

# Cloud Infinite

## Developer Manual

### Product Documentation



## Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice

 Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

---

# Contents

## Developer Manual

### Access Management

#### Overview

#### Authorization Granularity Details

#### Authorizing Sub-Accounts to Access CI Services

#### Configuring Data Persistence Permissions for Sub-Accounts

### Request Signature

### Monitoring and Alarms

# Developer Manual

## Access Management

### Overview

Last updated : 2024-01-31 16:54:06

You can create users or roles by using [Cloud Access Management \(CAM\)](#). By associating CAM preset or custom policies, you can specify which users have which access permissions to which resources. You can also specify that access permissions are only available under certain conditions.

Cloud Infinite (CI) has two major modules: image processing and media processing. For image processing, you can configure persistent permissions for your sub-account to process data when uploading or process COS data already stored in the cloud. To grant permissions to your sub-account to process data when downloading, please [submit a ticket](#) to contact us. For media processing, CI allows you to grant permissions at the resource level. You can grant permissions to allow a sub-account to manage a single resource through policy syntax.

See the basic concepts below. For more information, see the CAM [User Guide](#) documentation.

## Accounts

**Root account:** this account owns all the Tencent Cloud resources and has full access to these resources.

**Sub-account:** includes sub-users and collaborators.

**Sub-user:** created by a root account and is subordinate to the root account.

**Collaborator:** when a root account is added as the collaborator of the current root account, it becomes one of the sub-accounts of the current root account. The account can be switched from collaborator back to root account.

**Identity credentials:** include login credentials and access certificates.

**Login credentials:** refers to usernames and passwords.

**Access certificate:** refers to the Cloud API key pairs (SecretId and SecretKey).

## Resources and Permissions

**Resource:** an object that you can operating on, such as a COS bucket or an image in CI.

**Permission:** an authorization to allow or forbid some users to perform certain operations. By default, the root account has full access to all resources under it, while a sub-account has no access to any resources under its root account.

**Policy:** a set of syntax rules that define and describe one or more permissions. A root account grants access for users or user groups by associating policies with them. For more information, see [Access Policy Language Overview](#).

# Authorization Granularity Details

Last updated : 2024-01-31 16:54:06

You can grant a specified sub-account the API permission of a specified resource in CI. The authorization granularity of CI is divided into resource level and API level:

Resource-level API: It supports the authorization of a specific resource.

API-level API: It does not support the authorization of a specific resource.

For authentication through a resource-level API, CI will pass the specific 6-segment resource description to CAM for authentication, thereby supporting authorization and authentication of the specific resource.

For authentication through an API-level API, CI will not deliver the six-segment format of the specific resource to CAM for authentication. Instead, it will deliver the \* of any resource. To be more specific, if a resource is specified in the policy syntax during authorization but not delivered through the API for authentication, CAM will identify the API as out of the authorization scope, that is, having no permission.

The following lists all API permissions of CI:

## List Operation

API	Description	Authorization Granularity	6-Segment Resource Description
DescribeMediaBuckets	Queries buckets with media processing enabled	API level	*
DescribeRiskLibImage	Queries the list of images in the risky image library preset for content moderation	API level	*
DescribeRiskLibText	Queries the list of keywords in the risky text library preset for content moderation	API level	*

## Write Operation

### Basic features

--	--	--	--

API	Description	Authorization Granularity	6-Segment Resource Description
CreateCIBucket	Binds a bucket to CI	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
DeleteCIBucket	Unbinds a bucket from CI	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
SetDomain	Sets a CI domain name	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
SetCDNAccelerate	Sets a CDN acceleration domain name for CI	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
Set4XXResponse	Sets the 4xx image configuration in CI	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
SetRefer	Sets the bucket hotlink protection feature in CI	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
SetOriginProtect	Sets the original image protection feature in CI	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*

## Content moderation

API	Description	Authorization Granularity	6-Segment Resource Description
SetAuditingPicture	Sets automatic image moderation	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
SetAuditingVideo	Sets	Resource	qcs::ci:ap-

	automatic video moderation	level	shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
SetAuditingAudio	Sets automatic audio moderation	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
SetAuditingText	Sets automatic text moderation	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
SetAuditingDocument	Sets automatic file moderation	Resource level	qcs::ci:uid/\${appid}:bucket/examplebucket-1250000000/*
CreateAuditingJobs	Creates a video moderation job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
CreateAuditingPictureJob	Calls image moderation	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
CreateAuditingTextJob	Creates a text moderation job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
CreateAuditingAudioJob	Creates an audio moderation job	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/examplebucket/1250000000/*
CreateAuditingWebpageJob	Creates a webpage moderation job	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/\${bucket}/*
CreateAuditingDocumentJob	Creates a file moderation job	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/examplebucket/1250000000/*

CreateAuditingExistTask	Creates an existing data moderation job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
CancelAuditingExistTask	Cancels an existing data moderation job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
DeleteRiskLibText	Deletes the risky text library preset for content moderation	API level	*
DeleteRiskLibImage	Deletes an image in the risky image library preset for content moderation	API level	*
CreateRiskLibText	Adds a keyword to the risky text library for content moderation	API level	*
CreateRiskLibImage	Adds an image to the risky image library preset for content moderation	API level	*
CreateAuditingVirusJob	Creates a cloud virus	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/\${bucket}/*



	detection job		
--	------------------	--	--

## Media processing

API	Description	Authorization Granularity	6-Segment Resource Description
CreateMediaBucket	Enables media processing for a bucket	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
DeleteMediaBucket	Disables media processing for a bucket	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
CreateMediaTemplate	Creates a media processing template	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
UpdateMediaTemplate	Updates a media processing template	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
DeleteMediaTemplate	Deletes a media processing template	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
UpdateMediaQueue	Updates a media processing queue	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
CreateMediaJobs	Creates a media processing job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
CancelMediaJob	Cancels a media	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*

	processing job		
GenerateSnapshot	Captures video frames	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
GenerateMediaInfo	Queries the video information	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
DeleteMediaWorkflow	Deletes a workflow	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
UpdateMediaWorkflow	Updates a workflow	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
CreateMediaWorkflow	Creates a workflow	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*

## Image processing

API	Description	Authorization Granularity	6-Segment Resource Description
SetImageGuetzli	Enables image Guetzli compression	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example1250000000/*
SetImageAdvancedCompression	Enables image advanced compression	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example1250000000/*
SetImageBlindWatermark	Enables image blind watermarking	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example1250000000/*
SetImageStyleSeparator	Sets the image style separator	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example1250000000/*
SetImageStyle	Sets the	Resource	qcs::ci:ap-

	image style	level	shanghai:uid/1250000000:bucket/example-1250000000/*
--	-------------	-------	---

## File processing

API	Description	Authorization Granularity	6-Segment Resource Description
CreateDocProcessBucket	Enables file preview for a bucket	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
UpdateDocProcessQueue	Updates a file preview queue	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
DeleteDocProcessBucket	Disables file preview for a bucket	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
CancelDocProcessJob	Cancels a file preview job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
CreateDocProcessJobs	Creates a file preview job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*

## AI-based content recognition

API	Description	Authorization Granularity	6-Segment Resource Description
CreateDetectQRcodeJob	Calls the QR code recognition API	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
CreateDetectLabelJob	Calls the image tag recognition API	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
CreateFaceEffectJob	Calls the face	Resource	qcs::ci:ap-

	filter API	level	shanghai:uid/1250000000:bucket/example:1250000000/*
UpdateAsrQueue	Updates a speech recognition queue	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
DeleteAsrBucket	Disables speech recognition for a bucket	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
CreateAsrBucket	Enables speech recognition for a bucket	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
CreateAsrJobs	Creates a speech recognition job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
CreateGetActionSequenceJob	Calls the face verification - motion sequence acquisition API	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
CreateGetLiveCodeJob	Calls the face verification - verification code acquisition API	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
CreateLivenessRecognitionJob	Calls the face verification API	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
CreateIDCardOCRJob	Calls the ID card recognition API	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
CreateQRcodeGenerateJob	Calls the QR	Resource	qcs::ci:ap-

	code generation API	level	shanghai:uid/1250000000:bucket/example:1250000000/*
TriggerMediaWorkflow	Triggers a workflow	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
CreatePetDetectJob	Calls the pet recognition API	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
DeleteImage	Deletes an image in the image search library	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
AddImage	Adds an image to the image search library	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
CreateImageSearchBucket	Enables a bucket as the image search library	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/example:1250000000/*
CreateContentAnalysisJob	Calls the image tag recognition API	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/example:1250000000/*
CreateBodyJointsDetectJob	Calls the body keypoint analysis API	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/example:1250000000/*
CreateOCRJob	Calls the image OCR API	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/example:1250000000/*
CreateLicensePlateJob	Calls the license plate recognition API	Resource level	qcs::ci:uid/\${appid}:bucket/examplebucket-1250000000/*
CreateDetectFaceJob	Calls the face	Resource	qcs::ci:\${region}:uid/\${appid}:bucket/example

	detection API	level	1250000000/*
CreateAssessQualityJob	Calls the image quality analysis API	Resource level	qcs::ci::uid/\${appid}:bucket/examplebucket-1250000000/*
CreateAiRecognitionJob	Calls the all-in-one AI API	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/examp1250000000/*
CreateEnhanceImageJob	Calls the image enhancement API	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/\${buck
CancelPicProcessJob	Cancels an image processing job	Resource level	qcs::ci::uid/\${appid}:bucket/examplebucket-1250000000/*
CreatePicProcessJobs	Creates an image processing job	Resource level	qcs::ci::uid/\${appid}:bucket/examplebucket-1250000000/*
UpdatePicProcessQueue	Updates an image processing job queue	Resource level	qcs::ci::uid/\${appid}:bucket/examplebucket-1250000000/*
DeletePicProcessBucket	Disables image processing for a bucket	Resource level	qcs::ci::uid/\${appid}:bucket/examplebucket-1250000000/*
CreatePicProcessBucket	Enables image processing for a bucket	Resource level	qcs::ci::uid/\${appid}:bucket/examplebucket-1250000000/*

## Read Operation

### Basic features

API	Description	Authorization Granularity	6-Segment Resource Description
-----	-------------	---------------------------	--------------------------------

DescribeCIBuckets	Queries the CI binding status of a bucket	API level	*
DescribeDomain	Queries the information of a CI domain name	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
DescribeOriginProtect	Queries the enablement status of the original image protection feature in CI	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
DescribeCDNAccelerate	Queries the binding status of a CDN acceleration domain name in CI	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
Describe4XXResponse	Queries the status of 4xx image configuration in CI	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
DescribeRefer	Queries the status of bucket hotlink protection in CI	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*

## Content moderation

API	Description	Authorization Granularity	6-Segment Resource Description
DescribeAuditingJobs	Queries	Resource	qcs::ci:ap-

	video moderation jobs	level	shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingVideo	Queries whether automatic video moderation is enabled	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingAudio	Queries whether automatic audio moderation is enabled	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingPicture	Queries whether automatic image moderation is enabled	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingText	Queries whether automatic text moderation is enabled	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingDocument	Queries whether automatic file moderation is enabled	Resource level	qcs::ci::uid/\${appid}:bucket/examplebucket-1250000000/*
DescribeAuditingCallback	Queries the callback permission for content moderation	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingBlockRule	Queries	Resource	qcs::ci:ap-



	the blocking rule permission for content moderation	level	shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingPictureFiles	Queries the list of files for image moderation	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingVideoFiles	Queries the list of files for video moderation	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingAudioFiles	Queries the list of files for audio moderation	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingTextFiles	Queries the list of files for text moderation	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingExistTasks	Queries historical data moderation jobs	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingTextJob	Queries the result of a text moderation job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebu1250000000/*
DescribeAuditingAudioJob	Queries the result of an audio moderation job	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/example1250000000/*

DescribeAuditingDocumentJob	Queries the result of a file moderation job	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/example1250000000/*
DescribeAuditingWebpageJob	Queries the result of a webpage moderation job	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/\${bucke
DescribeAuditingDocumentFiles	Queries the result of a file moderation job	Resource level	qcs::ci:uid/\${appid}:bucket/examplebucket-1250000000/*
DescribeAuditingVirusJob	Queries the result of a cloud virus detection job	Resource level	qcs::ci:\${region}:uid/\${appid}:bucket/\${bucke

## Media processing

API	Description	Authorization Granularity	6-Segment Resource Descriptor
DescribeMediaTemplates	Queries media processing templates	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucke1250000000/*
DescribeMediaQueues	Queries media processing queues	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucke1250000000/*
DescribeMediaJob	Queries a media processing job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucke1250000000/*

DescribeMediaJobs	Queries the list of media processing jobs	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
DescribeMediaWorkflowExecutions	Queries workflow execution instances	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
DescribeMediaWorkflows	Queries the status of workflows	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
GetPrivateM3U8	Generates a private M3U8 video	API level	*
GenerateMediaInfo	Gets the media information	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
GenerateSnapshot	Calls a frame capturing job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
ModifyM3U8Token	Refreshes HLS encrypted token	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
CreateMediaTemplate	Creates a media processing template	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
UpdateMediaTemplate	Updates a media processing template	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
DeleteMediaTemplate	Deletes a media	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*

	processing template		
CreateMediaWorkflow	Creates a media processing workflow	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
UpdateMediaWorkflow	Updates a media processing workflow	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
DeleteMediaWorkflow	Deletes a media processing workflow	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
TriggerMediaWorkflow	Triggers a media processing workflow	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
UpdateMediaQueue	Updates a media processing queue	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
AddMediaQueue	Adds a media processing queue	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
CreateMediaBucket	Binding the media processing service	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
DescribeMediaBuckets	Queries the status of the media processing service	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
DeleteMediaBucket	Unbinding the media	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*

	processing service		
CreateMediaJobs	Submits a media processing job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
CancelMediaJob	Cancels a media processing job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
CreateInventoryTriggerJob	Initiates a batch media processing job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
DescribeInventoryTriggerJob	Queries a batch media processing job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
DescribeInventoryTriggerJobs	Queries the status batch operation jobs in batches	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
CancelInventoryTriggerJob	Cancels a batch operation job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
UpdateSpeedTranscodingProcessQueue	Updates an accelerated transcoding queue	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
DescribeSpeedTranscodingProcessQueues	Queries accelerated transcoding queues	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*

CreateSpeedTranscodingProcessQueues	Creates an accelerated transcoding queue	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/1250000000/*
-------------------------------------	--	----------------	--

## Image processing

API	Description	Authorization Granularity	6-Segment Resource Description
DescribeImageGuetzli	Queries the enablement status of image Guetzli compression for a bucket	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/ex1250000000/*
DescribeImageAdvancedCompression	Queries the enablement status of image advanced compression for a bucket	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/ex1250000000/*
DescribeImageBlindWatermark	Queries the enablement status of blind watermarking for a bucket	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/ex1250000000/*
DescribeImageStyleSeparator	Queries the image style separator	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/ex1250000000/*
DescribeImageStyles	Queries image styles	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/ex1250000000/*

## File processing

API	Description	Authorization Granularity	6-Segment Resource Description
-----	-------------	---------------------------	--------------------------------

DescribeDocProcessQueues	Queries file preview queues	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
DescribeDocProcessBucket	Queries the enablement status of file preview for a bucket	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
DescribeDocProcessJobs	Queries the list of file preview jobs	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
DescribeDocProcessJob	Queries a file preview job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*

## AI-based content recognition

API	Description	Authorization Granularity	6-Segment Resource Description
DescribeAsrQueues	Queries speech recognition queues	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
DescribeAsrBuckets	Queries the enablement status of speech recognition for buckets	API level	*
DescribeAsrJobs	Queries the list of speech recognition jobs	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*
DescribeAsrJob	Queries a speech recognition job	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket/1250000000/*

SearchImage	Searches for an image in the image search library	Resource level	qcs::ci:ap-shanghai:uid/1250000000:bucket/examplebucket-1250000000/*
DescribePicProcessJobs	Queries the list of image processing jobs	Resource level	qcs::ci::uid/\${appid}:bucket/examplebucket-1250000000/*
DescribePicProcessJob	Queries an image processing job	Resource level	qcs::ci::uid/\${appid}:bucket/examplebucket-1250000000/*
DescribePicProcessQueues	Queries image processing queues	Resource level	qcs::ci::uid/\${appid}:bucket/examplebucket-1250000000/*
DescribePicProcessBucket	Queries the list of buckets for image processing	Resource level	qcs::ci::uid/\${appid}:bucket/examplebucket-1250000000/*



# Authorizing Sub-Accounts to Access CI Services

Last updated : 2024-01-31 16:54:06

Cloud Infinite (CI) provides a suite of data processing services, among which CI storage is based on COS. Therefore, for a sub-account to use all CI services, it must be granted necessary read/write permissions both for CI and COS. Granting CI operation permissions to a sub-account involves three steps: [Create a Sub-Account and Grant CI Permissions](#), [Grant COS permissions to the Sub-Account](#), and [Use the Sub-Account to Process Data](#).

## Note:

A sub-account may need the following permissions to view data or change settings in the CI console:

Operation	Permission
Create a bucket	cos:PutBucket permission for COS bucket
Unbind a bucket	cos>DeleteBucket permission for COS bucket
View feature configuration	cos:GetBucket permission for COS bucket
Modify feature configuration	cos:PutObject permission for COS bucket

## Step 1. Create a Sub-Account and Grant CI Permissions

You can create a sub-account in CAM Console and grant CI access permissions for it. The specific procedure is as follows:

1. Log in to [CAM Console](#). In the left sidebar, choose **Users > User List**.
2. In the user list page, click **Create User**.
3. Click **Custom Create** to open the **Select Type** page.
4. Click **Access to resources and receive messages > Next** to enter the **Enter user info** page.
5. Enter user information. Here, you can create multiple sub-users, set the access type and console password, or perform other operations.
6. Click **Next** to set user permissions. Choose **Select policies from the policy list**, and then **QcloudCIFullAccess** for full CI access from the policy list. Click **Next**.
7. After confirming that the information is correct, click **OK** to create the sub-account.

## Step 2. Grant COS Permissions to the Sub-Account

To grant **COS resource access permissions** to the sub-account, associate a preset policy with it as follows:

1. Log in to [CAM Console](#). In the left sidebar, choose **Users > User List**.
2. In the user list page, find the sub-user that you just created, and click **Grant Permission** under **Operation**.
3. In the policy list, select the appropriate permission policy, and click **OK** to associate it with the sub-user

You can also grant COS permissions to a sub-account using a custom policy. For more information, see the [Authorization Management](#) document and policy examples.

## Step 3. Use the Sub-Account to Process Data

To use a sub-account to process data, you need the APPID of the root account and the SecretId and SecretKey of the sub-account.

1. Log in to [CAM Console](#) with your root account. In the left sidebar, choose **Users > User List**.
2. Click on the name of the sub-account whose SecretId and SecretKey you want.
3. Select the **API Key** tab where you can get the SecretId and SecretKey.

You can also get the API Key from the CAM console using a sub-account by granting it CAM read permission.

# Configuring Data Persistence Permissions for Sub-Accounts

Last updated : 2024-01-31 16:54:06

Currently, you can grant permissions to a Cloud Infinite (CI) sub-account to perform persistence operations by associating the write permission with given resources in COS. The following cases illustrate how to grant permissions to a CI sub-account to perform persistence operations on **all resources** and on **specific resources**.

## Note:

Before configuring data persistence permissions for your sub-account, you must associate the CI full read-write access **QcloudCIFullAccess**.

When configuring a custom policy, you can copy and paste the following reference policy into the **Edit Policy Content** input box and modify it based on the actual settings. For more information, see the [CAM Policy Syntax](#) document.

## Authorizing a sub-account to perform persistence operations on all resources

Assume that an enterprise account named CompanyExample (whose OwnerUin is 100000000001 and APPID is 1250000000) has a sub-account named Developer, which needs to perform persistence processing on all resources under CompanyExample.

In this case, the following solutions are available to authorize the CI sub-account to perform persistence processing by granting the write permission to all resources under the account CompanyExample in COS.

### Solution A:

CompanyExample grants the **QcloudCOSDataWriteOnly** preset policy to Developer. For more information on authorization, see [Authorization Management](#).

### Solution B:

Step 1: create the following policy by using policy syntax.



```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "action": [
        "cos:ListParts",
        "cos:PostObject",
        "cos:PutObject*",
        "cos:InitiateMultipartUpload",
        "cos:UploadPart",

```

```
        "cos:UploadPartCopy",
        "cos:CompleteMultipartUpload",
        "cos:AbortMultipartUpload"
    ],
    "resource": "*"
}
]
```

Step 2: associate this policy with the sub-account. For more information on authorization, see [Authorization Management](#).

## Authorizing a sub-account to perform persistence operations on resources under a specific directory

Assume that an enterprise account named CompanyExample (whose OwnerUin is 100000000001 and APPID is 12500000000) has a sub-account named Developer, which needs to perform persistence processing on resources under the doc directory in a bucket named examplebucket locating in the Shanghai region.

In this case, the following solutions are available to authorize the CI sub-account to perform persistence processing by granting the write permission to resources under the specified directory in COS.

### **Solution A:**

Configure policies and the ACL for resources in COS Console. For more information, see the [Adding Bucket Policies](#) document for COS.

### **Solution B:**

Step 1: create the following policy by using policy syntax.



```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "action": [
        "cos:ListParts",
        "cos:PostObject",
        "cos:PutObject*",
        "cos:InitiateMultipartUpload",
        "cos:UploadPart",

```

```
        "cos:UploadPartCopy",
        "cos:CompleteMultipartUpload",
        "cos:AbortMultipartUpload"
    ],
    "resource": "qcs::cos:ap-shanghai:uid/1250000000:examplebucket-1250000000"
}
]
```

Step 2: associate this policy with the sub-account. For more information on authorization, see [Authorization Management](#).

## Authorizing a sub-account to perform persistence operations on a specific resource

Assume that an enterprise account named CompanyExample (whose OwnerUin is 100000000001 and APPID is 1250000000) has a sub-account named Developer, which needs to perform persistence processing on the picture.jpg image file under the doc directory in a bucket named examplebucket locating in the Shanghai region.

In this case, the following solutions are available to authorize the CI sub-account to perform persistence processing by granting the write permission to the specified resource in COS.

### **Solution A:**

Configure policies and the ACL for resources in COS Console. For more information, see the [Adding Bucket Policies](#) document for COS.

### **Solution B:**

Step 1: create the following policy by using policy syntax.



```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "action": [
        "cos:ListParts",
        "cos:PostObject",
        "cos:PutObject*",
        "cos:InitiateMultipartUpload",
        "cos:UploadPart",

```



```
        "cos:UploadPartCopy",
        "cos:CompleteMultipartUpload",
        "cos:AbortMultipartUpload"
    ],
    "resource": "qcs::cos:ap-shanghai:uid/1250000000:examplebucket-1250000000/doc/"
}
]
```

Step 2: associate this policy with the sub-account. For more information on authorization, see [Authorization Management](#).

## Authorizing a sub-account to perform persistence operations on a specific resources with a specific prefix

Assume that an enterprise account named CompanyExample (whose OwnerUin is 100000000001 and APPID is 12500000000) has a sub-account named Developer, which needs to perform persistence processing on resources with the prefix of **test** under the doc directory in a bucket named examplebucket locating in the Shanghai region.

In this case, the following solutions are available to authorize the CI sub-account to perform persistence processing by granting the write permission to resources with the specified prefix in COS.

### Solution A:

Configure policies and the ACL for resources in COS Console. For more information, see the [Adding Bucket Policies](#) document for COS.

### Solution B:

Step 1: create the following policy by using policy syntax.



```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "action": [
        "cos:ListParts",
        "cos:PostObject",
        "cos:PutObject*",
        "cos:InitiateMultipartUpload",
        "cos:UploadPart",

```

```
        "cos:UploadPartCopy",
        "cos:CompleteMultipartUpload",
        "cos:AbortMultipartUpload"
    ],
    "resource": "qcs::cos:ap-shanghai:uid/1250000000:examplebucket-1250000000/doc/"
}
]
```

Step 2: associate this policy with the sub-account. For more information on authorization, see [Authorization Management](#).

## Authorizing a sub-account to perform persistence operations on all resources under a specific directory except specified files

Assume that an enterprise account named CompanyExample (whose OwnerUin is 100000000001 and APPID is 12500000000) has a sub-account named Developer, which needs to perform persistence processing on all resources under the doc directory in a bucket except the image file picture.jpg. The bucket is named examplebucket and locates in the Shanghai region.

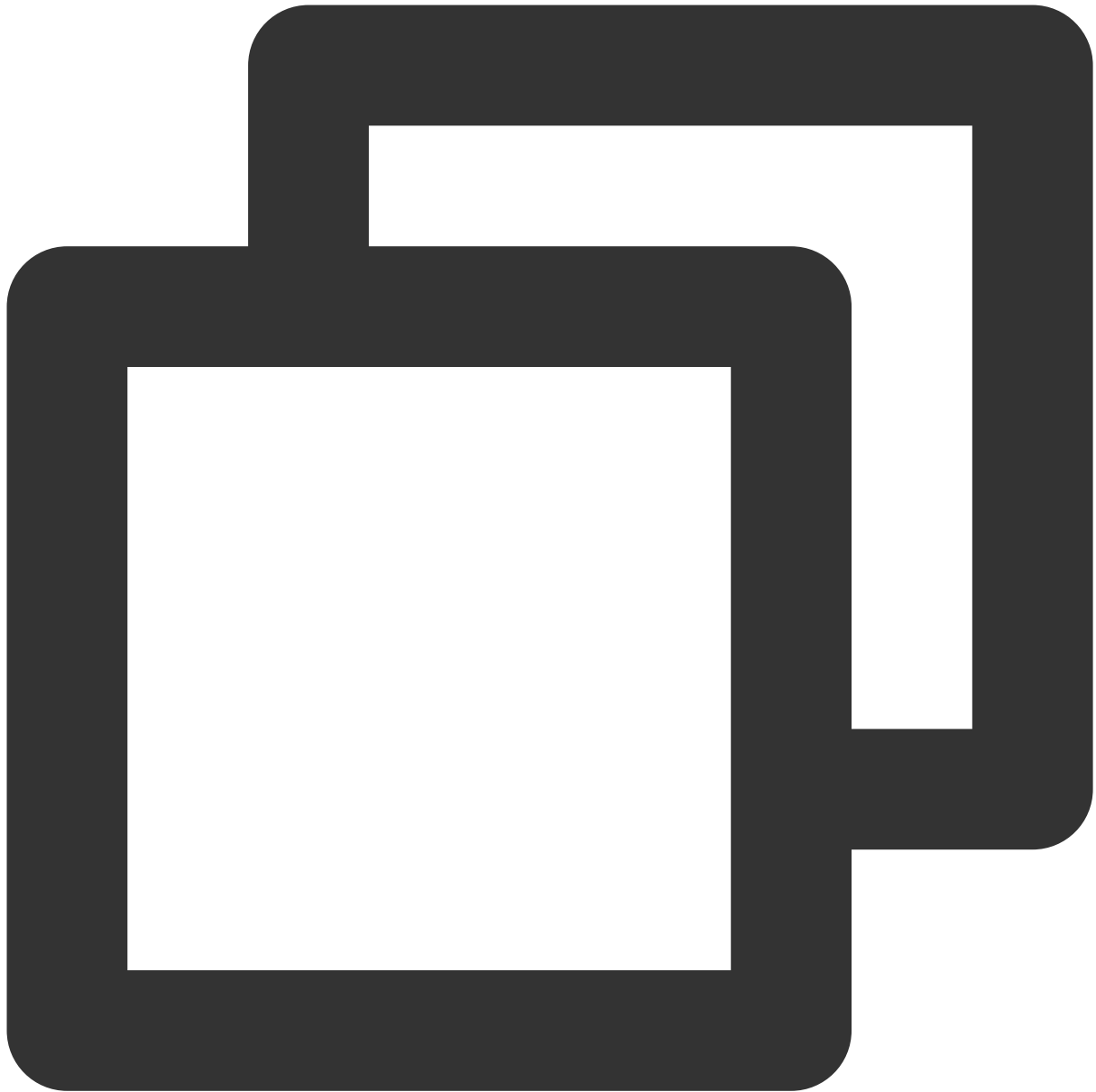
In this case, the following solutions are available to authorize the CI sub-account to perform persistence processing by granting the write permission to a given files under the specified directory in COS.

### **Solution A:**

Configure policies and the ACL in COS Console. For more information, see the [Adding Bucket Policies](#) document for COS.

### **Solution B:**

Step 1: create the following policy by using policy syntax.



```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "action": [
        "cos:ListParts",
        "cos:PostObject",
        "cos:PutObject*",
        "cos:InitiateMultipartUpload",
        "cos:UploadPart",

```

```
        "cos:UploadPartCopy",
        "cos:CompleteMultipartUpload",
        "cos:AbortMultipartUpload"
    ],
    "resource": "qcs::cos:ap-shanghai:uid/1250000000:examplebucket-1250000000",
},
{
    "effect": "deny",
    "action": [
        "cos:ListParts",
        "cos:PostObject",
        "cos:PutObject*",
        "cos:InitiateMultipartUpload",
        "cos:UploadPart",
        "cos:UploadPartCopy",
        "cos:CompleteMultipartUpload",
        "cos:AbortMultipartUpload"
    ],
    "resource": "qcs::cos:ap-shanghai:uid/1250000000:examplebucket-1250000000/doc/"
}
]
```

Step 2: associate this policy with the sub-account. For more information on authorization, see [Authorization Management](#).

# Request Signature

Last updated : 2024-01-31 16:54:06

CI uses signatures to verify the validity of requests. The developer authorizes a signature to the client for it to upload, download, and manage the specified resources.

When using CI, you can call RESTful APIs to initiate anonymous or signed HTTP requests. For signed requests, the server will validate the identity of the requester.

**Anonymous request:** an HTTP request that does not carry any authentication information, and is sent using RESTful APIs.

**Signed request:** an HTTP request that carries a signature. The server will authenticate requesters and only execute requests initiated by authenticated ones. If the authentication fails, the server will return an error message and deny the request.

CI uses the same signature algorithm as [Cloud Object Storage \(COS\)](#) and uses Hash-based message authentication code (HMAC) custom solutions for identity verification.

## Signature algorithm

Currently, CI has integrated with COS, meaning that you can use a COS domain to process images. The signature version required is different for CI and COS domains.

**CI domain:** formatted as `<BucketName-APPID>.<picture region>.myqcloud.com/<picture name>` (for example, `examplebucket-1250000000.picsh.myqcloud.com/picture.jpeg` ). It uses **JSON** signatures.

**COS domain:** formatted as `<BucketName-APPID>.cos.<Region>.myqcloud.com` (for example, `examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/picture.jpeg` ). It uses **XML** signatures. For more information about XML signatures, please see [Request Signature](#).

## Signature scenarios

Scenario		Applicable Signature
Processing data upon download	Hotlink protection not enabled	No signature verification
	Hotlink protection enabled	Signature verification
Processing data upon upload	Persistent processing	XML signatures
Bucket API calls	Query, activation, deletion, etc.	XML signatures
Content recognition	Detecting pornographic, political, or violent/terror content	XML signatures

## Signature tools

To generate a signature, you need the `APPID` (e.g., 1250000000), bucket name (e.g., `examplebucket-ci`), as well as the `SecretID` and `SecretKey` of the project.

The following describes how to obtain the information above:

1. Log in to the [CI console](#) and click **Bucket Management** on the left sidebar.
2. Click the name of the desired bucket.
3. Click **Bucket Configuration** to view the bucket name and bucket ID. If there is no bucket created for the current project, you can create one by referring to [Creating Buckets](#).
4. Log in to the [CAM console](#) and go to **Manage API Key** to get `SecretID` and `SecretKey`.

The signature calculation process of CI is the same as that of COS. You can use a COS signature tool to generate a signature version you need according to the **signature scenario**.

## Using a signature

Signed HTTP requests sent via RESTful APIs can pass the signature in the following ways:

1. Pass through a standard HTTP `Authorization` header, such as `Authorization: q-sign-algorithm=sha1&q-ak=...&q-sign-time=1557989753;1557996953&...&q-signature=...`
2. Pass as an HTTP request parameter (be sure to URL-encode), such as `/exampleobject?q-sign-algorithm=sha1&q-ak=...&q-sign-time=1557989753%3B1557996953&...&q-signature=...`

### Note:

In the example above, `...` are the signatures.

# Monitoring and Alarms

Last updated : 2024-01-31 16:54:06

## Overview

CI monitoring data, such as image processing usage, media job quantity, and content recognition request quantity, is collected and displayed by [CM](#). Therefore, you can go to the CI or [CM](#) console to view CI monitoring data in detail. This document describes how to get CI monitoring metrics. You can call CM APIs to get more detailed data. For more information, see the [CM documentation](#).

## Basic Features

CM provides the following modules for CI to implement monitoring and alarming.

Module	Capability	Main Feature
Monitoring overview	Displays the current status of the product	Provides general overview, alarm overview, and overall monitoring information
Alarm management	Supports alarm management and configuration	Supports creating new CI alarm policies, custom messages, and trigger templates
Monitoring platform	Displays the data of user-defined monitoring metrics	Allows you to customize the monitoring metrics and data to be reported
Cloud product monitoring	Displays the bucket monitoring view	Allows you to query the current monitoring views and data for each bucket, such as image processing usage, media job quantity, and content recognition request quantity

## Use Cases

**Daily management:** You can log in to the CM console to view the running status of CI in real time.

**Troubleshooting:** You can receive alarm notifications when the data of a monitoring metric reaches the threshold. It allows you to quickly notify the exceptions, find out the causes, and fix the issues.



## Setting and Querying via Console

You can create an alarm policy for CI in the [CM console](#). If the data of a monitoring metric reaches the specified threshold, you will receive an alarm notification. For detailed directions, see [Setting Monitoring Alarms](#).

## Querying Monitoring Data via APIs

You can view the monitoring data of CI by calling APIs.

## Monitoring Metrics

### Note:

CI monitoring data resides in Guangzhou region. Therefore, select "Guangzhou" for `Region` when you pull CI monitoring metric data, regardless of where the bucket resides.

When pulling data by using [API Explorer](#), select "ap-guangzhou" for the `Region` field.

When pulling data by using an SDK, enter "ap-guangzhou" for the `Region` field.

### Image processing

Metric	Meaning	Description	Unit	Dimension
image_requests	Total image processing requests	Total number of image processing requests, including basic processing, Guetzli image compression, image advanced compression, and blind watermarking	-	appid, bucket
image_basics_requests	Image basic processing requests	Total number of image basic processing requests	-	appid, bucket
image_guetzil_requests	Guetzil image compression	Number of Guetzil image	-	appid, bucket

	requests	compression requests		
image_compress_requests	Image advanced compression requests	Number of image advanced compression request counts	-	appid, bucket
image_imprint_requests	Blind watermarking requests	Number of blind watermarking requests	-	appid, bucket
image_basics_bytes	Total image basic processing usage	Total usage of image basic processing requests of all types	B	appid, bucket
image_basics_bytes_down	Image download usage	Usage of image basic processing requests of the GET type	B	appid, bucket
image_basics_bytes_up	Image upload usage	Usage of image basic processing requests of the PUT type	B	appid, bucket
image_put_2xx_response	2xx status codes for image upload	Number of PUT requests with a 2xx return code	-	appid, bucket
image_put_3xx_response	3xx status codes for image upload	Number of PUT requests with a 3xx return code	-	appid, bucket
image_put_4xx_response	4xx status codes for image upload	Number of PUT requests with a 4xx return code	-	appid, bucket
image_put_5xx_response	5xx status codes for image upload	Number of PUT requests with a 5xx return code	-	appid, bucket
image_get_2xx_response	2xx status codes for image download	Number of GET requests with a 2xx return code	-	appid, bucket

image_get_3xx_response	3xx status codes for image download	Number of GET requests with a 3xx return code	-	appid, bucket
image_get_4xx_response	4xx status codes for image download	Number of GET requests with a 4xx return code	-	appid, bucket
image_get_5xx_response	5xx status codes for image download	Number of GET requests with a 5xx return code	-	appid, bucket

**Note:**

1. For more information on 2xx, 3xx, 4xx, and 5xx status codes, see [Error Codes](#).
2. The statistical granularity ( `period` ) may vary by metric. You can call the [DescribeBaseMetrics](#) API to obtain the `period` supported by each metric.

**Media processing**

Metric	Meaning	Description	Unit	Dimension
media_tasks	Media processing jobs	Total number of media processing jobs of all types	-	appid, bucket
media_success_tasks	Successful media processing jobs	Number of successful media processing jobs of all types	-	appid, bucket
media_request_success_rate	Media processing request success rate	Percentage of successful media processing jobs to all jobs	%	appid, bucket
video_transcoding_times	Standard transcoding duration	Duration of standard transcoding	min	appid, bucket
format_conversion_times	Audio/Video remuxing duration	Duration of audio/video remuxing	min	appid, bucket
audio_transcoding_times	Audio transcoding duration	Duration of audio transcoding	min	appid, bucket

HD_transcoding_times	TESHD transcoding duration	Duration of TESHD transcoding	min	appid, bucket
----------------------	----------------------------	-------------------------------	-----	---------------

## Content recognition

Metric	Meaning	Description	Unit	Dimension
QR_code_recognition_requests	QR code recognition requests	Number of QR code recognition requests	-	appid, bucket
QR_code_recognition_success_requests	Successful QR code recognition requests	Number of successful QR code recognition requests	-	appid, bucket
QR_code_recognition_fail_requests	Failed QR code recognition requests	Number of failed QR code recognition requests	-	appid, bucket
image_tag_requests	Image tagging requests	Number of image tagging requests	-	appid, bucket
image_tag_success_requests	Successful image tagging requests	Number of successful image tagging requests	-	appid, bucket
image_tag_fail_requests	Failed image tagging requests	Number of failed image tagging requests	-	appid, bucket
image_quality_requests	Image quality assessment requests	Number of image quality assessment requests	-	appid, bucket
image_quality_success_requests	Successful image quality assessment requests	Number of successful image quality assessment requests	-	appid, bucket
image_quality_fail_requests	Failed image quality	Number of failed image quality	-	appid, bucket

	assessment requests	assessment requests		
audio_recognition_requests	Speech recognition requests	Number of speech recognition requests	-	appid, bucket
audio_recognition_success_requests	Successful speech recognition requests	Number of successful speech recognition requests	-	appid, bucket
audio_recognition_fail_requests	Failed speech recognition requests	Number of failed speech recognition requests	-	appid, bucket
face_check_requests	Face detection requests	Number of face detection requests	-	appid, bucket
face_check_success_requests	Successful face detection requests	Number of successful face detection requests	-	appid, bucket
face_check_fail_requests	Failed face detection requests	Number of failed face detection requests	-	appid, bucket
face_effects_requests	Face filter requests	Number of face filter requests	-	appid, bucket
face_effects__success_requests	Successful face filter requests	Number of successful face filter requests	-	appid, bucket
face_effects__fail_requests	Failed face filter requests	Number of failed face filter requests	-	appid, bucket
face_identity_check_requests	Face recognition requests	Number of face recognition requests	-	appid, bucket

face_identity_check_success_requests	Successful face recognition requests	Number of successful face recognition requests	-	appid, bucket
face_identity_check_fail_requests	Failed face recognition requests	Number of failed face recognition requests	-	appid, bucket
car_recognition_requests	Vehicle recognition requests	Number of vehicle recognition requests	-	appid, bucket
car_recognition_success_requests	Successful vehicle recognition requests	Number of successful vehicle recognition requests	-	appid, bucket
car_recognition_fail_requests	Failed vehicle recognition requests	Number of failed vehicle recognition requests	-	appid, bucket
OCR_recognition_requests	Image text recognition requests	Number of image text recognition requests	-	appid, bucket
OCR_recognition_success_requests	Successful image text recognition requests	Number of successful image text recognition requests	-	appid, bucket
OCR_recognition_fail_requests	Failed image text recognition requests	Number of failed image text recognition requests	-	appid, bucket

## File processing

Metric	Meaning	Description	Unit	Dimension
document_transcode_requests	File transcoding requests	Number of file transcoding requests	-	appid, bucket

document_transcode_success_requests	Successful file transcoding requests	Number of successful file transcoding requests	-	appid, bucket
document_transcode_fail_requests	Failed file transcoding requests	Number of failed file transcoding requests	-	appid, bucket
document_html_requests	File-to-HTML requests	Number of file-to-HTML requests	-	appid, bucket
document_html_success_requests	Successful file-to-HTML requests	Number of successful file-to-HTML requests	-	appid, bucket
document_html_fail_requests	Failed file-to-HTML requests	Number of failed file-to-HTML requests	-	appid, bucket

## Content moderation

Metric	Meaning	Description	Unit	Dimension
document_auditing_tasks	Total file moderation jobs	Total number of submitted file moderation jobs	-	appid, bucket
document_auditing_success_tasks	Successful file moderation jobs	Number of successful file moderation jobs	-	appid, bucket
document_auditing_fail_tasks	Failed file moderation jobs	Number of failed file moderation jobs	-	appid, bucket
document_auditing_callback_times	Total file moderation result callbacks	Total number of file moderation result callbacks	-	appid, bucket
document_auditing_callback_success_times	Successful file moderation result callbacks	Number of successful file moderation result callbacks	-	appid, bucket

document_auditing_callback_fail_times	Failed file moderation result callbacks	Number of failed file moderation result callbacks	-	appid, bucket
text_auditing_tasks	Text moderation jobs	Number of submitted text moderation jobs	-	appid, bucket
text_auditing_success_tasks	Successful text moderation jobs	Number of successful text moderation jobs	-	appid, bucket
text_auditing_fail_tasks	Failed text moderation jobs	Number of failed text moderation jobs	-	appid, bucket
text_auditing_callback_times	Total text moderation result callbacks	Total number of text moderation result callbacks	-	appid, bucket
text_auditing_callback_success_times	Successful text moderation result callbacks	Number of successful text moderation result callbacks	-	appid, bucket
text_auditing_callback_fail_times	Failed text moderation result callbacks	Number of failed text moderation result callbacks	-	appid, bucket
images_auditing_tasks	Image moderation jobs	Number of submitted image moderation jobs	-	appid, bucket
images_auditing_success_tasks	Successful image moderation jobs	Number of successful image moderation jobs	-	appid, bucket
images_auditing_fail_tasks	Failed image moderation jobs	Number of failed image moderation jobs	-	appid, bucket
images_auditing_callback_times	Total image moderation result callbacks	Total number of image moderation result callbacks	-	appid, bucket
images_auditing_callback_success_times	Successful image	Number of successful image	-	appid, bucket



	moderation result callbacks	moderation result callbacks		
images_auditing_callback_fail_times	Failed image moderation result callbacks	Number of failed image moderation result callbacks	-	appid, bucket
audio_auditing_tasks	Audio moderation jobs	Number of submitted audio moderation jobs	-	appid, bucket
audio_auditing_success_tasks	Successful audio moderation jobs	Number of successful audio moderation jobs	-	appid, bucket
audio_auditing_fail_tasks	Failed audio moderation jobs	Number of failed audio moderation jobs	-	appid, bucket
audio_auditing_callback_times	Total audio moderation result callbacks	Total number of audio moderation result callbacks	-	appid, bucket
audio_auditing_callback_success_times	Successful audio moderation result callbacks	Number of successful audio moderation result callbacks	-	appid, bucket
audio_auditing_callback_fail_times	Failed audio moderation result callbacks	Number of failed audio moderation result callbacks	-	appid, bucket
video_auditing_tasks	Video moderation jobs	Number of submitted video moderation jobs	-	appid, bucket
video_auditing_success_tasks	Successful video moderation jobs	Number of successful video moderation jobs	-	appid, bucket
video_auditing_fail_tasks	Failed video moderation jobs	Number of failed video moderation jobs	-	appid, bucket
video_auditing_callback_times	Total video moderation result callbacks	Total number of video moderation result callbacks	-	appid, bucket

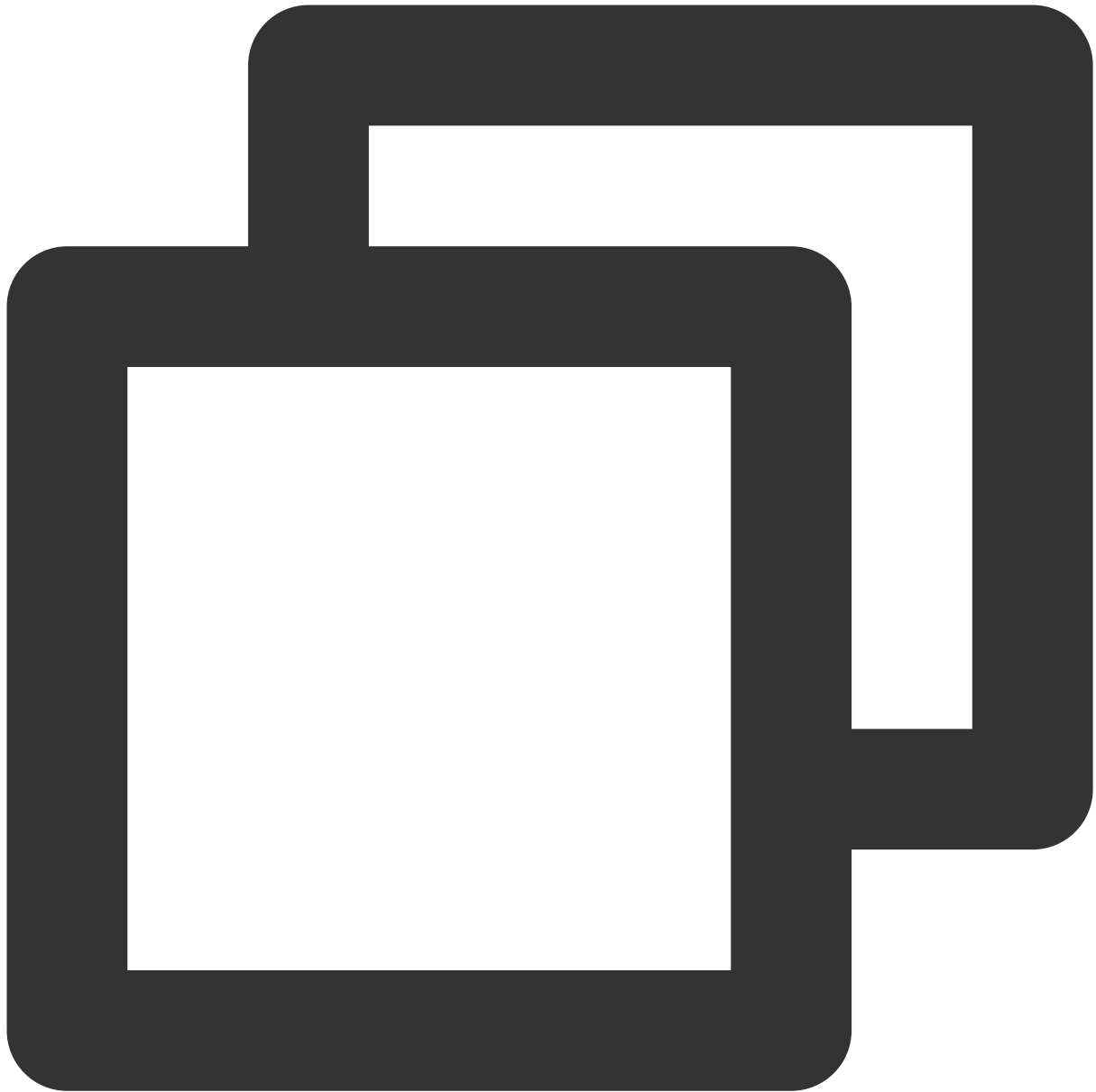
video_auditing_callback_success_times	Successful video moderation result callbacks	Number of successful video moderation result callbacks	-	appid, bucket
video_auditing_callback_fail_times	Failed video moderation result callbacks	Number of failed video moderation result callbacks	-	appid, bucket

## Traffic

Metric	Meaning	Description	Unit	Dimension
cdn_origin_traffic	CDN origin-pull traffic	Traffic generated by CI data transfer from the bucket to the CDN edge node	B	appid, bucket
internet_trafficUp	Public network outbound traffic	Traffic generated by CI data download from the bucket to the client over the internet	B	appid, bucket

## Input Parameter Description

To query the monitoring data of CI, use the following input parameters:



```
&Namespace=QCE/CI
&Instances.N.Dimensions.0.Name=appid
&Instances.N.Dimensions.0.Value=root account APPID
&Instances.N.Dimensions.1.Name=bucket
&Instances.N.Dimensions.1.Value=bucket name
```

### Monitoring description

**Monitoring interval:** CM supports multiple monitoring intervals, including real time, last 24 hours, last 7 days, and user-specified period, with time granularities of 1 minute, 5 minutes, 1 hour, and 1 day.

**Data storage:** 1-minute monitoring data can be stored for 15 days, 5-minute data for 31 days, 1-hour data for 93 days, and 1-day data for 186 days.

**Alarm display:** CM integrates the monitoring data of CI and displays the data in graphs. Alarm notifications can be sent to you according to the predefined alarm metrics of your product. In this way, you can stay informed of the overall running status.

**Alarm settings:** You can set the threshold for the monitoring metrics. When the monitoring data meets the alarm condition that is set, CM will send the alarm notifications to the specified users.