# Alibaba Cloud

**Apsara Stack Enterprise** 

Security Whitepaper

Product Version: 2012, Internal: V3.13.0

Document Version: 20210621

(-) Alibaba Cloud

### Legal disclaimer

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

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

# **Document conventions**

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

# **Table of Contents**

1.Security of Apsara Stack products	15
1.1. Elastic Compute Service (ECS)	15
1.1.1. Platform security	15
1.1.1.1. Security isolation	15
1.1.1.2. Authentication	16
1.1.1.2.1. Identity authentication	16
1.1.1.2.2. Access control	16
1.1.1.3. Data security	17
1.1.1.3.1. Overview	17
1.1.1.3.2. Triplicate storage technology	17
1.1.1.3.3. ECS disk encryption	18
1.1.1.4. Encrypted transmission	19
1.1.1.5. ARP spoofing prevention	19
1.1.2. Tenant security	19
1.1.2.1. Log audit	19
1.1.2.2. Secure images	19
1.1.2.3. Block storage	19
1.2. Container Service for Kubernetes	20
1.2.1. Platform security	20
1.2.1.1. Security isolation	20
1.2.1.2. Account authentication	20
1.2.1.3. Link security	20
1.2.2. Tenant security	21
1.2.2.1. Application security	21
1.2.2.2. Host security	23
1.3. Auto Scaling (ESS)	23

1.3.1. Platform security	23
1.3.1.1. Security isolation	23
1.3.1.2. Authentication	24
1.3.1.2.1. Authentication	24
1.3.1.2.2. Access control	24
1.3.2. Tenant security	24
1.3.2.1. Log audit	24
1.4. Resource Orchestration Service (ROS)	25
1.4.1. Platform security	25
1.4.1.1. Data security	25
1.4.1.2. Authentication	25
1.4.2. Tenant security	25
1.4.2.1. Log audit	25
1.5. Object Storage Service (OSS)	25
1.5.1. Platform security	25
1.5.1.1. Security isolation	25
1.5.1.2. Authentication and access control	25
1.5.1.2.1. Authentication	25
1.5.1.2.2. Access control	26
1.5.1.2.3. Support for RAM and STS	27
1.5.1.3. Data security	27
1.5.1.4. Data encryption	27
1.5.1.4.1. Server-side encryption	27
1.5.1.4.2. Client-side encryption	27
1.5.2. Tenant security	28
1.5.2.1. Key management	28
1.5.2.2. Log audit	28
1.5.2.3. Hotlink protection	28

1.6. Apsara File Storage NAS	28
1.6.1. Platform security	28
1.6.1.1. Security isolation	28
1.6.1.2. Authentication	29
1.6.1.3. Data security	31
1.6.2. Tenant security	31
1.6.2.1. Log audit	31
1.6.2.2. Directory-level ACLs	31
1.7. Tablestore	33
1.7.1. Platform security	33
1.7.1.1. Security isolation	33
1.7.1.2. Authentication	33
1.7.1.3. Data security	34
1.7.2. Tenant security	34
1.7.2.1. Key management	34
1.8. ApsaraDB RDS	34
1.8.1. Platform security	34
1.8.1.1. Secure isolation	34
1.8.1.2. Authentication	34
1.8.1.3. Data security	35
1.8.1.4. Data encryption	35
1.8.1.5. DDoS attack prevention	36
1.8.2. Tenant security	36
1.8.2.1. Log audit	36
1.8.2.2. IP address whitelist	37
1.8.2.3. Software update	37
1.9. Cloud Native Distributed Database PolarDB-X	37
1.9.1. Platform security	37

1.9.1.1. Security isolation	37
1.9.1.2. Authentication	37
1.9.2. Tenant security	38
1.9.2.1. IP address whitelist	38
1.9.2.2. Protection against high-risk SQL operations	38
1.9.2.3. Slow SQL audit	39
1.9.2.4. Performance monitoring	39
1.10. AnalyticDB for PostgreSQL	39
1.10.1. Platform security	39
1.10.1.1. Security isolation	39
1.10.1.2. Authentication	40
1.10.1.3. Primary and secondary nodes	40
1.10.2. Tenant security	40
1.10.2.1. Database account	40
1.10.2.2. IP address whitelists	40
1.10.2.3. SQL audit	40
1.10.2.4. Backup and restoration	41
1.10.2.5. Software upgrade	41
1.11. KVStore for Redis	41
1.11.1. Platform security protections	41
1.11.1.1. Security isolation	41
1.11.1.2. Authentication	42
1.11.1.3. Transmission encryption	42
1.11.2. Tenant security protections	42
1.11.2.1. Database account	42
1.11.2.2. IP address whitelist	42
1.11.2.3. Backup and recovery	43
1.11.2.4. Software upgrade	43

1.12. ApsaraDB for MongoDB	43
1.12.1. Platform security	43
1.12.1.1. Isolation	
1.12.1.2. Authentication	43
1.12.1.3. Data security	44
1.12.1.4. Data encryption	45
1.12.1.5. Anti-DDoS	45
1.12.2. Tenant security	45
1.12.2.1. Log audit	45
1.12.2.2. IP address whitelists	45
1.13. ApsaraDB for OceanBase	45
1.13.1. Platform security	45
1.13.1.1. Security isolation	45
1.13.1.2. Authentication	46
1.13.1.3. High-availability architecture	46
1.13.1.4. Compatibility	46
1.13.2. Tenant security	47
1.13.2.1. Database accounts	47
1.13.2.2. IP address whitelists	47
1.13.2.3. Log audit	47
1.13.2.4. Software upgrades	47
1.13.3. Notes on MySQL-related vulnerabilities in ApsaraDB fo	48
1.14. Data Transmission Service (DTS)	49
1.14.1. Platform security	49
1.14.1.1. Security isolation	49
1.14.1.2. Authentication	49
1.14.1.3. Transmission security	49
1.14.1.4. Data security	49

1.15. Data Management (DMS)	49
1.15.1. Platform security	49
1.15.1.1. Security isolation	49
1.15.1.2. Authentication	50
1.15.1.3. Data security	50
1.15.1.4. Data encryption	51
1.15.1.5. Limits on changes	51
1.15.2. Tenant security	52
1.15.2.1. Log audit	52
1.16. Server Load Balancer (SLB)	52
1.16.1. Platform security	52
1.16.1.1. Authentication	52
1.16.2. Tenant security	52
1.16.2.1. HTTPS	52
1.16.2.2. IP address whitelists	53
1.16.2.3. Log management	53
1.17. Virtual Private Cloud (VPC)	53
1.17.1. Platform security	53
1.17.1.1. Security isolation	53
1.17.1.2. Access control	53
1.17.2. Tenant security	53
1.17.2.1. Security groups	53
1.18. Log Service	53
1.18.1. Platform security	54
1.18.1.1. Security isolation	54
1.18.1.2. Authentication	54
1.18.1.3. Data security	54
1.18.1.4. Encrypted data transmission	55

1.18.1.5. Encrypt data	55
1.18.2. Tenant security	
1.18.2.1. Service monitoring	56
1.19. Key Management Service (KMS)	
1.19.1. Platform security	
1.19.1.1. Security isolation	56
1.19.1.2. Authentication	56
1.19.1.2.1. Authentication	56
1.19.1.2.2. Access control	57
1.19.1.2.3. RAM and STS support	57
1.19.1.3. Data security	57
1.19.1.4. Transmission encryption	57
1.20. Apsara Stack DNS	58
1.20.1. Tenant security	58
1.20.1.1. Tenant isolation	58
1.20.1.2. Network security hardening	58
1.20.1.3. Log audit	58
1.21. API Gateway	58
1.21.1. Platform security	58
1.21.1.1. Security isolation	58
1.21.1.2. Authentication	58
1.21.1.2.1. Authentication	58
1.21.1.2.2. API access control	59
1.21.1.2.3. RAM and STS support	59
1.21.1.3. Data security	59
1.21.1.4. Transmission encryption	
1.21.2. Tenant security	
1.21.2.1. Log audit	

1.21.2.2. IP address-based access control	60
1.22. Enterprise Distributed Application Service (EDAS)	60
1.22.1. Platform security	60
1.22.1.1. Authentication	60
1.22.1.2. Transmission encryption	61
1.22.2. Tenant security	62
1.23. MaxCompute	62
1.23.1. Platform security	62
1.23.1.1. Security isolation	62
1.23.1.2. Authentication	64
1.23.1.3. Data security	67
1.23.1.4. KMS-based storage encryption	68
1.23.2. Tenant security	72
1.23.2.1. Cross-project resource sharing	72
1.23.2.2. Column-level access control	75
1.23.2.3. Project protection	76
1.23.2.4. Log audit	78
1.24. DataWorks	78
1.24.1. Permission isolation for development and production e	78
1.24.2. Authentication and authorization	79
1.24.2.1. Access control	79
1.24.2.2. Permission management	79
1.24.3. Data encryption	80
1.24.4. Sensitive data protection	80
1.25. Realtime Compute	80
1.25.1. Platform security	81
1.25.1.1. Resource isolation	81
1.25.1.2. Authentication and authorization	81

1.25.1.3. Data security	81
1.25.1.4. Business process	81
1.26. Machine Learning Platform for AI	82
1.26.1. Security isolation	82
1.26.2. Authentication	83
1.26.2.1. Identity verification	83
1.26.2.2. Permission control	84
1.26.2.3. RAM and STS	84
1.26.3. Data security	85
1.26.4. Log audit	86
1.27. E-MapReduce (EMR)	86
1.27.1. Platform security	86
1.27.1.1. Security isolation	86
1.27.1.2. Authentication	86
1.27.1.3. Data security	89
1.27.2. Access control	89
1.28. DataHub	90
1.28.1. Platform security	90
1.28.1.1. Data isolation	90
1.28.1.2. Authentication	90
1.28.1.3. Data security	92
1.28.2. Tenant security	92
1.28.2.1. Log auditing	92
1.29. Quick BI	93
1.29.1. Tenant isolation	93
1.29.2. Authentication and authorization	93
1.29.2.1. Identity authentication	93
1.29.2.2. Permission management	

1.29.2.3. RAM authentication	94
1.29.3. Data security	94
1.29.4. Data encryption	94
1.29.5. Audit logs	94
1.29.6. Security best practices	95
1.30. Graph Analytics	95
1.30.1. Platform security	95
1.30.1.1. Security isolation	96
1.30.1.2. Authentication	96
1.30.1.3. Data security	96
1.30.1.4. Transmission encryption	96
1.30.1.5. System security	96
1.30.1.5.1. Vulnerability scanning mechanism	96
1.30.1.5.2. Update scheme for security vulnerabilities	96
1.30.1.5.3. System defense mechanism	97
1.30.1.6. Infrastructure security	97
1.30.1.7. Level-based data security	97
1.30.2. Tenant security	97
1.30.2.1. Log audit	97
1.31. Elasticsearch (on k8s)	97
1.31.1. Security isolation	97
1.31.2. Authentication and authorization	97
1.31.2.1. Identity authentication	97
1.31.2.2. Access control	98
1.31.3. Data security	98
1.31.4. Transmission encryption	98
1.32. Elasticsearch (on ECS)	98
1.32.1. Security isolation	98

1.32.2. Authentication and authorization	98
1.32.2.1. Identity authentication	98
1.32.2.2. Access control	98
1.32.3. Data security	98
1.32.4. Transmission encryption	00

# 1. Security of Apsara Stack products

# 1.1. Elastic Compute Service (ECS)

# 1.1.1. Platform security

# 1.1.1.1. Security isolation

Instance security isolation includes the following aspects:

#### **CPU** isolation

ECS supports the KVM hypervisor. Using VT-x virtualization, the hypervisor runs in vmx root mode while ECS instances run in vmx non-root mode. Hardware isolation prevents ECS instances from accessing the privileged resources of other instances.

#### Memory isolation

The hypervisor isolates memory on the virtualization layer. When ECS instances are running, extended page tables (EPT) ensure that ECS instances cannot access the memory resources of other instances.

After an ECS instance is released, all of its memory is cleared by the hypervisor to prevent other ECS instances from accessing the physical memory released by this ECS instance.

#### Storage isolation

On the virtualization layer, the hypervisor uses a device driver model to achieve I/O virtualization. ECS instances cannot directly access physical disks, and all I/O operations are intercepted and processed by the hypervisor. The hypervisor ensures that ECS instances can only access the allocated virtual disk space, thus realizing the security isolation of disk space between different ECS instances.

#### Network isolation

ECS uses Virtual Switches (VSwitches). Messages destined for an ECS instance are only sent to the VSwitch port corresponding to the virtual network interface of the ECS instance.

ECS instances, even those running in hybrid mode, are not able to receive or intercept messages intended for other ECS instances. Even if you set the network interface to the hybrid mode, the hypervisor does not transmit any traffic destined for an instance to any other instances.

Alibaba Cloud further uses Virtual Private Networks (VPCs) and security groups to isolate networks.

A security group is an additional security barrier for ECS instances provided by Alibaba Cloud. It implements a distributed virtual firewall with stateful packet inspection. A security group is independent of the operating system firewalls on the ECS instances within the group. The security group provides additional protection from outside of ECS instances. Security groups allow you to implement isolated security domains by configuring inbound or outbound policies for a single IP address or port.

A security group is a logical group that consists of a group of instances in the same region that share security requirements and have mutual access permissions. Security groups are used for network access control for one or more ECS instances and allow you to divide a cloud into separate security domains.

With the preceding isolation measures, even if two instances owned by the same user run on the same physical server, the two instances are unable to intercept each other's traffic.

In addition, we recommend that you encrypt data before saving it to ECS instance disks with either an encrypted file system or disk encryption. For more information, see ECS disk encryption.

#### 1.1.1.2. Authentication

### 1.1.1.2.1. Identity authentication

Account authentication uses identity credentials to verify the identities of users. An identity credential typically refers to a logon password or an AccessKey pair. You can obtain your AccessKey pair from the User Information page in the Apsara Uni-manager Management Console. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey ID is public and used to identify users. The AccessKey secret is a secret key used to encrypt the signature strings and verify the signature strings on a server. It is used to authenticate users and must be kept confidential.

ECS performs identity verification on each access request. Therefore, both HTTP and HTTPS requests must contain signature information. ECS performs symmetric encryption to verify the identity of the request sender by using the AccessKey ID and AccessKey secret.

#### 1.1.1.2.2. Access control

Resource Access Management (RAM) is a centralized user management and resource access control service provided by Apsara Stack. You can use RAM to create independent user accounts for your employees, systems, or applications and control their access to your cloud resources. Each RAM user can log on to the Apsara Uni-manager Management Console or call service APIs by using an independent logon password or AccessKey pair. By default, RAM users do not have any permissions on resources when they are created. Only an authorized RAM user can operate resources on behalf of the corresponding Apsara Stacktenant account.

You can use RAM to grant permissions based on the least privilege principle to reduce data security risks and avoid sharing your AccessKey pairs with other users. RAM enables an Alibaba Cloud account to have multiple independent RAM users. RAM also supports features such as multi-factor authentication (MFA), strong password policies, separation of console users from API users, custom fine-grained authorization policies, grouped authorization, and temporary authorization tokens, and temporary account disabling. RAM can be used to define fine-grained authorization at the API operation or resource ID level. RAM also supports various restrictive conditions on permission granting, such as constraints on source IP address, required SSL/TLS channel, access time period, and MFA.

We strongly recommend that you access and manage the operating systems in your ECS instances in a secure manner. You can use the SSH public key and private key pairs and properly maintain the private key. You must at least use a complex password, which can be set when you create the instance. In addition, you must use SSHv2 for remote logons and use the sudo commands to temporarily escalate permissions.

You can create RAM users and groups to manage access to cloud resources.

RAM can help you configure fine-grained access control on cloud resources. For example, you can enhance network security by attaching a policy to a RAM user group. If an access request does not originate from a corporate internal network specified in the policy, the access request is rejected.

Different permissions can be granted to RAM user groups to manage ECS resources. The following section describes two example groups.

• SysAdmins: This group requires permissions to create and manage ECS images, instances, snapshots, and security groups. A policy with full ECS management permissions can be attached to the SysAdmins group.

• Developers: This group requires permissions to use ECS instances. A policy that authorizes group members to call the Describelnstances, StartInstance, StopInstance, CreateInstance, and DeleteInstance API operations can be attached to the Developers group.

If a developer becomes a system administrator, you can move the developer account from the Developers group to the SysAdmins group.

ECS allows you to attach RAM roles to ECS instances to access cloud resources by using temporary Security Token Service (STS) credentials. Instance RAM roles allow ECS instances to assume RAM roles with specific access permissions.

Each RAM role can be assigned to an ECS instance, which can then access other cloud resources by using a temporary STS credential. This ensures the security of AccessKey pairs and allows you to implement fine-grained access control.

### 1.1.1.3. Data security

#### 1.1.1.3.1. Overview

Key Management Service (KMS) provides key management and encryption mechanisms for the storage of encrypted data necessary on the cloud platform. The encrypted data includes authorization credentials, passwords, and keys.

### 1.1.1.3.2. Triplicate storage technology

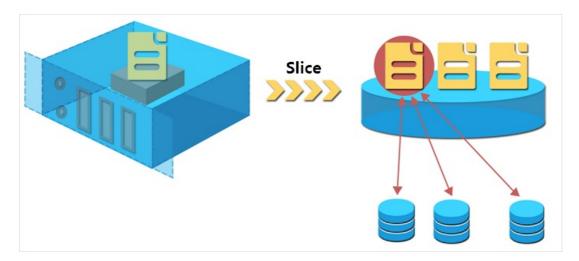
When ECS users read or write data from or to virtual disks, the operations are translated into reads or writes from or to the files stored in the Apsara Stack data system. Apsara Stack uses a flat design in which a linear address space is divided into slices, also called chunks. For each chunk, three replicas are created and stored on different nodes in the cluster to ensure the reliability of data.

Triplicate technology involves three key components: master, chunk server, and client. The write operation of an ECS user goes through several conversions, as shown in the following procedure:

- 1. The client determines the chunk corresponding to the write operation.
- 2. The client sends a request to the master to query the chunk servers on which to store the three chunk replicas.
- 3. The client sends a write request to the three corresponding chunk servers based on the results returned from the master.
- 4. If the write operation succeeds on all three chunk replicas, the client returns a success message to the user. Otherwise, the client returns a failure message.

The call distribution policy of the master takes into account the disk usage of all chunk servers in the cluster, the chunk server distribution on different switch racks, power supply, and machine load. The policy ensures that the three replicas of each chunk are distributed across different chunk servers on different switch racks. This effectively avoids data unavailability caused by the failure of chunk servers or racks where data is stored.

Triplicate backup



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

#### Automatic replication



To summarize, all your (add, modify, and delete) operations on cloud disk data are synchronized to the three chunk copies at the bottom layer. This approach ensures that user data stays reliable and consistent.

When data is deleted, Apsara Distributed File System reclaims the released storage space and makes it inaccessible to any users. Data in this storage space is erased completely before it is made available for other users. This procedure ensures that user data remains secure and confidential.

# 1.1.1.3.3. ECS disk encryption

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

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

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

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

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

### 1.1.1.4. Encrypted transmission

Apsara Stack uses the HTTPS protocol to ensure data transmission security. The Apsara Uni-manager Management Console uses HTTPS encryption for data transmission. All Apsara Stack services provide API access points that have HTTPS encryption enabled and allow users to use AccessKey pairs to call Apsara Stack service APIs. Apsara Stack uses standard SSL/TLS protocols for transmission and provides keys up to 256 bits in length to ensure that sensitive data can be encrypted and transmitted securely.

### 1.1.1.5. ARP spoofing prevention

ARP spoofing severely challenges traditional networks. Hackers can use ARP spoofing to imitate the routing address of another user and intercept confidential data.

To prevent ARP spoofing, Alibaba Cloud provides an ARP firewall at the Network Egress. Only MAC addresses that are allocated by the platform are authorized for communication.

## 1.1.2. Tenant security

### 1.1.2.1. Log audit

User authentication credentials and permission control are designed to avoid security issues. Security logs can help Alibaba Cloud users better understand and diagnose a variety of security situations. Alibaba Cloud provides centralized security log management for cloud resources operations. The logon and resource access operations of each account are logged, providing information about the operator, operation time, source IP address, resource object, operation name, and operation status. With all operation records saved, users can perform security analysis, intrusion detection, resource change tracking, and compliance audit. In a compliance audit, users may need to provide detailed operation records of the Apsara Stacktenant accounts and RAM users.

### 1.1.2.2. Secure images

Apsara Stack images integrate patches for all known high-risk vulnerabilities to prevent the host from being exposed to high risks after going online. After detecting a new high-risk vulnerability, Alibaba Cloud promptly updates images and delivers the updated images to customers. Besides, Alibaba Cloud also ensures image integrity and avoid malicious tempering by using a data verification algorithm.

Users can quickly upgrade their basic images after new high-risk vulnerabilities are detected. Moreover, users can upgrade the operating system or fix vulnerabilities of their ECS instances by themselves.

Alibaba Cloud strongly recommends that users employ Apsara Stack basic images as the first step to implement cloud migration without affecting business deployment.

### 1.1.2.3. Block storage

Block Storage is a low-latency, persistent, and high-reliability random block-level data storage service provided by Alibaba Cloud for ECS instances. Block storage automatically replicates data within a zone to prevent unavailability caused by unexpected hardware faults and to protect your business. Just like a physical hard disk, you can format block storage attached to an ECS instance, create a file system, and persistently store data there.

Block Storage automatically encrypts block storage devices used inside of virtual machines to ensure data is secure and stored in a distributed system.

### 1.2. Container Service for Kubernetes

# 1.2.1. Platform security

### 1.2.1.1. Security isolation

Container Service provides multiple security isolation methods to ensure cluster security.

#### **Exclusive Kubernetes clusters**

Kubernetes clusters you created through the Container Service console belong to you. Resources for deploying Kubernetes clusters, such as ECS and SLB instances, can be used only by the current Kubernetes clusters and are not shared with other users. Strong isolation at the physical level can avoid potential security risks arising from resource sharing.

#### ECS security group

The ECS instance used by each Kubernetes cluster belongs to the same ECS security group. Based on the least privilege principle, a security group contains only the following network access rules:

- Allow access to ECS instances over ICMP
- Allow access to ECS instances through pod CIDR blocks

#### Container network policies

In a Kubernetes cluster, pods on different nodes can communicate with each other by default. In some scenarios, the network intercommunication between different businesses is not allowed, and network policies must be introduced to reduce risks. In Kubernetes clusters, you can use the Canal network driver to implement the support for network policies.

#### 1.2.1.2. Account authentication

You can use the Container Service account authentication function to secure your containerized applications.

RAM user authorization is supported for Kubernetes clusters. You can perform RAM authorization to grant permissions on Kubernetes clusters to specific RAM users. This reduces the risk of exposing Apsara Stack tenant account data.

### Role-Based Access Control (RBAC)

RBAC uses the Kubernetes built-in API group for authentication management, allowing you to manage pods corresponding to different roles and role access permissions.

### 1.2.1.3. Link security

Container Service supports TLS certificate verification for link security.

The following communication links in Container Service Kubernetes clusters are verified by TLS certificates to prevent data tampering and eavesdropping on communications:

- kubelet on worker nodes actively communicates with apiserver on master nodes.
- apiserver on master nodes actively communicates with kubelet on worker nodes.

During initialization, a master node uses SSH tunnels to connect to the SSH service of other nodes (port 22).

# 1.2.2. Tenant security

# 1.2.2.1. Application security

Container Service for Kubernetes supports a wide range of application security policies.

#### Application security policies

Security policy	Description
Run a container as a non-root user	You can run applications in a container as a non-root user. This avoids container escapes and prevents attackers from obtaining host permissions.
Use secure base images	You can customize base images as required and enforce the use of approved base images within your organization. You can also use secure third-party images. We recommend that you use base images based on Alpine Linux. All Docker images are based on Alpine Linux.
Include minimal resources in images	Only resources necessary to run applications are included in images.
Configure Transport Layer Security (TLS) authentication for Docker daemons	TLS authentication is configured for Docker daemons and Docker Swarm API operations.
Prioritize CPU usage for containers	You can use the CPU sharing feature of Docker to prioritize CPU usage for containers. The feature allows a container to have priority over another for CPU usage and avoids high CPU usage by containers with a lower priority. This ensures expected performance of high-priority containers and prevents resource exhaustion attacks.
Limit container memory usage	By default, a container can consume all available memory resources on the Docker host. You can use the memory limit mechanism to prevent denial-of-service (DoS) attacks that may arise if a single container consumes all host resources. Specifically, you can add the -m or -memory option to the docker run command to run containers.

Security policy	Description
Limit container disk usage	By default, Docker images, container rootfs, and volumes are stored in the /var/lib/docker directory. They share the same file system with the host. The directory size varies depending on the content. You can mount the /var/lib/docker directory to the cloud storage, such as disks and Object Storage Service (OSS) buckets. This minimizes the impact on the root file system of the host.
Implement identity authentication	<ul> <li>Username-password pairs and certificates are used for user authentication.</li> <li>A logon failure processing mechanism can be used to limit the number of unauthorized logon attempts. If the number of failed logon attempts within a session exceeds the specified threshold, the session terminates.</li> <li>A session is automatically locked after the session stays in the idle state for a specified period of time.</li> <li>You must change the initial password when you log on to the system for the first time.</li> <li>The password must meet requirements. The new password cannot be the same as the last password. You are periodically prompted to change your password.</li> </ul>
Implement the security audit feature	<ul> <li>The system supports the account security audit feature. You can use this feature to create audit logs for system account modifications. These logs cannot be modified.</li> <li>All operations on the system platform are recorded in clear log entries, including the event date, time, initiator information, type, description, and result. Audit logs are periodically backed up and retained for at least six months.</li> </ul>
Ensure communication security and confidentiality	<ul> <li>TLS encryption is used in communication between system components.</li> <li>Confidential data is encrypted in the system. The data is sent to specific nodes only when required.</li> </ul>

Security policy	Description
	<ul> <li>The system uses a built-in multi-tenant permission management model. The model allows you to set different permissions based on teams and roles and manage fine-grained permissions on clusters and applications.</li> </ul>
Manage roles and permissions	<ul> <li>You can manage access to cluster resources for external account systems by using methods such as Lightweight Directory Access Protocol (LDAP).</li> </ul>
	<ul> <li>Each image repository supports access control based on multiple permissions, such as multi- tenant and read-only permissions.</li> </ul>

### 1.2.2.2. Host security

#### OS account requirements

An OS password must be at least 8 characters in length, and must contain at least three types of the following characters: uppercase letters, lowercase letters, numbers, and special characters. Weak passwords (such as regular or consecutive characters, employee IDs, and domain account prefixes) are not allowed. Your OS password is set to expire every 90 days by default.

#### A mechanism to limit unauthorized logon attempts

A logon failure processing mechanism can be used to limit the number of unauthorized logon attempts. When the number of failed logon attempts exceeds the specified threshold, the session terminates and exits.

#### Access control

Access control can be used to control user access to resources based on configured security policies.

• passwd file permission: 644

• shadow file permission: 000

• rc3.d file permission: 755

• profile file permission: 644

• profile.d folder permission: 755

#### Disabling and deletion of default accounts

You can delete redundant and expired accounts to prevent accounts from being shared. You can disable the following default accounts: sync, shutdown, and halt.

# 1.3. Auto Scaling (ESS)

# 1.3.1. Platform security

### 1.3.1.1. Security isolation

ESS implements user account-based isolation. You can manage scaling groups, configurations, and rules in your account, such as performing create, modify, and delete operations. ESS can use ECS instance resources only in your account for automatic scaling. ESS performs symmetric encryption by using AccessKey pairs to authenticate users who manage ECS instance resources. It authenticates each access request to ensure security isolation.

#### 1.3.1.2. Authentication

#### 1.3.1.2.1. Authentication

You can create an AccessKey pair in the Apsara Uni-manager Management Console. An AccessKey pair is composed of an AccessKey ID and an AccessKey secret. The AccessKey ID is a public key that is used to identify a user. The AccessKey secret is a key used to encrypt signature strings and verify those signature strings on the server. The AccessKey secret is used to authenticate the identities of users and must be kept confidential.

Auto Scaling authenticates each access request. Therefore, each request must contain signature information, regardless of whether it is sent over HTTP or HTTPS. Auto Scaling implements symmetric encryption to authenticate the identity of a request sender by using AccessKey pairs.

The AccessKey ID and AccessKey secret are issued by Alibaba Cloud to users. You can request and manage them on the official Alibaba Cloud website. The AccessKey ID indicates the identity of a user. The AccessKey secret is a key used to encrypt signature strings and verify those on the server. The AccessKey secret must be kept confidential.

#### 1.3.1.2.2. Access control

Resource Access Management (RAM) is a service that Alibaba Cloud provides to you to manage user identities and to control resource access. You can use RAM to create and manage user accounts, such as employee accounts, system accounts, and application accounts. You can also manage the operation permissions that these user accounts have on resources of your account. If multiple users in your enterprise operate resources collaboratively, RAM allows you to grant permissions to other users without sharing the AccessKey pair of your Apsara Stacktenant account with other users. Instead, you can grant users the minimum permissions necessary for them to complete their work. Thus, security risks to your enterprise information are reduced.

RAM allows you to create different roles and assign different permissions on cloud services to each role. Auto Scaling allows you to configure the RamRoleName parameter. You can configure this parameter to assign different roles to your ECS instances, allowing different instances to have permissions on different cloud services. Before configuring the RamRoleName parameter in Auto Scaling, you must ensure that the current RAM permission policy allows your ECS instance to act as the specified role. Otherwise, the scaling configuration cannot make the ECS instance available.

### 1.3.2. Tenant security

### 1.3.2.1. Log audit

ESS generates scaling activity logs that record information about each scaling activity, such as activity ID, status, status information, start time, end time, reason for activity, and details.

The states of a scaling activity include Rejected, In Progress, Successful, Warning, and Failed. Status information includes the status details. Reason for activity includes the results of scaling activities executed in a scaling group. Details include information about instances involved in a scaling activity.

# 1.4. Resource Orchestration Service (ROS)

# 1.4.1. Platform security

### 1.4.1.1. Data security

None

#### 1.4.1.2. Authentication

ROS supports RAM authentication.

Resource Access Management (RAM) is a resource access control service provided by Alibaba Cloud. You can use an Apsara Stack tenant account to create RAM users and grant them permissions on resources that belong to the tenant account.

# 1.4.2. Tenant security

### 1.4.2.1. Log audit

ROS displays detailed information about historical events, including event logs for stacks. The information includes the resource name, associated resource ID, resource type, resource status, status description, and event occurrence time. Event logs provide the information about changes to stacks.

# 1.5. Object Storage Service (OSS)

# 1.5.1. Platform security

### 1.5.1.1. Security isolation

OSS slices user data and discretely stores the sliced data in a distributed file system based on specific rules. The user data and its indexes are stored separately. OSS uses symmetric AccessKey pairs to authenticate users and verifies the signature in each HTTP request sent by users. If verification is successful, OSS reassembles the distributed data. This way, OSS implements data storage isolation between multiple tenants.

### 1.5.1.2. Authentication and access control

### 1.5.1.2.1. Authentication

You can create an AccessKey pair on Apsara Stack Management Console. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey ID is a public ID that uniquely identifies a user. The AccessKey secret is private and used to authenticate a user.

Before you send a request, you must generate a signature string for the request in the format specified by OSS. Then, you must encrypt the signature string by using your AccessKey secret to generate a verification code based on the HMAC algorithm. The verification code is timestamped to prevent replay attacks. After receiving the request, OSS finds the AccessKey secret based on your AccessKey ID, and uses the AccessKey secret to decrypt the signature string and verification code. Then, OSS calculates a verification code and compares it with the decrypted verification code. If the two verification codes are the same, OSS determines that the request is valid. Otherwise, OSS rejects the request and returns HTTP 403.

#### 1.5.1.2.2. Access control

OSS supports access control list (ACL) to control access permissions. An ACL is set based on resources. You can specify ACLs for buckets or objects. You can specify an ACL for a bucket when you create the bucket or for an object when you upload the object to OSS. You can also modify the ACLs of uploaded objects and created buckets.

Access to OSS resources can be initiated by the bucket owner or third party users. An owner owns a bucket. Third-party users are other users who access resources in the bucket. Access can be either anonymous or signed. If the access is initiated with an OSS request that does not contain identification information, the access is considered to be anonymous. A signed access is a request that contains signature information in the header or a URL that contains signature information as defined in OSS API documentation.

OSS provides access control for buckets and objects.

You can configure one of the following ACLs for a bucket:

- Public read/write: All users (including anonymous users) can perform write (Put Object, Get Object, and DeleteObject) operations on objects in the bucket.
- Public read: Only the bucket owner or authorized users can perform write operations (PutObject and DeleteObject) on objects in the bucket. Other users, including anonymous users, can only perform read operations (GetObject) from the objects in the bucket.
- Only the bucket owner or authorized users can perform read and write operations (PutObject, GetObject, and DeleteObject) on objects in the bucket. Other users cannot access the objects in the bucket without authorization.

**Note** If you do not configure the ACL of a bucket when you create the bucket, OSS sets the ACL of the bucket to private.

You can configure one of the following ACLs for an object:

- Public read/write: All users can perform read/write operations on the object.
- Public read: Only the object owner can perform read/write operations on the object. Others can perform read operations on the object.
- Private: Only the object owner can perform read/write operations on the object. Others cannot access the object.
- Default: The object inherits the ACL of the bucket.

**Note** If you do not configure the ACL of an object when you upload the object, OSS sets the ACL of the object to default.

### 1.5.1.2.3. Support for RAM and STS

OSS supports Resource Access Management (RAM) and Security Token Service (STS) authentication.

RAM is a resource access control service provided by Apsara Stack. RAM allows you to create RAM users under an Apsara Stack tenant account. The Apsara Stack tenant account can grant access permissions on resources to RAM users.

STS is service that provides temporary access credentials. You can use STS to generate a temporary access credential for a user and specify the permission and validity period of the credential. A credential becomes invalid after it expires.

### 1.5.1.3. Data security

An error may occur when data is transferred between the client and server. OSS supports CRC and MD5 verification to secure data.

#### CRC

OSS can return the CRC64 value of objects uploaded through any of the methods provided. The client can compare the CRC64 value with the locally calculated value to verify data integrity.

OSS calculates the CRC64 value for newly uploaded objects and stores the result as metadata of the object. OSS then adds the x-oss-hash-crc64ecma header to the returned response header, indicating its CRC64 value. This CRC64 value is calculated based on Standard ECMA-182.

#### MD5 verification

To check whether the object uploaded to OSS is consistent with the local file, attach the Content-MD5 field value to the upload request. The OSS server verifies the MD5 value. The upload can succeed only when the MD5 value of the object received by the OSS server is the same as the Content-MD5 field value. This method can ensure the consistency between objects.

### 1.5.1.4. Data encryption

### 1.5.1.4.1. Server-side encryption

OSS supports server-side encryption for uploaded data. When you upload data, OSS encrypts the data by using AES256 and permanently stores the encrypted data. When you download the data, OSS automatically decrypts the data, returns the original data, and declares in the header of the returned HTTP request that the data had been encrypted on the server.

To encrypt an object on the OSS server when you upload the object, you only need to add the x-oss-server-side-encryption header in the Put Object request and set its value to AES256.

# 1.5.1.4.2. Client-side encryption

OSS allows you to use client-side encryption to encrypt data before the data is sent to the server while the data encryption key (DEK) used is kept only on the local client. Other users cannot obtain the raw data without the DEK and enveloped data key (EDK), even if the data is leaked. OSS uses functions provided by SDKs to encrypt the data on local clients before the data is uploaded to the OSS bucket.

# 1.5.2. Tenant security

### 1.5.2.1. Key management

Apsara Stack Key Management Service (KMS) is a secure and highly available service that integrates hardware and software, and provides a key management system that can be extended to the cloud. KMS uses customer master keys (CMKs) to encrypt OSS objects and uses KMS API operations to generate data encryption keys (DEKs) in a centralized manner. You can define policies in KMS to control and monitor key usage. You can use these keys to protect data in OSS buckets.

### 1.5.2.2. Log audit

OSS automatically saves access logs. After access logging is enabled for a source bucket, OSS generates an object that contains access logs for that bucket (by hour), names the object based on predefined naming rules, and writes the object into the bucket specified by the user. These logs are used for later auditing and behavior analysis. Request logs contain information such as the request time, source IP address, request object, return code, and processing duration.

### 1.5.2.3. Hotlink protection

To prevent additional fees caused by unauthorized access to the resources in your bucket, you can configure hot link protection for your buckets on the Apsara Uni-manager Management Console or by using API operations.

You can set the following parameters to configure hotlink protection:

- Referer Whitelist: Only specified domain names are allowed to access OSS resources.
- Allow Empty Referer: If this parameter is set to disabled, a request is allowed to access OSS resources only if the request contains the Referer field configured in the HTTP or HTTPS header.

For example, if you add http://www.aliyun.com/ to the Referer whitelist of a bucket named oss-example, requests in which the Referer field is http://www.aliyun.com/ can access the objects in the bucket.

# 1.6. Apsara File Storage NAS

# 1.6.1. Platform security

### 1.6.1.1. Security isolation

#### **Network** isolation

NAS provides the permission group mechanism to control the networks over which NAS instances can be accessed. 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. In this way, networks are isolated from each other.

#### Storage isolation

In NAS, each mount point instance of a file system is mapped to a storage unit in the server storage pool. The storage units corresponding to different mount point instances are isolated from each other.

The access control module on the NAS server verifies the I/O requests of users based on the mapping between VPCs and NAS mount point instances. The module checks the storage unit information carried by each request against the storage unit information on the server for consistency. This ensures storage isolation on the server.

#### 1.6.1.2. Authentication

#### Permission control

NAS allows you to perform standard directory or file permission operations on a NAS instance. You can also configure read, write, or execution permissions for a user or user group. NAS supports two types of mount points: VPC and classic network. If you configure a VPC mount point for a NAS instance, only the ECS instances within the same VPC as the mount point can access the NAS instance. If you configure a classic network mount point, only the ECS instances under the same account as the mount point can access the NAS instance.

In NAS, the permission group acts as an IP address whitelist. You can add rules to the permission group of a NAS instance to allow users from specified IP addresses or address segments to access the NAS instance with different permissions.

VPC Default Permission Group is automatically generated for each account by default. This permission group allows all IP addresses in the VPC to access the mount point with full permissions. Full permissions include read and write permissions with no restrictions on the root user.

### ? Note

- Mount points in a classic network do not have a default permission group.
- When you add a permission group rule for a mount point in a classic network, you can only set the authorized IP address to a single IP address. You cannot set the authorized IP address to an IP address segment.

A permission group rule has four attributes, as listed in the following table.

#### Permission group rule attributes

Attribute	Value	Definition
Authorized IP Address	An IP address or IP address segment (You must specify an IP address for a permission group rule of a mount point in a classic network.)	The IP address or IP address segment of the authorized object.
Read and Write Permission	<ul><li>Read Only</li><li>Read/Write</li></ul>	The operation permissions of the authorized object on the NAS instance.

Attribute	Value	Definition
<ul> <li>Do Not Limit root User</li> <li>Limit root User</li> <li>Limit All Users</li> </ul>	Whether to restrict the permissions of the authorized object's Linux system users on the NAS instance.	
	Do Not Limit root User	Description:
	<ul> <li>Do Not Limit root User allows the root user to access the NAS instance.</li> </ul>	
	- Limit Alt Osers	<ul> <li>Limit root User considers the root user as nobody.</li> </ul>
		Limit All Users considers all users including root as nobody.
Priority	Valid values: 1 to 100. 1 indicates the highest priority.	When an authorized object matches multiple rules, the rule with the highest priority takes effect.

### Access control

30

NAS can work with RAM. You can make RAM settings in the NAS console to complete RAM authorization.

RAM allows you to grant the permissions on NAS instances to RAM users.

NAS operation permissions that can be granted to RAM users

Action	Description
DescriptFileSystems	Lists NAS instances.
DescriptMountTargets	Lists the mount points of a NAS instance.
DescriptAccessGroup	Lists the permission groups of a NAS instance.
DescriptAccessRule	Lists permission group rules.
CreateFileSystem	Creates a NAS instance.
CreateMountTarget	Adds a mount point to a NAS instance.
CreateAccessGroup	Creates a permission group.
CreateAccessRule	Adds a permission group rule.
DeleteFileSystem	Deletes a NAS instance.
DeleteMountTarget	Deletes a mount point.
DeleteAccessGroup	Deletes a permission group.
DeleteAccessRule	Deletes a permission group rule.
ModifyMountT argetStatus	Disables or enables a mount point.

Action	Description
ModifyMountT arget Access Group	Modifies the permission group of a mount point.
ModifyAccessGroup	Modifies a permission group.
ModifyAccessRule	Modifies a permission group rule.

### 1.6.1.3. Data security

#### Multi-copy data storage

NAS maintains multiple data copies to ensure data security.

User data: The NAS server stores three copies of user data. Services can continue running even when two copies are lost. The server monitors the number of copies in real time. When a data node is corrupted or a hard drive on a data node fails, the number of valid copies of some data in the cluster becomes less than 3. In this case, the server activates the replication mechanism to replicate data. This mechanism ensures that there are always three valid copies of each piece of data in the cluster.

The server prevents accidental silent errors by verifying that the stored data matches the check data. When the server detects a silent error, it replicates healthy copies to ensure the availability of three valid copies. This improves data reliability.

#### Data reclamation

The server reclaims the storage space that is released by the Delete operation. This reclaimed storage space is inaccessible to all users. The data stored in the storage space is erased before the space is reused. This fully guarantees data security.

## 1.6.2. Tenant security

## 1.6.2.1. Log audit

The NAS management system logs NAS instance operations, including creating and deleting NAS instances.

NAS logs are generated in real time and automatically stored on the server. A log contains detailed information about an operation, such as the executor and execution time. This information can be used for failure investigation and analysis.

### 1.6.2.2. Directory-level ACLs

This topic describes how to configure directory-level access control lists (ACLs). Only directory-level ACLs are available for Apsara File Storage NAS file systems.

#### **Prerequisites**

- You must mount NFSv4 file systems on all clients.
- You must use the alinas-acl tool to configure ACLs. We recommend that you do not change the file mode creation mask or use commands such as chmod to change file permissions. Otherwise, you may not obtain the expected results.

#### **Procedure**

The syntax of the mount command is sudo mount -t nfs -o vers=4.0 <the domain name of a mount target>:<the directory of an NAS file system> <a local directory>. For example, you can use mount -t nfs -o vers=4.0 014544bbf6-wdt41.cn-hangzhou.nas.aliyuncs.com://mnt to mount an NFSv4 file system.



- The value of the vers parameter changes based on the client version. If an error occurs when you set vers to 4.0, set vers to 4 instead.
- In some cases, ACLs are not enabled for a file system by default. To ensure ACLs are available for the file system, you can mount the file system again to enable ACLs.
- 2. Install the nfs4-acl-tools tool in CentOS.

sudo yum -y install nfs4-acl-tools

3. Make sure that Python 2.7 is installed.

python --version Python 2.7.5

4. Use the alinas-acl tool to configure an ACL.

./alinas\_acl set ./foo --add --user Alice --rule r #Grant the Alice user the read-only access to the foo file. ./alinas\_acl set ./foo -a -u Alice -r r #You can use the command to perform the same operation as the pre ceding command.

./alinas\_acl set ./dir --add --group Staff --rule rwx #Grant the Staff group the read, write, and execute acc ess to the dir directory.

./alinas\_acl set ./foo --add --user EVERYONE@ --rule none #Grant the EVERYONE@ principal no access to the foo file.

 $./a linas\_acl\,set\,./foo--add\,--user\,1001\,--rule\,none\,\#Grant\,the\,1001\,user\,principal\,no\,access\,to\,the\,foo\,file\,10$ 

./alinas\_acl set ./dir -d -u Bob #Revoke the Bob user access to the dir directory.

### ? Note

- To avoid a decrease in performance, we recommend that you configure an ACL for a directory rather than each file in the directory.
- We recommend that you add a maximum of 10 access control entries (ACEs) to an ACL.
- 5. View the ACL.

./alinas\_acl get ./foo #View the permissions on the foo file. # file: foo/ # owner: root # group: root OWNER@::rw- GROUP@::r-- EVERYONE@::--- Alice::r-- Staff:g:rwx 1001::--- Note When you configure an ACL, three special principals named OWNER@, GROUP@, and EVERYONE@ are automatically generated. These principals correspond to the user, group, and others classes of a file mode creation mask, respectively. The permissions that you specify for the file mode creation mask can be different from the permissions that you specify for an ACL. The actual permissions change based on the client version.

# 1.7. Tablestore

# 1.7.1. Platform security

### 1.7.1.1. Security isolation

This topic describes the security isolation methods of Tablestore, including network and storage isolation.

#### **Network** isolation

Tablestore supports instance-level VPC access control. The following types of VPC access settings are supported:

- Allows all network access: Access from the Internet and VPCs bound to the instance is allowed.
- Allows access from specific VPCs: Only access from VPCs bound to the instance is allowed.
- Allows access from the console and specific VPCs: Only access from VPCs bound to the instance and the Tablestore console is allowed. Access from other sources is denied.

#### Storage isolation

Tablestore uses a shared storage mechanism. This mechanism allows the instances of different users to share the same cluster resource. Tablestore uses data partitions as the smallest unit and supports the load balancing mechanism at the data partition level to isolate the impact between different instances.

#### 1.7.1.2. Authentication

This topic describes the authentication methods of Tablestore, including authentication and access control.

#### Authentication

Tablestore authenticates requests based on AccessKey pairs. Each valid Tablestore request must contain the correct AccessKey pair information. Tablestore authenticates each request from applications to prevent unauthorized data access and ensure data security.

#### Access control

Tablestore supports Security Token Service (STS), which allows you to control access of RAM users. STS is a temporary access credential service provided by Apsara Stack. It provides temporary access control. You can use STS to generate a temporary access credential. You can specify the permissions and validity period of the credential. The credential becomes invalid when it expires.

Tablestore supports authorization based on tables and API operations.

### 1.7.1.3. Data security

This topic describes the data security policies of Tablestore.

Tablestore is built on the Apsara Distributed File System and provides linear storage space. Linear addresses are sliced into chunks. For each chunk, three replicas are created and stored in different nodes in the cluster to ensure data reliability.

In Tablestore, data is serialized before it is written to the disks. Each data block is written to one or more chunks.

Apsara Distributed File System evaluates the disk usage of all nodes, the distribution of these nodes on different racks, the power supply, and the host loads to ensure that the chunk replicas are distributed to different hosts across different racks. This prevents host or rack faults from affecting service availability.

When a data node is damaged or a disk fault occurs on a data node, the chunk replica number becomes smaller than three. The Apsara Distributed File System starts the automatic replication process to replicate data among different service nodes when the replica number is smaller than three. This ensures that each chunk in the cluster has three valid replicas.

Write operations in Tablestore can be returned only after all three replicas are written to the disks. This ensures strong data consistency.

# 1.7.2. Tenant security

### 1.7.2.1. Key management

This topic describes how to manage keys. You can use keys to protect data stored in Tablestore.

Apsara Stack Key Management Service (KMS) is a secure and high-availability service that integrates hardware and software and provides a key management system that can be extended to the cloud.

KMS uses customer master keys (CMKs) to encrypt Tablestore tables and uses the KMS API to generate data encryption keys (DEKs) in a centralized manner. You can define policies in KMS to control and monitor key usage. You can use CMKs to protect data stored in Tablestore.

# 1.8. ApsaraDB RDS

# 1.8.1. Platform security

#### 1.8.1.1. Secure isolation

#### Tenant isolation

ApsaraDB RDS uses virtualization technology to isolate tenants. Each tenant can maintain their own database permissions independently. Apsara Stack also enhances security for servers that run databases. For example, users cannot use the databases to read or write operating system files. This prevents other users from accessing your data.

#### 1.8.1.2. Authentication

ApsaraDB RDS uses authentication to ensure data security.

#### Identity authentication

Account authentication uses your logon password or AccessKey pair to verify your identity. You can create an AccessKey pair from the Apsara Uni-manager Management Console. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey ID is a public key used for identification. The AccessKey secret is used to encrypt signature strings sent from the client and verify signature strings sent by the server. You must keep your AccessKey secret confidential.

ApsaraDB RDS authenticates each access request. Therefore, a request must contain signature information regardless of whether it is sent over HTTP or HTTPS. ApsaraDB RDS uses the AccessKey ID and AccessKey secret to implement symmetric encryption and authenticate the identity of a request sender. AccessKey pairs can be applied for and managed from Apsara Stack. The AccessKey secret must be kept strictly confidential.

#### Permission control

When an ApsaraDB RDS instance is created, the system does not create an initial database account. You can create standard database accounts by using the console or an API operation and configure read and write permissions for the databases. To implement fine-grained permission control, such as table-, view-, or field-based permissions, you can create a privileged database account by using the console or an API operation. You can then use the database client and privileged database account to create standard database accounts. A privileged database account can configure table-based read and write permissions for standard database accounts.

#### Access control

All ApsaraDB RDS instances that are created by an Apsara Stack tenant account are managed as resources of that account. By default, Apsara Stack tenant accounts have full access permissions on their resources.

ApsaraDB RDS supports Resource Access Management (RAM). You can use RAM to allow RAM users to access and manage ApsaraDB RDS resources in your account. ApsaraDB RDS can also use STS to offer temporary credentials for short-term access.

### 1.8.1.3. Data security

ApsaraDB RDS ensures data security by using hot standby, data backup, and log backup.

ApsaraDB RDS instances on High-availability Edition use two database nodes to implement hot standby. When the primary node fails, the secondary node immediately takes over the services. Database backups can be initiated anytime. To improve data traceability, ApsaraDB RDS can restore data to a previous point in time based on backup policies.

Automatic backups at regular intervals are required to ensure the integrity, reliability, and restorability of databases. ApsaraDB RDS provides data backup and log backup.

### 1.8.1.4. Data encryption

SSL

ApsaraDB RDS provides Secure Sockets Layer (SSL) for MySQL, SQL Server, PolarDB, and PostgreSQL. You can use the server root certificate to verify whether the destination database is an ApsaraDB RDS instance. This prevents man-in-the-middle attacks. ApsaraDB RDS also allows you to enable and update SSL certificates for servers to ensure security and validity.

Although ApsaraDB RDS can encrypt the connection between an application and a database, SSL cannot run properly until the application authenticates the server. SSL consumes extra CPU resources, which affects the throughput and response time of instances. The severity of the impact depends on the number of user connections and the frequency of data transfers.

### 1.8.1.5. DDoS attack prevention

ApsaraDB RDS prevents DDoS attacks by using the traffic scrubbing and blackhole filtering features.

When you access an ApsaraDB RDS instance from the Internet, the instance is vulnerable to DDoS attacks. When a DDoS attack is detected, the ApsaraDB RDS security system first scrubs inbound traffic. If traffic scrubbing is insufficient or if the blackhole triggering threshold is reached, blackhole filtering is triggered.

Traffic scrubbing and blackhole filtering must meet the following triggering conditions:

#### Traffic scrubbing

Traffic scrubbing targets traffic only from the Internet. Traffic is redirected from an IP address to the scrubbing device. Then, the scrubbing device checks whether the traffic is normal. The scrubbing device discards abnormal traffic and limits the traffic to the server. These operations mitigate damage on the server but have an impact on normal traffic.

ApsaraDB RDS automatically triggers and stops traffic scrubbing. Traffic scrubbing is triggered for a single ApsaraDB RDS instance if one of the following conditions is met:

- Packets per second (PPS) reaches 30,000.
- o Bits per second (BPS) reaches 180 Mbit/s.
- The number of new concurrent connections per second reaches 10,000.
- The number of active concurrent connections reaches 10,000.
- The number of inactive concurrent connections reaches 10,000.

#### • Blackhole filtering

Blackhole filtering targets traffic only from the Internet. If an ApsaraDB RDS instance is undergoing blackhole filtering, the instance cannot be accessed from the Internet, and connected applications are unavailable. Blackhole filtering is triggered for a single ApsaraDB RDS instance if one of the following conditions is met:

- o BPS reaches 2 Gbit/s.
- Traffic scrubbing is ineffective.

Blackhole filtering is automatically stopped 2.5 hours after it is triggered. Then, the instance undergoes traffic scrubbing. If a DDoS attack is still occurring, blackhole filtering is triggered again. Otherwise, the system restores the normal state.

# 1.8.2. Tenant security

### 1.8.2.1. Log audit

36

ApsaraDB RDS can audit logs to identify security issues.

ApsaraDB RDS allows you to view SQL transactions. You can audit SQL statements on a regular basis to identify and resolve issues. RDS Proxy records all SQL statements sent to ApsaraDB RDS, including the IP address, database name, user account used for execution, SQL statement, execution duration, number of returned records, and execution time of each statement.

### 1.8.2.2. IP address whitelist

ApsaraDB RDS uses IP address whitelists to prevent access from invalid IP addresses.

By default, ApsaraDB RDS instances are accessible from all IP addresses. Therefore, the default IP address whitelist contains only 0.0.0.0/0. You can add IP address whitelist rules by using the data security module in the console or calling an API operation. IP address whitelists can be updated without the need to restart ApsaraDB RDS instances. Whitelist updates do not affect the normal operation of instances. Multiple IP address whitelists can be configured for each instance. Each whitelist can contain up to 1,000 IP addresses or CIDR blocks.

## 1.8.2.3. Software update

ApsaraDB RDS supports post-restart update and mandatory update for software.

ApsaraDB RDS provides you with new versions of database software. In most cases, it is not required to immediately update software. The system updates the database software to the latest compatible version only when you manually restart an ApsaraDB RDS instance.

In rare cases such as critical bugs and security vulnerabilities, ApsaraDB RDS forces the database to update within the maintenance window of the instance. Such mandatory updates only result in temporary database disconnections, without adverse impact on the application if the database connection pool is correctly configured.

You can change the maintenance window by using the console or an API operation. This prevents mandatory updates during peak hours.

# 1.9. Cloud Native Distributed Database PolarDB-X

# 1.9.1. Platform security

## 1.9.1.1. Security isolation

#### **Network** isolation

PolarDB-X supports advanced control of network access by using a Virtual Private Cloud (VPC).

A VPC is a private network environment that you set. It strictly isolates network packets through underlying network protocols, and it controls access at the network layer. The VPC and IP address whitelist together greatly improve the security of PolarDB-X instances.

#### 1.9.1.2. Authentication

PolarDB-X provides a system to manage accounts and permissions, similar to that of MySQL. This system supports commands and functions such as GRANT, REVOKE, SHOW GRANTS, CREATE USER, DROP USER, and SET PASSWORD.

When you create a PolarDB-X database, by default, you can specify an account with all permissions. You can use this account to create one or more new accounts.

- You can grant permissions at the database and table levels. Currently, global permissions and column-level permissions are not supported.
- These eight statements of associated basic permissions are supported: CREATE, DROP, ALTER, INDEX, INSERT, DELETE, UPDATE, and SELECT.
- You can use user@'host' to match and verify access to a host.

Note However, if the business host is in a Virtual Private Cloud (VPC) network, the IP address cannot be obtained due to technical restrictions. In this case, we recommend that you change the format to user@'%'.

# 1.9.2. Tenant security

#### 1.9.2.1. IP address whitelist

PolarDB-X provides IP address whitelists to ensure secure access. You can configure an IP address whitelist for each PolarDB-X database.

The default setting of PolarDB-X instances allows access from any IP address. You can add IP addresses to the whitelist on the **Whitelist Settings** page in the console. You are required to restart PolarDB-X instance after you update the IP address whitelist, and your operations on the instance are not affected. You can set IP addresses or CIDR blocks in the IP address whitelist.

**Note** If the business host is in a Virtual Private Cloud (VPC) network, the IP address cannot be obtained due to technical restrictions. We recommend that you remove the IP address whitelist.

# 1.9.2.2. Protection against high-risk SQL operations

PolarDB-X prohibits high-risk operations such as full table deletion and full table update by default. You can temporarily skip this restriction by adding a hint. The following statements are prohibited by default:

- DELETE statements that do not contain the WHERE or LIMIT conditions.
- UPDATE statements that do not contain the WHERE or LIMIT conditions.

For example, the following statement is prohibited:

#### mysql> delete from tt;

ERR-CODE: [TDDL-4620][ERR\_FORBID\_EXECUTE\_DML\_ALL] Forbid execute DELETE ALL or UPDATE ALL sql. M ore: [http://middleware.alibaba-inc.com/faq/faqByFaqCode.html?faqCode=TDDL-4620]

After a hint is added, the statement is successfully executed.

mysql>/\*TDDL:FORBID\_EXECUTE\_DML\_ALL=false\*/delete from tt; Query OK, 10 row affected (0.21 sec)

### 1.9.2.3. Slow SQL audit

In the PolarDB-X console, you can query the slow SQL statements sent by an application to PolarDB-X. Slow SQL statements increase the response time (RT) of the entire link and reduce the throughput of PolarDB-X.

Contents of a slow SQL statement include the execution start time, database name, SQL statement, client IP address, and execution time. You can query details of slow SQL statements in the PolarDB-X console for optimization and adjustment.

## 1.9.2.4. Performance monitoring

The PolarDB-X console provides monitoring metrics in different dimensions. You can perform related operations based on the monitoring information.

There are two types of PolarDB-X monitoring information:

- Monitoring information about resources, including the CPU, memory, and network.
- Monitoring information about engines, including the logical queries per second (QPS), physical QPS, logical response time (RT) in milliseconds, physical RT in milliseconds, number of connections, and number of active threads.

The QPS and CPU performance of a PolarDB-X instance are in positive correlation. When PolarDB-X encounters a performance bottleneck, the CPU utilization of the PolarDB-X instance remains high. If the CPU utilization exceeds 90% or remains above 80%, the PolarDB-X instance faces a performance bottleneck. If there is no bottleneck in the PolarDB-X instance, the curent type of the PolarDB-X instance cannot meet the QPS performance requirements of the business. In this case, upgrade the instance.

# 1.10. AnalyticDB for PostgreSQL

# 1.10.1. Platform security

## 1.10.1.1. Security isolation

#### **Network** isolation

In Apsara Stack, you can use IP address whitelists to control access. You can also use a VPC to control network access.

A VPC is a private network environment that you can set in Apsara Stack. It strictly isolates network packets by using underlying network protocols and controls access at the network layer.

By default, AnalyticDB for PostgreSQL instances in a VPC are only accessible from the ECS instances within the same VPC. You can also apply for a public IP address to receive access requests from the Internet. This method is not recommended. The requests include but are not limited to:

• Access requests from ECS elastic IP addresses (EIPs).

• Access requests from the Internet egress of your data center.

**Note** IP address whitelists apply to all connections to AnalyticDB for PostgreSQL instances. We recommend that you configure whitelists before you apply for a public IP address.

#### Tenant isolation

AnalyticDB for PostgreSQL uses virtualization technology to isolate tenants. Each tenant can maintain their own database permissions independently. Apsara Stack also enhances security for servers that run databases. For example, users cannot use the databases to read or write operating system files. This prevents other users from accessing your data.

#### 1.10.1.2. Authentication

The AnalyticDB for PostgreSQL instances that you create by using your Apsara Stack tenant account are owned by the account. By default, Apsara Stack tenant accounts have full access permissions on their resources.

AnalyticDB for PostgreSQL supports Resource Access Management (RAM) and Security Token Service (STS). You can use RAM to grant access and management permissions on the AnalyticDB for PostgreSQL resources of your account to other RAM users. You can use STS to issue temporary access credentials to RAM users for short-term access to resources.

# 1.10.1.3. Primary and secondary nodes

Each AnalyticDB for PostgreSQL instance consists of a coordinator node and multiple compute nodes. Each node uses a primary/secondary architecture. If the primary node fails, the service is quickly switched to the secondary node. You can back up databases at any time. AnalyticDB for PostgreSQL can restore data from backup sets based on backup policies to improve data traceability.

# 1.10.2. Tenant security

#### 1.10.2.1. Database account

After you create an instance, you can create a superuser account in the console or by using an API operation. You can execute the GRANT statement to authorize other database accounts.

#### 1.10.2.2. IP address whitelists

By default, AnalyticDB for PostgreSOL instances block access from all IP addresses. The default IP address whitelist contains only 127.0.0.1. You can add IP addresses to a whitelist on the Security Controls page of the console or by using an API operation. The IP address whitelist can be updated without restarting the AnalyticDB for PostgreSQL instance. Whitelist updates do not affect the normal operation of the instance. You can configure multiple IP address whitelists. Each whitelist can contain up to 1,000 IP addresses or CIDR blocks.

# 1.10.2.3. SQL audit

AnalyticDB for PostgreSQL allows you to view SQL details. You can audit SQL operations on a regular basis to identify problems in a timely manner. The Proxy module records the information of all SQL statements that are sent to AnalyticDB for PostgreSQL, including the connected IP addresses, names of the accessed databases, accounts used for statement execution, SQL statements, execution duration, number of returned records, and execution time points.

## 1.10.2.4. Backup and restoration

For data integrity and reliability, a database must automatically back up data on a regular basis to ensure that data can be restored. AnalyticDB for PostgreSQL allows you to restore instances from backup sets.

## 1.10.2.5. Software upgrade

- AnalyticDB for PostgreSQL provides new versions of database software on a regular basis.
- Software upgrade is optional. It is carried out only upon your request.
- If the current database version that you are using has critical security risks, the AnalyticDB for PostgreSQL team will notify you and recommend that you schedule the upgrade. The AnalyticDB for PostgreSQL team provides full support throughout the upgrade.
- The update of AnalyticDB for PostgreSQL is typically completed within five minutes. During the upgrade, network interruptions may occur and the database may become read-only for about one minute. If the database reconnection settings or connection pools are properly configured, the upgrade has minimal impact on applications.

# 1.11. KVStore for Redis

# 1.11.1. Platform security protections

# 1.11.1.1. Security isolation

#### Tenant isolation

KVStore for Redis uses the virtualization technology to isolate tenants. Each tenant can maintain independent database permissions. Alibaba Cloud also increases security protections for the servers that run databases. For example, you cannot read from or write to system files by using the databases, so you cannot access other users' data.

#### **Network** isolation

In Apsara Stack, in addition to the whitelist, you can use Virtual Private Cloud (VPC) to restrict connections.

A VPC is a private network that you specify in Apsara Stack. The VPC strictly isolates your network packets based on network protocols and restricts connections at the network layer. You can use a virtual private network (VPN) or a leased line to connect server resources in your IDC to Alibaba Cloud, and use CIDR blocks in a VPC to prevent IP conflicts. In this way, your own servers and ECS instances can connect to KVStore for Redis instances at the same time. Protections based on the VPC and IP address whitelist improve the instance security.

By default, ECS instances in a VPC can only connect to KVStore for Redis instances in the same VPC. You can also request a public IP address to accept connections over a public network. We recommend that you do not use this connection method. The connection requests include but are not limited to:

- Those from ECS Elastic IP addresses (EIPs).
- Those from the public IP addresses in your own IDC.

Notice The IP whitelist is applicable to all types of connections to KVStore for Redis instances. We recommend that you set the whitelist before requesting the public IP address.

#### 1.11.1.2. Authentication

The instances that you create by using your Alibaba Cloud account are the resources under this account. By default, the Alibaba Cloud account is granted full operation permissions on all the resources under the account.

KVStore for Redis supports Resource Access Management (RAM) and Security Token Service (STS) services. By using RAM, you can create and manage RAM users. You can grant access and management permissions on KVStore for Redis resources under your Alibaba Cloud account to the RAM users. By using STS, you can manage short-term permissions granted to RAM users. You can use STS to grant permissions to temporary users.

# 1.11.1.3. Transmission encryption

KVStore for Redis provides secure encryption based on the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols. You can use the server root certificate from KVStore for Redis to verify that KVStore for Redis provides database services based on the target IP address and port. This can effectively prevent man-in-the-middle attacks (MITM). Also, KVStore for Redis allows you to enable and update SSL and TLS certificates for servers. Therefore, you can replace the SSL or TLS certificate to ensure security and validity.



- To use the transmission encryption feature, you must enable server verification in your application.
- Transmission encryption consumes extra CPU resources and affects the throughput and response time of KVStore for Redis instances. The performance depends on the number of connections and the data transfer frequency.

# 1.11.2. Tenant security protections

#### 1.11.2.1. Database account

To connect to KVStore for Redis, you must pass password authentication. The password is the access credential. KVStore for Redis optimizes the performance of transient connections. Therefore, when you enable password authentication, the performance of KVStore for Redis instances is not affected.

#### 1.11.2.2. IP address whitelist

KVStore for Redis allows you to use an IP address whitelist to restrict connections and secure your data. You can set an IP address whitelist for each KVStore for Redis instance.

By default, KVStore for Redis instances block connections from all IP addresses or CIDR blocks. In this case, the IP address whitelist is set to 127.0.0.1. To add an IP address or CIDR block to the whitelist, in the KVStore for Redis console, choose Instance Information > Change Whitelist. After you modify the IP address whitelist, you do not need to restart the instance, so you can still run the instance normally. You can specify multiple IP address groups for a whitelist. Each group contains a maximum of 1,000 IP addresses or CIDR blocks.

## 1.11.2.3. Backup and recovery

Databases require regular automatic backups to guarantee data integrity, reliability, and restorability. KVStore for Redis supports instance recovery based on backup sets.

# 1.11.2.4. Software upgrade

- KVSt ore for Redis regularly provides database upgrades.
- The upgrades are not mandatory. Databases upgrade to the specified version only when you request.
- When the KVStore for Redis team determines that your version has major security risks, KVStore for Redis notifies you to enable the upgrade. The KVStore for Redis team supports the whole upgrade process.
- KVStore for Redis completes the upgrade within five minutes. During the upgrade, temporary disconnections may occur, and the instance may stay in read-only status for one minute. If you have correctly configured the database reconnection or connection pool for your application, the upgrade does not affect your application.

# 1.12. ApsaraDB for MongoDB

# 1.12.1. Platform security

#### 1.12.1.1. Isolation

#### Network isolation

ApsaraDB for MongoDB allows you to use a VPC for higher-level network isolation.

A VPC is a private network that you configure on Apsara Stack. It strictly isolates your network packets by using underlying network protocols to implement access control at the network layer.

#### Tenant isolation

ApsaraDB for MongoDB uses virtualization technologies to isolate tenants. Each tenant has separate database permissions. Apsara Stack also enhances the security of database servers. For example, Apsara Stack prohibits read and write operations on system files by using a database. This ensures that you cannot access the data of other users.

#### 1.12.1.2. Authentication

#### Identity authentication

Account authentication uses an identity credential to verify the real identity of a user. An identity credential usually refers to a logon password or AccessKey pair. You can create AccessKey pairs in the ApsaraDB for MongoDB console. An AccessKey pair contains an AccessKey ID and an AccessKey secret. The AccessKey ID is public and represents the user identity. The AssessKey secret is used to encrypt the signature string. The server also uses the AccessKey secret to verify the signature string. Make sure that you keep the AccessKey secret confidential.

ApsaraDB for MongoDB performs identity authentication on each access request. Therefore, all HTTP and HTTPS requests must contain the signature information. ApsaraDB for MongoDB implements symmetric encryption through the Access Key ID and Access Key secret to authenticate the request sender. You can apply for an AccessKey ID and an AccessKey secret in the Apsara Uni-manager Operations Console. The AccessKey ID represents the user identity. The AssessKey secret is used to encrypt the signature string. The server also uses the AccessKey secret to verify the signature string. Make sure that you keep the AccessKey secret confidential.

#### Permission management

To log on to an ApsaraDB for MongoDB instance, you must pass the username and password authentication. After an ApsaraDB for MongoDB instance is created, a root account is created by default. You can either specify the password for the root account when you create an instance or reset the password after you create an instance.

The root account has all management permissions on an ApsaraDB for MongoDB instance. You can log on to the database as the root user to add or delete accounts, and grant permissions to other accounts.

#### Access control

ApsaraDB for MongoDB instances that you create by using your Alibaba Cloud tenant account are considered resources under this account. An Alibaba Cloud tenant account has full operation permissions on its resources by default.

ApsaraDB for MongoDB supports Resource Access Management (RAM). RAM allows you to grant access and management permissions on the ApsaraDB for MongoDB resources under your account to RAM users.

## 1.12.1.3. Data security

ApsaraDB for MongoDB uses a high-availability architecture that features a three-node replica set. The three data nodes are located on different physical servers. The secondary and standby nodes automatically synchronize data from the primary node. Services are provided by the primary and secondary nodes. 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.

ApsaraDB for MongoDB can automatically create backups. You can quickly restore data to ensure data integrity and reliability.

You can set the frequency to create full physical backups during one week (at least twice per week) and the start time and end time of the backups. In addition, you can perform full physical backups in the ApsaraDB for MongoDB console, or through APIs at any time based on your O&M requirements.

The system automatically backs up incremental logs generated by an ApsaraDB for MongoDB instance. The combination of full backups and incremental logs enables you to restore data to a specific second in time within the backup retention period.

## 1.12.1.4. Data encryption

#### SSL

ApsaraDB for MongoDB provides Secure Sockets Layer (SSL). You can prevent man-in-the-middle attacks by using the server root certificate to verify whether the destination database is an ApsaraDB for MongoDB instance. ApsaraDB for MongoDB also allows you to enable and update SSL certificates for servers to ensure data security and validity.

#### 1.12.1.5. Anti-DDoS

This feature monitors inbound network traffic in real time. When large volumes of malicious traffic is identified, it scrubs traffic through IP filtering. If traffic scrubbing fails, it triggers the black hole threshold.

# 1.12.2. Tenant security

## 1.12.2.1. Log audit

This feature records all operations that a client performs on a connected database. It provides references for fault analysis, behavior analysis, and security audit. This feature helps you obtain data execution information for analysis. Audit logs will gradually become an essential regulatory requirement of Finance Cloud and other core businesses.

#### 1.12.2.2. IP address whitelists

ApsaraDB for MongoDB allows you to configure IP address whitelists for each ApsaraDB for MongoDB instance to implement network access control.

The default whitelist of an ApsaraDB for MongoDB instance contains 0.0.0.0/0, which indicates that the instance is accessible from all IP addresses. You can use OpenAPI Explorer or the ApsaraDB for MongoDB console to configure an IP address whitelist. Updating an IP address whitelist does not restart the instance, so your business is not affected.

# 1.13. ApsaraDB for OceanBase

# 1.13.1. Platform security

# 1.13.1.1. Security isolation

#### Multi-tenant data isolation

ApsaraDB for OceanBase implements multi-tenant data isolation in databases. This allows one ApsaraDB for OceanBase cluster to serve multiple tenants. Cross-tenant data access is not allowed. This eliminates the risks of data leakage and data breach for each tenant. Each tenant has exclusive use of the resources that are allocated to the tenant. The frontend applications of each tenant offer stable performance. The performance is measured by the following performance metrics: response time, transactions per second (TPS), and queries per second (QPS). These performance metrics for each tenant are not affected by the service loads of the other tenants.

#### 1.13.1.2. Authentication

#### Strong authentication support

When you create a tenant, the system automatically creates a super account that has all the permissions. To log on to your database, use the "<Account>@<Tenant>#<Cluster name>" format. You can use the account that has all the permissions to create accounts and grant the created accounts different permissions.

- You can grant permissions based on the following levels:
  - o Tenant level: The permissions apply to all the databases of the tenant.
  - o Database level: The permissions apply to all the objects in the specified database.
  - Table level: The permissions apply to all the columns in the specified table.
- You can grant the created accounts the following basic permissions: CREATE, DROP, ALTER, INDEX, INSERT, DELETE, UPDATE, and SELECT.

# 1.13.1.3. High-availability architecture

In ApsaraDB for OceanBase, each data record is stored in more than 50% of at least three servers. For example, if three servers are used, each data record must be stored in two of the three servers. Each write transaction is valid only if the transaction is stored in more than 50% of all the servers. Therefore, no data loss occurs if only a minority of all the servers fail. This ensures that a recovery point objective (RPO) of zero can be achieved. In addition, ApsaraDB for OceanBase uses the Paxos protocol at the underlying layer to ensure high availability. If the primary server fails, a new primary server is automatically elected by the remaining servers based on the Paxos protocol. This ensures automatic switchovers and service continuity. In production environments, the recovery time objective (RTO) can be less than 30 seconds.

ApsaraDB for OceanBase retains multiple replicas and uses the Paxos protocol. This allows you to deploy ApsaraDB for OceanBase across data centers in different regions and implement high-availability features such as active geo-redundancy. ApsaraDB for OceanBase supports the following typical deployment solutions: Three Data Centers in One Region, Three Data Centers Across Two Regions, and Five Data Centers Across Three Regions. This allows ApsaraDB for OceanBase to meet the various business requirements for disaster recovery across data centers and zones.

# 1.13.1.4. Compatibility

ApsaraDB for OceanBase is compatible with most of the MySQL 5.6 features. This allows you to migrate MySQL-based services to ApsaraDB for OceanBase based on minimal or zero code modifications. In ApsaraDB for OceanBase, you can create partitioned tables and use subpartitions. This serves as an alternative to the sharding solutions of traditional databases. These features improve the efficiency of application development and data migration.

ApsaraDB for OceanBase is compatible with MySQL in the following aspects:

- APIs: Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC) APIs are used in ApsaraDB for OceanBase. This allows you to access ApsaraDB for OceanBase by using MySQL drivers.
- Data objects: ApsaraDB for OceanBase supports standard SQL objects and objects that are specific to MySQL. The standard SQL objects include databases, tables, views, and auto-increment columns. ApsaraDB for OceanBase implements multitenancy in database systems.

46 > Document Version: 20210621

#### SQL statements:

- ApsaraDB for OceanBase supports the standard SQL statements that you can execute to add, delete, modify, and query data.
- ApsaraDB for OceanBase supports MySQL-specific statements that are frequently used in applications, such as REPLACE and INSERT ON DUPLICATE KEY UPDATE.
- ApsaraDB for OceanBase supports MySQL-specific options, such as the IGNORE option in DML statements and the HINT option that specifies the indexes to be used in SELECT statements.
- System objects: System objects include system views, variables, and functions.
- Transactions: ApsaraDB for OceanBase uses the multi-version concurrency control (MVCC) protocol. This allows you to perform data reads or writes in parallel. ApsaraDB for OceanBase supports the read committed isolation level.

# 1.13.2. Tenant security

#### 1.13.2.1. Database accounts

When you log on to the ApsaraDB for OceanBase console, username and password authentication is required. After you log on to the console, you can create tenants, manage clusters, and perform operations and maintenance (O&M) tasks on clusters.

#### 1.13.2.2. IP address whitelists

ApsaraDB for OceanBase allows you to use whitelists to implement security access control. You can configure a whitelist for each ApsaraDB for OceanBase tenant.

By default, a newly created ApsaraDB for OceanBase instance is accessible from all IP addresses. You can execute the following statement to modify the default access setting: ALTER TENANT tenantname SET VARIABLES ob tcp invited nodes = '192.168.0.0/16,10.125.227.255/255.255.252.0'.

## 1.13.2.3. Log audit

ApsaraDB for OceanBase allows you to audit SQL logs. This provides an efficient method for you to identify issues. The gv\$sql\_audit file records the details of each SQL statement that is routed to ApsaraDB for OceanBase. The details include the corresponding server IP address, database name, user account, SQL statement, start time of execution, execution duration, and queuing time.

# 1.13.2.4. Software upgrades

- ApsaraDB for OceanBase releases software versions on a regular basis.
- Software upgrades are optional. Software upgrades are implemented only after you send requests to upgrade software to the specified versions.
- If the assessment results of the ApsaraDB for OceanBase team show that your version is experiencing major security risks, the team notifies you of the risks. The team also recommends that you schedule and perform software upgrades.
- The required time for upgrades depends on cluster sizes. In most cases, each upgrade requires dozens of minutes to an hour. Your services are not affected during the upgrade processes.

# 1.13.3. Notes on MySQL-related vulnerabilities in ApsaraDB for OceanBase

#### Notes on MySQL-related vulnerabilities in ApsaraDB for OceanBase

To Alibaba Cloud Apsara Stack users:

A recent security audit report revealed that ApsaraDB for OceanBase used MySQL 5.6.25. This MySQL version is at the risk of being attacked.

The version number 5.6.25 is used as a logical version number to implement the compatibility between ApsaraDB for OceanBase and MySQL. ApsaraDB for OceanBase is compatible with MySQL 5.6.25 and does not use MySQL 5.6.25. ApsaraDB for OceanBase uses a proprietary kernel and is not exposed to the security vulnerabilities that are caused by MySQL 5.6.25.

The explanation applies if MySQL server-related vulnerabilities are detected in ApsaraDB for OceanBase.

Beijing Ant Yun Financial Services Co., Ltd. May 16, 2019

# 1.14. Data Transmission Service (DTS)

# 1.14.1. Platform security

## 1.14.1.1. Security isolation

DTS uses independent processes and files to isolate instances and data between tenants. For example, users are not allowed to read/write OS files of instances so that users cannot access data of other users.

#### 1.14.1.2. Authentication

You can use your Alibaba Cloud account to create a DTS instance. The resources of the DTS instance are owned by the Alibaba Cloud account. The account has full access permissions on its DTS resources by default.

DTS supports RAM for Alibaba Cloud. You can assign permissions to access and manage DTS resources to RAM users. RAM enables you to assign permissions as needed and helps enterprises minimize information security risks.

## 1.14.1.3. Transmission security

To enhance data transmission security, Data Transmission Service (DTS) uses self-defined log formats.

In DTS, data is encrypted for secure transmission. For example, data is encrypted during incremental data synchronization between the data reading module and the data synchronization module.

DTS also supports HTTPS to effectively improve access security.

# 1.14.1.4. Data security

When you use DTS to synchronize or subscribe to incremental data, many pieces of incremental data are stored on the DTS server. The incremental data is serialized and stored based on the storage format defined in DTS. The DTS-defined storage format helps to enhance data security.

**Note** After data is written to the DTS server, it is stored for seven days and automatically deleted after this period of time expires.

# 1.15. Data Management (DMS)

# 1.15.1. Platform security

# 1.15.1.1. Security isolation

DMS controls access at the following four levels: Apsara Stack tenant accounts, users, IP whitelists, and permissions.

- Required. DMS uses RAM users or Apsara Stack tenant accounts as the first-level authentication.
- Required. Users are grouped into different enterprises. These enterprises are isolated between each other. A user of an enterprise cannot access the resources of another enterprise even if the user

passes the first-level authentication.

- Required. DMS uses IP whitelists to control access from IP addresses over the Internet.
- Required. After a user passes the third-level authentication, the user still cannot query or update databases and tables if the user does not have the required permissions.

#### 1.15.1.2. Authentication

Authentication for Apsara Stack tenant accounts

Before you use Data Management (DMS), you must log on to the Apsara Uni-manager Management Console by using your account and password. If your logon session expires, logon fails, or you switch to another account, you can no longer access DMS. You must use your Apsara Stack tenant account to log on to DMS again. You can use DMS after you log on to the DMS console with your Alibaba Cloud tenant account.

Access control over database accounts

After you log on to the DMS console by using your Alibaba Cloud tenant account, DMS checks your account permissions when you add a database instance to DMS. One of the DMS users must be the owner of a database to be accessed, or the owner of the database has granted permissions to the current logon user. Otherwise, the current user cannot add the database to DMS for subsequent management.

## 1.15.1.3. Data security

#### Security control over data

Dat a security for queries in the SQLConsole.

- Fine-grained access control: DMS allows you to control access to databases at the database, table, or field level without the need to use your database accounts and passwords. Field-level access control allows you to manage several sensitive fields, such as the ID card, mobile phone number, and password.
- Maximum number of returned entries in a single query: If the number of returned entries in an SQL query exceeds the limit, only the specified maximum number of entries can be returned. This prevents data transmission on a large scale. This feature applies to all DMS users on a global scale.
- Maximum number of daily returned entries: If the total number of daily returned entries exceeds the limit, you can no longer issue a query. This prevents data transmission on a large scale. This feature applies to all DMS users and can be changed for a single user.
- Maximum number of daily queries: If the total number of daily queries exceeds the limit, you can no longer issue a query. This prevents data transmission on a large scale. This feature applies to all DMS users and can be changed for a single user.
- Maximum timeout period of a single SQL query: If the execution time of an SQL query exceeds the maximum time period, the system stops the SQL query. This prevents business interruption that is caused by compromised database performance. This feature applies to all DMS users and can be changed for a single instance.
- Maximum size of an SQL table to be queried: The system stops an SQL query for a table if the following conditions are true: 1. The size of the table to be queried exceeds the limit. 2. SQL fails to use an index to perform the query and must transverse the entire table to complete the query. This avoids business interruption that is caused by compromised database performance. This feature applies to all DMS users and can be changed at the database level on a global scale.

## 1.15.1.4. Data encryption

KMS-based encryption

Server-side encryption is used to protect static data. Apsara Stack Key Management Service (KMS) is a secure and high-availability service that integrates hardware and software and provides a key management system that can be extended to the cloud. KMS uses customer master keys (CMKs) to encrypt the database keys, sensitive information, and sensitive settings of DMS. KMS calls the required API operations to generate data encryption keys (DEKs) at a time. You can define policies in KMS to control and monitor key usage.

• Support for HTTPS and SSL

DMS supports HTTPS and SSL connections between user-side browsers and DMS servers. This ensures that data is not intercepted during data transmission.

## 1.15.1.5. Limits on changes

#### Limits on data changes

- Approval for changes: If you do not have the required permissions to change a destination database, you cannot submit a ticket to request changes. This avoids unexpected operations from unauthorized operators. This feature applies to all DMS users. After you obtain the required permissions, you can submit a ticket.
- Accuracy for changes: The syntax of all the SQL statements in a ticket that is submitted to request changes must be valid. If one of the SQL statements is invalid, the system rejects the ticket. This avoids unexpected changes that are caused by invalid syntax. This feature applies to all DMS users.
- Number of affected rows: After the syntax of SQL statements in a ticket is verified, the system checks the number of affected rows. If the actual number of affected rows is different from the estimated number of affected rows, a notification is sent. This avoids unexpected changes that are caused by the deviation between the specified SQL logic and the actual results. This feature applies to all DMS users.
- Process of changes: You can configure approval processes based on the number of affected rows or change type. This ensures security and development efficiency. This feature applies to all DMS users and can be changed for instances.
- Change scripts for transactions: You can enable the transaction feature on demand when changes are being executed on relational databases such as MySQL. If all changes are successful, these changes are applied. If one of these changes fails, all the changes roll back. This ensures transaction compliance. This feature applies to all DMS users and can be enabled or disabled on demand. This feature is unavailable for PolarDB-X (previously called DRDS).
- Execution of change scripts on schedule: If you need to execute changes during off-peak hours, you can configure a schedule that runs a change task at a specified time and returns results. This avoids man-made errors. This feature applies to all DMS users and can be enabled on demand.
- Backup of change scripts: Before you execute a change script, you can back up all the rows to be changed by using an insert script. This allows you to roll back all the changes and restore all the rows that are backed up immediately after unexpected changes occur. This feature applies to all DMS users. In each ticket, you can request to back up a maximum of 500,000 rows.
- Risk management prior to the execution of changes: Before you execute a change, the system checks the metadata lock of the destination table. If the system fails to obtain the metadata lock within 10 seconds, the system retries. After three failed retries, the related database is deemed busy and the

change task fails. This prevents the system from adding extra workloads to the database. This also prevents business interruption that is caused by compromised database performance. This feature applies to all DMS users.

- Risk management prior to the execution of changes: Before you execute a change, the system checks
  the number of active connections to the destination database. If the number exceeds the limit, the
  system retries. After three failed retries, the database is deemed busy and the change task fails. This
  prevents the system from adding extra workloads to the database. This also prevents business
  interruption that is caused by compromised database performance. This feature applies to all DMS
  users.
- Management during the execution of changes: After an SQL query is completed, the system switches
  to the sleep mode. The period of time in sleep mode is used to balance integrated workload across
  multiple periods of time. This ensures smooth changes. This prevents the system from adding extra
  workloads to the database. This also prevents business interruption that is caused by compromised
  database performance. This feature applies to all DMS users.

# 1.15.2. Tenant security

## 1.15.2.1. Log audit

DMS allows you to audit all operational logs. To audit these logs across multiple dimensions, choose **System Management > Security > Operation audit**. Each operational log includes several fields, such as the operator, operation time, operation, and execution impact.

# 1.16. Server Load Balancer (SLB)

# 1.16.1. Platform security

#### 1.16.1.1. Authentication

SLB instances created using an Apsara Stack tenant account are owned by the account. By default, the tenant account has full permissions on these resources.

You can manage SLB instances by using RAM. You can authorize a RAM user to access and manage SLB resources owned by the tenant account. SLB also supports STS, which enables temporary access to SLB resources.

# 1.16.2. Tenant security

#### 1.16.2.1. HTTPS

SLB supports HTTPS load balancing to forward HTTPS requests.

SLB supports HTTPS, SSL, and TLS load balancing:

- SLB provides centralized certificate and key management for services that require certificate
  authentication. This eliminates the need to deploy certificate management systems on individual ECS
  instances.
- All decryption operations are performed on SLB, reducing the CPU usage of backend ECS instances.

SLB provides centralized certificate management that allows you to store certificates and keys. All private keys uploaded to the certificate management system are encrypted.

#### 1.16.2.2. IP address whitelists

SLB masks the IP addresses of backend servers and provides only the IP address of the SLB instance for external use.

SLB also provides the whitelist function. By adding a whitelist, you can control which IP addresses can access the SLB service.

## 1.16.2.3. Log management

Server Load Balancer (SLB) provides the log management feature that allows you to view the operation and health check logs of an SLB instance.

# 1.17. Virtual Private Cloud (VPC)

# 1.17.1. Platform security

# 1.17.1.1. Security isolation

VPCs are isolated through tunneling technology. The isolation effect between VPCs is the same as that of the traditional VLANs. Broadcast domain isolation can be achieved on ECS instances and NICs. VPCs, like VLANs, are isolated at the network layer. Meanwhile, VPCs divide different security domains for access control.

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.

#### 1.17.1.2. Access control

VPCs support RAM. RAM allows you to grant access and management permissions on your VPC resources to RAM users.

VPCs also support STS, which issues temporary credentials to grant temporary access permissions.

# 1.17.2. Tenant security

# 1.17.2.1. Security groups

VPCs use security groups to divide network security domains and implement Layer 3 access control. The security groups act as virtual firewalls and provide stateful inspection and packet filtering features.

VPCs are isolated by default and can be connected to each other through peering connections.

# 1.18. Log Service

# 1.18.1. Platform security

# 1.18.1.1. Security isolation

Logtail supports multi-tenant isolation. Compared with mainstream open-source collection agents, Logtail has a more refined architecture. A fixed number of threads are used by Logtail to discover events, read data, parse data, and send data. The number of threads does not increase as the number of configurations increases. All configurations operate in the same execution environment. However, Log Service uses multiple technical methods to guarantee processing isolation of configurations, fair scheduling of configurations, reliability and controllability of data collection, and high cost performance of resources.

The characteristics of Logtail multi-tenant isolation are as follows:

- Logtail schedules data collection based on time slices to guarantee isolation and fairness of configuration data endpoints.
- Logtail supports multi-level feedback queues for resource usage to guarantee the isolation and fairness of processing flows and configurations with extremely low resource consumption.
- Logtail supports non-blocking mode of event processing to guarantee high reliability even if log files are rotated when configuration is blocked or data collection is stopped.
- Logtail supports different throttling mechanisms for various configurations, policies of stopping collection, dynamic configuration updates to guarantee a high level of control for data collection.

#### 1.18.1.2. Authentication

To ensure the security of log data, this topic describes the authentication method. The process of authentication works the same as that of the identity authentication. This authentication method allows you to delete a Resource Access Management (RAM) user. Each HTTP request of a Log Service API operation must pass security verification to ensure log data security. The security authentication is based on the Alibaba Cloud AccessKey and is done by using the symmetric encryption algorithm.

The following procedure shows how authentication works:

- 1. The requester generates a signature string based on the API request content (including HTTP header and body).
- 2. The requester uses Alibaba Cloud AccessKey pair (AccessKey ID and AccessKey secret) to sign the signature string generated in the first step. A digital signature is generated for this API request.
- 3. The requester sends both the API request content and digital signature to the server.
- 4. After the server receives the request, the server repeats Step 1 and Step 2 to compute the expected digital signature for this request.
  - ? Note The server retrieves the AccessKey pair used by this request from the background.
- 5. The server compares the expected digital signature with the digital signature sent from the requester. If they are the same, the request passes security authentication. Otherwise, the request is rejected.

## 1.18.1.3. Data security

The operation of collecting Log Service user data is mapped to reading and writing operations of files which are stored in the Apsara Stack data system.

Apsara Stack uses a flat network in which a linear address space is divided into slices, which are also called chunks. Each chunk is duplicated into three copies. Each copy is stored on different nodes in the cluster to ensure the reliability of data.

The triplicate technology used for the Apsara Stack data system involves three key components: master, chunk server, and client. Each write operation performed by an ECS user undergoes several processes before the client executes the operation. The following section describes how the client executes this operation:

- 1. The client determines the location of the chunk corresponding to the write operation.
- 2. The client sends a request to the master to query the storage locations (chunk servers) of all three chunk replicas.
- 3. The client sends write requests to the chunk servers that are returned from the master.
- 4. If the write operation succeeds on all three chunk replicas, the client returns a success message. Otherwise, the client returns a failure message.

The distribution strategy of the master takes all factors of the entire system into account, such as chunk server disk usage, chunk server distribution across different racks, power distribution conditions, and machine workloads. This strategy ensures that each chunk replica is distributed on different chunk servers across different racks, which effectively prevent data unavailability if a rack or chunk server fails.

In the cases of data node damage or hard disk failures on data nodes, the total number of valid replicas of some chunks become less than three. In these cases, the master replicates data between chunk servers to maintain three valid replicas of all chunks in the cluster.

All user operations including addition, modification, and deletion of data are synchronized to the three copies. This mechanism ensures the reliability and consistency of user data.

Furthermore, when you delete data, the released storage space is reclaimed by Apsara Distributed File System and is not accessible to users. Data erasure is performed before the storage space is reused. This mechanism provides the highest level of data security.

# 1.18.1.4. Encrypted data transmission

Log Service ensures your data security in transmission by using the following methods:

- You use your Alibaba Cloud AccessKey to perform Log Service authentication. To prevent data tampering during data transmission, the Logtail client obtains your Alibaba Cloud AccessKey to sign all log data packets before they are sent. The Logtail client also uses the HMAC SHA1 signature algorithm for authentication.
  - Note The Logtail client obtains your Alibaba Cloud AccessKey over HTTPS to ensure the security of your AccessKey.
- The API layer uses the signature authentication mechanism to control access permissions and ensure the security of your data.
- Log Service applies HTTPS and SSL to the network connection between a client and a server to ensure that data is not monitored or stolen during transmission. Log Service communicates with the client over HTTPS to ensure data security.

# 1.18.1.5. Encrypt data

Log Service allows you to use Key Management Service (KMS) to encrypt data for secure storage.

Log Service encrypts data by using KMS. The data encryption mechanism has the following characteristics:

- Log Service supports the Advanced Encryption Standard (AES) encryption algorithm.
- You can create and manage a customer master key (CMK) in the KMS console to ensure the security of the CMK.
- Log Service supports the following encryption types:
  - Service key-based encryption: Log service generates an independent service key for each Logstore.
     The service key never expires.
  - Bring Your Own Key (BYOK) encryption: You can create a CMK in the KMS console and grant the
    relevant permissions to Log Service. When Log Service calls a KMS API operation, this CMK is used to
    create a key for data encryption. If the CMK is deleted or disabled, the BYOK key becomes invalid.

# 1.18.2. Tenant security

# 1.18.2.1. Service monitoring

Log Service monitors machine group status and log collection status of Logtail in real time.

• Machine group status

Log Service monitors the heart beat status of all the servers in your machine group in real time. The server status can be **OK** or **Fail**. **Fail** indicates that the machine group is in an abnormal state and cannot collect logs.

• Log collection status

When you use Logtail to collect logs, Log Service sends alerts through **Collection Error Diagnosis** if errors such as failed regular expression parsing, invalid file path, and insufficient shard capacity occur. Alerts contain information about the errors, such as the time of occurrence, the IP address of the server, the number of errors, and the types of errors.

# 1.19. Key Management Service (KMS)

# 1.19.1. Platform security

# 1.19.1.1. Security isolation

This topic describes the security isolation of KMS.

**Note** No instances are deployed in KMS. Therefore, resource isolation caused by instance virtualization do not occur.

CMKs are the only resources in KMS. You can use CMKs by calling API operations but cannot directly access the CMK data. Security isolation is implemented at the network layer of the API.

#### 1.19.1.2. Authentication

56

#### 1.19.1.2.1. Authentication

You can create an AccessKey pair in the Apsara Uni-manager Management Console. An AccessKey pair consists of an AccessKey ID and AccessKey secret. The AccessKey ID is public and uniquely identifies a user, whereas the AccessKey secret is private and used to authenticate a user.

Before you send a request to KMS, you must generate a signature string for the request in the format specified by KMS. Then, you must encrypt the signature string by using your AccessKey secret to generate a verification code based on the Hash-based Message Authentication Code (HMAC) algorithm. The verification code carries a timestamp to prevent replay attacks. After KMS receives the request, KMS finds the AccessKey secret based on your AccessKey ID, and uses the AccessKey secret to extract the signature string and verification code. If the generated verification code is the same as that in the request, KMS determines that the request is valid. Otherwise, KMS rejects the request and returns HTTP status code 403.

#### 1.19.1.2.2. Access control

Resource Access Management (RAM) is used for access control in KMS. You can use RAM policies to define different identity types and grant RAM users permissions on KMS.

Permissions on KMS have the following basic elements:

- Action: the Action parameter in KMS API requests. For example, you can specify this parameter to create, delete, modify, or query keys, as well as to encrypt or decrypt data by using keys. Each API operation corresponds to an action. You can grant permissions on each action to an identity.
- Resource: Keys are the only resources of KMS. Key IDs are used to identify resources.

## 1.19.1.2.3. RAM and STS support

KMS supports RAM and Security Token Service (STS) authentication.

RAM is a resource access control service provided by Alibaba Cloud. It allows you to create RAM users under your Apsara Stack tenant account and grant them permissions to access resources.

STS is an Alibaba Cloud service that provides temporary access credentials. It is used for short-term access control. You can use STS to generate a temporary access credential. You can specify the permissions and validity period of the credential. The credential expires automatically upon its expiration date.

# 1.19.1.3. Data security

KMS uses multiple mechanisms to ensure the security of your data.

Data in KMS refers to the customer master keys (CMKs) that are created and managed in KMS. CMKs are stored in ApsaraDB RDS servers in primary/secondary mode. Each primary or secondary server has its own redundancy and backup mechanism. This way, ApsaraDB RDS implements hierarchical redundancy for CMK data.

The key material of CMKs is encrypted by KMS before the material is stored on disks. KMS provides a hierarchical key architecture and automatically rotates upper-layer keys.

# 1.19.1.4. Transmission encryption

KMS implements end-to-end encryption for data transmission.

When you send a request to KMS, you must use HTTPS to ensure the privacy and integrity of the exchanged information.

# 1.20. Apsara Stack DNS

# 1.20.1. Tenant security

#### 1.20.1.1. Tenant isolation

Apsara Stack DNS isolates data by tenant. After receiving a DNS request, Apsara Stack DNS determines whether the request is authorized based on the AccessKey pair contained in the request. If it is, Apsara Stack DNS processes the data. Tunnel IDs are used to associate VPCs with zones. The backend DNS server responds to the users' DNS requests based on tunnel IDs. This helps isolate data among tenants.

# 1.20.1.2. Network security hardening

Apsara Stack DNS provides recursive resolution. To protect against potential security risks from the Internet, Apsara Stack DNS reinforces the security of domain name resolution. Specifically, only outbound traffic is allowed, and all inbound traffic from the Internet is discarded.

## 1.20.1.3. Log audit

Log audit is an essential step to ensure security. Apsara Stack DNS provides you with detailed log information. All operations are logged in real time. If a security breach occurs, you can query log entries to trace the activities of the attacker and analyze the security breach.

# 1.21. API Gateway

# 1.21.1. Platform security

# 1.21.1.1. Security isolation

API Gateway isolates resources by tenant. Resources belong only to the tenant account they are created in. Resources are isolated between different tenants.

## 1.21.1.2. Authentication

#### 1.21.1.2.1. Authentication

You can create an app in the API Gateway console. After you create an app, a key pair that consists of an AppKey and an AppSecret is automatically assigned to the app. The AppKey is a public key that is used to identify your identity. The AppSecret is a private key that is used to authenticate your identity.

Before you send a request to API Gateway, you must generate a signature string for the request in the format specified by API Gateway. Then, you must use your AppSecret to encrypt the signature string based on the HMAC algorithm to generate a verification code. The verification code contains a timestamp to prevent replay attacks. After API Gateway receives the request, API Gateway finds the AppSecret that corresponds to the AppKey, and extracts the signature string and verification code in the same way. If the calculated verification code is the same as the one received, the request is valid. Otherwise, API Gateway rejects the request and returns an HTTP 403 error.

#### 1.21.1.2.2. API access control

API Gateway users consist of API providers and third-party users (API callers). The provider of an API can request a third-party user to provide an Appld. After the API provider authorizes the application specified by Appld to call the API, the third-party user can immediately initiate an access request to the API by using the AppKey and AppSecret of the application.

An access request must carry the signature of the user who accesses the API. A signature-based access request is a request that contains signature information in the header as stipulated in the API Gateway documentation.

## 1.21.1.2.3. RAM and STS support

API Gateway allows you to manage APIs through RAM and STS.

Resource Access Management (RAM) is a resource access control service provided by Alibaba Cloud. You can use an Apsara Stack tenant account to create RAM user accounts and grant them permissions to access the resources that belong to the tenant account.

Security Token Service (STS) is a temporary access credential service provided by Alibaba Cloud. It provides temporary access control. You can use STS to generate a temporary access credential. You can determine the permissions and validity period of the credential. The access credential expires automatically upon its expiration date.

## 1.21.1.3. Data security

API Gateway uses signature authentication to ensure the consistency and integrity of user data when an API is called. In addition, API Gateway provides the data cleansing function. During data cleansing, invalid parameters are cleaned to ensure the security of requests.

API Gateway requires the caller to add user identity information to the API request, and add an encrypted signature to the data for transmission. After receiving an API request, API Gateway verifies the user identity and checks data for integrity and consistency.

In addition, API Gateway can clean invalid parameters that are not preset by users. This ensures that only valid and secure API requests are approved.

# 1.21.1.4. Transmission encryption

API Gateway supports HTTPS to ensure the security of data during transmission.

# 1.21.2. Tenant security

# 1.21.2.1. Log audit

Log Service is a log management service provided by Alibaba Cloud. API Gateway can be used together with Log Service to provide you with access, monitoring, and audit information query and display functions. API Gateway records API requests in real time and synchronizes the logs to Log Service on a regular basis. Request logs contain information such as the request time, source IP address, requested object, returned code, and processing duration.

#### 1.21.2.2. IP address-based access control

API Gateway allows you to set a blacklist and a whitelist based on the ClientIP of a caller.

An IP address-based access control policy takes effect immediately after you bind it to an API. This can prevent API requests from unauthorized IP addresses.

- Blacklist: prohibits API requests from the specified IP addresses.
- Whitelist: permits the API requests from only the specified IP addresses.

# 1.22. Enterprise Distributed Application Service (EDAS)

# 1.22.1. Platform security

The platform-side security design mainly includes authentication and transmission encryption.

#### 1.22.1.1. Authentication

Authentication Enterprise Distributed Application Service (EDAS) mainly includes permission control, access control, and API authentication.

#### **ACL**

#### • EDAS Agent

EDAS Agent implements authentication and authorization at two layers: it restricts authentication from the consumer to provider and implements the principle of least privilege by authenticating the AccessKey ID and AccessKey secret of delivered commands.

#### DAuth

- Security credentials: EDAS DAuth can generate the App\_KEY and App\_SECRET for EDAS access requests to implement authentication control.
- Authentication settings: Granular policy settings for EDAS authorization are supported, including authorization enabling, signature verification, log switch, custom logs, log cache, and log detection.

#### Diamond Server

Diamond Server uses the App\_KEY and App\_SECRET generated by DAuth to authenticate access requests to HTTP and HTTPS APIs.

#### · Config Server

- Throttling: HTTP requests sent to Config Server can be throttled to prevent Config Server from being unstable under excessive workloads.
- Authentication: Config Server uses the App\_KEY and App\_SECRET generated by DAuth to authenticate access requests to HTTP and HTTPS APIs.

#### Access control

EDAS RAM authorization can retrieve your on-demand AccessKey ID and AccessKey secret through STS. You can create ECS instances by using the on-demand AccessKey ID and AccessKey secret to access your ECS API. On-demand AccessKey ID and AccessKey secret have a validity period and must be regenerated after expiration. The ECS API permissions are restricted. You can call API operations only with the specified API operation permissions. The restrictions avoid excessive permissions caused by the use of ECS RAM FullAccess.

#### API authentication

API access permissions are authenticated using the AccessKey ID and AccessKey secret of the RAM user. The EDAS API uses the custom key-based Hash Message Authentication Code (HMAC) HTTPS solution for authentication. To authenticate a request, you must combine certain elements in the request to form a string. Then, use the EDAS key to calculate the HMAC of the string. Generally, this process is called request signing. The output HMAC algorithm is called signature because the algorithm simulates the security attributes of real signatures. Finally, you can use the EDAS API syntax to add the signature as a request parameter.

After the system receives the authenticated request, it extracts the EDAS key and uses the key to calculate the signature of the received message by using the same method. Then, the system compares the calculated signature with the signature of the requester. If both signatures match, the system considers that the requester has the access permission on the EDAS key and issues the key to the requester as the issuance authority of the delegator. If the signatures do not match, the request is discarded, and the system returns an error message.

## 1.22.1.2. Transmission encryption

Enterprise Distributed Application Service (EDAS) uses the TLS certificate deployment solution to realize full link encryption. Considering the validity period of the certificate, EDAS supports certificate update.

#### **EDAS** management security

EDAS management involves the following roles:

- EDAS Console: allows you to create clusters and deploy packages in the EDAS console.
- EDAS Agent: is an EDAS client deployed on your Elastic Compute Service (ECS) instance and supports various EDAS operations. EDAS Agent consists of two components: StarAgent and EDAS Agent.
- EDAS Server: receives commands from EDAS Console and delivers them to EDAS Agent.

The security process of EDAS management is as follows:

- 1. When EDAS Console receives your deployment command (for example, deploying a WAR package), EDAS Console performs API authentication and uses the AccessKey ID and AccessKey secret of a Resource Access Management (RAM) user to connect to the EDAS Server API.
- 2. EDAS Server sends the encrypted commands to EDAS Agent through the encrypted channel.

#### **EDAS RPC security**

The EDAS remote procedure call (RPC) process involves the following roles:

- EDAS DAuth: generates the AccessKey ID and AccessKey secret of an EDAS user and performs actions such as authentication.
- EDAS Config Server: provides information including service registration, IP addresses, and called APIs.
- EDAS Dubbo and High-speed Service Framework (HSF): are distributed RPC products.
- EDAS Pandora: is a lightweight container isolation service for class isolation and loading.

#### The EDAS RPC process is as follows:

- 1. Both the provider and consumer start the Pandora process. Pandora provides call services for HSF and Dubbo.
- 2. On the ECS instance to which it belongs, the provider registers a service with EDAS Config Server. During service registration with EDAS Config Server, TLS is used to protect link security, and the EDAS AccessKey ID and AccessKey secret are used for authentication. Only specified users can publish called APIs to register with EDAS Config Server.
- 3. The consumer pulls related information from EDAS Config Server, including the service name and service IP address provided by the provider.
- 4. The consumer directly calls the Pandora process of the provider to access the service, uses the EDAS AccessKey ID and AccessKey secret for authentication to call the service.

#### **EDAS Docker security**

The EDAS Docker call process is as follows:

- 1. Create a Docker cluster in EDAS by calling the Docker API. EDAS uses the RAM Security Token Service (STS) on-demand token for authentication.
- 2. Create cluster certificates. One certificate is generated for each cluster.
- 3. Create resources, including ECS and Server Load Balancer (SLB) instances, and use the RAM STS ondemand token to call the APIs for ECS or SLB instances. Authorize the required APIs.
- 4. Configure nodes.
  - i. Generate node certificates. The system generates a root certificate for a cluster and uses cloud-init to directly generate node certificates based on the root certificate.
  - ii. Use cloud-init to install Docker.
  - iii. Install system services such as EDAS Agent by using the EDAS installation scripts. Then you can apply the EDAS management process to manage these services.

# 1.22.2. Tenant security

API auditing is the main security function on the tenant side.

ActionTrail records the operations performed on resources under users' cloud accounts, supports operation record query, and stores record files in user-specified OSS buckets. The operation records stored by ActionTrail can be used for security analysis, resource change tracking, and compliance audit.

ActionTrail collects API call records of cloud services, including the records of API calls initiated in the console). It standardizes the operation records and saves them to user-specified OSS buckets as files. You can manage the record files by using all the management functions provided by OSS, such as authorization, lifecycle management, and archiving management.

# 1.23. MaxCompute

# 1.23.1. Platform security

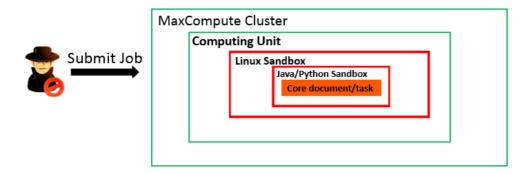
# 1.23.1.1. Security isolation

This topic describes security isolation for MaxCompute platform security.

MaxCompute can be used to handle security issues in multi-tenant scenarios. It integrates the authentication system of Alibaba Cloud to authenticate users by using symmetric AccessKey pairs. MaxCompute verifies the signature in each HTTP request, and discretely stores and isolates data of different users in Apsara Distributed File System. This allows MaxCompute to meet the requirements for multi-tenant collaboration, data sharing, data confidentiality, and data security.

MaxCompute runs all computing jobs in individual sandboxes. The sandbox architecture has multiple layers from the kernel layer to the KVM virtualization layer. The sandboxes use an authentication mechanism to ensure data security and prevent server faults caused by misoperations or malicious operations.

Sandbox-based protection



#### Network isolation

MaxCompute is a big data platform provided by Alibaba Cloud to process large amounts of data. It complies with isolation standards to ensure data security. MaxCompute supports virtual private clouds (VPCs), which allow you to isolate data. Limits are imposed when you use MaxCompute in VPCs.

#### Limits:

- The classic network, VPCs, and the Internet are isolated from each other. Users can access only the endpoints and virtual IP addresses (VIPs) of their own networks.
- Domain names that pass the verification on the three types of networks can be used to access the projects that do not have VPC IDs or IP address whitelists configured.
- Projects that have VPC IDs configured are accessible only to the specified VPCs.
- Projects that have IP address whitelists configured are accessible only to the hosts whose IP addresses are added to the IP address whitelists.
- If a request is sent by a proxy server, the request is allowed or denied based on the VPC ID and IP address of the last-hop proxy server.

Elasticsearch on MaxCompute is an enterprise-class full-text retrieval system developed by Alibaba Cloud. It must also comply with isolation standards to ensure data security. Therefore, Elasticsearch on MaxCompute supports VPCs. Limits are imposed when you use Elasticsearch on MaxCompute in VPCs.

#### Limits:

- The classic network, VPCs, and the Internet are isolated from each other. Users can access only the endpoints and VIPs of their own networks.
- Domain names that pass the verification on the three types of networks can be used to access the projects that do not have VPC IDs or IP address whitelists configured.
- When an Elasticsearch cluster is started in a MaxCompute project, they share the same whitelist of

VPCs.

• By default, if an Elasticsearch cluster is started, the cluster occupies all resources. If you want to start more Elasticsearch clusters, you must scale up the MaxCompute project or scale down the Elasticsearch cluster.

When you deploy MaxCompute in Apsara Stack, a MaxCompute project is automatically created and an Elasticsearch cluster is automatically started in the project. You can start your Elasticsearch cluster in your MaxCompute project. After the cluster is started, you must apply for a domain name and VIP and verify the Elasticsearch cluster deployed in a VPC in the Elasticsearch frontend.

#### 1.23.1.2. Authentication

This topic describes authentication for MaxCompute platform security.

#### Identity authentication

You can create an AccessKey pair in the Apsara Uni-manager Management Console. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey ID is public and used to identify a user, whereas the AccessKey secret is private and used to authenticate the user identity.

Before the client sends a request to MaxCompute, the client generates a string to be signed in the format specified by MaxCompute, uses the AccessKey secret to encrypt the string, and then generates a signature for the request. After MaxCompute receives the request, it identifies the AccessKey secret based on the AccessKey ID and generates a signature. If the signature is the same as that sent by the client, the request is valid. Otherwise, MaxCompute rejects the request and returns an HTTP 403 error.

#### Access control

You can use an Apsara Stack tenant account or a RAM user to access MaxCompute resources. Different RAM users can be created within one Apsara Stack tenant account. MaxCompute checks the permissions of your Apsara Stack tenant account or RAM user each time you access its resources.

- If you access a resource by using your Apsara Stack tenant account, MaxCompute checks whether the account is the resource owner. A resource is accessible only to its owner.
- If you access a resource as a RAM user, MaxCompute checks whether the Apsara Stack tenant account to which the RAM user belongs is the resource owner and whether the RAM user has been granted permissions on that resource.

**Note** If an Apsara Stack tenant account other than the resource owner and its RAM users are granted permissions on the resource, they can also access the resource.

MaxCompute supports the following two authorization mechanisms to control access from RAM users:

ACL-based authorization is an object-based authorization mechanism. An access control list (ACL) contains access permissions on an object. It is a resource of the object. The ACL takes effect only when the object exists. If the object is deleted, the ACL of the object is also deleted. ACL-based authorization is similar to the authorization mechanism that is implemented by using the GRANT and REVOKE statements defined in SQL-92. You can execute these statements to grant or revoke permissions on an object.

Note ACL-based authorization allows you to separately manage field-level permissions. ACL fields and tables are independent authorization objects and contain complete authorization information. You can separately grant permissions on ACL fields and tables and specify expiration time and conditions. You can separately view authorization information and revoke permissions.

Policy-based authorization is the other authorization mechanism that MaxCompute supports. An
access control policy contains access permissions on both an object and a subject. It is a resource of
the subject. Policy-based authorization can be performed on objects and subjects that do not exist
or are uncertain. The system does not automatically change or delete policies associated with an
object when the object is deleted. MaxCompute uses a custom language to specify access control
policies for objects.

#### Hierarchical authorization of management permissions

MaxCompute supports hierarchical authorization of management permissions. Permission objects and operations are used to define permissions on management operations. You can use existing access control mechanisms, such as policy-based or ACL-based authorization, to control permissions on management operations.

Permission objects include policies, ACLs, project configurations, and projects. Permission operations are the actions that are performed on permission objects, such as the PUT action in a PUT policy and the CREATE action on projects.

Fine-grained management permissions support more complex access control scenarios, such as allowing only specific clients to perform management operations.

# Fine-grained column-level authorization based on permission model 2.0

To implement fine-grained permission management and enhance data security, MaxCompute provides fine-grained column-level authorization based on permission model 2.0.

MaxCompute provides centralized management and query APIs for fine-grained column-level authorization based on permission model 2.0.

- MaxCompute and DataWorks allow for fine-grained tenant authorization and access control on data shared across tenants by using packages.
- MaxCompute allows you to authorize users to access tables and track the queries on those tables.

The following examples show how to configure fine-grained column-level authorization based on permission model 2.0.

**Note** Fine-grained authorization cannot be implemented by using access control policies because they cannot ensure data security. The following examples show how to implement fine-grained authorization by using ACLs.

Syntax for ACL-based fine-grained authorization

• Grant or revoke permissions on columns in a table within a project

grant <privileges> on table <name>(<column\_list>) to|from USER|ROLE <user|role name>; -- Grant permissions.

revoke <privileges> on table <name>(<column\_list>) to|from USER|ROLE <user|role name>;

-- Revoke permissions.

#### Description:

- If you grant permissions on a table, permissions are granted on all columns in the table, including the added columns and columns whose names are changed.
- If you grant or revoke permissions on columns, permissions on the columns can be differentiated from other permissions or merged with similar permissions.
  - If you grant the SELECT permission on col1 and col2 and grant the DESCRIBE permission on col2 and col3, both authorization statements are valid.
  - If you grant the SELECT permission on col1 and col2 and grant the SELECT permission on col3 and col4, the SELECT permission takes effect on col1, col2, col3, and col4.
- Only project owners and other authorized users can grant permissions.
- Add columns to a package across projects

add table <name>(<column list>) to package pkgdel1 with privileges <privilege list>;

**Note** If you add the same column to a package multiple times, permissions on the column can be differentiated from other permissions or merged with similar permissions.

#### Description:

- The syntax to allow the addition of columns to the package remains unchanged.
- The syntax to install or uninstall the package remains unchanged. However, permissions on added columns must be taken into account.
- o Only project owners and other authorized users can grant permissions.
- Grant or revoke permissions on columns across projects

grant <privileges> on table <name>(<column\_list>) to|from USER|ROLE <user|role name>
PRIVILEGEPROPERTIES("refobject"="true", "refproject"="<pri>project\_name>", "package"="<package name>
");

-- Grant permissions.

revoke <privileges> on table <name>(<column\_list>) to|from USER|ROLE <user|role name> PRIVILEGEPROPERTIES("refobject"="true", "refproject"="project\_name>", "package"="<package name>").

-- Revoke permissions.

#### Description:

- The rules used to grant or revoke permissions within a project take effect for cross-project authorization only when a table is added to multiple packages.
- You can grant or revoke permissions only on columns that are added to the packages. However, permissions on columns in the tables that are added to multiple packages must be taken into account.
- $\circ~$  Only project owners and other authorized users can grant permissions.

Authentication policy: Policy-based authentication uses the same logic as ACL-based authentication. Make sure that you consider how these policies interact with column-level permissions.

Permission query: You can query permissions within a project, query package permissions, and query permissions across projects.

• Query permissions within a project

show grants for <user|role name>; #The syntax does not change. However, the result is displayed at the column level.

show grants for table <name>(columns);

show grants on table <name>(columns) for user|role <name>;

**Note** If a column is specified, permissions only on this column are displayed.

• Query package permissions

```
describe package <pkg name>;
describe package <pkg name> PRIVILEGEPROPERTIES ("allowedonly"="true");
describe package <pkg name> PRIVILEGEPROPERTIES ("contentonly"="true");
```

**?** Note The preceding commands return results at the column level.

• Query permissions across projects

```
show grants for <user|role name> PRIVILEGEPROPERTIES ("refobject"="true", "refproject"="<project>"); #
The syntax does not change. However, the result is displayed at the column level.
show grants for table <name>(columns) PRIVILEGEPROPERTIES("refobject"="true", "refproject"="<project"="true");
show grants on table <name>(columns) for user|role <name> PRIVILEGEPROPERTIES ("refobject"="true", "refproject"="true");
```

Audit policy: Relevant information is included in audit logs.

#### Access control

MaxCompute supports RAM authorization.

Resource Access Management (RAM) is a resource access control service provided by Alibaba Cloud. You can use your Apsara Stack tenant account to create RAM users and grant them permissions to access specific resources owned by the account.

# 1.23.1.3. Data security

This topic describes data security for MaxCompute platform security.

MaxCompute has passed an independent third-party audit on compliance with the trust services criteria for security, availability, and confidentiality established by American Institute of Certified Public Accountants (AICPA).

Apsara Stack provides a flat storage system in which a linear address space is divided into chunks. Each chunk is replicated into three copies. These copies are stored on different data nodes of the storage cluster to ensure data reliability.

The data storage system for Apsara Stack involves three key components: master, chunk server, and client. Write operations in MaxCompute are processed and executed by the client. Procedure:

- 1. The client determines the chunk requested by the write operation.
- 2. The client sends a request to the master to query the chunk servers where the three chunk copies are stored.
- 3. The master returns the addresses of the chunk servers. Then, the client sends the write request to

the chunk servers.

4. If the write operation succeeds in all three chunk copies, the client returns a success message. Otherwise, the client returns a failure message.

The master takes into account the disk usage of all chunk servers in the cluster, distribution of chunk servers in different server racks, power supply status, and server load. This ensures that the three copies of a chunk are distributed on different chunk servers in different racks. This way, data is still accessible even if a chunk server or rack is faulty.

If a data node or its hard disks are faulty, some chunks may have less than three valid copies. In this case, the master replicates data between chunk servers to ensure that each chunk in the cluster has three valid copies.

All data operations in MaxCompute, such as addition, modification, and deletion, are synchronized to the three copies. This mechanism ensures data reliability and consistency.

After you delete data, the storage space is reclaimed by Apsara Distributed File System. Before the storage space is released, it is inaccessible to all users, and Apsara Distributed File System clears data from it. This provides maximum protection for your data.

# 1.23.1.4. KMS-based storage encryption

This topic describes KMS-based storage encryption for MaxCompute platform security.

#### **Background information**

As MaxCompute is deployed more and more widely around the world, its security requirements increase and the need to protect sensitive data, such as user privacy and financial information, becomes more and more crucial. MaxCompute must implement data-at-rest encryption to meet compliance and regulatory requirements.

MaxCompute provides storage encryption. It uses Apsara Distributed File System to encrypt, store, decrypt, and manage user data, and uses Key Management Service (KMS) to ensure the security of user data and keys.

MaxCompute storage encryption provides an additional layer of security and minimizes the damages caused by data loss. Even if encrypted data is lost or stolen, no meaningful content can be extracted from the data.



- MaxCompute storage encryption enables transparent encryption and decryption by using customer master keys (CMKs) to simplify user operations.
- MaxCompute allows you to configure the storage encryption settings by project. After encryption settings are configured for a project, all data subsequently written to the project is encrypted to reduce security risks.
- MaxCompute storage encryption is backward compatible. It allows unencrypted projects to be encrypted and allows both encrypted and unencrypted data to coexist in a project.

#### Description

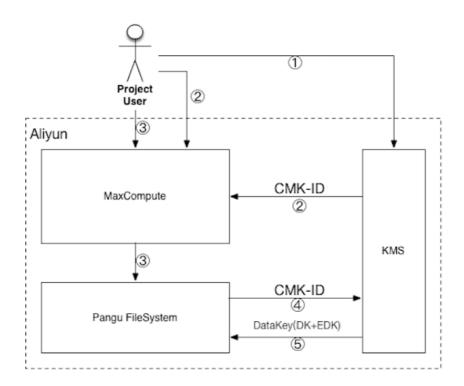
• MaxCompute uses projects as its basic operational unit and stores the table data of projects in encrypted form. Only full table encryption is supported. Resources and volumes cannot be encrypted.

- The types of tasks that support storage encryption are SQL Task 2.0, including the service mode, MergeTask, and Tunnel. After storage encryption is enabled for a project, table data written by using these types of tasks is stored in encrypted form.
- AES-CTR, AES-256, RC4, and SM4 encryption algorithms are supported.
- MaxCompute is connected to KMS to ensure key security. You must activate KMS to generate and manage keys for encryption and decryption. After you submit a request to enable storage encryption, MaxCompute automatically connects to KMS to generate the keys required for encryption.
  - **?** Note Projects that belong to the same project owner share the same key.
- You can read encrypted and unencrypted data without the need to change task types. Encrypted and unencrypted data can coexist in a project.

### Procedure to encrypt data

- 1. Activate KMS.
- 2. Enable MaxCompute storage encryption.
  - **Note** When you enable storage encryption, the system asks KMS to create a CMK. The CMK is used to protect the data key used for encryption.
- 3. After KMS is activated and storage encryption is enabled, submit jobs in MaxCompute for processing. After the jobs are complete, MaxCompute uses Apsara Distributed File System to encrypt the data for storage.
- 4. Apsara Distributed File System provides KMS with the created CMK to obtain the data key used for encryption.
- 5. The data key obtained from KMS consists of a Data Key (DK) and an Enveloped Data Key (EDK). A DK is a plaintext key used to encrypt data, and an EDK is a ciphertext key generated after envelope encryption is implemented on the DK. After Apsara Distributed File System encrypts data by using the DK, it stores the encrypted data and EDK to complete the data encryption process.

The following figure shows how to encrypt data.

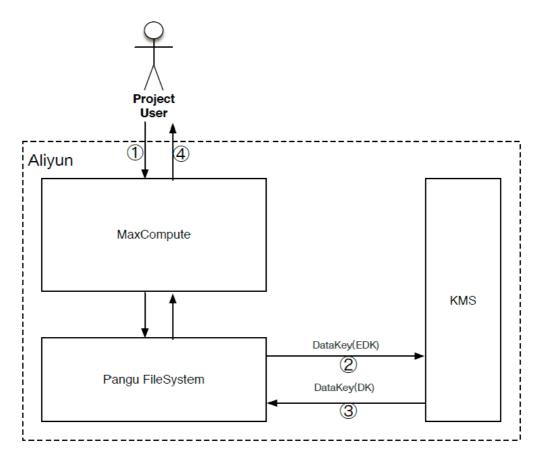


#### Procedure to process encrypted data

If you use MaxCompute to process encrypted data, the system automatically decrypts the data. You do not need to perform other operations to decrypt data.

- 1. Submit a MaxCompute job to process encrypted data.
- 2. Apsara Distributed File System reads the EDK of the encrypted data and sends it to KMS to obtain the DK.
  - Note To ensure data security, you cannot directly use the EDK to decrypt data.
- 3. Apsara Distributed File System decrypts data based on the received DK.
- 4. MaxCompute processes the decrypted data and returns the results.

The following figure shows how to process the encrypted data.



#### Use of CMKs for storage encryption

MaxCompute supports CMKs to meet your business and security requirements in different scenarios. When you create a project, you can specify a CMK to encrypt data.

Procedure to use a CMK to encrypt and decrypt data:

- 1. When you create a project, send a request to enable storage encryption.
- 2. Specify the CMK ID used by the project.

You can use MaxCompute to create a CMK or use the CMK that you created or uploaded in KMS.

Note If you use the CMK that you created or uploaded in KMS, you must authorize MaxCompute to use the CMK.

- 3. Select an encryption algorithm.
- 4. Configure the other settings required to create the project.
- 5. After the project is created, storage encryption takes effect. Data written to MaxCompute by using SQL and Tunnel tasks is stored in encrypted form.

#### Usage notes

If you enable storage encryption for a project, take note of the following rules:

- You must activate KMS by using your Apsara Stack tenant account. If KMS is not activated, a message appears, indicating that storage encryption cannot be enabled.
- When you submit a request to enable storage encryption for a project, you must specify an

encryption algorithm. If you do not specify the encryption algorithm, the default algorithm AES-CTR is used.

- A MaxCompute production engineer must enable storage encryption for a project and specify the encryption algorithm by using AdminConsole. Then, AdminConsole automatically accesses KMS to generate the CMK required for encryption.
- After the project configuration takes effect, the table data that is generated or imported by using the tasks that support storage encryption is stored in encrypted form.
- User data stored before storage encryption is enabled cannot be automatically encrypted. If automatic encryption of the data is required, you must write the data again after storage encryption is enabled.

#### Special notes

- After storage encryption takes effect, data encryption and decryption operations are automatically performed. No additional operations are required to use the data.
- In addition to the supported task types mentioned in the "Description" section, other task types such as OpenMR can be used but do not support data encryption.
- After storage encryption takes effect for a project, you can query your keys in KMS. However, you cannot modify the keys or encryption algorithms.
- Storage encryption can be disabled. After storage encryption is disabled, new data is no longer stored in encrypted form. Encrypted data remains encrypted until it is overwritten. The existing encrypted data can still be read in KMS.
- MaxCompute must use an STS token of a RAM role to access your keys. Therefore, you must evaluate the load on RAM.

#### Remarks

You can configure the storage encryption settings in the Apsara Uni-manager Management Console and Apsara Uni-manager Operations Console. You can enable storage encryption when you create a project in the Apsara Uni-manager Management Console and encrypt data when you manage projects in the Apsara Uni-manager Operations Console.

# 1.23.2. Tenant security

## 1.23.2.1. Cross-project resource sharing

This topic describes cross-project resource sharing for MaxCompute tenant security.

Assume that you are the owner or administrator who assumes the admin role of a project and a user requests to access the resources of your project. If the user belongs to your project team, we recommend that you grant permissions to the user by using the authorization management feature. If the user does not belong to your project team, you can share resources with the user across projects by using packages.

You can share data and resources across projects by using packages.

The administrator of Project A can create a package that includes all objects required by Project B. Then, the administrator of Project A can grant Project B the permissions to install the package. Then, the administrator of Project B installs the package in Project B and determines whether to grant the permissions on the package to other users in Project B.

Examples of operations that can be performed by package creators and users:

#### For package creators

create package <pkgname>;

-- Create a package.



- Only project owners have the permissions to perform this operation.
- A package name can be up to 128 characters in length.

add project\_object to package package\_name [with privileges privileges] remove project\_object from package package\_name project\_object ::= table table\_name | instance inst\_name | function func\_name | resource res\_name privileges ::= action\_item1, action\_item2, ... -- Add resources to the package.

### ? Note

- o A project is not a valid object. A project cannot be added to a package.
- In addition to objects, the operation permissions on the objects are also added to the
  package. If you do not use [with privileges privileges] to specify permissions, the object is
  read-only. Only the READ, DESCRIBE, and SELECT permissions are granted on the object. An
  object and its permissions are inseparable and cannot be changed after you add them to
  a package. If you want to change them, you must remove the object from the package
  and add it again.

allow project <pri>priname> to install package <pkgname> [using label <number>];

-- Grant the permissions on the package to another project.

disallow project <pri>priname> to install package <pkgname>;

-- Revoke the permissions on the package from another project.

delete package <pkgname>;

-- Delete a package.

show packages;

-- Query the packages.

describe package <pkgname>;

- -- Query details about a package.
- For package users

install package <pkgname>;

-- Install a package.



- Only project owners have the permissions to perform this operation.

#### uninstall package <pkgname>;

-- Uninstall a package.

Note To uninstall a package, you must specify pkgname in the following format: <project Name>.<project Name>.

#### show packages;

-- Query the packages that have been created and installed.

#### describe package <pkgname>;

-- Query details about a package.

An installed package is an independent object in MaxCompute. To access resources in a package shared by the user of another project, you must have read permissions on the package. If you do not have the read permissions on the package, you must send a request to the project owner or administrator to apply for the permissions. The project owner or administrator can use ACL-based or policy-based authorization to grant permissions.

Use ACL-based authorization to allow odps\_test@aliyun.com to access the resources in the package. Example:

use prj2; install package prj1.testpkg; grant read on package prj1.testpackage to user aliyun\$odps\_test@aliyun.com;

Use policy-based authorization to allow users in the prj2 project to access the resources in the package. Example:

use prj2; install package prj1.testpkg; put policy /tmp/policy.txt;

```
Note /tmp/policy.txt contains the following data:

{
  "Version": "1",
  "Statement":
  [{
    "Effect":"Allow",
    "Principal":"*",
    "Action":"odps:Read",
    "Resource":"acs:odps:*:projects/prj2/packages/prj1.testpkg"
  }]
  }
}
```

### 1.23.2.2. Column-level access control

This topic describes column-level access control for MaxCompute tenant security.

Label-based security (LabelSecurity) is a mandatory access control (MAC) policy for a project. It allows project administrators to control user access to sensitive data at the column level.

LabelSecurity classifies both data and users who want to access the data into different levels. Data is classified into the following levels based on its sensitivity:

- Level 0: unclassified
- Level 1: confidential
- Level 2: sensitive
- Level 3: highly sensitive

MaxCompute adopts the preceding sensitivity levels. Project owners must define their own standards to determine the data sensitivity levels and access permission levels. By default, the data sensitivity level and access permission level of all users are 0.

LabelSecurity allows project administrators to label table columns and views. A table can have columns with different sensitivity levels. By default, the sensitivity level of a new view is 0. The sensitivity levels of views and tables that correspond to the views are independent of each other.

LabelSecurity applies the following default security policies based on the sensitivity levels of data and users:

- No-ReadUp: Users are not allowed to read data that has higher sensitivity levels than their levels, unless they are explicitly authorized.
- Trusted-User: Users are allowed to write data to columns regardless of the sensitivity levels. The default sensitivity level of a new column is 0.

### ? Note

- Traditional MAC systems use complex security policies to prevent unauthorized data operations in projects. For example, the No-WriteDown policy only allows a user to write data to columns that have higher sensitivity levels than the user level. By default, MaxCompute does not support the No-WriteDown policy. This reduces the costs for project owners to manage data sensitivity levels. Project owners can set SetObjectCreatorHasGrantP ermission to false to implement a policy similar to No-WriteDown.
- If you want to prevent data transfer between projects, you can enable project protection. After the settings take effect, users are able to access data only within their own projects and data cannot be transferred to other projects.

LabelSecurity is disabled for a project by default. Project owners can enable it based on business requirements. After LabelSecurity is enabled, the preceding default security policies take effect. In this case, users must have the SELECT permission and the required level to read sensitive data in the tables.

Run the following command to enable or disable LabelSecurity:

Set LabelSecurity=true|false;

- -- This command is used to enable or disable LabelSecurity. The default value is false.
- --Only project owners can run this command. Other operations can be performed by users who assume the admin role.

### 1.23.2.3. Project protection

This topic describes project protection for MaxCompute tenant security.

Users who are authorized to access data in multiple projects can transfer data across these projects. If a project contains highly sensitive data that cannot be shared with other projects, the administrator can set projectProtection to true to allow only inbound data flows.

Configuration command:

#### set projectProtection=true

-- This command allows only inbound data flows.

Note By default, projectProtection is set to false. You must manually set it to true to enable project protection.

### Data transfer across projects after project protection is enabled

After project protection is enabled for your project, you may need to transfer the data of a table to another project and make sure that the table does not contain sensitive data. MaxCompute provides two methods for you to transfer the data:

Configure an exception policy

When a project owner enables project protection, the owner can run the following command to configure an exception policy:

set ProjectProtection=true WITH EXCEPTION <policyFile>

**Note** An exception policy is different from policy-based authorization despite the fact that both operations have the same syntax. This exception policy describes an exception in the project protection mechanism. Access requests that meet the description of the exception policy can ignore project protection rules.

#### Example:

-- Allow the Alice@aliyun.com user to perform a SELECT operation on the alipay.table\_test table and tran sfer data out of the alipay project.

### ? Note

- The exception policy is not a common authorization method. In this example, if Alice does
  not have the SELECT permission on the alipay.table\_test table, Alice cannot transfer data
  even if the preceding exception policy is configured.
- Project protection is a method to control data transfer, not to control data access. Data transfer control is effective only if users can access their desired data.

#### Configure trusted projects

If the current project is protected, data can be transferred to its trusted project. The data transfer does not violate the project protection rules. If multiple projects are specified as trusted projects for each other, they form a trusted project group. Data can be transferred only within the group.

Commands to manage trusted projects:

```
list trustedprojects;
-- Query all trusted projects added to the current project.
add trustedproject <projectname>;
-- Add a trusted project to the current project.
remove trustedproject <projectname>;
-- Remove a trusted project from the current project.
```

### Resource sharing and project protection

MaxCompute supports both package-based resource sharing and project protection. However, their features are mutually exclusive.

Resource sharing takes precedence over project protection. If a data object is shared with users in another project, the data object is not limited by project protection.

### More checks to prevent data from being transferred out of projects

To prevent data from being transferred out of projects, set projectProtection to true and verify the following items:

- No trusted projects are added. If a trusted project is added, assess the potential risks.
- No exception policies are configured. If an exception policy is configured, assess the potential risks, especially data leaks caused by time-of-check to time-of-use (TOC2TOU).
- No data is shared by using packages. If a package is used to share data, make sure that the package does not contain sensitive data.

### 1.23.2.4. Log audit

This topic describes log audit for MaxCompute tenant security.

MaxCompute allows you to audit logs generated for different users and stores the logs in metadata warehouses.

In a metadata warehouse, MaxCompute is used to analyze its running status, and metadata stored in MaxCompute is aggregated to a table. This allows you to query and collect statistics on the metadata. The metadata includes static data, operational logs, and security information.

- Static data is permanently written to the data warehouse.
- Operational logs record task processes and are stored only in one partition.
- Security information originates from Tablestore and includes whitelists and ACLs.

### 1.24. DataWorks

# 1.24.1. Permission isolation for development and production environments

DataWorks manages code and configurations by workspace. Two workspace modes are available, which are Basic Mode and Standard Mode.

A workspace created in Standard Mode can isolate the development and production environments. Take the MaxCompute engine as an example. A workspace created in Standard Mode requires two MaxCompute projects: one for the development environment and the other for the production environment. Data in the two environments is completely isolated from each other.

Developers can only operate development environment data on the DataStudio page of DataWorks. Changes to production environment data take effect only after an administration expert publishes the changes. A workspace created in Standard Mode allows you to strictly control table permissions. Developers are prohibited from arbitrarily operating tables in the production environment to guarantee data security.

The development and production environments are integrated for a workspace created in Basic Mode. This type of workspace features fast iteration. Code takes effect immediately after being submitted, without requiring you to publish it. However, permissions of the development and production environments are not isolated.

### 1.24.2. Authentication and authorization

### 1.24.2.1. Access control

### Logon control

You can use your Alibaba Cloud account to log on to the Resource Access Management (RAM) console and create multiple RAM users. By creating policies and attaching them to RAM users, you can enable RAM users who meet specified conditions to access DataWorks. For example, you can specify that only RAM users who use the specified IP address or Classless Inter-Domain Routing (CIDR) block, enable multifactor authentication (MFA), and use the HTTPS access protocol can access DataWorks.

By specifying the IP addresses or CIDR blocks that have access to DataWorks, you can further prevent unauthorized access and ensure data and business security. For example, when your AccessKey is inadvertently lost or stolen, DataWorks can prevent access from unauthorized IP addresses (such as IP addresses that are not in your internal network) before you create a new AccessKey.

#### Sandbox isolation

A workspace is a basic unit for isolating user data in DataWorks. All nodes in the workspace run in the sandbox to prevent data leakage. Sandbox isolation also prevents developers from jeopardizing third-party data stores on the public network due to unauthorized use of external resources. By default, DataWorks only allows the following access scenarios:

- In DataStudio, developers can only access a specified compute engine.
- In Data Integration, developers can only access data stores that have been registered.

If developers need to access external resources outside the workspace in addition to the above two scenarios, the workspace administrator must add the resources to the sandbox whitelist in advance.

# 1.24.2.2. Permission management

### Role management

DataWorks defines seven roles in permission management, including owner, administrator, developer, O&M engineer, deployment engineer, security administrator, and guest.

Role	Permission description
Owner	Indicates the user who owns a workspace. The owner has all permissions of a workspace.
Administrator	Indicates an administrative user entrusted by the owner. An administrator has all permissions of a workspace except for deleting the workspace.
Developer	Indicates a user who operates the development environment. A developer has the permissions to develop nodes and workflows and operate data in the development environment.

Role	Permission description
O&M engineer	Indicates a user who operates the production environment. An O&M engineer has the permissions to terminate, rerun, and deploy nodes in the production environment.
Deployment engineer	Indicates a user who connects the development and production environments. A deployment engineer has the permissions to publish code from the development environment to the production environment.
Security administrator	Indicates a data security manager. A security administrator has the permissions to manage the configuration in the Data Protection module.
Guest	Indicates a user with the minimum permission. A guest can only view code instead of performing any other operations.

### Permission management

DataWorks allows you to manage data permissions on a workspace. You can authorize permissions by table or field and view and audit permissions.

#### Data download control

DataWorks gives you full control over configuration data download to reduce the risk of data leakage and ensure data security.

# 1.24.3. Data encryption

All underlying data of DataWorks is encrypted for storage and transmission, including user code, workflow configuration, and data store connection information. Only authorized users can view, use, and modify the data.

# 1.24.4. Sensitive data protection

DataWorks supports data identification, sensitive data discovery, data classification and grading, desensitization, access monitoring, risk discovery and alerting, and audit.

- Data identification: automatically identifies sensitive data in a workspace based on preset rules.
- Data classification and grading: allows you to define different levels of data confidentiality and provide separate access control permissions for each level.
- Desensitization: desensitizes sensitive data by masking, aliases, and hashing.
- Access monitoring: monitors the access to and export of sensitive data.
- Risk discovery: monit ors sensitive dat a access behavior in specific scenarios.

# 1.25. Realtime Compute

# 1.25.1. Platform security

### 1.25.1.1. Resource isolation

Realtime Compute projects are isolated based on permissions. Only authorized users can access or perform operations on a project and its sub-products.

Realtime Compute allows you to isolate resources at the project level. For example, if a sharp increase in streaming data inputs results in a higher CPU usage for the job of a user's project, the CPU usage of your job is not impacted. This is enabled by the application of virtualization technologies at the underlying layer of Realtime Compute.

### 1.25.1.2. Authentication and authorization

### Realtime Compute accounts

You can only log on to the Realtime Compute console using Alibaba Cloud accounts, which are managed based on the username, password, and signature key. The accounts comply with the existing security system of Alibaba Cloud. To ensure the security of accounts, the HTTPS protocol is used for transmission.

#### Data store accounts

In Realtime Compute, the accounts of data stores are required to create source and result tables. We provide Resource Access Management (RAM) and Security Token Service (STS) to prevent your business data from leaking due to the loss of account information.

### 1.25.1.3. Data security

Realtime Compute ensures the security of Realtime Compute system data and business data.

### System data security

The security of system data is ensured by Realtime Compute. The following measures have been taken to ensure the security of system data:

- The HTTPS protocol is used for transmission.
- The Advanced Encryption Standard (AES) is used to encrypt the information about the connection with data stores. This helps to prevent sensitive information from leaking.
- Comprehensive attack tests have been performed to ensure high-level security.

### **Business data security**

Realtime Compute does not store business data of users. The security of business data is ensured by external Alibaba Cloud data stores. For more information, see security models and best security practices of Alibaba Cloud data stores.

## 1.25.1.4. Business process

### **Business process security**

Realtime Compute defines a strict development process for streaming data analysis, and provides separate pages for data development and administration in its console. This guarantees a complete and secure business process while minimizing adverse effects on user experience.

Code versions

Realtime Compute allows you to compare code versions and roll back to an earlier version. This helps you trace the code and troubleshoot faults.

• Standalone debugging tool in the IDE

Realtime Compute offers a debugging tool in the integrated development environment (IDE), which allows you to debug the code without affecting online data. With this tool, you can specify data for source tables, dimension tables, and result tables to create a job, and then debug the data offline. This ensures that running jobs are not affected.

Job publishing process

Realtime Compute offers a job publishing process that prevents running jobs from being affected by the changes of offline code. After you debug the new code, you can publish the job and view it on the Administration page of the Realtime Compute platform. The running job does not automatically use the new code. Instead, you must confirm the changes, terminate the running job, and start the job again using the new code. In this way, you can exercise a complete control over the code that is used for the job to be published.

# 1.26. Machine Learning Platform for AI

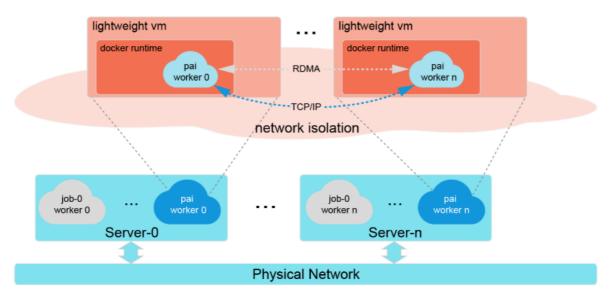
# 1.26.1. Security isolation

This topic describes the security isolation technology of Machine Learning Platform for AI.

#### Tenant isolation

When multiple tenants use Apsara Stack Machine Learning Platform for AI, the computing and storage resources assigned to each tenant are isolated and secured by using the multi-tenancy technology.

• Network virtualization plays a crucial role in tenant isolation. Based on the virtual network established at the upper layer of the physical network, Machine Learning Platform for AI ensures that computing resources used for different algorithm jobs are isolated. This allows each tenant to independently manage computing and storage resources in a distributed cluster. When you submit a distributed deep learning job, the system uses the PCle technology to connect a GPU instance to a Docker instance, as shown in the following figure. This guarantees the performance of GPU computing without compromising resource isolation.



When you activate Machine Learning Platform for AI, the DataWorks service is also activated.
 DataWorks creates a dedicated gateway resource group for each tenant based on centralized resource management. The gateway resource group allows you to securely run SQL, MapReduce, and other types of tasks. This also avoids mutual impacts among different tenants.

#### Service isolation of EAS

Elastic Algorithm Service (EAS) stores AccessKey pairs of tenants who deploy services. The AccessKey pairs are stored in ApsaraDB RDS. EAS verifies each request and denies requests that fail the verification. EAS uses the container technology to isolate resources. The resources used by a tenant, such as memory and CPU resources, are limited. Therefore, services deployed by different tenants on the same server do not affect each other.

# 1.26.2. Authentication

## 1.26.2.1. Identity verification

This topic describes the identity verification mechanism for Machine Learning Studio and Elastic Algorithm Service (EAS) of Machine Learning Platform for AI.

### Identity verification for Machine Learning Platform for AI

When you access a Machine Learning Platform for AI application from a browser, your identity is verified in the following steps:

- All requests from the browser are transmitted to LoginFilter. LoginFilter calls the API of the DataWorks SSO module to verify your identity. If you have not logged on to the Apsara Unimanager Management Console, you are redirected to the logon page. After you log on, the logon information is written to the related cookies.
- 2. Each request carries a token that is used to prevent Cross-Site Request Forgery (CSRF) attacks.
- 3. All requests to add, delete, modify, or query resources are verified to ensure that you have permissions to access the resources. This prevents unauthorized actions.
- 4. When the DataWorks scheduling module or other third-party modules call the API of Machine Learning Platform for AI, the token center of DataWorks is used to verify user permissions for all requests.

### Identity verification for EAS

You can obtain an AccessKey pair in the console. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey ID is public and is used to identify a user. The AccessKey secret is private and is used to verify the identity of a user.

When you send a request to EAS, your identity is verified in the following steps:

- 1. You generate a signature string from the request to be sent. The signature string must be in the format that is required by EAS.
- 2. You use your AccessKey secret and the HMAC algorithm to encrypt the signature string and then generate a verification code. The verification code carries a timestamp to prevent replay attacks.
- 3. After EAS receives the request, it finds the AccessKey secret that corresponds to the AccessKey ID. Then, EAS generates a signature string and a verification code from the request in the same way.
  - If the verification code generated by EAS is identical to the given one, EAS considers the request valid.
  - If the verification code generated by EAS is different from the given one, EAS denies the request and returns an HTTP 403 error.

### 1.26.2.2. Permission control

This topic describes the permission control policies of Machine Learning Platform for AI.

After you use Elastic Algorithm Service (EAS) to deploy a service as an API, the system generates a token that is used to call the API. You must provide this token when you call the API. This token is private. Make sure that the token is kept only by the service consumer.

### 1.26.2.3. RAM and STS

Resource Access Management (RAM) is a resource access control service provided by Apsara Stack. When you use RAM, you do not need to share the AccessKey pair of your Apsara Stacktenant account with other tenants, and you can grant the least privilege to RAM users. RAM allows you to use your Apsara Stacktenant account to create RAM users and assign resource access permissions to the RAM users. All RAM users that you create belong to your Apsara Stacktenant account and all resources created by the RAM users belong to the Apsara Stacktenant account.

Security Token Service (STS) is a temporary access token service provided by Apsara Stack to manage temporary access permissions. STS can generate a temporary access token for a user. The access permissions and expiration date of the token are defined by the user. The access token automatically expires upon the expiration date.

When you use the deep learning module, you can go to the OSS quick authorization page in the Apsara Uni-manager Management Console to perform quick authorization.

- 1. OSS quick authorization grants read and write permissions on an OSS bucket to the MaxCompute service account odps.aliyuncs.com.
- 2. After the service account is authorized, the role **AliyunODPSPAIDef ault Role** is created in RAM. Each role has a globally unique resource identifier named **RoleArn**. The format is acs:ram::\$account ID:role/\$roleName.
- 3. After the role is created, Machine Learning Platform for AI uses the AccessKey pair of the account odps.aliyuncs.com to call the AssumeRole operation of STS.
- 4. If the call is successful, you obtain a temporary AccessKey pair (AccessKeyId and AccessKeySecret)

and a temporary STS token (SecurityToken). The default validity period is 3,600 seconds. After you obtain the temporary AccessKey pair and STS token, you can read data from and write data to the OSS bucket where your project is deployed.

AliyunODPSPAIDefault Role is described in the following code block:

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

"Version": "1"
```

The following code block describes the permissions that are granted to the role:

```
{
"Version": "1",
"Statement": [
{
   "Action": [
     "oss:GetObject",
     "oss:ListObjects",
     "oss:DeleteObject",
     "oss:Parts",
     "oss:PutObject",
     "oss:AbortMultipartUpload"
   ],
   "Resource": "*",
   "Effect": "Allow"
}
]
```

# 1.26.3. Data security

# Data security of Machine Learning Platform for AI applications

Machine Learning Platform for AI uses MaxCompute to compute and store big data. User data is stored in MaxCompute projects. Projects that belong to different users are isolated from each other.

For financial institutes and other scenarios that require high data security, MaxCompute provides a mechanism that allows you to enable or disable data protection for a MaxCompute project. You can use this feature to allow or prohibit data exports from a specified project.

### Data security of EAS

Elastic Algorithm Service (EAS) uses three types of data:

- EAS metadata: ApsaraDB RDS stores the metadata and ensures data security.
- Kubernetes metadata: The metadata is stored in etcd. etcd provides services and runs with three replicas. To access etcd, you need certificates that are kept by the Kubernetes cluster administrator.
- Monitoring data: The monitoring data is stored in Apsara Stack disks. The type of disk determines the type of data backup.

# 1.26.4. Log audit

Machine Learning Platform for AI provides a request log. User access records are automatically written into a designated file in the specified format based on the frequency of user access. The user access log is used for auditing or action analysis. The request log contains information such as the request time, source IP address, request method, request URL, request user ID, processing duration, and error code.

# 1.27. E-MapReduce (EMR)

# 1.27.1. Platform security

# 1.27.1.1. Security isolation

E-MapReduce (EMR) isolates user data by tenant. Tenants can obtain only the metadata settings for themselves. Metadata is associated with business data. Therefore, a tenant can query only the business data that is associated with their metadata.

### 1.27.1.2. Authentication

This topic describes Kerberos authentication and Lightweight Directory Access Protocol (LDAP) authentication supported by EMR.

#### Kerberos authentication

EMR supports Kerberos authentication. If you run open source components on a cluster in Kerberos security mode, only authenticated clients can access services, such as Hadoop Distributed File System (HDFS), on the cluster.

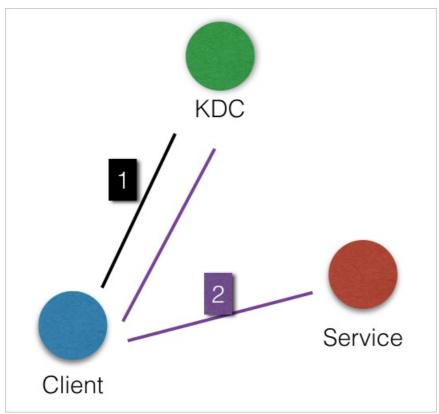
Kerberos is a network authentication protocol. EMR uses Hadoop Authentication Service (HAS) to implement Kerberos authentication. Open source big data ecosystems such as Hadoop and Spark support only built-in Kerberos for security authentication. HAS provides a new authentication method, Kerberos-based token authentication. This new authentication method enables Hadoop and Spark to connect to existing authentication and authorization systems. This way, the related authentication methods other than Kerberos can also be used on Hadoop and Spark. Kerberos-based token authentication also hides the complexity of Kerberos from users. Kerberos-based token authentication supports most components in big data ecosystems and requires few or no changes to components. The original Kerberos authentication method can still be used for all components.

#### How Kerberos authentication works

Kerberos is an identity authentication protocol based on symmetric-key cryptography. As an independent third-party identity authentication service, Kerberos provides identity authentication and supports single sign-on (SSO). After a client is authenticated, it can access multiple services, such as HBase and HDFS.

Kerberos authentication includes two steps. 1. Key Distribution Center (KDC) authenticates a client. 2. A service authenticates the client.





- KDC: indicates a Kerberos server.
- Client: indicates a client (principal) that needs to access an application service on a cluster. KDC and the service authenticate the identity of the client.
- Service: indicates a service integrated with Kerberos, such as HDFS, YARN, or HBase.

#### Authenticate a client by KDC

A client can access a service for which Kerberos authentication is enabled only after the client is authenticated by KDC. After the authentication, the client obtains a Ticket Granting Ticket (TGT). Then, the client can use the TGT to access a service for which Kerberos authentication is enabled.

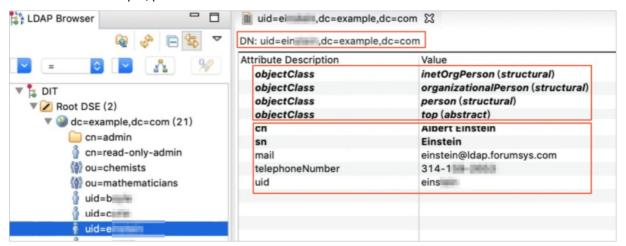
#### Authenticate a client by a service

After a client obtains a TGT, the client can use the TGT to access services. The client uses the TGT and the name of the service to be accessed such as HDFS to obtain a Service Granting Ticket (SGT) from KDC. Then, the client uses the SGT to access the service. The service uses relevant information to authenticate the client. After the authentication succeeds, the client can access the service.

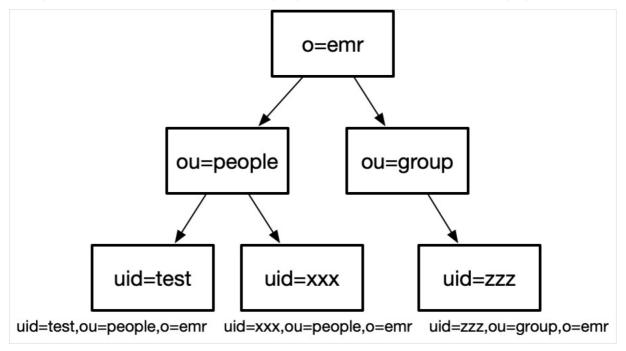
#### LDAP authentication

LDAP is short for Lightweight Directory Access Protocol. LDAP-compliant directory services include ApacheDS, Active Directory, and Red Hat Directory Service. By default, ApacheDS is built in EMR.

An LDAP directory stores data in a tree structure. Each node in the directory has a unique path name, which is called distinguished name (DN). Each node has one or more object Classes, such as Organizational Unit (ou). The object Class of a user node is person. Each object Class contains multiple attributes. For example, person contains the cn and uid attributes.



In EMR OpenLDAP, user information is stored in the ou=people,o=emr directory. For example, the storage structure of user test is *uid=test,ou=people,o=emr*, as shown in the following figure.



A component deployed in an EMR cluster supports LDAP authentication only after the component is connected to an LDAP server.

After you connect a component, such as Knox, Hue, or Ranger Admin, to an LDAP server, you can use the username and password of the LDAP server to log on to the web UI of the component.

Services such as HiveServer2, Spark ThriftServer, and Presto Coordinator can connect to an LDAP server only after LDAP authentication is enabled for them. After LDAP authentication is enabled for a service, you can access a client of the service, such as the Beeline client, and run commands only by using the username and password of the LDAP server.

### 1.27.1.3. Data security

This topic describes the data security of EMR.

Hadoop Distributed File System (HDFS) stores files as data blocks. Each data block has multiple replicas (three by default). We recommend that you store replicas on different racks. In most cases, the replication factor of HDFS is 3. One replica is placed on a node of the local rack, another replica is placed on a different node of the same rack, and the last replica is placed on a node of a remote rack.

HDFS scans the replicas on a regular basis. If a replica is found missing, HDFS copies the data and generates a new replica. If data on a node is found missing, HDFS restores all data on the node.

HDFS is a reliable data storage system that provides storage for massive amounts of data.

The synchronization interval equals the maximum time range allowed for data loss. For example, if data is synchronized on an hourly basis, the maximum time range allowed for data loss is 1 hour. If data is synchronized at 30-minute intervals, the maximum time range allowed for data loss is 30 minutes.

### 1.27.2. Access control

This topic describes the access control methods of EMR.

After you create users, you can grant permissions on components to the users. Access control is available regardless of whether authentication is enabled.

#### HDFS authorization

After HDFS authorization is enabled, only users who are granted the required permissions can access HDFS and perform operations such as reading data and creating folders.

#### • YARN authorization

YARN supports service-level and queue-level authorization.

#### o Service-level authorization

- Service-level authorization controls the permissions of specified users on cluster services, such as job submissions.
- Authorization policies are configured in the hadoop-policy.xml file.
- Service-level authorization must be verified before other authorizations, such as the authorization of the permissions to perform actions on HDFS or submit a YARN job for queuing.

#### o Queue-level authorization

YARN uses queues to control the actions that can be performed on resources. The following types of queue schedulers are available: Capacity Scheduler and Fair Scheduler.

#### • Hive authorization

Hive has the following two built-in authorization schemes. You can use both of them at the same time.

- Storage-based authorization
- SQL standards-based authorization

#### HBase authorization

If HBase authorization is not enabled, all accounts can access HBase clusters to perform all operations, such as disabling tables, dropping tables, and performing major compaction. For clusters for which Kerberos authentication is disabled, users can use a forged identity to access cluster services. This is the case even if HBase authorization is enabled. For security reasons, we recommend that you create clusters that have Kerberos authentication enabled.

#### • Kafka authorization

If Kafka authorization is enabled but Kafka authentication (such as Kerberos authentication or simple username-password cryptography) is disabled, users can use a forged identity to access cluster services. For security reasons, we recommend that you create Kafka clusters that have Kerberos authentication enabled.

#### Ranger

Apache Ranger is a centralized security framework that supports fine-grained access control across Hadoop components, such as HDFS, Hive, HBase, and Kafka. You can manage access policies in the Apache Ranger console.

## 1.28. DataHub

# 1.28.1. Platform security

### 1.28.1.1. Data isolation

This topic describes the isolation scheme of DataHub.

DataHub implements symmetric encryption based on AccessKey pairs (AccessKey ID and AccessKey secret) to verify the identity of requesters. Each HTTP request is signed and authenticated. DataHub isolates the data of different users by using the Apsara Distributed File System.

DataHub ensures that user data and metadata are stored in separate storage spaces.

### 1.28.1.2. Authentication

This topic describes the authentication methods of DataHub.

#### Authentication

You can create AccessKey pairs in the Apsara Uni-manager Management Console. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey ID identifies a user, whereas the AccessKey secret is used to authenticate the user identity. The AccessKey secret must be kept strictly confidential.

Before you send a request to DataHub, you need to create a string to be signed in the format specified by DataHub and then create a signature for the request by using the AccessKey secret. After DataHub receives the request, it identifies the AccessKey secret that corresponds to the AccessKey ID. Then, it extracts the signature string and verification code in the same way. If the extracted verification code is the same as the one received, the request is valid. Otherwise, DataHub rejects the request and returns an HTTP 403 error.

#### **ACL**

You can access DataHub resources by using Alibaba Cloud accounts or RAM users. Different RAM users can be created in one Alibaba Cloud account. An Alibaba Cloud account can grant RAM users the permission to access DataHub resources.

- When you access a DataHub resource by using an Alibaba Cloud account, DataHub checks whether the account is the owner of the corresponding resource. Only the resource owner has the permission to access the corresponding resource.
- When you access a DataHub resource as a RAM user, DataHub checks whether you are authorized by the corresponding Alibaba Cloud account to access the resource and whether the Alibaba Cloud account owns the resource. For more information about RAM authorization, see RAM and STS authorization.



Note

DataHub does not support resource authorization between Alibaba Cloud accounts.

#### Access control

DataHub supports Resource Access Management (RAM) and Security Token Service (STS) authorization.

RAM is a resource access control service provided by Alibaba Cloud. You can use your Alibaba Cloud account to create RAM users and grant them the permissions to access specific resources that are owned by the account.

STS is an Alibaba Cloud service that provides temporary access credentials. It is used for short-term access control. You can use STS to generate a temporary access credential. You can specify the permissions and validity period of the credential. The credential becomes invalid when it expires.

DataHub resources are authorized by using RAM authorization policies. You must specify Action, Resource, and Effect in a policy. The following script shows a sample policy.

```
{
"Version": "1",
"Statement": [
{
    "Action": [ "dhs:GetRecords"],
    "Resource": "acs:dhs:cn-hangzhou:1001:projects/A/topics/B",
    "Effect": "Allow"
}
]
```

Note The format of Resource is acs:dhs:{Region}:{User}:{DataHubResource}.

The preceding sample policy indicates the control of access to a DataHub resource that resides in a specific region. The policy grants RAM user 1001 the permission to read data from Topic B of Project A that resides in the China (Hangzhou) region.

Dat aHub supports fine-grained permission control policies. You can control resource access based on your needs.

### 1.28.1.3. Data security

This topic describes the data security scheme of DataHub.

Apsara Stack uses a flat network in which a linear address space is divided into slices. Slices are also called chunks. For each chunk, three replicas are created and stored on different nodes in the cluster to ensure dat a reliability.

The triplicate technology used for the Apsara Stack data system involves three key components: master, chunk server, and client. Each write operation in DataHub is executed by the client. The following steps show the execution process.

- 1. The client determines the chunk of the write operation.
- 2. The client sends a request to the master to guery the chunk servers where the three chunk replicas are stored.
- 3. The master returns the addresses of the chunk servers, and then the client sends the write request to the chunk servers.
- 4. If the write operation succeeds in all three chunk replicas, the client returns a success message. Otherwise, the client returns a failure message.

The master analyzes the following information: the disk usage of all chunk servers in the cluster, distribution of chunk servers in different switch racks, power supply status, and machine load. The master ensures that the three replicas of a chunk are distributed on different chunk servers in different racks. This effectively prevents a single point of failure caused by the failure of a chunk server or rack.

If a data node or its hard disks are faulty, the total number of valid replicas of some chunks may become less than three. In this case, the master replicates data between chunk servers to make sure that each chunk in the cluster has three valid replicas.

All operations on data in DataHub, including inserting, updating, and deleting data, are synchronized to the three replicas. This mechanism guarantees data reliability and consistency.

In addition, when data is deleted, the released storage space is reclaimed by Apsara Distributed File System. During this period, the space is inaccessible for all users. Data erasure is performed on the space before it is released for further usage. This mechanism provides a high level of protection for user data.

# 1.28.2. Tenant security

# 1.28.2.1. Log auditing

This topic describes the log auditing scheme of DataHub.

DataHub logs various types of information for security and troubleshooting purposes. DataHub creates topics to store log data, such as queries per second (QPS), request duration, processing duration, source IP address, and status code. DataHub displays some of the statistics in the console to help you analyze issues.

# 1.29. Quick BI

### 1.29.1. Tenant isolation

This topic describes the tenant isolation feature of Quick BI.

As a multi-tenant SaaS product on the cloud, Quick BI supports tenant isolation. Quick BI maps a tenant to an organization which is the most basic concept of Quick BI. Any data, resource, or user must belong to an organization. Data, resources, and users of an organization are isolated from those of other organizations.

# 1.29.2. Authentication and authorization

This topic describes the authentication and authorization feature of Quick Bl.

## 1.29.2.1. Identity authentication

Quick BI supports two access modes: logon mode and logon-free mode.

### Logon mode

You must create an account in the Apsara Uni-manager Management Console, log on to the console, and then access Quick BI.

In logon mode, Quick BI carries the session ID for each request. After Quick BI receives a request, it verifies the identity of the requestor and provides services only after the verification is passed. If the verification fails, Quick BI does not provide services and the logon page appears.

### Logon-free mode

You can create an AccessKey pair in the Apsara Uni-manager Management Console. An AccessKey pair consists of an AccessKey ID and an AccessKey secret.

- The AccessKey ID is a public key used to identify users.
- The AccessKey secret is a private key used to authenticate users.

Before you send a request to Quick BI, you must generate a signature string for the request in the format specified by Quick BI. Then, you must encrypt the signature string by using your AccessKey secret to generate a verification code based on the HMAC algorithm. The verification code is timestamped to prevent replay attacks.

After Quick BI receives the request, it finds the AccessKey secret based on the AccessKey ID, and extracts the signature string and verification code. If the obtained verification code is the same as that in the request, the request is considered valid. Otherwise, Quick BI rejects the request.

# 1.29.2.2. Permission management

Quick BI supports two permission management methods: data object permission management and row-level permission management.

### Data object permission management

Data objects include data sources, datasets, workbooks (including classic workbooks), dashboards, and BI portals. Data objects are managed by workspace. Users in a workspace can be granted the following permissions:

- Management permission: allows you to manage workspace members and resources.
- Development permission: allows you to manage data sources, datasets, and reports, and view reports in the workspace.
- Analysis permission: allows you to manage reports and view all reports in the workspace.
- View permission: allows you to view all reports in the workspace.

### Row-level permission management

Quick BI allows you to configure row-level permissions for fields in a dataset, so that different users can view only the data within their permissions.

If row-level permission management is enabled for a field in a dataset, users must have permissions on this field to access the charts generated based on this dataset.

### 1.29.2.3. RAM authentication

Quick BI supports RAM authentication.

Resource Access Management (RAM) is a resource access control service provided by Alibaba Cloud. It allows you to create RAM users of an Apsara Stack tenant account and grant them permissions to access resources.

# 1.29.3. Data security

This topic describes the data security feature of Quick BI.

Data security of Quick BI involves user business data and Quick BI metadata.

User business data refers to the data that users analyze by using Quick BI. Quick BI provides real-time online analysis services to connect to data sources. User business data is not stored in disks, which does not cause data security risks.

Quick BI metadata refers to the data objects created by users in the Quick BI console. These data describe only the processing procedures and presentation forms of business data. Quick BI metadata is stored in a mini RDS database. This is an industrial-grade database that does not provide external access interfaces. The security and stability of this database are guaranteed by a dedicated team, thus the data security risks are extremely low.

# 1.29.4. Data encryption

This topic describes the data encryption feature of Quick Bl.

Sensitive user data, including user information and data source configurations, is encrypted by using the envelope encryption technology of KMS. All Quick BI features use HTTPS to ensure secure data transmission. This meets the requirements for encrypted transmission of sensitive data.

# 1.29.5. Audit logs

This topic describes the audit logs feature of Quick BI.

Quick BI provides detailed audit logs to record user behavior, such as report access, report editing, and authorized operations. A single behavior is recorded as an event. An audit log contains the organization and the workspace in which the event occurs. It also contains the operator, operation type, object, source IP address, time, and operation range of the event.

If you have activated ActionTrail, Quick BI can deliver events to ActionTrail based on organization owners for professional auditing.

# 1.29.6. Security best practices

This topic describes the security best practices of Quick BI.

### Access the system with a non-super administrator account.

After you purchase Quick BI and create an organization in Quick BI as a super administrator (quickbi\_admin), we recommend that you use a RAM user to perform daily development and O&M. You can grant the RAM user the required permissions.

### Create workspaces and specify administrators for the workspaces

Quick BI uses workspaces to manage data objects and users in organizations. Generally, a workspace corresponds to a theme, a project, or a real department of an enterprise. A workspace administrator can manage most of the tasks of the super administrator, including data object management and workspace member management.

### Grant minimum permissions to workspace members

A Quick BI workspace provides the following roles: administrator, developer, analyst, and viewer. You can attach roles to workspace members to grant them minimum permissions.

- For users who need only to view reports, attach the viewer role to them.
- For users who need to analyze data and develop reports, but not create or modify dataset models, attach the analyst role to them.
- For users who need to create or modify dataset models and add data sources, attach the developer role to them. Users with the developer role can access the core data assets in the workspace, so attach the developer role with caution.
- We recommend that you grant workspace administrator permissions only to core management personnel.

### Take note of the validity period of permissions

Quick BI allows you to apply for permissions on reports that you were not authorized to access. You must specify the validity period of the permissions. Approvers must pay attention to the validity period of permissions. Make sure that user permissions can be revoked regularly and the permissions are under control.

# 1.30. Graph Analytics

# 1.30.1. Platform security

### 1.30.1.1. Security isolation

Graph Analytics supports multi-tenant isolation. Users are walled off from seeing the settings or querying the data created by other users. Tenants can only obtain their own metadata settings. Different metadata corresponds to different business data, so one tenant can only query the business data corresponding to its own metadata.

### 1.30.1.2. Authentication

### Identity authentication

Graph Analytics currently supports two authentication methods in Apsara Stack:

- Authentication in Graph Analytics: Log on with the user password created in the Graph Analytics user system. To prevent password leakage, the password is MD5 encrypted and encrypted when transmitted on the network.
- Authentication of external users: Graph Analytics is connected to the user systems of customers. These systems are responsible for the security of their own users.

#### Permission control

Graph Analytics controls the access permissions to functional modules, data rows, and columns based on the permissions of users.

### 1.30.1.3. Data security

Graph Analytics is deployed in distributed clusters. The control nodes and compute nodes are separated, which helps to avoid a single point of failure (SPOF). In addition, data is synchronized among the clusters through a distributed cache to prevent data loss during the failover.

### 1.30.1.4. Transmission encryption

Graph Analytics provides Web services by using the HTTPS protocol. HTTPS is a secure and reliable data transmission protocol. It can effectively prevent security issues that may occur when data is transmitted over networks.

# 1.30.1.5. System security

96

# 1.30.1.5.1. Vulnerability scanning mechanism

Before launched, Graph Analytics has been scanned by Apsara Stack for security vulnerabilities in the following aspects:

- System security scanning: targets the security vulnerabilities in the operating system of Graph Analytics.
- Middleware dependency scanning: targets the security vulnerabilities in the middleware used by Graph Analytics.
- Code vulnerability scanning: targets the security vulnerabilities in Graph Analytics's own code and the third-party open-source framework.

# 1.30.1.5.2. Update scheme for security vulnerabilities

For security vulnerabilities detected by Alibaba Cloud security groups, Apsara Stack security tests, or other methods, the research and development team of Graph Analytics will perform immediate version updates or iterative updates based on the severity of the vulnerability. This ensures that the new version conforms to the production security regulations of Alibaba Cloud.

# 1.30.1.5.3. System defense mechanism

Graph Analytics is launched in the Apsara Stack environment. The system defense mechanism of Graph Analytics depends on the defense mechanism of Apsara Stack systems.

### 1.30.1.6. Infrastructure security

Graph Analytics is launched in the Apsara Stack environment and its infrastructure is secured by Apsara Stack security facilities. For more information, see the infrastructure security topic in Alibaba Cloud Security White Paper.

### 1.30.1.7. Level-based data security

Graph Analytics meets the Apsara Stack V3 requirements for security level 4. Graph Analytics has obtained the following security certificates from the Apsara Infrastructure Management Framework: cacert.pem, privatecloud.pem, privatecloud key.pem, and privkey.pem.

For Apsara Stack versions later than V3.3, you can connect to Graph Analytics by using the HTTPS protocol.

# 1.30.2. Tenant security

# 1.30.2.1. Log audit

All user requests are logged, so they can be used for audit or specific behavior analysis. The logs include information such as the user name, IP addresses, operation content, and operation statuses.

# 1.31. Elasticsearch (on k8s)

# 1.31.1. Security isolation

Each Elasticsearch cluster uses independent storage. You can configure a password for a cluster and use this password to access the cluster. Clusters are independent of each other. The Elasticsearch process runs in a container and does not support the root account. Elasticsearch uses Java virtual machines (JVMs) to isolate resources. You can access only the directories of data and configuration files even if third-party plug-ins are used.

## 1.31.2. Authentication and authorization

## 1.31.2.1. Identity authentication

Elasticsearch provides the security authentication feature to authenticate user identity. You can access Elasticsearch clusters by using the HTTP basic access authentication method. If the account or password that is configured on a client is invalid, you cannot use the client to access your cluster. Elasticsearch does not store passwords in plaintext. It uses the bcrypt algorithm to check whether passwords are valid.

### 1.31.2.2. Access control

After you configure security settings for an Elasticsearch cluster, you can use its built-in role-based access control (RBAC) system to perform access control for the cluster at the field level. In addition, you can use the Kibana console to manage users, roles, and permissions.

# 1.31.3. Data security

The underlying layer of Elasticsearch uses shards to store data and dynamically modifies the number of replicas for the shards. Data is replicated from shards to replicas to ensure data security. If a data node in a cluster fails or a hard disk on a data node in the cluster fails, the cluster can detect damage to shards. The health states of related indexes and the entire cluster change from green to yellow. In addition, the replication mechanism is automatically triggered to clone a shard based on existing data. The cluster then returns to the green health state after all data is restored.

# 1.31.4. Transmission encryption

Nodes in an Elasticsearch cluster communicate with each other over SSL. This way, attackers cannot intercept network traffic or launch attacks to the system.

# 1.32. Elasticsearch (on ECS)

# 1.32.1. Security isolation

Each Elasticsearch cluster uses independent storage. You can configure a password for a cluster and use this password to access the cluster. Clusters are independent of each other. The Elasticsearch process runs in a container and does not support the root account. Elasticsearch uses Java virtual machines (JVMs) to isolate resources. You can access only the directories of data and configuration files even if third-party plug-ins are used.

# 1.32.2. Authentication and authorization

### 1.32.2.1. Identity authentication

Elasticsearch provides the security authentication feature to authenticate user identity. You can access Elasticsearch clusters by using the HTTP basic access authentication method. If the account or password that is configured on a client is invalid, you cannot use the client to access your cluster. Elasticsearch does not store passwords in plaintext. It uses the bcrypt algorithm to check whether passwords are valid.

### 1.32.2.2. Access control

After you configure security settings for an Elasticsearch cluster, you can use its built-in role-based access control (RBAC) system to perform access control for the cluster at the field level. In addition, you can use the Kibana console to manage users, roles, and permissions.

# 1.32.3. Data security

The underlying layer of Elasticsearch uses shards to store data and dynamically modifies the number of replicas for the shards. Data is replicated from shards to replicas to ensure data security. If a data node in a cluster fails or a hard disk on a data node in the cluster fails, the cluster can detect damage to shards. The health states of related indexes and the entire cluster change from green to yellow. In addition, the replication mechanism is automatically triggered to clone a shard based on existing data. The cluster then returns to the green health state after all data is restored.

# 1.32.4. Transmission encryption

Nodes in an Elasticsearch cluster communicate with each other over SSL. This way, attackers cannot intercept network traffic or launch attacks to the system.