Cloud Access Management

Product Documentation

Contents

Cloud Access Management 5
- Ops Management Guide 5
  - Product Architecture 5
    - Core Feature Components 5
      - Basic Platform 5
      - Users and Permissions 6
      - CAM 7
      - Cloud API Management 8
      - License Management 9
    - Product Architecture Diagram 10
      - Basic Platform 10
      - CAM 12
      - Cloud API Management 14
      - License Management 15
    - Deployment Architecture Diagram 16
      - CAM 16
    - Business Flow Diagram 17
      - Basic Platform 17
      - Users and Permissions 19
      - CAM 21
      - Cloud API Management 22
      - License Management 23
    - Core Logic Overview 24
      - Basic Platform 24
      - Users and Permissions 26
      - CAM 29
      - Cloud API Management 30
      - License Management 31
  - Product Dependencies 34
    - Users and Permissions 34
    - Audit Logs 35
    - CAM 36
    - Tag Management 38
    - Cloud API Management 39
    - License Management 41
  - Troubleshooting 42
    - Impact Scope of Failures 42
      - Tag Management 42
      - License Management 43
    - Failure Recovery 44
      - Scenario 1: Error When Opening the Users and Permissions Page or Creating an Account on the Operation Side 44
      - Scenario 2: Tag Service Page Error 46
      - Scenario 3: Tags Not Displayed After Being Added to a Resource on the Product Page 48
      - Scenario 4: Cloud API Product Unavailable, Affecting Platform Management Workload 54
  - Daily Inspection 56
    - Cloud Product Inspection 56
      - Viewing Inspection Items 56
      - Viewing Inspection Tasks 67
      - Viewing Health Check Results 73
    - TCenter Inspection Item Checklist 76
    - Inspection Handling 115
      - Inspection Item 1: Basic Platform Feature Inspection 115
      - Inspection Item 2: Component Status Inspection 118
  - Daily Monitoring 121
    - Common Monitoring Metrics 121
    - Alarm Handling 123
      - Contingency Plan for Pod Restart 123
      - Contingency Plan for Pod CPU Utilization (Relative to the Limit) Exceeding the Threshold 126
      - Contingency Plan for Pod Actual Memory Utilization (Relative to the Limit) Exceeding the Threshold 129
  - Emergency Plan 132
    - User and Permission: No Permission to Access Product Pages After Product Permissions Are Bound 132
  - Appendix A: Common Operations 135
    - Common Operations 135
    - Obtaining Service Logs 137
    - Viewing Pod Rendering Status 138
    - Obtaining Operation Logs 140
    - Technical Metrics 141
      - User and Permission 141
    - Performance Metrics 142
      - User and Permission 142
- Product White Paper 143
  - Product Overview 143
  - Product Strengths 144
  - Use Cases 145
  - Product Architecture 146
  - Features 147
  - Usage Recommendations 149
- Product Introduction 157
  - Overview of CAM 157
  - Product Features 158
  - Use Cases 159
  - Use Limits 160
  - Products Supported by CAM 161
- Operation Guide 165
  - Overview 165
  - User Management 166
    - User Type 166
    - Root Account 167
    - Sub-account 168

- Creating Sub-users 168
- Sub-user Permission Settings 170
- Sub-user API Key Management 172
- Sub-user Security Credentials 174
- Sub-user Login 174
- Resetting Log-in Password for Sub-Users 175
- Security Protection for Sub-Users 176
- Subscribing Messages for Sub-users 180
- Deleting Sub-users 181
- User Information 182
- User Group 183
- Creating User Groups 183
- User Management 184
- User Group Permission Settings 186
- Deleting a User Group 187
- User Settings 188
- Password Rules 188
- Login Policies 189
- Policy Management 190
- Relevant Definitions 190
- Permission 190
- Policy 191
- Authorization Guide 192
- Creating Custom Policies 192
- Authorization Management 194
- IP-based Access Restriction 196
- Syntax Logic 198
- Element Reference 198
- Syntax Structure 200
- Evaluation Logic 205
- Resource Description Methods 207
- Policy Variable 210
- Conditions 212
- Role Management 218
- Role Overview 218
- Basic Concepts 219
- Creating Roles 221
- Modifying Roles 222
- Deleting Role 223
- Authorizing Roles 224
- Granting Role Assumption Policy to Sub-Accounts 225
- Best Practices 226
- Access Keys 227
- Viewing the Current User's Access Keys 227
- Troubleshooting 228
- Creating Policy Based on Fault Report 228
- Enterprise Authentication Login Management 232
- WeCom Account 238
- Best Practice 240
- CMQ Case Studies 240
- Authorize the sub-account to have full permissions for the message service 240
- Authorize Sub-account Full Permissions to Access the TDMQ it creates 241
- Authorize sub-account read permission for a specific TDMQ 242
- CVM Relevant Cases 243
- Authorize Sub-account Full Permissions to CVM 243
- Authorizing Sub-account Read-only Permission to CVM 244
- Authorize sub-accounts to have read-only permission of CVM related resources 245
- Authorize Sub-account Permission to Perform Operations on CBSs 247
- Authorize Sub-account Permission to Perform Operations on Security Groups 248
- Authorizing a Sub Account to Perform Operations on EIP Addresses 250
- Authorize sub-accounts to have the operation permission CVM 252
- Authorize sub-accounts to have the operation permission of CVM i 253
- Authorize the sub-account to have all the permissions of CVM, but not including payment permissions 254
- Grant Sub-account View Permissions for Unsupported Project Products 255
- API Documentation 256
- Cloud Access Management(cam) 256
- version(2019-01-16) 256
- API overview 256
- CallingMethod 258
- signaturev1 258
- signaturev3 266
- request 275
- result 276
- PublicParameter 279
- Identity Provider APIs. 281
- CreateOauthProvider 281
- GetUserAccessToken 283
- RefreshUserToken 285
- UpdateOauthProvider 287
- VerifyUserAccessToken 290
- Other APIs 292
- AttachRolePolicies 292
- AttachRolePolicy 294
- AttachRolesPolicy 296
- CreatePolicy 298
- CreateRole 299
- DeletePolicy 300
- DeleteRole 302
- DescribeRoleList 304
- DetachGroupPolicies 306
- DetachGroupsPolicy 308

- DetachUsersPolicy 310
- GetPolicy 312
- GetRole 314
- GetServiceRoleInfo 316
- ListAttachedGroupPolicies 319
- ListAttachedRolePolicies 321
- ListEntitiesForPolicy 322
- ListPolicies 323
- UpdateAssumeRolePolicy 325
- UpdatePolicy 326

• User APIs 327

- GetPasswordRules 327
- GetSubsGroup 329
- GetUinBySecretId 331
- UpdatePasswordRules 333

• Data structure 335

• ErrorCode 352

Ops Management Guide
Product Architecture
Core Feature Components
Basic Platform

Component ID	Component Feature
tcloud-tcenter-project-org	Project management/Resource management
ocloud-tcenter-location	Platform deployment information service

Users and Permissions

Component ID	Component Feature
ocloud-tcenter-identityaccess	Core component, responsible for account read/write, login, verification, and MFA authentication
ocloud-tcenter-identityccdb	Responsible for account synchronization
ocloud-tcenter-open-identity	Responsible for enterprise authentication login
ocloud-tcenter-mc-cas-portal	Responsible for operation side CAS login

CAM

Component ID	Component Feature
tcloud-tcenter-cam	Tenant-side CAM
ocloud-tcenter-cam	Operation-side CAM
product-tcenter-support-cam	Virtual machine (VM)-based standalone CAM (optional)
tcloud-tcenter-support-cam	Containerized standalone CAM (optional)

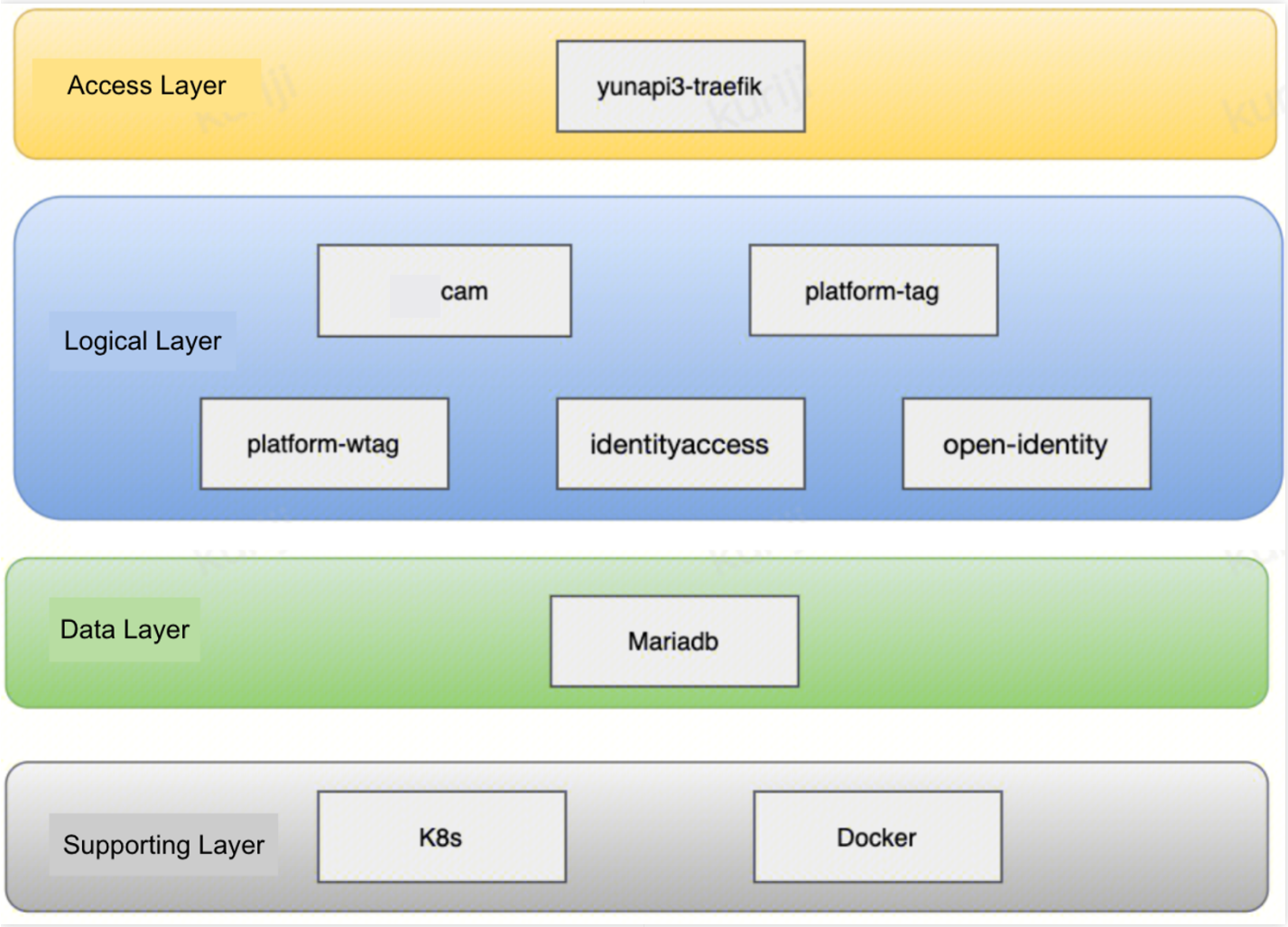
Cloud API Management

Component ID	Component Feature
ocloud-tcenter-yunapi3-traefik	Cloud API gateway service for the operation side
tcloud-tcenter-yunapi3-traefik	Cloud API gateway service for the tenant side
ocloud-tcenter-capi-internal	Gateway service for internal service calls
ocloud-tcenter-yunapi3-yuntu	Responsible for cloud API management on the operation side
tcloud-tcenter-yunapi3-tyuntu	Responsible for cloud API management on the tenant side
image-preset-yunapi	Cloud API preset data import/export tool (apitools)
ocloud-tcenter-location	Responsible for managing deployment information of platform products

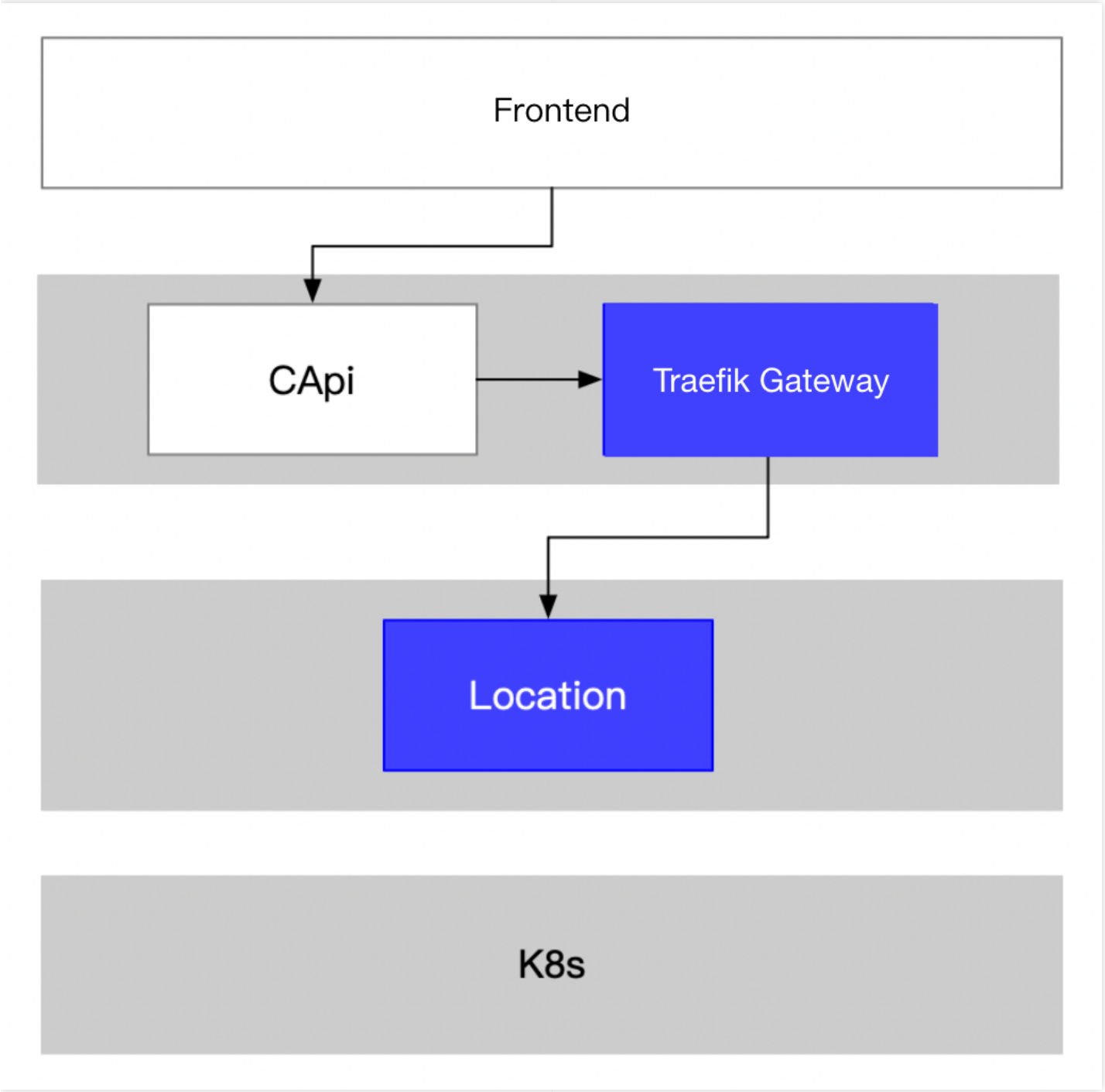
License Management

Component ID	Component Feature
ocloud-tcenter-license	License backend service component
dbsql-tcenter-license	License service database component
ocloud-tcenter-license-crd	LicenseClient CRD registration component

Product Architecture Diagram
Basic Platform

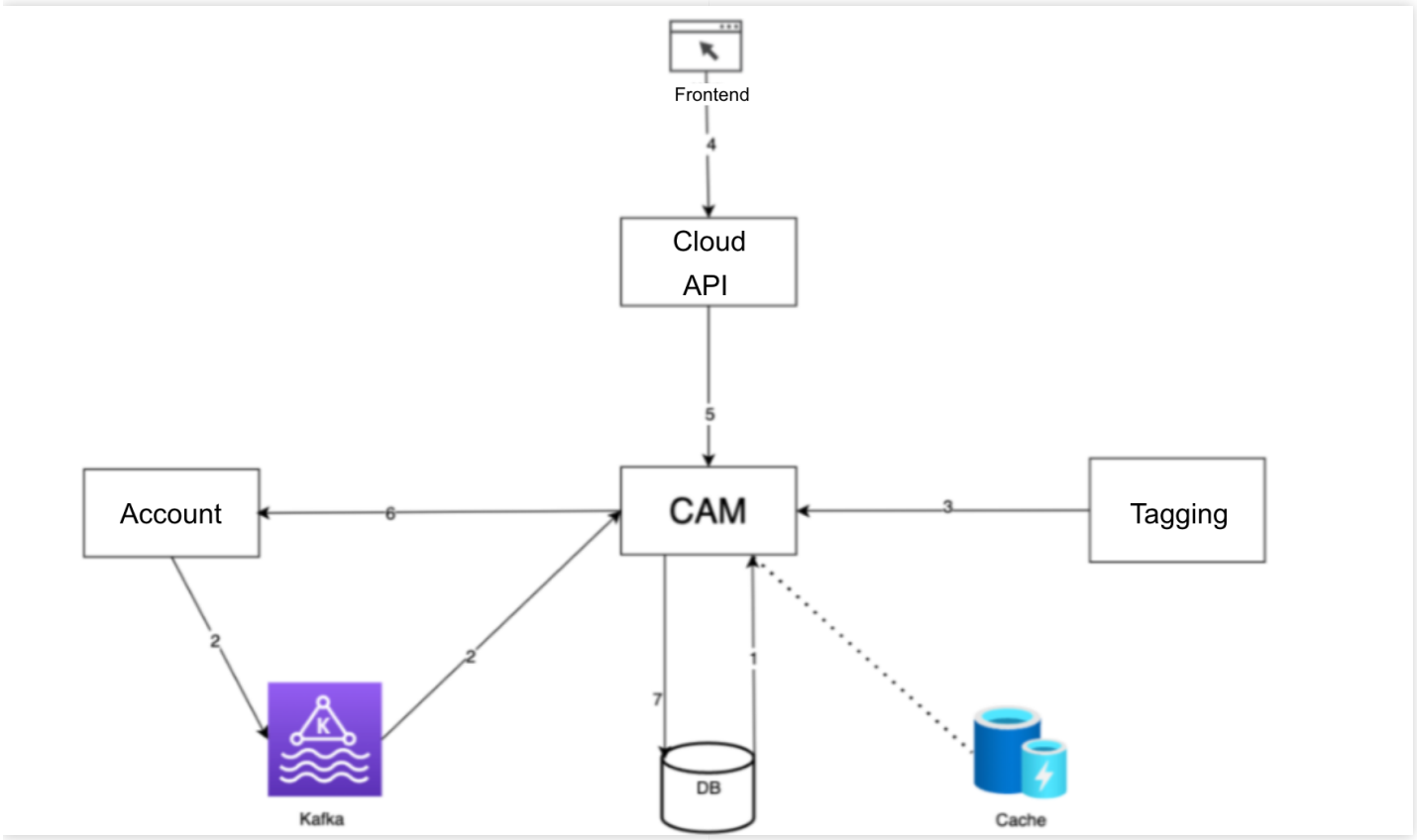


Location Architecture

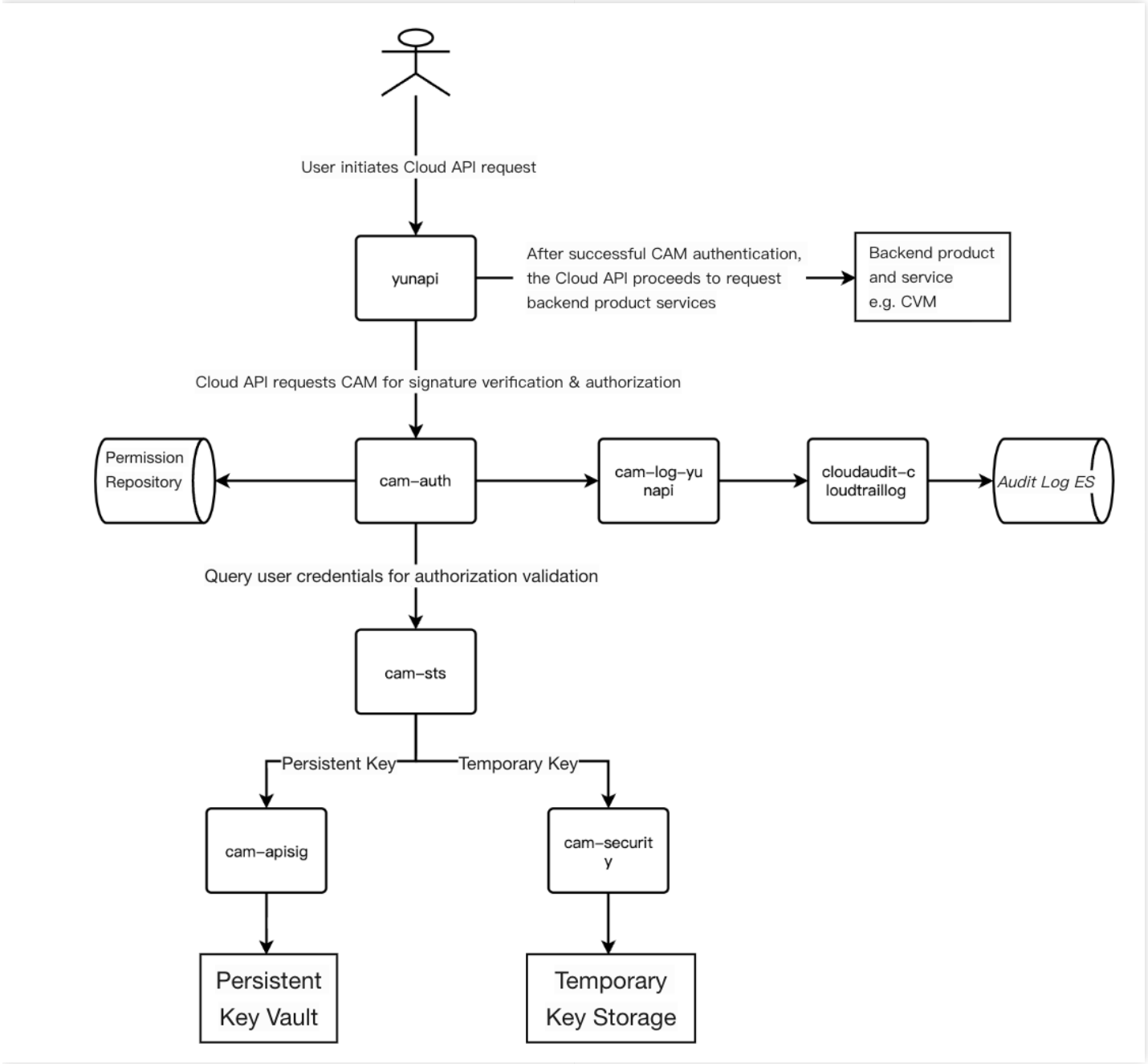

Frontend
CApi
Traefik Gateway
Location
K8s

CAM

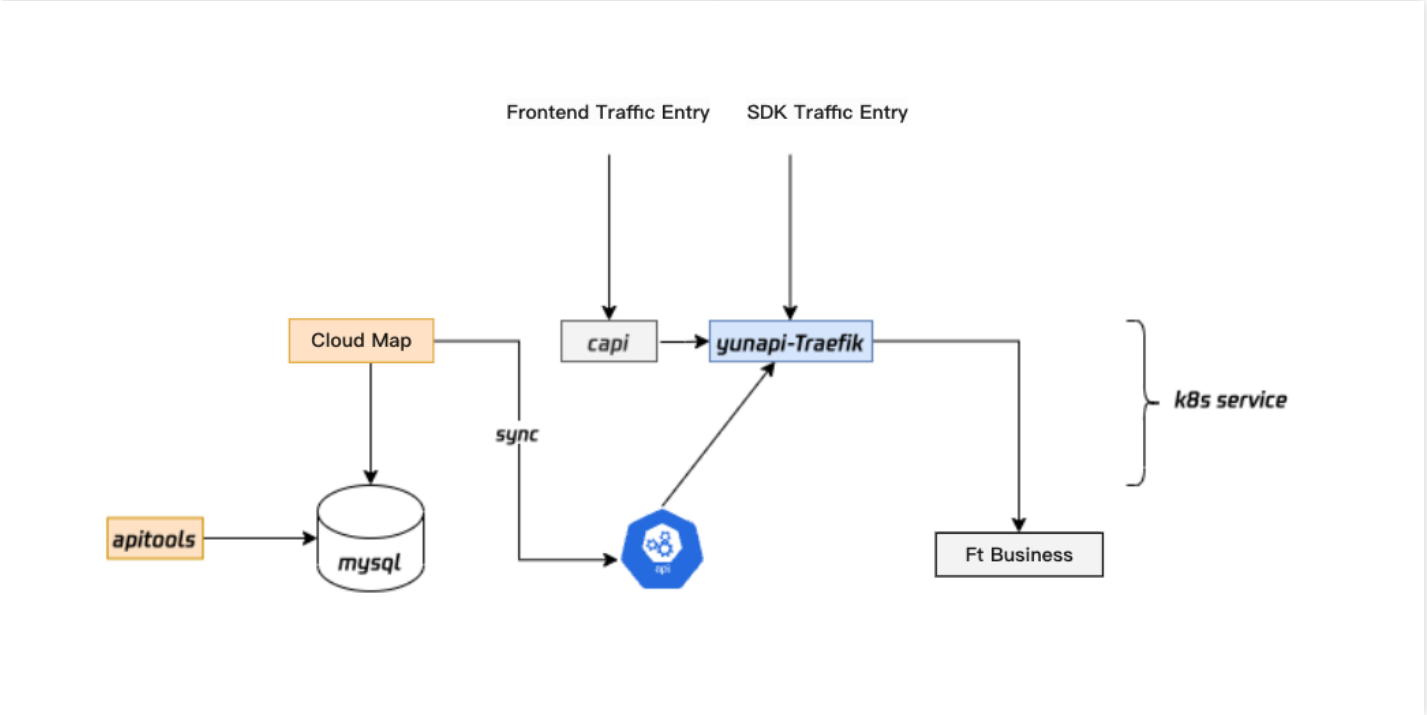
CAM mainly depends on services such as account, tag, and TLSQL.



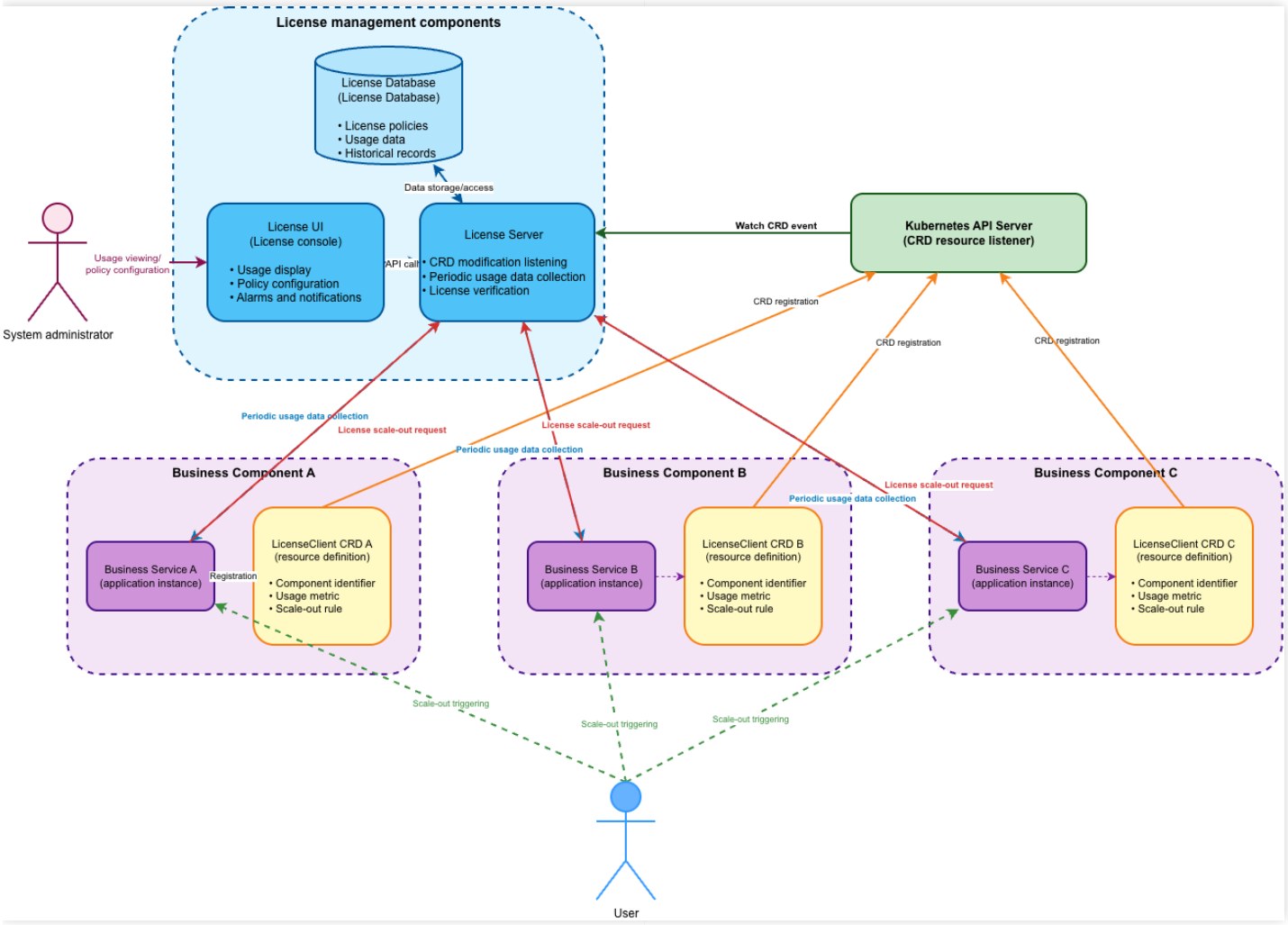
User authentication requests are forwarded by the cloud API to CAM. CAM queries the user key and performs the authentication verification.


User initiates Cloud API request
yunapi
After successful CAM authentication, the Cloud API proceeds to request backend product services
Backend product and service e.g. CVM
Cloud API requests CAM for signature verification & authorization
Permission Repository
cam-auth
cam-log-yunapi
cloudaudit-cloudtraillog
Audit Log ES
Query user credentials for authorization validation
cam-sts
Persistent Key
Temporary Key
cam-apisig
cam-security
Persistent Key Vault
Temporary Key Storage

Cloud API Management

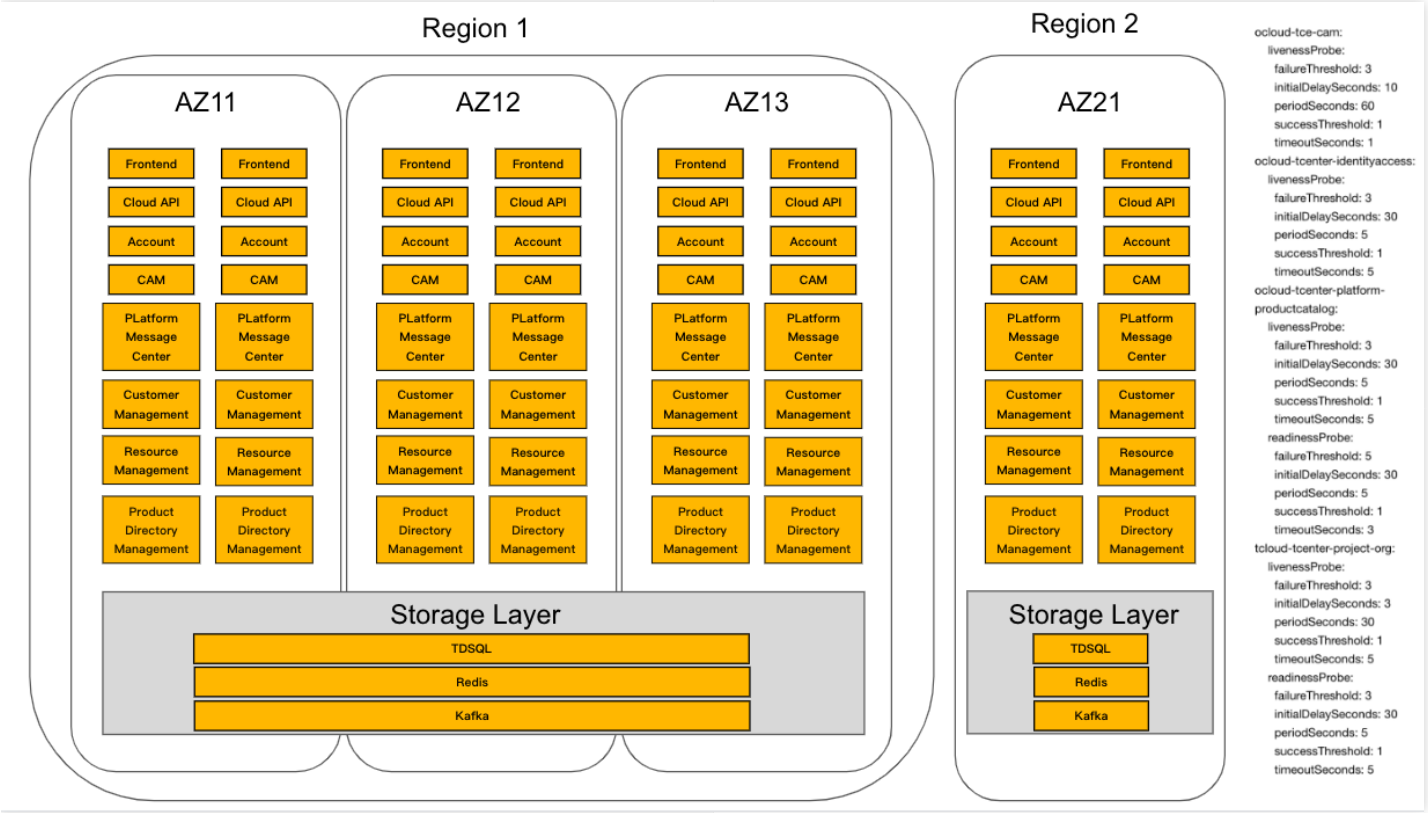


License Management



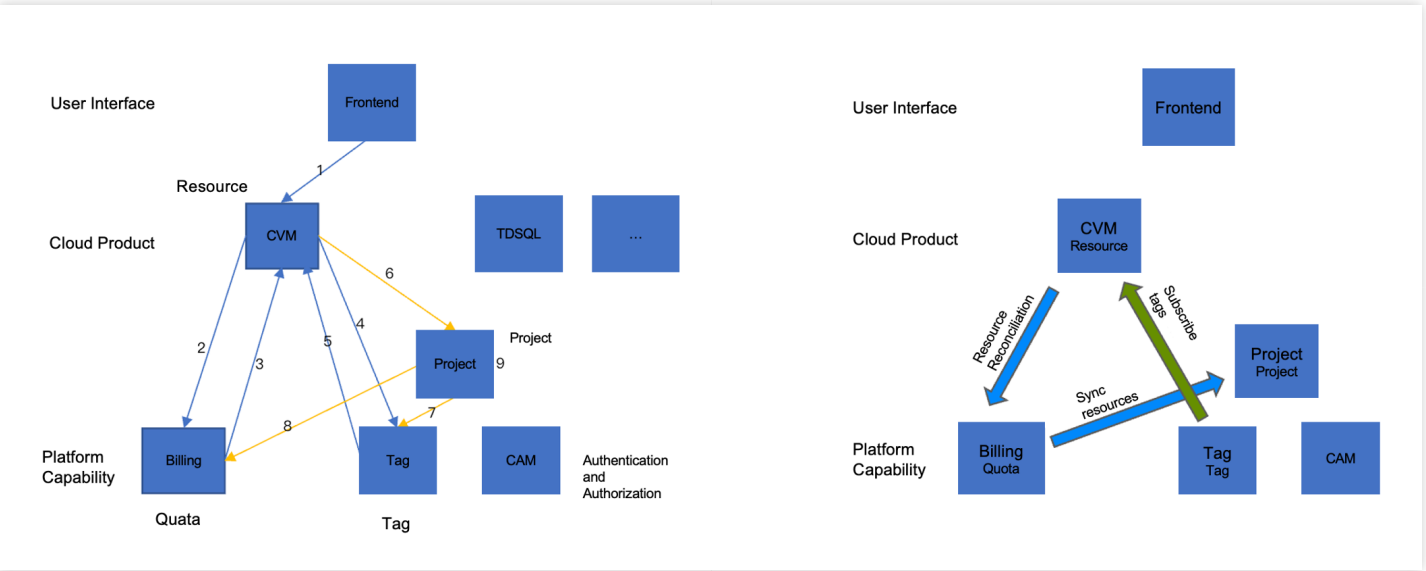
Deployment Architecture Diagram
CAM

The CAM service is a global-level stateless service. The standalone editions of CAM (with component names product-tcenter-support-cam and tcloud-tcenter-support-cam) are region-level services. The CAM service relies on supporting databases for storage and can be evenly deployed across all regions and availability zones. All components support horizontal scaling.

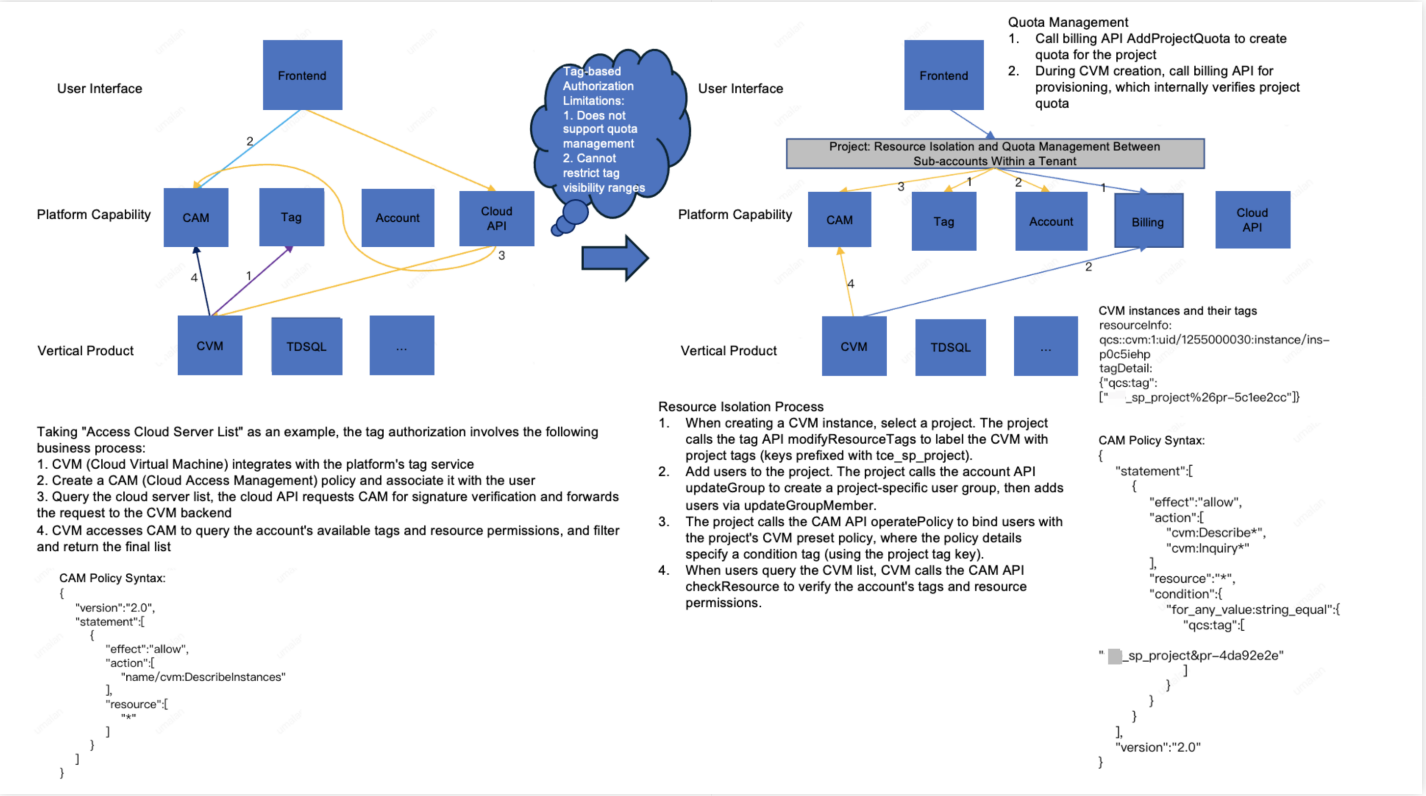


Business Flow Diagram
Basic Platform

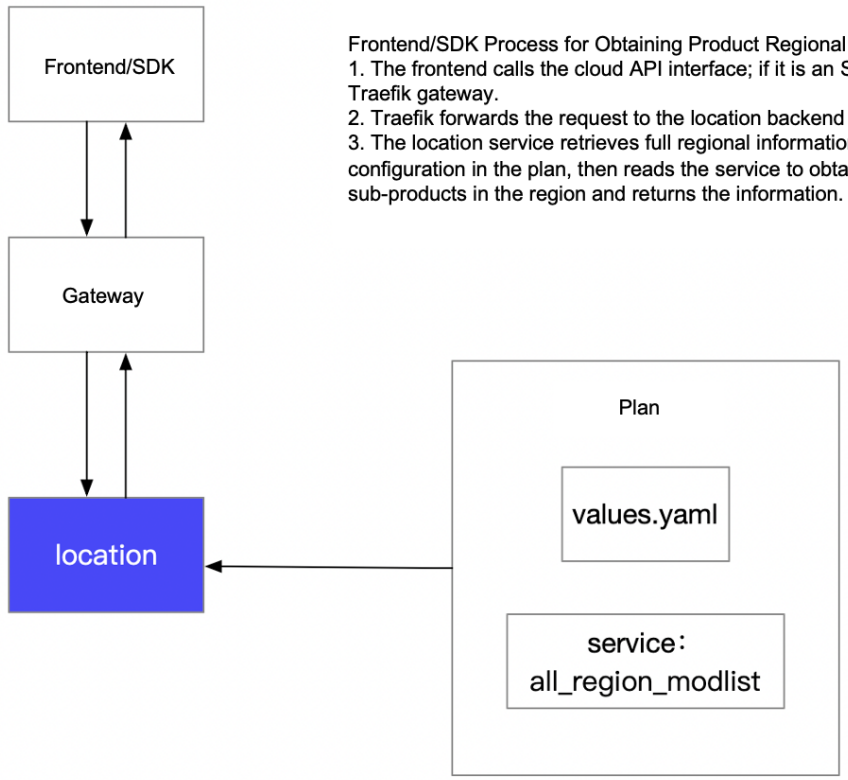
The project mainly interacts with billing, CAM, and various access resources. The basic workload flow is shown as follows:



Using CVM as an example, the following process explains how the project achieves resource isolation:



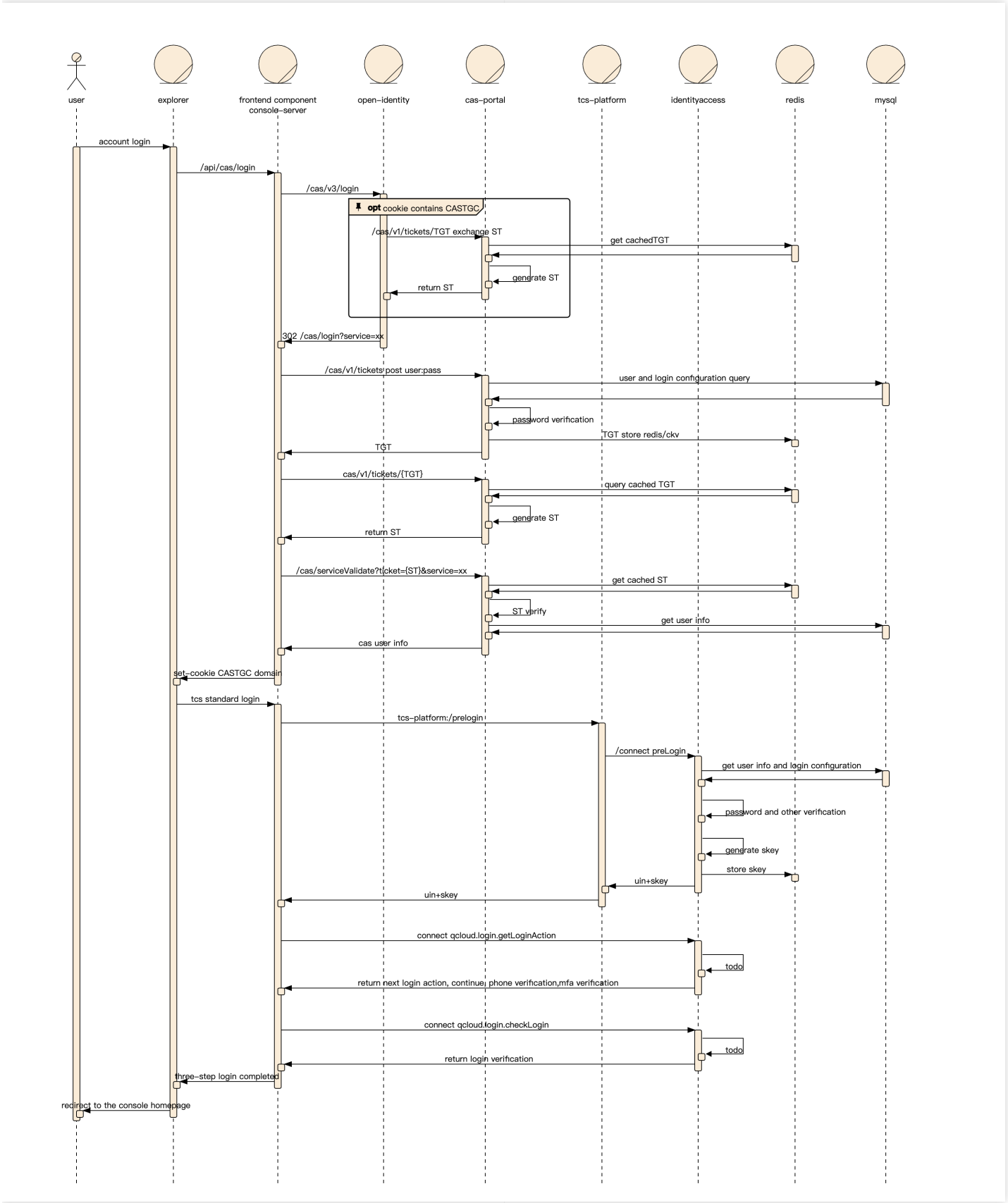
Users query the location service to obtain product deployment information by region:



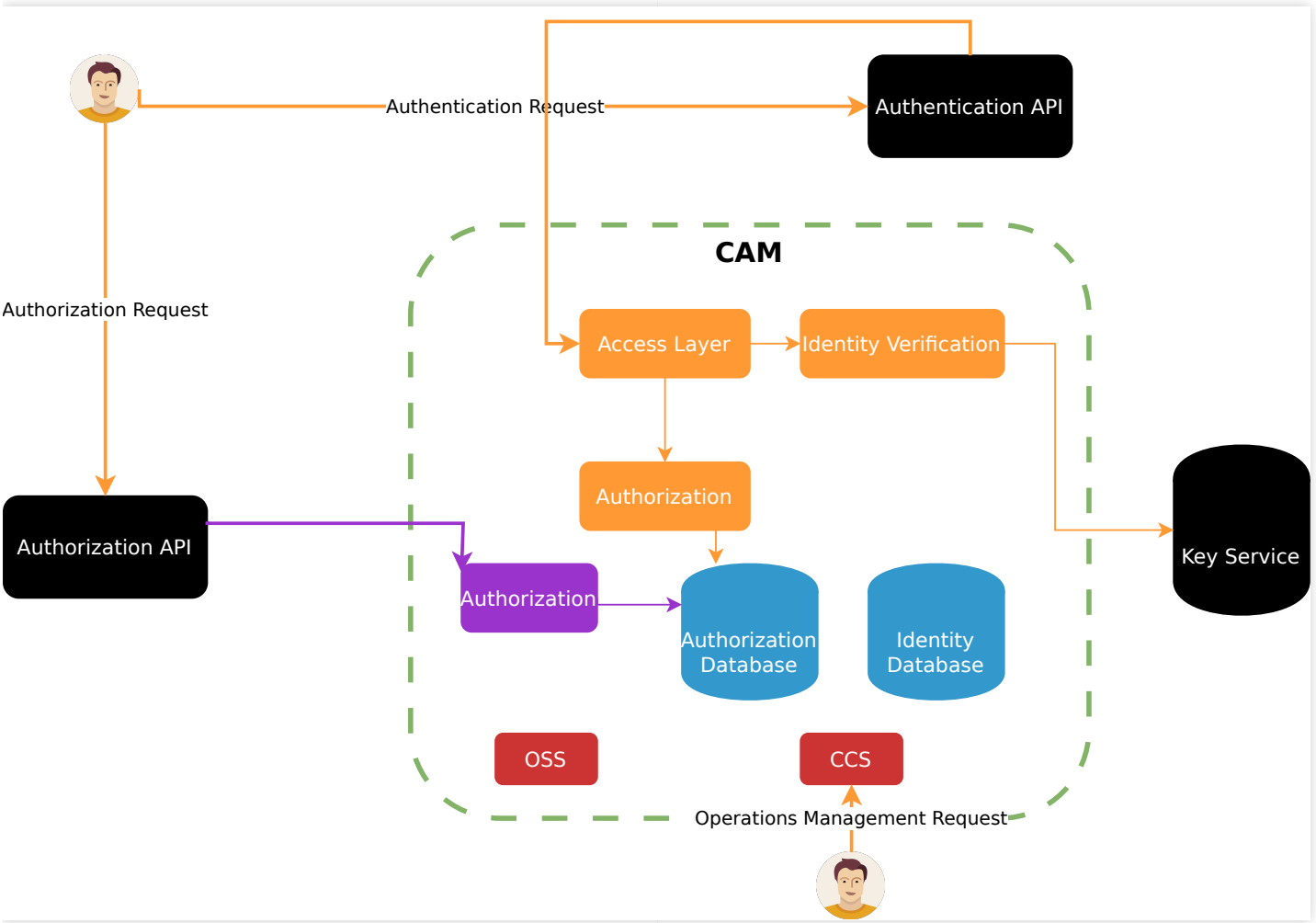
Frontend/SDK Process for Obtaining Product Regional Deployment Information:
1. The frontend calls the cloud API interface; if it is an SDK call, it directly accesses the Traefik gateway.
2. Traefik forwards the request to the location backend service.
3. The location service retrieves full regional information based on the region configuration in the plan, then reads the service to obtain the deployed products and sub-products in the region and returns the information.

Users and Permissions

The core login flow on the operation side is as follows:


user
explorer
frontend component
console-server
open-identity
cas-portal
tcs-platform
identityaccess
redis
mysql
account login
/api/cas/login
/cas/v3/login
opt cookie contains CASTGC
/cas/v1/tickets/TGT exchange ST
get cachedTGT
generate ST
return ST
302 /cas/login?service=xx
/cas/v1/tickets post user:pass
user and login configuration query
password verification
TGT store redis/ckv
TGT
cas/v1/tickets/{TGT}
query cached TGT
generate ST
return ST
/cas/serviceValidate?ticket={ST}&service=xx
get cached ST
ST verify
get user info
cas user info
set-cookie CASTGC domain
tcs standard login
tcs-platform:/prelogin
/connect preLogin
get user info and login configuration
password and other verification
generate skey
store skey
uin+skey
uin+skey
connect qcloud.login.getLoginAction
todo
return next login action, continue, phone verification,mfa verification
connect qcloud.login.checkLogin
todo
return login verification
three-step login completed
redirect to the console homepage

CAM

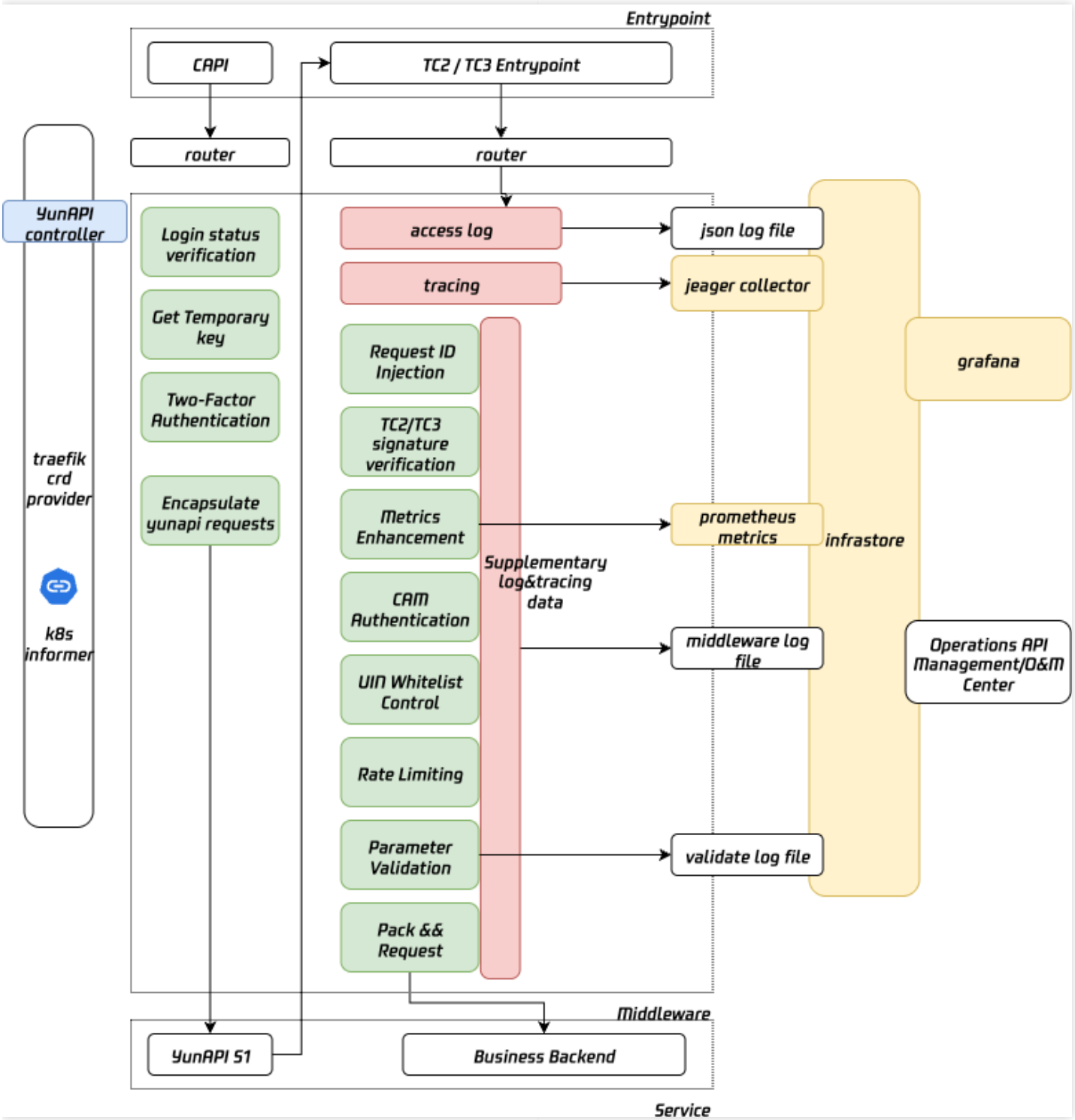


Authentication consists of two parts:

- Identity verification: The use of keys ensures that requests are not tampered with during transmission. The request URL contains the key ID and signature key. On the server side, the system obtains the user's key based on the key ID, recalculates the signature, and compares it with the incoming one. If the signatures match, the request is considered valid.
- Permission control: Permissions can be defined at the resource level or defined more precisely at the API level.

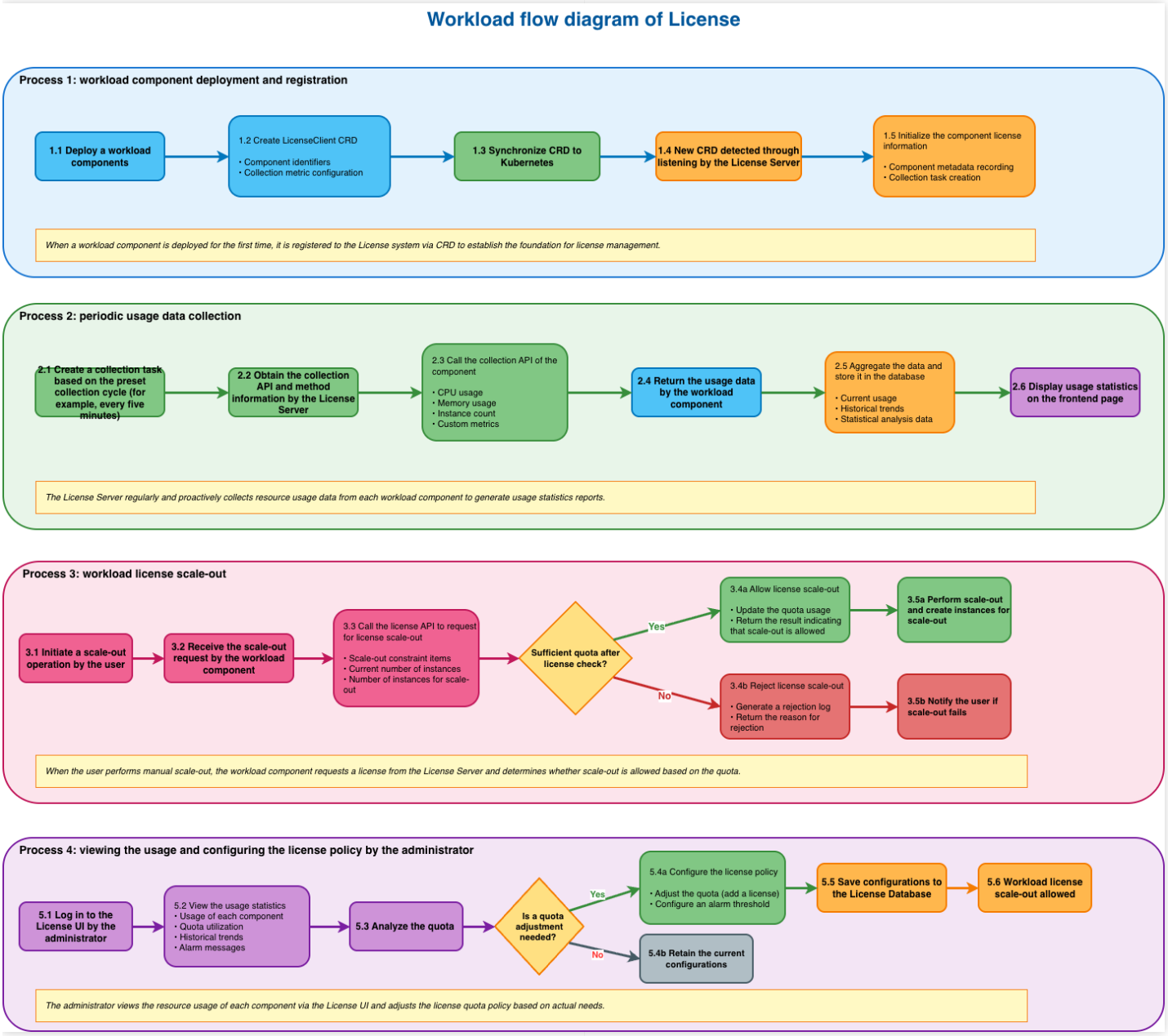
Cloud API Management

The core access path: frontend > CAPI > Traefik > FT backend services.



License Management

Workload Flow Diagram



Core Logic Overview
Basic Platform

Core Logic for Users to Query a Resource List

1. View the list of resources the user has access to via the API logic.cam.checkResource, which calls CAM to query all user groups the user has joined, named tce_project_${ProjectName}. This returns a list of associated projects and a tagList, where the key is tce_sp_project and the value is the project name.
2. If the user has the corresponding permissions, query the API to fetch resources by tags based on the tagList, and load the resources with the relevant tags using qcloud.tag.getResourcesByTags.
3. Based on the resource data obtained from the above steps, filter the resources (using deny or allow logic) against all resources under the root account to determine which resources should be displayed. For resources with the tag key tce_sp_project, when you verify the policy, the tag should correspond to the project name.

Adding Users and Assigning Permissions

1. Add users to the project-specific user group tce_project_${ProjectName} , then assign the corresponding policy permissions to the users.
2. On the Add Member page, only users not already part of the project can be added. For users already in the project, policies can only be edited via the table.

> Note:
>
> The policy list is maintained via the project's metadata, displaying the available policies that can be assigned. When you authorize users, CAM is queried. If the policy for the project does not exist, it is created and then assigned to the user. CAM policy naming convention: tce_sp_project_${ProjectName}*${PolicyName}. CAM applies special handling to policies with the tce_project prefix. When policy names are displayed, the prefix is removed; that is, all policies starting with tce_sp_project* should be listed.

Logic for Users to Query the Location Service to Obtain Product Deployment Information by Region

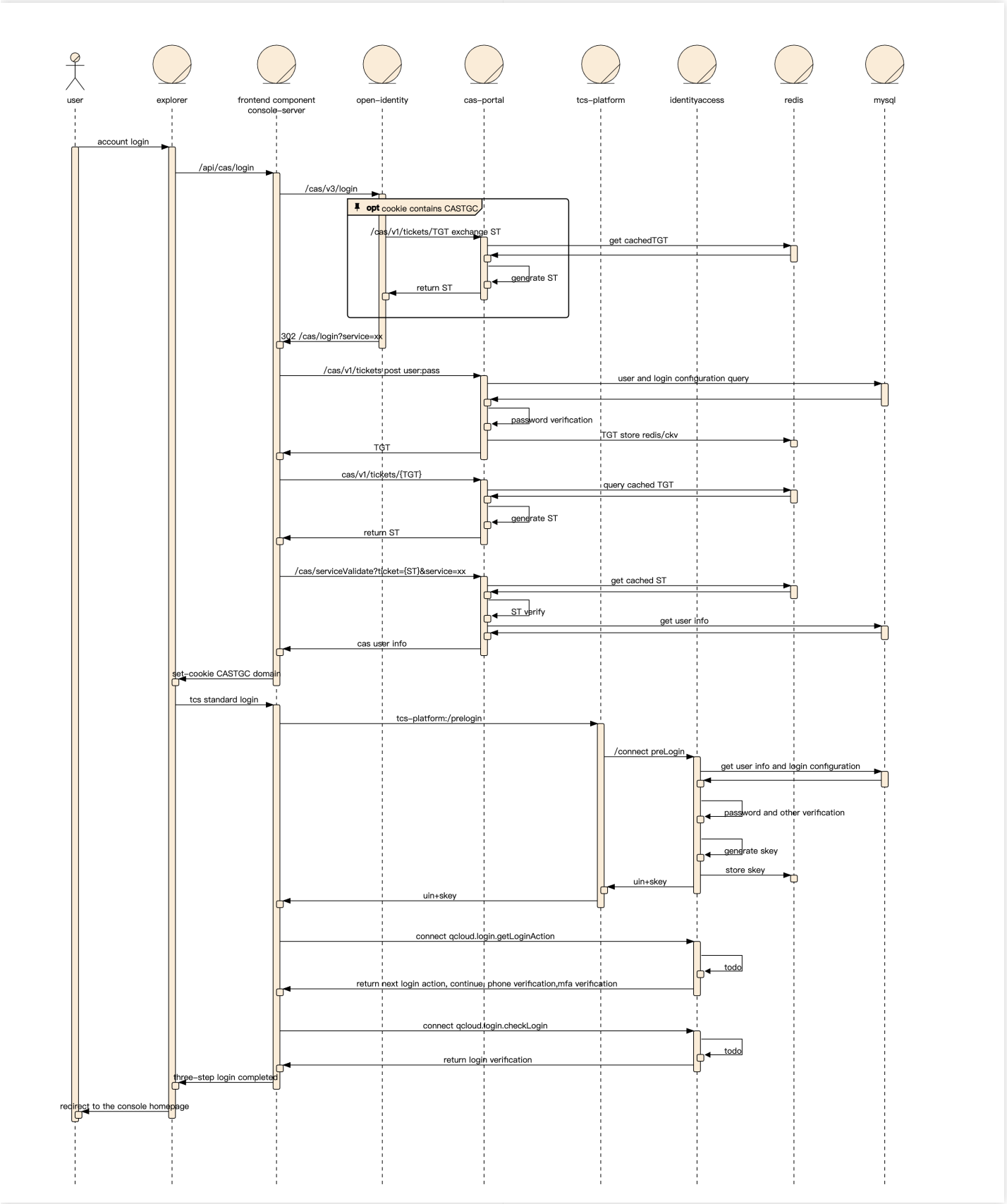
1. The user queries the full product list in all regions.

2. If the user specifies certain regions, those specified regions are filtered out from the full region configuration.
3. Select and apply appropriate policies according to different conditions:
    - If input.ProductID is a special product ID, a "full" list is supported.
    - If input.ProductID is empty, the full product list is returned. Process the region allowlist as follows:
        - If it is not a CMGT request, the WhitelistStrategy is applied.
    - If it is an operation side request, the DeploymentStrategy is applied.
    - By default, both DeploymentStrategy and WhitelistStrategy are applied.
      DeploymentStrategy: Filters out regions and availability zones (AZs) where the product or subproduct is not deployed, based on the deployment information of products and subproducts.
      WhitelistStrategy: Filters and updates the status of regions and AZs based on the allowlist configuration:

- No allowlist configured: Not restricted by allowlist; all access is allowed.
- Configured as full access: All UINs are allowed to access.
- Configured as in effect: Only UINs in the allowlist are allowed access.
- Configured as invalid: No UIN is allowed access.
  ==> Equivalent conversion:
- Configured as in effect: UINs not in the allowlist are denied access.
- Configured as invalid: All UINs are denied access.

Users and Permissions

Core login logic:



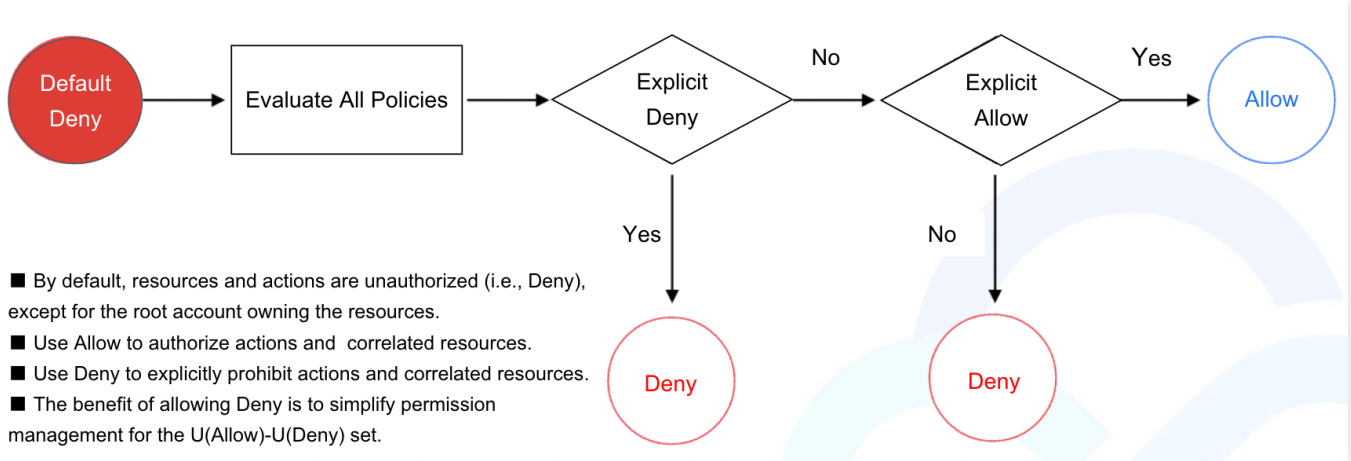
Upon user login, the system first checks whether the cookie contains a TGT. If the TGT exists, an ST is generated using the TGT. The ST is then used to retrieve user information and call the login API to log in to the console directly.

If the cookie does not contain a TGT, the user is redirected to the login page and follows the standard cloud platform login process.
During the cloud platform login process, the system checks whether the account is locked, whether the username and password are correct, whether the password has expired, and whether login protection is enabled.
The login flow also involves calling TDSQL, writing the skey to Redis® after login, and reporting login information to Kafka.

CAM

Vertical products integrate with the cloud API according to specifications, and cloud APIs can be configured as either signature authentication or signature-only without authentication, depending on the requirement:

1. For cloud APIs that have signature authentication enabled, the cloud API calls CAM to perform signature authentication.
2. For cloud APIs set to signature-only without authentication, the cloud API passes through the request directly to the vertical product, and the vertical product may choose to call CAM itself for signature authentication.
   The cloud API or vertical product extracts information such as account, API, resource, and conditions from the request, then calls the CAM API to determine whether the current logged-in account has permission to access the specified resource via the cloud API.
   CAM supports general authentication, anonymous authentication, cross-account authentication, token-based authentication, tag-based authentication, and list-based authentication. The common authentication process is as follows:

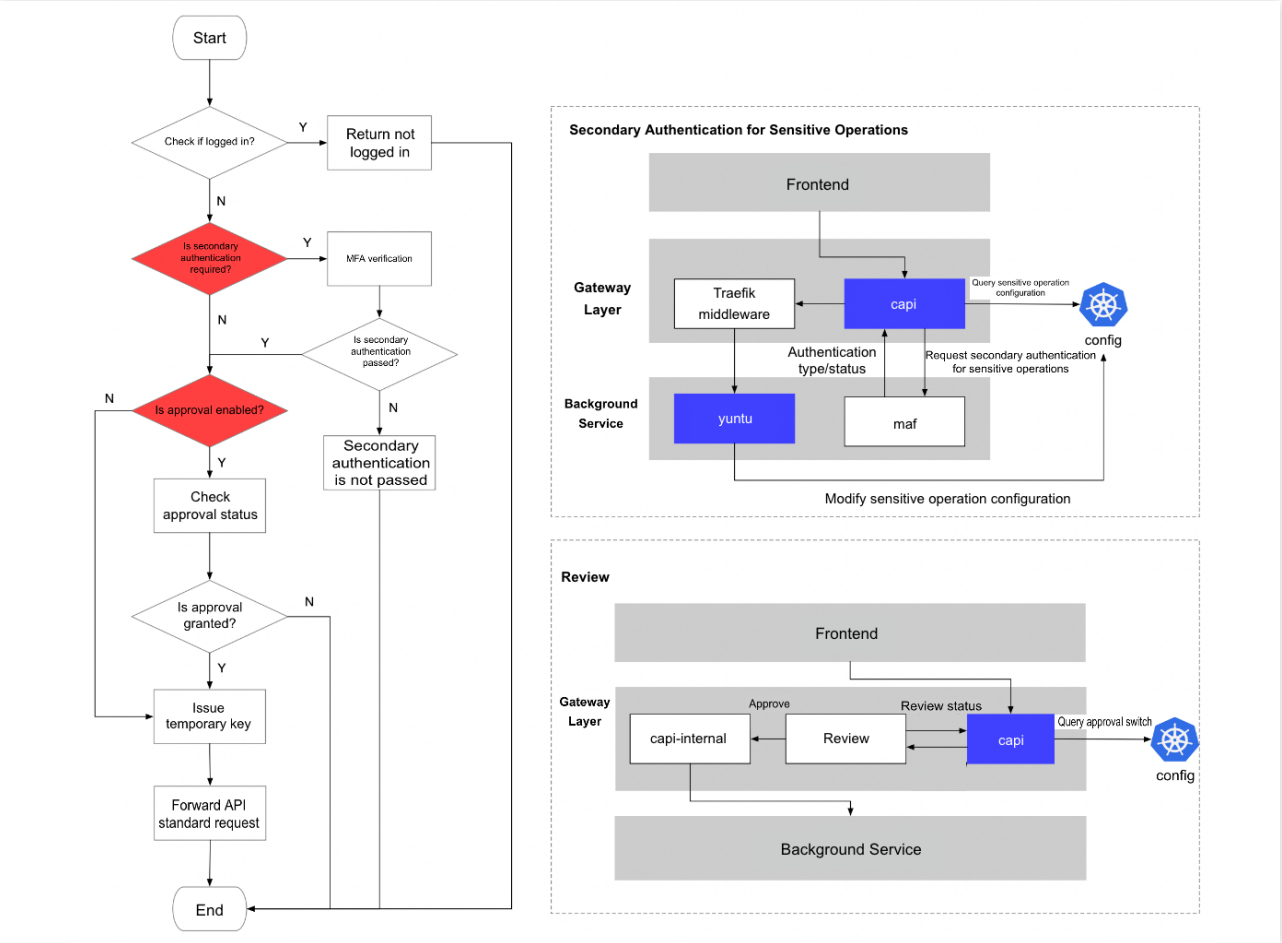


List-based authentication uses tags to implement resource isolation. Vertical products directly call the CAM checkResource API to query the resource instance and tag lists that the current account has permission to access.

Cloud API Management

The core process of capi in controlling cloud API requests is as follows:

- Login check process: Based on the UIN and SKey, the system requests the authentication service to verify the login status.
- Secondary authentication process for sensitive operations: The gateway checks whether the API is a sensitive operation and whether secondary authentication is enabled. If the secondary authentication is enabled, it returns a special error code to the frontend. The frontend pops up a window prompting the user to enter a code and resubmit the request. The gateway passes through the code to MFA for verification.
- Approval check process:
  - The gateway checks whether the approval switch is enabled. If the switch is enabled, it calls the approval service to create an approval record. Subsequent requests will call the approval service to check whether the approval status is passed.
  - If the approval passes, it will call capi-internal to simulate a frontend request to the gateway to make the request take effect.



License Management

Component Registration and License Initialization

Purpose: Automatically discover and register new workload components.
Process:

1. The License Controller listens to resource modification events of LicenseClient CRD on the Kubernetes API Server.
2. The License Server reads the component identifier from CRD and collects configurations and other information.
3. The License Server writes the license information and quota configurations to the License Database and then updates the CRD status to Registered.
4. The License Server sends a notification of the registration result via the event mechanism.

Periodic Usage Collection

Purpose: Periodically collect the resource usage information on components.
Process:

1. The scheduler of the License Server triggers a collection task every five minutes.
2. The License Server obtains the list of all registered LicenseClient resources from Kubernetes.
3. The License Server concurrently calls the collection APIs exposed by workload components to obtain the constraint item usage information (such as the CPU usage, memory usage, and number of instances).
4. The License Server aggregates collection results, calculates the quota utilization values, and writes them to the License Database by time.
5. The License UI reads data from the database and displays the data in statistical reports and trend charts.

License Check for Scale-Out

Purpose: Check whether the quota is sufficient before component scale-out.
Process:

1. A workload component initiates a scale-out request (containing the component ID, target number of instances, and other information).
2. The License Server verifies the validity of the request.

3. The License Server queries the total quota for the component and the current usage from the License Database.
4. The License Server performs a quota check. If the sum of the scale-out amount and the current usage does not exceed the total quota , the request is allowed. Otherwise, the request is rejected.
5. The License Server returns the result (allowed or rejected) and records audit logs.

Quota Policy and Alarms

Alarm rules:

- Warning (reaching 80%): The message is in yellow. It is recommended to perform scale-out promptly.
- Alarm (reaching 100%): The message is in red. Scale-out is not allowed, and the issue needs to be solved immediately.

Notification channels: site message, mail, SMS message, and announcement.

Administrator operations:

1. Log in to the License UI to view the real-time usage, historical trends, and alarm status.
2. Modify the component quota, adjust the alarm threshold, and set a validity period.
3. The License Server listens to policy modification events and loads new configurations in real time.
4. All modification operation records are stored in the License Database for auditing and tracking.

Key Design Principles

Principle	Description
Driven by events	Automatically trigger the registration process based on CRD events.
Asynchronous concurrency	Concurrently call component APIs during collection to improve efficiency.
Quota priority	Check the quota before scale-out to prevent resource overuse.
Multi-level alarms	Provide warnings and alarms and send notifications to the administrator promptly.
Full auditing	Generate logs for all operations (registration, collection, check, and modification).
Real-time synchronization	Enable policy modifications to take effect immediately without the need to restart the system.

Core Data Flow

CRD modification event
  |
Listen by the server > Read configurations > Write configurations to the database > Update the CRD status
  |
Trigger scheduled collection
  |
Obtain the component list by the server > Call collection APIs concurrently > Aggregate the usage data > Write data to the database
  |
Check based on alarm rules > Trigger a notification (platform announcement/email/WeChat message)
  |
Display on the UI > View/Configure by the administrator > Modify the policy > Synchronize new configurations by the server

Product Dependencies
Users and Permissions

Dependent Product Name	Dependency Type	Dependency Relationship Description	Impact Level
Supporting TDSQL	Database	Depend on serviceid: dbsql-ocloud-tcenter-identityaccess dbsql-ocloud-tcenter-open-identity Specific database: t_auth2 Store user account information for the operation side.	High
Supporting Kafka	Cloud message queue	Depend on serviceid: kafka-ocloud-tcenter-identityaccess. - identity-group-topic - sen_cloudaudit Store audit information.	High
Supporting Redis®	Redis® cache	Depend on serviceId: ckv-tcenter-identityaccess. Store user login cache information.	High
CAM	CAM authentication	Account creation depends on CAM service authorization and other operations.	High
cas	CAS login	CAS login is required on the operation side.	High
National Cryptography Configuration	National cryptography configuration dependency	Depend on encryption and decryption for storage, supporting configuration encryption and decryption, transmission encryption and decryption, and signature verification.	High
foyo	Foyo configuration center	Depend on Foyo configuration. Used by operation side for configuration management view.	Medium

Audit Logs

Dependent Product Name	Dependency Type	Dependency Relationship Description	Impact Level
Supporting Kafka	Cloud message queue	Feature: Store audit logs. Depend on serviceid: kafka-tcenter-ocloud-cam kafka-cam Related topics: kafka-cam - auth_cloudaudit - signature_cloudaudit - sen_cloudaudit - cloudapi_req_production_json_log - audit_track - update_org_members kafka-tcenter-ocloud-cam - auth_ocloud_cloudaudit - signature_ocloud_cloudaudit - sen_ocloud_cloudaudit - cloudapi_ocloud_req_production_json_log	High
Supporting ES	Index management	Feature: Store audit logs. Depend on serviceid: es-ocloud-cloudaudit es-cloudaudit	High
Supporting Redis®	Data cache	Feature: Store cache data. Depend on serviceid: ckv-tcenter-ocloud-cam-log-yunapi ckv-cam-log-yunapi	High
CAM	Data dependency	Report specific audit logs.	High

CAM

Dependent Product Name	Dependency Type	Dependency Relationship Description	Impact Level
Supporting TDSQL	Database	Depend on serviceid: dbsql-ocloud-tcenter-cam dbsql-tcloud-tcenter-cam Specific database: cam	High
Supporting Kafka	Cloud message queue	Feature: Store authentication and audit log information. Depend on serviceid: kafka-tcloud-tcenter-identityaccess kafka-ocloud-tcenter-identityaccess kafka-tcenter-ocloud-cam kafka-cam Related topics: kafka-tcloud-tcenter-identityaccess - tcloud-identity-group-topic - tcloud_sen_cloudaudit kafka-ocloud-tcenter-identityaccess - identity-group-topic - sen_cloudaudit kafka-tcenter-ocloud-cam - auth_ocloud_cloudaudit - signature_ocloud_cloudaudit - sen_ocloud_cloudaudit - cloudapi_ocloud_req_production_json_log kafka-cam - auth_cloudaudit - signature_cloudaudit - sen_cloudaudit - cloudapi_req_production_json_log - audit_track - update_org_members	High
Tag	API dependency	Depend on serviceid: tcloud-tcenter-platform-tag	High

Dependent Product Name	Dependency Type	Dependency Relationship Description	Impact Level
Account	API dependency	Depend on serviceid: tcloud-tcenter-platform-account	High

Tag Management

Dependent Product Name	Dependency Type	Dependency Relationship Description	Impact Level
Supporting TDSQL	Database	Depend on serviceid: dbsql-tcenter-platform-account Specific database: qcloudTag	High
Supporting CMQ	Cloud message queue	Depend on service ID: cmq-tcenter-wtag. The service subscribes to CMQ to obtain tag information.	High

Cloud API Management

Dependent Product Name	Dependency Type	Dependency Relationship Description	Impact Level
Supporting TDSQL hp_api_formal	Database	Depend on serviceid: dbsql-tcenter-tcloud_api3, dbsql-tcenter-ocloud_api3 Specific database: hp_api_formal: Store yunapi API and backend service address information. api_sync: CRD synchronization information.	High
Kafka	Cloud message queue	Depend on service: cloudaudit_kafka. Topics: cloudapi_ocloud_req_production_json_log (operation side). cloudapi_req_production_json_log (tenant side). Used for cloud audit requirements.	High
CAM service	CAM authentication service	serviceid: ocloud-tcenter-cam-auth (operation side). tcloud-tcenter-cam-auth (tenant side). Enabling yunapi authentication requires the CAM service for authentication. CAM write: ocloud-tcenter-cam-grant-write CAM read: ocloud-tcenter-cam-grant Internal API calls the CAM API to perform read and write operations on CAM role information.	High
CAM service	CAM service for obtaining temporary keys	serviceid: tcloud-tcenter-cam-sts (tenant side). ocloud-tcenter-cam-sts (tenant side). Get a temporary key	High
Account	Account login authentication	serviceid: tcloud-tcenter-tcs-authn-80 (tenant side). ocloud-tcenter-platform-verify (operation side). Verify login status when calling cloud API.	High

Dependent Product Name	Dependency Type	Dependency Relationship Description	Impact Level
Account	Secondary authentication for sensitive operations	serviceid: tcloud-tcenter-platform-mfa (tenant side). ocloud-tcenter-platform-mfa (operation side).	High
Approval	Approval service	serviceid: ocloud-tcenter-approval (tenant side). ocloud-bsp-approval-svr (operation side).	High
Key	Key Management Service	serviceid: passwd-secret	Low

License Management

Dependent Product Name	Dependency Type	Dependency Description	Impact Level
Supporting TDSQL	Database	Dependent service ID: dbsql-tcenter-license Data storage	High
Key	Key service	Dependent service ID: passwd-secret Storage encryption/decryption, supporting configuration encryption/decryption, transmission encryption/decryption, signing, and signature verification	High
Message center	Message service	Dependent service ID: ocloud-tcenter-message-svr Alarm message sending	Medium

Troubleshooting
Impact Scope of Failures
Tag Management

No.	Scenario	Impact on the Cloud Product Data Plane	Impact on the Platform Control Plane
1	Tag page error	None	Affect customers' ability to view the tag page.
2	Tags not displayed after being added to a resource on the product page	None	Affect customers' ability to assign tags to specific resources.

License Management

No.	Scenario	Impact on the Cloud Product Data Plane	Impact on the Cloud Product Business Plane	Impact on the Platform Control Plane
1	License backend service down	None.	Cloud product scale-out is allowed by default if the license backend service is down.	Viewing the license-related information by customers is affected.
2	Slow license backend service response	None.	Cloud product scale-out is not smooth or times out.	None.
3	License data collection exception	The display of certain products that depend on the collected license usage data is affected.	The calculation for scale-out of certain products that depend on the collected license usage data encounters errors (the usage data is not updated).	None. The system prompts such exceptions of the corresponding product.

Failure Recovery
Scenario 1: Error When Opening the Users and Permissions Page or Creating an Account on the Operation Side

Scenario Description of Failure Symptoms

An error occurs when opening the User and Permissions page or when creating an account on the operation side.

Impact Scope of the Failure

- Impact on Cloud Platform product control: Affects the display of existing accounts and the creation of accounts. It may also impact login on the operation side.
- Impact on user's workload: None.

Failure Location and Analysis

1. The status of the related pod is abnormal.
2. Service rendering abnormality and database abnormality.

Failure Emergency Response Steps

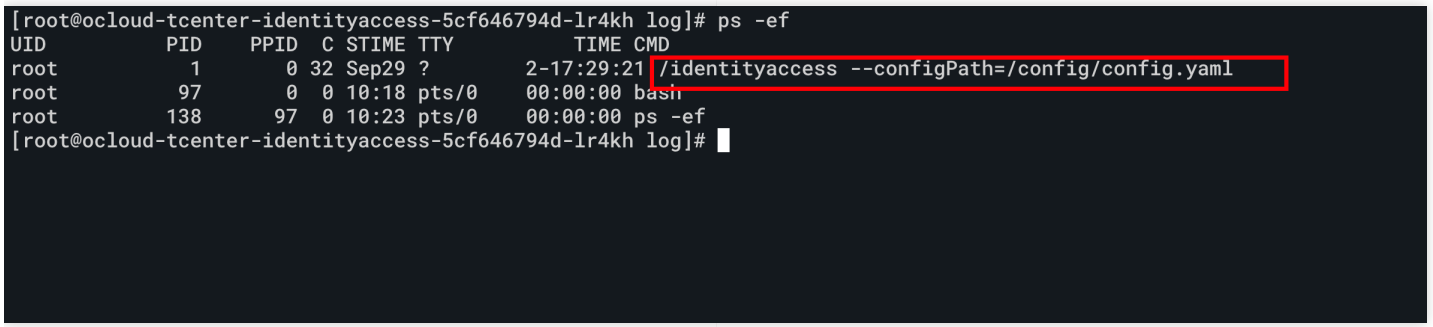
1. To quickly recover the workload, restart the related pod services.

```
kubectl get pod -n tce | grep -E 'cloud-tcenter-identityaccess|cloud-tcenter-open-identity'
kubectl delete pod -n tce ocloud-tcenter-identityaccess-xxxx/ocloud-tcenter-open-identity-xxx
```

2. If restarting the pod does not recover the workload, enter the pod and view the logs for further troubleshooting.

```
kubectl get pod -n tce | grep -E 'cloud-tcenter-identityaccess|cloud-tcenter-open-identity'
kubectl exec -it -n tce ocloud-tcenter-identityaccess-xxxx bash
```

```
# Check if the main service process is running.
ps -ef
cd /data/log/
# Query logs by error reqid.
grep -ir $reqid app.log
```



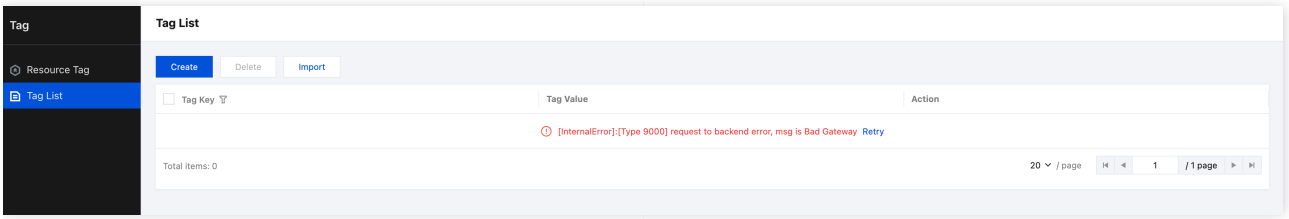
Failure Recovery Verification

The error on the page disappears, or the service feature is recovered.

Scenario 2: Tag Service Page Error

Scenario Description of Failure Symptoms

The tag service page reports an error:



Impact Scope of the Failure

- Impact on Cloud Platform product control: Affects tag querying and modification.
- Impact on user's workload: None.

Failure Location and Analysis

Backend service abnormality, containerized deployment:

1. The status of the pod is abnormal.
2. A single API feature of the service is abnormal.

Failure Emergency Response Steps

1. Check the pod status of the service: Check whether the pod is started normally and whether the pod status is normal.

   ```
   kubectl get pod -n tce | grep tag
   ```

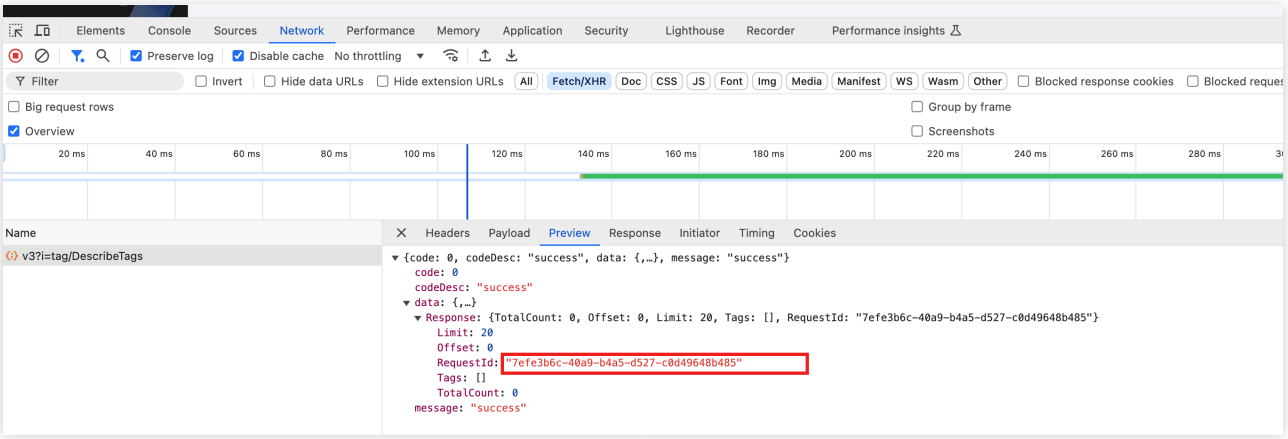
   - If the pod status is abnormal, delete and restart the pod.

     ```
     kubectl delete pod -n tce tcloud-tcenter-platform-tag-xxxxx
     ```

   - If the pod status is normal, use the RequestId from the page to query the service logs and troubleshoot the root cause:

     ```
     kubectl get pod -n tce | grep tcloud-tcenter-platform-tag
     ```

kubectl exec -it -n tce tcloud-tcenter-platform-tag-xxxx bash
cd /data/log/swoole_tag
grep -ir "{RequestId}" ./*



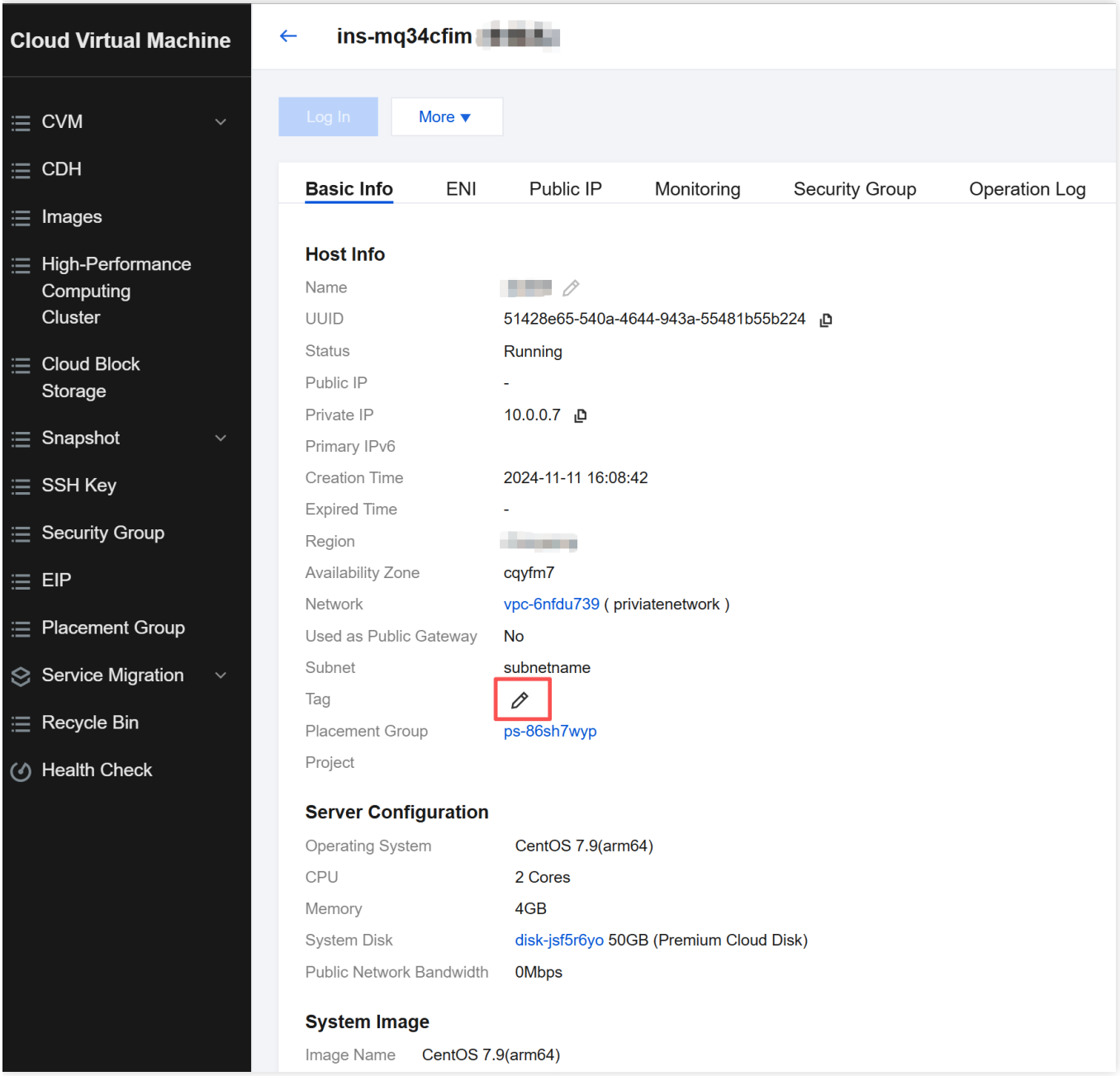
Failure Recovery Verification

The page feature is recovered.

Scenario 3: Tags Not Displayed After Being Added to a Resource on the Product Page

Scenario Description of Failure Symptoms

For example, after a tag is assigned to a specific instance on the CVM page, the page displays a success message, but the tag information is not displayed on the CVM page.



Impact Scope of the Failure

- Impact on Cloud Platform product control: Affects customers' ability to assign tags to specific resources.
- Impact on user's workload: None.

Failure Location and Analysis

1. Confirm whether the CMQ service is operating normally:
   kubectl get pod -n sso | grep cmq



2. Check whether the CMQ configuration in the tag service is correct:

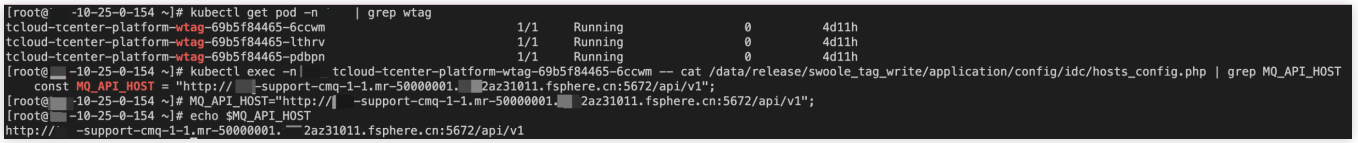
Obtain the pod name.

kubectl get pod -n tce | grep wtag

Select any running pod to view configuration.

kubectl exec -n tce tcloud-tcenter-platform-wtag-69b5f84465-6ccwm -- cat /data/release/swoole_tag_write/application/config/idc/hosts_config.php | grep MQ_API_HOST

Set the obtained MQ_API_HOST as an environment variable.



3. View mq subscription status:

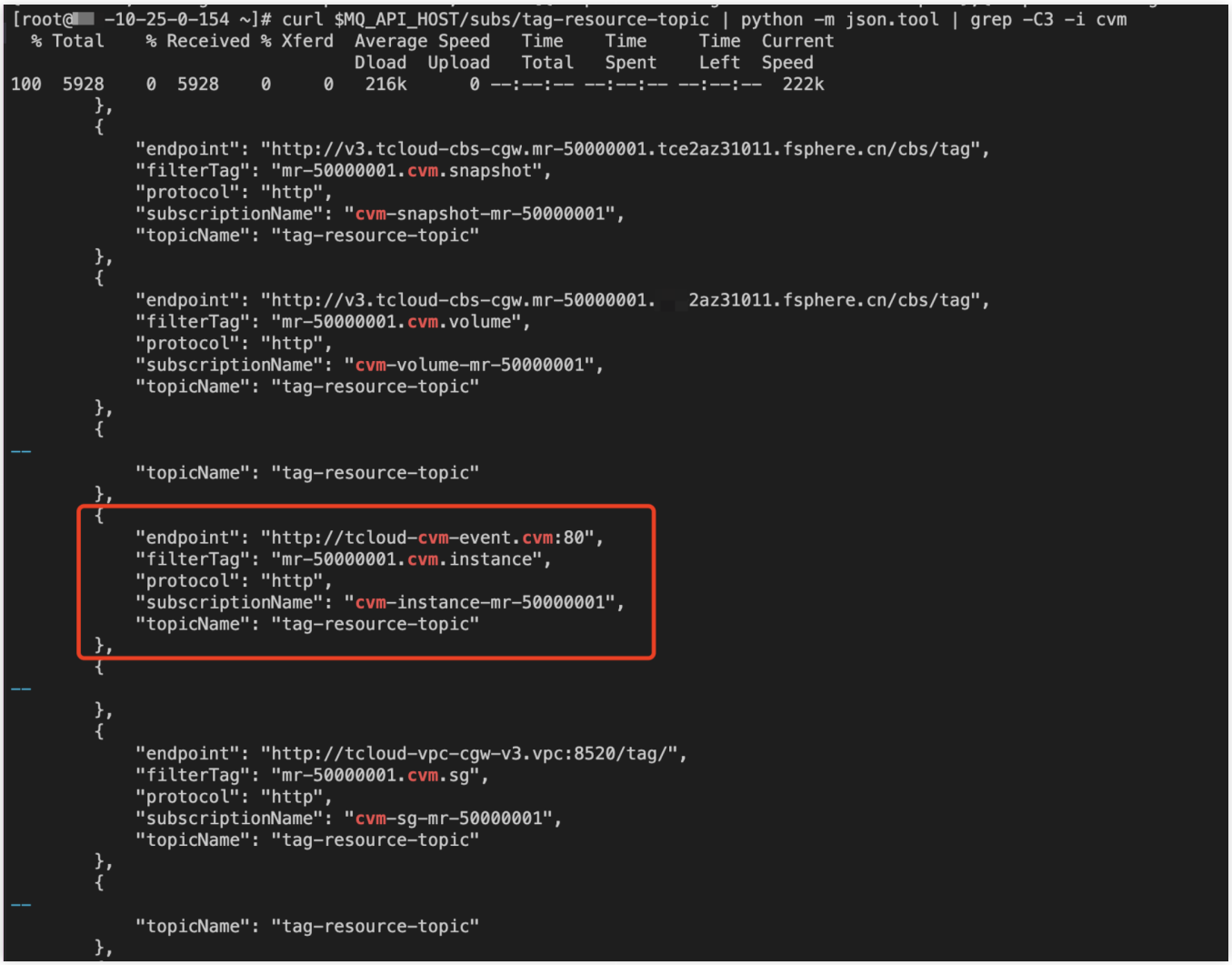
Confirm that tag-resource-topic exists; if not, submit a ticket to the platform side for handling.

curl $MQ_API_HOST/topics

```
[root@tcs-10-25-0-154 ~]# curl $MQ_API_HOST/topics
{"code":200,"message":"list topic success","data":[{"topicName":"tag-resource-ocloud-topic"},{"topicName":"tag-resource-topic"}]}[
```

Check subscription status. For example, check CVM's subscription status. If there is no subscription, submit a ticket to the CVM product side for handling.

curl $MQ_API_HOST/subs/tag-resource-topic | python -m json.tool | grep -C3 -i cvm



After confirming the subscription is normal, check whether CMQ callback logs are triggered.

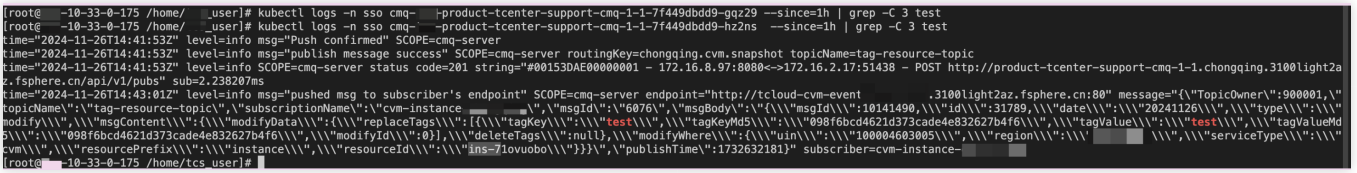
First, check the CMQ pod.

kubectl get pod -n sso | grep cmq

View logs; since=1h means logs from the past

hour. "test" is a sample tag name, or you can filter using instance ID. Multiple pods need to be filtered. If the following log appears, callback is successful. Submit a CVM ticket for further troubleshooting. If no log, submit a ticket to the platform side for handling.

kubectl logs -n sso cmq-tce-tce-support-cmq-1-1-xxx --since=1h | grep -C 3 test

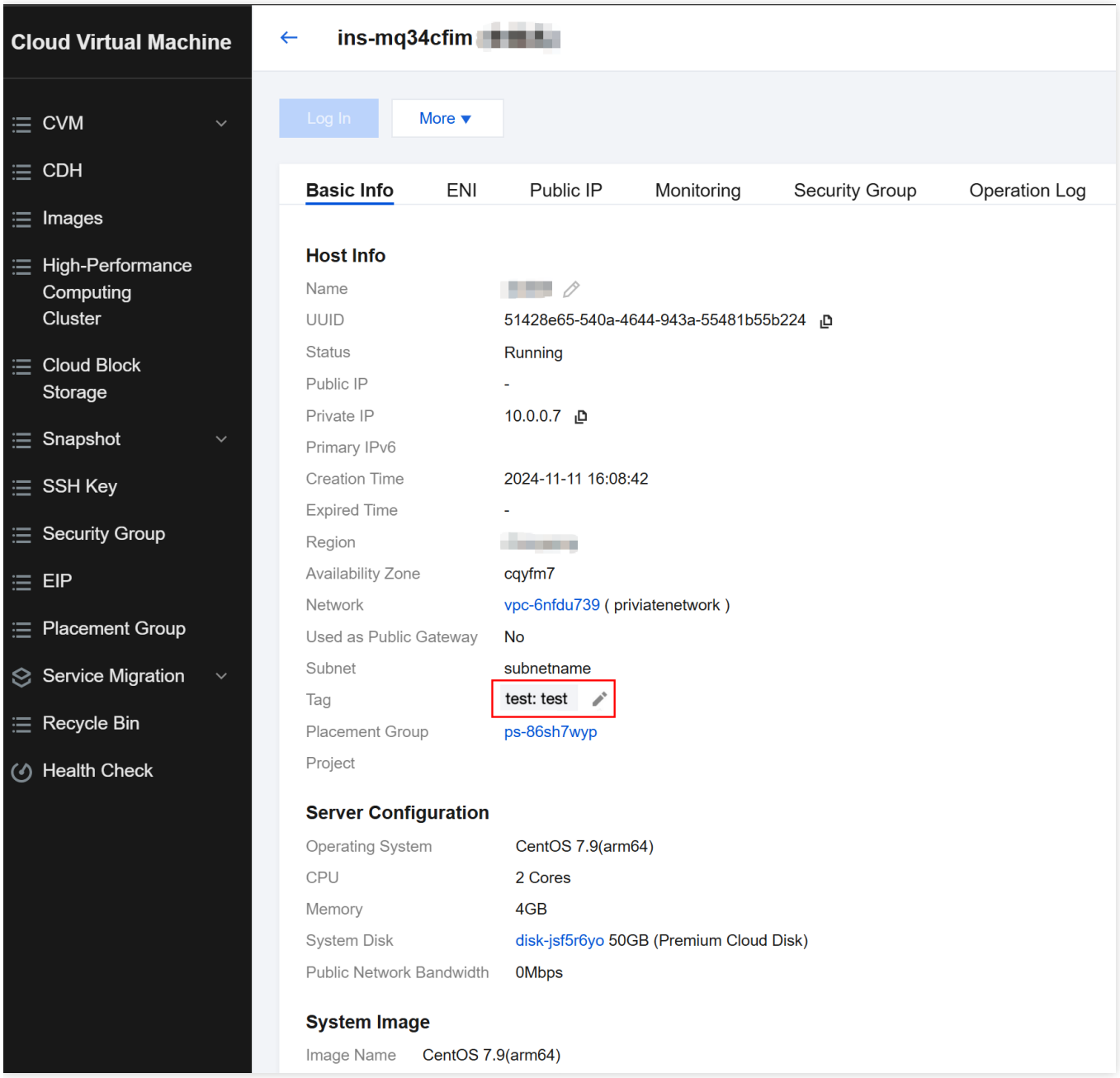


Failure Emergency Response Steps

1. If the CMQ topic is missing, you can create it manually:
   curl -X POST -d '{"topicName":"tag-resource-topic"}' $MQ_API_HOST/topics
2. If the CMQ subscription is missing, you need to confirm with the relevant product side to check which components need to be restarted to re-establish the subscription.
3. f the previous two steps are normal and there is no callback message from CMQ, as a temporary measure, consider restarting the pods related to the tag and CMQ to check if recovery is possible:
   kubectl rollout restart deploy -n tce tcloud-tcenter-platform-tag
   kubectl rollout restart deploy -n tce tcloud-tcenter-platform-wtag
   kubectl rollout restart deploy -n sso cmq-tce-tce-support-cmq-1-1

Failure Recovery Verification

The tag added to the corresponding resource on the product page can now be displayed correctly.


Cloud Virtual Machine
CVM
CDH
Images
High-Performance Computing Cluster
Cloud Block Storage
Snapshot
SSH Key
Security Group
EIP
Placement Group
Service Migration
Recycle Bin
Health Check
ins-mq34cfim
Log In
More
Basic Info
ENI
Public IP
Monitoring
Security Group
Operation Log
Host Info
Name
UUID 51428e65-540a-4644-943a-55481b55b224
Status Running
Public IP -
Private IP 10.0.0.7
Primary IPv6
Creation Time 2024-11-11 16:08:42
Expired Time -
Region
Availability Zone cqyfm7
Network vpc-6nfdu739 (priviatenetwork)
Used as Public Gateway No
Subnet subnetname
Tag test: test
Placement Group ps-86sh7wyp
Project
Server Configuration
Operating System CentOS 7.9(arm64)
CPU 2 Cores
Memory 4GB
System Disk disk-jsf5r6yo 50GB (Premium Cloud Disk)
Public Network Bandwidth 0Mbps
System Image
Image Name CentOS 7.9(arm64)

Scenario 4: Cloud API Product Unavailable, Affecting Platform Management Workload

Scenario Description of Failure Symptoms

All API calls via yunapi result in errors.

Impact Scope of the Failure

- Impact on Cloud Platform product control: The control is unavailable; all API calls through yunapi are unavailable.
- Impact on users' workload: No impact on the customer's existing data workload usage.

Failure Location and Analysis

- Check if the yunapi pod status is normal.
- Check if the network environment is functioning properly.

Failure Emergency Response Steps

1. Check if the pod status is normal. If the pod status is abnormal, you can delete and restart it to quickly recover the workload.

```
kubectl get pod -n tce | grep tcenter-yunapi3
kubectl delete pod -n tce xxxx--tcenter-yunapi3-traefik-xxxxx
```

2. If restarting does not recover the workload, access the container to check if the network connection from yunapi-traefik to other services is normal. If there is a network issue, fix the network. If the issue persists, view the service logs and recover the workload.

```
kubectl exec -it -n tce tcloud-tcenter-yunapi3-traefik-xxxx bash
cd /usr/local/services/traefik/log
# Filter logs by page reqid.
grep -ir "reqid" traefik.log
 grep -ir "reqid" *
```

Failure Recovery Verification

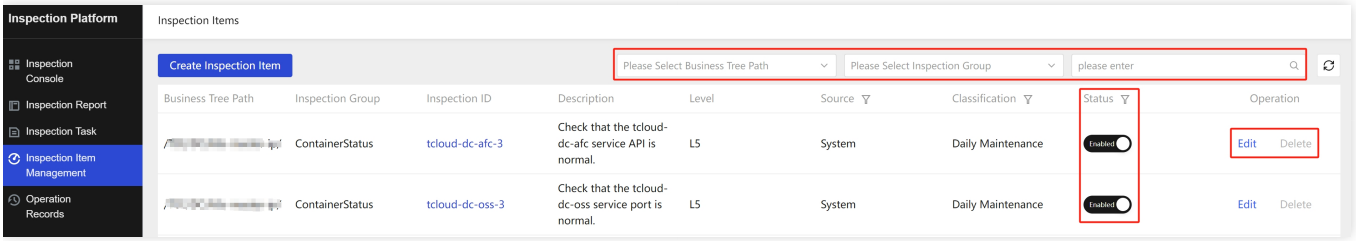
The page successfully calls the API through yunapi.

Daily Inspection
Cloud Product Inspection
Viewing Inspection Items

Navigate to Inspection Platform > Inspection Item Management, where inspection items are already built-in for cloud products, and you can use these inspection items to check the status of cloud products.

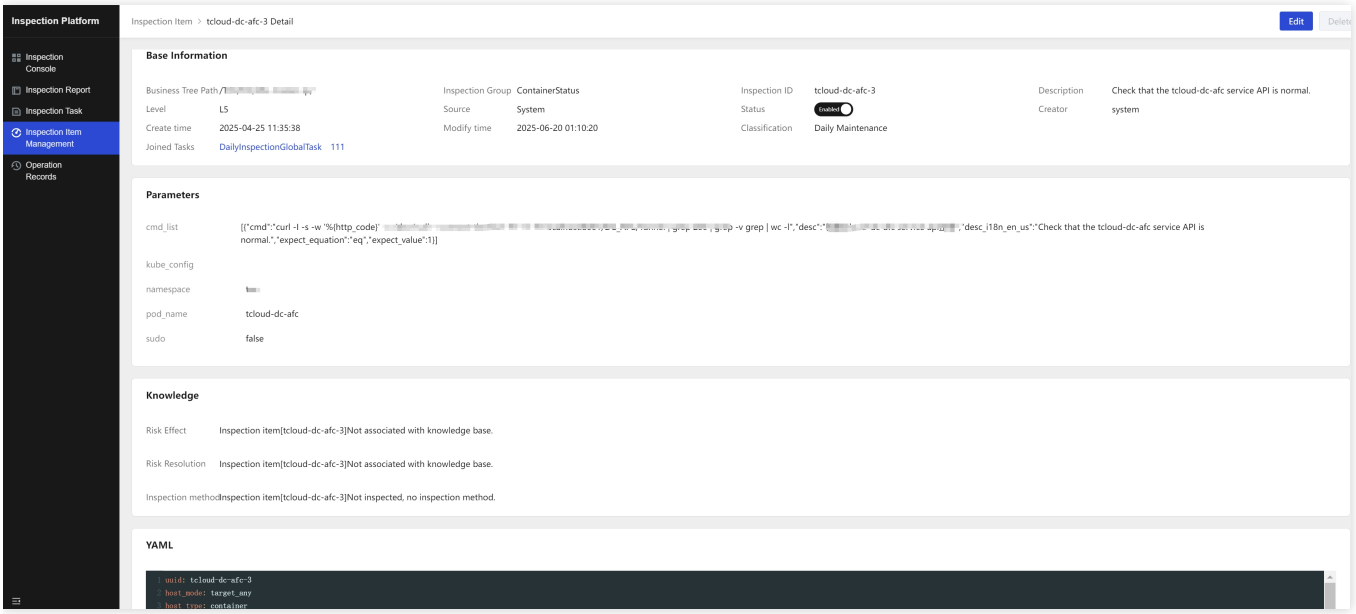
1 Inspection Item List

This module summarizes the list of inspection items on the current platform, allowing you to enable/disable, edit, add, and delete inspection items. By using the dropdown list to select the business tree path or inspection item group, you can filter the inspection items. Additionally, you can search for inspection items using keywords such as the inspection item description or inspection item group.



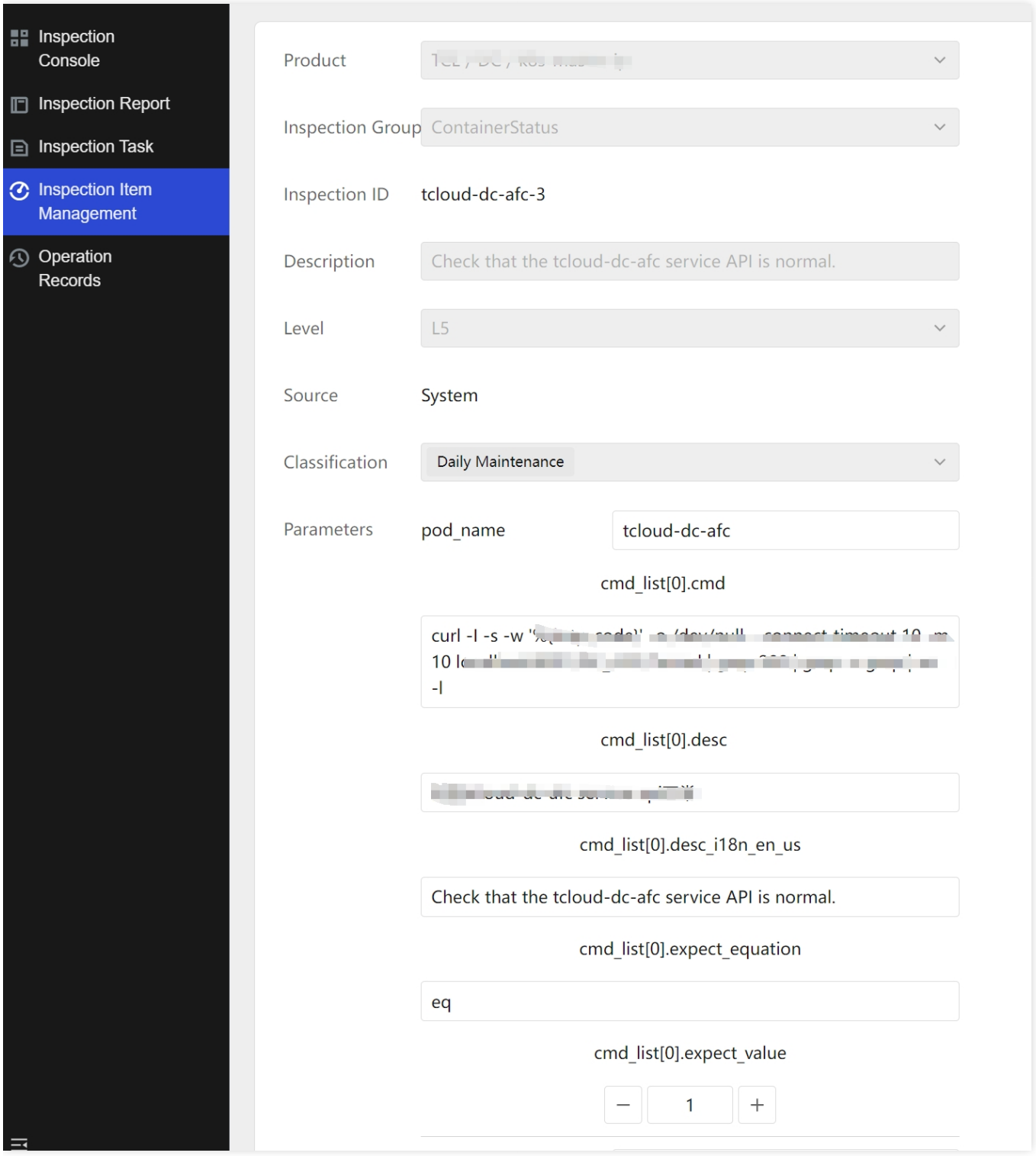
2 View Inspection Item Detailed Information

In the inspection item list, click the Inspection Item ID you want to view to enter the inspection item details page. The page displays inspection item information, including basic information, parameters, knowledge base, and YAML.



3 Edit Inspection Items

On the inspection item management page, click Edit to perform this operation.

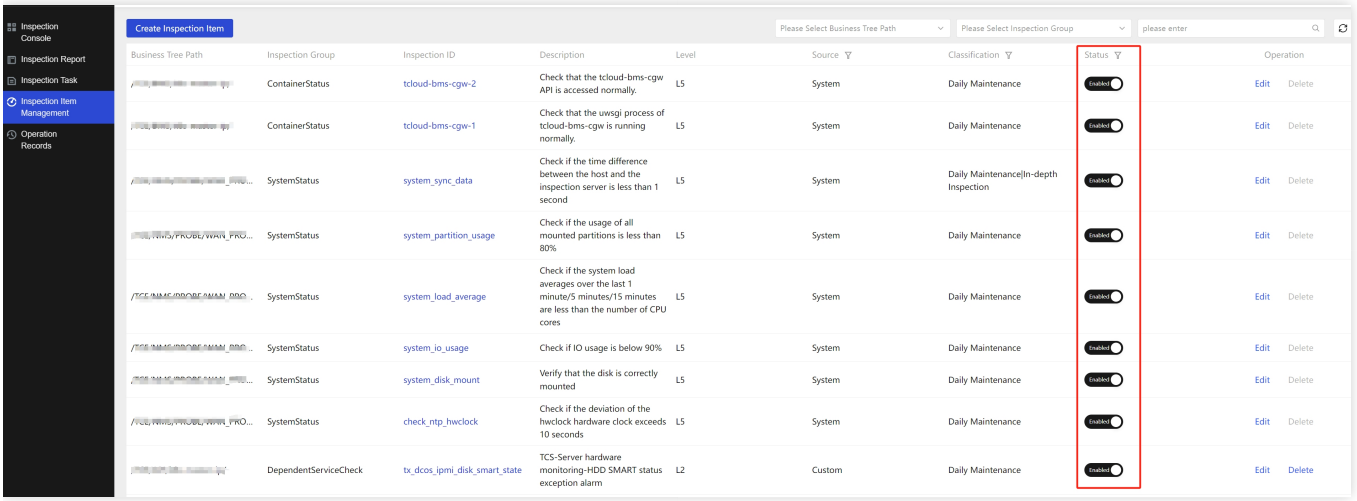


4 Enable/Disable Inspection Items

Enabling: Click the button switch in the disabled status to enable an inspection item that has been previously disabled. Once enabled, the inspection item will be executed during the next run of associated inspection tasks it has been added to.

Disabling: Click the button switch in the enabled status to disable an inspection task that has been previously

enabled. Once disabled, the inspection item will not be executed during the next run of associated inspection tasks it has been added to, and related abnormal items will be automatically cleared.

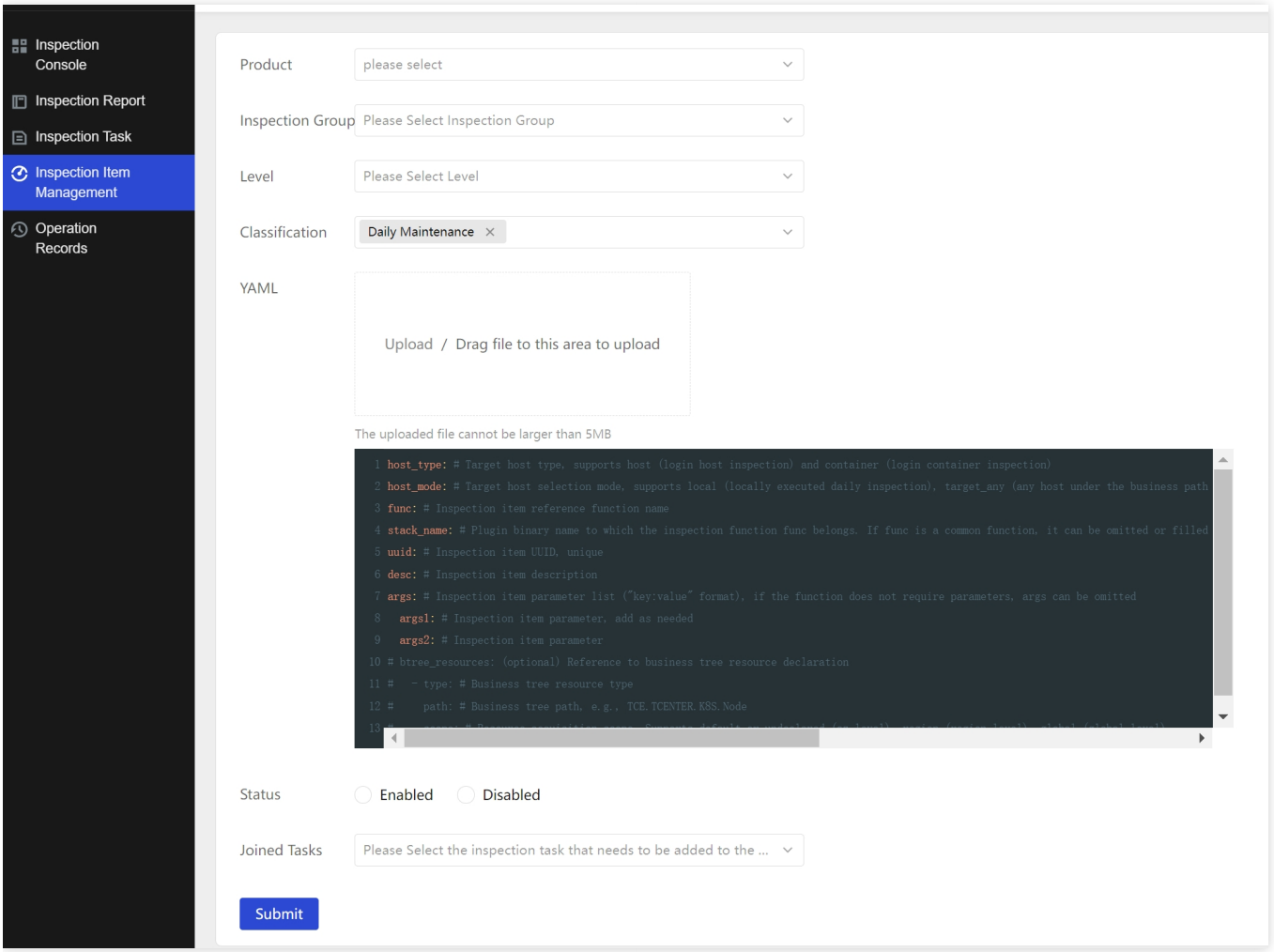


5 Add Custom Inspection Items

Automated management of inspection items is supported. Below is the guide for creating a custom inspection item.

Starting Adding Custom Inspection Items

Navigate to Inspection Platform > Inspection Item Management, and click Create Inspection Item.



The process for writing inspection items is as follows:

1. Inspection item product: Select the inspection target for the current inspection item.
2. Inspection item group: Select the group name for the inspection item.
3. Inspection item level: Select the inspection item level.
4. Category: Select the category (the default is Daily Ops).
5. Configuring the YAML File
    i. Find the appropriate inspection plugin and inspection function based on the inspection method.
        i. For remote command execution, typically use the system plugin's system_exec_command function.
        ii. For SQL execution, typically use the MySQL plugin's check_field_value function.
        iii. ......
    ii. Declare parameters for the inspection function.
    iii. Declare the description information and UUID (try to use meaningful English names for the UUID).
6. Enable inspection items and add them to the inspection task.

Configure the YAML file.

(1) Basic format

The YAML file contains the following fields:

host_type: host # Target host type. It can be set to host (for logging in to a host for inspections) or container (for logging in to a container for inspections). When the current path is k8s-master-ip or tianxun, host_type s hould be set to container.
host_mode: target_all # Selection mode for target hosts. It supports local (execute locally), target_any (select any host in the workload path of the current inspection item for execution), and target_all (select all hosts in the workload path of the current inspection item for execution).
func: MysqlExec # Name of the function referenced by the inspection item.
stack_name: checks-base # Binary name of the plugin of the inspection function func. If func is a general fun ction, this field can be omitted or filled in as checks-base. If it is a technology stack-specific function, the bin ary name of the technology stack should be filled in, such as tce-compute.
uuid: tx_cvm_podcheck_1 # (Required) UUID of the inspection item.
desc: Inspection item description # (Required) Description of the inspection item.
args: # Inspection item parameter list ("key: value" format). If func does not require any parameters, args can be omitted.
  database: yhimage
  sql: show tables;
btree_resources: (Optional) Declaration for reference of the workload tree resources.
- type: host # Type of the workload tree resources.
path: TCE.TCENTER.K8S.Node # Path of the workload tree.
scope: region # Resource acquisition scope. It supports the default value or undeclared (for AZ level), re gion (for region level), and global (for global level).

(2) Reference to workload tree resources (btree_resources)

host_type: host
host_mode: local
func: ExecuteShResultNum
btree_resources: # Here is the declaration for the host resources within the current region in the path TCE.TC ENTER.K8S.Node for the workload tree.
  - type: host
    path: TCE.TCENTER.K8S.Node
    scope: region
args:
  sh_list:
    - sh_name: "tgwadm_whitelist_check.sh"
      params: '{{ .btree_resources.host | map(.ip) | join(" ") }}' # Here, .btree_resources.host is used to referenc e the declared resource. The syntax is in jq format.
      expect_equation: eq
      expect_value: 1

(3) Common function formats

- Execute the shell command and compare the output result with the expected number.

  stack_name: checks-base
  func: SystemCmdResultNum
  args:
    cmd_list:

```
    cmd: "uptime | awk '{print $3/365}'"
    desc: "Whether the server has been running for more than 3 years."
    expect_equation: 'le'
    expect_value: 3
```

- Execute the shell command and compare the output result with the expected string.

```
stack_name: checks-base
func: SystemCmdResultStr
args:
  cmd_list:
    cmd: kubectl get secret -n cert-manager cert-manager-webhook-tls -o json | jq '.data["tls.crt"]' | x
args echo | base64 -d | openssl x509  -checkend 3888000
    desc: "Check whether the certificate expiration time for cert-manager is 45 days later."
    expect_equation: 'eq'
    expect_value: 'Certificate will not expire'
```

- Check whether the pod is alive.

```
stack_name: checks-base
func: PodActiveCheck
args:
  pod_name: 'tcloud-argus2-adp-synchronizer'
```

- Check whether the pod is alive, and execute shell commands inside the pod.

```
stack_name: checks-base
func: PodActiveCheck
args:
  cmd_list:
    - cmd: ps aux | grep 'gw-controller-monitor' | grep -v grep | awk '{print $11}' |sort | wc -l
      desc: Check whether the ocloud-vpc-nfv-gw-controller gw-controller-monitor process is running
normally.
      expect_equation: ge
      expect_value: 1
  pod_name: ocloud-vpc-nfv-gw-controller
```

- Copy the shell script and execute it, then check the execution results.

```
stack_name: checks-base
func: ExecuteShResultNum
args:
  sh_list:
    - sh_name: "tgwadm_whitelist_check.sh"
```

```
params: '"ip1" "ip2"'
expect_equation: eq
expect_value: 1
```

(4) Complete function list

- checks-base

```
CheckOsVersion
ClusterAllocationCheck
CpuFrequency
CpuUsage
CrontabGrep
DiskCapacity
DiskMount
DiskUsage
ElasticsearchCheck
EtcdCheck
ExecuteShResultNum
ExecuteShResultNumProxy
ExecuteShResultStr
ExecuteShResultStrProxy
GetK8sCert
GetOcloudCert
GetServerCert
GetTcloudCert
HttpApiResult
HttpApiResultNum
HttpApiResultStr
IOUsage
LoadAverage
MemSwapUsage
MemUsage
MysqlCheck
MysqlCheckDemo
MysqlCheckFieldValue
MysqlExec
PartitionUsage
PingCheck
PodActiveCheck
PodImageCheck
PodLogCheck
PodNodeCheck
PodReplicaCheck
PodRestartCheck
PodUsageCheck
PortCheck
ProcessCheck
```

PvcUsageCheck
RedisConfigCheck
RedisExecCheck
ServiceActiveCheck
SyncDate
SysLogCheck
SystemCmdResult
SystemCmdResultNum
SystemCmdResultNumProxy
SystemCmdResultStr
SystemCmdResultStrProxy
SystemModeCheck
ZookeeperCheck

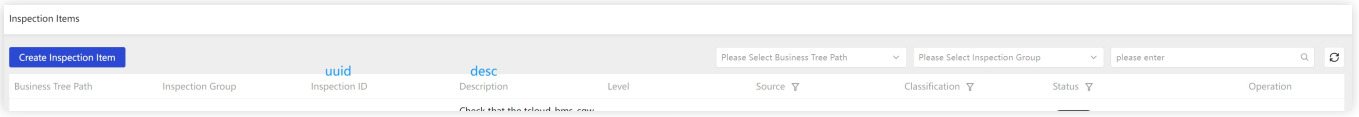
(5) Examples

Example 1: tx_tcenter_check_cert-manager_expire.

```
uuid: tx_tcenter_check_cert-manager_expire
host_mode: target_all
host_type: host
stack_name: tce-base
func: SystemCmdResultStr
args:
   cmd_list:
      - cmd: kubectl get secret -n cert-manager cert-manager-webhook-tls -o json | jq '.data["tls.crt"]' | xargs
echo | base64 -d | openssl x509  -checkend 3888000
        desc: Check whether the certificate expiration time for cert-manager is 45 days later.
        expect_equation: eq
        expect_value: Certificate will not expire
desc: Check whether the certificate expiration time for cert-manager is 45 days later.
btree_resources: []
```

Take the above inspection item as an example:

- This inspection item's UUID is set to tx_tcenter_check_cert-manager_expire, and its desc is set to 'Check whether the certificate expiration time for cert-manager is 45 days later'. The UUID and desc will be displayed on the inspection list page.



- This inspection item's host_mode is set to target_all, indicating that this inspection item will ultimately be executed on all machines in the associated business tree path. If it were set to target_any, this inspection item would be executed on any machine in the associated business tree path.
- This inspection item's host_type is set to host, indicating that the target of this inspection item is a host. In addition to host, the inspection platform also supports a container type for host_type, indicating that the

target is a container or special node.

- This inspection item's stack_name is set to tce-base, and its func is set to SystemCmdResultStr. This indicates that the implementation of the inspection is located in a binary file named tce-base, and the function being called is named SystemCmdResultStr.
- The subsequent args field contains parameters passed to the SystemCmdResultStr function. Similar to section 2.2, the SystemCmdResultStr function supports executing multiple shell commands, with each command specified using cmd. The expected output for these commands is 'Certificate will not expire'.

Example 2: tx_cbs_pod_processor

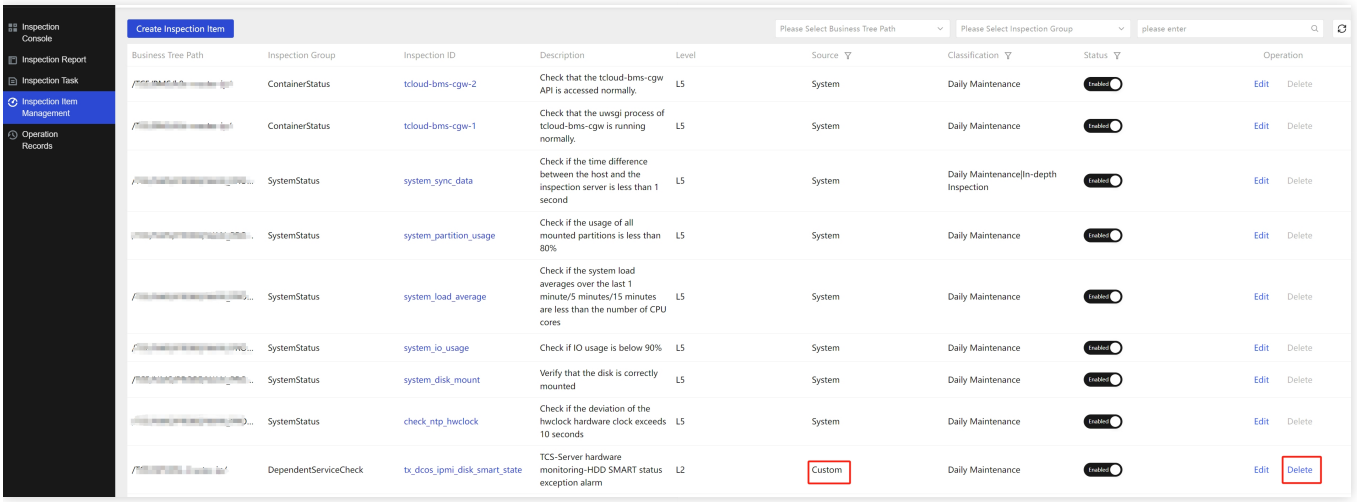
```
uuid: tx_cbs_pod_processor
host_mode: target_any
host_type: container
stack_name: tce-storage
func: PodActiveCheck
args:
  cmd_list:
    - cmd: df -h | grep '/' |tr -d '%' | awk '{if ($5>80)print}' | wc -l
      desc: Check whether the number of mounted partitions whose pod container disk utilization is greater
than 80% is 0.
      expect_equation: eq
      expect_value: 0
  pod_name: ocloud-cbs-processor
desc: The pod ocloud-cbs-processor container status is running, and the disk utilization is less than 80%.
btree_resources: []
```

Here, we only list the parts that differ between this inspection item and the previous one:

- The current inspection item's host_mode is set to target_any, and its host_type is set to container. This indicates that the inspection needs to be executed only once and should run inside the container (pod).
- The current inspection item's func is set to PodActiveCheck, and its args is set to the parameters passed to PodActiveCheck. This function also supports the simultaneous declaration of multiple shell commands.

6. Deleting Inspection Items

Click delete to delete a custom inspection item. System inspection items cannot be deleted. If you do not need to use them, you can disable them.

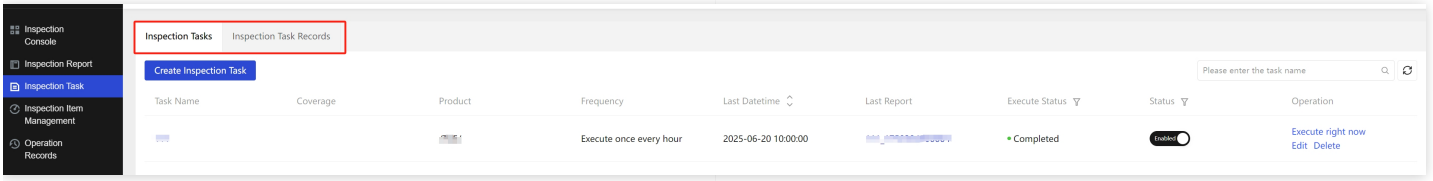

Inspection Console
Inspection Report
Inspection Task
Inspection Item Management
Operation Records
Create Inspection Item
Please Select Business Tree Path
Please Select Inspection Group
please enter
Business Tree Path
Inspection Group
Inspection ID
Description
Level
Source
Classification
Status
Operation
ContainerStatus
tcloud-bms-cgw-2
Check that the tcloud-bms-cgw API is accessed normally.
L5
System
Daily Maintenance
Enabled
Edit
Delete
ContainerStatus
tcloud-bms-cgw-1
Check that the uwsgi process of tcloud-bms-cgw is running normally.
L5
System
Daily Maintenance
Enabled
Edit
Delete
SystemStatus
system_sync_data
Check if the time difference between the host and the inspection server is less than 1 second
L5
System
Daily Maintenance|In-depth Inspection
Enabled
Edit
Delete
SystemStatus
system_partition_usage
Check if the usage of all mounted partitions is less than 80%
L5
System
Daily Maintenance
Enabled
Edit
Delete
SystemStatus
system_load_average
Check if the system load averages over the last 1 minute/5 minutes/15 minutes are less than the number of CPU cores
L5
System
Daily Maintenance
Enabled
Edit
Delete
SystemStatus
system_io_usage
Check if IO usage is below 90%
L5
System
Daily Maintenance
Enabled
Edit
Delete
SystemStatus
system_disk_mount
Verify that the disk is correctly mounted
L5
System
Daily Maintenance
Enabled
Edit
Delete
SystemStatus
check_ntp_hwclock
Check if the deviation of the hwclock hardware clock exceeds 10 seconds
L5
System
Daily Maintenance
Enabled
Edit
Delete
DependentServiceCheck
tx_dcos_ipmi_disk_smart_state
TCS-Server hardware monitoring-HDD SMART status exception alarm
L2
Custom
Daily Maintenance
Enabled
Edit
Delete

Viewing Inspection Tasks

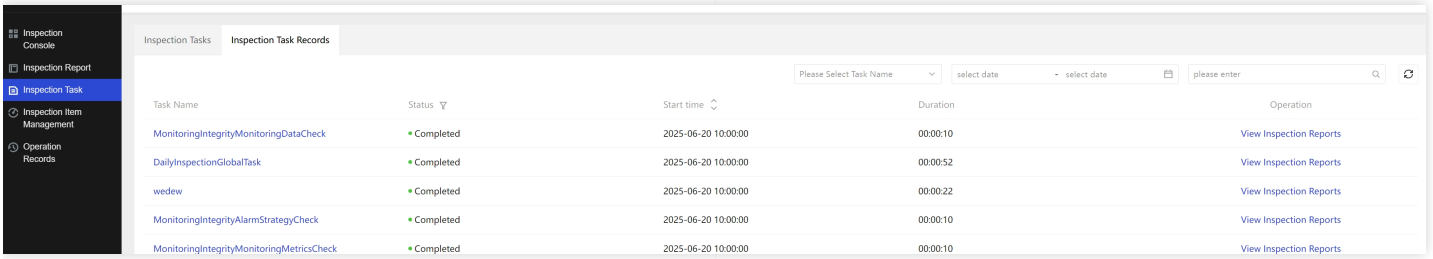
Navigate to Inspection Platform > Inspection Tasks. It includes both the system built-in inspection tasks and user-defined inspection tasks.

1. List of Inspection Tasks

Manages all inspection tasks and inspection task execution records on the platform and supports visualized management of inspection task execution. On the interface, you can directly view the inspection task execution records and perform operations such as inspection task creation, enabling/disabling, immediate execution, editing, and deletion.



">



2. Inspection Task Details Page

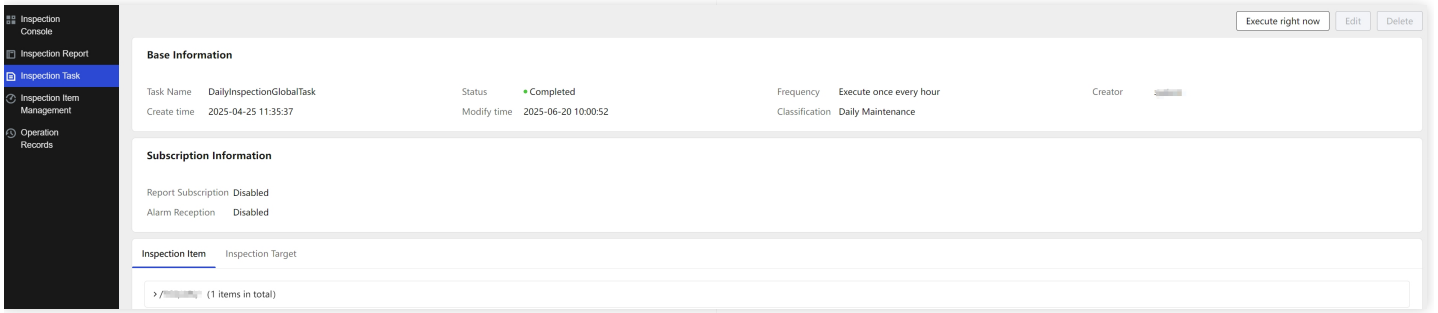
In the inspection task list, click the Task Name you want to view to enter the inspection task detail page.

(1) Basic Information displays the task name, status, inspection frequency, creation time, and other information of the inspection task.

(2) Subscription Information displays the report subscription and alarm subscription configuration status of the inspection task.

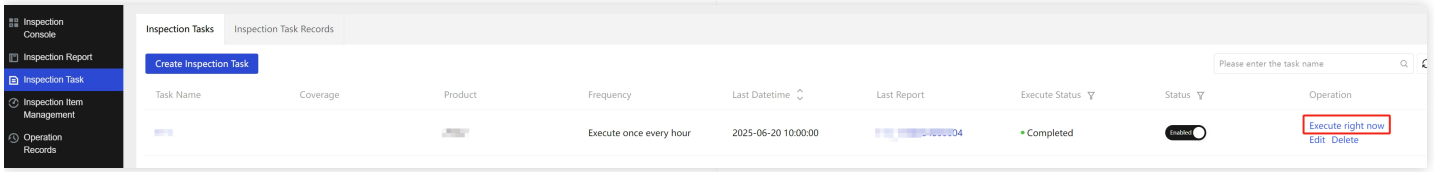
(3) Inspection Item List displays inspection items in the task based on the product node.

(4) Inspection Target List displays inspection targets in the task based on the product node.



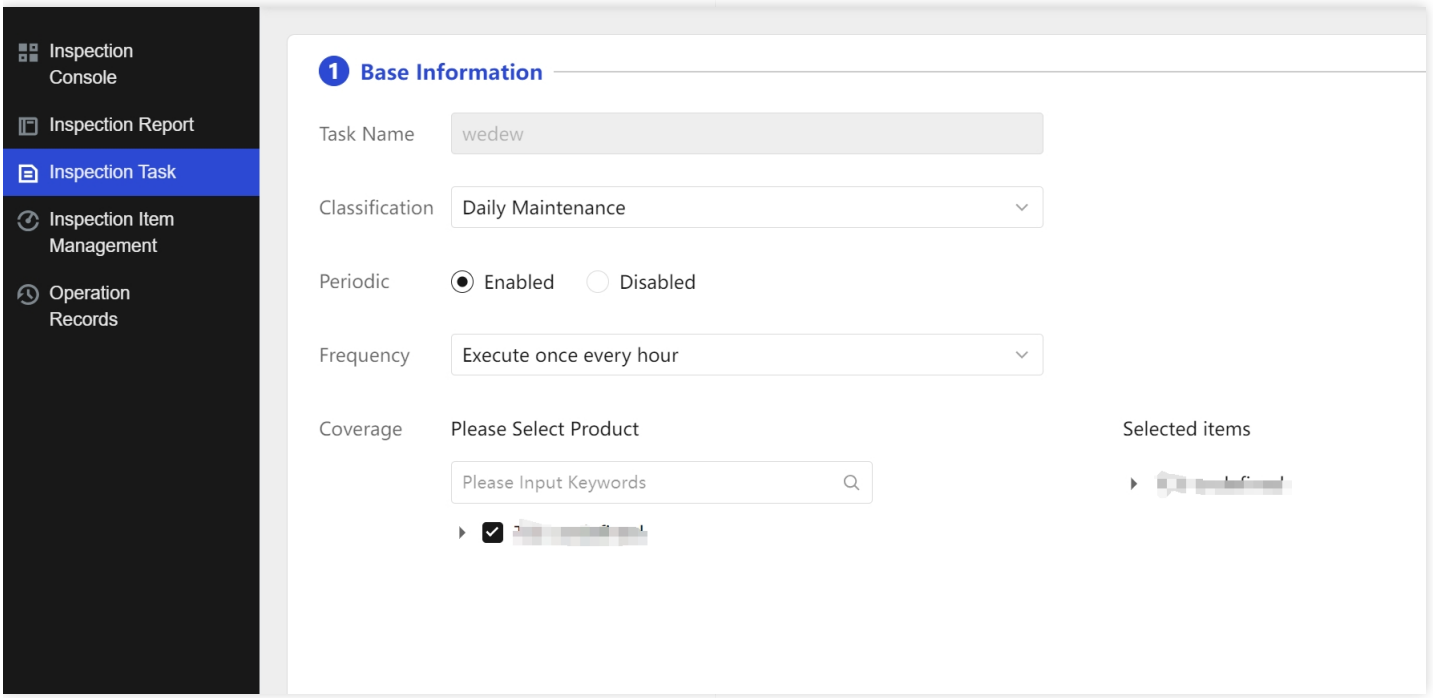
3 Execute inspection tasks

Click the immediate execution button for the inspection task. After confirmation, the execution of the inspection task will be triggered immediately. After execution, the inspection results will be displayed in the inspection console and the inspection report module.



4 Edit Inspection Task

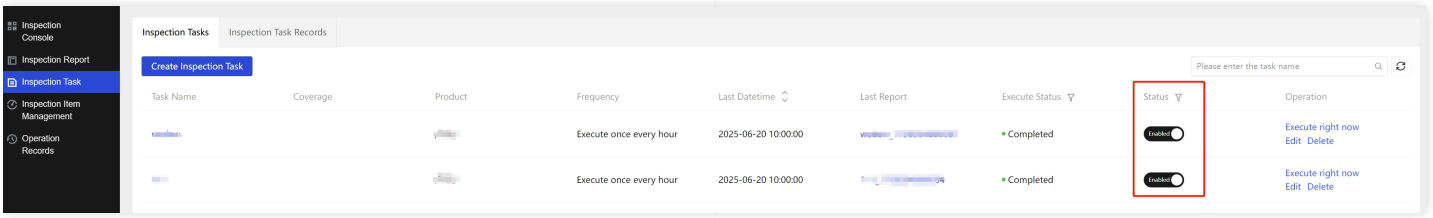
Click the Edit button for the inspection task to edit/modify the task.



When submitting the modifications, select Execute Task Immediately. If you select Yes, the modified inspection task will be executed immediately and will follow the updated information; if you select No, the modified inspection task will take effect the next time it is executed.

5 Enable/disable inspection task

Enabling: Click the toggle button in the disabled status to enable a disabled inspection task. The inspection task will be executed according to the inspection frequency.
Disabling: Click the toggle button in the enabled status to disable an enabled inspection task. The inspection task will no longer be executed.



6 Create Inspection Task

Support visualized management of inspection tasks. The guidelines for creating a custom inspection task are shown below.

Access it via Inspection Platform - Inspection Task and click Create Inspection Task.

(1) Fill in the Basic Information of the Inspection Task

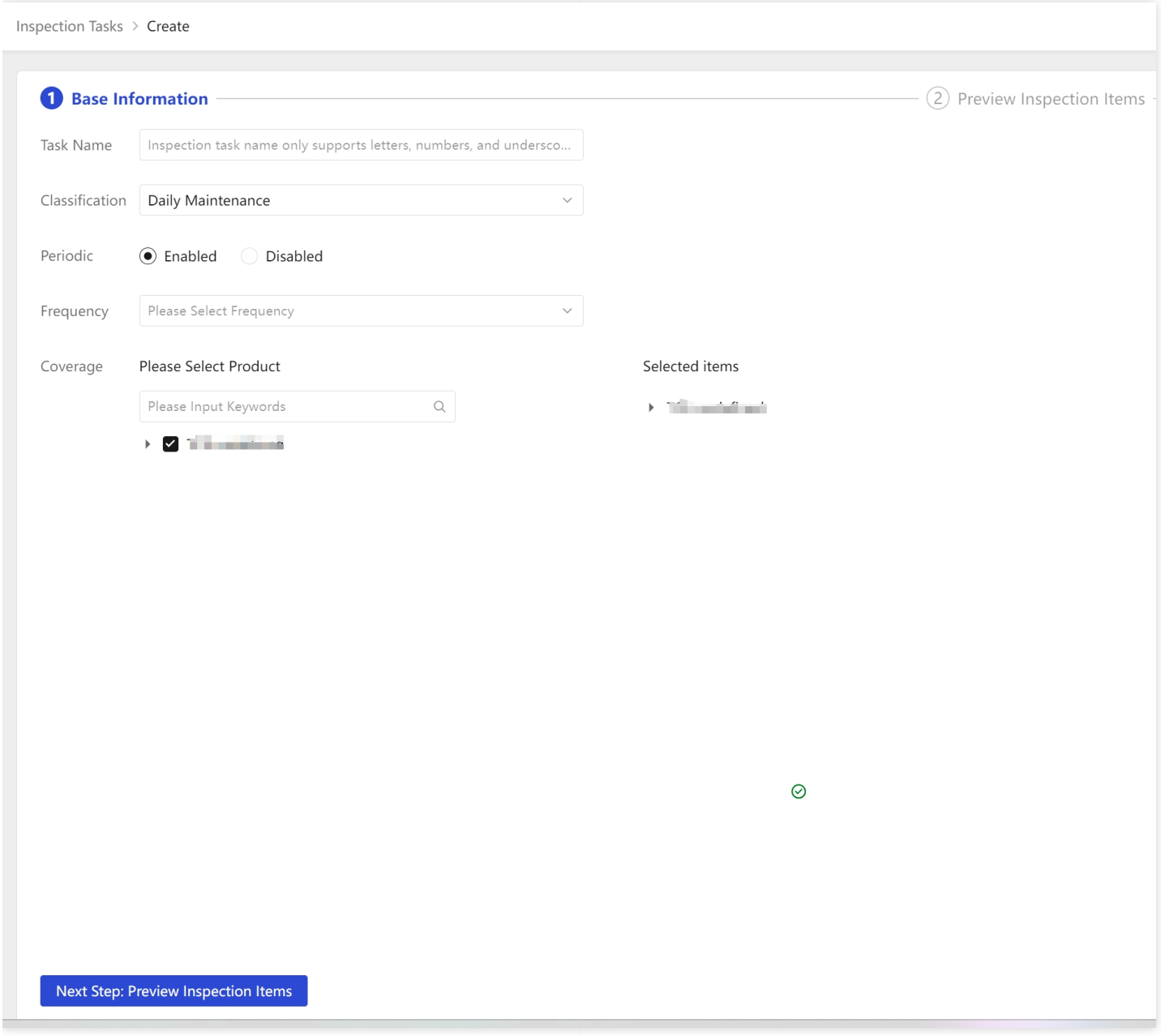
Task name: Enter the name of the inspection task.

Category: Select in the dropdown list.

Scheduled automatic inspections: Select whether to enable scheduled inspections.

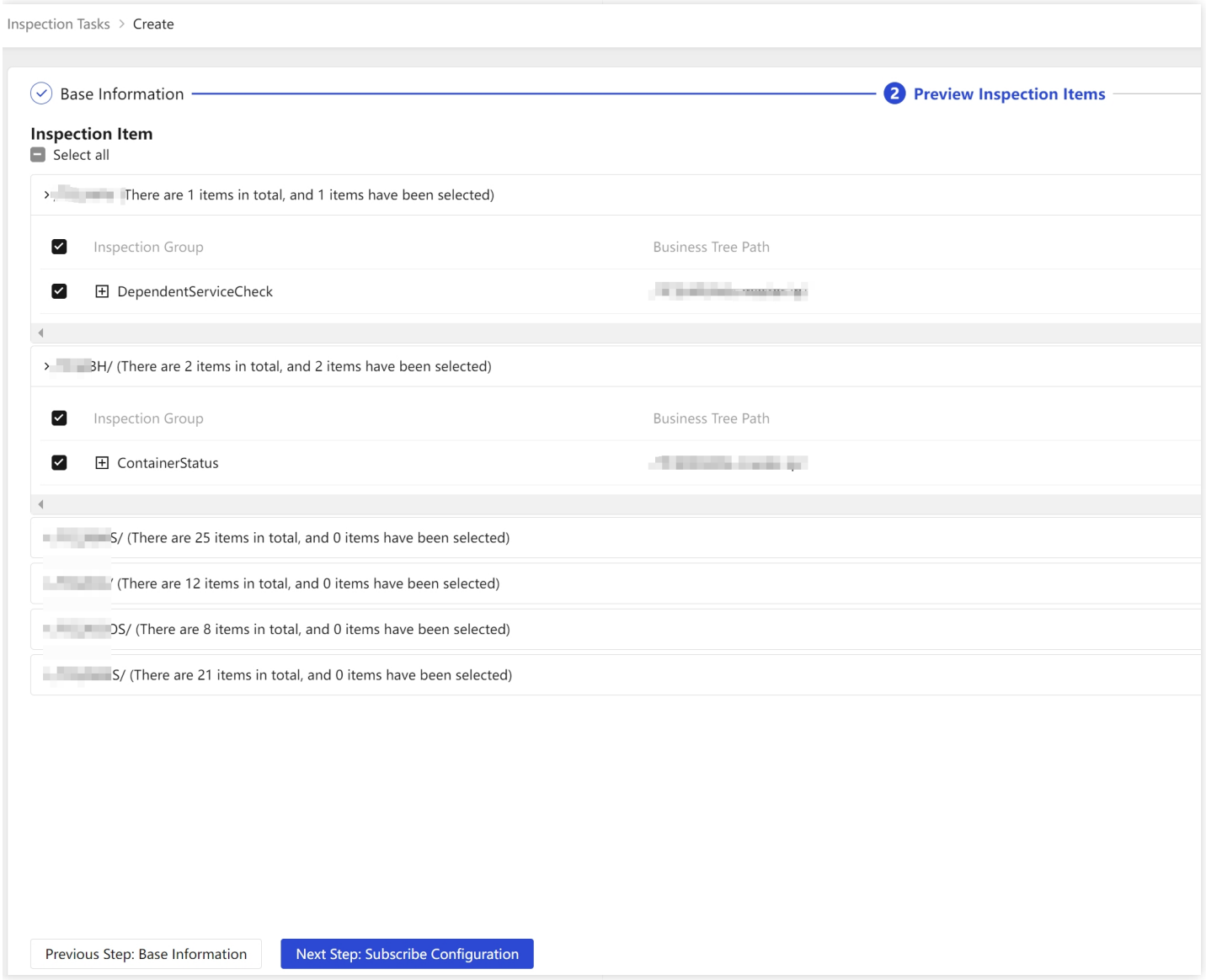
Inspection frequency: Select the frequency of automatic inspections for the inspection task in the dropdown list.

Inspection scope: Select the business tree scope for this inspection task according to information such as the product cluster and product node. You need to select business tree nodes that contain machines.



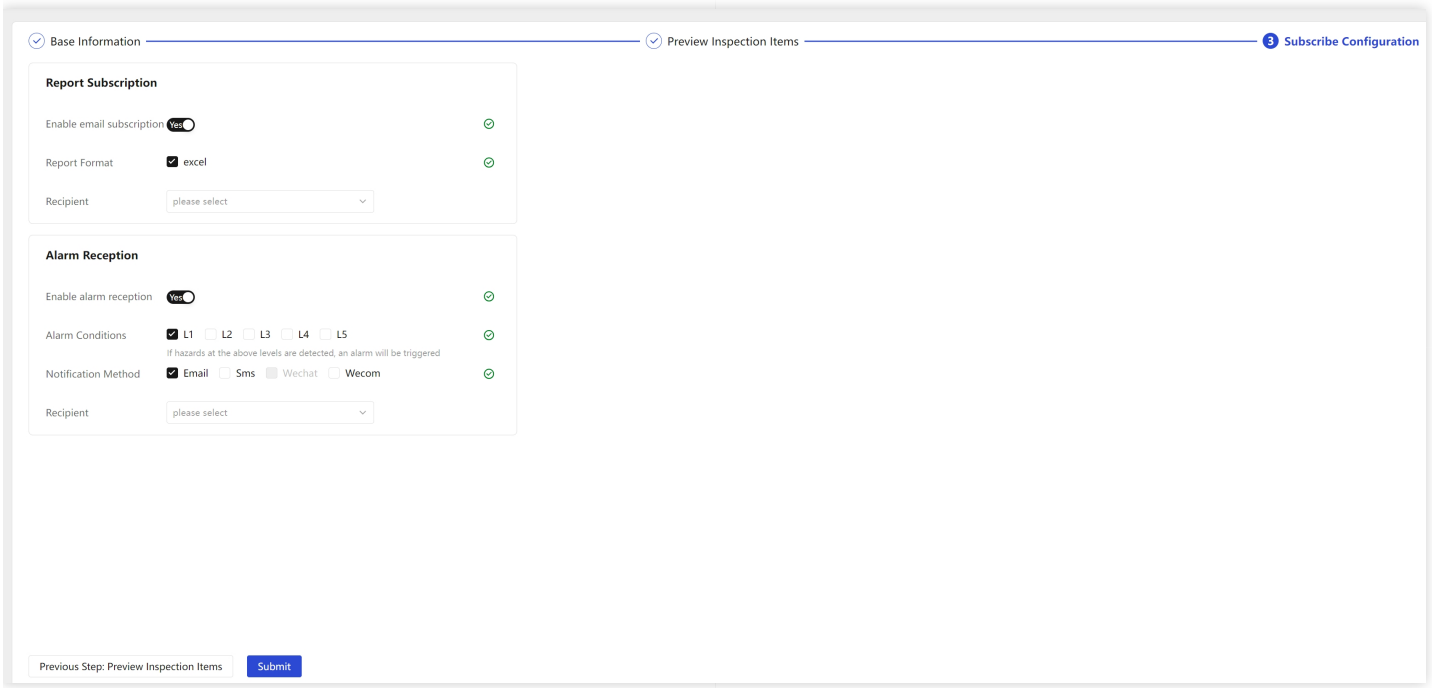
(2) Select Inspection Items

Check the inspection items to be executed.



(3) Setting the Subscription

Set the information for Report Subscription and Alarm Reception according to your needs.

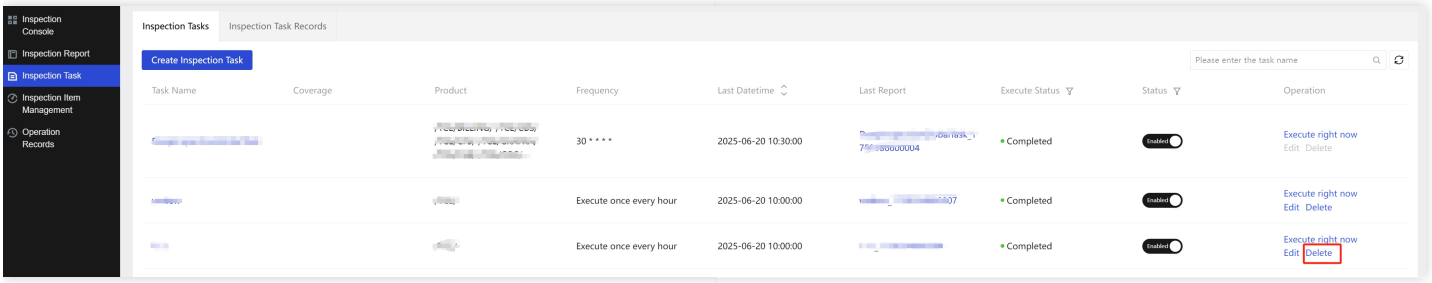


(4) Submitting an Inspection Task

When submitting the inspection task, you can select whether to execute it immediately. After selecting immediate execution and submitting it, the inspection task will be executed immediately.

7 Delete Task

Click the Delete button for the inspection task to delete the selected inspection task. Once deleted, the task cannot be recovered. Therefore, proceed with caution. Inspection reports that are generated by the executed inspection tasks will not be deleted. (Note: Default inspection tasks initially provided by the system cannot be deleted.)

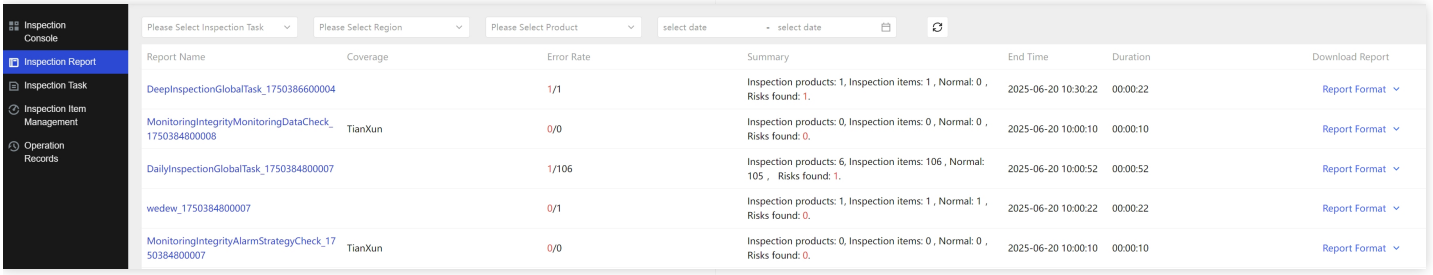


Viewing Health Check Results

Choose Inspection Platform > Inspection Report. The execution results of inspection tasks will be displayed in the form of inspection reports. You can directly view the inspection results on the frontend page, or download inspection reports to review the inspection results.

1 Inspection Report List

You can view the inspection report list data, including the report name, inspection coverage, hazard detection ratio, result summary, completion time, and inspection duration. On the page, you can query data by inspection task, site, inspected product, and time.

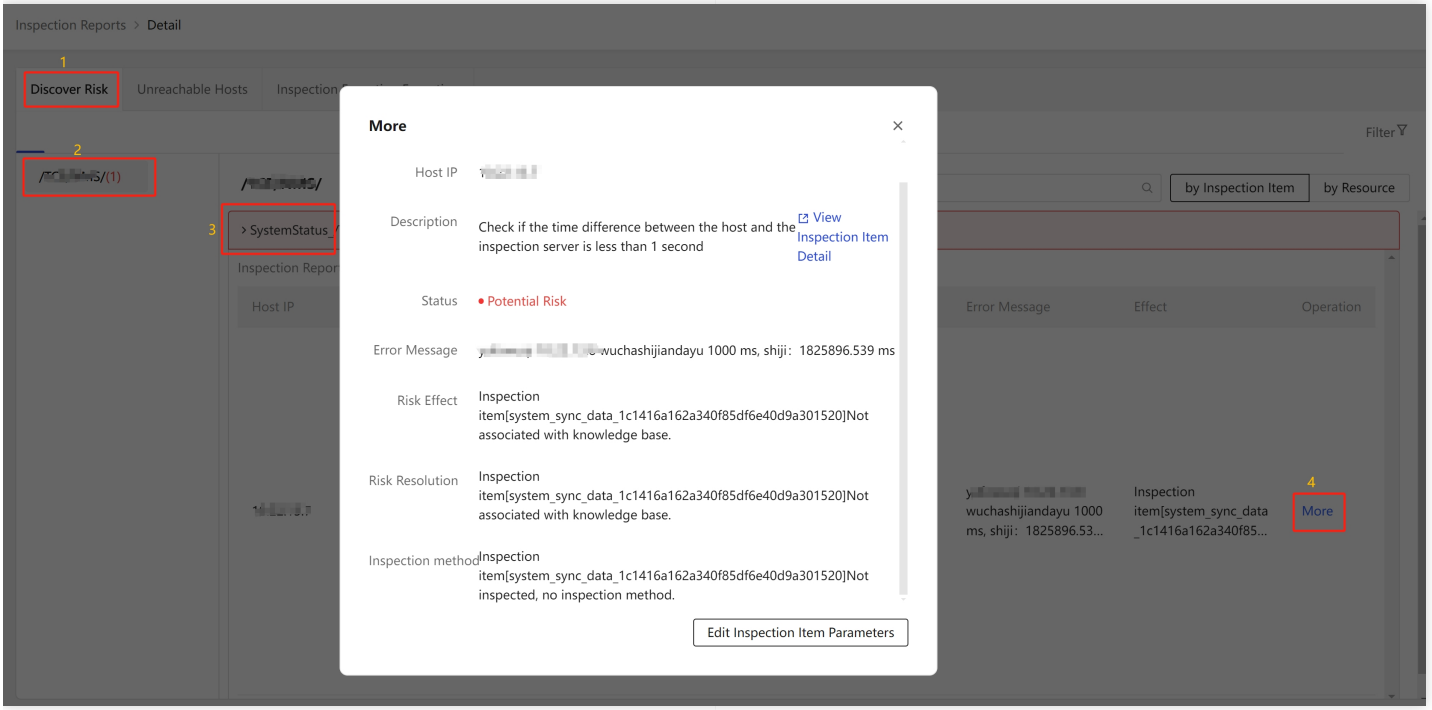


2 Details Page of Inspection Report

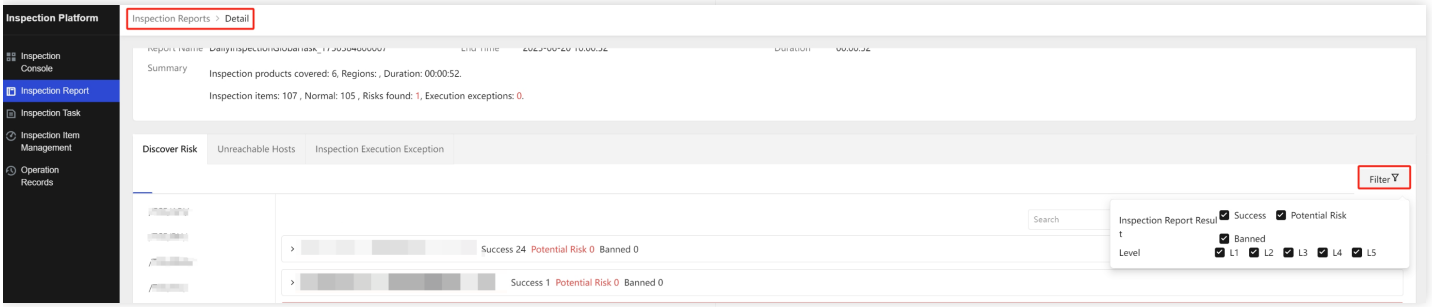
Click Report Name in the inspection report list to enter the inspection report details page, where the current inspection summary, identified hazards, unreachable host list, and abnormal inspection executions are displayed.

(1) Identified Hazards

If a red number appears after the product name on the current page and it is not 0, it indicates that the current inspection task has identified hazards in the inspected product. You can click the corresponding Product to expand the red inspection group, and then click More for inspection items with abnormal status to view detailed inspection results.

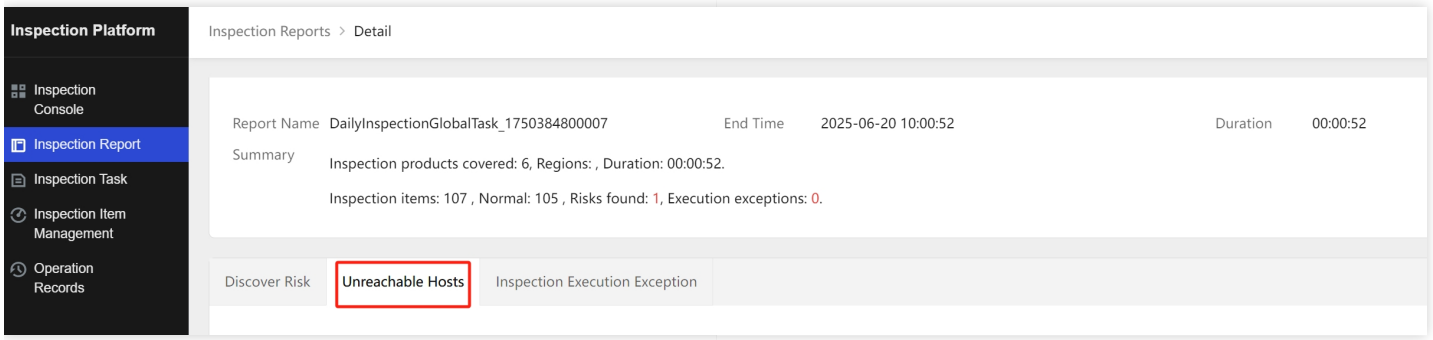


On the Inspection Report Details page, you can filter by condition by checking the Inspection Result Status and Inspection Item Grading.



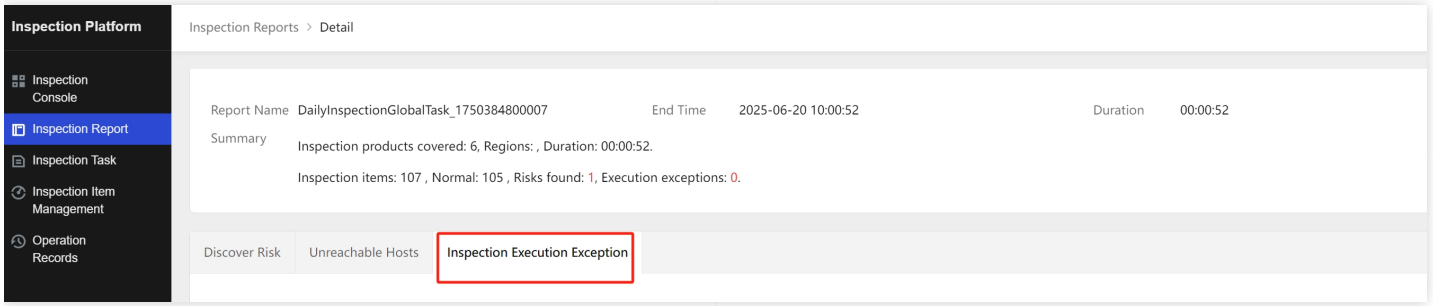
(2) Unreachable Host List

Click Unreachable Host List on the inspection report details page. You can see the information of hosts that failed to connect during the current inspection task. Use the error messages in the list to troubleshoot issues such as network connectivity or password-free configuration settings.



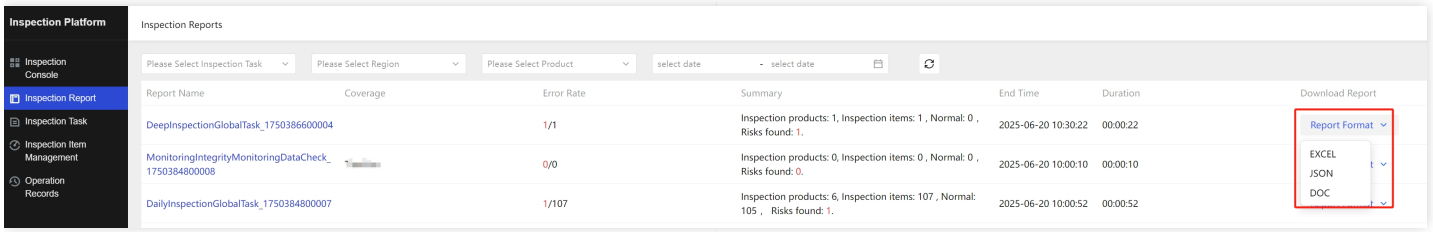
(3) Abnormal Inspection Executions

Click Inspection Execution Anomaly Items on the inspection report details page. You can see a summary of inspection items where host retrieval failed or inspection execution failed within the current inspection report.



3 Download the Inspection Report

Go to the Inspection Report page and click Report Format to download the inspection report in the corresponding format.



TCenter Inspection Item Checklist

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
check_ntp_hwclock	Checking whether the hwclock (hardware clock) deviation exceeds 10 seconds	System parameter check	Daily Ops	TCENTER/Zookeeper/zookeeper TCENTER/Zookeeper/instance TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/TDSQL/Instance TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/ips

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				TCENTER/CSP/gw_hosts TCENTER/CRedis/interface TCENTER/CRedis/cache
drms_tce-tcenter-hdfs_check_enable_topology_4f970b7a	Checking whether rack awareness is enabled	Disaster recovery check	Disaster recovery check	TCENTER/HDFS/namenode
drms_tce-tcenter-k8s_check_apiserver_39c40585	Checking the rdata configuration of the apiserver	Disaster recovery check (standalone)	Disaster recovery check	TCENTER/K8S/Master
drms_tce-tcenter-k8s_check_cluster_global_dnsnameservers_178906da	Checking the dnsNameservers in the cluster global configuration	Disaster recovery check (standalone)	Disaster recovery check	TCENTER/K8S/Master
drms_tce-tcenter-k8s_check_coredns_c728ad1d	Checking the health check configuration for CoreDNS	Disaster recovery check (standalone)	Disaster recovery check	TCENTER/K8S/Master
drms_tce-tcenter-k8s_check_kube_dns_d5dbbfbf	Checking the Kube DNS	Disaster recovery check (standalone)	Disaster recovery check	TCENTER/K8S/Master
drms_tce-tcenter-k8s_check_registry_1bd407c1	Checking the tcs-internal-registry	Disaster recovery check (standalone)	Disaster recovery check	TCENTER/K8S/Master
system_cpu_usage	Checking whether the CPU utilization is greater than 90%	System parameter check	Daily Ops	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
system_disk_mount	Checking whether disks are correctly mounted	System parameter check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
system_io_usage	Checking whether the I/O utilization is less than 90%	System parameter check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
system_load_average	Checking whether the average system load for the last 1 minute/5 minutes/15 minutes is less than the number of CPU cores	System parameter check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
system_mem_usage	Checking whether the memory utilization is less than 80%	System parameter check	Daily Ops and in-depth inspection	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product-tcenter-support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
system_partition_usage	Checking whether the utilization of all mount partitions is less than 80%	System parameter check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product-tcenter-support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
system_sync_data	Checking whether the time difference between the host and the inspection server is less than 1 second	System parameter check	Daily Ops and in-depth inspection	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
tx_barad_es_ping	Checking whether the Elasticsearch node ping latency is below 500 ms	Dependency service check	Daily Ops	TCENTER/ Elasticsearch/ etcd
tx_barad_zk_config_state	Checking the node configuration of the ZooKeeper cluster	Dependency service check	Daily Ops	TCENTER/ Zookeeper/ instance
tx_barad_zk_node_count	Checking whether the number of Kafka directories in the ZooKeeper cluster is set to 3	Dependency service check	Daily Ops	TCENTER/ Zookeeper/ instance
tx_barad_zk_node_exist	Checking whether the storm110, Kafka, and kfkSpout directories exist in the ZooKeeper cluster/ directory	Dependency service check	Daily Ops	TCENTER/ Zookeeper/ instance
tx_barad_zk_partition_usage	Checking whether the disk utilization of the ZooKeeper cluster for both / and /data is below the threshold	Dependency service check	Daily Ops	TCENTER/ Zookeeper/ instance
tx_barad_zk_snapshot_state	Checking the snapshot size of the ZooKeeper cluster	Dependency service check	Daily Ops	TCENTER/ Zookeeper/ instance
tx_barad_zk_state	Checking the node status of the ZooKeeper cluster	Dependency service check	Daily Ops	TCENTER/ Zookeeper/ instance
tx_check_mysql_status_cam_auth	Checking the database of the TCenter cam auth component	Dirty data check	Daily Ops	TCENTER/k8s-master-ip
tx_check_mysql_status_cam_grant	Checking the database of the TCenter cam grant component	Dirty data check	Daily Ops	TCENTER/k8s-master-ip
tx_check_mysql_status_cam_list	Checking the database of the TCenter cam list component	Dirty data check	Daily Ops	TCENTER/k8s-master-ip
tx_check_mysql_status_cam_sts	Checking the database of the TCenter cam sts component	Dirty data check	Daily Ops	TCENTER/k8s-master-ip

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_check_mysql_status_platform_account	Checking the database of the TCenter platform account component	Dirty data check	Daily Ops	TCENTER/k8s-master-ip
tx_get_svc-ignore-sync-ip	Checking whether infra.tce.io and ignore-sync-ip are configured in the SVC	Security check	Daily Ops	TCENTER/K8S/Master
tx_kafka_check_replicationfactor	Checking whether the number of Kafka replicas on physical servers is not equal to 1	Service status check	Daily Ops	TCENTER/Kafka/kafka
tx_pass_tmp_safe	Checking whether the passwords for users root and tunnel_user have been stored in the database and whether a password risk assessment has been conducted	Security check	Daily Ops	TCENTER/K8S/Master
tx_Redis®_base_cc_sql_interface_balance_status	Checking for the uneven distribution of interfaces in the newcc database	Dirty data check	Daily Ops	TCENTER/CRedis/oss
tx_Redis®_base_cc_sql_status_interface_procs_capacity	Checking whether the interface utilization in the newcc database is greater than 80%	Dirty data check	Daily Ops	TCENTER/CRedis/oss
tx_Redis®_base_cc_sql_status_Redis®_procs_capacity	Checking whether the Redis® process utilization in the newcc database is greater than 80%	Dirty data check	Daily Ops	TCENTER/CRedis/oss
tx_Redis®_base_cc_sql_status_task_state	Checking whether the event tasks in the newcc database are normal	Dirty data check	Daily Ops	TCENTER/CRedis/oss
tx_Redis®_base_interface_status_cli_num	[Collection] Checking whether the number of interface connections exceeds 5,000	Capacity check	Daily Ops	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_latency	[Collection] Checking whether the interface latency is below 200 ms	IDC network check	Daily Ops	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_maxfd	[Alarm] Checking whether the maximum limit for interface fd is configured to exceed 10,000	System parameter check	Daily Ops	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_mem	[Collection] Checking whether the interface memory is greater than 5 GB	Capacity check	Daily Ops	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_rafficlimit	[Alarm] Checking whether the interface has generated limit logs due to traffic throttling	Capacity check	Daily Ops	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_slowlog	[Alarm] Checking whether the interface has recorded over 50 slow queries within one hour	Capacity check	Daily Ops	TCENTER/CRedis/interface

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_Redis®_base_interface_status_stats_avg_latency_6	[Alarm] Checking whether the single-process request time of the interface exceeds 500 ms	Capacity check	Daily Ops	TCENTER/ CRedis/ interface
tx_Redis®_base_interface_status_stats_clirates	[Alarm] Checking whether the single-process connection utilization on the interface is greater than 80%	Capacity check	Daily Ops	TCENTER/ CRedis/ interface
tx_Redis®_base_interface_status_stats_input_limit_rate	[Alarm] Checking whether the single-process input bandwidth utilization on the interface is greater than 80%	Capacity check	Daily Ops	TCENTER/ CRedis/ interface
tx_Redis®_base_interface_status_stats_output_limit_rate	[Alarm] Checking whether the single-process output bandwidth utilization on the interface is greater than 80%	Capacity check	Daily Ops	TCENTER/ CRedis/ interface
tx_Redis®_base_interface_status_stats_qps	[Alarm] Checking whether the single-process QPS on the interface exceeds 50,000	Capacity check	Daily Ops	TCENTER/ CRedis/ interface
tx_Redis®_base_server_status_check_client_biggest_input_buf_status	[Collection] Checking whether the maximum input buffer of Redis® exceeds 100 MB	Capacity check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_base_server_status_check_client_longest_output_list_status	[Collection] Checking whether the number of objects in the Redis® output buffer exceeds 1,000	Capacity check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_base_server_status_check_connected_clients_ratio_status	[Alarm] Checking whether the number of clients connected to Redis® is greater than 50% of the upper limit	Capacity check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_base_server_status_check_instantaneous_input_kbps_status	[Collection] Checking whether the input bandwidth of Redis® exceeds 500 Mbps	Capacity check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_base_server_status_check_instantaneous_ops_per_sec_status	[Collection] Checking whether the request QPS of Redis® is higher than 60,000	Capacity check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_base_server_status_check_instantaneous_output_kbps_status	[Collection] Checking whether the output bandwidth of Redis® exceeds 500 Mbps	Capacity check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_base_server_status_check_latest_fork_usec_status	[Collection] Checking whether the fork latency of Redis® is longer than 1 second	Capacity check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_base_server_status_check_logdiskbusy_status	[Alarm] Checking whether a busy blockage has occurred during the	Capacity check	Daily Ops	TCENTER/ CRedis/cache

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
	persistence of Redis®			
tx_Redis®_base_server_status_check_master_aof_enable	[Alarm] Checking whether AOF is enabled for the master Redis® instance with a performance impact	Service configuration check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_base_server_status_check_Redis®_limit_maxfd	[Alarm] Checking whether the fd limit set for Redis® exceeds 10,000	Capacity check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_base_server_status_check_repl_offset_status	[Alarm] Checking whether the primary-secondary offset of Redis® is greater than 10 MB	Service configuration check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_base_server_status_check_slowlog_status	[Collection] Checking whether the number of slow queries in Redis® exceeds 20	Capacity check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_base_server_status_check_used_memory_status	[Collection] Checking whether the memory usage of Redis® exceeds 30 GB	Capacity check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_cc_agent_check_cc_agent_timer	Checking whether the agent process (cc_agent_timer) that the Redis® cc depends on is active	Process check	Daily Ops	TCENTER/ Product/ product-tcenter-support-cRedis®/ oss_nodes
tx_Redis®_cc_agent_check_cc_monitor_py	Checking whether the agent process (cc_monitor_py) that the Redis® cc depends on is active	Process check	Daily Ops	TCENTER/ Product/ product-tcenter-support-cRedis®/ oss_nodes
tx_Redis®_cc_agent_check_defunct_proc	Checking for zombie cc_monitor_py processes in the Redis® cc	Process check	Daily Ops	TCENTER/ Product/ product-tcenter-support-cRedis®/ oss_nodes
tx_Redis®_cc_agent_check_influxdb	Checking whether the agent process (influxd) that the Redis® cc depends on is active	Process check	Daily Ops	TCENTER/ Product/ product-tcenter-support-cRedis®/ oss_nodes
tx_Redis®_cc_agent_check_mul_server	Checking whether the agent process (mul_server) that the Redis® cc depends on is active	Process check	Daily Ops	TCENTER/ Product/ product-tcenter-support-cRedis®/ oss_nodes
tx_Redis®_cc_agent_check_qds_center	Checking whether the agent process (qds_center)	Process check	Daily Ops	TCENTER/ Product/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
	that the Redis® cc depends on is active			product-tcenter-support-cRedis®/oss_nodes
tx_Redis®_cc_agent_check_web	Checking whether the agent process (web) that the Redis® cc depends on is active	Process check	Daily Ops	TCENTER/Product/product-tcenter-support-cRedis®/oss_nodes
tx_Redis®_cc_agent_version_statics_cc_monitor_py	[Collection] Checking the version of cc_monitor_py for the cc	Process check	Daily Ops	TCENTER/Product/product-tcenter-support-cRedis®/oss_nodes
tx_Redis®_cc_sql_status_app_nums	Checking for the existence of dirty instance data in the instances of the newcc database	Database check	Daily Ops	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_app_state	Checking whether the instances in the newcc database are normal	Database check	Daily Ops	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_interface_machine_status	Checking for non-1 statuses of interface machines in the newcc database	Database check	Daily Ops	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_interface_proc_status	Checking for non-1 statuses of interface processes in the newcc database	Database check	Daily Ops	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_machine_salerate	Checking whether the machine sales rate in the newcc database is greater than 80%	Database check	Daily Ops	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_master_port	Checking whether the port configuration for the master process in the newcc database is correct	Database check	Daily Ops	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_master_slave	Checking for inconsistencies between primary and secondary processes in the Redis®_procs_t of the newcc database	Database check	Daily Ops	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_Redis®_machine_status	Checking for non-1 statuses of Redis® machines in the newcc database	Database check	Daily Ops	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_Redis®_proc_status	Checking for non-1 statuses of Redis® processes in the newcc database	Database check	Daily Ops	TCENTER/CRedis/oss

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_Redis®_cc_sql_status_replication_check	Checking whether the number of replicas is consistent in the newcc database	Database check	Daily Ops	TCENTER/ CRedis/oss
tx_Redis®_cc_sql_status_scale_check	Checking whether the number of shards is consistent in the newcc database	Database check	Daily Ops	TCENTER/ CRedis/oss
tx_Redis®_check_agent_status	Checking for any agents that have not been started	Service status check	Daily Ops	TCENTER/ CRedis/oss
tx_Redis®_check_cluster_fail_nodes_status	Checking whether the cluster contains failed nodes that have not been successfully forgotten	Service status check	Daily Ops	TCENTER/ CRedis/oss
tx_Redis®_interface_version_statics	[Collection] Checking the interface version information	Service status check	Daily Ops	TCENTER/ CRedis/ interface
tx_Redis®_machine_hwcheck_disk	Checking for disk exceptions logged in the past 30 minutes	Hardware status check	Daily Ops	TCENTER/ CRedis/ interface
tx_Redis®_machine_hwcheck_eth	Checking for NIC exceptions logged in the past 30 minutes	Hardware status check	Daily Ops	TCENTER/ CRedis/ interface
tx_Redis®_machine_hwcheck_mem	Checking for memory exceptions logged in the past 30 minutes	Hardware status check	Daily Ops	TCENTER/ CRedis/ interface
tx_Redis®_machine_hwcheck_nvme	Checking for NVMe exceptions logged in the past 30 minutes	Hardware status check	Daily Ops	TCENTER/ CRedis/ interface
tx_Redis®_machine_hwcheck_raid	Checking for RAID exceptions logged in the past 30 minutes	Hardware status check	Daily Ops	TCENTER/ CRedis/ interface
tx_Redis®_machine_survival_status	Checking the machine liveness status	Service status check	Daily Ops	TCENTER/ CRedis/oss
tx_Redis®_master_slave_consistency	Checking for the existence of dirty data in primary-secondary consistency	Service status check	Daily Ops	TCENTER/ CRedis/oss
tx_Redis®_server_status_cluster_fail_nodes	[Alarm] Checking whether there are failed nodes in the Redis® cluster	Service status check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_server_status_cluster_slots_ok	[Alarm] Checking whether the cluster slot status of Redis® is abnormal	Service status check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_server_status_cluster_state	[Alarm] Checking whether the cluster info status of Redis® is abnormal	Service status check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_server_status_memused_rate	[Alarm] Checking whether the Redis® memory utilization is greater than 80%	Service status check	Daily Ops	TCENTER/ CRedis/cache
tx_Redis®_server_status_stand_proc	[Alarm] Checking whether a single-point process exists in the Redis® cluster	Service status check	Daily Ops	TCENTER/ CRedis/cache

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_Redis®_server_version_statics	[Collection] Checking the cache process version of Redis®	Service status check	Daily Ops	TCENTER/ CRedis/cache
tx_tcenter_ckv_check_port	Checking whether the CKV port exists	Dependency service check	Daily Ops	TCENTER/ CRedis/cache
tx_tcenter_ckv_check_time	Checking whether the CKV time is normal	Dependency service check	Daily Ops	TCENTER/ CRedis/cache
tx_tcenter_csp_mgmt_pod_nginx	Checking whether the nginx process is running normally in the csp-pod-mgmt Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_mgmt_pod_pm2	Checking whether the PM2 process is running normally in the csp-pod-mgmt Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_mon_ceph_mgr	Checking whether the ceph-mgr process is running normally	Service status check	Daily Ops	TCENTER/CSP/ monitor_hosts
tx_tcenter_csp_mon_ceph_mon	Checking whether the ceph-mon process is running normally	Service status check	Daily Ops	TCENTER/CSP/ monitor_hosts
tx_tcenter_csp_mon_moira_server	Checking whether the moira server process is running normally	Service status check	Daily Ops	TCENTER/CSP/ monitor_hosts
tx_tcenter_csp_mon_moira_server_api	Checking whether the moira server api service is normal	Service status check	Daily Ops	TCENTER/CSP/ monitor_hosts
tx_tcenter_csp_mon_pod_ceph_mgr	Checking whether the ceph-mgr process is running normally in the csp-pod-mon Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_mon_pod_ceph_mon	Checking whether the ceph-mon process is running normally in the csp-pod-mon Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_mon_pod_moira_server	Checking whether the moira server process is running normally in the csp-pod-mon Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_mon_pod_moira_server_api	Checking whether the moira server API service is running normally in the csp-pod-mon Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_mon_pod_node_exporter	Checking whether the node_exporter process is running normally in the csp-pod-mon Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_mon_pod_prometheus	Checking whether the prometheus process is running normally in the csp-pod-mon Pod	Container check	Daily Ops	TCENTER/k8s-master-ip

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_tcenter_csp_rgw_moira_agent	Checking whether the moira agent is running normally	Service status check	Daily Ops	TCENTER/CSP/gw_hosts
tx_tcenter_csp_rgw_nginx	Checking whether csp_nginx is running normally	Service status check	Daily Ops	TCENTER/CSP/gw_hosts
tx_tcenter_csp_rgw_pod_moira_agent	Checking whether the moira agent is running normally in the csp-pod-rgw Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_rgw_pod_nginx	Checking whether the csp_nginx process is running normally in the csp-pod-rgw Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_rgw_pod_node_exporter	Checking whether the node_exporter process is running normally in the csp-pod-rgw Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_rgw_pod_rgw_api	Checking whether the RGW API service is running normally in the csp-pod-rgw Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_rgw_radosgw	Checking whether the RGW process is running normally	Service status check	Daily Ops	TCENTER/CSP/gw_hosts
tx_tcenter_csp_rgw_rgw_api	Checking whether the RGW API service is normal	Service status check	Daily Ops	TCENTER/CSP/gw_hosts
tx_tcenter_csp_store_ambari_agent	Checking whether the ambari-agent service is running normally	Service status check	Daily Ops	TCENTER/CSP/ips
tx_tcenter_csp_store_ambari_agent_crontab	Checking whether the ambari-agent scheduled task is normally configured in crontab	Service status check	Daily Ops	TCENTER/CSP/ips
tx_tcenter_csp_store_ceph_down	Checking whether any Ceph OSD is down	Service status check	Daily Ops	TCENTER/CSP/ips
tx_tcenter_csp_store_ceph_health	Checking the cluster health status	Service status check	Daily Ops	TCENTER/CSP/ips
tx_tcenter_csp_store_ceph_size	Checking the number of replicas of the CSP cluster	Service status check	Daily Ops	TCENTER/CSP/ips
tx_tcenter_csp_store_ceph_usage	Checking the utilization of the CSP cluster	Service status check	Daily Ops	TCENTER/CSP/ips
tx_tcenter_csp_store_crond	Checking whether crond is running normally	Service status check	Daily Ops	TCENTER/CSP/ips
tx_tcenter_csp_store_pod_ambari_agent	Checking whether the ambari-agent service is running normally in the csp-pod-osd Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_store_pod_ambari_agent_crontab	Checking whether the ambari-agent scheduled task is normally configured in crontab for the csp-pod-osd Pod	Container check	Daily Ops	TCENTER/k8s-master-ip

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_tcenter_csp_store_pod_ceph_down	Checking whether any Ceph OSD in the csp-pod-osd Pod is down	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_store_pod_ceph_health	Checking whether the cluster in the csp-pod-osd Pod is healthy	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_csp_store_pod_crond	Checking whether crond is running normally in the csp-pod-osd Pod	Container check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_es_allocation_oss	Checking whether the disk utilization of the Elasticsearch cluster does not exceed the threshold	Service status check	Daily Ops	TCENTER/ Elasticsearch/ oss
tx_tcenter_es_breaker_oss	Checking whether the parameter settings of the circuit breaker are less than 80%	Service status check	Daily Ops	TCENTER/ Elasticsearch/ oss
tx_tcenter_es_check_agent	Checking whether the agent process is running normally	Process check	Daily Ops and in-depth inspection	TCENTER/ Elasticsearch/ data
tx_tcenter_es_check_api	Checking whether the API can be accessed normally	API check	Daily Ops and in-depth inspection	TCENTER/ Elasticsearch/ oss
tx_tcenter_es_check_ces_node_info	Checking whether the.ces_node_info file has not been updated for 3 days	Service configuration check	Daily Ops	TCENTER/ Elasticsearch/ data
tx_tcenter_es_check_etcd_cluster_health	Checking whether the etcd cluster is healthy	Container check	Daily Ops and in-depth inspection	TCENTER/K8S/ Master
tx_tcenter_es_check_etcd_cluster_health	Checking whether the etcd cluster is healthy	Service status check	Daily Ops and in-depth inspection	TCENTER/ Elasticsearch/ etcd
tx_tcenter_es_check_master	Checking whether the master process is running normally	Process check	Daily Ops and in-depth inspection	TCENTER/ Elasticsearch/ oss
tx_tcenter_es_check_master_cron	Checking whether the master scheduled task exists	Service configuration check	Daily Ops	TCENTER/ Elasticsearch/ oss
tx_tcenter_es_check_node_status	Checking whether each node is running normally	Service status check	Daily Ops	TCENTER/ Elasticsearch/ etcd
tx_tcenter_es_cluster_each_shards_oss	Checking whether the disk utilization of the Elasticsearch cluster does not exceed the threshold	Service status check	Daily Ops and in-depth inspection	TCENTER/ Elasticsearch/ oss

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_tcenter_es_cluster_total_shards_oss	Checking whether the disk utilization of the Elasticsearch cluster does not exceed the threshold	Service status check	Daily Ops	TCENTER/ Elasticsearch/ oss
tx_tcenter_es_health_oss	Checking whether the health status of the Elasticsearch cluster is green	Service status check	Daily Ops	TCENTER/ Elasticsearch/ oss
tx_tcenter_es_jvm_mem_oss	Checking whether the JVM memory utilization of the Elasticsearch cluster does not exceed the threshold	Service status check	Daily Ops	TCENTER/ Elasticsearch/ oss
tx_tcenter_es_node_oss	Checking whether the health status of the Elasticsearch cluster is green	Service status check	Daily Ops	TCENTER/ Elasticsearch/ oss
tx_tcenter_es_shards_store_oss	Checking whether each shard of the supporting ES node is less than 50 GB	Service status check	Daily Ops	TCENTER/ Elasticsearch/ oss
tx_tcenter_etcd_dbsize	Checking whether dbSize is less than 1.6 GB	Service status check	Daily Ops	TCENTER/K8S/ Master TCENTER/K8S/ Etcd
tx_tcenter_etcd_member_status	Checking whether the member status is normal	Service status check	Daily Ops	TCENTER/K8S/ Etcd
tx_tcenter_hdfs_check_datanode_cluster_state	Checking whether the HDFS datanodes cluster is running normally	Service status check	Daily Ops and in-depth inspection	TCENTER/ HDFS/ datanode
tx_tcenter_hdfs_check_dfs_usage	Checking whether the DFS utilization of the HDFS cluster is less than 80%	Service status check	Daily Ops and in-depth inspection	TCENTER/ HDFS/ datanode
tx_tcenter_hdfs_check_kinit_state	Checking whether the kinit task for HDFS has been executed successfully	Service status check	Daily Ops	TCENTER/ HDFS/ datanode
tx_tcenter_hdfs_check_missing_blocks	Checking whether there are no missing blocks in the HDFS cluster	Service status check	Daily Ops and in-depth inspection	TCENTER/ HDFS/ datanode
tx_tcenter_hdfs_check_namenode_cluster_state	Checking whether the HDFS namenodes cluster is running normally	Service status check	Daily Ops and in-depth inspection	TCENTER/ HDFS/ datanode
tx_tcenter_hdfs_check_read_state	Checking whether the HDFS read operation is normal	Service status check	Daily Ops and in-depth inspection	TCENTER/ HDFS/ datanode
tx_tcenter_hdfs_check_safe_mode	Checking whether HDFS is in the security mode	System parameter check	Daily Ops and in-depth inspection	TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_tcenter_hdfs_check_write_state	Checking whether the HDFS write operation is normal	Service status check	Daily Ops and in-depth inspection	TCENTER/ HDFS/ datanode
tx_tcenter_hdfs_crontab_check_hdfs_journalnode	Checking the hdfs-jn scheduled task	Service configuration check	Daily Ops	TCENTER/ HDFS/ journalnode
tx_tcenter_hdfs_crontab_check_kinit	Checking whether HDFS is configured with a kinit scheduled task	Service status check	Daily Ops	TCENTER/ HDFS/ journalnode
tx_tcenter_hdfs_kerberos	Checking the processes of Kerberos	Service status check	Daily Ops	TCENTER/ HDFS/ namenode
tx_tcenter_hdfs_keytable	Checking the keytabe	Service status check	Daily Ops	TCENTER/ HDFS/ namenode
tx_tcenter_hdfs_keytable_check_node	Checking the keytab files of all nodes	Service status check	Daily Ops	TCENTER/ HDFS/ namenode TCENTER/ HDFS/ journalnode TCENTER/ HDFS/ datanode
tx_tcenter_hdfs_process_check_journalnode	Checking whether the hdfs-jn process is running normally	Process check	Daily Ops	TCENTER/ HDFS/ journalnode
tx_tcenter_k8s_apiserver-etcd-client_expire	Checking whether the apiserver-etcd-client certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/ Master
tx_tcenter_k8s_apiserver-kubelet-client_expire	Checking whether the apiserver-kubelet-client certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/ Master
tx_tcenter_k8s_apiserver-loopback-client_expire	Checking whether the apiserver-loopback-client certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/ Master
tx_tcenter_k8s_apisever_expire	Checking whether the apiserver certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/ Master
tx_tcenter_k8s_ca_expire	Checking whether the ca certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/ Master
tx_tcenter_k8s_cert-manager_expire	Checking whether the cert-manager certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_k8s_etcd-ca_expire	Checking whether the etcd-ca certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/ Etcd

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_tcenter_k8s_etcd-peer_expire	Checking whether the etcd-peer certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/Etcd
tx_tcenter_k8s_etcd-server_expire	Checking whether the etcd-server certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/Etcd
tx_tcenter_k8s_kubelet-ca_expire	Checking whether the kubelet-ca certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/Master
tx_tcenter_k8s_kubelet-client_expire	Checking whether the kubelet-client certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcenter_k8s_kubelet_expire	Checking whether the kubelet certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcenter_k8s_mtu_check	Checking whether there is no risk for MTU	Service configuration check	Daily Ops and in-depth inspection	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcenter_k8s_ocloud-tcenter-base-vpc-dns_analysis	Checking whether ocloud-tcenter-base-vpc-dns can resolve domain names	Service configuration check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_k8s_registry-ca_expire	Checking whether the registry-ca certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/Master
tx_tcenter_k8s_registry_expire	Checking whether the registry certificate will expire in 45 days	Certificate and license check	Daily Ops	TCENTER/K8S/Master
tx_tcenter_kafka_check_data_size	Checking Kafka	Dirty data check	Daily Ops	TCENTER/Kafka/kafka
tx_tcenter_kafka_check_port	Checking Kafka	Port check	Daily Ops	TCENTER/Kafka/kafka
tx_tcenter_kafka_check_topic	Checking Kafka	Dirty data check	Daily Ops	TCENTER/Kafka/kafka
tx_tcenter_kafka_partition_leader	Checking whether the partition leader of Kafka is not -1	Service status check	Daily Ops	TCENTER/Kafka/kafka
tx_tcenter_kubelet_check_docker	Checking the docker process status	Process check	Daily Ops	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcenter_kubelet_check_kubelet_status	Checking the kubelet process status	Process check	Daily Ops	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcenter_kubelet_tcs_cgroup_memory_nokmem	Checking whether the cgroup.memory=nokmem configuration is completed on all TCS nodes	Service configuration check	Daily Ops	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcenter_mongo_cluster_config_state	Checking the MongoDB cluster	Dirty data check	Daily Ops	TCENTER/MongoDB/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				instance
tx_tcenter_mongo_cluster_read_state	Checking the MongoDB cluster	Dirty data check	Daily Ops	TCENTER/ MongoDB/ instance
tx_tcenter_mongo_cluster_shard_state	Checking the MongoDB cluster	Dirty data check	Daily Ops	TCENTER/ MongoDB/ instance
tx_tcenter_mongo_cluster_write_state	Checking the MongoDB cluster	Dirty data check	Daily Ops	TCENTER/ MongoDB/ instance
tx_tcenter_mq_check_cluster_state	Checking whether the cluster status is normal	Service status check	Daily Ops	TCENTER/ RabbitMQ/ RabbitMQ
tx_tcenter_mq_check_cron	Checking the scheduled tasks of MQ	Service status check	Daily Ops	TCENTER/ RabbitMQ/ RabbitMQ
tx_tcenter_mq_check_cron	Checking the scheduled tasks of MQ	Service configuration check	Daily Ops	TCENTER/ Product/ product-tcenter-support-mq/ mq_nodes
tx_tcenter_mq_check_list_queues_state	Checking whether the status of mq list_queues is normal	Service status check	Daily Ops	TCENTER/ RabbitMQ/ RabbitMQ
tx_tcenter_mq_cluster_brainsplit	Checking whether a split-brain error does not occur in the MQ cluster	Service status check	Daily Ops	TCENTER/ RabbitMQ/ RabbitMQ
tx_tcenter_mq_report_state	Checking the MQ node	Service status check	Daily Ops	TCENTER/ RabbitMQ/ RabbitMQ
tx_tcenter_mq_report_state	Checking the MQ node	Dirty data check	Daily Ops	TCENTER/ Product/ product-tcenter-support-mq/ mq_nodes
tx_tcenter_mq_state	Checking the MQ node	Service status check	Daily Ops and in-depth inspection	TCENTER/ RabbitMQ/ RabbitMQ
tx_tcenter_mq_state	Checking the MQ node	Dirty data check	Daily Ops and in-depth inspection	TCENTER/ Product/ product-tcenter-support-mq/ mq_nodes
tx_tcenter_mq_version_check_3100	Checking the physical supporting RabbitMQ version of the cloud platform	Service configuration check	Daily Ops	TCENTER/ RabbitMQ/ RabbitMQ
tx_tcenter_ntp_check_crontab_ntpdate	Checking whether the ntpdate scheduled task is configured in crontab	Time service check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/journalnode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/oss TCENTER/CRedis/interface TCENTER/CRedis/cache

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_tcenter_ntp_check_ntpd_is_enabled	Checking whether the ntpd startup script is normal	Time service check	Daily Ops	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/journalnode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/oss TCENTER/CRedis/interface

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				TCENTER/ CRedis/cache
tx_tcenter_ntp_check_ntpd_offset	Checking whether the last synchronization offset of ntpd is within 100 ms	Time service check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ journalnode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/oss

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
tx_tcenter_ntp_check_ntpd_when	Checking whether the last synchronization time of ntpd is within 1024s	Time service check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ journalnode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				gw_hosts TCENTER/ CRedis/oss TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
tx_tcenter_ntp_check_process_ntpd_and_chrony	Checking whether ntpd or chrony is used	Time service check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ journalnode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/oss TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_tcenter_pod_workload_node_check	Checking whether workloads are distributed across the available nodes in a Kubernetes cluster without overlapping	Container check	Daily Ops	TCENTER/K8S/Master
tx_tcenter_system_check_defunct_process	Checking whether the number of zombie processes on the server is less than 50	System parameter check	Daily Ops	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
tx_tcenter_system_cpu_idle	Checking whether the CPU idle rate of the server is greater than 30%	System parameter check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
tx_tcenter_system_disk_read_write	Checking the disk read-write availability of the server	System parameter check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_tcenter_system_fd_usage	Checking whether the fd usage of the server is less than 80%	System parameter check	Daily Ops	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
tx_tcenter_system_inode_usage	Checking whether the inode utilization of the server is less than 80%	System parameter check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
tx_tcenter_system_netdev_recv_pkg	Checking whether the number of NIC inbound packets of the server is less than 5 GB	System parameter check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				CRedis/ interface TCENTER/ CRedis/cache
tx_tcenter_system_netdev_send_pkg	Checking whether the number of NIC outbound packets of the server is less than 5 GB	System parameter check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				TCENTER/ CRedis/cache
tx_tcenter_system_network_sockets	Checking whether the number of network socket connections of the server is less than 200,000	System parameter check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_tcenter_system_tcenter_io_await	Checking whether the I/O wait time of the server is less than 1second	System parameter check	Daily Ops	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product-tcenter-support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
tx_tcenter_system_tcenter_io_util	Checking whether the I/O utilization of the server is	System parameter	Daily Ops	TCENTER/ TDSQL/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
	less than 60%	check		scheduler TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Master TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/CSP/ monitor_hosts TCENTER/CSP/ gw_hosts
tx_tcenter_tcs_cgroup_memory_nokmem_status	Checking whether the cgroup.memory=nokmem configuration is effective on all TCS nodes	Service configuration check	Daily Ops	TCENTER/K8S/ Master
tx_tcenter_tcs_check_node_mem_cpu	Checking whether the CPU:memory ratio of the node is greater than 90%	Container check	Daily Ops	TCENTER/K8S/ Master
tx_tcenter_tcs_check_node_mem_cpu	Checking whether the CPU:memory ratio of the node is greater than 90%	Service status check	Daily Ops	TCENTER/k8s- master-ip
tx_tcenter_tcs_check_node_status	Checking whether the node is ready	Container check	Daily Ops and in-depth inspection	TCENTER/K8S/ Master
tx_tcenter_tcs_check_node_status	Checking whether the node is ready	Service status check	Daily Ops and in-depth inspection	TCENTER/k8s- master-ip
tx_tcenter_tcs_master_disk_usage	Checking whether the disk utilization is less than 60%	Process check	Daily Ops	TCENTER/K8S/ Master
tx_tcenter_tdsql_check_proc	Checking the configurations of EWP processes and scheduled tasks	Process check	Daily Ops	TCENTER/ TDSQL/oss TCENTER/ TDSQL/ Instance
tx_tcenter_tdsql_ioutil	Checking whether the disk utilization is greater than 60%	Capacity check	Daily Ops	TCENTER/ TDSQL/oss TCENTER/ TDSQL/

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
				Instance
tx_tcenter_tdsql_load_average	Checking whether the average system load for the last 1 minute/5 minutes/15 minutes is less than the number of CPU cores	Capacity check	Daily Ops	TCENTER/ TDSQL/oss TCENTER/ TDSQL/ Instance
tx_tcenter_tdsql_login	Checking whether the user's TDSQL exists and whether the TDSQL can be logged in to	System parameter check	Daily Ops	TCENTER/ TDSQL/ Instance
tx_tcenter_tdsql_mem_swap_usage	Checking whether the swap usage is greater than 60%	Capacity check	Daily Ops	TCENTER/ TDSQL/oss TCENTER/ TDSQL/ Instance
tx_tcenter_tdsql_mem_usage	Checking whether the memory utilization exceeds the threshold	Capacity check	Daily Ops	TCENTER/ TDSQL/oss TCENTER/ TDSQL/ Instance
tx_tcenter_tdsql_partition_usage	Checking whether the disk utilization of the mounted / and /data partitions is less than the threshold	Capacity check	Daily Ops	TCENTER/ TDSQL/oss TCENTER/ TDSQL/ Instance
tx_tcenter_tdsql_rc_local	Checking whether rc.local has execution permissions	System parameter check	Daily Ops	TCENTER/ TDSQL/ Instance
tx_tcenter_tke_check_cls_fail_by_k8s_api	Checking whether there are TKE clusters that failed to be created or are stuck in the creation process for a long period of time	Service status check	Daily Ops	TCENTER/k8s-master-ip
tx_tcenter_vpcdns_dig_pod	Checking the domain name resolution of base-vpc-dns	Service status check	Daily Ops	TCENTER/ KubeResource/ ocloud-tcenter-base-vpc-dns
tx_tcenter_zk_check_conn_num	Checking whether the number of connections to ZooKeeper is less than 500	Dependency service check	Daily Ops	TCENTER/ Zookeeper/ instance
tx_tcenter_zk_check_conn_num	Checking whether the number of connections to ZooKeeper is less than 500	Service status check	Daily Ops	TCENTER/ Zookeeper/ zookeeper
tx_tcenter_zk_check_status	Checking the ZooKeeper status	Dependency service check	Daily Ops and in-depth inspection	TCENTER/ Zookeeper/ instance
tx_tcenter_zk_check_zk_outstanding_requests	Checking whether the number of queued requests in the ZooKeeper cluster exceeds 10	Dependency service check	Daily Ops and in-depth inspection	TCENTER/ Zookeeper/ instance

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_tcenter_zk_check_zk_outstanding_requests	Checking whether the number of queued requests in the ZooKeeper cluster exceeds 10	Service status check	Daily Ops and in-depth inspection	TCENTER/ Zookeeper/ zookeeper
tx_tcenter_zk_check_znode_num	Checking whether the number of znodes exceeds 100,000	Dependency service check	Daily Ops and in-depth inspection	TCENTER/ Zookeeper/ instance
tx_tcenter_zk_check_znode_num	Checking whether the number of znodes exceeds 100,000	Service status check	Daily Ops and in-depth inspection	TCENTER/ Zookeeper/ zookeeper
tx_tcenter_zk_config	Checking the node configuration of the ZooKeeper cluster	Dependency service check	Daily Ops	TCENTER/ Zookeeper/ instance
tx_tcenter_zk_multi_version	Checking whether multiple versions of ZooKeeper are deployed in a single cluster	Dependency service check	Daily Ops	TCENTER/ Zookeeper/ instance
tx_tcenter_zk_no_sync_data	Checking whether the number of unsynchronized data items in ZooKeeper is greater than 10	Dependency service check	Daily Ops	TCENTER/ Zookeeper/ instance
tx_tcenter_zk_no_sync_data	Checking whether the number of unsynchronized data items in ZooKeeper is greater than 10	Service status check	Daily Ops	TCENTER/ Zookeeper/ zookeeper
tx_tcenter_zk_partition_usage	Checking whether the disk utilization of the ZooKeeper cluster for both / and /data is below the threshold	Dependency service check	Daily Ops	TCENTER/ Zookeeper/ instance
tx_tcenter_zk_snapshot	Checking the snapshot size of the ZooKeeper cluster	Dependency service check	Daily Ops	TCENTER/ Zookeeper/ instance
tx_tcenter_zk_snapshot_size	Checking whether the size of ZooKeeper snapshots exceeds 1 GB	Dependency service check	Daily Ops and in-depth inspection	TCENTER/ Zookeeper/ instance
tx_tcenter_zk_snapshot_size	Checking whether the size of ZooKeeper snapshots exceeds 1 GB	Service status check	Daily Ops and in-depth inspection	TCENTER/ Zookeeper/ zookeeper
tx_tcenter_zk_state	Checking the node status of the ZooKeeper cluster	Dependency service check	Daily Ops and in-depth inspection	TCENTER/ Zookeeper/ instance
tx_tcs_check_pajero	Checking the running status of pajero	Service configuration check	Daily Ops and in-depth inspection	TCENTER/K8S/ Master
tx_tcs_check_pvc_usage	Checking whether the PVC utilization is greater than 85%	Service configuration check	Daily Ops	TCENTER/K8S/ Node TCENTER/K8S/ Master

Inspection Item ID	Inspection Item Description	Inspection Item Group	Category	Business Tree Path
tx_tcs_check_tunl_vip	Checking the integrity of the TCS tunnel	Service configuration check	Daily Ops and in-depth inspection	TCENTER/K8S/Node
tx_tcs_dig_underlay	Checking the resolution of TCS underlay domain names	Process check	Daily Ops	TCENTER/K8S/Master

Inspection Handling
Inspection Item 1: Basic Platform Feature Inspection

Scenarios

Most components of TCenter run in containers, and most of these containers have their own health detection mechanisms. Therefore, it is important to first confirm whether the containers are running properly.

Prerequisites

The login addresses and account passwords of the tenant console and operation console have been obtained.

Operation Steps

1. Log in to the tenant console and confirm that there are no errors on the main page.
2. Select Personal Information > Account Center/Cloud Access Management/Security Settings, and enter the tenant-side page of the basic platform. In the left sidebar, click each submenu item sequentially to check if the pages display correctly.
3. Log in to the operation console and confirm that there are no errors on the main page.
4. Choose Platform Management > User and Permission/Cloud API Management/CAM Management, and enter the operation-side page of the basic platform. In the left sidebar, click each submenu item sequentially to check if the pages display correctly.
   If the page is displayed abnormally or errors occur on the page, see Exception Handling for handling.

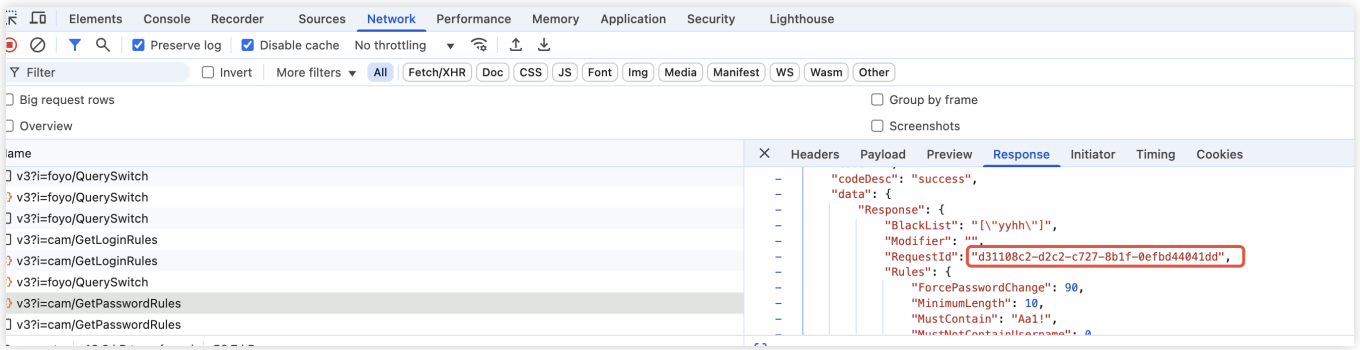
Exception Handling

1. Check the pod status of the service.

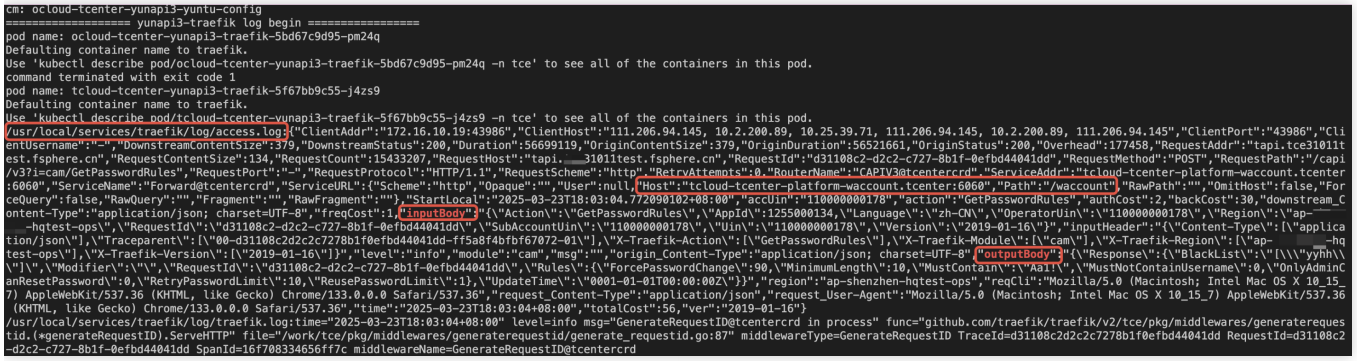
```
# Check if the pod is started normally and if the pod status is normal.
kubectl get pod -n tce | grep cloud-tcenter
# Since all TCenter services are stateless, if there are any pods in an abnormal status, you can quickly recover them using the delete command. Here, ocloud-tcenter-identityaccess is used as an example.
kubectl delete pod -n tce ocloud-tcenter-identityaccess-xxx
```

1. View the service logs.

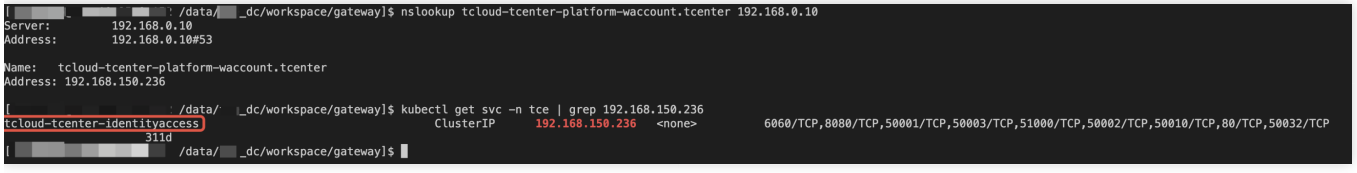
Since many pages are involved, and almost all of them are served by containerized components that forward requests via cloud API, a common troubleshooting method is provided:



Obtain the cloud API log search script.
mkdir -p /data/tce_dc/workspace/gateway/
cd /data/tce_dc/workspace/gateway/
kubectl -ntce cp `kubectl get pod -n tce | grep ocloud-tcenter-yunapi3-yuntu | awk '{print $1}' | tail -n 1`:/usr/local/services/yuntu/bin/traefik_tool.sh /data/tce_dc/workspace/gateway/traefik_tool.sh
chmod +x traefik_tool.sh
./traefik_tool.sh log ${reqid}

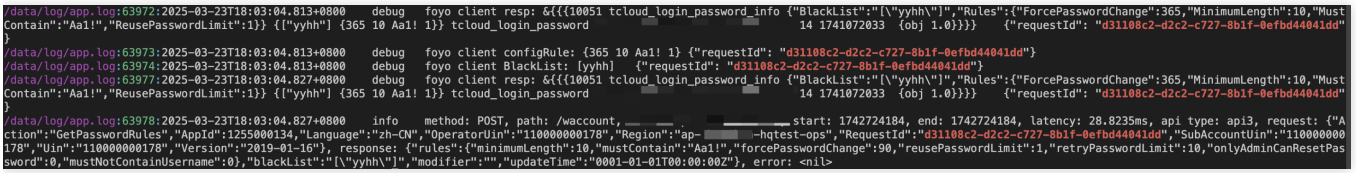


Pay special attention to the red boxes in the screenshot, including:
(1) Backend request address (tcloud-tcenter-platform-waccount.tcenter:6060/waccount).
(2) Input parameters ({"Action":"GetPasswordRules","AppId":1255000134,"Language":"zh-CN","OperatorUin":"110000000178","Region":"ap-hqtest-ops","RequestId":"d31108c2-d2c2-c727-8b1f-0efbd44041dd","SubAccountUin":"110000000178","Uin":"110000000178","Version":"2019-01-16"}).
For further log analysis, you need to enter the backend workload container for inspection, for example:



nslookup tcloud-tcenter-platform-waccount.tcenter 192.168.0.10

kubectl get svc -n tce | grep 192.168.150.236
kubectl get pod -n tce | grep tcloud-tcenter-identityaccess
kubectl exec -it -n tce tcloud-tcenter-identityaccess-xxxx bash
grep -rn ${reqid} /data/log/



Inspection Item 2: Component Status Inspection

Scenarios

Most components of TCenter run in containers, and most of these containers have their own health detection mechanisms. Therefore, it is important to first confirm whether the containers are running properly.

Prerequisites

You can log in to the Kubernetes cluster nodes and execute kubectl commands.

Operation Steps

1. Log in to the Kubernetes cluster node and run the following command to check if the service status is normal.

```
# Check if the pod is started normally and if the pod status is normal.
kubectl get pod -n tce | grep cloud-tcenter
```

Exception Handling

If the container's runtime status is abnormal, you can confirm the reason for the abnormality using the following method.

```
# Use tcloud-tcenter-console-web-xx as an example.
kubectl describe pod -n tce tcloud-tcenter-console-web-xxx | grep -A 5  -E 'Last State:|Events:'
```

```
    Last State:     Terminated
      Reason:       Completed
      Exit Code:    0
      Started:      Sun, 23 Mar 2025 18:24:17 +0800
      Finished:     Sun, 23 Mar 2025 18:25:47 +0800
    Ready:          False
--
Events:
  Type     Reason     Age                  From     Message
  ----     ------     ----                 ----     -------
  Warning  Unhealthy  13m (x6959 over 4d)  kubelet  (combined from similar events): Liveness probe failed: [2025-03-23T18:16:47] cwp_cloudapi healthchk exit 1
```

Focus on the Last State.Reason and Exit Code. If the container is abnormal due to resource limitations, such as an out-of-memory (OOM) issue, oomkilled is displayed, and the exit code is 137. If the status in the figure is not 137, it is usually caused by the failure of the service health detection mechanism in the cloud platform scenario. Based on the events section, it can be confirmed that the root cause is a liveness detection failure.

```
# Check container resource limits and health check settings.
kubectl describe pod -n tce tcloud-tcenter-console-web-xxx | grep -A 4 -E 'Limits|Liveness|Readiness'
```

```
    Limits:
      cpu:     4
      memory:  16G
    Requests:
      cpu:        1
--
    Liveness:     exec [/bin/bash /■ /healthchk.sh] delay=30s timeout=1s period=30s #success=1 #failure=3
    Readiness:    exec [/bin/bash /■ /healthchk.sh] delay=30s timeout=1s period=30s #success=1 #failure=3
```

(1) If the container is affected by resource limitations, you can temporarily solve the issue by adjusting the component's specifications. For a long-term solution, you can submit a ticket to the relevant technology stack to check whether the resource usage is unreasonable or if the default settings are unreasonable. The typical processing rocedures are as follows:

```
# Modify the limits part to adjust the corresponding rule information (using tcloud-tcenter-console-we
b-xx as an example).
# For regular Kubernetes components, modify the deploy, sts, ds, and other custom resources by defa
ult. In the cloud platform scenario, applications use the open application model (OAM), and the upper-
level controllers for regular services are all comp.
kubectl edit comp -n tce tcloud-tcenter-console-web
# Confirm that the container starts normally.
kubectl get pod -n tce | grep tcloud-tcenter-console-web
# Confirm the historical resource usage of components. Log in to the operation side, choose Monitorin
g System (CWhistle) > Cloud Product Metric > TCS/Pod, and select pod_cpu_limit_usage/pod_memory
_nocache_limit_usage for Monitoring Metric, global for Cluster, tce for Namespace, and tcloud-tcenter-
console-web for Container to view the information.
```

(2) If the container cannot start due to a health check issue, you need to view the service logs to confirm the reason. A typical approach is as follows (After preliminary troubleshooting, you can submit a ticket to the relevant technology stack for further support):

kubectl get pod -n tce | grep tcloud-tcenter-console-web
Enter the container using the following command when the pod is in a 0/1 Running status.
kubectl exec -it -n tce tcloud-tcenter-console-web-xxx bash
tail -n 100 /data/log/console/console.server.log
If the container is always in the CrashLoopBackOff status, try running the kubectl logs command to v iew the logs. (The service should support printing the logs to stdout.)
kubectl logs -n tce tcloud-tcenter-console-web-xxx

Daily Monitoring
Common Monitoring Metrics

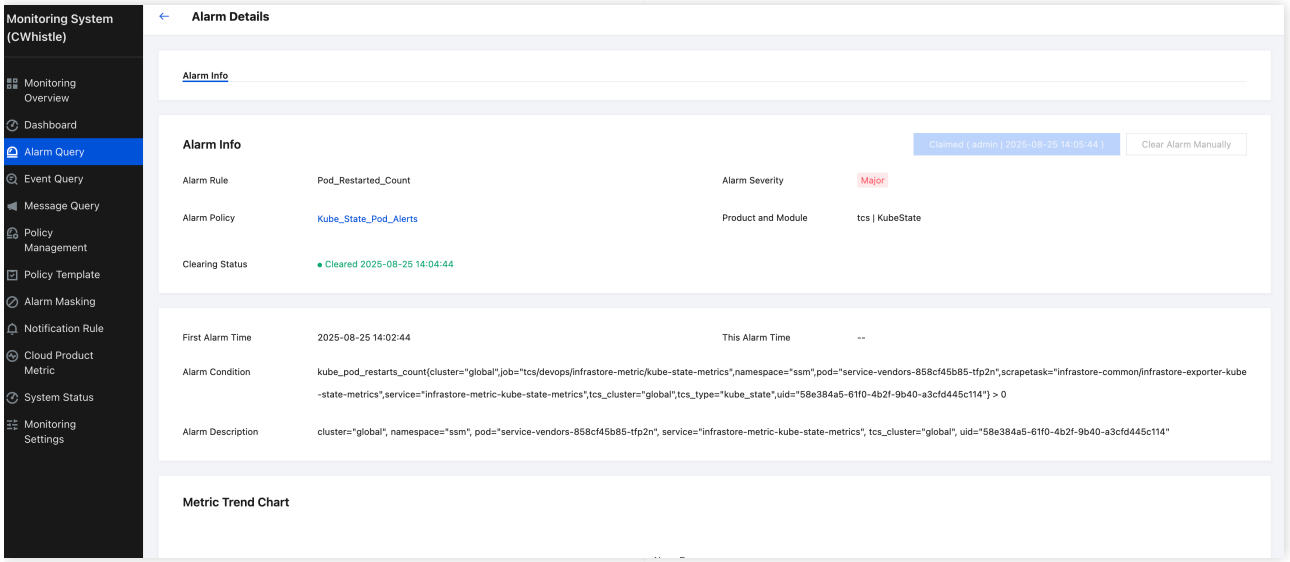
Metric/Event Name	Category	Unit	Metric Type	Threshold	Contingency Plan
Pod restart	Container Monitoring		The container has restarted.		-
Pod CPU utilization (relative to the limit) exceeding the threshold	Container Monitoring	%	The CPU utilization of the pod is high and exceeds the configured threshold of the limit.	95%	-
Pod actual memory utilization (relative to the limit) exceeding the threshold	Container Monitoring	%	The actual memory utilization of the pod is high and exceeds the configured threshold of the limit.	95%	-

CAM Monitoring Metrics

Metric Name	Type	Tag	Purpose
http_requests_total	Counter	action	Total HTTP requests categorized by API action
http_requests_latency	Histogram	action	HTTP request latency distribution (Buckets: 1–4096 ms exponential distribution)
http_requests_latency_range_total	Counter	action, section	Number of HTTP requests by latency range (0–300 ms/300–600 ms/600 ms–1s/1s–2s/2s–3s/3s+)
http_error_count	Counter	action	Number of HTTP error requests
cam_auth_requests_total	Counter	mode, returncode	Total CAM authentication requests categorized by authentication mode and return code

Metric Name	Type	Tag	Purpose
cam_auth_requests_latency	Histogram	mode, returncode	CAM authentication request latency distribution (Buckets: 1/10/100/500/1000/3000 ms)
cam_auth_requests_latency_section	Counter	uin, owneruin, section	Number of CAM authentication requests by latency range; the specific uin/owneruin is recorded when the request latency exceeds 1s
cam_db_slow_queries_total	Counter	action, duration_range, sql	Database slow query statistics categorized by operation duration range and SQL

Alarm Handling
Contingency Plan for Pod Restart



Alarm Description

Pod restart:

- Containers related to basic platform workload (ocloud-osp-passwd-svr): Pod restarts due to certain abnormalities.
- Containers related to audit log workload (t|ocloud-cloudaudit): Pod restarts due to certain abnormalities.
- Containers related to CAM workload (t|ocloud-tcenter-cam/tcloud-tcenter-support-cam): Pod restarts due to certain abnormalities.
- Containers related to tag workload (tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag): Pod restarts due to certain abnormalities.
- Containers related to cloud API management workload (cloud-tcenter-yunapi3): Pod restarts due to certain abnormalities.

Alarm Attributes

Module	Name	Event Level	Manual/Automatic Clearing
Default alarm policy for tcs-kube_state metric monitoring	Pod restart	Level 2 alarm	Automatic clearing

Impact on the System

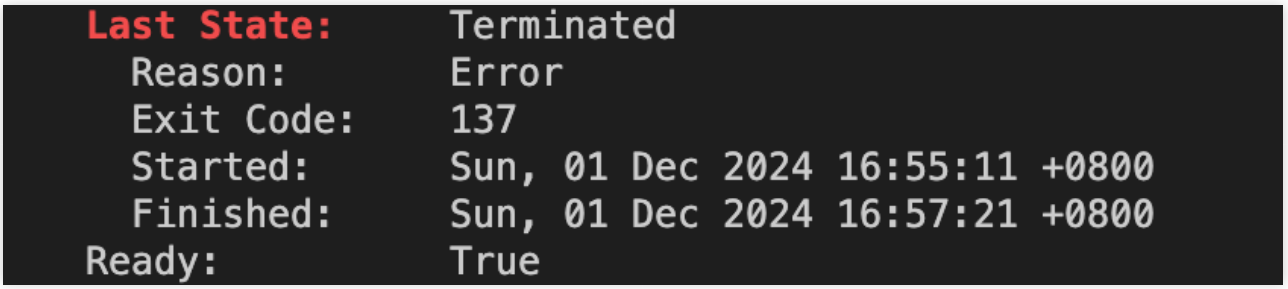
No impact.

Possible Causes

No.	Cause	Description
1	Resource Limitations	- Memory limit: This is usually accompanied by the "Pod actual memory utilization (relative to the limit) exceeds the threshold" alarm. If the memory used by the container exceeds the defined limit, Kubernetes will kill the container, resulting in a restart. In such cases, the cause of insufficient memory needs to be confirmed with technical support. Possible causes include: - The memory limit assigned to the pod is too small and needs to be increased. - The application may have a memory leak, causing usage to increase gradually and eventually exceed the limit. The program's memory reclamation mechanism needs optimization. - CPU limit: This is usually accompanied by the "Pod CPU utilization (relative to the limit) exceeds the threshold" alarm. Although the CPU limit does not directly cause container restarts, if the container fails to function properly due to insufficient CPU resources, it may crash. In such cases, the cause of insufficient CPU resources needs to be confirmed with technical support. Possible causes include: - The CPU resource limit assigned to the pod is too small and needs to be increased. - The application may have abnormalities, causing its CPU usage to increase gradually and eventually exceed the limit. The program logic needs optimization.
2	Health check failure	Liveness probe: If a liveness probe is configured and detects that the container is unhealthy, Kubernetes will restart the container. For container health issues, please contact technical support to confirm whether the workload service is functioning normally and to identify the root cause.
3	Node issue	If a node fault occurs, all services running on that node will be abnormal. This type of issue is handled at the TCS layer and does not require attention from the business side.

Processing Procedures

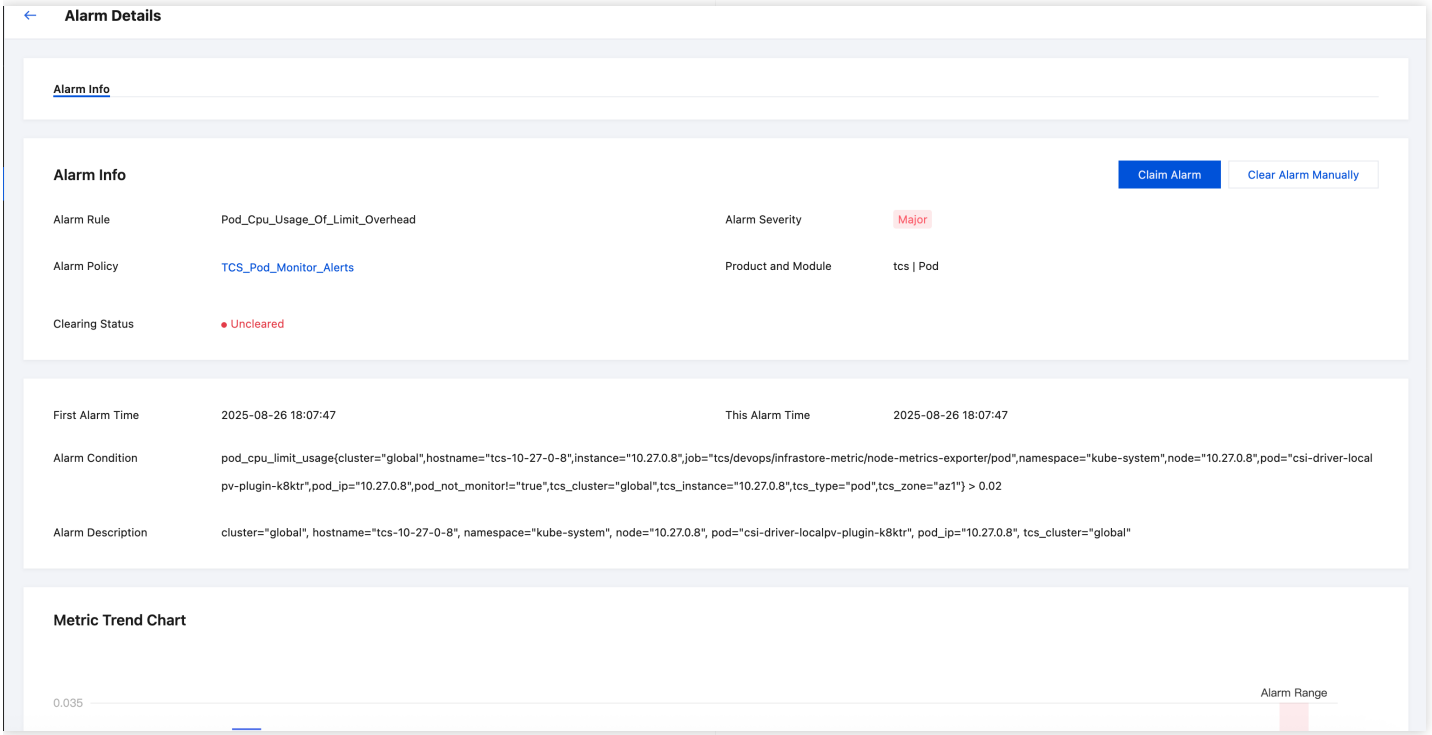
1. Query the status information of the pod when its most recent run ended:
   - Basic platform workload: kubectl describe pod -n tce ocloud-osp-passwd-svr-xxx | grep -A 5 'Last State:'
   - Audit log workload: *kubectl describe pod -n tce t|ocloud-cloudaudit-xxx | grep -A 5 'Last State:'*
   - CAM workload: kubectl describe pod -n tce t|ocloud-tcenter-cam/tcloud-tcenter-support-cam-xxx | grep -A 5 'Last State:'
   - Tag workload: kubectl describe pod -n tce tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag-xxx | grep -A 5 'Last State:'
   - Cloud API management workload: kubectl describe pod -n tce cloud-tcenter-yunapi3-xxx | grep -A 5 'Last State:'



> Note:
>
> The main focus here is on the reason and exit code sections:
>
> - exitCode: The exit code of the container.
> - reason: the reason for termination, such as Error (an error occurred) or OOMKilled (terminated due to out-of-memory issues).

2. Check the event status of the pod:
   - Basic platform workload: kubectl describe pod -n tce ocloud-osp-passwd-svr-xxx | grep -A 10 'Events:'
   - Audit log workload: kubectl describe pod -n tce t|ocloud-cloudaudit-xxx | grep -A 10 'Events:'
   - CAM workload: kubectl describe pod -n tce t|ocloud-tcenter-cam/tcloud-tcenter-support-cam-xxx | grep -A 10 'Events:'
   - Tag workload: kubectl describe pod -n tce tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag-xxx | grep -A 10 'Events:'
   - Cloud API management workload: kubectl describe pod -n tce cloud-tcenter-yunapi3-xxx | grep -A 10 'Events:'
3. Collect log information from the fault occurrence time and contact technical support for assistance. Log path: /data/log.

Contingency Plan for Pod CPU Utilization (Relative to the Limit) Exceeding the Threshold



Alarm Description

Pod CPU utilization (relative to the limit) exceeds the threshold:

- Containers related to basic platform workload (cloud-tcenter): Due to certain abnormalities, CPU utilization becomes high, exceeding 95% of the configured limit.
- Containers related to audit log workload (t|ocloud-cloudaudit): Due to certain abnormalities, CPU utilization becomes high, exceeding 95% of the configured limit.
- Containers related to CAM workload (t|ocloud-tcenter-cam/tcloud-tcenter-support-cam): Due to certain abnormalities, CPU utilization becomes high, exceeding 95% of the configured limit.
- Containers related to tag workload (tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag): Due to certain abnormalities, CPU utilization becomes high, exceeding 95% of the configured limit.
- Containers related to cloud API management workload (cloud-tcenter-yunapi3): Due to certain abnormalities, CPU utilization becomes high, exceeding 95% of the configured limit.

Alarm Attributes

Module	Name	Event Level	Manual/ Automatic Clearing
Default alarm policy for tcs-pod metric monitoring	Pod CPU utilization (relative to the limit) exceeding the threshold	Level 2 alarm	Automatic clearing

Impact on the System

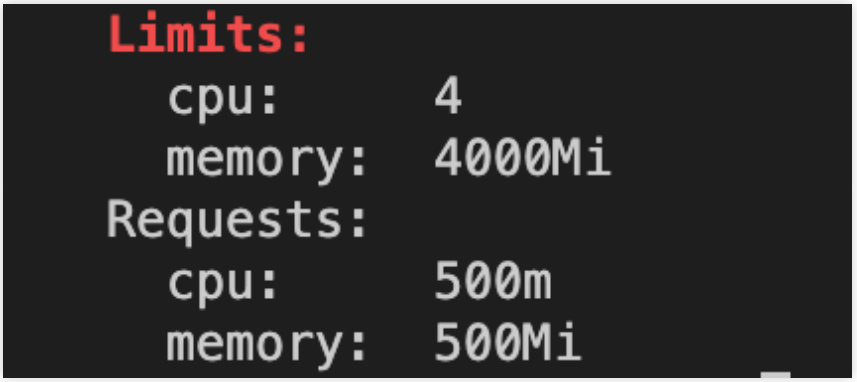
No impact.

Possible Causes

- The CPU resource limit assigned to the pod is too small and needs to be increased.
- The application may have abnormalities, causing its CPU usage to increase gradually and eventually exceed the limit. The program logic needs optimization.

Processing Procedures

1. Check pod resource limits:
    - Basic platform workload***: kubectl describe pod -n tce cloud-tcenter-xxx | grep -A 5 'Limits:'***
    - Audit log workload: *kubectl describe pod -n tce t|ocloud-cloudaudit-xxx | grep -A 5 'Limits:'*
    - CAM workload: *kubectl describe pod -n tce t|ocloud-tcenter-cam/tcloud-tcenter-support-cam-xxx | grep -A 5 'Limits:'*
    - Tag workload: *kubectl describe pod -n tce tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag-xxx | grep -A 5 'Limits:'*
    - Cloud API management workload: *kubectl describe pod -n tce cloud-tcenter-yunapi3-xxx | grep -A 5 'Limits:'*

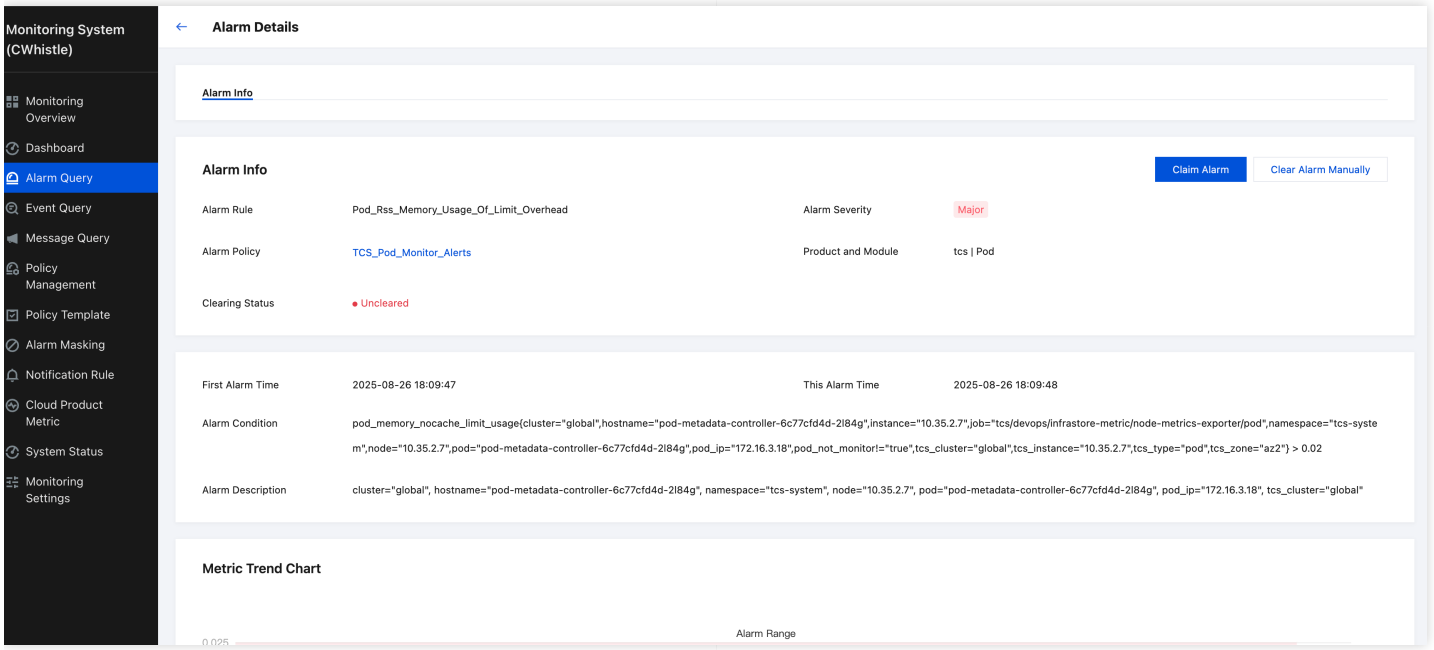


2. View historical monitoring information of pod:
   Log in to the operation side, choose Monitoring System (CWhistle) > Cloud Product Metric > TCS/Pod, query the CPU utilization monitoring data of the Pod that triggered the alarm around the time the alarm was triggered or for a longer period, and determine whether the behavior is normal.

Time Range	The day of the fault
Monitoring Metric	pod_cpu_limit_usage
Cluster	global
Namespace	tce
Pod	◦ Basic platform workload: cloud-tcenter-xxx ◦ Audit log workload: Pod: t|ocloud-cloudaudit-xxx ◦ CAM workload: Pod: t|ocloud-tcenter-cam/tcloud-tcenter-support-cam-xxx ◦ Tag workload: Pod: tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag-xxx ◦ Cloud API management workload: Pod: cloud-tcenter-yunapi3-xxx

3. Collect relevant information and contact the technical team for support.

Contingency Plan for Pod Actual Memory Utilization (Relative to the Limit) Exceeding the Threshold



Alarm Description

Pod actual memory utilization (relative to the limit) exceeds the threshold:

- Containers related to basic platform workload (cloud-tcenter): Due to certain abnormalities, memory utilization becomes high, exceeding 95% of the configured limit.
- Containers related to audit log workload (t|ocloud-cloudaudit): Due to certain abnormalities, memory utilization becomes high, exceeding 95% of the configured limit.
- Containers related to CAM workload (t/ocloud-tcenter-cam/tcloud-tcenter-support-cam): Due to certain abnormalities, memory utilization becomes high, exceeding 95% of the configured limit.
- Containers related to tag workload (tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag): Due to certain abnormalities, memory utilization becomes high, exceeding 95% of the configured limit.
- Containers related to cloud API management workload (cloud-tcenter-yunapi3): Due to certain abnormalities, memory utilization becomes high, exceeding 95% of the configured limit.

Alarm Attributes

Module	Event Name	Event Level	Manual/ Automatic Clearing
Default alarm policy for tcs-pod metric monitoring	Pod actual memory utilization (relative to the limit) exceeding the threshold	Level 2 alarm	Automatic clearing

Impact on the System

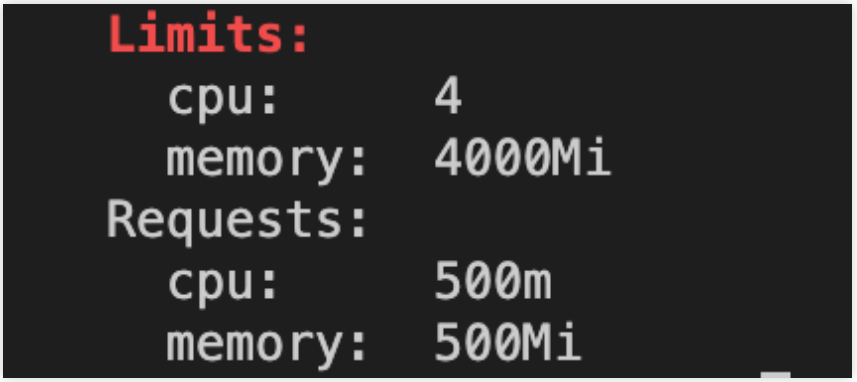
No impact.

Possible Causes

- The memory limit assigned to the pod is too small and needs to be increased.
- The application may have a memory leak, causing usage to increase gradually and eventually exceed the limit. The program's memory reclamation mechanism needs optimization.

Processing Procedures

1. Check pod resource limits:
    - Basic platform workload: *kubectl describe pod -n tce cloud-tcenter-xxx | grep -A 5 'Limits:'*
    - Audit log workload: *kubectl describe pod -n tce t|ocloud-cloudaudit-xxx | grep -A 5 'Limits:'*
    - CAM workload: *kubectl describe pod -n tce t|ocloud-tcenter-cam/tcloud-tcenter-support-cam | grep -A 5 'Limits:'*
    - Tag workload: *kubectl describe pod -n tce tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag-xxx | grep -A 5 'Limits:'*
    - Cloud API management workload: *kubectl describe pod -n tce cloud-tcenter-yunapi3-xxx | grep -A 5 'Limits:'*



2. View historical monitoring information of pod:
   Log in to the operation side, choose Monitoring System (CWhistle) > Cloud Product Metric > TCS/Pod, query the memory utilization monitoring data of the Pod that triggered the alarm around the time the alarm was triggered or for a longer period, and determine whether the behavior is normal.

Time Range	The day of the fault
Monitoring Metric	pod_memory_nocache_limit_usage
Cluster	global
Namespace	tce
Pod	◦ Basic platform workload: cloud-tcenter-xxx ◦ Audit log workload: t|ocloud-cloudaudit-xxx ◦ CAM workload: t|ocloud-tcenter-cam/tcloud-tcenter-support-cam ◦ Tag workload: tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag-xxx ◦ Cloud API management workload: cloud-tcenter-yunapi3-xxx

3. Collect relevant information and contact the technical team for support.

Emergency Plan
User and Permission: No Permission to Access Product Pages After Product Permissions Are Bound

Scenario Description of Failure Symptoms

No permission to access product pages after product permissions are bound.

Failure Location and Analysis

1. Confirm whether the permission policy is normal: Check on the operation side whether the corresponding policy syntax includes actions and APIs for page access.

2. Confirm whether the policy information in the database is normal:
   Log in to the CAM database, query the policy bound to the user by UIN in the cExtension table, and check if the API required by the user exists in actionName. If not, query the data for the service corresponding to the API by serviceType in the cServicePerm table, and check if the required API exists in apiEnName.

   ```
   dbsql_info dbsql-tcenter-cam-cauth/dbsql-tcenter-ocloud-cam-cauth
   select * from cExtension where uin='xxx' and actionName like "%xxx%"\G
   select * from cExtension where uin=xxx and strategyId in (xxx);
   select * from `cServicePerm` where serviceType='xxx'\G
   select * from cServicePerm where serviceType in (xxx);
   select * from cServicePerm where permId in (xxx) and apiEnName like '%xxx%'\G
   select distinct permId from cStrategyPerm where strategyId=xxx;
   ```

3. Check whether the policy information is normal: View the details of the bound policy in the cStrategy table.

   ```
   set names latin1
   select * from cStrategy where strategyName = 'cam_permissions_for_xxx'\G
   select * from cStrategy where  strategyInfo like '%xxx%'\G
   select * from cStrategy where strategyId="xxx";
   ```

4. Check the record information on the policy binding task:
   Check whether the policy binding record in the jobdetail table is abnormal.

   ```
   select * from cExtendJobDetail where uin=xxx;
   select * from cExtendJobDetail where uin=xxx order by updateTime desc limit 10;
   ```

Check whether the record of permission policy addition via a user group is abnormal.

```
select * from cReceiverInfo where uin=xxx; // uid
select * from cGroupMemberInfo where uid=xxx; // groupId
select * from cExtendJobDetail where groupId in (xxx) order by updateTime desc limit 20; //
select * from cRelatedStrategy where groupId=xxx;
```

5. Further analyze the issue by reproducing it through requests:

   i. Use the account with the issue to perform the action that triggers the issue. Then, use a request ID without permissions to query the log in the cam-auth component.
   ii. When the log is queried, modify the content of the log by adding "debug": 1.
   iii. Run curl 127.0.0.1:9502 -d'' to initiate a request with modified parameters.


```
{
  "version": 1,
  "componentName": "API",
  "eventId": "4627b684-bcfb-4206-a72a-a46a9ac173a1",
  "timestamp": 1744355915,
  "interface": {
    "para": {
      "version": "2.0",
      "method": "POST",
      "mode": 0,
      "debug": 1,
      "url": "cbs.yunapi3.3100arm-baseline.fsphere.cn/?
Action=DescribeDepotTransferTaskOverview&DepotIds.0=16777291&Language=zh-
CN&Nonce=1744355915&OIDCIdToken=&OIDCRefreshToken=&Region=[redacted]&Request
Client=CAPI_GO_SDK&RequestSource=&SecretId=AKIDB0IgYdiELni7MCTKBrV0nqcNqLeRIt
yd&SignatureMethod=HmacSHA256&Timestamp=1744355915&Token=81d4a1873a606507
d8b5db2a915acdce447cb8d710001&Version=2017-03-12",
      "header": "authorization:q-sign-algorithm=api_sha256&q-
ak=AKIDB0IgYdiELni7MCTKBrV0nqcNqLeRItyd&q-sign-time=1744355615;1744356215&q-
key-time=1744355615;1744356215&q-header-list=&q-url-param-
list=Action;DepotIds.0;Language;Nonce;OIDCIdToken;OIDCRefreshToken;Region;RequestClie
nt;RequestSource;SecretId;SignatureMethod;Timestamp;Token;Version&q-
```


Failure Emergency Response Steps

1. If the preset data permission policy is missing, manually update the preset component.

```
kaleido_preset update /data/tce_dc/software/lastest/preset/preset-xxx
```

2. If dirty data exists, clear the dirty data.

   i. Back up the original data.

```
tad job dbBackup --servicebindings=dbsql-tcenter-cam-cauth
tad job dbBackup --servicebindings=dbsql-tcenter-ocloud-cam-cauth
```

   ii. Delete the dirty data in cServicePerm.

   iii. Refresh the preset component.

```
kaleido_preset update preset-
```

   iv. Check whether cStrategyPerm and cExtension are referenced correctly.

Appendix A: Common Operations
Common Operations

Restarting a Pod

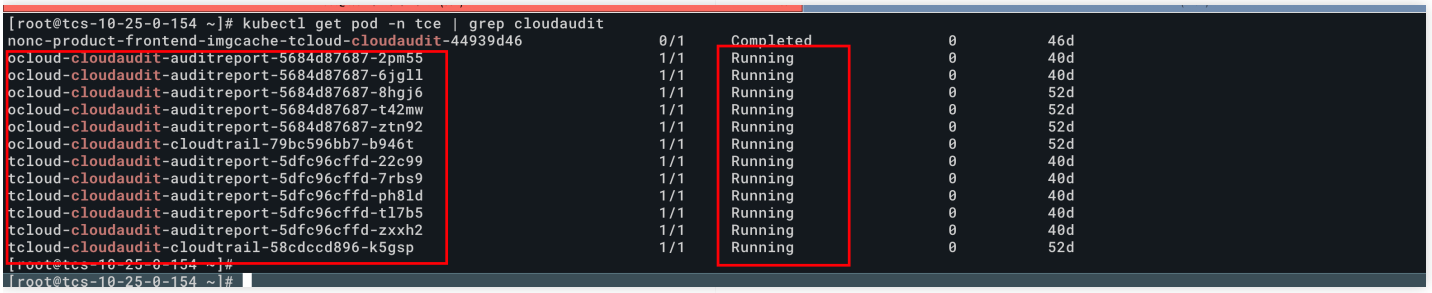
Operation Description

Used as a quick workload recovery attempt when there are page errors or the pod is in an abnormal status.

Processing Procedures

Query the audit service pods and delete the pods that need to be deleted.

kubectl get pod -n tce | grep cloudaudit
kubectl delete pod -n tce tcloud-cloudaudit-xxxxx



Viewing the Service Logs

Operation Description

View the service logs for troubleshooting.

Processing Procedures

Query the pod, enter the container, and view logs.

kubectl get pod -n tce | grep cloudaudit

kubectl exec -it -n tce tcloud-cloudaudit-auditreport-xxxxx bash
cd /data/log/python/
vim xxxxxxx-log.txt

kubectl exec -it -n tce tcloud-cloudaudit-cloudtrail-xxxxx bash

```
cd /data/log/cloudtrail/detail
vim daily_xxxxxx.log
```

Obtaining Service Logs

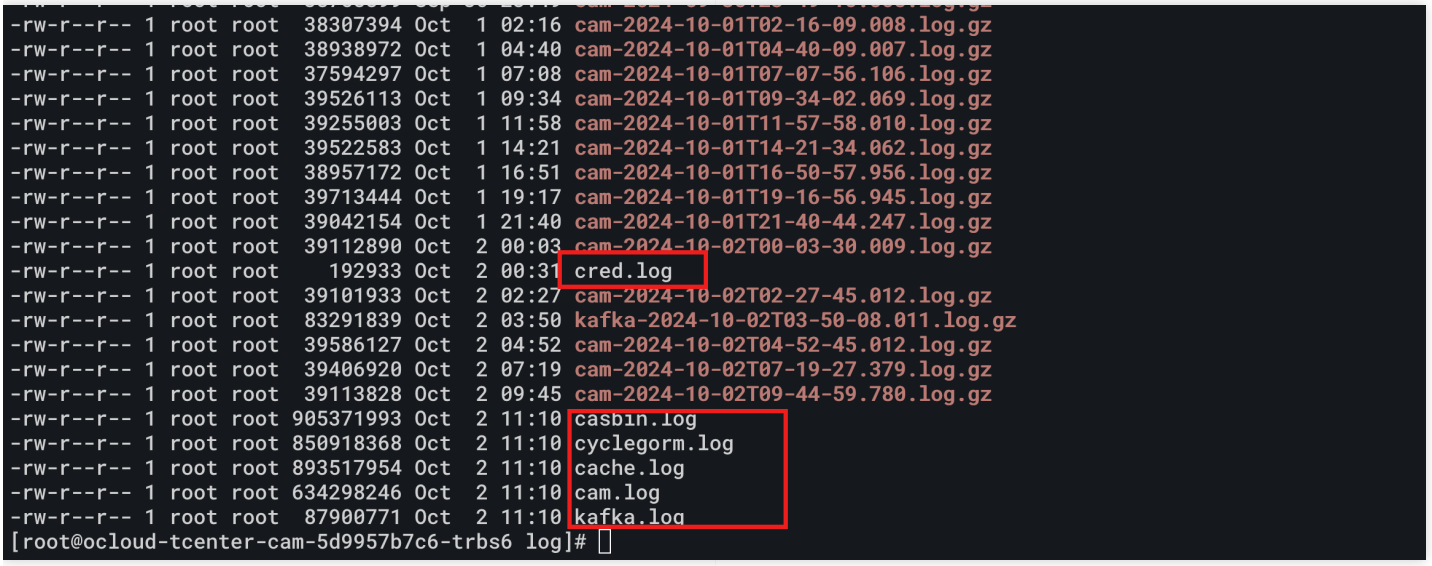
Operation Description

When the CAM service is abnormal, you need to view the logs for troubleshooting.

Processing Procedures

In containerized deployments, enter tcloud-tcenter-cam or ocloud-tcenter-cam, then enter the /data/log directory to view cam.log and other relevant logs.

kubectl get pod -n tce | grep cloud-tcenter-cam
kubectl exec -it -n tce ocloud-tcenter-cam-xxxxxxx bash
cd /data/log/
tail -f cam.log and other logs.



Viewing Pod Rendering Status

Operation Description

Daily inspection of pod status.

Processing Procedures

1. Enter the pod and view the rendering status:

kubectl get pod -ntce | grep -E 'tcenter-platform-tag|tcenter-platform-wtag'
Enter the pod and view the rendering status.
kubectl exec -it -n tce tcloud-tcenter-platform-tag-xxxxx bash
cat /tce/conf/config/tce.config.center/sdk.json

kubectl exec -it -n tce tcloud-tcenter-platform-tag-wtag-xxxxx bash
Enter the pod and view the rendering status.
cat /tce/conf/config/tce.config.center/sdk.json
cat /data/release/swoole_tag/application/config/idc/hosts_config.php

2. Database rendering of the tcloud-tcenter-platform-tag service:



3. Check whether the database rendering of *tcloud-tcenter-platform-tag-wtag* is normal:



4. Check whether the rendering of CMQ is normal:

```
[root@tcloud-tcenter-platform-wtag-59f5c4565-5pkwb detail]# cat /data/release/swoole_tag_write/application/config/idc/hosts_config.php
<?php

class HostsConfig{
    public static function getResourceUrl(){
        $config = array();
        $config['gz']['cvm']['cvm'] = 'http:///interfaces/interface.php';
        return $config;
    }
    const ACCOUNT_URL = "http://tcloud-tcenter-platform-account.tcenter:6060/account";
    const WHITE_LIST_URL = "http://tcloud-tcenter-platform-whitelist.tcenter:6060/whitelist";
    const CAM_TAG_URL = "http://tcloud-tcenter-cam-grant-write.tcenter/camTag";
    const CMQ_DOMAIN = "";
    const CMQ_URL = "";
    const CMQ_SECRET_ID = "";
    const CMQ_SECRET_KEY = "";
    const CMQ_TOPIC_NAME = "tag-resource-topic";
    const CMQ_TOPIC_NAME1 = "tag-resource-topic";
    const CMQ_TOPIC_NAME_PROJECT = "tag-project-resource-topic";
    const MQ_API_HOST = "http://   -support-cmq-1-1.ap-        -hqtest-ops.   3101 1test.fsphere.cn:5672/api/v1";
    const CMSI_URL = "";

    const CAM_AUTH_URL = "http://tcloud-tcenter-cam-auth.tcenter:9502";
}


[root@tcloud-tcenter-platform-wtag-59f5c4565-5pkwb detail]#
```

Obtaining Operation Logs

Operation Description

View the pod. Enter the pod to view logs.

Processing Procedures

```
kubectl get pod -n tce | grep tcenter-yunapi3
kubectl exec -it -n tce t/ocloud-tcenter-yunapi3-traefik-xxxx bash
cd /usr/local/services/traefik/log
# Filter logs by page reqid.
grep -ir "reqid" traefik.log
 grep -ir "reqid" *
```

Technical Metrics
User and Permission

- Success rate of user creation: 99.9%.
- Success rate of user group query: 99.9%.
- Success rate of policy creation: 99.9%.
- Success rate of policy binding: 99%.

Performance Metrics
User and Permission

Performance Metric	• User list query: 300 ms. • Creation: 1s. • User group list viewing: 300 ms. • User group creation: 1s. • Policy list viewing: 200 ms. • Custom policy creation: 1.5s. • Policy binding to users: 800 ms. • CPU utilization: 5%. • Memory utilization: 10%.
Product Limit Parameter	• Up to 1,000 custom policies can be created with a root account. • A single user can be associated with up to 5,000 policies.

Product White Paper
Product Overview

Tcenter and Business Management Platform (Basic Edition) (BMP) form a one-stop operation management platform built for enterprise customers. The platform provides core capabilities such as unified account and permission management, resource management, and product management to achieve centralized and automated management of the private cloud environment, helping customers simplify the management processes for personnel, resources, and products, thereby enhancing the operational efficiency of cloud businesses and enabling enterprises to focus more on core business innovation and development.

Product Strengths

Fine-grained operations

- The usage and cost data for cloud products can be provided for customers, and quota management capabilities can be provided.
- Group-based resource management is supported to simplify the complexities of resource grouping and authorization within a single tenant.
- Multi-level organizations within an enterprise can be matched, enabling unified operation management of multiple tenants within the enterprise.

Security operations

- Control over high-risk operations can be achieved through the login protection, operation protection, and approval management features.
- The cloud auditing feature enables auditing of all operations.
- Fine-grained permission management is provided to meet the permission control requirements within the enterprise.

Openness

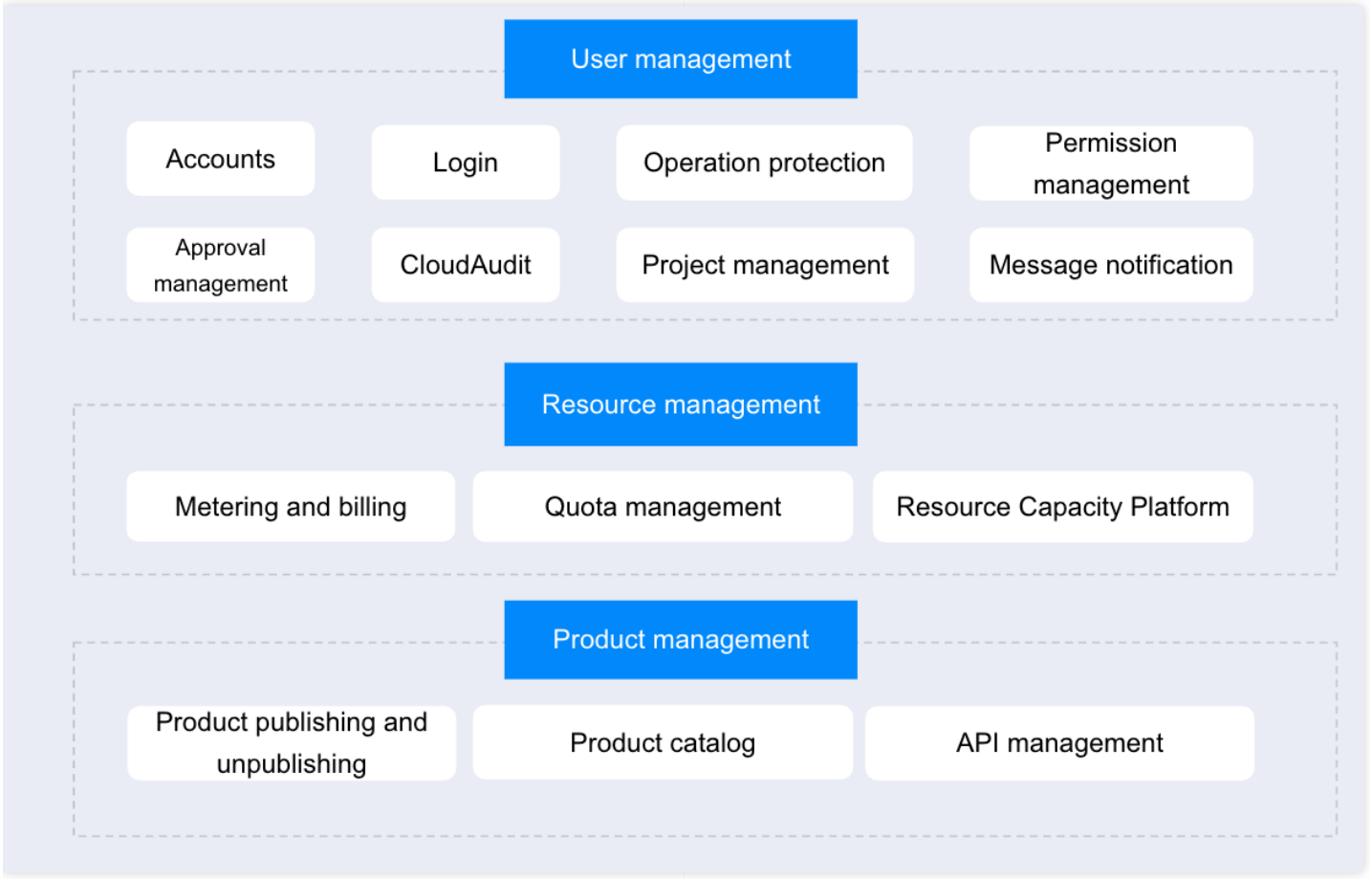
APIs of all products are fully open, enabling customers to conduct secondary development with these APIs.

Use Cases

1. Unified operations
Case description: For small and medium-sized enterprises, the management is relatively centralized without departmental divisions.
Solution description: Enterprises can create two tenants, one as a test environment and the other as a production environment. Under the test tenant or production tenant, projects can be created as needed, and different resources and accounts can be added to corresponding projects. Resources in a project can only be managed by members within the project. This solution helps customers achieve group management of resources within tenants.
2. Security operations
Case description: For financial and insurance industries, cybersecurity classified protection requirements should be met, and traceable operations and fine-grained permissions are required.
Solution description: The cloud auditing feature can record all operation records of users; the permission management feature helps customers achieve permission control at the operation level and resource level; the login protection, operation protection, and approval management features help customers control high-risk operations.

Product Architecture

Tcenter and BMP primarily provide the capabilities of user management, resource management, and product management. The specific feature architecture is as follows.



Features

- A secure and compliant cloud usage and management process is provided for customers: Capabilities, including account, login, operation protection, permission management, approval management, cloud auditing, project management, and message notification, are provided, ensuring secure and compliant cloud usage and management for customers.
- Fine-grained operation capabilities: Through billing and metering, quota management, and the capacity platform, fine-grained resource management can be achieved.
- Openness: Cloud APIs are fully open, enabling customers to conduct secondary development.
  Account: The platform supports multi-tenancy and resource isolation between tenants. The accounts of a tenant are divided into root accounts and sub-accounts, and the permissions of sub-accounts should be assigned by administrators.
  Login: The platform supports login protection. When users log in to the cloud platform, identity verification (such as SMS verification code and token) is required. This prevents unauthorized access even if passwords are stolen, ensuring maximum account security.
  Operation protection: The operation protection feature effectively ensures the security of account resources. Before performing sensitive operations, cloud users should complete identity verification (such as SMS verification code and token), and can only proceed with the operation after successful verification.
  Permission management: The private cloud provides permission management at the API level and resource level, helping customers securely and conveniently manage access to cloud services and resources.
  Approval management: The platform supports custom approval flow orchestration. High-risk operations should be approved by administrators before they can be executed.
  Cloud auditing: Users' access and usage behaviors of cloud products and services through the cloud console, API, and SDK can be audited to meet security and compliance audit requirements.
  Project management: The complexity issues of resource grouping, authorization management, and project-based cost allocation within a single tenant account can be resolved.
  Billing and metering: The statistics of resource instance usage information for each tenant are supported, and unified features such as aggregated statistics, display, query, and filtering are provided.
  Quota management: Through quota management, resource quotas for various cloud products can be uniformly allocated to tenants.
  Resource capacity platform: This platform provides customers with capacity visualization, capacity alarms, and capacity prediction features, enabling customers to intuitively perceive the resource usage of the entire resource pool and make timely decisions (such as scale-out) based on workload conditions.
  Report platform: This platform provides customers with unified and automated report services. It

converts scattered resource data on the cloud platform into standardized reports, helping customers regularly and clearly understand the basic conditions of resources.
Product listing and removal: The listing, grayscale listing, and removal operations can be performed on cloud products. After a product is listed, all tenants can access the product on the cloud platform; after the grayscale listing operation is performed, the product is only visible to tenants on the grayscale list; after a product is removed, the product becomes invisible to all tenants in the console.
Product catalog: Product catalogs visible to tenants can be configured for specified tenants.
API management: APIs of all products on the platform are fully open, with API documentation provided, enabling customers to conduct secondary development with these APIs.
Message: Operation messages or alarm messages can be promptly sent to users.

Usage Recommendations

Basic Platform

Software Usage Recommendations

Metric Name	Default Value	Value Range
Session timeout time on the tenant side	30 minutes	30 to 1440 minutes
Session timeout time on the operation side	30 minutes	30 to 1440 minutes
Tenant-side account temporary locking time	60 minutes	10 to 1440 minutes
Operation-side account temporary locking time	60 minutes	10 to 1440 minutes
Maximum incorrect password attempts during tenant-side login	10	3 to 10
Maximum incorrect password attempts during operation-side login	10	3 to 10
Multi-end login switch on the tenant side	Disabled	Disabled or enabled
Multi-end login switch on the operation side	Disabled	Disabled or enabled
Graphic verification code for operation-side login	Disabled	Disabled or enabled
Graphic verification code for tenant-side login	Disabled	Disabled or enabled
Tenant-side watermark	Disabled	Disabled or enabled

Metric Name	Default Value	Value Range
Operation-side watermark	Disabled	Disabled or enabled

Users and Permissions

Software Usage Recommendations

Metric Name	Default Value	Value Range
Number of accounts per tenant	10,000	Non-adjustable
Number of user groups per tenant	1,000	Non-adjustable
Number of user groups a sub-account can join	300	Non-adjustable
Number of users in a user group	1,000	Non-adjustable
Number of custom policies per tenant	1,000	Non-adjustable
Number of policies that can be associated with a user, user group, or role	5,000	Non-adjustable
Maximum number of characters in a policy syntax	4,096 characters	Non-adjustable
SSO protocol	-	OAuth 2.0, SAML, CAS, and LDAP protocols
Password composition	Lowercase letters and digits	At least two types of uppercase and lowercase letters, digits, and special characters
Minimum password length	10 characters	10–32 characters
Password validity period	90 days	0 (no limit)–365 days

Metric Name	Default Value	Value Range
Reuse limit	1 time (cannot be the same as the previous password by default)	1–24 times

- It is recommended to perform a quarterly scan and cleanup of tenant root accounts and sub-accounts that have been inactive for an extended period (≥ 90 days). The cleanup follows a three-step process: "disable first, observe for 30 days, and delete". Before cleanup, confirm that the account has no active AK/SK and no associated resources.
- It is not recommended to use the root account for resource operations. Instead, create sub-accounts and assign policies based on the principle of least privilege, and use sub-accounts with limited permissions to manage your cloud resources.

Cloud API Management

Software Usage Recommendations

Metric Name	Default Value	Value Range
Multi-language SDKs provided by Cloud API	-	Cloud API provides SDKs in the following languages: Java, Python, PHP, Golang, and Node.js
API timeout	5000 ms	10 - 30000 ms
API call frequency per tenant	20 calls per second	Non-adjustable
Number of parameters per API	-	0 - 20
Number of parameters per complex object	-	0 - 20
Request size	1024 KB	1024 - 10240 KB

- Product APIs provided by the cloud platform can only be called to query resources. Editing and deleting resources are not supported.
- APIs of user-defined products can be called to add, edit, and delete resources.
- Cloud API keys should be rotated regularly, and deprecated keys should be promptly disabled to

prevent security risks caused by key leakage.
- The SDK version must match the Cloud API version to avoid SDK call failures.

Resource Management

Software Usage Recommendations

The scope of products integrated into project management is as follows.

Cloud Product	Resource Type
CVM	Cloud virtual machine
CFS	File system
CBS	Cloud disk
	Snapshot
Cloud Storage Private (CSP)	Bucket
Cloud Object Storage (COS)	Bucket
Redis®	Instance
TDSQL for MySQL	Instance
TDSQL for PostgreSQL	Instance
Cloud Load Balancer (CLB)	LB instance
Distributed Transaction Framework (DTF)	Transaction
TDMQ for CKafka (CKafka)	Instance
Cloud Task (TCT)	Deployment group
Big Data Suite (TBDS)	Instance

Metering Management

Software Usage Recommendations

The metering and billing module supports the following products. If Billing Management is deployed for your project, you can bill products on a pay-as-you-go basis.

Category	Product
Compute	- CVM - BMS
Network	- Cloud Load Balancer (CLB) - Elastic IP (EIP) - NAT Gateway (NATGW) - NAT64 Gateway - VPN Gateway - Direct Connect (DC) - Peering Connection (PC) - IDC Bandwidth
Storage	- CBS - CBS Snapshot - Cloud Object Storage (COS) - Cloud Storage Private (CSP) - CFS - TurboFS - BRC - Cloud Log Service (CLS)
Database	- TDSQL for MySQL - TDSQL for PostgreSQL - Redis®
Middleware	- TDMQ for Apache Pulsar (TDMQ Pulsar) - TDMQ for CKafka (CKafka) - TDMQ for Apache RocketMQ (TDMQ RocketMQ)
Security	- Cloud Workload Protection Platform (CWP) - Web Application Firewall (WAF) - Security Operation Center (SOC) - Cloud Bastion Host (CBH) - Data Security Audit (DSAuditSaaS) - Key Management Service (KMS) - Secrets Manager (SSM) - Cloud Hardware Security Module (CloudHSM) - Cloud Firewall (CFW)

Quotas for products supported by metering and billing:

Cloud Product	Quota
CVM	Number of instances
	Number of CPU cores
	Memory size
BMS	Number of servers
CBS	Number of instances
	Amount of storage space
CBS Snapshot	Number of snapshots
CFS	Number of instances
TurboFS	Number of instances
	Amount of storage space
BRC	Number of backup points
TDSQL for MySQL	Number of instances
	Number of CPU cores
	Amount of storage space
TDSQL for PostgreSQL	Number of instances
	Number of CPU cores
	Amount of storage space
Redis®	Number of instances
	Memory

CloudAudit

Software Usage Recommendations

- Calls to all cloud APIs can be recorded.
- API call records within up to one year can be queried.
- For financial and cybersecurity classified protection scenarios requiring longer-term retention of audit logs, you should deliver the logs to COS or equivalent storage via CloudAudit to achieve long-term retention, rather than relying solely on CloudAudit for storage. On the operation side, logs can be delivered to Kafka. On the tenant side, logs can be delivered to COS/CSP.

Approval Management

Software Usage Recommendations

Range of the products that have been accessed by Approval Management:

Side	Product
Tenant side	CVM
	Cloud Load Balancer (CLB)
	CBS
	Virtual Private Cloud (VPC)
	Kubernetes Engine (TKE)
	BMS
	MariaDB
	TDSQL for MySQL
	TDMQ for CKafka (CKafka)
	TDMQ for Apache Pulsar (TDMQ Pulsar)
	Direct Connect (DC)
	CFS

Side	Product
	Resource Management
Operation side	Tcenter
	BMS
	Data Center Operating System (DCOS)
	Cloud Load Balancer (CLB)
	Virtual Private Cloud (VPC)
	Direct Connect (DC)
	CVM
	Job Tools
	Cloud Workload Protection Platform (CWP)
	CBS
	CFS
	Redis®
	Product Ops Center
	Cloud Object Storage (COS)
	Network Management System (NMS)
	Password Bank
	Agent Management
	Web Application Firewall (WAF)

Tags

Tags can be applied to cloud product resource instances to enable categorized resource management.

Product Introduction
Overview of CAM

Cloud Access Management (CAM) is a web service provided by the cloud platform, primarily designed to assist customers in securely managing access permissions for resources under their cloud platform accounts. Users can create, manage, and terminate users (groups) through CAM, and use identity management and policy management to control the permissions of other users using cloud platform resources.

Product Features

CAM provides the following support features:

Authorized access to root account resources

Allows you to authorize resources of the root account to other users, including sub-accounts and other root accounts, without sharing the identity credentials of the root account.

Refined permission management

You can grant different levels of access permissions for different resources to different users. For example, some sub-accounts can be granted permission to operate a specific CVM virtual machine, while other accounts or root accounts can be granted permission to operate CVM in a specific region. Resources, access permissions, and users can all be packaged in batches.

Final consistency

CAM currently supports multiple regions of the cloud platform. It achieves cross-region data synchronization by replicating policy data. Although CAM policy changes are submitted promptly, cross-region policy synchronization may cause a delay in policy enforcement. Additionally, CAM uses caching to improve performance (currently a one-minute cache), so updates take effect after the cache expires.

Use Cases

Sub-account permission management for enterprises

Employees in different positions within the enterprise need to have minimized access permissions to the enterprise's cloud resources.
Scene: An enterprise owns numerous cloud resources, including CVM, VPC instances, COS buckets, and objects. The enterprise has many employees, including developers, testers, and operations personnel. Some developers need read-write permissions to cloud resources related to their projects, testers need read/write permissions to testing cloud resources related to their projects, and operations personnel are responsible for purchasing machines and daily operations. When there are changes in an employee's responsibilities or project involvement, the corresponding permissions will be terminated.

Use Limits

Item	Limit Value
Number of user groups in a root account	300
Number of sub-accounts in a root account	2000
Number of user groups that a sub-account can join	300
Number of sub-accounts in a user group	300
Number of custom policies that can be created by a root account	1000
Maximum number of characters in a policy syntax	4096

> Note:
>
> 1. The number of custom policies that can be created by a root account includes COS custom policies. If you encounter a Exceed the Maximum Number of Custom Policies (Limit is 1500) prompt and the CAM custom policy number has not reached the limit, you can go to the COS bucket list - console, click on the bucket name to enter the permission management to check if the ACL (Access Control List) number exceeds the limit.
> 2. The number of policies directly associated with a user or user group includes COS custom policies. If you encounter a Policy Association Failure prompt and the number of associated policies within CAM has not reached the limit, you can go to the COS bucket list - console, click on the bucket name to enter the permission management to check if the ACL (Access Control List) number exceeds the limit.

Products Supported by CAM

Introduction

Cloud Access Management already supports permission management for most Cloud Services. This document mainly introduces relevant information about the cloud product services that support Cloud Access Management (CAM). Specific dimensions include authorization granularity, console, authorization based on tags, reference documentation, etc. The following list separately lists the services supported by CAM under various major product categories on the cloud platform. Definitions for the information in the table are as follows:

- Service: The name of the cloud service that supports CAM, click on the link to the corresponding product service documentation for quick access to relevant information.
- Authorization Granularity: The minimum authorization granularity provided by the current service. Authorization granularity is divided into three levels of granularity: service level, operation level, and resource level.
- Service Level: Defines whether access permissions to the entire service are granted, divided into allowing full operational permissions for the service or denying all operational permissions for the service.
- Operation Level: Defines whether access permissions to specific interfaces (APIs) of the service are granted, for example: authorizing an account to perform read-only operations on the Cloud Virtual Machine service.
- Resource Level: Defines whether access permissions to specific resources are granted. This is the finest authorization granularity, for example: authorizing an account to only read and write operations on a specific Cloud Virtual Machine.
- Console: Indicates whether sub-accounts can access the current service through the console. "?" indicates support, - indicates temporary lack of support.
- Authorization Based on Tags: Indicates whether the current service supports permission management based on tags. "?" indicates support, "-" indicates temporary lack of support.
- Reference Documentation: Link to documents related to CAM for the current service. "-" indicates none available.

Computing

Service	Authorization Granularity	Console	Authorization Based on Tags	Service Roles	Reference Documentation

Service	Authorization Granularity	Console	Authorization Based on Tags	Service Roles	Reference Documentation
Cloud Virtual Machine	Resource Level	✓	✓	-	

|Bare Metal Server|Resource Level|✓|-|-|-|

Storage

Service	Authorization Granularity	Console	Authorization Based on Tags	Service Roles	Reference Documentation
COS	Resource Level	✓	✓	-	
Cloud File Storage	Resource Level	✓	✓	-	
Cloud Block Storage	Resource Level	✓	✓	-	-

Network

Service	Authorization Granularity	Console	Authorization Based on Tags	Service Roles	Reference Documentation
Load balancing	Resource Level	✓	✓	-	
Virtual Private Cloud (VPC)	Resource Level	✓	✓	-	

Management and Audit

Service	Authorization Granularity	Console	Authorization Based on Tags	Service Roles	Reference Documentation

Service	Authorization Granularity	Console	Authorization Based on Tags	Service Roles	Reference Documentation
Access Management	Operation Level	✓	-	-	
Cloud Audit	Operation Level	✓	-	-	-

Monitoring and Operations

Service	Authorization Granularity	Console	Authorization Based on Tags	Service Roles	Reference Documentation
Cloud Monitor	Operation Level	✓	-	-	-

Public Service

Service	Authorization Granularity	Console	Authorization Based on Tags	Service Roles	Reference Documentation
VPC DNS (VPCDNS)	Service Level	✓	-	-	-

Security

Service	Authorization Granularity	Console	Authorization Based on Tags	Service Roles	Reference Documentation
CWPP (CWP)	Service Level	✓	-	-	-
Secrets Manager (SSM)	Service Level	✓	-	-	-
Web Application Firewall (WAF)	Service Level	✓	-	-	-
Cloud Firewall (CFW)	Service Level	✓	-	-	-

Operations Platform

Service	Authorization Granularity	Console	Authorization Based on Tags	Service Roles	Reference Documentation
Cloud Audit	Service Level	✓	-	-	-
Access Management	Service Level	✓	-	-	-
Tag	Service Level	✓	-	-	-

Operation Guide
Overview

Log in to the cloud access management console and select Overview in the left sidebar to access the overview page. The overview page displays the quantities of users, user groups, custom policies, and roles, as well as the creation entrance and cloud access management guide.

User Management
User Type

A CAM user is an entity you created on the cloud platform, and each CAM user is associated with only one cloud account. The identity of your registered cloud account is the root account. You can use User Management to create sub-accounts with different permissions for assistance. Sub-accounts are categorized into sub-users and message recipients.

Account Type	Root account	Sub-account	
		Sub-user	Message Recipient
Definition	Possess all resources of the cloud platform and can access any of its resources. You are not advised to use the root account for resource operations. Instead, you should create sub-accounts, assign policies based on the principle of least privilege, and use sub-accounts with limited permissions to manage your cloud resources.	Created by the root account and fully owned by the root account that creates the sub-user.	Only has the message receiving feature.
Console Access	✔	✔	-
Programmatic Access	Default has full policy access.	✔	-
Policy Authorization	✔	✔	-
Message Notification	✔	✔	✔

Root Account

Overview

This document describes the root account permission settings and message reception. You can follow the steps below to understand the root account permissions and how to modify the message reception method.

Prerequisites

You have registered a cloud platform account, namely the root account.

Directions

Root Account Does Not Require Authorization

The root account automatically possesses all resources under the cloud platform account without requiring authorization and can access any resource at will. Therefore, it is not recommended to perform resource operations using the root account. Instead, you should create sub-accounts and assign policies based on the principle of least privilege, allowing sub-accounts with limited permission scopes to manage your cloud resources.

Root Account Message Reception

The security phone number and security email registered when you sign up for the cloud platform root account will serve as the initial message reception methods. If you modify the security phone number or security email within the Account Center > Console, the contact phone number or contact email used for cloud platform message notifications in the Cloud Access Management (CAM) > Console will not be synchronized.

> Note: To avoid any losses caused by missed messages, please promptly visit the Cloud Access Management console to confirm whether the contact phone number or contact email used for message subscriptions meets your expectations.

Sub-account
Creating Sub-users

Scenarios

This document describes how to create a sub-user and set permissions for it, allowing it to manage resources under the root account within the granted permission scope.

Operation Guide

Creating Sub-users via Console

You can log in to the cloud console to create a sub-user and set permissions for it.

1. Log in to the CAM console, and in the left sidebar, choose User Management > User to enter the User page.

2. On the User page, click Create User. In the displayed Select User Type dialog box, select Sub-user and enter the Create Sub-user page.

3. On the page for entering user information, under Set User Information, enter the username (required), nickname, mobile phone number, and email address.

   > Note: Up to 10 users can be created at a time by clicking Add User. - Usernames of sub-users cannot be modified once they are set since they are used for login.

4. Under Access Type, set the sub-user's access method.

   - Programmatic access: Enable SecretId and SecretKey for the sub-user to manage root account resources within its permission scope using cloud APIs, SDKs, and other development tools.
   - Cloud platform management console access: Enable password, and sub-users will manage root account resources within their permission scope by logging into the cloud console.

     > Note: For account security, it is advised to enable login protection and operation protection.

5. Set Console Password, Password Reset Required, Login Protection, and Operation Protection.

6. Click Create to create a sub-user. After successful creation, the system will automatically enter the Set

Permissions page.

7. In Set Permissions, you can choose:
    - Adding to an existing user group: Create a user group and add the sub-user to it, or join an existing user group.
    - Copying existing user permissions: Copy permissions from an existing user to the current user.
    - Granting permissions based on the policy list: Select a policy and grant the corresponding policy permissions to the current sub-user.
8. Click Next: Complete to enter the User Permission Setting page.
9. On the User Permission Settings page, you can choose different methods to set permissions for the new sub-user as needed. After associating with the policy, the sub-user will obtain the permissions described in the policy.
    - Adding to an existing user group: It is the best practice for managing user permissions by job function to add the sub-user to a group. In this way, you can obtain permissions. By adding the sub-user to an existing user group or creating a user group, the sub-user can be associated with the policies attached to that group.
    - Copying existing user permissions: Associate a sub-user with policies by copying the permissions of an existing user. Click Copy Existing User Permissions, select the user whose permissions need to be copied, and the sub-user can be associated with the policies attached to the copied user.
    - Granting permissions based on the policy list: Click Authorize from the Policy List, and select the policies to be associated.
10. Click Next: Complete.
11. On the Completion page, click Complete to complete the creation of a sub-user. A page indicates that the new sub-user has been successfully created is displayed.
12. On this page, you can obtain the sub-user information through the following method.

- Click Download Security Credentials to save some information to a local Excel file.

Creating Sub-users via APIs

You can use access keys to call the AddUser API to add a sub-user and set permissions for it.

Sub-user Permission Settings

Overview

This documentation describes how to authorize and revoke policies associated with sub-users. Sub-users will manage resources under the root account within the granted permissions.

Directions

Authorize Association of Policies for Sub-users

Direct Association

You can directly associate policies with users to obtain the permissions included in the policies.

1. Log in to the Cloud Access Management Console, select User Management > User to enter the User Management page.
2. On the User Management page, click on the user to enter the user details page.
3. Click on Associated Policies to view the policies already associated.
4. Click on Associate Policy, select policies from the pop-up policy list.
5. Click OK to complete the direct association of policies for sub-users.

Associated with Groups

You can add users to user groups, and users will automatically obtain the permissions associated with the user group. The policies obtained through this method are associated with groups. To revoke policies associated with groups, remove users from the respective user groups.

1. Log in to the Cloud Access Management Console, select User Management > User Group to enter the User Management page.
2. On the User Group Management page, click on the user group to enter the user group details page.
3. Click on Associated Policies to view the policies already associated.
4. Click on Associate Policy, select policies from the pop-up policy list.
5. Click OK to complete the authorization of policies associated with user groups.

Revoking Associated Policies for Sub-users

Directly Revoking Associated Policies for Sub-users

You can directly revoke policies associated with users to revoke the permissions associated with the user.

1. Log in to the Cloud Access Management Console, select User Management > User to enter the User

Management page.
2. On the User Management page, click on the user to enter the user details page.
3. Click on Associated Policies to view the policies already associated.
4. Click Revoke Policy.
5. Click OK to complete the direct revocation of policies associated with sub-users.

Removing Users from Groups

You can remove users from groups to revoke policies associated with the users.

1. Log in to the Cloud Access Management Console, select User Management > User Groups to enter the User Group Management page.
2. On the User Group Management page, click on the user group to enter the user group details page.
3. Click on Associated Policies to view the policies already associated.
4. Click Revoke Policy.
5. Click OK to complete the direct revocation of policies associated with sub-users.

Sub-user API Key Management

Scenarios

This section describes how to manage cloud API keys of sub-users. Sub-users can call cloud APIs through cloud API keys.

Note: Keys can only be created for sub-users but not for the root account. Using the root account is not recommended in actual projects.

Operation Steps

Creating a Key for a Sub-user

1. Log in to the CAM console and choose User Management > User to enter the User Management page.
2. On the User Management page, click a user to enter the User Details page, and click API Key Management.
3. Click Create Key and select a key creation method: system generation or custom creation.
4. A newly created key is added to the key list.

Disabling a Sub-user Key

1. Log in to the CAM console and choose User Management > User to enter the User Management page.
2. On the User Management page, click a user to enter the User Details page, and click API Key Management.
3. In the Operation column of the key list, click Disable.
4. In the operation confirmation pop-up window, click Confirm.
5. After the key is disabled, the user cannot call cloud APIs through this key.

Enabling a Sub-user Key

1. Log in to the CAM console and choose User Management > User to enter the User Management page.
2. On the User Management page, click a user to enter the User Details page, and click API Key Management.
3. In the Operation column of the key list, click Enable.
4. After the key is enabled, the user can call cloud APIs through this key.

Deleting a Sub-user Key

1. Log in to the CAM console and choose User Management > User to enter the User Management page.
2. On the User Management page, click a user to enter the User Details page, and click API Key Management.
3. In the Operation column of the key list, click Delete. Note: The key should be disabled before deletion.
4. After the key is deleted, the user cannot call cloud APIs through this key.

Sub-user Security Credentials
Sub-user Login

Scenarios

This document describes how to log in as a sub-user or WeCom sub-user. Upon successful login, the sub-user will manage the resources under the root account within the permission scope.

Operation Steps

Sub-user Login

Logging in with Account and Password

You can use the following steps to log in as a custom sub-user with an account and password. Upon successful login, you can manage the resources under the root account within the set permissions.

1. Go to the Sub-user Login page to log in with an account.

2. On the Sub-user Login page, enter the root account ID, sub-user name, and login password. > The root account ID is the ID to which the sub-user belongs. The account ID (for example: 100001234567) is the UUID for the account on the cloud platform. Please contact the root account to view this information in the Account Information section.

3. Click Log In to log in as a sub-user with an account and password.

Resetting Log-in Password for Sub-Users

Overview

This document describes the process of modifying a sub-user's password, enabling them to access resources under the root account of the sub-user management with the new password.

Directions

1. In User Management > User, select the sub-user for whom you need to modify the password, select the specific Username and enter the user details page.
2. On the user details page, go to Security Settings > Console Password, and click on Manage Password.
3. In the pop-up window for Manage Console Access, set the current user password.
    - If the current sub-user needs to access the cloud through the console, please set Console Access to Enable.
    - If you need to set a new password for the sub-user, you can do so in two ways.
    - If you choose Auto-generated Password in Console Password, the system will automatically generate a console log-in password. You can copy and save it, or click Download .csv to save the password if needed.
    - If you choose Custom Password in Access Password, input the console log-in password that you have set for the sub-user.
    - If you want the current user to reset the password by themselves, check User Must Reset Password at Next Log-in. The sub-user will be prompted to reset the console login password after the next successful login.

Security Protection for Sub-Users

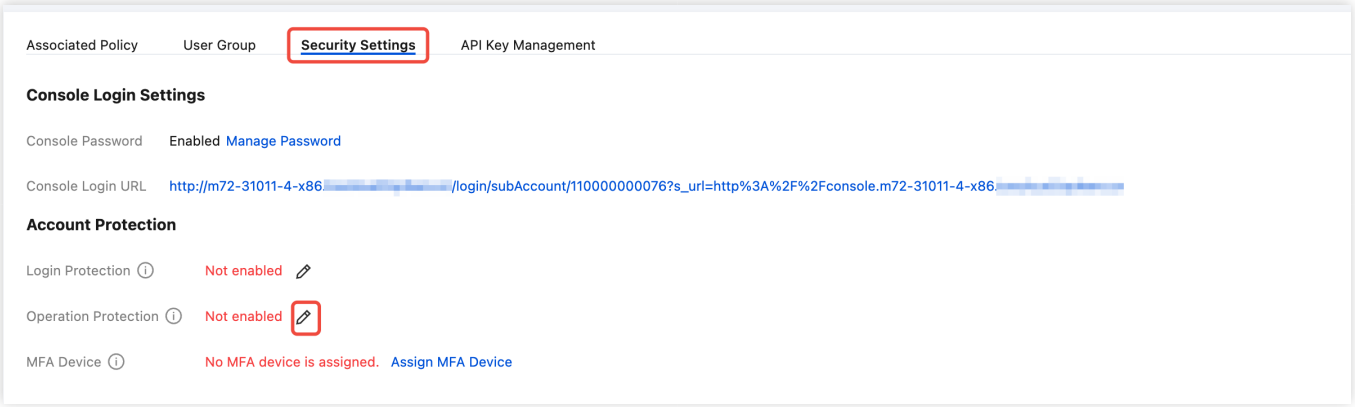
Scenarios

This document describes how to enable and disable security protection for sub-users. Sub-users decide whether to conduct security verification based on the settings.

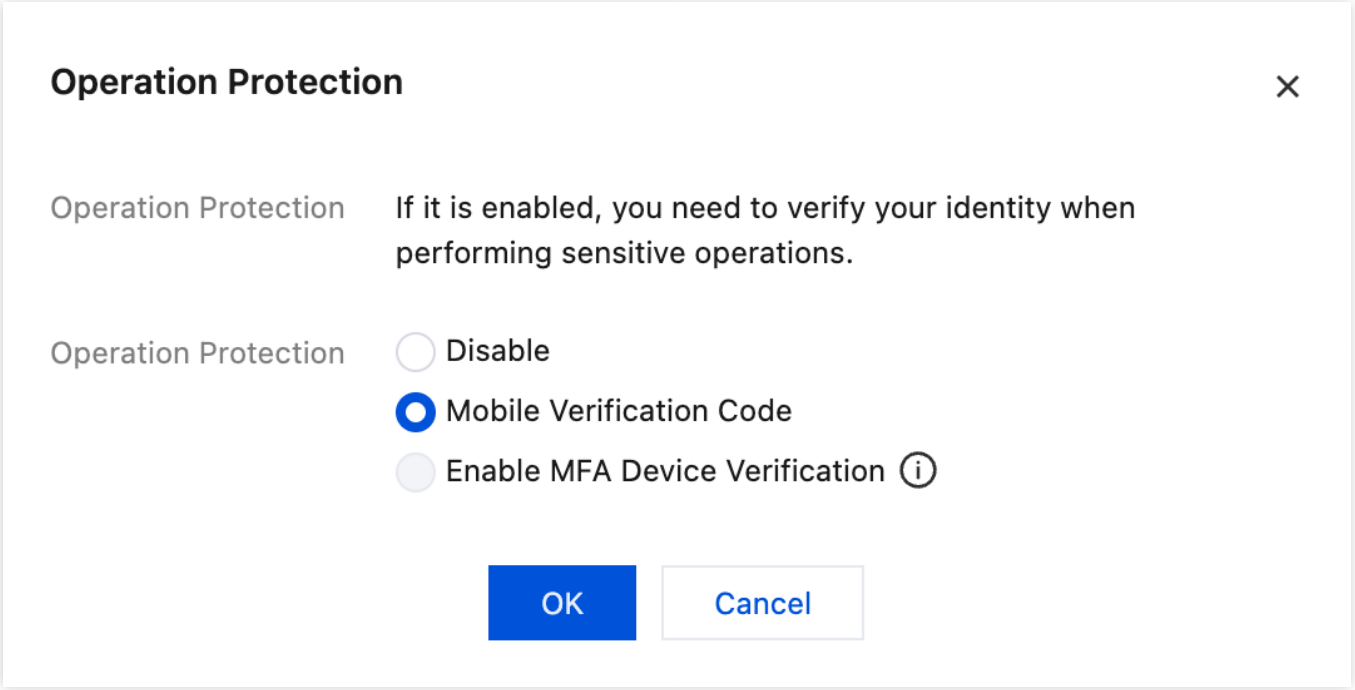
Operation Steps

Enabling Operation Protection for Sub-Users

1. Log in to the CAM console, and in the left sidebar, choose User Management > User to go to the User List page.
2. On the User List page, locate the sub-user for whom you want to enable operation protection.
3. Click the username to go to the User Details page.
4. On the User Details page, click Security Settings to go to the Security Settings page.
5. On the Security Settings page, click the edit icon next to Account Protection > Operation Protection, as shown below:



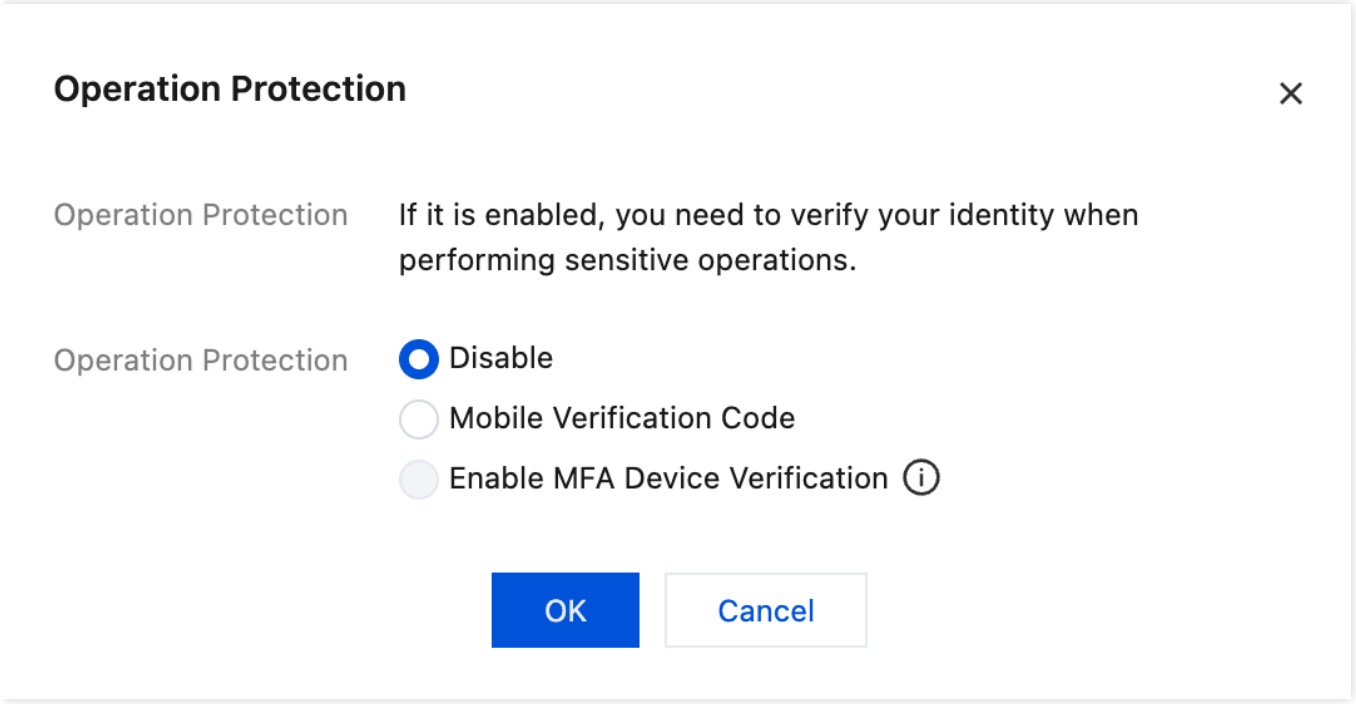
6.In the pop-up Operation Protection window, select the types of protection to enable and apply the corresponding operation protection for the current sub-user.



7.Click OK to complete the enabling of operation protection for the sub-user.

Disabling Operation Protection for Sub-Users

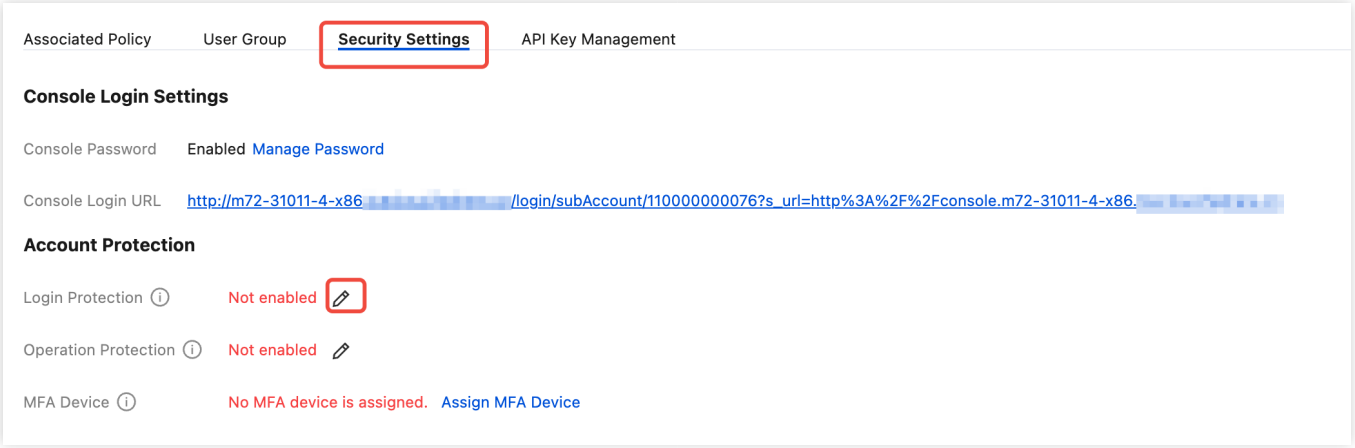
1.See steps 1 to 5 in Enabling Operation Protection to go to the Identity Security window.
2.In the pop-up Identity Security window, set the operation protection type to Disabled.



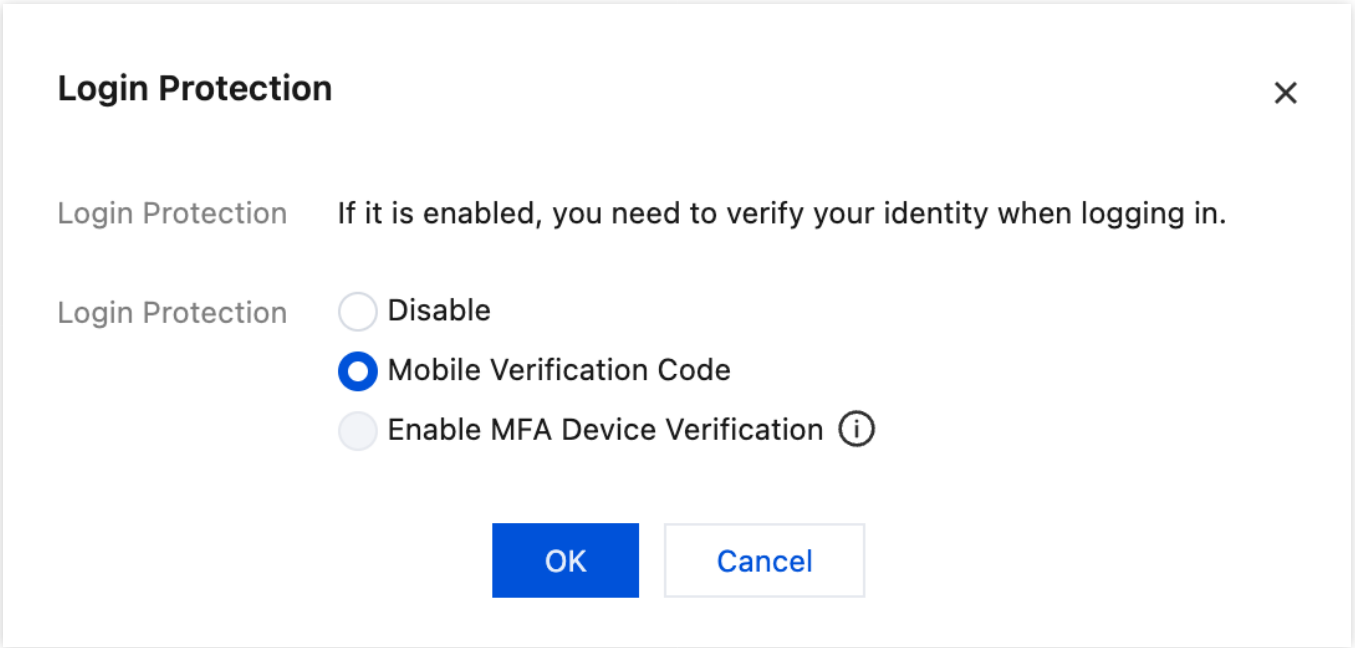
3.Click OK to complete the disabling of operation protection for the sub-user.

Enabling Login Protection for Sub-Users

1. Log in to the CAM console, and in the left sidebar, choose User Management > User to go to the User List page.
2. On the User List page, locate the sub-user for whom you want to enable login protection.
3. Click the username to go to the User Details page.
4. On the User Details page, click Security Settings to go to the Security Settings page.
5. On the Security Settings page, click the edit icon next to Account Protection > Login Protection, as shown below:



6.In the pop-up Login Protection window, select the types of protection to enable and apply the corresponding login protection for the current sub-user.

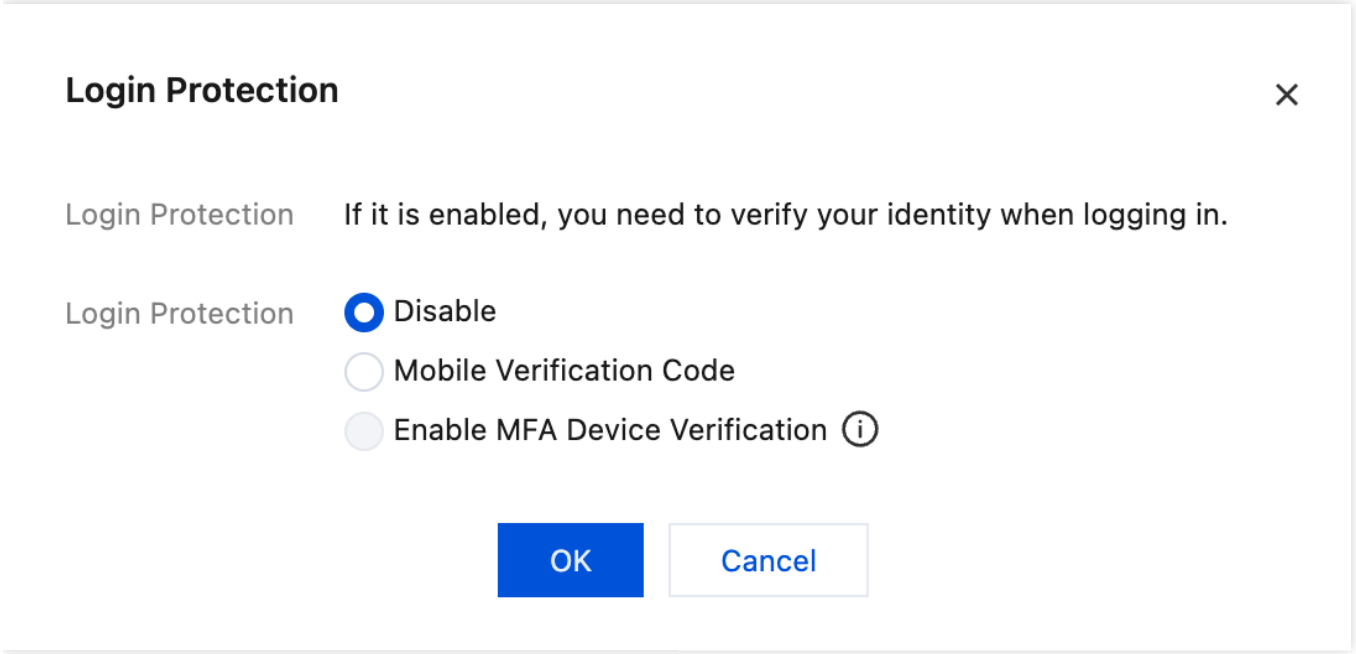


7.Click OK to complete the enabling of login protection for the sub-user.

Disabling Login Protection for Sub-Users

1.See steps 1 to 5 in Enabling Operation Protection to go to the Identity Security window.

2.In the pop-up Identity Security window, set the login protection type to Disabled.



3.Click OK to complete the disabling of login protection for the sub-user.

Subscribing Messages for Sub-users

Overview

This documentation describes how to validate message channels and set up message subscription for sub-users. In order for a sub-user to receive messages, they need to validate their message channels and subscribe to the relevant message notifications once the validation is complete.

Operation Guide

Set up Message Subscription

1. Go to the Message Center and click on Message Subscription.
2. Select message types
3. Click Add Recipient
4. In the Add Recipient window that appears, select the recipients or recipient groups that you want to subscribe to.
5. Click OK to complete the message subscription setup.

Deleting Sub-users

Overview

This document describes how to delete single or multiple sub-users. After deletion, the sub-users will no longer have management permissions for the root account.

Prerequisites

Log in to the Cloud Access Management console, select User Management > User to access the user management page.

Directions

Delete a Single Sub-user

1. On the user management page, locate the sub-user that needs to be deleted.
2. Click Delete in the right-hand operation column.
3. In the prompted window for deleting the user, confirm that the API key under the current sub-user is disabled and deleted. For details, please refer to the access keys.
4. Click Confirm Deletion to complete the deletion of a single sub-user.

Delete Multiple Sub-users

1. On the user list management page, tick the box next to the sub-users that need to be deleted.
2. Click on Delete in the upper left corner.
3. In the prompted window for deleting the user, confirm that the API key under the selected sub-users is disabled and deleted. For details, please see the access key section.
4. Click Confirm Deletion to complete the deletion of multiple sub-users.

User Information

Scenarios

This document describes how to view and modify the username, remarks, mobile phone number, and other information of sub-accounts.

Viewing User Information

1. Log in to the CAM console, choose User Management > User to enter the User Management page, and locate the sub-account for which you want to view its user information.
2. Click the username to enter the User Details page.
3. You can view the user information (including username, remarks, mobile phone number, email address, and whether WeChat notifications are allowed) at the top of the page.

Modifying User Information

1. Log in to the CAM console, choose User Management > User to enter the User Management page, and locate the sub-account for which you want to modify its user information.
2. Click the username to enter the User Details page, and click Edit in the upper right corner.
3. In the displayed window for editing information, modify the corresponding user information.
    - Modify the username of the current user. Usernames of sub-users cannot be modified because they are used for login.
    - Contact mobile number: Modify the bound mobile information for the current sub-account. This mobile number can be used to receive notifications from the root account and verify the identity before sensitive operations.
    - Contact email: Modify the bound email information for the current sub-account. This email can be used to receive notifications from the root account.
4. Click OK to complete the modification of user information. You can search for your sub-account via the modified username, mobile phone number, and email address on the User List Management page.

User Group
Creating User Groups

Creating User Groups

1. Log in to the CAM console, choose User Management > User Group to enter the User Group Management page.

2. Click Create a New User Group to enter the page for entering user group information.

3. On the page for entering user group information, enter the user group name and remarks. The user group name is required. > Note: You can search by user group name or remarks in the user group list to quickly and accurately locate the corresponding group among numerous user groups.

4. Click OK to successfully create a user group.

5. Click the name of the newly created user group to enter the User Group Information Settings page.

6. On the Associated Policies page, click Associate Policy to enter the Policy Management page.

7. Select the desired policy (multiple selections are allowed), and click OK to successfully associate the policies.

8. Under Associated Policies, you can view related settings of the user group. If there are any errors, you can click Remove to modify them.

Associated Documents

If you want to learn how to manage sub-users through user groups for group authorization, see User Management and User Group Permission Settings.
If you want to learn how to create sub-users, see Creating a Custom Sub-User.

User Management

Overview

This document describes how to add or remove individual or multiple users to/from a user group.

Prerequisites

You have logged in to the Cloud Access Management Console and selected User Management > User Groups to enter the user group page.

Directions

Add User to User Group

Adding a Single User to a User Group

1. On the user group page, locate the user group to which you want to add users.
2. Click Add User in the right-hand operation column.
3. In the pop-up window for adding users, select the users you want to add.
4. Click OK to complete the operation of adding user to the user group.

Adding Multiple Users to Multiple User Groups

1. On the user group page, select the user groups you need to add users to.
2. Click [Add User] in the top left corner.
3. In the pop-up window for adding users, select the users you want to add.
4. Click OK to complete the operation of adding user to the user group.

Remove Users from User Group

Removing a Single User from a User Group

1. On the user group page, locate the user group from which you want to remove users.
2. Click on the user group name to enter the user group details page.
3. In the user group details page, click Added Users to enter the user list page.
4. In the user list page, locate the user you want to remove, and click Remove from Group in the right-hand operation column.
5. Click Remove User to complete the operation of removing a single user from the user group.

Removing Multiple Users from a User Group

1. On the user group page, locate the user group from which you want to remove users.
2. Click on the user group name to enter the user group details page.
3. In the user group details page, click Added Users to enter the user list page.
4. In the user list page, select the users you want to remove.
5. Click Remove User > Confirm Removal to complete the operation of removing multiple users from the user group.

User Group Permission Settings

Overview

This document describes how to grant and revoke policies associated with user groups. Sub-accounts under the user group will manage resources within the granted permissions of the root account.

Prerequisites

Log in to the Cloud Access Management console, select User Management > User Groups to access the User Group Management page.

Directions

Add policies to a user group

1. On the User Group Management page, click on the user group name to access the user group details page.
2. On the user group details page, click Associated Policies to access the permission management page.
3. On the permission management page, click on Attach Policies.
4. In the pop-up box, select the policies to add (multiple selections are possible), then click OK to complete the process of adding policies to the user group.

Remove policies from a user group

1. On the User Group Management page, click on the user group name to access the user group details page.
2. On the user group details page, click Associated Policies to access the permission management page.
3. In the list, locate the policy to be removed and click Remove on the right.
4. After confirming, click Confirm Deletion to complete the process of removing policies from the user group.

Deleting a User Group

Overview

This documentation describes how to delete a user group. After deletion, sub-accounts under the user group will no longer have the permissions obtained through the user group.

Directions

Deleting a Single User Group

1. Select User Management > User Group to enter the user group management console page.
2. On the user group management console page, find the user group to be deleted.
3. Click Delete in the operation column on the right to complete the deletion of a user group.

User Settings
Password Rules

Overview

This section describes how to set the password validity period for sub-users.

Directions

> Description
>
> - The password rules set in this step only apply to sub-users who log in using passwords.
> - After the login password expires, the sub-user will be unable to log in through other means and must reset the login password.

1. Select User Management > User Settings to enter the security settings page.
2. Click the edit button of the Password Rules Settings module to modify password rules, password length, password validity period, and password repetition limit.
    - Password Rules: Must contain at least two of the following: uppercase letters, lowercase letters, numbers, and special characters.
    - Minimum Password Length: Default minimum length is 10 characters, maximum length can be set to 32 characters.
    - Password Validity Period: 0 days means no restriction, maximum can be set to 365 days.
    - Repeat Limit: Limits the repetition of new passwords with historical passwords, defaulting to non-repetition with the previous one. (Maximum of 24 non-repeated passwords)
    - Password Blocklist: Users are prohibited from setting passwords that contain strings set in the blacklist, with a maximum of 10 strings allowed.
3. Click the OK button to complete the password complexity settings. Passwords need to be reset when the validity period is reached, calculated from the last password modification.

Login Policies

This document describes how to set the session timeout duration for sub-users.

Prerequisites

Login should be conducted with a root account or a sub-account with administrator permissions.

Operation Steps

1. Choose User Management > User Settings to enter the User Settings page.
2. In the Login Policy area, click the edition button for the Session Timeout Duration module to modify the session timeout duration.
3. After the changes have been made, click OK.

Policy Management
Relevant Definitions
Permission

Permissions describe that, under certain conditions, certain actions are allowed for or denied from accessing specific resources.

By default, the root account is the resource owner and has access permissions to all resources under its name; sub-accounts have no access permissions to any resources; resource creators do not automatically have access permissions to the resources they create and require authorization from the resource owner.

A policy is a syntax rule that defines and describes one or more permissions. CAM supports two types of policies: preset policies and custom policies. Preset policies, created and managed by the cloud platform, are collections of common permissions, such as super administrators and cloud resource administrators. Such policies are read-only and not writable. Custom policies are created by users to provide a more specific description of resource management permissions. Preset policies cannot target specific resources and are coarse-grained, whereas custom policies offer the flexibility to meet users' differentiated permission management requirements.

Authorization is completed by binding one or more policies to a user or user group. The policy involved in the authorization can be either a preset policy or a custom policy.

Policy

A policy is a syntax rule used to define and describe one or more permissions. Cloud policies include preset policies and custom policies. CAM provides various methods from different perspectives for creating and managing policies. If you need to add permissions to CAM users or groups, you can directly associate preset policies or create custom policies and then associate them with CAM users or groups. Each policy can contain multiple permissions, and you can attach multiple policies to a CAM user or group.

Preset Policy

Preset policies are created and managed by the cloud. They are collections of common permissions frequently used by users, such as super administrators and full read and write permissions for resources. These policies apply to a broad range of operation objects with coarse-grained operations. As policies preset by the system, they cannot be edited by users.

Custom Policy

Custom policies are created by users to provide a more specific description of resource management permissions. These policies allow for fine-grained permission division and offer the flexibility to meet user requirements for differentiated permission management. For example, you can associate a policy with a specific database administrator to grant it the authority to manage cloud database instances but not CVM instances.

Authorization Guide
Creating Custom Policies

Scenarios

This document describes how to create a custom policy using different methods. Custom policies allow for fine-grained permission allocation, and can flexibly meet the users' requirements for differentiated permission management.

Prerequisites

The user has logged in to the CAM console and entered the Policy Management page.

Operation Steps

Creating Policies by Using Policy Generator

During policy creation by using a policy generator, services and operations are selected from the policy wizard, resources are defined, and the policy syntax is generated automatically. This method is preferred due to its simplicity and flexibility.

1. On the Policy Management page, click Create Custom Policy in the upper left corner.

2. In the pop-up window for selecting a creation method, click Create by Policy Generator to enter the page of selecting services and operations.

3. On the page of selecting services and operations, provide the following information.

    - Service (required): Select the product to be added.
    - Operation (required): Select the operations you want to authorize.
    - Resource (required): Enter the resource you want to authorize in the six-segment resource style. Cloud products with operation-level or service-level authorization granularity do not support entering specific six-segment resources, and you can simply enter "*". For the resource description method of cloud products with resource-level authorization granularity, see the Access Management Guide document for the corresponding product in CAM-supported Products (/document/product/879033/10588). For the authorization granularity supported by cloud products, see Authorization Granularity in CAM-supported Products (/document/

product/879033/10588).
- Condition (optional): Set the effective conditions for the above authorization of sub-accounts. For details, see Effective Conditions (/document/product/20472/10608).

> Note:
>
> Multiple statements can be added to a single policy.

4. Choose Add Statement > Next to enter the Policy Editing page.

5. On the Policy Editing page, add information such as the policy name and description, and then confirm the policy content. The console automatically generates the policy name and content.

> Note:
>
> - The policy name is policygen by default, with a numerical suffix generated based on the creation date. You can also customize the name as needed.
> - The policy content corresponds to the services and operations in step 3. You can modify it as needed.

6. Click Finish to complete the creation of a custom policy by using the policy generator.

Tag-based Authorization

Policies for tag-based authorization quickly grant users or user groups access to resources with a specific tag attribute.

1. On the Policy Management page, click Create Custom Policy in the upper left corner.
2. In the pop-up window for selecting the creation method, click Tag-based Authorization to enter the Tag-based Authorization page.
   - Users/User Groups: Select users and user groups you want to authorize. You can select only users or user groups.
   - For Binding Tags: (Required) Click Add, and enter a tag key and a tag value in the pop-up input box.
   - The following service resources have the corresponding operation permissions: Select service and operation to grant permissions, which are management permissions by default.
3. On the Tag-based Authorization page, select the following information and click Next to enter the Review page.
4. On the Review page, confirm the policy name and policy content, and then click Finish to complete the creation of a custom policy for tag-based authorization. The default policy name and policy content are automatically generated by the console. The policy name is policygen by default, and a numerical suffix is generated based on the creation date.

Authorization Management

Scenarios

This document describes how to associate users/user groups/roles with policies and how to associate policies with users/user groups/roles. Upon successful association, users/user groups/roles will obtain corresponding permissions through the policies.

Prerequisites

The user has logged in to the CAM console.

Operation Steps

Associating Users/User Groups/Roles with Policies

1. On the CAM console, click Policy Management on the left to enter the Policy Management page.
2. On the Policy Management page, select preset policies or custom policies as needed in Policy Type.
3. Locate the desired preset policy, and click Associate Users/User Groups/Roles in the Operation column on the right.
4. In the displayed Associate Users/User Groups/Roles dialog box, select user, user group, or role in Type as needed.
5. Select the users/user groups/roles to be associated.
6. Click OK to complete the association of users with policies.

Associating Policies with Users/User Groups

Associating Policies with Users

1. On the CAM console, click User Management on the left to enter the User Management page.
2. Select a specified user, click the username to enter the User Details page.
3. Under Associated Policies, click Associate Policy to open the Associate Policy dialog box.
4. In the Associate Policy dialog box, select the desired policy.

5. Click OK to complete the association of custom policies with users.

Associating Policies with User Groups

1. On the CAM console, click User Group on the left to enter the User Management page.
2. Find the user group to be authorized and click its name to enter the User Group Details page.
3. Under Associated Policies, click Associate Policy to open the Associate Policy dialog box.
4. In the displayed Associate Policy dialog box, select the target policy.
5. Click OK to complete the association of preset policies with user groups.

IP-based Access Restriction

Scenarios

This document describes how to restrict sub-accounts' access actions based on IP through custom policies. After successful setting, the sub-account will be allowed for or denied from managing resources under the root account based on the set IP.

Prerequisites

The involved product supports IP-based workload access restriction. For details, see the FAQs.

Operation Steps

1. Enter the Policy Management page and click Create Custom Policy in the upper left corner.
2. In the pop-up window for selecting a creation method, click Create by Policy Generator to enter the page of selecting services and operations.
3. On the page of selecting services and operations, provide the following information.

- Effect (required): Select "Allow". If you select "Deny", the user or user group cannot obtain authorization.
- Service (required): Select the product to be added.
- Action (required): Select the product permissions as needed.
- Resource (required): You can enter it by referring to the Resource Description Method.
- Condition: Select conditions as needed and enter the IP address. Multiple restrictions can be added. For example, if Allow is selected for Effect, authorization is only granted to users or groups using the specified IP address.

Use Case

The policy syntax is as follows.

```
{
    "version": "2.0",
    "statement": [
        {
```

```
            "effect": "allow",
            "action": "cos:PutObject",
            "resource": "*",
            "condition": {
                "ip_equal": {
                    "qcs:ip": [
                        "10.217.182.3/24",
                        "111.21.33.72/24"
                    ]
                }
            }
        }
    ]
}
```

Syntax Logic
Element Reference

A policy consists of several elements used to describe specific authorization information. Core elements include principal, action, resource, condition, and effect. The reserved word element can only be lowercase. There are no specific order requirements in description. For policies without specific conditional constraints, the condition element is optional. The principal element cannot be written in the console; it can only be used in policy management APIs and parameters related to policy syntax.

1.Version

Describes the syntax version of a policy. This element is required. Currently, the value can only be "2.0".

2.Principal

Describes the entity that the policy authorizes. It includes users (developers, sub-accounts, and anonymous users) and user groups. In the future, it will include more entities such as roles, federated users, and others. This element is only supported in policy syntax-related parameters in policy management APIs.

3.Statement

Describes the detailed information of one or more permissions. This element includes action, resource, condition, effect, and other related elements representing a single permission or a set of permissions. A policy should contain exactly one statement element.

4.Action

Describes the allowed or denied actions. An action can be an API (described by a name prefix) or a feature set (a set of specific APIs described by a permid prefix). This element is required.

5.Resource

Describes the detailed data to be authorized. A resource is described in a six-segment format. Detailed resource definitions vary by product. For more information on how to specify a resource, see the product documentation corresponding to the resources for which you are writing a statement. This element is required.

6.Condition

Describes the constraints for policy enforcement. A condition consists of an operator, key, and value. Condition values can include time, IP address, and other parameters. Some services allow you to specify other values in the condition. This element is optional.

7.Effect

Describes whether the result of the statement is "allow" or "deny". The possible values are allow and deny. This element is required.

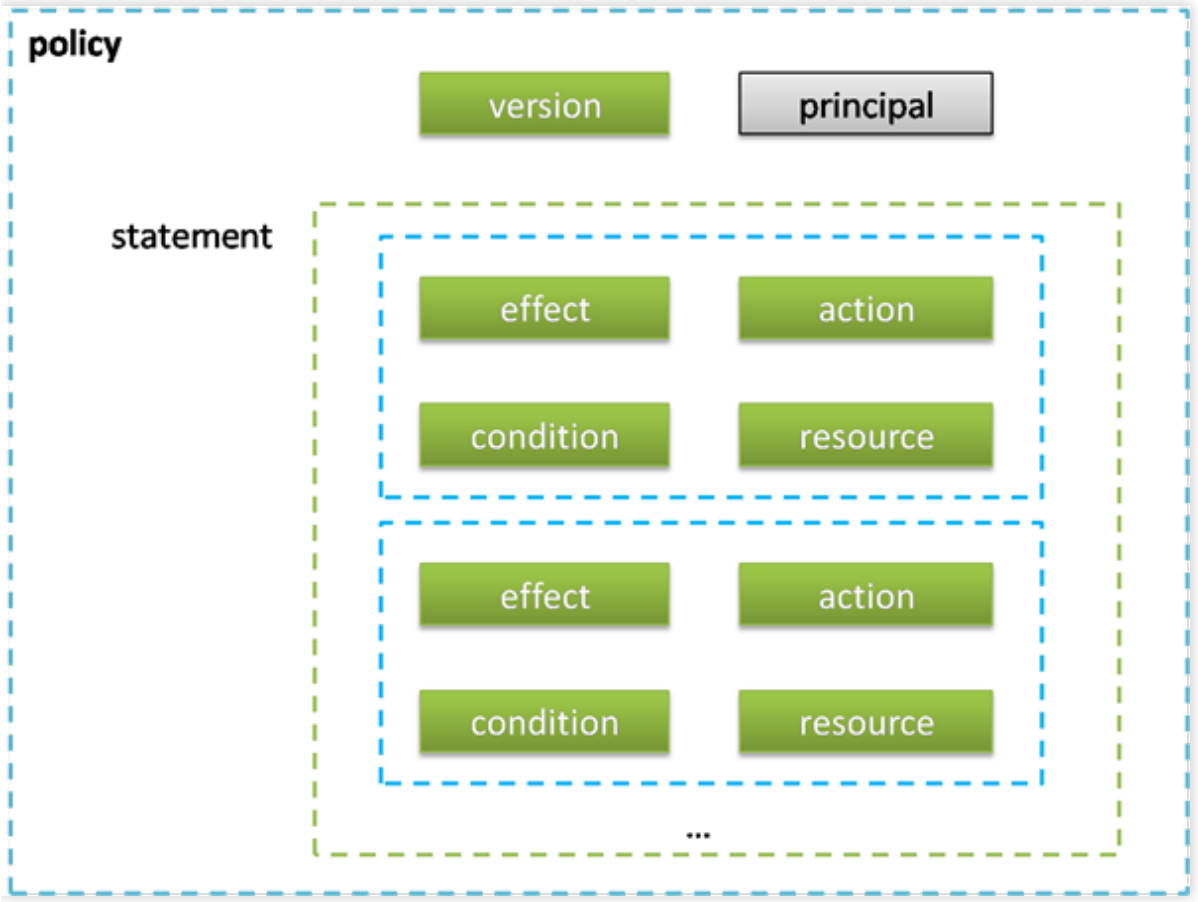
8.Policy Example

In this example, the description is as follows: The sub-account (ID: 3232523) and the group (ID: 18825) of the developer (ID: 1238423) have permissions to read and write all objects in Cloud Object Storage (COS) bucketA in region 1 and the object object2 in COS bucketB in region 2 when the access IP address is within the range of 10.121.2.*, as well as the permission to send messages.

```
{
    "version": "2.0",
    "principal": {
        "qcs": [
            "qcs::cam::uin/1238423:uin/3232523",
            "qcs::cam::uin/1238423:groupid/18825"
        ]
    },
    "statement": [
        {
            "effect": "allow",
            "action": [
                "name/cos:PutObject",
                "permid/280655"
            ],
            "resource": [
                "qcs::cos:bj:uid/1238423:prefix//1238423/bucketA/*",
                "qcs::cos:gz:uid/1238423:prefix//1238423/bucketB/object2"
            ],
            "condition": {
                "ip_equal": {
                    "qcs:ip": "10.121.2.10/24"
                }
            }
        },
        {
            "effect": "allow",
            "action": "name/cmqqueue:Sendmessages",
            "resource": "*"
        }
    ]
}
```

Syntax Structure

The syntax structure of the entire policy is shown in the figure below. A policy consists of a version and statements, and may also include principal information. The principal information is only used for policy syntax-related parameters in policy management APIs.
A statement consists of several substatements. Each substatement includes four elements: action, resource, condition (optional), and effect.



JSON Format

The policy syntax is based on the JSON format. If the created or updated policy fails to meet the JSON format, it will not be submitted successfully, so users need to ensure that the JSON format is correct. The JSON format standard is defined in RFC7159, and you can also check the policy format using an online JSON verification program.

Syntax Conventions

The syntax description has the following conventions:

- The following characters are JSON characters included in the policy syntax.

{ } [ ] " , :

- The following characters are special characters used to describe the policy syntax and are not included in a policy.

  = < > ( ) |

- When multiple values are allowed for an element, comma delimiters and ellipsis are used. Example:

  [<resource_string>, < resource_string>, ...]
  <principal_map> = { <principal_map_entry>, > <principal_map_entry>, ... }

When the policy syntax allows multiple values, it can contain just one value. In this case, the trailing comma needs to be omitted, and the square brackets "[]" are optional. Example:

"resource": [<resource_string>]
"resource": <resource_string>

- The question mark "?" after the element indicates that the element is optional. Example:

  <condition_block?>

- If the element includes enum values, they are separated by a vertical bar "|", and the range of values is defined within parentheses "()". Example:

  ("allow" | "deny")

- String elements are enclosed in double quotes. Example:

  <version_block> = "version" : "2.0"

Syntax Description

policy={
<version_block><principal_block?>,
<statement_block>

```
}<version_block>="version": "2.0"<statement_block>="statement": [
    <statement>,
    <statement>,
    ...
]<statement>={
    <effect_block>,
    <action_block>,
    <resource_block>,
    <condition_block?>
}<effect_block>="effect": ("allow"|"deny")<principal_block>="principal": ("*"|<principal_map>)<princi
pal_map>={
    <principal_map_entry>,
    <principal_map_entry>,
    ...
}<principal_map_entry>="qcs": [
    <principal_id_string>,
    <principal_id_string>,
    ...
]<action_block>="action": ("*"|[
    <action_string>,
    <action_string>,
    ...
])<resource_block>="resource": ("*"|[
    <resource_string>,
    <resource_string>,
    ...
])<condition_block>="condition": {
    <condition_map>
}<condition_map>{
    <condition_type_string>: {
        <condition_key_string>: <condition_value_list>
    },
    <condition_type_string>: {
        <condition_key_string>: <condition_value_list>
    },
    ...
}<condition_value_list>=[
    <condition_value>,
    <condition_value>,
    ...
]<condition_value>=("string"|"number")
```

Syntax description:

- A policy can contain multiple statements.

  The maximum length of a policy is 4,096 characters (excluding spaces). For details, see Restrictions.

  The display order of blocks is not restricted. For example, in a policy, a version_block can follow an

effect_block.

- The currently supported syntax version is 2.0.
- Writing the principal_block element in the console is not allowed. Principal is supported only in policy management APIs and policy syntax-related parameters.
- Both the action and resource support lists, with action also supporting permid (a set of product-defined actions).
- A condition can include a single condition or a logical combination of multiple sub-condition blocks. Each effective condition consists of the condition operator (condition_type), condition key (condition_key), and condition value (condition_value).
- The effect of each policy statement is either deny or allow. When a policy statement contains both allow and deny, the deny-override principle is applied.

String Description

The element string of syntax description is described as follows:

action_string

It consists of description scope, service type, and action name.

```
//All actions of all products
"action":"*"
"action":"*:*"
// All actions of COS products
"action":"cos:*"
// Action named GetBucketPolicy of COS products
"action":"cos:GetBucketPolicy"
// Action for matching some buckets of COS products
"action":"cos:*Bucket*"
//List of actions with action set ID 280649
"action":"permid/280649"
// Action list named GetBucketPolicy\PutBucketPolicy\DeleteBucketPolicy of COS products
"action":["cos:GetBucketPolicy","cos:PutBucketPolicy","cos: DeleteBucketPolicy"]
```

Where, permid is the action set ID defined for each product. For details, see the relevant product documentation.

resource_string

Resource is described in a six-segment format.

```
qcs: project :serviceType:region:account:resource
```

Examples are as follows:

// For object resources of a COS product, located in Luzon, with resource owner UID 10001234, resource name bucket1/object2, and resource prefix prefix, its resource description is as follows.
qcs::cos:sh:uid/10001234:prefix//10001234/bucket1/object2
// For a queue of CMQ products, located in Luzon, with resource owner UIN 12345678, resource name 12345678/queueName1, and resource prefix queueName, its resource description is as follows.
qcs::cmqqueue:sh:uin/12345678:queueName/12345678/queueName1
// For CVM of a CVM product, located in Luzon, with resource owner UIN 12345678, resource name ins-abcdefg, and resource prefix instance, its resource description is as follows.
qcs::cvm:sh:uin/12345678:instance/ins-abcdefg

For details, see the resource description method on the Supported Resource-level Permissions page of each product.

condition_type_string

Condition operators, describing the type of test conditions. Examples include string_equal, string_not_equal, date_equal, date_not_equal, ip_equal, ip_not_equal, numeric_equal, and numeric_not_equal. Here is an example:

```
"condition":{
        "string_equal":{"cvm:region":["CAX01","ACX01"]},
        "ip_equal":{"qcs:ip":"10.131.12.12/24"}
}
```

condition_key_string

Condition key, representing the application of condition operators on its value to determine if the condition is met. CAM defines a set of condition keys that can be used in all products, including qcs:current_time, qcs:ip, qcs:uin, and qcs:owner_uin. For details, see Conditions.

principal_id_string

For CAM, the user is also its resource. Hence, the principal is also described in the six-segment format. The following is an example. For details, see Resource Description Methods.

```
"principal":   {"qcs":["qcs::cam::uin/1238423:uin/3232",
            "qcs::cam::uin/1238423:groupid/13"]}
```

Evaluation Logic

CAM determines whether to allow or deny a cloud platform user's access to cloud resources based on the following evaluation logic.

1. By default, all requests are denied.
2. CAM checks all policies associated with the current user.
3. It checks whether there is a match with any policy. If yes, it proceeds to the next evaluation step; otherwise, it makes a final decision to deny access to cloud resources.
4. It checks whether there is a matching deny policy. If yes, it makes a final decision to deny access to cloud resources; otherwise, it proceeds to the next evaluation step.
5. It checks whether there is a matching allow policy. If yes, it makes a final decision to allow access to cloud resources; otherwise, it makes a final decision to deny access to cloud resources.
    - Root accounts are granted access to all resources under their name by default.
    - Certain general policies are associated with all CAM users by default. For details, see the general policy table below.
    - All other policies should be explicitly specified, including allow and deny policies.
    - The workload supporting cross-account resource access involves permission transmission scenarios where root account A grants permissions to root account B's sub-account to access root account A's resources. In this case, CAM verifies whether A grants B the permission and whether B grants its sub-account the permission. Under the premise that both conditions are met, the sub-account of B can access A's resources.
      The table of the currently supported general policies is as follows:

Policy	Definition
Multi-factor authentication (MFA) is required for the key query.	{ "principal":" *",* *"action":"account:QueryKeyBySecretId",* *"resource":""*, "condition":{"string_equal":{"mfa":"0"}} }
MFA is required for setting sensitive operations.	{ "principal":" *",* *"action":"account:SetSafeAuthFlag",* *"resource":""*, "condition":{"string_equal":{"mfa":"0"}} }
MFA is required for token binding.	{ "principal":" *",*

Policy	Definition
	"action":"account:BindToken", *"resource":""*, "condition":{"string_equal":{"mfa":"0"}} }
MFA is required for token unbinding.	{ "principal":" *",* *"action":"account:UnbindToken",* *"resource":""*, "condition":{"string_equal":{"mfa":"0"}} }
MFA is required for email address changing.	{ "principal":" *",* *"action":"account:ModifyMail",* *"resource":""*, "condition":{"string_equal":{"mfa":"0"}} }
MFA is required for mobile phone number changing.	{ "principal":" *",* *"action":"account:ModifyPhoneNum",* *"resource":""*, "condition":{"string_equal":{"mfa":"0"}} }

Resource Description Methods

The resource element describes one or more operation objects, such as CVM resources, COS buckets, and others. This document mainly introduces the resource description information of CAM.

Six-Segment Format

All resources can adopt the following six-segment description method. Each product has its own resources and corresponding resource definition details. For information on how to specify resources, see the corresponding product documentation. The six-segment definition method is as follows:

```
qcs:project_id:service_type:region:account:resource
```

Where:

- qcs is the abbreviation of qcloud service, indicating that it is the cloud resource of the cloud platform. This field is required.
- project_id describes project information and is only compatible with early CAM logic. The current policy syntax prohibits entering this information.
- service_type describes the abbreviation of the product, such as CVM and CDN. For product test details, see the corresponding product documentation. The value * indicates all products. This field is required.
- region describes the region information. If the value is empty, it represents all regions. For the cloud platform's new unified region naming method, see Region and AZ. The existing region naming method of the cloud platform is defined as follows.

Region Abbreviation	Description
ACX01	Angeles City
CAX01	Caloocan City

- account describes the root account information of the resource owner. Currently, there are two ways to describe the resource owner: UIN and UID.
- UIN refers to the QQ number of the root account and is expressed as uin/${uin} , such as uin/12345678;
- UID refers to the app ID of the root account and is expressed as uid/${appid} , such as uid/10001234.
- When the value is empty, it represents the root account to which the CAM user who created the policy belongs. Currently, the resource owners of COS and CAS workloads can only be described in UID (you can ignore this step if it is not involved). The resource owners of other workloads can only be described in UIN.

- resource describes the resource details of each product.

1. There are several description methods, and this field is required.

2. Represents the resource ID under a specific resource subclass. For example, instance/ins-abcdefg under VPC products.

   <resource_type>/<resource_id>

3. Represents the path-included resource ID under a specific resource subclass, such as prefix//10001234/bucket1/object2 under COS products. This method supports directory-level prefix matching. For example, prefix//10001234/bucket1/* indicates all objects under bucket1.

   <resource_type>/<resource_path>

4. Represents all resources under a specific resource subclass, such as instance/* .

   <resource_type>/*

5. Represents all resources under a specific product. ```

-

6. In some scenarios, the resource element can also be described as `*`. The meaning is defined as follo ws. For details, see the corresponding product documentation.
7. When the action requires association with the resource, the resource is defined as `*`, which represe nts the association with all resources.
8. When the action does not require association with the resource, the resource needs to be defined a s `*`.

CAM Resource Definition

CAM contains resources such as users, groups, and policies. The description methods of CAM resource s are as follows:

Root account:

``` bash
qcs::cam::uin/164256472:uin/164256472
```

Or

qcs::cam::uin/164256472:root

Sub-account:

qcs::cam::uin/164256472:uin/73829520

Group:

qcs::cam::uin/164256472:groupid/2340

All users:

qcs::cam::anonymous:anonymous

Or

*

Policies:

qcs::cam:: uin/12345678:policyid/*

Or

qcs::cam:: uin/12345678:policyid/12423

Important Resource Notes

- The resource owner needs to be the root account. If a sub-account creates the resource, it will not automatically have the access permission to the resource unless it obtains authorization from the resource owner.
- COS, CAS, and other workloads support cross-account authorization of resource access permissions. The authorized account can authorize resources to its sub-accounts by means of permission transfer.

Policy Variable

Use Cases

Scenario assumption: You want to grant each CAM user access to the resources they create. For example, you want to set that the creator of a COS resource has access to that resource by default.
If the resource owner (root account) authorizes each resource individually to the resource creator, the authorization cost is high, requiring writing policies for each resource type and authorizing them to the creator. In this case, you can achieve your requirement by using policy variables. Add placeholder information describing the creator in the resource definition of the policy, and this placeholder is the policy variable. During authentication, the policy variable will be replaced with context information from the request itself.

You can grant resource read permissions to the resource creator as follows.

```
{
      "version":"2.0",
      "statement":
       {
         "effect":"allow",
         "action":"name/cos:Read*",
         "resource":"qcs::cos::uid/1238423:prefix/${uin}/*"
       }
}
```

- Policy variable includes the creator's UIN in the path of each resource. For example, if a user (UIN is 12356) creates a bucket (name is test), the corresponding resource description is as follows.

qcs::cos::uid/1238423:prefix/12356/test

- When a user (UIN is 12356) accesses the resource, during authentication, the placeholders in the corresponding policy information will be replaced with the visitor.

qcs::cos::uid/1238423:prefix/12356/

- Based on the prefix of the resource (qcs::cos::uid/1238423:prefix/12356/) in the policy, you can access the resource qcs::cos::uid/1238423:prefix/12356/test .

Policy Variable Location

Resource element location: Policy variables can be used in the last segment of the six-segment resource description.
Condition element location: Policy variables can be used in condition values.

The following policy indicates that the VPC creator has access to the resource.

```
{
    "version":"2.0",
    "statement":
     {
       "effect":"allow",
       "action":"name/vpc:*",
       "resource":"qcs::vpc::uin/12357:vpc/*"
       "condition":{"string_equal":{"qcs:create_uin":"${uin}"}}
     }
}
```

Policy Variable List

The list of the currently supported policy variables is as follows:

Variable	Description
${uin}	The sub-account UIN of the current visitor. In the case where the visitor is the root account, it is the same as the root account's UIN.
${owner_uin}	The UIN of the root account to which the current visitor belongs.
${app_id}	The app ID of the root account to which the current visitor belongs.

Conditions

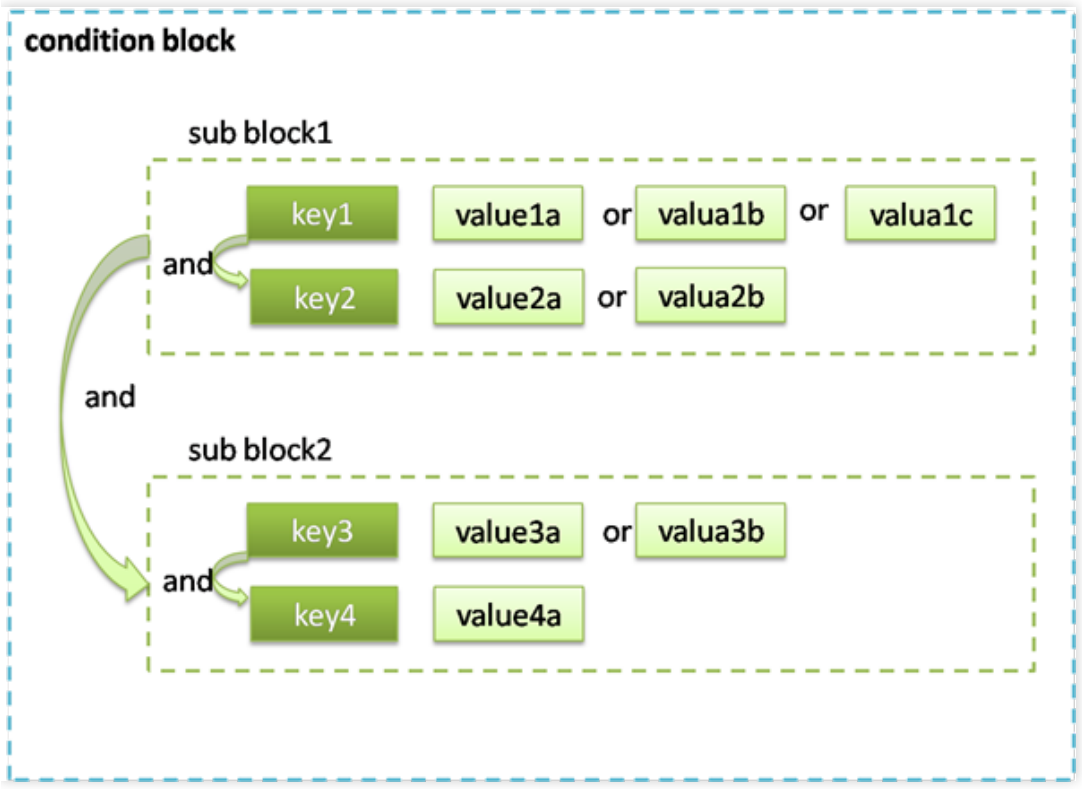
Use Cases

In many scenarios, you may need to define the conditions of the created policies further.

- Scenario 1: When CAM users call the cloud API, their access sources need to be restricted. In this case, IP conditions should be added based on existing policies.
- Scenario 2: When CAM users call the VPC peering connection API, in addition to verifying whether they have the access permission to the peering connection API and its resources, it is also necessary to confirm whether they have the access permission to the VPC associated with the peering connection.

Syntax Structure

The syntax structure of conditions is shown in the following figure. A condition block can be composed of several sub-condition blocks. Each sub-condition block corresponds to a condition operator and several condition keys, and each condition key corresponds to several condition values.



Evaluation Logic

The evaluation logic for the conditions to take effect is as follows.

1. A condition key corresponds to multiple condition values, and the condition takes effect as long as the corresponding key value in the context meets any of the condition values under the action of the associated condition operator.
2. When there are multiple condition keys in a sub-condition block, the sub-condition block takes effect only when all conditions corresponding to each condition key take effect.

3. If a condition contains multiple sub-condition blocks, the condition takes effect only when all sub-condition blocks take effect.
4. For condition operators with the "_if_exist" suffix, the condition takes effect even if the context does not contain the condition key associated with the condition operator.
5. for_all_value: a qualifier-constrained condition operator, applicable in scenarios where a condition key in the context includes multiple values. The condition takes effect only when each value of the condition key in the context takes effect under the action of the associated condition operator.
6. for_any_value: a qualifier-constrained condition operator, applicable in scenarios where a condition key in the context includes multiple values. The condition takes effect as long as any of the values of the condition key in the context take effect under the action of the associated condition operator.

Example

1. The following example indicates that the user should be within the 10.217.182.3/24 or 111.21.33.72/24 IP range to call the cloud API.

```
{
 "version": "2.0",
 "statement": {
    "effect": "allow",
    "action": "cos:PutObject",
    "resource": "*",
    "condition": {
       "ip_equal": {
          "qcs:ip": [
             "10.217.182.3/24",
             "111.21.33.72/24"
          ]
       }
    }
 }
}
```

2. The following example describes that a VPC is allowed to bind to the specified NAT gateway, and the VPC should be located in Shanghai.

```
{
 "version": "2.0",
 "statement": {
    "effect": "allow",
    "action": "name/vpc:AcceptVpcPeeringConnection",
    "resource": "qcs::vpc:sh::pcx/2341",
    "condition": {
       "string_equal_if_exist": {
          "vpc:region": "ACX01"
       }
    }
```

```
    }
  }
```

List of Condition Operators

The following table shows information about condition operators, condition names, and examples. Please refer to the corresponding product documentation for custom condition keys for each product.

Condition Operator	Description	Condition Name	Example
string_equal	String is equal to a specified value (case sensitive).	qcs:tag	{"string_equal":{"qcs:tag/tag_name1":"tag_value1"}}
string_not_equal	String is not equal to a specified value (case sensitive).	qcs:tag	{"string_not_equal":{"qcs:tag/tag_name1":"tag_value1"}}
string_equal_ignore_case	String is equal to a specified value (case insensitive).	qcs:tag	{"string_equal_ignore_case":{"qcs:tag/tag_name1":"tag_value1"}}
string_not_equal_ignore_case	String is not equal to a specified value (case insensitive).	qcs:tag	{"string_not_equal_ignore_case":{"qcs:tag/tag_name1":"tag_value1"}}
string_like	String matches a specified value (case sensitive).	qcs:tag	{"string_like":{"qcs:tag/tag_name1":"tag_value1"}}
string_not_like	String mismatches or is not equal to a specified value (case sensitive).	qcs:tag	{"string_not_like":{"qcs:tag/tag_name1":"tag_value1"}}

Condition Operator	Description	Condition Name	Example
date_not_equal	Time is not equal to a specified value.	qcs:current_time	{"date_not_equal": {"qcs:current_time":"2016-06-01T00:01:00Z"}}
date_greater_than	Time is greater than a specified value.	qcs:current_time	{" date_greater_than ": {"qcs:current_time":"2016-06-01T00:01:00Z"}}
date_greater_than_equal	Time is greater than or equal to a specified value.	qcs:current_time	{" date_greater_than_equal ": {"qcs:current_time":"2016-06-01T00:01:00Z"}}
date_less_than	Time is less than a specified value.	qcs:current_time	{" date_less_than ": {"qcs:current_time":"2016-06-01T 00:01:00Z"}}
date_less_than_equal	Time is less than or equal to a specified value.	qcs:current_time	{" date_less_than ": {"qcs:current_time":"2016-06-01T 00:01:00Z"}}
date_less_than_equal	Time is less than or equal to a specified value.	qcs:current_time	{"date_less_than_equal ": {"qcs:current_time":"2016-06-01T00:01:00Z"}}
ip_equal	IP is equal to a specified value.	qcs:ip	{"ip_equal":{"qcs:ip ":"10.121.2.10/24"}}
ip_not_equal	IP is not equal to a specified value.	qcs:ip	{"ip_not_equal":{"qcs:ip ":["10.121.2.10/24", "10.121.2.20/24"]}}
numeric_not_equal	Value is not equal to a specified value.	qcs:mfa	{" numeric_not_equal":{"mfa":1}}

Condition Operator	Description	Condition Name	Example
numeric_greater_than	Value is greater than a specified value.		{"numeric_greater_than ": {"cvm_system_disk_size":10}}
numeric_greater_than_equal	Value is greater than or equal to a specified value.		{"numeric_greater_than_equal ": {"cvm_system_disk_size":10}}
numeric_less_than	Value is less than a specified value.		{"numeric_less_than ": {"cvm_system_disk_size":10}}
numeric_less_than_equal	Value is less than or equal to a specified value.		{"numeric_less_than_equal ": {"cvm_system_disk_size":10}}
numeric_equal	Value is equal to a specified value.	qcs:mfa	{" numeric_equal":{"mfa":1}}
numeric_greater_than	Value is greater than a specified value.		{"numeric_greater_than ":{"some_key":11}}
bool_equal	Boolean matches a specified value.	-	-
null_equal	There is a condition key empty match.	-	-

Note:

1. The date format should follow the standard ISO 8601, and the UTC time should be adopted.
2. The IP format should conform to the CIDR specification.
3. The condition operator (except null_equal) with the suffix "_if_exist" indicates that the condition still takes effect

even if the context does not contain the corresponding key value.

4. for_all_value: a qualifier used with condition operators, indicating that a condition takes effect only when every value of the condition key in the context meets the requirement.
5. for_any_value: a qualifier used with condition operators, indicating that the condition takes effect if any value of the condition key in the context meets the requirement.
6. Some workloads do not support the conditions or only support partial conditions. For details, see the workload documentation.

Role Management
Role Overview

What is a Role?

A role can be considered as a virtual account within Converge Cloud. Roles can also be granted policies, possessing permissions to allow or deny actions within Converge Cloud. Roles can be assumed by any Converge Cloud account and are not exclusively associated with a bind account. Roles do not have associated persistent credentials (passwords or access tokens). The root account only needs to use persistent credentials when applying for a role. When a user takes on a role, temporary credentials are dynamically created and provided to the user for accessing the corresponding resources through temporary tokens to sign API calls to access the user's cloud resources.

Use Scenes for Roles

Cloud Account Role

Cloud account roles are used to enable resource access across tenants. For example, in certain delegated operations scenes, Tenant A can use roles to allow Tenant B to access specific resources within Tenant A.

Basic Concepts

Before using roles, you need to learn some basic terms, including role, service role, custom role, and permission policy.

Role

A virtual identity with a set of permissions, used to grant access to services, operations, and resources in the Converge Cloud to role carriers. These permissions are attached to a role, not to a specific user or user group.

CAM supports the following two types of roles:

- Service (preset) roles: roles predefined by services. After service roles obtain user authorization, services can access user resources by assuming service roles.
- Custom roles: roles defined by users. Users can freely determine role carriers and role permissions. Roles can be used by the following users:
- Root accounts that can act as roles.
- Sub-users and collaborators that can act as roles.

Service Role

Service roles, a unique type of CAM preset roles, are directly provided by various product services. The associated permissions of service roles are predefined by relevant product services. After you grant a service role to a relevant product service, the service gains full authority to act on your behalf and call other product services within the scope of the role's permissions. Service roles allow you to use services more easily because you do not need to add permissions during the role granting process manually, and you only need to decide whether to grant the service the permissions associated with the role.

In the process of granting service roles to relevant product services, the relevant permissions and role carriers for service roles have been defined. Unless otherwise defined, only the service can assume the role. The predefined elements of a service role include the role name, role carrier, and permission policy.

Cloud Account Role

Cloud account roles are CAM roles customized by users. The role name, role carrier, and role permissions of a custom role are all determined by the user. Custom roles allow you to assign access permissions to your cloud resources more freely and flexibly, with the role carrier being the root account of another tenant.

The object to which you grant a role only obtains the relevant permissions when the role is used, avoiding potential security issues associated with persistent keys.

Permission Policy

Permission documentation in JSON format. In a permission policy, you can define the operations and resources that a role can use. This documentation's rules rely on the CAM policy language rules.

Trust Policy

Permission documentation in JSON format. In a trust policy, you can define the objects that can assume roles and the prerequisites for assuming roles. This documentation's rules rely on the CAM policy language rules.

Creating Roles

Scenarios

This document explains how to create roles. Once a role is created, it can manage the resources under the root account within the granted permission scope.

Prerequisites

The user has logged in to the CAM console.

Operation Steps

1. Choose Cloud Access Management > Role > Role Management, click Create Role, enter the corresponding parameters as prompted, and click Next: Configure Role Policies.

2. In the Policy List, select the policies you want to add to the current role, and click Next: Review.

3. After confirming the role-related information, click Finish.

Modifying Roles

Scenarios

This document explains how to modify role-associated policies. Once the policies are successfully modified, the role can manage the resources under the root account within the granted permission scope according to the current settings.

Prerequisites

The user has logged in to the CAM console.

Operation Steps

Modifying Role Description

1. On the Role Management page, click the name of the role you want to modify to enter the Role Details page.
2. In the Role Information area, click  to edit the role description.
3. Click Save to update the role description.

Modifying Associated Policies

1. On the Role Management page, click the name of the role you want to modify to enter the Role Details page.
2. In the Associated Policies tab, click Associate Policy.
3. In the displayed Associate Policy dialog box, select the policies you want to add to the current role and click OK to add the role-associated policies.

Disassociating Associated Policies

1. On the Role Management page, click the name of the role you want to modify to enter the Role Details page.
2. In the Associated Policies tab, click Remove in the Operation column corresponding to the policy you want to remove.
3. In the displayed Remove Policy dialog box, click Confirm Removal. The role will no longer have the permissions described in the policy.

Deleting Role

Overview

This documentation describes how to delete a role. Once a role is deleted, it will no longer have access to the resources managed by the corresponding permission management account.

Directions

1. Log in to the Access Management (CAM) Console and go to the Role Management page.
2. Click Delete button in the Operation column corresponding to the role that you want to delete.
3. In the Delete Role dialog box that appears, click OK to delete the role. This action will also remove any policies and authorizations associated with the service role.

Authorizing Roles

1. Choose CAM > Role and access the Role Authorization list. A list of all roles that the current tenant can assume is displayed.
2. By clicking Authorize Assuming in the operation column, the administrator can specify which sub-users under the current tenant are allowed to access resources under the corresponding tenant by assuming specified roles.
3. After being authorized to assume a role, the sub-user can switch roles upon logging in to the console.
4. After clicking Switch Role, the sub-user can select the name of the role to switch to on the page. Only the roles that this sub-user has permission to switch to are listed here.

Granting Role Assumption Policy to Sub-Accounts

The root account acting as the role carrier can allow its sub-accounts to assume roles. Here, we provide an example to illustrate how to create and grant role assumption policies to sub-accounts.
Scenario assumption: Company A has an Ops engineer position and wants to outsource this position to Company B. This position requires the operation of all CVM resources in Company A's region.
You can follow the steps below:

1. Create a new policy named DevOpsPolicy under Company A's account, and associate it with APIs for Ops operations.
2. Create a role with roleName being DevOpsRole under Company A's account CompanyExampleA, set the role carrier to Company B, and associate the role with the policy DevOpsPolicy.
3. The administrator of Company B can log in to the console and authorize a sub-user user01 to assume the DevOpsRole role to access Company A. The specific operations are as follows: Under the role authorization menu, click Authorize Assuming and select the sub-user user01.
4. After logging into the console, the sub-user user01 of Company B can click the Switch Role button in the upper right corner, select the role DevOpsRole, and log into Company A's tenant by assuming the role.

Best Practices

When creating a role, you can choose the root account as the role carrier and bind authorization policies to the role. The root account acting as the carrier can grant permissions to its CAM sub-account to assume the role by creating permission policies. Subsequently, the CAM sub-account can log in to the corresponding root account console via role switching to perform operations within the authorized scope, or initiate cross-account requests through the cloud API.

Scenarios

Scenario assumption: There are two root accounts, A and B, in an enterprise. The security administrator m has a CAM sub-user a under account A, and m wants to use this sub-account to maintain the security information under account B. In this case, we can perform the following steps:

1. Create a security Ops role (role) under account B and specify the role carrier as root account A.
2. The administrator of account A logs in to the platform and authorizes m to assume role in the role list of CAM.
3. Employee m logs in as CAM sub-user a.
4. Employee m chooses Switch role in the console and logs in to the console using the security role (role).
5. Perform security Ops-related operations.
6. If employee m needs to perform security Ops-related operations on multiple root accounts at the same time, the above steps can be followed to grant employee m the corresponding security Ops permissions for the respective root accounts.

Access Keys
Viewing the Current User's Access Keys

Scenarios

This document describes how to view the API key information of the current user.

Prerequisites

The user has logged in to the CAM console and entered the cloud API key management page.

Operation Steps

The root account or sub-account can view and copy the SecretId and SecretKey information of the API keys of the current account. With SecretId and SecretKey, APIs, SDKs, or other development tools can be used to manage resources under the root account within the scope of permissions.

1. Enter the API Key Management page, and copy the SecretId from the key pair column.
2. In the key pair column, click Display to complete the authentication and obtain the SecretKey.

> Note:
>
> If your sub-account needs to self-manage API keys, grant it permissions for the CamFullAccess policy.

Troubleshooting
Creating Policy Based on Fault Report

Creating Policy Based on Fault Report

Overview

This document describes how to create a policy for troubleshooting through fault reports. After the fault is cleared, sub-accounts will manage resources under the root account within the newly set permission range.

Example

When a sub-account with QcloudCVMReadOnlyAccess policy tries to reinstall the CVM, an error message will be displayed as follows:

If you wish to authorize a sub-account to continue operations, you can create and associate a custom policy for it based on the current error information.

Directions

1. Enter the Policies page in the CAM console, click Create Definition Policy.

2. In the pop-up window for selecting a creation method, click Create by Policy Builder to enter the page of selecting service and operation.

3. On the page of selecting service and operation, add the following information,
    - Effect (required): According to the authorization effect, choose to 'Allow' or 'Deny'. In this example, select Allow.
    - Service (required): Select the product to be authorized according to its English abbreviation. In this example, for cvm in the operation of the error message, you will select Cloud Virtual Machin from the product list.
    - Operation (required): select the operation you want to authorize. In this example, it corresponds to "ResetInstance" in the error message operation.

- Resources (required): Fill in the six-segment resource of the resources you want to authorize.
- Condition (optional): Set the effective conditions for the above authorization of sub-accounts, such as access by specified IP. In this example, it does not need to be filled in.

4. Click Add Statement > Next to go to the policy editing page.

5. On the policy editing page, add the policy name and remarks and confirm the content of the policy. The policy name and content are automatically generated by the console.
   - The policy name defaults to policygen, with a numerical suffix generated based on the creation date. You can also customize this as needed.
   - The content of the policy corresponds to the services and operations in step 3. You can modify it according to your actual needs

6. Click Create Policy to complete the operation of creating a custom policy by policy builder.

7. Authorize the sub-account. After successful authorization, the sub-account will obtain corresponding permissions and remove the fault.

SMS Verification Failure

Symptom Description

When binding or changing mobile numbers, resetting passwords, etc., no verification information is received on the mobile.

Possible Causes

The main reasons for not receiving verification information on the mobile include:

- Incorrect mobile phone number and area code.
- The phone system automatically hides the content based on keywords.
- Abnormal reception caused by the mobile phone number itself, such as arrears and network failure.

Steps

1. Confirm whether the mobile phone number is correct.
    - If yes, go to the next step.
    - If no, change the mobile number.
2. Verify whether the mobile phone is out of service.
    - If yes, make a payment or change your mobile phone number.
    - If no, go to the next step.
3. Confirm whether the authentication SMS is blocked as spam.
    - If yes, disable the SMS Interception in the application software.
    - If no, go to the next step.
4. Abnormal network communication may cause SMS loss. Confirm whether the network communication is abnormal.

Email Verification Failure

Symptom Description

When binding or changing emails, resetting passwords, etc., the email does not receive verification information.

Possible Causes

The main reasons for not receiving verification information in the email include:

- The email address is incorrect.
- The email system automatically hides the content based on keywords.
- The email system has special restrictions that cause reception to fail. For example, a company's in-house email system prohibits receiving third-party emails.

Steps

1. Confirm whether the email address is correct.
    - If yes, go to the next step.
    - If no, change the email address.
2. Confirm whether the authentication information is regarded as spam and stored in the trash can.
    - Yes, set the Cloud's email to the allowlist.

- If no, go to the next step.

3. Abnormal network communication may cause email loss. Please confirm whether the network communication is abnormal.
   - If yes, obtain it again or try again later.
   - If no, go to the next step.
4. Confirm whether you are using a company's in-house email system with third-party mail restrictions.

Enterprise Authentication Login Management

Access Preparations

- The enterprise should provide relevant servers based on the authentication type.
- The container network of the cloud platform needs to have access to the network where the enterprise server is located. Please do not enable enterprise account login if the network is not properly configured, as this will prevent tenant-side users from logging in.

Access to CAS

1. Log in to the tenant console and choose Cloud Access Management > Enterprise Authentication Management.

2. Configure the relevant information required for enterprise user authentication with CAS.
    - Description of User Information Field Matching:
      The CAS server should transmit user information from the account server to the platform through the CAS verification ticket URL. The user information includes username, nickname, mobile phone number, and email address. Different servers may use varying field names for the same data. To address this, the integrating party can perform field name conversion during user information field matching. Among them, the key in json configuration represents the field recognized by the platform, and the value represents the field name returned by the server. For example, if the username field in the user information returned by the server is named "user", configure the value as user. If the server uses userName for this field, configure the value as userName.
    - Description of Account Synchronization:
      The server transmits user information to the platform. After receiving the username from the server, the platform checks whether a sub-user with the same username exists under the current tenant. If such a sub-user exists, this sub-user is used for platform login directly. If no such sub-user exists, the platform checks whether account synchronization is enabled. If this feature is enabled, the platform automatically creates a sub-user with the same username under the current tenant. If this feature is disabled, the platform prompts that no such sub-user

exists and instructs the user to contact the administrator for sub-user creation.

3. Configure the enterprise CAS login and verification URLs (the network of the URLs should be reachable from the Converge Cloud network). For detailed explanations of these URLs, see the CAS protocol documentation.

4. Enable enterprise user authentication.

5. For CAS verification of ticket URL, xml cas:serviceResponse should be returned, which should include the username, nickname, mobile phone number, and email address. Otherwise, the access will fail. The field names for username, nickname, mobile number, and email can be customized by customers and mapped in the user information field matching (see step 2).

Parameter Name	Type	Required or Not	Description
cas:user_id	String	Yes	Enterprise user login name, containing 1 to 50 characters that include English letters, digits, and special characters (_ and -). Spaces are not supported.
cas:user_name	String	Yes	Enterprise user nickname.
cas:email	String	Yes	Email, which should conform to the email format specification.
cas:phone	Int	No	Mobile phone number.
cas:country_code	String	No	Mobile country/region code.

Verification on the enterprise account server: serviceValidate.
Example of a response in XML format:

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
   <cas:authenticationSuccess>
      <cas:user>chopin1</cas:user>
         <cas:attributes>
               <cas:user_id>chopin1</cas:user_id>
               <cas:user_name>chopin1</cas:user_name>
               <cas:email>chopin1@mail.com</cas:email>
               <cas:phone>13999999999</cas:phonel>
              <cas:country_code>86</cas:country_code>
         </cas:attributes>
   </cas:authenticationSuccess>
</cas:serviceResponse>
```

6. Verify and configure network-layer communication:
   - Enter the master node of the Converge Cloud cluster, and perform the following operations on the ocloud-tcenter-mc-idplogin Pod container shell terminal:
     a. Verify whether the Pod can reach the enterprise CAS server network. Note: For the following URLs, enter the enterprise's CAS server validation address.
     b. curl -v http://cas.convergecloud.com/cas/serviceValidte
   - Verify whether the network in the above steps is reachable. If yes, the configuration is completed.
   - If the network is unreachable, verify whether containers in the ocloud-tcenter-mc-idplogin Pod can access the public network or the enterprise CAS server network. If the network is reachable but the domain name is unreachable, configure the domain name for access in kube-dns. If the network is unreachable, contact the deployment and delivery team to configure the network for proper connection with the enterprise network.
7. After the configuration is completed and enterprise user authentication login is enabled, users need to enter the root account ID when they log in to the tenant side. Currently, enterprise authentication login information is configured at the tenant granularity, meaning tenants can configure their own account server. Therefore, it is necessary to first enter the root account ID, and the platform will query the corresponding server information based on the root account ID. After the correct root account ID is entered, the platform will automatically redirect to the enterprise CAS server. If users have already logged in to the CAS server, they can directly enter the console as a logged-in user. If users have not logged in to the CAS server, the platform will open the login page on the server. Users need to enter the username and password on this login page to log in to the CAS server before entering the console as a logged-in user.

Access to OAuth

1. Log in to the Converge Cloud tenant console and click Cloud Access Management to enter the Enterprise Authentication Login Management page.

2. Configure the relevant information required for enterprise user authentication with OAuth.

3. Configure the enterprise OAuth remarks, Client ID, Client Secret, URL for verifying the authorization information via OAuth, URL for obtaining access_token, URL for obtaining the user information, and user information field matching. Enable account synchronization.

- Description of User Information Field Matching:
  The OAuth server should transmit user information from the account server to the platform through the URL for obtaining user information. The user information includes username, nickname, mobile phone number, and email address. Different servers may use varying field names for the same data. To address this, the integrating party can perform field name conversion during user information field matching. Among them, the key in the JSON configuration represents the field recognized by the platform, and the value represents the field name returned by the server. For example, if the username field in the user information returned by the server is named "user", configure the value as user. If the server uses userName for this field, configure the value as userName.
- Description of Account Synchronization:
  The server transmits user information to the platform. After receiving the username from the server, the platform checks whether a sub-user with the same username exists under the current tenant. If such sub-user exists, this sub-user is used for platform login directly. If no such sub-user exists, the platform checks whether account synchronization is enabled. If this feature is enabled, the platform automatically creates a sub-user with the same username under the current tenant. If this feature is disabled, the platform prompts that no such sub-user exists and instructs the user to contact the administrator for sub-user creation (as shown in the figure below).

4. Enable enterprise user authentication.

5. Verify and configure network-layer communication:
   - Enter the master node of the Converge Cloud cluster, and perform the following operations on the ocloud-tcenter-mc-idplogin Pod container shell terminal:
     a. Verify whether the Pod can reach the enterprise CAS server network. Note: For the following URLs, enter the enterprise's CAS server validation address.
     b. curl -v http://cas.convergecloud.com/cas/serviceValidte
   - Verify whether the network in the above steps is reachable. If yes, the configuration is completed.
   - If the network is unreachable, verify whether the ocloud-tcenter-mc-idplogin Pod container can access the public network or the enterprise CAS server network. If the network is reachable but

the domain name is unreachable, configure the domain name to be accessed in kube-dns. If the network is unreachable, contact the deployment and delivery team to configure the network for proper connectivity with the enterprise network.

6. After the configuration is completed and account synchronization is enabled, users need to enter the root account ID when they log in to the tenant side. Currently, enterprise authentication login information is configured at the tenant granularity, meaning tenants can configure their own account server. Therefore, it is necessary to first enter the root account ID, and the platform will query the corresponding server information based on the root account ID. After the correct root account ID is entered, the platform will automatically redirect to the enterprise server. If users have already logged in to the server, they can directly enter the console as a logged-in user. If users have not logged in to the server, the platform will open the login page on the server. Users need to enter the username and password on this login page to log in to the server before entering the console as a logged-in user.

Access to LDAP

1. Log in to the Converge Cloud tenant console and click Cloud Access Management to enter the Enterprise Authentication Login Management page.

2. Configure the relevant information required for enterprise user authentication with LDAP.
   For the values of Server Address, Connection Type, Base Directory, Administrator Account, Administrator Password, and Filter Conditions, see the LDAP protocol documentation.

- Description of User Information Field Matching:
  The LDAP server should transmit user information from the account server to the platform. The user information includes username, nickname, mobile phone number, and email address. Different servers may use varying field names for the same data. To address this, the integrating party can perform field name conversion during user information field matching. Among them, the key in the JSON configuration represents the field recognized by the platform, and the value represents the field name returned by the server. For example, if the username field in the user information returned by the server is named "user", configure the value as user. If the server uses userName for this field, configure the value as userName.

- Description of Account Synchronization:
  The server transmits user information to the platform. After receiving the username from the server, the platform checks whether a sub-user with the same username exists under the current tenant. If such sub-user exists, this sub-user is used for platform login directly. If no such sub-user exists, the

platform checks whether account synchronization is enabled. If this feature is enabled, the platform automatically creates a sub-user with the same username under the current tenant. If this feature is disabled, the platform prompts that no such sub-user exists and instructs the user to contact the administrator for sub-user creation (as shown in the figure below).

- LDAP Access Test
  After the configuration is completed, you can enter the server username and password to perform a test. If the test is successful, the successful information will be displayed (as shown in the figure).

  If the test fails, a failure message will be displayed based on the reason for the failure.

3. After the configuration is completed and account synchronization is enabled, users need to enter the root account ID when they log in to the tenant side. Currently, enterprise authentication login information is configured at the tenant granularity, meaning tenants can configure their own account server. Therefore, it is necessary to first enter the root account ID, and the platform will query the corresponding server information based on the root account ID.

4. After the root account ID is entered, users need to enter the username and password for the LDAP server. The platform will verify the entered username and password on the server. If the verification is successful, users can enter the console as a logged-in user. If the verification fails, they cannot enter the console.

WeCom Account

1. Joint Account

Joint accounts are used to obtain the WeCom member account information.

1.1 WeCom

1. Users need to register WeCom first and create custom applications of WeCom.
2. Obtain relevant information of WeCom and custom applications: Corpid, AgentId, and CorpSecret.
3. Log in to the cloud platform and choose Cloud Access Management > Joint Account > WeCom to enter the Management page.
4. Click Associate WeCom Account, and the dialog box for associating a WeCom account will be displayed.
5. Enter AppName, Corpid, AgentId, and CorpSecret in the dialog box.

1.2 Viewing WeCom Members

1. Log in to the cloud platform and choose Cloud Access Management > Joint Account > WeCom to enter the Management page.
2. In the WeCom information list, click Click to View to view the authorized WeCom members of the custom application.

2. User

Associate the account information of WeCom members with cloud platform users to send messages to cloud platform users via WeCom.

2.1. Associating a User Account with WeCom

1. Log in to the cloud platform and choose Cloud Access Management > User to enter the Management page.

2. Click Associate WeCom in the Operation column of the user list, and the WeCom member dialog box will be displayed.

3. Select the WeCom member to be associated in the dialog box.

4. Click OK to complete the configuration.

Best Practice
CMQ Case Studies
Authorize the sub-account to have full permissions for the message service

Authorize the sub-account to have full permissions for the message service

The enterprise account CompanyExample has a sub-account Developer, which needs to have all permissions on the message queue under the name of the enterprise account CompanyExample. No matter whether the message queue is a topic model or a queue model, it can be read and written.

Scheme A:

The enterprise account CompanyExample directly grants the sub-account Developer with the predefined policy QCloudCmqQueueFullAccess and QCloudCmqTopicFullAccess. For authorization methods, refer to Authorization Management.

Scheme B:

Step 1: Create the following policies by policy syntax.

```
{
  "version": "2.0",
  "statement":
   {
      "effect": "allow",
      "action": ["cmqtopic:*","camqueue:*"]
      "resource": "*"
   }
}
```

Step 2: Authorize the policy to the sub-account. For authorization methods, refer to Authorization Management.

Authorize Sub-account Full Permissions to Access the TDMQ it creates

Authorize Sub-account Full Permissions to Access the TDMQ it creates

The enterprise account CompanyExample has a sub-account Developer that wants access to the TDMQ it creates.

Scheme A:

The enterprise account CompanyExample directly authorizes the predefined policyQCloudCmqQueueCreaterFullAccess and QCloudCmqTopicCreatorFullAccess to the sub-account Developer. For authorization methods, refer to Authorization Management.

Scheme B:

Step 1: Create the following policies by policy syntax

```
{
    "version": "2.0",
    "statement":
    [
      {
         "effect": "allow",
         "action": "cmqtopic:*",
         "resource": "qcs::cmqtopic:::topicName/uin/${uin}/*"
      },
      {
         "effect": "allow",
         "action": "cmqqueue:*",
         "resource": "qcs::cmqqueue:::queueName/uin/${uin}/*"
      }
    ]
}
```

Step 2: Authorize the policy to the sub-account. For authorization methods, refer to Authorization Management.

Authorize sub-account read permission for a specific TDMQ

Authorize sub-account read permission for a specific TDMQ

The enterprise account CompanyExample (ownerUin: 12345678) has a TDMQ and wishes its sub-account, Developer, to have access to this message queue.

Step 1: Create the following policies by policy syntax

```
{
    "version": "2.0",
    "statement":
    {
        "action": "cmqqueue:SendMessage",
        "resource":"qcs::cmqqueue:::queueName/uin/12345678/test-caten",
        "effect": "allow"
    }
}
```

Step 2: Authorize the policy to the sub-account.

CVM Relevant Cases
Authorize Sub-account Full Permissions to CVM

Authorize Sub-account Full Permissions to CVM

The enterprise account CompanyExample (ownerUin: 12345678) has a sub-account Developer, which needs to have full administrative permissions on the CVM service of the enterprise account CompanyExample (all operation permissions such as creation, management, order placement and payment for cloud server).

Scheme A:

The enterprise account CompanyExample directly authorizes the predefined policy QcloudCVMFullAccess and QcloudCVMFinanceAccess to the sub-account Developer.

Scheme B:

Step 1: Create the following policies by policy syntax

```
{
    "version": "2.0",
    "statement":[
        {
            "effect": "allow",
            "action": "cvm:*",
            "resource": "*"
        },
        {
              "effect": "allow",
              "action": "finance:*",
              "resource": "qcs::cvm:::*"
        }
    ]
}
```

Step 2: Authorize the policy to the sub-account. For authorization methods, refer to Authorization Management.

Authorizing Sub-account Read-only Permission to CVM

Authorizing Sub-account Read-only Permission to CVM

The enterprise account CompanyExample (ownerUin: 12345678) has a sub-account Developer, which needs to have the permission to query CVM instances of the CVM service of the enterprise account CompanyExample, but does not have the permission to create, delete, and power on/off.

Scheme A:

The enterprise account CompanyExample directly authorizes the predefined policy QcloudCVMInnerReadOnlyAccess to the sub-account Developer.

Scheme B:

Step 1: Create the following policies by policy syntax

```
{
  "version": "2.0",
  "statement":
   {
      "effect": "allow",
      "action": [
            "cvm:Describe*",
            "cvm:Inquiry*"
            ],
      "resource": "*"
   }
}
```

Step 2: Authorize the policy to the sub-account.

Authorize sub-accounts to have read-only permission of CVM related resources

Authorize sub-accounts to have read-only permission of CVM related resources

The enterprise account CompanyExample (ownerUin: 12345678) has a sub-account Developer. This sub-account needs to have the permission to query CVM instances and related resources (VPC, Cloud Load Balancer) of the CVM service of the enterprise account CompanyExample, but does not have the permission to create, delete, or power on/off.

Scheme A:

The enterprise account CompanyExample directly authorizes the predefined policy QcloudCVMReadOnlyAccess to its sub-account Developer.

Scheme B:

Step 1: Create the following policies by policy syntax

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:Describe*",
                "cvm:Inquiry*"
            ],
            "resource": "*",
            "effect": "allow"
        },
        {
            "action": [
                "vpc:Describe*",
                "vpc:Inquiry*",
                "vpc:Get*"
            ],
            "resource": "*",
            "effect": "allow"
        },
        {
            "action": [
                "clb:Describe*"
            ],
            "resource": "*",
            "effect": "allow"
```

```
        },
        {
            "effect": "allow",
            "action": "monitor:*",
            "resource": "*"
        }
    ]
}
```

Step 2: Authorize the policy to the sub-account.

Authorize Sub-account Permission to Perform Operations on CBSs

Authorize Sub-account Permission to Perform Operations on CBSs

The enterprise account CompanyExample (ownerUin: 12345678) has a sub-account Developer, which needs permissions to view CBS information in the CVM Console, create CBS, and use CBS under the CompanyExample enterprise account.

Scheme A:

The enterprise account CompanyExample directly authorizes the predefined policy QcloudCDCFullAccess to the sub-account Developer.

Step 1: Create the following policies by policy syntax

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:CreateCbsStorages",
                "cvm:AttachCbsStorages",
                "cvm:DetachCbsStorages",
                "cvm:ModifyCbsStorageAttributes",
                "cvm:DescribeCbsStorages",
                "cvm:DescribeInstancesCbsNum",
                "cvm:RenewCbsStorage",
                "cvm:ResizeCbsStorage"
                ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

Step 2: Authorize the policy to the sub-account.

Note: If you do not allow the sub-account to modify CBS attributes, please remove "cvm:ModifyCbsStorageAttributes" from the above policy syntax.

Authorize Sub-account Permission to Perform Operations on Security Groups

Authorize Sub-account Permission to Perform Operations on Security Groups

The enterprise account CompanyExample (ownerUin: 12345678) has a sub-account Developer, which needs to have the permission to view and use security groups in the CVM console for the enterprise account CompanyExample.

The following policies allow sub-accounts to have the permission to create and delete security groups in the CVM console.

Step 1: Create the following policies by policy syntax

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:DeleteSecurityGroup",
                "cvm:CreateSecurityGroup"
            ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

Step 2: Authorize the policy to the sub-account

The following policies allow sub-accounts to create, delete and modify security group policies in the CVM console.

Step 1: Create the following policies by policy syntax

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:ModifySecurityGroupPolicy",
                "cvm:CreateSecurityGroupPolicy",
                "cvm:DeleteSecurityGroupPolicy"
            ],
            "resource": "*",
```

```
            "effect": "allow"
        }
    ]
}
```

Step 2: Authorize the policy to the sub-account.

Authorizing a Sub Account to Perform Operations on EIP Addresses

Authorizing a Sub Account to Perform Operations on EIP Addresses

The enterprise account CompanyExample (ownerUin: 12345678) has a sub account Developer, which needs to have the permissions to view and use the EIP addresses in the CVM console for the CVM service of the enterprise account CompanyExample.

Step 1: Create the following policy by policy syntax.

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:AllocateAddresses",
                "cvm:AssociateAddress",
                "cvm:DescribeAddresses",
                "cvm:DisassociateAddress",
                "cvm:ModifyAddressAttribute",
                "cvm:ReleaseAddresses"
            ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

Step 2: Grant the policy to the sub account.

The following policy allows the sub account to view EIP addresses and assign them to instances and associate them with each other. The sub account can modify the attributes of EIP addresses, cancel their association, or release them.

Step 1: Create the following policy by policy syntax.

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:DescribeAddresses",
                "cvm:AllocateAddresses",
                "cvm:AssociateAddress"
```

```
            ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

Step 2: Grant the policy to the sub account. /130622231683297280/130622242107191296).

Authorize sub-accounts to have the operation permission CVM

Authorize sub-accounts to have the operation permission of a specific CVM

The enterprise account CompanyExample (ownerUin: 12345678) has a sub-account Developer, which needs to have the operation permissions on the specified CVM machine (ID: ins-1, Luzon region) of the enterprise account CompanyExample.

Step 1: Create the following policies by policy syntax

```
{
    "version": "2.0",
    "statement": [
        {
            "action": "cvm:*",
            "resource": "qcs::cvm:gz::instance/ins-1",
            "effect": "allow"
        }
    ]
}
```

Step 2: Authorize the policy to the sub-account.

Authorize sub-accounts to have the operation permission of CVM i

Authorize sub-accounts to have the operation permission of CVM in a specific region

The enterprise account CompanyExample (ownerUin: 12345678) has a sub-account Developer, which needs to have the operation permissions on all machines in Luzon of the enterprise account CompanyExample.

Step 1: The enterprise account CompanyExample directly authorizes the predefined policy QcloudCVMReadOnlyAccess to the sub-account Developer.

Step 2: Create the following policies by policy syntax

```
{
    "version": "2.0",
    "statement": [
        {
            "action": "cvm:*",
            "resource": "qcs::cvm:CAX01::*",
            "effect": "allow"
        }
    ]
}
```

Step 2: Authorize the policy to the sub-account.

Authorize the sub-account to have all the permissions of CVM, but not including payment permissions

Authorize the sub-account to have all the permissions of CVM, but not including payment permissions

The enterprise account CompanyExample (ownerUin: 12345678) has a sub-account Developer. This sub-account needs to have all permissions (creation, management and other operations) for the CVM service of the enterprise account CompanyExample, excluding payment permission. It can place an order but cannot pay.

Scheme A:

The enterprise account CompanyExample directly authorizes the predefined policy QcloudCDCFullAccess to the sub-account Developer.

Scheme B:

Step 1: Create the following policies by policy syntax

```
{
  "version": "2.0",
  "statement":[
      {
          "effect": "allow",
          "action": "cvm:*",
          "resource": "*"
      }
  ]
}
```

Step 2: Authorize the policy to the sub-account.

Grant Sub-account View Permissions for Unsupported Project Products

Case Scenario

The enterprise account CompanyExample (ownerUin: 12345678) has a sub-account Developer, which needs to be authorized to view products such as snapshots, images, VPC and Elastic IP in the console.

Operation Guide

Grant Sub-account View Permissions for Unsupported Project Products

API Documentation
Cloud Access Management(cam) version(2019-01-16)
API overview

API version

V3

Identity Provider APIs.

Interface name	Interface function
CreateOauthProvider	CreateOauthProvider
GetUserAccessToken	GetUserAccessToken
RefreshUserToken	RefreshUserToken
UpdateOauthProvider	UpdateOauthProvider
VerifyUserAccessToken	VerifyUserAccessToken

Other APIs

Interface name	Interface function
AttachRolePolicies	AttachRolePolicies
AttachRolePolicy	AttachRolePolicy
AttachRolesPolicy	AttachRolesPolicy
CreatePolicy	CreatePolicy
CreateRole	CreateRole
DeletePolicy	DeletePolicy

Interface name	Interface function
DeleteRole	DeleteRole
DescribeRoleList	DescribeRoleList
DetachGroupPolicies	DetachGroupPolicies
DetachGroupsPolicy	DetachGroupsPolicy
DetachUsersPolicy	DetachUsersPolicy
GetPolicy	GetPolicy
GetRole	GetRole
GetServiceRoleInfo	GetServiceRoleInfo
ListAttachedGroupPolicies	ListAttachedGroupPolicies
ListAttachedRolePolicies	ListAttachedRolePolicies
ListEntitiesForPolicy	ListEntitiesForPolicy
ListPolicies	ListPolicies
UpdateAssumeRolePolicy	UpdateAssumeRolePolicy
UpdatePolicy	UpdatePolicy

User APIs

Interface name	Interface function
GetPasswordRules	GetPasswordRules
GetSubsGroup	GetSubsGroup
GetUinBySecretId	GetUinBySecretId
UpdatePasswordRules	UpdatePasswordRules

CallingMethod signaturev1

Converge Cloud authenticates each access request, i.e. each request needs to include authentication information (Signature) in the common parameters to verify the identity of the requester.
The Signature is generated by the security credentials which include SecretId and SecretKey. If you don't have the security credentials yet, go to the API Key page to apply for them; otherwise, you cannot invoke the API.

1. Applying for Security Credentials

Before using the API for the first time, go to the API Key page to apply for security credentials.
Security credentials consist of SecretId and SecretKey:

- SecretId is used to identify the API requester.
- SecretKey is used to encrypt the signature string and verify it on the server.
- You must keep your security credentials private and avoid disclosure.

You can apply for the security credentials through the following steps:

1. Log in to the Converge Cloud Console.
2. Go to the Cloud API Key page.
3. On the API Key Management page, click Create Key to create a SecretId/SecretKey pair.

Note: Each account can have up to two pairs of SecretId/SecretKey.

2. Generating a Signature

With the SecretId and SecretKey, a signature can be generated. The following describes how to generate a signature:

Assume that the SecretId and SecretKey are:

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

Note: This is just an example. For actual operations, please use your own SecretId and SecretKey.

Take the Cloud Virtual Machine's request to view the instance list (DescribeInstances) as an example. When

you invoke this API, the request parameters may be as follows:

Parameter name	Description	Parameter value
Action	Method name	DescribeInstances
SecretId	Key ID	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	Current timestamp	1465185768
Nonce	Random positive integer	11886
Region	Region where the instance is located	luzon
InstanceIds.0	ID of the instance to query	ins-09dx96dg
Offset	Offset	0
Limit	Allowed maximum output	20
Version	API version number	2017-03-12

2.1. Sorting Parameters

First, sort all the request parameters in an ascending lexicographical order (ASCII code) by their names. Notes: (1) Parameters are sorted by their names instead of their values; (2) The parameters are sorted based on ASCII code, not in an alphabetical order or by values. For example, InstanceIds.2 should be arranged after InstanceIds.12. You can complete the sorting process using a sorting function in a programming language, such as the ksort function in PHP. The parameters in the example are sorted as follows:

```
{
    'Action' : 'DescribeInstances',
    'InstanceIds.0' : 'ins-09dx96dg',
    'Limit' : 20,
    'Nonce' : 11886,
    'Offset' : 0,
    'Region' : 'luzon',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    'Timestamp' : 1465185768,
    'Version': '2017-03-12',
}
```

When developing in another programming language, you can sort these sample parameters and it will work as long as you obtain the same results.

2.2. Concatenating a Request String

This step generates a request string.
Format the request parameters sorted in the previous step into the form of "parameter name"="parameter value". For example, for the Action parameter, its parameter name is "Action" and its parameter value is "DescribeInstances", so it will become Action=DescribeInstances after formatted.
Note: The "parameter value" is the original value but not the value after URL encoding.

Then, concatenate the formatted parameters with "&". The resulting request string is as follows:

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.3. Concatenating the Signature Original String

This step generates a signature original string.
The signature original string consists of the following parameters:

1. HTTP method: POST and GET modes are supported, and GET is used here for the request. Please note that the method name should be in all capital letters.
2. Request host: the domain name of the request to view the list of instances (DescribeInstances) is cvm.convergecloud.com. The actual request domain name varies by the module to which the API belongs. For more information, see the instructions of the specific API.
3. Request path: The request path in the current version of Cloud API is fixed to / .
4. Request string: the request string generated in the previous step.

The concatenation rule of the signature original string is: Request method + request host + request path + ? + request string

The concatenation result of the example is:

```
GETcvm.convergecloud.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.4. Generating a Signature String

This step generates a signature string.
First, use the HMAC-SHA1 algorithm to sign thesignature original stringobtained in the previous step, and then encode the generated signature using Base64 to obtain the final signature.

The specific code is as follows with the PHP language being used as an example:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.convergecloud.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit
=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPL
E&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

The final signature is:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

When developing in another programming language, you can sign and verify the original in the example above and it works as long as you get the same results.

3. Encoding a Signature String

The generated signature string cannot be directly used as a request parameter and must be URL encoded.

For example, if the signature string generated in the previous step is EliP9YW3pW28FpsEdkXt/+WcGeI= , the final signature string request parameter (Signature) is EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%3d, which will be used to generate the final request URL.

**Note: If your request method is GET, or the request method is POST and the Content-Type is application/x-www-form-urlencoded, then all the request parameter values need to be URL encoded (except the parameter key and the symbol of =) when sending the request. Non-ASCII characters need to be encoded with UTF-8 before URL encoding. **

Note: The network libraries of some programming languages automatically URL encode all parameters, in which case there is no need to URL encode the signature string; otherwise, two rounds of URL encoding will cause the signature to fail.

Note: Other parameter values also need to be encoded using RFC 3986. Use %XY in percent-encoding for special characters such as Chinese characters, where "X" and "Y" are hexadecimal characters (0-9 and uppercase A-F), and using lowercase will cause an error.

4. Signature Failure

The following situational error codes for signature failure may occur. Please resolve the errors accordingly.

Error code	Error description
AuthFailure.SignatureExpire	The signature is expired

Error code	Error description
AuthFailure.SecretIdNotFound	The key does not exist
AuthFailure.SignatureFailure	Signature error
AuthFailure.TokenFailure	Token error
AuthFailure.InvalidSecretId	Invalid key (not a Cloud API key type)

5. Signature Demo

When calling API 3.0, you are recommended to use the corresponding Converge Cloud SDK 3.0, Swhich encapsulates the signature process, enabling you to focus on only the specific APIs provided by the product when developing. See SDK Center for more information. Currently, the following programming languages are supported:

- Python
- Java
- PHP
- Go
- Node

To further explain the signing process, we will use a programming language to implement the process described above. The request domain name, API and parameter values in the sample are used here. This goal of this example is only to provide additional clarification for the signature process, please see the SDK for actual usage.

The final output URL might be: https://cvm.convergecloud.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

Note: The key in the example is fictitious, and the timestamp is not the current time of the system, so if this URL is opened in the browser or called using commands such as curl, an authentication error will be returned: Signature expired. In order to get a URL that can work properly, you need to replace the SecretId and SecretKey in the example with your real credentials and use the current time of the system as the Timestamp.

Note: In the example below, even if you use the same programming language, the order of the parameters in the URL may be different for each execution. However, the order does not matter, as long as all the parameters are included in the URL and the signature is calculated correctly.

Note: The following code is only applicable to API 3.0. It cannot be directly used in other signature processes. Even with an older API, signature calculation errors may occur due to the differences in details. Please refer to the corresponding documentation.

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class CloudAPIDemo {
    private final static String CHARSET = "UTF-8";

    public static String sign(String s, String key, String method) throws Exception {
        Mac mac = Mac.getInstance(method);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
        mac.init(secretKeySpec);
        byte[] hash = mac.doFinal(s.getBytes(CHARSET));
        return DatatypeConverter.printBase64Binary(hash);
    }

    public static String getStringToSign(TreeMap<String, Object> params) {
        StringBuilder s2s = new StringBuilder("GETcvm.convergecloud.com/?");
        // When signing, the parameters need to be sorted in lexicographical order. TreeMap is used here
to guarantee the correct order.
        for (String k : params.keySet()) {
            s2s.append(k).append("=").append(params.get(k).toString()).append("&");
        }
        return s2s.toString().substring(0, s2s.length() - 1);
    }

    public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException
{
        StringBuilder url = new StringBuilder("https://cvm.convergecloud.com/?");
        // There is no requirement for the order of the parameters in the actual request URL.
        for (String k : params.keySet()) {
            // The request string needs to be URL encoded. As the Key is all in English letters, only the valu
e is URL encoded here.
            url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).app
end("&");
        }
        return url.toString().substring(0, url.length() - 1);
```

```
    }

    public static void main(String[] args) throws Exception {
        TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap enables automati
c sorting
        // A random number should be used when actually calling, for example: params.put("Nonce", new
Random().nextInt(java.lang.Integer.MAX_VALUE));
        params.put("Nonce", 11886); // Common parameter
        // The current time of the system should be used when actually calling, for example:   params.put
("Timestamp", System.currentTimeMillis() / 1000);
        params.put("Timestamp", 1465185768); // Common parameter
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // Common parameter
        params.put("Action", "DescribeInstances"); // Common parameter
        params.put("Version", "2017-03-12"); // Common parameter
        params.put("Region", "luzon"); // Common parameter
        params.put("Limit", 20); // Business parameter
        params.put("Offset", 0); // Business parameter
        params.put("InstanceIds.0", "ins-09dx96dg"); // Business parameter
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // Common parameter
        System.out.println(getUrl(params));
    }
}
```

Python

Note: If running in a Python 2 environment, the following requests dependency package must be installed first: pip install requests .

```
# -*- coding: utf8 -*-
import base64
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/?"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
```

```
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.convergecloud.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'luzon',
        'SecretId' : secret_id,
        'Timestamp' : 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # n actual invocation would occur here, which may incur fees after success
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

signaturev3

Converge Cloud The API will authenticate each access request, meaning that each request must include signature information (Signature) in the public request parameters to verify the requester's identity.
The signature information is generated by the security credentials, which include SecretId and SecretKey . If the user does not have security credentials, please apply for them on the Cloud API Key page, otherwise they cannot call the Cloud API interfaces.

1. Apply for security credentials.

Before using the Cloud API for the first time, please go to the Cloud API Key page to apply for a security credential.
Security credentials include SecretId and SecretKey:

- SecretId The security credentials are used to identify the identity of the API caller.
- SecretKey The secret key is used to encrypt and sign the string for server-side verification of the signature.
- Users must strictly protect their security credentials and avoid disclosure them.

The specific steps to apply for security credentials are as follows:

1. Log in to Converge Cloud Administration Center Console
2. Go to the console page for cloud API keys
3. On the Cloud API Key page, click New to create a pair of SecretId/SecretKey

Note: Developer accounts can have up to two pairs SecretId / SecretKey.

2. TC3-HMAC-SHA256 signature method

Note: For GET method, only Content-Type: application/x-www-form-urlencoded protocol format is supported. For POST method, Content-Type: application/json and Content-Type: multipart/form-data are currently supported. By default, all business interfaces are supported in json format. Only specific business interfaces are supported in multipart format. In this case, the interface cannot be called in json format. Refer to the specific business interface document description.

The following takes Cloud Virtual Machine querying Region-1 instance list as an example to introduce the signature calculation process step by step. We only use two parameters for the query instance list: Limit and Offset, called using the GET method.

Assume that the SecretId and SecretKey of the user are AKIDz 8krbsJ 5yKBZQpn 74WFkmLPx 3EXAMPLE and Gu 5t9xGARNpq 86cd 98joQYCN3EXAMPLE, respectively.

2.1. splice specification request string

Concatenate CanonicalRequest strings in the following format:

```
CanonicalRequest =
  HTTPRequestMethod + '\n' +
  CanonicalURI + '\n' +
  CanonicalQueryString + '\n' +
  CanonicalHeaders + '\n' +
  SignedHeaders + '\n' +
  HashedRequestPayload
```

- HTTPRequestMethod: HTTP request methods (GET, POST), GET in this example;
- CanonicalURI: URI parameter, API 3.0 fixed to forward slash (/);
- CanonicalQueryString: Query string in the URL that initiates the HTTP request. For POST requests, it is fixed as an empty string. For GET requests, it is the question mark (?) in the URL. The following string content, this example value is: Limit=10&Offset=0. Note: CanonicalQueryString needs to be URL encoded.
- CanonicalHeaders: The header information of participating signature, at least including host and content-type, can also be added with customized header to improve the uniqueness and security of its own request. Splicing rules: 1) head key and value are uniformly converted to lowercase, and the first and last spaces are removed, and spliced according to the format of key:value\n;2) multiple heads are spliced according to the dictionary sorting of head key (lowercase). In this example: content-type: application/x-www-form-urlencoded\nhost: cvm.convergecloud.com\n
- SignedHeaders: The header information involved in the signature indicates which headers participate in the signature of this request, and the header content contained in CanonicalHeaders is one-to-one corresponding. content-type and host are required headers. constructed as: 1) The head key is uniformly converted to lowercase; 2) Multiple header keys (lowercase) are concatenated according to dictionary sorting and separated by semicolons (;). In this case: content-type;host
- HashedRequestPayload: The hash value of the request body, calculated as Lowercase(HexEncode(Hash.SHA256(RequestPayload))), SHA256 hash the entire payload of the HTTP request body, then hexadecimal encoding, and finally converting the encoding string to lowercase letters. Note: For GET requests, RequestPayload is fixed as an empty string. For POST requests, RequestPayload is the payload of the HTTP request body.

According to the above rules, the resulting canonical request string in the example is as follows (for clarity, \nnewlines are replaced by a new line printed separately):

```
GET
```

```
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.convergecloud.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. Concatenate the string to be signed

Concatenate the string to be signed in the following format:

```
StringToSign =
  Algorithm + \n +
  RequestTimestamp + \n +
  CredentialScope + \n +
  HashedCanonicalRequest
```

- Algorithm: Signature algorithm, currently fixed as TC3-HMAC-SHA256;
- RequestTimestamp: Request timestamp, i.e. the X-TC-Timestamp value of the request header. The request in the example above is 1539084154;
- CredentialScope: Credential range in the format Date/service/tc3_request, containing date, requested service and termination string (tc3_request). **Date is the date of UTC standard time, the value needs to be consistent with UTC standard time converted by common parameter X-TC-Timestamp **;service is the product name, which must be consistent with the product domain name called, such as cvm. The above example request, the value is 2018-10-09/cvm/tc3_request;
- HashedCanonicalRequest: The hash value of the canonical request string obtained by concatenating the above steps is calculated by Lowercase(HexEncode(Hash.SHA256(CanonicalRequest))).

note:

> 1. Date must be calculated from timestamp X-TC-Timestamp and time zone UTC+0. Adding local time zone information to the system, such as East Zone 8, will cause calls to succeed during the day and night, but calls to fail in the early hours of the morning. Assuming the timestamp is 1551113065, the time in the East Eight District is 2019-02-26 00:44:25, but the calculated Date with UTC+0 should be 2019-02-25 instead of 2019-02-26.
> 2. Timestamp must be the current system time and ensure that the system time and standard time are synchronized. If the difference exceeds five minutes, it will fail. If it is not synchronized with the standard time for a long time, it may cause the request to fail after running for a period of time (return signature expiration error).

According to the above rules, the string to be signed in the example is as follows (for clarity,\nnewlines are

replaced by new lines printed):

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. calculate the signature

1. Compute the derived signature key, pseudocode as follows

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey: Original SecretKey;
- Date: That is, the Date field information in Credential, as shown in the example above, is 2018-10-09.;
- Service: That is, the Service field information in Credential, as shown in the example above, is cvm;

2. Calculate the signature, pseudocode as follows

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

- SecretSigning: That is, the derivative signature key calculated above;
- StringToSign: That is, the character string to be signed calculated in step 2;

2.4. Splicing Authorization

Splice as follows Authorization:

```
Authorization =
    Algorithm + ' ' +
    'Credential=' + SecretId + '/' + CredentialScope + ', ' +
    'SignedHeaders=' + SignedHeaders + ', '
    'Signature=' + Signature
```

- Algorithm: Signature method, fixed as TC3-HMAC-SHA256;
- SecretId: SecretId in key pair;
- CredentialScope: See above, Scope of Voucher;
- SignedHeaders: See above, header information for participating signatures;
- Signature: signature value

According to the above rules, the values obtained in the example are:

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

The final complete invocation information is as follows:

```
https://cvm.convergecloud.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.convergecloud.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: luzon
```

3. signing failed

According to the actual situation, there are the following signature failure error codes, please handle according to the actual situation

error code	error description
AuthFailure.SignatureExpire	signature expiration
AuthFailure.SecretIdNotFound	key is not present
AuthFailure.SignatureFailure	signature error
AuthFailure.TokenFailure	token error
AuthFailure.InvalidSecretId	Illegal key (not cloud API key type)

4. Signature Demo

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
```

```
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.convergecloud.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(byte[] key, String msg) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA256");
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
        mac.init(secretKeySpec);
        return mac.doFinal(msg.getBytes(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.convergecloud.com";
        String region = "luzon";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        // Pay attention to the time zone, otherwise it is easy to make mistakes.
        sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
        String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

        // ************* Step 1: Assemble the specification request string *************
```

```
        String httpRequestMethod = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\n"
                + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
        System.out.println(canonicalRequest);

        // ************* Step 2: Stitching the string to be signed *************
        String credentialScope = date + "/" + service + "/" + "tc3_request";
        String hashedCanonicalRequest = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* Step 3: Calculate the signature *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        System.out.println(signature);

        // ************* Step 4: Splicing Authorization *************
        String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
                + "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
        System.out.println(authorization);

        TreeMap<String, String> headers = new TreeMap<String, String>();
        headers.put("Authorization", authorization);
        headers.put("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

```
# -*- coding: utf-8 -*-
```

```
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# key parameters
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.convergecloud.com"
endpoint = "https://" + host
region = "luzon"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* Step 1: Assemble the specification request string *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
payload = ""
if http_request_method == "POST":
    canonical_querystring = ""
    ct = "json"
    payload = json.dumps(params)
canonical_headers = "content-type:application/%s\nhost:%s\n" % (ct, host)
signed_headers = "content-type;host"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
                     canonical_uri + "\n" +
                     canonical_querystring + "\n" +
                     canonical_headers + "\n" +
                     signed_headers + "\n" +
                     hashed_request_payload)
print(canonical_request)

# ************* Step 2: Stitching the string to be signed *************
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
                  str(timestamp) + "\n" +
                  credential_scope + "\n" +
                  hashed_canonical_request)
print(string_to_sign)
```

```
# ************* Step 3: Calculate the signature *************
# Compute signature digest function
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ************* Step 4: Splicing Authorization *************
authorization = (algorithm + " " +
                 "Credential=" + secret_id + "/" + credential_scope + ", " +
                 "SignedHeaders=" + signed_headers + ", " +
                 "Signature=" + signature)
print(authorization)

# Common parameters added to request header
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

request

1. Service Address

Region refers to the geographical area of a physical data center. Converge Cloud ensures complete isolation between different regions, providing maximum stability and fault tolerance. To reduce access latency and improve download speed, it is recommended to choose the region closest to your customers.

You can view the complete region list by using the API interface "DescribeRegions".

2. Communications Protocol

All the Converge Cloud APIs communicate via HTTPS, providing highly secure communication tunnels.

3. Request Methods

Supported HTTP request methods:

- POST (recommended)
- GET

PThe Content-Type types supported by POST requests:

- application/json (recommended). The TC3-HMAC-SHA256 signature algorithm must be used.
- application/x-www-form-urlencoded. The HmacSHA1 or HmacSHA256 signature algorithm must be used.
- multipart/form-data (only supported by certain APIs). You must use TC3-HMAC-SHA256 to calculate the signature.

The size of a GET request packet is up to 32 KB. The size of a POST request is up to 1 MB when the HmacSHA1 or HmacSHA256 signature algorithm is used, and up to 10 MB when TC3-HMAC-SHA256 is used.

4. Character Encoding

Only UTF-8 encoding is used.

result

Response for Successful Requests

For example, when calling CAM API (version: 2017-03-12) to view the status of instances (DescribeInstancesStatus), if the request has succeeded, you may see the response as shown below:


```
{
    "Response": {
        "TotalCount": 0,
        "InstanceStatusSet": [],
        "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
    }
}
```


- The API will return Response, which contains RequestId, as long as it processes the request. It does not matter if the request is successful or not.
- RequestId is the unique ID of an API request. Contact us with this ID when an exception occurs.
- Except for the fixed fields, all fields are action-specified. For the definitions of action-specified fields, see the corresponding API documentation. In this example, TotalCount and InstanceStatusSet are the fields specified by the API DescribeInstancesStatus. 0 TotalCount means that the requester owns 0 CVM instance so the InstanceStatusSet is empty.

Response for Failed Requests

If the request has failed, you may see the response as shown below:


```
{
    "Response": {
        "Error": {
            "Code": "AuthFailure.SignatureFailure",
            "Message": "The provided credentials could not be validated. Please check your signature is cor
rect."
        },
        "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
    }
}
```


- The presence of the Error field indicates that the request has failed. A response for a failed request will include Error, Code and Message fields.

- Code is the code of the error that helps you identify the cause and solution. There are two types of error codes so you may find the code in either common error codes or API-specified error codes.
- Message explains the cause of the error. Note that the returned messages are subject to service updates. The information the messages provide may not be up-to-date and should not be the only source of reference.
- RequestId is the unique ID of an API request. Contact us with this ID when an exception occurs.

Common Error Codes

If there is an Error field in the response, it means that the API call failed. The Code field in Error indicates the error code. The following table lists the common error codes that all actions can return.

Error Code	Description
AuthFailure.InvalidSecretId	Invalid key (not a API key type).
AuthFailure.MFAFailure	MFA failed.
AuthFailure.SecretIdNotFound	The key does not exist.
AuthFailure.SignatureExpire	Signature expired.
AuthFailure.SignatureFailure	Signature error.
AuthFailure.TokenFailure	Token error.
AuthFailure.UnauthorizedOperation	The request does not have CAM authorization.
DryRunOperation	DryRun Operation. It means that the request would have succeeded, but the DryRun parameter was used.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidAction	The API does not exist.
InvalidParameter	Incorrect parameter.
InvalidParameterValue	Invalid parameter value.
LimitExceeded	Quota limit exceeded.
MissingParameter	A parameter is missing.
NoSuchVersion	The API version does not exist.

Error Code	Description
RequestLimitExceeded	The number of requests exceeds the frequency limit.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	Resource is unavailable.
UnauthorizedOperation	Unauthorized operation.
UnknownParameter	Unknown parameter.
UnsupportedOperation	Unsupported operation.
UnsupportedProtocol	hHTTPS request method error. Only GET and POST requests are supported.
UnsupportedRegion	API does not support the requested region.

PublicParameter

Common parameters are parameters used to identify users and interface authentication purposes. If not necessary, these parameters are not described in the separate interface document of each interface, but these parameters need to be carried in each request before the request can be normally initiated.

signature method v3

When using TC3-HMAC-SHA256 signature method, the common parameters need to be uniformly placed in the HTTP Header request header, as follows:

parameter name	type	required	described
X-TC-Action	String	Yes	The name of the operation interface. Refer to the description of the public parameter Action in the input parameters of the interface documentation for the value. For example, the value for the query instance list interface of the cloud server is DescribeInstances.
X-TC-Region	String	Yes	region parameters, Data used to identify which region you wish to operate on. The region value accepted by the interface refers to the description of the input parameter common parameter Region in the interface document. Note: Some interfaces do not need to pass this parameter, which is specifically stated in the interface documentation. In this case, even passing this parameter will not take effect.
X-TC-Timestamp	Integer	Yes	The current UNIX timestamp, which records the time when the API request was initiated. for example 1529223702. Note: If the time difference from the server exceeds 5 minutes, it will cause a signature expiration error.
X-TC-Version	String	Yes	Version of API for operation. Refer to the description of the public parameter Version in the interface document for the value. For example, the version of the cloud server. 2017-03-12.
Authorization	String	Yes	HTTP The standard authentication header field is "Authorization". For example: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 Here, - TC3-HMAC-SHA256: Signature method, currently fixed as this value, - Credential: Credential: Signature credential; AKIDEXAMPLE is the SecretId; Date is a date in UTC time, and this value must match the value of X-TC-Timestamp (a common parameter) in UTC time format; service is the name of the product/service, and is generally a domain name prefix. For example,cvm; - SignedHeaders: The headers that contains the authentication information; content-type and host are the required headers; - Signature: Signature digest.
X-TC-Token	String	No	The token used for temporary certificates needs to be used in conjunction with a temporary key. Temporary keys and tokens need to be obtained by calling the access management service interface. Long-term keys do not require tokens.

signature method v1

When using HmacSHA1 and HmacSHA256 signature methods, common parameters need to be placed uniformly in the request string, as follows

parameter name	type	required	described
Action	String	Yes	The name of the interface to be operated. The value refers to the explanation of the public parameter Action in the interface documentation. For example, for the cloud server's query instance list interface, the value is DescribeInstances .
Region	String	Yes	The region parameter is used to identify which region's data you want to operate on. The region value accepted by the interface is specified in the input parameter of the public parameter Region in the interface documentation. Note: Some interfaces do not require this parameter to be passed, which will be specifically mentioned in the interface documentation. Even if this parameter is passed, it will not take effect in this case.
Timestamp	Integer	Yes	The current UNIX timestamp, which can record the time when the API request is initiated. For example, 1529223702. If the difference between the current time is too large, it will cause a signature expiration error.
Nonce	Integer	Yes	A random positive integer combined with Timestamp is used to prevent replay attacks.
SecretId	String	Yes	The SecretId is an identifier applied for the cloud API key to identify the user's identity. A SecretId corresponds to a unique SecretKey , and the SecretKey is used to generate the request signature Signature .
Signature	String	Yes	Request signature is used to verify the legitimacy of this request. Users need to calculate it based on the actual input parameters. The specific calculation method can be found in the interface authentication documentation.
Version	String	Yes	The version of the API being operated on. The value refers to the description of the public parameter Version in the interface documentation. For example, the version of the cloud server. 2017-03-12.
SignatureMethod	String	No	Signature method, currently supports HmacSHA256 and HmacSHA1. HmacSHA256 algorithm is used to verify signatures only if this parameter is specified as HmacSHA 256, otherwise HmacSHA1 is used to verify signatures.
Token	String	No	Token used for temporary certificate needs to be used together with temporary key. The temporary key and Token need to be obtained from the Access Management service invocation interface. Long-term keys do not require tokens.

zone list

A region (Region) refers to the geographical area of a physical data center. Converge Cloud Translation: To deliver verification that completely isolates different regions, ensuring maximum stability and fault tolerance between regions. In order to reduce access latency and improve download speed, it is recommended that you choose the region closest to your customers.

You can view the complete region list by using the API interface Query Region List.

Identity Provider APIs. CreateOauthProvider

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

API for adding OAuth configuration

Default interface request rate limit: 20 times per second.

Interface update time: 2022-12-12 19:35:47.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: CreateOauthProvider
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Name	Yes	No	String	Identity provider (enterprise) name. Example value: oauthserver Example: name
Desc	No	No	String	Remarks. Example value: mark Example: desc
ClientId	Yes	No	String	ID of the registered application. Example value: c1aa4fbce389206a7061 Example: c1aa4fbce389206a7061
ClientSecret	Yes	No	String	Key of the registered application. Example value: 837c669d84f2fe64b11a8853d521a17970df55ca Example: 944218ad060c8f2483d4d5bbfa64b6256b78aa79
AuthorizeUri	Yes	No	String	URL for OAuth authorization information. Example value: http://test.com/oauth/authorize Example: https://github.com/login/oauth/authorize
AccessTokenUri	Yes	No	String	URL for obtaining access_token. Example value: http://test.com/oauth/token Example: https://github.com/login/oauth/access_token
GetUserInfoUri	Yes	No	String	URL for obtaining user information. Example value: http://test.com/oauth/userinfo Example: https://api.github.com/user
UserNameField	Yes	No	String	Field name corresponding to the login account. Example value: username Example: user
NickNameField	No	No	String	Field name corresponding to the nickname. Example value: nickname Example: nick
PhoneNumField	Yes	No	String	Field name corresponding to the phone number. Example value: phone Example: phone
EmailField	Yes	No	String	Field name corresponding to the Email. Example value: email Example: email
IsSyncIdpUser	No	No	Int64	Whether to synchronize IdP users. 0: not synchronize; 1: synchronize. Example value: 0. Example: 0

3. Output Parameters

Parameter name	Type	Description
Id	Int64	ID Example: 1
Name	String	Name Example: name
SAMLProviderArn	String	SAMLProviderArn Example: ""
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
FailedOperation.IdentityExist	Identity authentication failed.

GetUserAccessToken

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

Obtain the access token of third-party open platform for a user.

Default interface request rate limit: 20 times per second.

Interface update time: 2022-08-11 19:05:48.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: GetUserAccessToken
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
UserAuthCode	Yes	No	String	Authorization code. Example: ""
OpenAccessToken	No	No	String	Third- party access token, used for complex authorization. Example: d1e459383977a892fa7b6c549d2f76f8

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
AppId	String	app id Example: 1255000002
UserOpenId	String	Third-party openId. Example: e5fcfd7cf67d62c85d38b10171641c57
UserUnionId	String	Third-party unionId. Example: a64e5abe67a8712c348c547a2760f9ce
UserAccessToken	String	Third-party access token. Example: 905dd67ae2a97e9870364a87d7872da2
ExpiresAt	Int64	Expired At Example: 1730433390
UserRefreshToken	String	refresh token Example: 905dd67ae2a97e9870364a87d7872da2
Scope	String	Authorization scope. Example: login
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.ParamError	Invalid input parameter.
InternalError.SystemError	Internal error. Please check and try again.
FailedOperation.AuthCodeError	Authorization code exception.

RefreshUserToken

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

Refreshes the third-party access_token of a user.

Default interface request rate limit: 20 times per second.

Interface update time: 2022-08-11 19:05:48.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: RefreshUserToken
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
UserRefreshToken	Yes	No	String	User refresh token. Example: d1e459383977a892fa7b6c549d2f76f8
UserOpenId	Yes	No	String	User openId. Example: e5fcfd7cf67d62c85d38b10171641c57

3. Output Parameters

Parameter name	Type	Description
UserAccessToken	String	Third-party access_token.

Parameter name	Type	Description
		Example: d1e459383977a892fa7b6c549d2f76f8
ExpiresAt	Int64	Expired At Example: 1730433390
AppId	String	appId Example: 1255000002
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
FailedOperation.RefreshTokenError	An error occurred when refreshing user token.
InvalidParameter.ParamError	Invalid input parameter.
InternalError.SystemError	Internal error. Please check and try again.

UpdateOauthProvider

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

API for updating OAuth configuration information

Default interface request rate limit: 20 times per second.

Interface update time: 2022-11-16 17:05:50.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: UpdateOauthProvider
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Desc	No	No	String	Description Example: desc
Name	Yes	No	String	Identity provider (enterprise) name. Example: name
Id	Yes	No	Int64	id Example: 1
OwnerUin	Yes	No	Int64	OwnerUin Example: 110000000001
ClientId	Yes	No	String	ID for app registration. Example: c1aa4fbce389206a7061

Parameter name	Required	Allow NULL	Type	Description
ClientSecret	Yes	No	String	Key for app registration. Example: 944218ad060c8f2483d4d5bbfa64b6256b78aa79
AuthorizeUri	Yes	No	String	URL for OAuth authorization. Example: https://github.com/login/oauth/authorize
AccessTokenUri	Yes	No	String	Obtains access_token URL. Example: https://github.com/login/oauth/access_token
GetUserInfoUri	Yes	No	String	Obtains user information URL. Example: https://api.github.com/user
UserNameField	Yes	No	String	Field name for login account. Example: user
NickNameField	No	No	String	Field name for nickname. Example: nick
PhoneNumField	Yes	No	String	Field name for mobile number. Example: phone
EmailField	Yes	No	String	Field name for email address. Example: email
IsSyncIdpUser	Yes	No	Int64	Whether to Synchronize IdP User Data Example: 0

3. Output Parameters

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

There is no error code related to business logic for this interface, For other error codes, see public

ErrorCode.

VerifyUserAccessToken

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

Verifies the access_token of third-party open platform for a user.

Default interface request rate limit: 20 times per second.

Interface update time: 2022-08-11 19:05:48.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: VerifyUserAccessToken
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
UserAccessToken	Yes	No	String	access token Example: d1e459383977a892fa7b6c549d2f76f8
UserOpenId	Yes	No	String	open id Example: e5fcfd7cf67d62c85d38b10171641c57

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
UserOpenId	String	Third-party platform openId. Example: e5fcfd7cf67d62c85d38b10171641c57
UserUnionId	String	Third-party platform unionId. Example: d1e459383977a892fa7b6c549d2f76f8
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
FailedOperation.UserrAccessTokenError	An error occurred when accessing token.
InvalidParameter.ParamError	Invalid input parameter.
InternalError.SystemError	Internal error. Please check and try again.

Other APIs
AttachRolePolicies

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

API for binding multiple policies to a role

Default interface request rate limit: 20 times per second.

Interface update time: 2022-08-11 19:02:25.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: AttachRolePolicies
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
RoleId	No	No	Uint64	Role ID (either the role ID or role name must be specified). Example: 4611686018427387904
RoleName	No	No	String	Role name (either the role name or role ID must be specified). Example: QCS_RoleName
PolicyId	No	No	Array of Uint64	List of policy IDs (either the policy IDs or policy names must be specified). Example: [1001,1002]
PolicyName	No	No	Array of String	List of policy names (either the policy names or policy IDs must be specified). Example: ["policygen-1","policygen-2"]

3. Output Parameters

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
FailedOperation.PolicyFull	The number of user policies has exceeded the limit.
InvalidParameter.OperatePoliciesOverLimit	Too many policies in one operation.
InvalidParameter.PasswordLengthTooShort	Password too short.
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.ParamError	Invalid input parameter.
InvalidParameter.PolicyIdNotExist	Policy ID does not exist.

AttachRolePolicy

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API (AttachRolePolicy) is used to associate a policy with a role.

Default interface request rate limit: 20 times per second.

Interface update time: 2022-08-11 19:02:25.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: AttachRolePolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
PolicyId	No	No	Uint64	Policy ID. Either PolicyId or PolicyName must be specified. Example: 1
AttachRoleId	No	No	String	Role ID, used to specify a role. Either AttachRoleId or AttachRoleName must be specified. Example: 4611686018427397905
AttachRoleName	No	No	String	Role name, used to specify a role. Either AttachRoleId or AttachRoleName must be specified. Example: QCS_RoleName
PolicyName	No	No	String	Policy name. Either PolicyId or PolicyName must be specified. Example: policygen-20141112201913

3. Output Parameters

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.PolicyIdNotExist	Policy ID does not exist.
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.RoleNotExist	Role does not exist.

Error Code	Description
InvalidParameter.AttachmentFull	Limit on number of authorization object associated policies in the principal field reached.

AttachRolesPolicy

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

Bind multiple roles to the policy.

Default interface request rate limit: 20 times per second.

Interface update time: 2023-02-09 15:36:51.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: AttachRolesPolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
RoleId	No	No	Array of Uint64	Role ID (either the role ID or role name must be specified). Example: [4611686018427387904,4611686018427387905]
RoleName	No	No	Array of String	Role name (either the role name or role ID must be specified). Example: ["role-1","role-2"]
PolicyId	No	No	Uint64	Policy ID (either policy ID or policy name is required) Example: 1000
PolicyName	No	No	String	Policy Name (either policy ID or policy name is required) Example: policygen-20141112201913

3. Output Parameters

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
FailedOperation.PolicyFull	The number of user policies has exceeded the limit.
InvalidParameter.OperatePoliciesOverLimit	Too many policies in one operation.
InvalidParameter.PasswordLengthTooShort	Password too short.
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.ParamError	Invalid input parameter.
InvalidParameter.PolicyIdNotExist	Policy ID does not exist.

CreatePolicy

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API (CreatePolicy) is used to create a policy.

Default interface request rate limit: 20 times per second.

Interface update time: 2022-12-02 16:17:55.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: CreatePolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
PolicyName	Yes	No	String	Policy name. Example: policygen-20141112201913
Description	No	No	String	Policy description. Example: Test description information.
PolicyDocument	Yes	No	String	Policy document. Example: {"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service": ["cloudaudit.<no value>","cls.<no value>"]}}]}. Among which, principal indicates the object authorized by the role. To obtain this parameter, see the details on obtaining role (https:// <no value>/document/product/598/36221), output parameter RoleInfo. Example: {"version":"2.0","statement":[{"effect":"allow","action":"cvm:Describe*","resource":"*"}]}
CreateMode	No	No	Uint64	Creation mode. 1: preset policy sourced from dbsql; 2: role policy created from a policy template by using a COS backend API; 3: policy created by using the policy generator; 4: tag-related. Example: 1

3. Output Parameters

Parameter name	Type	Description
PolicyId	Uint64	Policy ID Example: 1000
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.ParamError	Invalid input parameter.
ResourceNotFound.UserNotExist	User does not exist.
FailedOperation.PolicyFull	The number of user policies has exceeded the limit.
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.ActionNotExist	The action does not exist.

CreateRole

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API (CreateRole) is used to create a role.

Default interface request rate limit: 20 times per second.

Interface update time: 2022-08-11 19:02:25.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: CreateRole
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
RoleName	Yes	No	String	Role name. Example: QCS
PolicyDocument	Yes	No	String	Policy document. Example: {"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.<no value>","cls.<no value>"]}}]}. Among which, principal indicates the object authorized by the role. To obtain this parameter, see the details on obtaining role (https://<no value>/document/product/598/36221), output parameter RoleInfo. Example: {"version":"2.0","statement":[{"action":"sts:AssumeRole","effect":"allow","principal":{"qcs":["qcs::cam::uin/110000000007:root"]}}]}
Description	No	No	String	Role description. Example: adesc
ConsoleLogin	No	No	Uint64	Whether to allow login. 1: allow; 0: disallow. Example: 0
SessionDuration	No	No	Uint64	Maximum validity period of the temporary key for role application. (Value range: 0-43200) Example: 3600
RoleType	No	No	String	Role type (system

3. Output Parameters

Parameter name	Type	Description
RoleId	String	Role ID. Example: 4611686018427388476
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.AttachmentFull	Limit on number of authorization object associated policies in the principal field reached.
InvalidParameter.ConditionError	condition field of policy document invalid.
InvalidParameter.DescriptionLengthOverlimit	The input parameter description cannot exceed 300 bytes in length.
InvalidParameter.PrincipalError	principal field of policy document invalid.
InvalidParameter.RoleFull	Limit on number of roles reached.
InvalidParameter.RoleNameError	Role name invalid.
InvalidParameter.RoleNameInUse	A role with the same name already exists.
InvalidParameter.UserNotExist	User object does not exist
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.ParamError	Invalid input parameter.

DeletePolicy

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

API for deleting policies

Default interface request rate limit: 20 times per second.

Interface update time: 2024-11-11 16:48:23.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DeletePolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
PolicyId	No	No	Array of Uint64	Array of policy IDs. Batch deletion policies are supported. Example: [1000,1001]

3. Output Parameters

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.ParamError	Invalid input parameter.
InvalidParameter.PolicyIdError	Input parameter PolicyId invalid.
InvalidParameter.PolicyIdNotExist	Policy ID does not exist.
ResourceNotFound.NotFound	The resource does not exist.
ResourceNotFound.PolicyIdNotFound	Resources specified by PolicyId does not exist.

DeleteRole

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API (DeleteRole) is used to delete a specified role.

Default interface request rate limit: 20 times per second.

Interface update time: 2022-08-11 19:02:25.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DeleteRole
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
RoleId	No	No	String	Role ID, used to specify a role. Either RoleId or RoleName must be specified. Example: 4611686018427844696
RoleName	No	No	String	Role name, used to specify a role. Either RoleId or RoleName must be specified. Example: QCS_RoleName

3. Output Parameters

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InternalError.SystemError	Internal error. Please check and try again.

Error Code	Description
InvalidParameter.RoleNotExist	Role does not exist.
InvalidParameter.ParamError	Invalid input parameter.

DescribeRoleList

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API (DescribeRoleList) is used to get the role list under the account.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-02 21:20:38.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeRoleList
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Page	Yes	No	Uint64	Page number, starting from 1. Example: 1
Rp	Yes	No	Uint64	Number of rows per page, which cannot be greater than 200. Example: 5
Service	No	No	String	Filters service account entity by role. Example: cvm
Keyword	No	No	Array of String	Filters by role name or role description. Example: ["rolename","roleid"]

3. Output Parameters

Parameter	Type	Description

name		
List	Array of RoleInfo	List of role details. Example:View
TotalNum	Uint64	Total number of roles. Example: 14
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.ParamError	Invalid input parameter.

DetachGroupPolicies

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

API for unbinding multiple policies from a user group

Default interface request rate limit: 20 times per second.

Interface update time: 2024-11-11 16:49:40.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DetachGroupPolicies
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
GroupId	Yes	No	Uint64	User group ID. Example: 1000
PolicyId	Yes	No	Array of Uint64	List of policy IDs. Example: [1000,1001]

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.ParamError	Invalid input parameter.
InvalidParameter.PolicyIdNotExist	Policy ID does not exist.
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.PolicyIdError	Input parameter PolicyId invalid.

DetachGroupsPolicy

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

API for unbinding a policy from multiple user groups

Default interface request rate limit: 20 times per second.

Interface update time: 2024-11-11 16:50:02.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DetachGroupsPolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
GroupId	Yes	No	Array of Uint64	List of user group IDs. Example: [1000,1001]
PolicyId	Yes	No	Uint64	Policy ID. Example: 1000

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.ParamError	Invalid input parameter.
InvalidParameter.PolicyIdNotExist	Policy ID does not exist.
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.PolicyIdError	Input parameter PolicyId invalid.

DetachUsersPolicy

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

API for unbinding a policy from multiple users

Default interface request rate limit: 20 times per second.

Interface update time: 2024-11-11 16:51:47.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DetachUsersPolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
TargetUin	Yes	No	Array of Uint64	List of target user IDs. Example: [1000,1001]
PolicyId	Yes	No	Uint64	Policy ID. Example: 1000

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.ParamError	Invalid input parameter.
InvalidParameter.PolicyIdNotExist	Policy ID does not exist.
FailedOperation.PolicyFull	The number of user policies has exceeded the limit.
InternalError.SystemError	Internal error. Please check and try again.

GetPolicy

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API (GetPolicy) is used to query and view policy details.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-02 21:20:38.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: GetPolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
PolicyId	Yes	No	Uint64	Policy ID. Example: 17698703

3. Output Parameters

Parameter name	Type	Description
PolicyName	String	Policy name. Example: policygen-20141112201913
Description	String	Policy description. Example: Test policy.
Type	Uint64	1: custom policy, 2: preset policy. Example: 1
AddTime	Datetime	Creation time. Example: 2019-04-29 21:18:28
UpdateTime	Datetime	Last update time. Example: 2019-04-29 21:28:32
PolicyDocument	String	Policy document. Example: {"version":"2.0","statement":[{"effect":"allow","action":["name\/cos: *"],"resource":[*""]}]}
PresetAlias	String	Description Example: Remarks
IsServiceLinkedRolePolicy	Uint64	Whether it is a service-related policy. Example: 1
CreateMode	Uint64	1: preset policy sourced from the dbsql; 2: role policy created based on a policy template by using a COS backend API; 3: policy created by using the policy generator; 4: tag-related policy. Example: 1
IsCheck	Uint64	Whether the key has been verified. 1: yes; 0: no. Example: 1
PolicyId	Uint64	Policy ID. Example: 1000
ServiceType	String	Service Name Example: cvm
Attachments	Uint64	number of associated user Example: 1
IsAttached	Uint64	not null when you need to query whether the marked entity is associated with the policy. 0: not associated with the policy; 1: associated with the policy. Example: 1
Deactived	Uint64	Deactivation status. 1: deactivated; 0: not deactivated. Example: 0
DeactivedDetail	Array of String	List of deactivated products. Example: ["deacproduct"]
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
ResourceNotFound.PolicyIdNotFound	Resources specified by PolicyId does not exist.
InvalidParameter.ParamError	Invalid input parameter.

Error Code	Description
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.PolicyIdError	Input parameter PolicyId invalid.

GetRole

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API (GetRole) is used to get the details of a specified role.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-02 21:20:38.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: GetRole
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
RoleId	No	No	String	Role ID, used to specify a role. Either RoleId or RoleName must be specified. Example: 4611686018427844696
RoleName	No	No	String	Role name, used to specify a role. Either RoleId or RoleName must be specified. Example: QCSRole

3. Output Parameters

Parameter name	Type	Description
RoleInfo	RoleInfo	Role details. Example:View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.ParamError	Invalid input parameter.
InvalidParameter.RoleNotExist	Role does not exist.

GetServiceRoleInfo

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

API for obtaining the information of a service role

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-02 21:20:38.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: GetServiceRoleInfo
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
RoleName	No	No	String	Role name. Example: QCS
PolicyName	No	No	Array of String	List of policy names. Example: ["policygen-1","policygen-2"]

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
RoleName	String	Role name. Example: QCS
ServiceType	String	Suitable Business Type Example: cvm
ServiceTypeEn	String	Business type in English. Example: cvm
RoleDesc	String	Role description. Example: QCS
RoleDescEn	String	Role description in English. Example: QCS
PolicyName	String	Name of the preset policy. Example: policygen-20141112201913
Remark	String	Description Example: remark
EnRemark	String	English Description Example: ""
PolicyList	Array of RolePolicyList	Policy list Example:View
RoleDescI18n	String	Role description that supports internationalization. Example: ""
Id	String	Role ID Example: ""
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.ParamError	Invalid input parameter.

Error Code	Description
InternalError.SystemError	Internal error. Please check and try again.

ListAttachedGroupPolicies

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

API for obtaining the list of policies associated with a user group

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-02 21:20:38.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: ListAttachedGroupPolicies
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
TargetGroupId	Yes	No	Uint64	User group ID. Example: 3349
Page	Yes	No	Uint64	Page number, starting from 1. The default value is 1. Example: 1
Rp	Yes	No	Uint64	Size per page. Default value: 20. Example: 10

3. Output Parameters

Parameter name	Type	Description
TotalNum	Uint64	Total number of policies. Example: 1
List	Array of AttachPolicyInfo	Policy list Example:View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.ParamError	Invalid input parameter.
InternalError.SystemError	Internal error. Please check and try again.

ListAttachedRolePolicies

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API (ListAttachedRolePolicies) is used to obtain the list of the policies associated with a role.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-02 21:20:38.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: ListAttachedRolePolicies
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
RoleId	No	No	String	Role ID, used to specify a role. Either RoleId or RoleName must be specified. Example: 4611686018427397905
RoleName	No	No	String	Role name, used to specify a role. Either RoleId or RoleName must be specified. Example: role_data
PolicyType	No	No	String	Filters by policy type, If you set this field to User, only custom policies are queried. If you set this field to QCS, only preset policies are queried. Example: 1
Page	Yes	No	Uint64	Page number, starting from 1. Example: 1
Rp	Yes	No	Uint64	Number of rows per page, which cannot be greater than 200. Example: 10

3. Output Parameters

Parameter name	Type	Description
List	Array of AttachedPolicyOfRole	List of role-associated policies. Example:View
TotalNum	Uint64	Total number of role-associated policies. Example: 1
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.ParamError	Invalid input parameter.

ListEntitiesForPolicy

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API (ListEntitiesForPolicy) is used to query the list of entities associated with a policy.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-02 21:20:38.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: ListEntitiesForPolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
PolicyId	Yes	No	Uint64	Policy ID. Example: 524497
Page	No	No	Uint64	Page number, starting from 1. The default value is 1. Example: 1
Rp	No	No	Uint64	Size per page. Default value: 20. Example: 10
EntityFilter	No	No	String	Valid values: 'All', 'User', 'Group', and 'Role'. 'All' means to obtain all entity types. 'User' means to obtain only sub-accounts. 'Group' means to obtain only user groups. 'Role' means to obtain only roles. The default value is 'All'. Example: All

3. Output Parameters

Parameter name	Type	Description
TotalNum	Uint64	Total number of entities. Example: 1
List	Array of AttachEntityOfPolicy	List of entities. Example:View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.PolicyIdError	Input parameter PolicyId invalid.
InvalidParameter.ParamError	Invalid input parameter.
InternalError.SystemError	Internal error. Please check and try again.

ListPolicies

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API is used to query the policy list.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-09 17:18:22.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: ListPolicies
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Rp	No	No	Uint64	Number of policies per page. The default value is 20. The value must be greater than 0 and less than or equal to 200. Example: 1
Page	No	No	Uint64	Page number. The default value is 1. Value range: 1-200. Example: 10
Scope	No	No	String	Valid values: 'All', 'QCS' and 'Local'. 'All' means to obtain all policies. 'QCS' means to obtain only preset policies. 'Local' means to obtain only custom policies. The default value is 'All'. Example: All
Keyword	No	No	String	Matches by policy name. Example: name
TargetUin	No	No	Uint64	Matches by UIN. Example: 10093
TargetGroupId	No	No	Uint64	Matches by ID. Example: 10093
TargetRoleId	No	No	Uint64	Matches by role ID. Example: 10093
ServiceType	No	No	String	Matches by product ID, such as cvm. Example: cvm
FlagUin	No	No	Uint64	Associates by UIN flag. Example: 10093
FlagGroupId	No	No	Uint64	Associates by GroupId flag. Example: 10093
FlagRoleId	No	No	Uint64	Associates by Role ID flag. Example: 10093
Order	No	No	String	Policy sorting parameter. Example: desc
ProjectVisible	No	No	Uint64	Project visibility Example: 1
Client	No	No	String	Client Example: console

3. Output Parameters

Parameter name	Type	Description
TotalNum	Uint64	Total number of policies. Example: 239
List	Array of StrategyInfo	Policy array. Each member of the array includes the policyId, policyName, addTime, type, description, createMode fields. Where: policyId: ID policyName: policy name addTime: creation time of the policy Type: 1 means custom policy and 2 means preset policy. description: policy description createMode: 1 indicates that the policy is created according to the business permissions, and other values indicate that the policy syntax can be viewed and the policy can be updated through the policy syntax. Attachments: number of associated user ServiceType: product associate with the policy IsAttached: not null when you need to query whether the marked entity is associated with the policy. 0: not associated with the policy; 1: associated with the policy. Example:View
ServiceTypeList	Array of String	Service List Example: ["cvm"]
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.ParamError	Invalid input parameter.
InvalidParameter.UinError	Uin field invalid.
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.ScopeError	Scope field invalid.

Error Code	Description
InvalidParameter.KeywordError	Keyword field invalid.
InvalidParameter.GroupIdError	GroupId field invalid.
InvalidParameter.ServiceTypeError	ServiceType field invalid.

UpdateAssumeRolePolicy

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API (UpdateAssumeRolePolicy) is used to modify the trust policy of a role.

Default interface request rate limit: 20 times per second.

Interface update time: 2022-08-11 19:02:25.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: UpdateAssumeRolePolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
RoleId	No	No	String	Role ID, used to specify a role. Either RoleId or RoleName must be specified. Example: 4611686018427731422
RoleName	No	No	String	Role name, used to specify a role. Either RoleId or RoleName must be specified. Example: QCS_RoleName
PolicyDocument	Yes	No	String	Policy document. Example: {"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service": ["cloudaudit.<no value>","cls.<no value>"]}}]}. Among which, principal indicates the object authorized by the role. To obtain this parameter, see the details on obtaining role (https:// <no value>/document/product/598/36221), output parameter RoleInfo. Example: {"version":"2.0","statement":[{"action":"sts:AssumeRole","effect":"allow","principal":{"qcs":["qcs::cam::uin/110000000007:root"]}}]}

3. Output Parameters

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.ConditionError	condition field of policy document invalid.
InvalidParameter.VersionError	Version field of policy document invalid.
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.AttachmentFull	Limit on number of authorization object associated policies in the principal field reached.
InvalidParameter.PrincipalError	principal field of policy document invalid.
InvalidParameter.RoleNotExist	Role does not exist.
InvalidParameter.ParamError	Invalid input parameter.

UpdatePolicy

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API (UpdatePolicy) is used to update a policy.
This API will update the default version of an existing policy instead of creating a new one. If no policy exists, a default version will be created.

Default interface request rate limit: 20 times per second.

Interface update time: 2022-08-11 19:02:25.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: UpdatePolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
PolicyId	Yes	No	Uint64	Policy ID. Example: 17698703
PolicyName	No	No	String	Policy name. Example: QCS_PolicyName
Description	No	No	String	Policy description. Example: Policy_Description
PolicyDocument	No	No	String	Policy document. Example: {"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service": ["cloudaudit.<no value>","cls.<no value>"]}}]}. Among which, principal indicates the object authorized by the role. To obtain this parameter, see the details on obtaining role (https:// <no value>/document/product/598/36221), output parameter RoleInfo. Example: {"version":"2.0","statement":[{"effect":"allow","action":"cvm:Describe*","resource":"*"}]}

3. Output Parameters

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.PolicyNameError	PolicyName field invalid.
InvalidParameter.PrincipalError	principal field of policy document invalid.
InvalidParameter.ConditionError	condition field of policy document invalid.
InvalidParameter.ParamError	Invalid input parameter.
InvalidParameter.PolicyDocumentError	PolicyDocument field invalid.
InvalidParameter.ActionError	Action field of policy document invalid.
ResourceNotFound.PolicyIdNotFound	Resources specified by PolicyId does not exist.
InvalidParameter.NotSupportProduct	CAM does not support the resource type specified in the policy document.
InvalidParameter.PolicyDocumentLengthOverLimit	PolicyDocument field exceeds length restriction.
ResourceNotFound.UserNotExist	User does not exist.
InternalError.SystemError	Internal error. Please check and try again.
InvalidParameter.StatementError	Statement field of policy document invalid.
InvalidParameter.VersionError	Version field of policy document invalid.
InvalidParameter.EffectError	Effect field of policy document invalid.
InvalidParameter.PolicyIdNotExist	Policy ID does not exist.
InvalidParameter.PolicyIdError	Input parameter PolicyId invalid.
InvalidParameter.ResourceError	Resource field of policy document invalid.
InvalidParameter.UserNotExist	User object does not exist
ResourceNotFound.GroupNotExist	The user group does not exist.
ResourceNotFound.NotFound	The resource does not exist.
FailedOperation.PolicyNameInUse	The policy name specified by the PolicyName field already exists.
InvalidParameter.AttachmentFull	Limit on number of authorization object associated policies in the principal field reached.
UnauthorizedOperation	Unauthorized operation.
InvalidParameter.DescriptionLengthOverlimit	The input parameter description cannot exceed 300 bytes in length.

User APIs
GetPasswordRules

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This (GetPasswordRules) is used to obtain the password setting rules of a user.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-23 16:09:45.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: GetPasswordRules
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.

3. Output Parameters

Parameter name	Type	Description
Rules	PasswordRules	List of password rules. Example:View

Parameter name	Type	Description
UpdateTime	String	Update time. Example: 2019-04-29 21:18:28
Modifier	String	Modifier. Example: 130000000001
BlackList	String	Blocklist string list in JSON format. Example: aaa
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

There is no error code related to business logic for this interface, For other error codes, see public ErrorCode.

GetSubsGroup

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

Queries the list of user groups to which a sub-account belongs

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-23 15:09:13.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: GetSubsGroup
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Uid	Yes	No	Uint64	Recipient user ID. Example: 1258042
Rp	Yes	No	Int64	Number of entries per page. Example: 20
Page	Yes	No	Int64	Number of pages. Example: 1

3. Output Parameters

Parameter name	Type	Description
TotalNum	String	Total number. Example: 1
GroupInfo	Array of GroupInfo	List of user information. Example:View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InternalError.SystemError	Internal error. Please check and try again.

GetUinBySecretId

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

This API is used to query UIN based on SecretId.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-03-17 18:04:03.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: GetUinBySecretId
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
ApiSecretId	Yes	No	String	Key ID Example: AK****

3. Output Parameters

Parameter name	Type	Description
Uin	Uint64	User ID Example: 100

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.ParamError	Invalid input parameter.

UpdatePasswordRules

1. Interface Description

Interface request domain name: cam.api3.convergecloud.com.

Updates user password setting rules.

Default interface request rate limit: 20 times per second.

Interface update time: 2022-08-11 19:06:02.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: UpdatePasswordRules
Version	Yes	No	String	Public parameter, The value of this interface: 2019-01-16
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Rules	Yes	No	PasswordRules	Password setting rules. Example:View
BlackList	No	No	String	Blocklist string list in JSON format. Example: aaa

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

There is no error code related to business logic for this interface, For other error codes, see public ErrorCode.

Data structure

ServicePermItem

API

Referenced by the following interfaces: GetServicePermList

name	required	NULL allowed	type	description
AddTime	no	no	Datetime	Creation time. Example: 2014-08-03 12:00:00
ApiAddr	no	no	String	ApiAddr Example: ApiAddr
ApiZhName	no	no	String	Description Example: ADMIN
AuthFunction	no	no	String	Authentication Interface Example: AuthFunction
CWildcardName	no	no	String	CWildcardName Example: CWildcardName
InterfaceEnName	no	no	String	API name Example: CVM
InterfaceLevel	no	no	String	Authentication Granularity, 0: Interface Level, 1: Resource Level Example: 1
IsAuthBusiness	no	no	String	Authentication Method, 0: Cloud API Forwards Authentication, 1: Business Invokes Authentication Interface Example: 1
IsNeedObject	no	no	String	IsNeedObject Example: 1
IsSeen	no	no	Uint64	IsSeen Example: 1
IsSeenAtGenerator	no	no	String	Whether the Policy Generator Is Visible Example: 1
IsSpResource	no	no	String	IsSpResource Example: 1
IsUserSet	no	no	String	IsUserSet Example: 1
PermId	no	no	String	Id Example: 1000
ReadWriteDetail	no	no	String	Interface Category Example: API category. 0: read; 1: write; 2: mark; 3: list.
ResourceType	no	yes	String	Resource Category Example: cvm
UpdateTime	no	no	String	Update time. Example: 2014-08-03 12:00:00
Weight	no	no	String	Weight Example: 1
Writter	no	no	String	Operator Example: admin
ServiceName	no	no	String	Service Name Example: cvm
ProductShortCode	no	no	String	ProductShortCode Example: p_cvm
ProductShortName	no	no	String	ProductShortName Example: p_cvm
ServiceType	no	no	String	Service Type Example: cvm
Interface	no	no	String	API name Example: CVM
InterfaceName	no	no	String	API name Example: CVM

ApiKeyDetail

Details of a persistent key.

Referenced by the following interfaces: GetApiKey

name	required	NULL allowed	type	description
SecretId	no	no	String	Key ID. Example: AK**********

name	required	NULL allowed	type	description
SecretKey	no	no	String	Persistent key. Example: TR**********
CreateTime	no	no	Uint64	Creation time (timestamp). Example: 2014-08-03 12:00:00
Status	no	no	Uint64	Status (2: valid; 3: disabled). Example: 1
Source	no	no	Uint64	Source. The default value is 0. Example: 1
Remark	no	yes	String	Description. Example: aremark

ExtAttr

User attribute combination.

Referenced by the following interfaces: DescribeSubReceiver

name	required	NULL allowed	type	description
NeedResetToken	yes	no	Int64	Needs to reset the MFA token. Example: 0
NeedResetStoken	yes	no	Int64	Needs to reset the MFA stoken. Example: 0

UpdateGroupInfo

API for updating user group information

Referenced by the following interfaces: UpdateSubAccount

name	required	NULL allowed	type	description
GroupId	yes	no	Uint64	User group ID. Example: 15094
Uid	yes	no	Uint64	User ID. Example: 1258042
GroupName	no	no	String	User group name. Example: name
Channel	no	no	Int64	Channel for receiving messages. 0: none; 1: SMS; 2: email; 3: SMS and email. Example: 0

RolePolicyList

Role Policy List

Referenced by the following interfaces: GetServiceRoleInfo

name	required	NULL allowed	type	description
IsHidden	no	no	Uint64	Status Example: 1
PolicyId	no	no	String	Policy ID Example: 100
PolicyName	no	no	String	Policy name. Example: policygen-20141112201913

ListOpenPlatform

Detailed information of a third-party platform.

Referenced by the following interfaces: ListOpenPlatforms

name	required	NULL allowed	type	description
OpenId	yes	no	Int64	openid Example: 1
AppId	yes	no	String	app id Example: 1255000002
OpenName	yes	no	String	app name Example: ""
OpenLogo	yes	no	String	open logo Example: ""
OpenHome	yes	no	String	Homepage of the third-party platform. Example: ""

name	required	NULL allowed	type	description
OpenType	yes	no	Int64	Authorization type. Example: 0
Uin	yes	no	Uint64	Applies for accounts. Example: 110000000001
Status	yes	no	Int64	Status Example: 0
Domain	yes	no	String	Domain name of the third-party platform. Example: ""
State	yes	no	Int64	Frozen status. 0: non-frozen; 1: frozen. Example: 0
Modifier	yes	no	String	Modifier. Example: 110000000001
ModifyTime	yes	no	String	Update time. Example: 2019-04-29 21:18:28
CreateTime	yes	no	String	Creation time. Example: 2019-04-29 21:18:28
Memo	yes	no	String	Description Example: ""

ServiceItem

Service

Referenced by the following interfaces: GetServiceList

name	required	NULL allowed	type	description
AddTime	no	no	Datetime	Creation time. Example: 2014-08-03 12:00:00
ArnDocument	no	no	String	ArnDocument Example: https://www.example.com/document/product/378/8222965
ColConf	no	yes	String	ColConf Example: ColConf
DefAddr	no	yes	String	DefAddr Example: DefAddr
DefaultStrategyList	no	no	String	Default policy Example: DefaultStrategyList
IsAllowDefProj	no	no	String	IsAllowDefProj Example: 1
IsDisProject	no	no	String	IsDisProject Example: 1
IsDisZone	no	no	String	IsDisZone Example: 1
IsSeen	no	no	String	Whether it is visible. Example: 1
Online	no	no	String	Online Example: Online
QueryAddr	no	no	String	QueryAddr Example: cvm
QueryInterface	no	no	String	QueryInterface Example: cvm
ServiceEnName	no	no	String	Service English Name Example: cvm
ServiceName	no	no	String	Service Name Example: cvm
ServiceType	no	no	String	Service Type Example: cvm
SynInterface	no	no	String	SynInterface Example: cvm
UpdateTime	no	no	Datetime	Change Time Example: 2014-08-03 12:00:00
Weight	no	no	String	Weight Example: 1
WhiteKey	no	no	String	WhiteKey Example: white1
Writter	no	no	String	Creator Example: admin

name	required	NULL allowed	type	description
ResourceTypeList	no	no	Array of ResourceTypeItem	Resource Type Array Example:View
Type	no	no	String	Type Example: cvm

ServiceApiInfo

Information of a service and its APIs.

Referenced by the following interfaces: GetServiceApiList

name	required	NULL allowed	type	description
Name	yes	no	String	Service Name Example: User and Permission
ServiceType	yes	no	String	Service ID. Example: cam
ArnDocument	yes	yes	String	Service introduction document link. Example: https://www.example.com/document/product/378/8965
ApiList	no	yes	Array of ServiceApiListInfo	List of API information. Example:View
ConditionKeyList	no	yes	Array of String	List of condition rules. Example: ["qcs:ip"]

StrategyInfoForAction

Strategy details

Referenced by the following interfaces: ListPoliciesByAction

name	required	NULL allowed	type	description
PolicyId	no	no	Uint64	Policy ID. Example: 1
PolicyName	no	no	String	Policy name. Example: AdministratorAccess
AddTime	no	no	Datetime	Policy creation time. Example: 2016-06-02 19:40:09
Type	no	no	Uint64	Policy type. 1: custom policy; 2: preset policy. Example: 1
Description	no	no	String	Policy description. Example: This policy allows you to manage all users and their permissions, related financial info, and cloud service assets under this account.
CreateMode	no	no	Uint64	Creation source. 1: created via the console; 2: created via policy syntax. Example: 1
Attachments	no	yes	Uint64	number of associated user Example: 1
ServiceType	no	no	String	product associate with the policy Example: cam
IsAttached	no	yes	Uint64	not null when you need to query whether the marked entity is associated with the policy. 0: not associated with the policy; 1: associated with the policy. Example: 1
Deactived	no	yes	Uint64	Deactivation status. 1: deactivated; 0: not deactivated. Example: 0
DeactivedDetail	no	yes	Array of String	List of deactivated products. Example: ["cvm"]
IsCheck	no	yes	String	Whether security verification is performed. 1: yes; 0: no. Example: 1
PolicyDocument	no	no	String	Policy Syntax Example: {"version":"2.0","statement":[{"action":["cam:*","account:*"],"resource":["*"],"effect":"allow"}]}
UpdateTime	no	no	String	Modified At Example: 2016-06-02 19:40:09

FilterItem

Filter data structure.

Referenced by the following interfaces: ListMaskedSubAccounts, ListSubAccounts

name	required	NULL allowed	type	description
Operator	yes	no	String	Operator. Example: **

name	required	NULL allowed	type	description
Attr	yes	no	String	Attribute. Example: ""
Value	yes	no	String	Match value. Example: ""

ApiKey

List of API key data.

Referenced by the following interfaces: CreateApiKey, CreateCollApiKey, QueryApiKey, QueryCollApiKey, QueryKeyBySecretId

name	required	NULL allowed	type	description
SecretId	no	no	String	Key ID. Example: AKID***wdew
CreateTime	no	no	Uint64	Creation time (timestamp). Example: 1615990212
Status	no	no	Uint64	Status (2: valid; 3: disabled). Example: 2
SecretKey	no	no	String	Persistent key. Example: ACCF***wdew
Source	no	no	Uint64	Source Example: 0
Remark	no	no	String	Description Example: remark

GroupMember

User group member.

Referenced by the following interfaces: AddUserToGroup, RemoveUserFromGroup, UpdateGroupMember

name	required	NULL allowed	type	description
Uid	yes	no	String	User ID. Example: 100
GroupId	yes	no	String	Group ID. Example: 100

AttachedPolicyOfRole

Policy associated with the role

Referenced by the following interfaces: ListAttachedRolePolicies

name	required	NULL allowed	type	description
PolicyId	yes	no	Uint64	Policy ID. Example: 1000
PolicyName	yes	no	String	Policy name Example: policygen-20141112201913
AddTime	yes	no	String	Binding time. Example: 2014-08-03 12:00:00
PolicyType	yes	yes	String	Policy type. User: custom policy; QCS: preset policy. Example: Filter by policy type. User: querying custom policies only; QCS: querying preset policies only.
CreateMode	yes	no	Uint64	Policy creation method. 1: created by product feature or project permission; other values: created by policy syntax. Example: 1

GroupMeta

User group metainformation.

Referenced by the following interfaces: GetUserGroupList, ListAllUserGroup

name	required	NULL allowed	type	description
GroupId	no	yes	Int64	User group ID. Example: 100
GroupName	no	yes	String	User group name. Example: group1

SubAccountFilter

Information of a sub-account with filters.

Referenced by the following interfaces: GetGroupList, GetGroupsSubAccount, GetSubsGroup, ListGroups, ListGroupsForConsole, ListMaskedSubAccounts, ListSubAccounts

name	required	NULL allowed	type	description
Uid	yes	yes	Uint64	Sub-user UIDs. Example: 1258042
Uin	yes	yes	Uint64	User UIN. Example: 130000000001
Name	yes	yes	String	Username. Example: username
Remark	yes	yes	String	Description Example: remark
CanLogin	yes	yes	Uint64	Whether the account can be logged in to. Example: 1
PhoneNum	yes	yes	String	Mobile number. Example: 11111111111
CountryCode	yes	yes	String	Area code. Example: 86
PhoneFlag	yes	yes	Int64	Whether the phone number is verified. Example: 1
Email	yes	yes	String	Email address Example: mail@mail.com
EmailFlag	yes	yes	Int64	Whether the email address is verified. Example: 1
UserType	yes	yes	Int64	User type. Example: 0
CreateTime	yes	yes	String	Creation time. Example: 2019-04-29 21:18:28
IsReceiverOwner	yes	yes	Int64	Whether the account id the message recipient. Example: 1
SystemType	yes	yes	String	Type Example: subaccount
NeedResetPassword	yes	yes	Int64	Whether the password needs to be reset. Example: 0
ConsoleLogin	yes	yes	Int64	Whether console login is allowed. Example: 1
WxzsStatus	yes	yes	Int64	WeChat official account following status. Example: 0
PermType	yes	yes	Array of String	Permission type. Example: ["0"]
NickName	yes	yes	String	Nickname. Example: nick
QywxUserId	no	no	String	Enterprise WeChat User ID Example: qywxuserid
UserAttributeAndValues	no	yes	Array of AccountAttributeAndValue	Extended Attribute Example:View
Status	no	no	Int64	Status Example: 0
LoginStatus	no	no	Int64	Login status Example: 0
VerifyPhone	no	no	String	VerifyPhone Example: 11111111111
VerifyEmail	no	no	String	VerifyEmail Example: mail@mail.com
VerifyCountryCode	no	no	String	VerifyCountryCode Example: 86

OwnerInfo

Root account information.

Referenced by the following interfaces: ListMaskedSubAccounts, ListSubAccounts

name	required	NULL allowed	type	description
Uin	yes	yes	Uint64	Root account UIN. Example: 130000000001
UserName	yes	yes	String	User Name Example: username
CheckStatus	yes	no	Uint64	Verification status. Example: 0

UserInfo

User Info

Referenced by the following interfaces: AddSubAccount, ListGroups

name	required	NULL allowed	type	description
CanLogin	no	no	String	Sub-account type. 0: not allowed to log in to the console; 1: allowed to log in to the console. Example: 1
CountryCode	no	no	String	Area code. Example: 86
Detail	no	no	AccountDetail	Details. Example:View
Name	no	no	String	Name. Example: name
PhoneNum	no	no	String	Phone number. Example: 11111111111
SystemType	no	no	String	Account type. Example: Subaccount
Email	no	no	String	Security email address. Example: mail@mail.com
NickName	no	no	String	Nickname. Example: nickname
Remark	no	no	String	Remarks. Example: remark
WxzsStatus	no	no	Int64	WeChat login status. Example: 0
ContactMail	no	no	String	Contact email address. Example: mail@mail.com
IsReceiverOwner	no	no	Int64	Whether it is the root account. 1: root account, 0: sub-account. Example: 1
IdentifyType	no	no	Int64	Identity type. Example: 0

Receiver

Message recipient information.

Referenced by the following interfaces: DescribeSubAccountContacts

name	required	NULL allowed	type	description
Uid	yes	no	Uint64	id Example: 1258042
Name	yes	no	String	Name. Example: username
Remark	yes	no	String	Description Example: remark
PhoneNumber	yes	no	String	Mobile phone number Example: 11111111111
PhoneFlag	yes	no	Int64	Whether the mobile number is verified. Example: 1
Email	yes	no	String	Email address Example: mail@mail.com
EmailFlag	yes	no	Int64	Whether the email address is verified. Example: 1
IsReceiverOwner	yes	no	Int64	Whether the recipient is the main contact. Example: 1
WechatFlag	no	no	Int64	Whether to allow the receiving of notifications on WeChat. Example: 0
Uin	yes	no	Uint64	Account UIN. Example: 130000000001

SubAccountInfo

Sub-account user information.

Referenced by the following interfaces: UpdateSubAccount

name	required	NULL allowed	type	description
CanLogin	no	no	String	Whether the account can logged in to. 0: no; 1: yes. Example: 1

name	required	NULL allowed	type	description
ConsoleLogin	no	no	String	Console login or not. 1: yes. Example: 1
CountryCode	no	no	String	Country code. Example: 86
Name	no	no	String	User Name Example: Name
NeedResetPassword	no	no	String	Whether the password needs to be reset. 1: yes. Example: 0
PhoneNum	no	no	String	Phone number Example: 11111111111
Remark	no	no	String	Description Example: Remarks
SystemType	no	no	String	Account type. Example: subaccount
Uid	no	no	String	Recipient UID. Example: 1280504
Uin	no	no	String	Unique ID of the account. Example: 110000000001
Password	no	no	String	Password. Example: password
WxzsStatus	no	no	Int64	WeChat message status. Example: 0
UserType	no	no	Int64	User type. Example: 0
Email	no	no	String	Email Example: mail@mail.com
Account	no	no	String	User Name Example: username
Lang	no	no	String	Language. Example: en-US
NickName	no	no	String	Nickname. Example: Alias

GroupData

Information about a user group.

Referenced by the following interfaces: GetAllSubUser

name	required	NULL allowed	type	description
GroupId	no	no	Int64	User group ID. Example: 1000
GroupName	yes	no	String	User group name. Example: group1
GroupNum	yes	no	Int64	Number of members in a user group. Example: 100
Channel	yes	no	Int64	Created channel. Example: 0
GroupMem	no	no	Array of Uint64	Group Member UID Example: [1280504,1280505]

GroupInfoWithRemark

GroupInfoWithRemark

Referenced by the following interfaces: ListUsersForGroup

name	required	NULL allowed	type	description
GroupId	no	no	Int64	GroupId Example: 100
GroupName	no	no	String	GroupName Example: name
Remark	no	no	String	Remark Example: remark

GroupUserInfo

User information under a user group.

Referenced by the following interfaces: GetGroupsSubAccount

name	required	NULL allowed	type	description
Uid	yes	yes	Uint64	Recipient user ID. Example: 1258042
Uin	yes	yes	Uint64	Unique ID of the account. Example: 130000000001
Name	yes	yes	String	User Name Example: username
PhoneNum	yes	yes	String	Phone number Example: 11111111111
CountryCode	yes	yes	String	Country code. Example: 86
PhoneFlag	yes	yes	String	Mobile number flag. Example: 1
Email	yes	yes	String	Email address Example: mail@mail.com
EmailFlag	yes	yes	String	Email address flag. Example: 1
UserType	yes	yes	Int64	User type. Example: 0
CreateTime	yes	yes	String	Creation time. Example: 2019-04-29 21:18:28
IsReceiverOwner	yes	yes	String	Whether the recipient is a root account. Example: 1
SystemType	yes	yes	String	Account type. Example: subaccount
NickName	yes	yes	String	Nickname. Example: nick

CasProviderItem

CAS server configuration information.

Referenced by the following interfaces: DescribeCasProvider

name	required	NULL allowed	type	description
Id	yes	no	Int64	provider id Example: 1
CreateUin	yes	no	Uint64	Creator account UIN. Example: 110000000001
OwnerUin	yes	no	Uint64	UIN of the root account. Example: 110000000001
Name	yes	no	String	Parameter Example: name
Desc	yes	no	String	Description. Example: desc
ProviderType	yes	no	Int64	Provider type. Example: 8
Status	yes	no	Int64	Status Example: 0
ModifyTime	yes	no	String	Modified At Example: 2019-04-29 21:18:28
CreateTime	yes	no	String	Creation time. Example: 2019-04-29 21:18:28
SAMLMetadata	yes	no	String	SAML metadata. Example: ""
SAMLEntityId	yes	no	String	SAML instance ID. Example: ""
SAMLSingleSignOn	yes	no	String	SAML login redirect. Example: ""
SAMLSingleLogout	yes	no	String	SAML logout redirect. Example: ""
SAMLKeys	yes	no	String	SAML keyword. Example: ""
Cas	yes	no	String	Cas Example: ""
CasRoot	yes	no	String	CAS root URL. Example: https://cas.yfm13.fsphere.cn/

name	required	NULL allowed	type	description
CasLoginUrl	yes	no	String	CAS login URL. Example: https://cas.yfm13.fsphere.cn/cas/login
CasValidateUrl	yes	no	String	CAS verification URL. Example: https://cas.yfm13.fsphere.cn/cas/serviceValidate
CasLogoutUrl	yes	no	String	CAS logout URL. Example: https://cas.yfm13.fsphere.cn/cas/logout
Oauth	yes	no	String	OAuth configuration. Example: ""

UserGroup

User and user group information.

Referenced by the following interfaces: GetUserGroupList, ListAllUserGroup

name	required	NULL allowed	type	description
Uid	yes	yes	Uint64	Recipient user ID. Example: 1258042
Uin	yes	yes	Uint64	Unique ID of the account. Example: 110000000001
Name	yes	yes	String	User Name Example: name
IsReceiverOwner	yes	yes	Int64	Whether the recipient is a root account. Example: 1
Group	yes	yes	Array of GroupMeta	User Group information Example:View

AccountAttributeAndValue

Extended Attribute Value

Referenced by the following interfaces: GetGroupList, GetGroupsSubAccount, GetSubsGroup, ListGroups, ListGroupsForConsole, ListMaskedSubAccounts, ListSubAccounts

name	required	NULL allowed	type	description
AttributeName	no	no	String	Attribute name. Example: ""
Attribute	no	no	String	Attribute. Example: ""
AttributeId	no	no	Int64	Attribute ID Example: ""
ValueId	no	no	Int64	Value ID Example: ""
Uin	no	no	Int64	uin Example: 110000000001
Value	no	no	String	Value Example: ""

ServiceApiListInfo

API information of the service.

Referenced by the following interfaces: GetServiceApiList

name	required	NULL allowed	type	description
Name	yes	no	String	API name. Example: CreateInstance
IsNeedObject	yes	yes	String	Whether it needs to be associated with an object. Example: ["1"]
Desc	yes	yes	String	Description. Example: Create Instance
ReadWriteDetail	yes	yes	Uint64	API type. 0: read; 1: write; 2: tag; 3: list. Example: 1
InterfaceLevel	yes	yes	Uint64	Authorization granularity. 0: API level; 1: resource level. Example: 1
ResourceExample	yes	yes	String	Example of resource in 6-segment format. Example: [qcs::cvm:region:uin/110000001207:volume/*]

AttachedStrategyInfo

Policy information

Referenced by the following interfaces: DescribeAttachedEntityPolicies

name	required	NULL allowed	type	description
PolicyId	no	no	Uint64	Policy ID. Example: 1000
PolicyName	no	no	String	Policy name. Example: policygen-20141112201913
AddTime	no	yes	Datetime	Policy creation time. Example: 2014-08-03 12:00:00
CreateMode	no	yes	Uint64	Creation source. 1: created via the console; 2: created via policy syntax. Example: 1
Description	no	yes	String	Policy description. Example: adesc

AttachedStrategyInfoPack

Policy information package.

Referenced by the following interfaces: DescribeAttachedEntityPolicies

name	required	NULL allowed	type	description
List	no	no	Array of AttachedStrategyInfo	Policy array. Each member of the array includes the policyId, policyName, addTime, type, description, createMode fields. Where: Example:View
TotalNum	no	no	Uint64	Number of policies. Example: 100
Id	no	no	String	When the input parameter Type is 1, this field refers to the UIN. When the input parameter Type is 2, this field refers to the groupId. Example: 100

ListGroupUserInfo

ListGroupUserInfo

Referenced by the following interfaces: DescribeGroups, ListUsersForGroup

name	required	NULL allowed	type	description
CountryCode	no	no	String	CountryCode Example: 86
CreateTime	no	no	String	CreateTime Example: 2025-01-17 14:21:07
Email	no	no	String	Email Example: mail@mail.com
EmailFlag	no	yes	Int64	EmailFlag Example: 0
IsReceiverOwner	no	no	Int64	IsReceiverOwner Example: 0
JoinTime	no	no	String	JoinTime Example: 2025-01-17 14:21:07
Name	no	no	String	Name Example: Name
NickName	no	no	String	NickName Example: NickName
PhoneFlag	no	no	Int64	PhoneFlag Example: 0
PhoneNum	no	no	String	PhoneNum Example: 11111111111
SystemType	no	no	Int64	SystemType Example: 0
Uid	no	no	Uint64	Uid Example: 1258683
Uin	no	no	Uint64	Uin Example: 130000000640
UserType	no	no	Int64	UserType Example: 3
QywxUserId	no	no	String	QywxUserId Example: ""

UserData

User Information.

Referenced by the following interfaces: GetAllSubUser

name	required	NULL allowed	type	description
Uid	yes	no	Uint64	Sub-user ID. Example: 1258042
Uin	yes	no	Uint64	Account UIN. Example: 110000000001
Name	yes	no	String	Username. Example: name
PhoneNum	yes	no	String	Mobile number. Example: 11111111111
CountryCode	yes	no	String	Area code. Example: 86
PhoneFlag	yes	no	Int64	Mark for verifying a phone number. Example: 1
Email	yes	no	String	Email addresses. Example: mail@mail.com
EmailFlag	yes	no	Int64	Specifies whether an email address is verified. Example: 1
UserType	yes	no	Int64	User type. Example: 0
CreateTime	yes	no	String	Creation time. Example: 2019-04-29 21:18:28
WechatFlag	yes	no	Int64	Wechat identifier. Example: 0
SystemType	yes	no	String	Type of an account system. Example: Subaccount
IsReceiverOwner	yes	no	Int64	Whether the account is a root account. Example: 1
PermType	no	no	Array of String	PermType Example: ["0"]

AddSubAccountDetail

New User Details

Referenced by the following interfaces: AddSubAccount

name	required	NULL allowed	type	description
Name	no	no	String	User Name Example: name
Token	no	no	String	token Example: ""
Password	no	no	String	Password. Example: password
SecretId	no	no	String	secretid Example: ""
SecretKey	no	no	String	secretKey Example: ""
NickName	no	no	String	Nickname. Example: nickname
Uin	no	no	Uint64	uin Example: 110000000001
Nickname	no	no	String	Nickname Example: nickname

UserList

Sub-account list.

Referenced by the following interfaces: ListUsersForPolicy

name	required	NULL allowed	type	description
Name	yes	no	String	Sub-account username. Example: admin
SubAccountUin	yes	no	String	Sub-account UIN. Example: 1000

AttributeInfo

Attribute.

Referenced by the following interfaces: AddAttributeValues

name	required	NULL allowed	type	description
AttributeName	no	yes	String	Attribute name. Example: ""
Attribute	no	yes	String	Attribute. Example: ""
AttributeValue	no	yes	String	Attribute value. Example: ""

UserLists

GetUserListByUinList

Referenced by the following interfaces: GetUserListByUinList

name	required	NULL allowed	type	description
CountryCode	no	no	String	CountryCode Example: 86
IsAuthed	no	no	Int64	IsAuthed Example: 0
UserCamUid	no	no	Int64	UserCamUid Example: 1280504
UserCellphone	no	no	String	UserCellphone Example: 11111111111
UserEmail	no	no	String	UserEmail Example: mail@mail.com
UserId	no	no	String	UserId Example: 1280504
UserUin	no	no	Uint64	UserUin Example: 110000000001
UserName	no	no	String	UserName Example: username
UserIsLocked	no	no	Int64	UserIsLocked Example: 1

AccountDetail

Account details.

Referenced by the following interfaces: AddSubAccount, ListGroups

name	required	NULL allowed	type	description
ActionFlag	no	no	ActionLoginFlag	Flag for sensitive operations. Example:View
ConsoleLogin	no	no	String	Whether it is allowed to log in to the console. 0: No; 1: Yes. Example value: 1 Example: 1
LoginFlag	no	no	ActionLoginFlag	Login protection. Example:View
NeedResetPassword	no	no	String	Whether it is required to reset the password. 0: No; 1: Yes. Example value: 0 Example: 0
Password	no	no	String	User password. Example value: password Example: password
UseApi	no	no	String	Whether to use APIs. 0: No; 1: Yes. Example value: 1 Example: 1
TokenType	no	no	Int64	Whether to assign a device type. 0: No; 2: Yes. Example value: 0 Example: 0

AttachedUserPolicyGroupInfo

Information of the policies (managed via group) that are associated with the user.

Referenced by the following interfaces: ListAttachedUserAllPolicies

name	required	NULL allowed	type	description
GroupId	no	no	Uint64	Group ID. Example: 10012

name	required	NULL allowed	type	description
GroupName	no	no	String	Group name. Example: group_one

ActionLoginFlag

Flag for sensitive login operations.

Referenced by the following interfaces: AddSubAccount, ListGroups

name	required	NULL allowed	type	description
Phone	no	no	String	Telephone. 0: sensitive operation protection not enabled; 1: sensitive operation protection enabled. Example value: 0 Example: 0
Stoken	no	no	String	Soft token. 0: sensitive operation protection not enabled; 1: sensitive operation protection enabled. Example value: 0 Example: 0
Token	no	no	String	Hard token. 0: sensitive operation protection not enabled; 1: sensitive operation protection enabled. Example value: 0 Example: 0
Ukey	no	no	String	ukey. 0: sensitive operation protection not enabled; 1: sensitive operation protection enabled. Example value: 0. Example: 0

RoleInfo

Role details

Referenced by the following interfaces: DescribeRoleList, GetRole

name	required	NULL allowed	type	description
RoleId	yes	no	String	Role ID. Example: 12
RoleName	yes	no	String	Role name. Example: QCS
PolicyDocument	yes	no	String	Policy document of the role. Example: {\"version\":\"2.0\",\"statement\":[{\"effect\":\"allow\",\"action\":[\"cvm:ModifyDiskAttributes\"],\"resource\":[\"*\"],\"condition\":{\"ip_equal\":{\"qcs:ip\":[\"1.1.1.1\"]}}},{\"effect\":\"allow\",\"action\":[\"cos:PutObject\"],\"resource\":[\"qcs::cos::uid/1255000018:rules-package-ftp-1255000018/waf_rules.zip\"],\"condition\":{\"ip_equal\":{\"qcs:ip\":[\"1.1.1.1\"]}}}]}
Description	yes	no	String	Role description. Example: Description
AddTime	yes	no	String	Creation time of the role. Example: 2020-01-01 01:01:01
UpdateTime	yes	no	String	Last update time of the role. Example: 2020-01-01 01:01:01
DeletionTaskId	no	yes	String	Compatible with public cloud field, no meaning Example: 1
ConsoleLogin	yes	no	Uint64	Whether the role is allowed to log in. 1: yes; 0: no. Example: 1
RoleType	no	yes	String	Role Type: user, system, or service_linked. Example: user
SessionDuration	no	yes	Uint64	Valid time Example: 7200

ResourceTypeItem

Resource Type

Referenced by the following interfaces: GetServiceList

name	required	NULL allowed	type	description
ResourceEnName	no	no	String	English Name Example: ""
ResourceName	no	no	String	Name Example: ""
ResourceType	no	no	String	Resource Type Example: ""
ServiceType	no	no	String	Service Type Example: ""

AttachPolicyInfo

Associated policy

Referenced by the following interfaces: ListAttachedGroupPolicies

name	required	NULL allowed	type	description
PolicyId	yes	no	Uint64	Policy ID. Example: 1000
PolicyName	yes	yes	String	Policy name Example: policygen-20141112201913
AddTime	yes	yes	Datetime	Creation time. Example: 2014-08-03 12:00:00
CreateMode	yes	yes	Uint64	Creation source. 1: created via the console; 2: created via policy syntax. Example: 1

StrategyInfo

Policy information

Referenced by the following interfaces: ListPolicies

name	required	NULL allowed	type	description
PolicyId	yes	no	Uint64	Policy ID. Example: 1
PolicyName	yes	no	String	Policy name. Example: AdministratorAccess
AddTime	yes	yes	Datetime	Policy creation time. Example: 2016-06-02 19:40:09
Type	yes	no	Uint64	Policy type. 1: custom policy; 2: preset policy. Example: 1
Description	yes	yes	String	Policy description. Example: This policy allows you to manage all users and their permissions, related financial info, and cloud service assets under this account.
CreateMode	yes	no	Uint64	Creation source. 1: created via the console; 2: created via policy syntax. Example: 1
Attachments	no	no	Uint64	number of associated user Example: 1
ServiceType	yes	yes	String	product associate with the policy Example: cam
IsAttached	no	yes	Uint64	not null when you need to query whether the marked entity is associated with the policy. 0: not associated with the policy; 1: associated with the policy. Example: 1
Deactived	no	yes	Uint64	Deactivation status. 1: deactivated; 0: not deactivated. Example: 0
DeactivedDetail	no	yes	Array of String	List of deactivated products. Example: ["deacproduct"]
IsCheck	yes	yes	Uint64	Whether security verification is performed. 1: yes; 0: no. Example: 1
PolicyDocument	no	no	String	Policy Syntax Example: {\"version\":\"2.0\",\"statement\":[{\"effect\":\"allow\",\"action\":[\"cvm:ModifyDiskAttributes\"],\"resource\":[\"*\"],\"condition\":{\"ip_equal\":{\"qcs:ip\":[\"1.1.1.1\"]}}},{\"effect\":\"allow\",\"action\":[\"cos:PutObject\"],\"resource\":[\"qcs::cos::uid/1255000018:rules-package-ftp-1255000018/waf_rules.zip\"],\"condition\":{\"ip_equal\":{\"qcs:ip\":[\"1.1.1.1\"]}}}]}
UpdateTime	no	no	String	Modified At Example: 2016-06-02 19:40:09

AttachedUserPolicy

Details of policies associated with the user

Referenced by the following interfaces: ListAttachedUserAllPolicies

name	required	NULL allowed	type	description
PolicyId	no	no	Uint64	Policy ID. Example: 1000
PolicyName	no	no	String	Policy name. Example: policygen-20141112201913
Description	no	no	String	Policy description. Example: adesc
AddTime	no	no	String	Creation time. Example: 2014-08-03 12:00:00
StrategyType	no	no	Uint64	Policy type (1: custom policy; 2: preset policy). Example: 1
CreateMode	no	no	Uint64	Creation mode (1: created by product or project permission; other values: created by policy syntax). Example: 1
Groups	no	yes	Array of AttachedUserPolicyGroupInfo	Information of association via group. Example:View

DescribeGroupsInfo

DescribeGroupsInfo

Referenced by the following interfaces: DescribeGroups

name	required	NULL allowed	type	description
Channel	no	no	Int64	Channel Example: 3
CreateTime	no	no	String	CreateTime Example: 2025-02-21 11:30:44
GroupId	no	no	Int64	GroupId Example: 100
GroupName	no	no	String	GroupName Example: name
GroupNum	no	no	Int64	GroupNum Example: 3
GroupType	no	no	Int64	GroupType Example: 0
Remark	no	no	String	Remark Example: Remark
UserInfo	no	no	Array of ListGroupUserInfo	UserInfo Example:View

OwnerAccountAttribute

Root account attribute.

Referenced by the following interfaces: UpdateOwnerAccount

name	required	NULL allowed	type	description
Remark	no	no	String	Attribute. Example: remark

Filter

Filter combination.

Referenced by the following interfaces: ListMaskedSubAccounts, ListSubAccounts

name	required	NULL allowed	type	description
Keywords	no	no	Array of FilterItem	Filter combination. Example:View
Operator	yes	no	String	Operator. Example: ""

GroupUidUinInfo

User group and user information.

Referenced by the following interfaces: DeleteSubAccount

name	required	NULL allowed	type	description
Uid	yes	no	Uint64	Sub-user UIDs. Example: 1258042
Uin	yes	no	Uint64	Sub-user UIN. Example: 130000000001
GroupId	yes	no	Int64	User group ID. If there is no group, set this field to -1. Specifying a group ID here means to remove the user from the group. Example: 15094

AttachEntityOfPolicy

The entity associated with the policy

Referenced by the following interfaces: ListEntitiesForPolicy

name	required	NULL allowed	type	description
Id	yes	no	String	Entity ID. Example: 1000001
Name	yes	yes	String	Entity name. Example: policygen-20141112201913
Uin	yes	yes	Uint64	Entity UIN. Example: 3449203261

name	required	NULL allowed	type	description
RelatedType	yes	no	Uint64	Association type. 1: user association; 2: user group association. Example: 1

PasswordRules

Password rules.

Referenced by the following interfaces: GetPasswordRules, UpdatePasswordRules

name	required	NULL allowed	type	description
MinimumLength	yes	no	Int64	Minimum password length. Example: 10
MustContain	yes	no	String	Must contain Example: Aa!
ForcePasswordChange	yes	no	Int64	Password validity period. Example: 0
ReusePasswordLimit	yes	no	Int64	Number of password reuses. Example: 2
RetryPasswordLimit	no	no	Int64	Maximum number of failed login attempts with password. Example: 2
OnlyAdminCanResetPassword	no	no	Int64	Whether only the admin can reset the password. Example: 0
MustNotContainUsername	no	no	Int64	It should not contain the username. Example: 0

GroupInfo

User Group information

Referenced by the following interfaces: GetGroupList, GetGroupsSubAccount, GetSubsGroup, ListGroups, ListGroupsForConsole, ListMaskedSubAccounts, ListSubAccounts

name	required	NULL allowed	type	description
GroupId	no	no	Uint64	Group ID. Example: 15094
GroupName	no	no	String	Group name. Example: groupname
Channel	no	yes	Int64	Channel for receiving messages. 0: none; 1: SMS; 2: email; 3: SMS and email. Example: 0
Remark	no	yes	String	Description Example: remark
CreateTime	no	yes	String	Creation time. Example: 2019-04-29 21:18:28
UserInfo	no	yes	Array of SubAccountFilter	Member information of the user group. Example:View
GroupType	no	yes	Int64	Type of the user group. 0: custom; 1: preset. Example: 0
GroupNum	no	no	Int64	Number of group members. Example: 2

ErrorCode

Feature Description

If there is an Error field in the response, it means that the API call failed. For example:

```
{
    "Response": {
        "Error": {
            "Code": "AuthFailure.SignatureFailure",
            "Message": "The provided credentials could not be validated. Please check your signature is cor
rect."
        },
        "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
    }
}
```

Code in Error indicates the error code, and Message indicates the specific information of the error.

Error Code List

Common Error Codes

Error codes	Description
AuthFailure.InvalidSecretId	Invalid key (not a Cloud API key type).
AuthFailure.MFAFailure	MFA failed.
AuthFailure.SecretIdNotFound	Key does not exist. Check if the key has been deleted or disabled in the console, and if not, check if the key is correctly entered. Note that whitespaces should not exist before or after the key.
AuthFailure.SignatureExpire	Signature expired. Timestamp and server time cannot differ by more than five minutes. Please ensure your current local time matches the standard time.
AuthFailure.SignatureFailure	Invalid signature. Signature calculation error. Please ensure you've followed the signature calculation process described in the Signature API documentation.
AuthFailure.TokenFailure	Token error.

Error codes	Description
AuthFailure.UnauthorizedOperation	The request does not have CAM authorization.
DryRunOperation	DryRun Operation. It means that the request would have succeeded, but the DryRun parameter was used.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidAction	The API does not exist.
InvalidParameter	Incorrect parameter.
InvalidParameterValue	Invalid parameter value.
LimitExceeded	Quota limit exceeded.
MissingParameter	A parameter is missing.
NoSuchVersion	The API version does not exist.
RequestLimitExceeded	The number of requests exceeds the frequency limit.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	Resource is unavailable.
UnauthorizedOperation	Unauthorized operation.
UnknownParameter	Unknown parameter.
UnsupportedOperation	Unsupported operation.
UnsupportedProtocol	HTTPS request method error. Only GET and POST requests are supported.
UnsupportedRegion	API does not support the requested region.

Service Error Codes

Error codes	Description
ResourceNotFound.URLError	URL parsing exception.

Error codes	Description
FailedOperation.AddIdentityError	Failed to add identity.
FailedOperation.NameAlreadyExist	Username already exists
FailedOperation.AccountSettingConfigError	Exceptional metadata is configured for the account.
InvalidParameter.ServiceTypeError	ServiceType field invalid.
ResourceNotFound.NotFound	The resource does not exist.
ResourceNotFound.GroupNotExist	The user group does not exist.
InvalidParameter.PolicyDocumentLengthOverLimit	PolicyDocument field exceeds length restriction.
InvalidParameter.PolicyIdNotExist	Policy ID does not exist.
FailedOperation.SetLoginRuleFail	Failed to set a login policy.
FailedOperation.SubAccountHasKey	The sub-account has a key.
FailedOperation.PolicyFull	The number of user policies has exceeded the limit.
FailedOperation.SkeyExpired	Skey has expired.
InvalidParameter.RoleNotExist	Role does not exist.
InvalidParameter.ActionNotExist	The action does not exist.
FailedOperation.NoneValue	Configuration is missing.
FailedOperation.PolicyNameInUse	The policy name specified by the PolicyName field already exists.
FailedOperation.IdentityExist	Identity authentication failed.
InvalidParameter.PasswordLengthTooShort	Password too short.
FailedOperation.InAsyncModifyError	Data modification is in progress.
InvalidParameter.PolicyDocumentError	PolicyDocument field invalid.
ResourceNotFound.PolicyIdNotFound	Resources specified by PolicyId does not exist.
InvalidParameter.OperatePoliciesOverLimit	Too many policies in one operation.
InvalidParameter.GroupIdError	GroupId field invalid.

Error codes	Description
FailedOperation.UnknownAccountSettingKey	Unknown account configuration item.
ResourceNotFound.RecordNotExist	The record does not exist
FailedOperation.UserrAccessTokenError	An error occurred when accessing token.
InvalidParameter.NotSupportProduct	CAM does not support the resource type specified in the policy document.
FailedOperation.AccountGroupNameNotMatch	The group cannot match the configured group name.
InvalidParameter.PolicyIdError	Input parameter PolicyId invalid.
InvalidParameter.DescriptionLengthOverlimit	The input parameter description cannot exceed 300 bytes in length.
FailedOperation.UnknownAccountSettingGroup	Unknown account configuration group.
FailedOperation.Accesskey	An error occurred while manipulating the access key
ResourceUnavailable.IDPMaxLimit	Reached the maximum number of idPs allowed.
InvalidParameter.ParamError	Invalid input parameter.
InvalidParameter.ActionError	Action field of policy document invalid.
UnauthorizedOperation	Unauthorized operation.
InvalidParameter.RoleNameError	Role name invalid.
ResourceNotFound.IdentityNotExist	The Identity does not exist.
InvalidParameter.ResourceError	Resource field of policy document invalid.
FailedOperation.CheckPasswordError	Password checking failed.
InvalidParameter.RoleNameInUse	A role with the same name already exists.
InvalidParameter.CreateGroupErr	Failed to create the user group
InvalidParameter.EffectError	Effect field of policy document invalid.
InvalidParameter.KeywordError	Keyword field invalid.
InvalidParameter.VersionError	Version field of policy document invalid.
InvalidParameter.PolicyNameError	PolicyName field invalid.

Error codes	Description
InvalidParameter.ConditionError	condition field of policy document invalid.
FailedOperation.RefreshTokenError	An error occurred when refreshing user token.
InvalidParameter.UinError	Uin field invalid.
ResourceNotFound.UserNotExist	User does not exist.
InvalidParameter.ScopeError	Scope field invalid.
InvalidParameter.StatementError	Statement field of policy document invalid.
InternalError.SystemError	Internal error. Please check and try again.
FailedOperation.SecretIdExist	SecretId already exists.
InvalidParameter.UserNotExist	User object does not exist
FailedOperation.AuthCodeError	Authorization code exception.
InvalidParameter.RoleFull	Limit on number of roles reached.
InvalidParameter.PrincipalError	principal field of policy document invalid.
FailedOperation.AccountSettingValueCalculateError	An exception occurred while calculating a configuration value.
InvalidParameter.AttachmentFull	Limit on number of authorization object associated policies in the principal field reached.