

# Batch Compute

## User Guide

### 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



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

## User Guide

How to create images usable by BatchCompute

Windows Custom Images

Environment Variables

Entering COS & CFS Paths

Event Notice & Callback

Cloud Access Management

# User Guide

## How to create images usable by BatchCompute

Last updated : 2024-01-13 11:19:29

### Overview

BatchCompute relies on Cloud-init service to initialize CVMs. Hence the image used for BatchCompute should have Cloud-init installed and configured. Otherwise, the job execution may fail, or CVMs cannot be created in the compute environment. (With Cloud-init, you can customize the configuration of the CVM when it is first initialized.)

Note the following while installing and configuring Cloud-init:

**Newly created** Linux CVM/image: all current versions of Tencent Cloud CentOS and Ubuntu public images support Cloud-init by default. When you create CVMs and custom images with these public images, you needn't manually install and configure Cloud-init.

**Existing** Linux CVM/image: if the CVMs or custom images are created earlier, you need to manually install Cloud-init.

See [Installing Cloud-Init on Linux](#).

Windows image: you need to create Windows CVMs and images via marketplace images. See [Windows Custom Images](#).

Common operating system images that already contain Cloud-init are as follows:

img-31tjrtph (CentOS 7.2 64-bit)

img-er9shcln (Windows Server 2012 R2 Standard 64-bit English)

img-pyqx34y1 (Ubuntu Server 16.04.1 LTS 64-bit)

# Windows Custom Images

Last updated : 2024-01-13 11:19:29

## Overview

To use Windows services, you need to create custom images from existing Windows Server images. This document introduces how to create a Windows custom image.

## Prerequisites

You have registered a Tencent Cloud account. If not, go to the [Sign up](#) page.

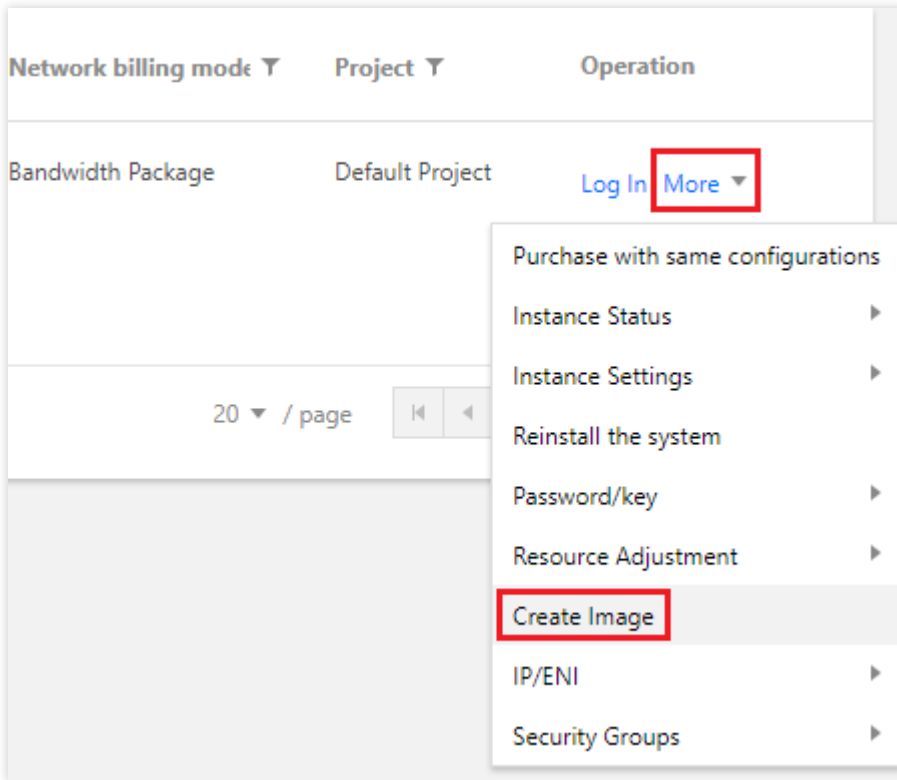
## Directions

### Installing required software on a CVM

Log in to the [CVM Console](#) to view details of the CVM you have created. After remote login, install all necessary software on the CVM and test the connectivity.

### Creating a custom image

1. Locate the CVM for which you want to create a custom image, click **More** -> **Create image** under the **Operation** column, as shown below:

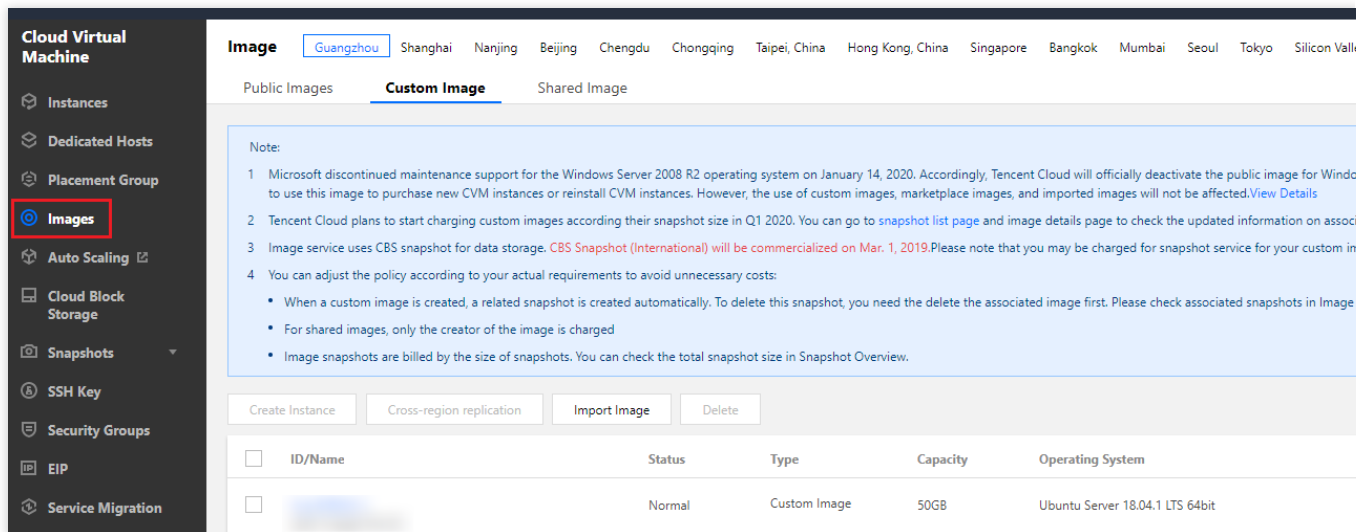


2. In the pop-up window, enter the image name and description, and click **Create Image**.

3. After the image is created, click **Images** on the left sidebar to view custom images, as shown below:

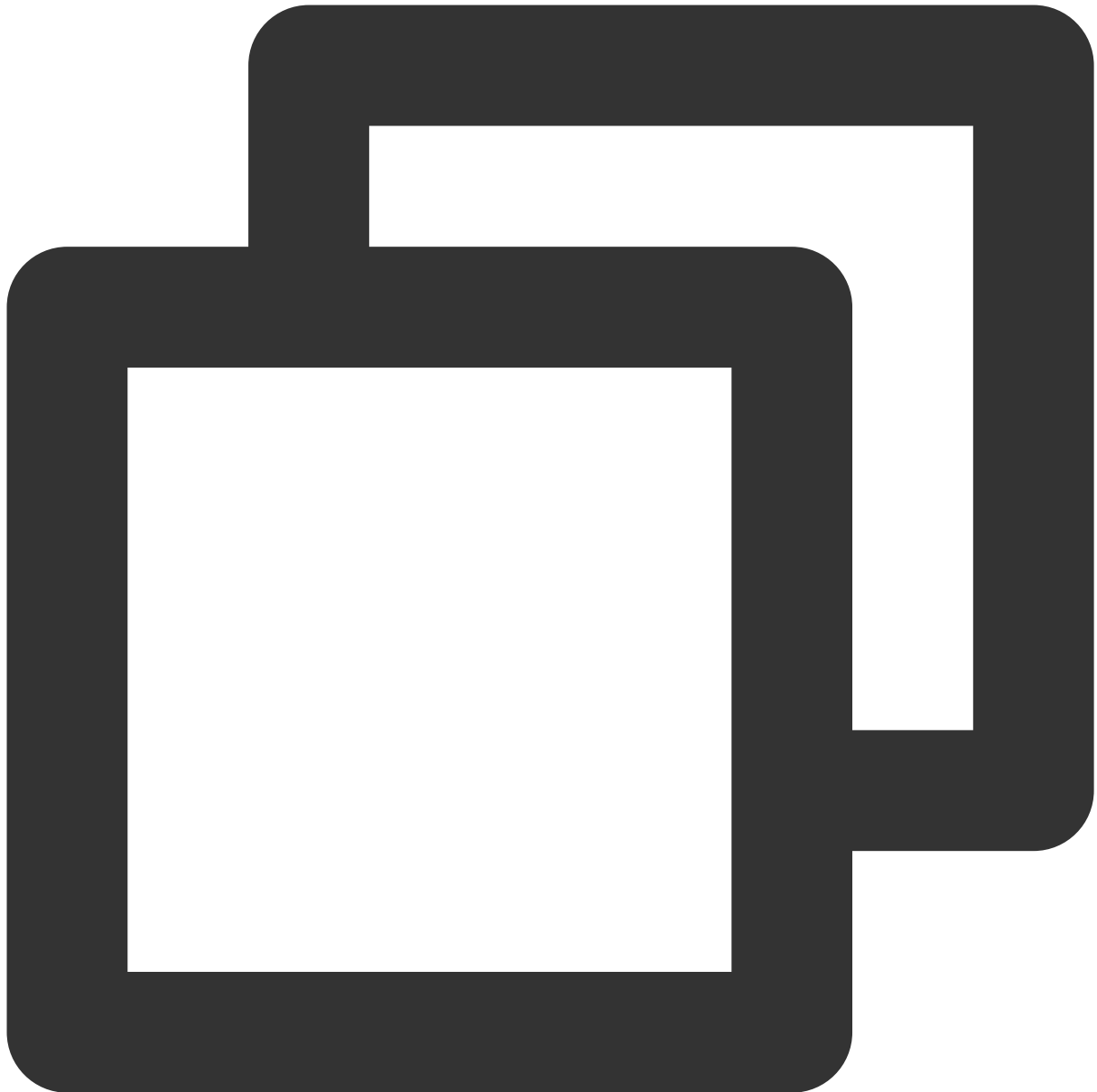
**Note:**

You can obtain the custom image ID on the **Image** page.



**Using a custom image to submit the test job**

You can obtain and modify the official sample to create a Batch Compute environment under your account. See below to learn about each configuration item in the compute environment.



```
tccli batch SubmitJob --version 2017-03-12 --Job '{
  "JobName": "TestJob",          // Job name
  "JobDescription": "for test ", // Job description
  "Priority": "1",               // Job priority
  "Tasks": [                    // Task list (this example only has one task)
    {
      "TaskName": "Task1",      // Task 1 name
      "Application": {          // Task execute command
        "DeliveryForm": "LOCAL", // Run the local command
        "Command": "python -c \\\"fib=lambda n:1 if n<=2 else fib(n-1)+fib(n-2)\\\" "
      }
    }
  ],
}
```

```
"ComputeEnv": {          // Compute environment configuration
  "EnvType": "MANAGED",  // Compute environment types: managed and u
  "EnvData": {          // Specific configuration (current type is mana
    "InstanceType": "S1.SMALL1",    // CVM instance type
    "ImageId": "",          // CVM image ID (Replace with your custom i
  }
},
"RedirectInfo": {       // Configuration of standard output redirection
  "StdoutRedirectPath": "cos://dondonbatchv5-1251783334.cosgz.myqclou
  "StderrRedirectPath": "cos://dondonbatchv5-1251783334.cosgz.myqclou
}
}
]
}'
--Placement'{
  "Zone": "ap-guangzhou-2"    // Availability zone (to be replaced)
}'
```

You can use the following sample codes. Refer to the “Job Configuration” section in [Using CLI - Submit a Job](#), and replace the **To be replaced** fields with your actual information. For example, replace `ImageId` with your custom image ID.





```
tccli batch SubmitJob --version 2017-03-12 --Job '{"JobName": "TestJob", "JobDesc
```

# Environment Variables

Last updated : 2024-01-13 11:19:28

## Summary

BatchCompute provides the instances of a task with the task-related environment variable information to allow user program to execute different computing tasks based on the environment variables.

## Details

Variable	Name	Description
BATCH_JOB_ID	Job ID	ID of the job to which an instance belongs. It is included in the returned result after the job is submitted, such as job-n4ohivif
BATCH_TASK_NAME	Task name	Name of the task to which an instance belongs. It is specified when the job is submitted, such as <code>"TaskName"</code> ; <code>"Task1"</code> ;
BATCH_TASK_INSTANCE_NUM	Total number of instances of a task	The total number of instances requested to run concurrently in the task to which the instances belong, such as <code>"TaskInstanceNum"</code> ; <code>5</code>
BATCH_TASK_INSTANCE_INDEX	Task instance index	The index of an instance in the task to which the instance belongs; for example, if five instances are specified to run concurrently in a task, then the indices of these instances are 0, 1, 2, 3, 4

# Entering COS & CFS Paths

Last updated : 2024-01-13 11:19:29

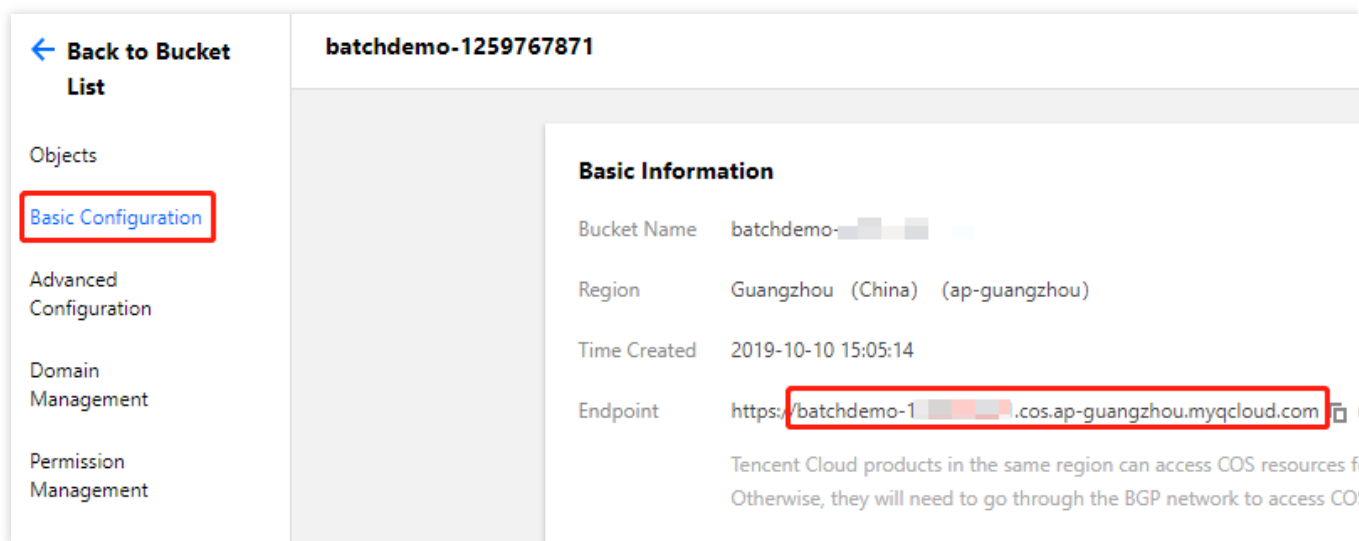
## Overview

You need to enter COS and CFS paths for execution logs ( `StdOut` and `StdErr` ) and remote storage mappings in BatchCompute, which is slightly different from access to COS buckets or files in HTTP mode.

## COS Path Description

### Only endpoints in COS XML API format

For BatchCompute paths, only endpoints in the XML API format are supported as shown below:



The screenshot shows the console interface for a COS bucket. On the left, there is a navigation menu with options: Back to Bucket List, Objects, Basic Configuration (highlighted with a red box), Advanced Configuration, Domain Management, and Permission Management. The main content area displays the bucket name 'batchdemo-1259767871' and a 'Basic Information' section. The 'Endpoint' field is highlighted with a red box and shows the URL 'https://batchdemo-1[redacted].cos.ap-guangzhou.myqcloud.com'. Below the endpoint, there is a note: 'Tencent Cloud products in the same region can access COS resources f... Otherwise, they will need to go through the BGP network to access CO...'

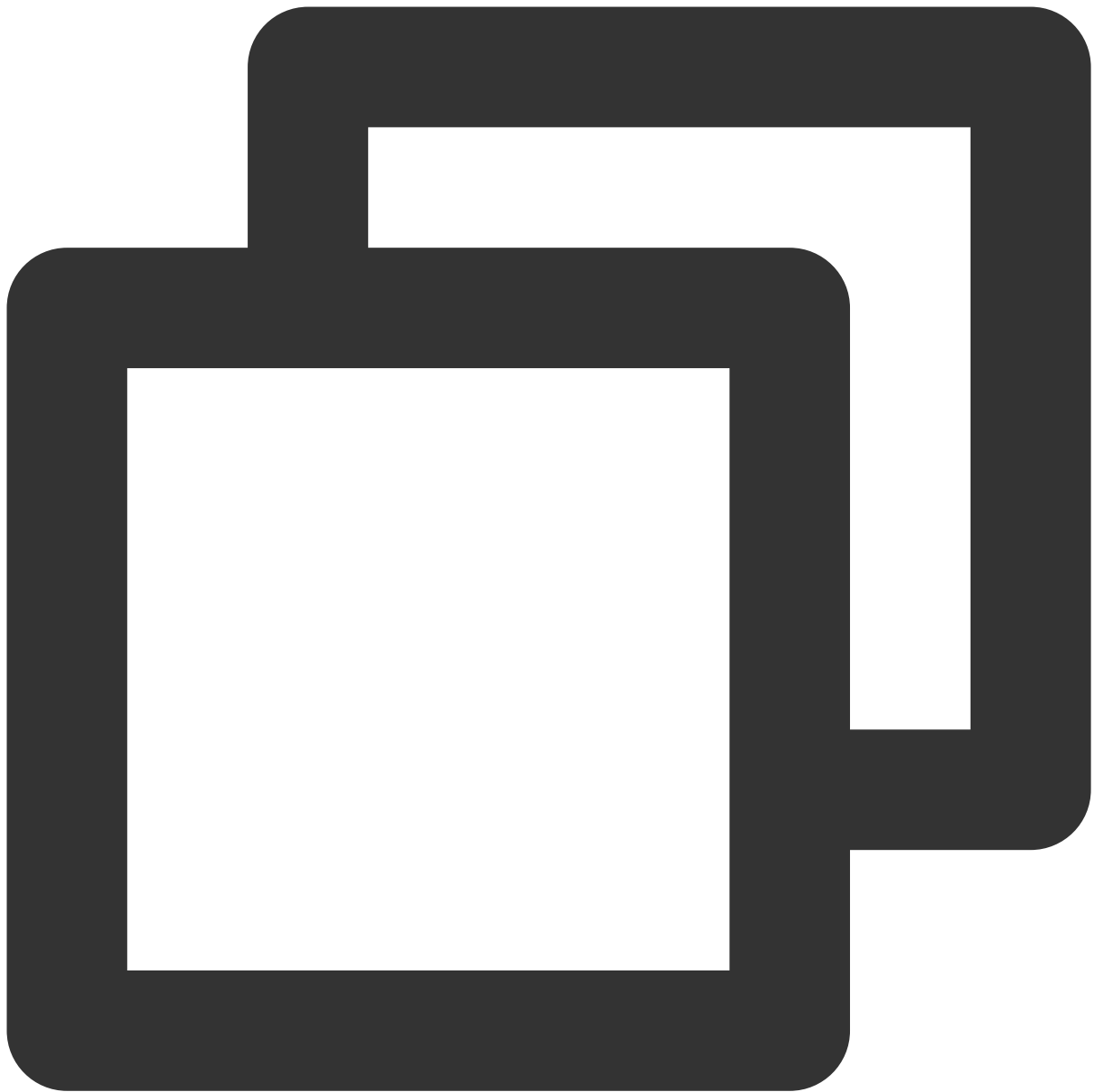
Basic Information	
Bucket Name	batchdemo-[redacted]
Region	Guangzhou (China) (ap-guangzhou)
Time Created	2019-10-10 15:05:14
Endpoint	https://batchdemo-1[redacted].cos.ap-guangzhou.myqcloud.com

### Prefixed with cos://

The obtained COS endpoint is as shown below:

The screenshot shows the Tencent Cloud COS console interface. On the left, a sidebar contains navigation options: 'Back to Bucket List', 'Objects', 'Basic Configuration' (highlighted with a red box), 'Advanced Configuration', 'Domain Management', and 'Permission Management'. The main content area displays the configuration for a bucket named 'batchdemo-1259767871'. Under the 'Basic Information' section, the following details are listed: Bucket Name (batchdemo-...), Region (Guangzhou (China) (ap-guangzhou)), Time Created (2019-10-10 15:05:14), and Endpoint (https://batchdemo-1...cos.ap-guangzhou.myqcloud.com). The endpoint URL is also highlighted with a red box. Below the endpoint, there is a note: 'Tencent Cloud products in the same region can access COS resources f. Otherwise, they will need to go through the BGP network to access CO'.

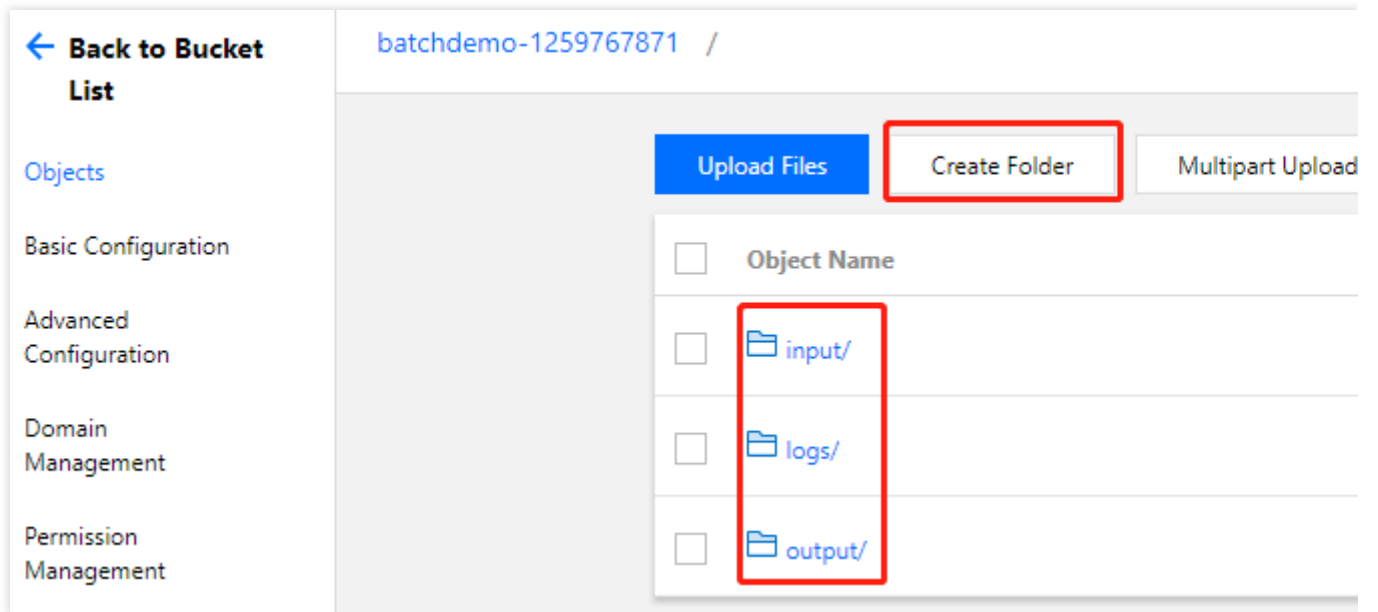
Prefix BatchCompute paths with `cos://` and suffix them with `/`.



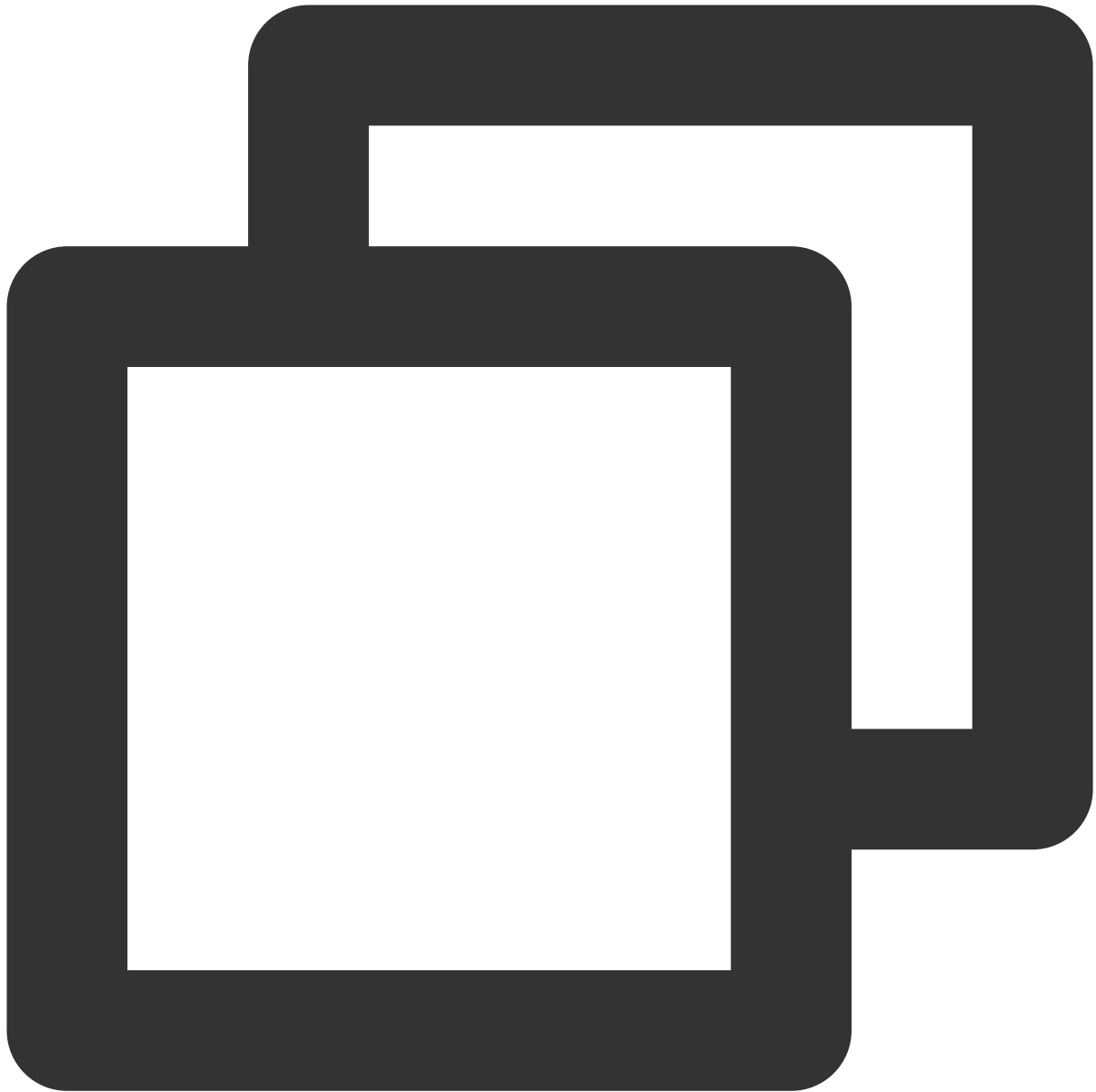
```
cos://batchdemo-125178xxxx.cos.ap-guangzhou.myqcloud.com/
```

### Mounting subdirectory

Add subdirectories to a bucket path as common file directories. The following figure shows the subdirectories created in the bucket.



For directory mounting, the COS paths are as follows:



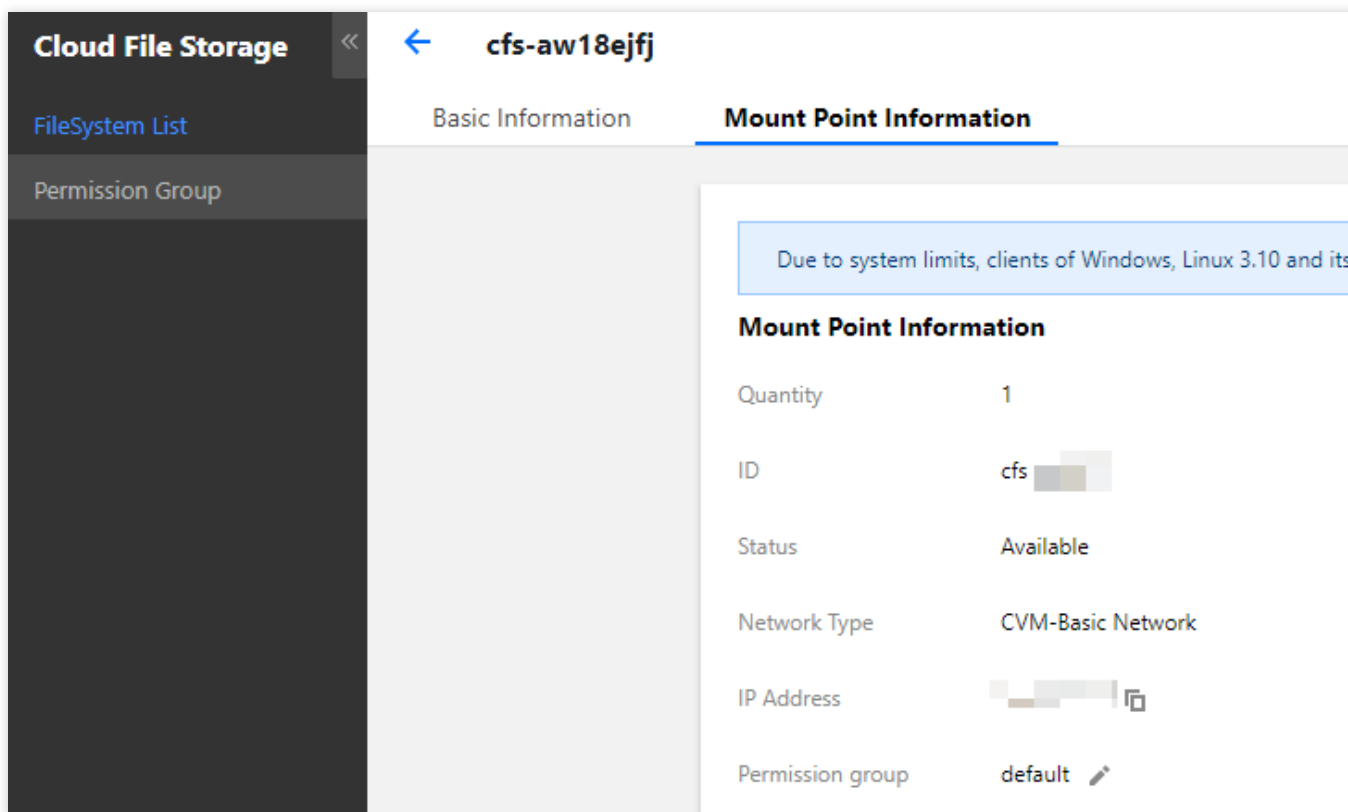
```
cos://batchdemo-125178xxxx.cos.ap-guangzhou.myqcloud.com/logs/  
cos://batchdemo-125178xxxx.cos.ap-guangzhou.myqcloud.com/input/  
cos://batchdemo-125178xxxx.cos.ap-guangzhou.myqcloud.com/output/
```

### Supporting intra-region buckets

COS has regional attributes. Data transfer between the storage server and Cloud Virtual Machine (CVM) is efficient only when your BatchCompute job and COS bucket are in the same region.

## CFS Path Description

In remote storage mappings, you can configure CFS/NAS paths to automatically mount to a local path as shown below:



### Prefixed with `cfs://` or `nfs://`

An example of an obtained CFS path is `10.66.xxx.xxx`. Prefix a BatchCompute path with `cfs://` or `nfs://`.

### Note :

Suffix the path with `/` and ensure that your CFS/NAS and BatchCompute job are on the same network.





`cfs://10.66.xxx.xxx/`

# Event Notice & Callback

Last updated : 2024-01-13 11:19:29

## Summary

BatchCompute (Batch) supports throwing of events generated in the jobs and compute environment in the form of Cloud Message Queue (CMQ). For example, when events such as job execution success/failure and compute environment node creation success/failure/exception occur, notifications and callbacks can be made possible through the topic subscription mechanism of CMQ.

## How to Use

A compute environment listening event is used as an example below to demonstrate how to register a compute environment-related event in three steps.

### 1. Create a CMQ topic

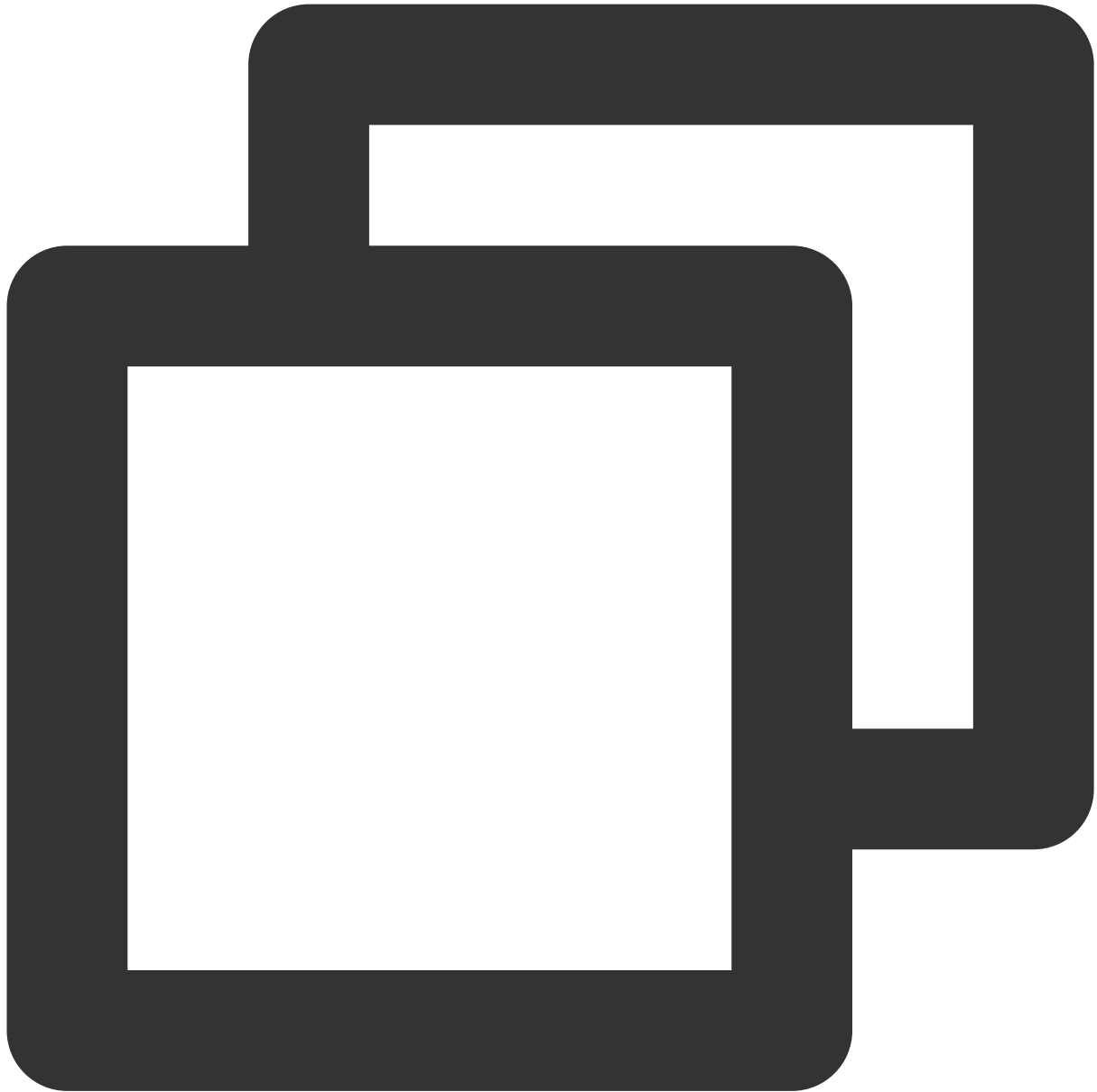
Log in to the [CMQ Console](#) and create a new topic.

The screenshot shows the Tencent Cloud CMQ console interface. On the left, the 'Messaging Service CMQ' sidebar is visible with 'Queue Service' and 'Topic Subscription' (checked) options. The main area is titled 'Topic Subscription' and shows a region dropdown set to 'Guangzhou'. A blue information box contains the following text: 'To ensure the security of data transmission, it is recommended to use HTTPS. Queue: https://cmq-queue-gz.api.qcloud.com Topic: https://cmq-topic-gz.api.qcloud.com'. Below this, there are buttons for 'Create', 'Edit resource tag', and 'API request ac'. A table below lists topics with columns 'ID/Name', 'Mo...', and 'Messag'. One topic is listed with 'topic' and 't' in the ID/Name column, a bar chart icon in the Mo... column, and '1 day' in the Messag column. This row is highlighted with a red box.

ID/Name	Mo...	Messag
topic t		1 day

### 2. Create a compute environment and associate the CMQ topic

Add a **notifications** field when submitting a job (SubmitJob) or creating a compute environment (Create) and specify the **event\_name** of the event to be listened to. Multiple events can be specified.



```
"notifications": [  
  {  
    "event_configs": [  
      {  
        "event_name": "JobFailed" // Event name  
      },  
      {  
        "event_name": "JobSucceed",  
      }  
    ]  
  }  
]
```

```
    "event_vars": [          // Custom event parameter
      {
        "name": "jobSucceed",
        "value": "Success"
      }
    ]
  },
],
"topic_name": "job-message" // CMQ topic name
}
],
```

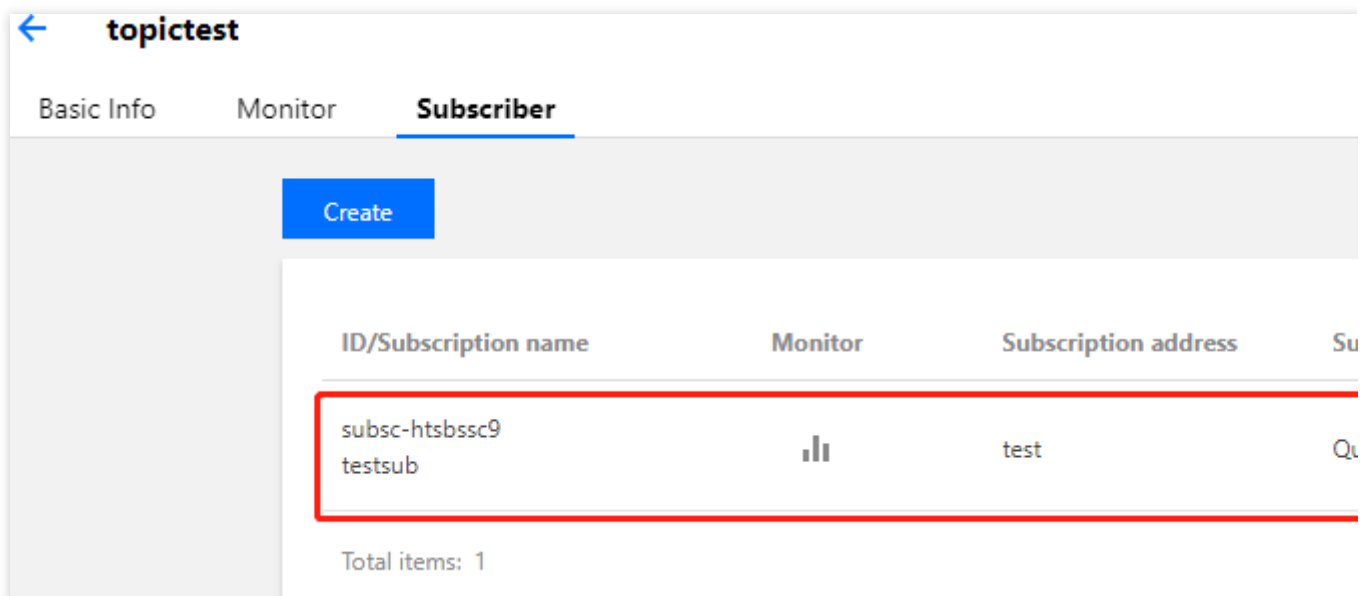
Currently, it is only supported to associate CMQ topics when creating a compute environment via API or CLI. Doing so in the console will be supported in the future.

`event_vars`: In addition to the fixed message generated by the event, custom parameters can be added.

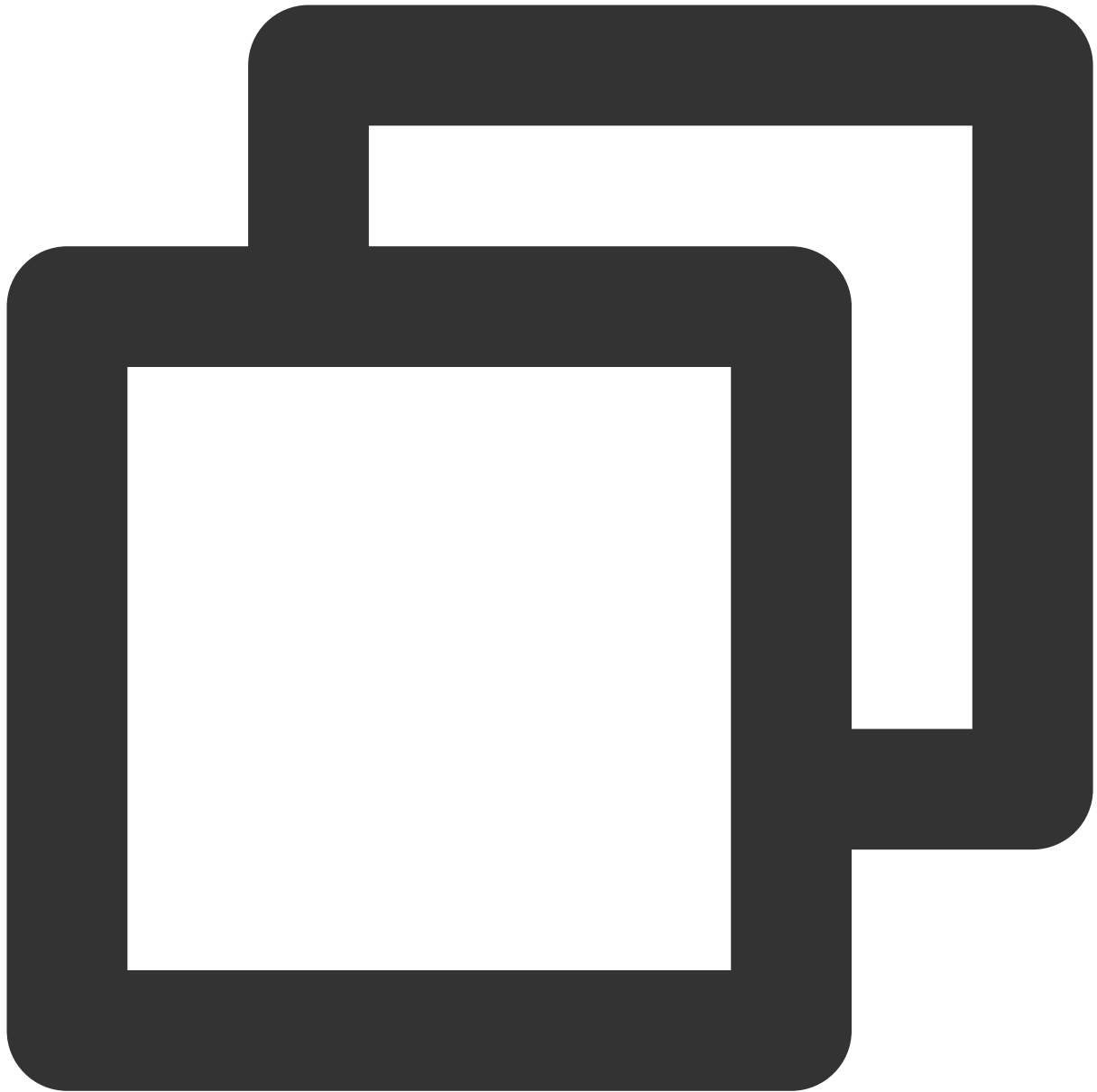
`topic_name`: Name of the associated CMQ topic (**Note: This is not the ID**). All event messages will be delivered to this topic which will then forward the messages to all subscribers.

### 3. Set up a subscriber and test

Add a subscriber to the newly created topic in the [CMQ Console](#). For quick and easy viewing, you can specify an existing message queue.



The message structure is as follows. If a message queue is specified as the subscriber, you can quickly view the messages sent to the topic by Batch in [CMQ Console > Message Receiving](#) where the message content requires Base64 processing.



```
{
  "Events": [{
    "EventVersion": "1.0",
    "EventTime": "2018-06-15T14:43:17Z",
    "Region": "ap-guangzhou",
    "Batch": {
      "ComputeNodeId": "node-0iy7wxyo",
      "EnvId": "env-ptoxdb1t",
      "ComputeNodeState": "CREATED",
      "Mem": 8,
      "ResourceCreatedTime": "2018-06-15T14:43:18Z",
```

```

      "EnvName": "batch-env",
      "ComputeNodeInstanceId": "ins-9rikj9kw",
      "Cpu": 4
    },
    "EventName": "COMPUTE_NODE_CREATED",
    "EventVars": []
  }]
}

```

## Job-related Events

Type	Description
JOB_RUNNING	Job is running
JOB_SUCCEED	Job succeeded
JOB_FAILED	Job failed
JOB_FAILED_INTERRUPTED	Job is interrupted due to failure
TASK_RUNNING	Task is running
TASK_SUCCEED	Task succeeded
TASK_FAILED	Task failed
TASK_FAILED_INTERRUPTED	Task is interrupted due to failure
TASK_INSTANCE_RUNNING	Task instance is running
TASK_INSTANCE_SUCCEED	Task instance succeeded
TASK_INSTANCE_FAILED	Task instance failed
TASK_INSTANCE_FAILED_INTERRUPTED	Task instance is interrupted due to failure

For more information, see [SubmitJob](#).

## Compute Environment-related Events

Type	Description
COMPUTE_ENV_CREATED	Compute environment created
COMPUTE_ENV_DELETED	Compute environment deleted

---

COMPUTE_NODE_CREATED	Compute node successfully created
COMPUTE_NODE_CREATION_FAILED	Compute node creation failed
COMPUTE_NODE_RUNNING	Compute node is running
COMPUTE_NODE_ABNORMAL	Compute node is abnormal
COMPUTE_NODE_DELETING	Deleting compute node

For more information, see [CreateComputeEnv](#).

# Cloud Access Management

Last updated : 2024-01-13 11:19:29

## Overview

[Cloud Access Management \(CAM\)](#) is a web-based Tencent Cloud service that helps you with the security management of access permissions for resources under your Tencent Cloud account. With CAM, you can create, manage, and terminate users or user groups, and can use identity and policy management to control the permissions other users have to use Tencent Cloud resources. Policies can be used to authorize or block the use of specified resources by users to complete specified tasks. When you use CAM, you can associate policies with a user or user group to perform permissions control.

BatchCompute is already integrated with CAM. You can use CAM to control the permissions for resources related to BatchCompute.

## Related concepts

### CAM users

[CAM users](#) are entities that you create in Tencent Cloud, and each CAM user is associated with only one Tencent Cloud account. The Tencent Cloud account you register is the root account. You can also create sub-accounts with different permissions in [Cloud Access Management](#). There are 3 types of sub-accounts: [sub-users](#), [collaborators](#), and [message recipients](#).

### Policies

[The policy](#) is the syntax rule used to define and describe one or more permissions. CAM supports two types of policies: preset policy and custom policy.

**Preset policies:** Policies created and managed by Tencent Cloud. These are some common permission sets that are frequently used by users, such as full read and write permissions for resources. Preset policies have a wide range of operation objects, coarse operation granularity, and are preset by the system. They cannot be edited by users.

**Custom policies:** Policies created by users. These permit fine-grained division of permissions. For example, a usage policy is associated with a sub-account that gives the sub-account management permissions for the computing environment of BatchCompute, but no management permissions for TencentDB instances.

### Resources

[Resource](#) is an element of policies that describes one or multiple operation objects. For example, BatchCompute computing environment or jobs.



## Preset policies of BatchCompute

Preset Policies	Permissions Granted
QcloudBatchFullAccess	Users associated with this policy will have full read and write access permissions to all BatchCompute resources.
QcloudBatchReadOnlyAccess	Users associated with this policy will have read-only permission to all BatchCompute resources.
QcloudFullAccessForBatchRole	Users associated with this policy will have access to other Tencent Cloud products while using BatchCompute. This includes access permissions for products such as CVM, VPC, COS, CMQ Topic, CMQ Queue, CLS, and Cloud Monitor.

## Authorizable resource types

Resource-level permissions specify which resources a user can operate. For example, you can authorize a user to have operation permissions for computing environments in Guangzhou region. The resource types of BatchCompute that can be authorized through CAM are as follows:

Resource Type	Resource Description Method in Authorization Policy
Job-related	qcs::batch:\${region}:uin/\${uin}:job/\${jobId}/*
Computing environment-related	qcs::batch:\${region}:uin/\${uin}:computeenv/\${envId}/*
Task template-related	qcs::batch:\${region}:uin/\${uin}:tasktemplate/\${taskTemplateId}/*

The following table lists the resource-level permissions operations APIs that BatchCompute supports, as well as the resource path supported by each operation. When setting the resource path, you must modify variable parameters such as `${region}`、`${uin}`、`${envId}`、`${jobId}`、and `${taskTemplateId}` according to your actual parameter information. You can also use `*` in the path as a wildcard character. For information about related concepts in CAM policies such as region, action, account, and resource, see [Element Reference](#).

## Note :

Any BatchCompute API operation not listed in the table does not support resource-level permission. For such operations, you can still authorize a user to perform it, but you must specify `*` as the resource element in the policy statement.

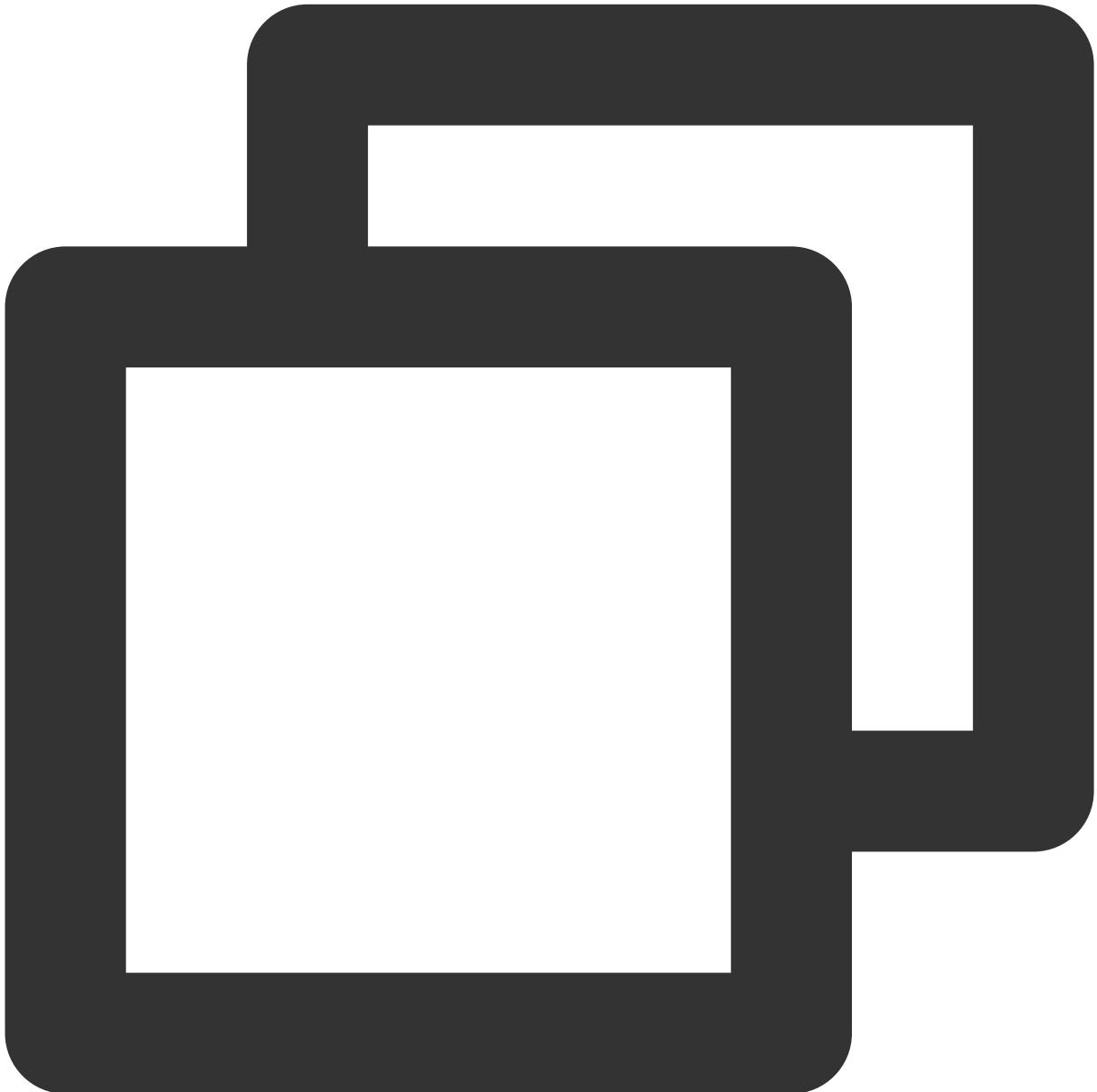
API: action	Resource path: resource
CreateCpmComputeEnv	qcs::batch:\${region}:uin/\${uin}:computeenv/*
CreateComputeEnv	qcs::batch:\${region}:uin/\${uin}:computeenv/*
DeleteComputeEnv	qcs::batch:\${region}:uin/\${uin}:computeenv/\${envId}

ModifyComputeEnv	qcs::batch:\${region}:uin/\${uin}:computeenv/\${envId}
TerminateComputeNode	qcs::batch:\${region}:uin/\${uin}:computeenv/\${envId}
TerminateComputeNodes	qcs::batch:\${region}:uin/\${uin}:computeenv/\${envId}
DescribeComputeEnv	qcs::batch:\${region}:uin/\${uin}:computeenv/\${envId}
DescribeComputeEnvActivities	qcs::batch:\${region}:uin/\${uin}:computeenv/\${envId}
DescribeComputeEnvCreateInfo	qcs::batch:\${region}:uin/\${uin}:computeenv/\${envId}
DescribeComputeEnvCreateInfos	qcs::batch:\${region}:uin/\${uin}:computeenv/\${envId}
DescribeComputeEnvs	qcs::batch:\${region}:uin/\${uin}:computeenv/\${envId}
SubmitJob	qcs::batch:\${region}:uin/\${uin}:job/*
RetryJobs	qcs::batch:\${region}:uin/\${uin}:job/\${jobId}
DeleteJob	qcs::batch:\${region}:uin/\${uin}:job/\${jobId}
TerminateJob	qcs::batch:\${region}:uin/\${uin}:job/\${jobId}
TerminateTaskInstance	qcs::batch:\${region}:uin/\${uin}:job/\${jobId}
DescribeJob	qcs::batch:\${region}:uin/\${uin}:job/\${jobId}
DescribeJobs	qcs::batch:\${region}:uin/\${uin}:job/\${jobId}
DescribeJobSubmitInfo	qcs::batch:\${region}:uin/\${uin}:job/\${jobId}
DescribeTask	qcs::batch:\${region}:uin/\${uin}:job/\${jobId}
DescribeTaskLogs	qcs::batch:\${region}:uin/\${uin}:job/\${jobId}
CreateTaskTemplate	qcs::batch:\${region}:uin/\${uin}:tasktemplate/*
DeleteTaskTemplates	qcs::batch:\${region}:uin/\${uin}:tasktemplate/\${taskTemplateId}
ModifyTaskTemplate	qcs::batch:\${region}:uin/\${uin}:tasktemplate/\${taskTemplateId}
DescribeTaskTemplates	qcs::batch:\${region}:uin/\${uin}:tasktemplate/\${taskTemplateId}
DetachInstances	qcs::batch:\${region}:uin/\${uin}:computeenv/\${envId}
AttachInstances	qcs::batch:\${region}:uin/\${uin}:computeenv/\${envId}

## BatchCompute CAM policy example

The following section provides two specific examples that display how to use CAM to control permissions for BatchCompute:

Create a policy to prohibit access to any computing environment in Guangzhou region.



```
# In this example, `${uin}` must be substituted with the account information, and `  
{  
"version": "2.0",
```

```
"statement": [  
  {  
    "effect": "allow",  
    "action": [  
      "name/batch:*"  
    ],  
    "resource": [  
      "qcs::batch:${region}:uin/${uin}:computeenv/*"  
    ]  
  },  
  {  
    "effect": "deny",  
    "action": [  
      "name/batch:*"  
    ],  
    "resource": [  
      "qcs::batch:${region}:uin/${uin}:computeenv/${envId}"  
    ]  
  }  
]  
}
```

Policy created: grant access permissions on jobs' reading APIs in all regions.



```
# In this example, `${uin}` must be substituted with the account information
```

```
{  
  "version": "2.0",  
  "statement": [  
    {  
      "effect": "allow",  
      "action": [  
        "name/batch:DescribeJobs",  
        "name/batch:DescribeJobSubmitInfo",  
        "name/batch:DescribeJob",
```

```
        "name/batch:DescribeTask",
        "name/batch:DescribeTaskLogs"
    ],
    "resource": [
        "qcs::batch:${region}:uin/${uin}:job/*"
    ]
}
]
}
```