Cloud File Storage

Product Documentation

Contents

Cloud File Storage 3
- Product Introduction 3
  - CFS Overview 3
  - Product Strengths 4
  - Storage Types, Performance, and Specifications 5
  - Use Cases 7
  - System Limits 8
- Purchase Guide 10
  - Billing Overview 10
  - Viewing Bills 11
- Getting Started 12
  - Creating File Systems and Mount Targets 12
  - Using CFS on a Linux Client 14
  - Using CFS on a Windows Client 22
  - Installing CFS Service Monitoring Plugin on Clients 27
  - Using the Console or CFS Client Assistant to Mount File Systems 29
- Operation Guide 31
  - Cloud Access Management(CAM) 31
  - Managing File System 39
  - Permission Management 41
  - Monitoring File Systems 43
  - Managing Snapshots 45
    - Creating Snapshot 45
    - Scheduled Snapshot 46
    - Creating a File System Using a Snapshot 48
    - Deleting Snapshots 50
  - Using Tags 51
  - Guide for Cross-AZ and Cross-Network Access 54
  - Automatically Mounting File Systems 56
- Best Practices 62
  - Using CFS on Containers 62
- API Documentation 65
  - NAS(cfs) 65
    - version(2019-07-19) 65
      - API overview 65
      - CallingMethod 68
        - signaturev1 68
        - signaturev3 76
        - request 85
        - result 86
        - PublicParameter 89
      - APIs 91
        - BindAutoSnapshotPolicy 91
        - DescribeAutoSnapshotPolicies 93
        - DescribeCfsTags 95
        - UpdateCfsDrStatus 97
      - File system APIs 99
        - CreateCfsFileSystem 99
        - DeleteCfsFileSystem 102
        - DescribeCfsFileSystemClients 105
        - DescribeCfsFileSystems 107
        - DescribeCfsFileSystemsByVpc 110
        - DescribeMountTargets 113
        - UpdateAutoSnapshotPolicy 115
        - UpdateCfsFileSystemName 117
        - UpdateCfsFileSystemPGroup 119
        - UpdateCfsFileSystemSizeLimit 122
      - Other APIs 123
        - CopyFileSystemSnapshotCrossRegion 123
        - CreateAutoSnapshotPolicy 125
        - DeleteAutoSnapshotPolicy 127
        - UnbindAutoSnapshotPolicy 129
      - Permission Group APIs 131
        - BatchAddCfsRules 131
        - BatchDeleteCfsRule 134
        - CreateCfsPGroup 137
        - CreateCfsRule 139
        - DeleteCfsPGroup 140
        - DeleteCfsRule 142
        - DescribeCfsPGroups 145
        - DescribeCfsRules 147
        - UpdateCfsPGroup 149
        - UpdateCfsRule 150
      - Service APIs 151
        - DescribeAvailableZoneInfo 151
        - DescribeCfsServiceStatus 153
        - SignUpCfsService 154
      - Data structure 155
      - ErrorCode 159

Product Introduction
CFS Overview

Introduction

Cloud File Storage (CFS) provides a scalable shared file storage service that can be used with services such as Cloud Virtual Machine (CVM). CFS offers standard NFS file system access protocols to provide shared data sources for multiple CVM instances. It supports elastic capacity expansion and performance scaling. CFS can be mounted on existing applications without modification. As a highly available and reliable distributed file system, CFS is suitable for various scenarios such as big data analysis, media processing, and content management.

CFS is easy to integrate, eliminating a need to adjust the business structure or make complex configurations. To integrate and use CFS, complete three steps only: creating a file system and its mount point, launching a file system client on a server, and mounting the created file system.

Product Features

Integrated Management

CFS supports NFS v3.0/v4.0 protocol and POSIX access syntax (such as strong data consistency and file locking). Users can mount a file system by running the standard mount command on the corresponding operating system.

Automatic Expansion

CFS can automatically expand the storage capacity of a file system based on the file size without interrupting requests and applications during the process, thereby ensuring exclusive use of storage resources while reducing management workload.

Security Settings

CFS features extremely high availability and persistence. Each file stored in a CFS instance has 3 redundant copies. It supports access from VPC and access control.

Pay-as-You-Go

CFS is metered and billed by the actual usage of an instance. It allows multiple CVM instances to share the same storage space via the NFS protocol.

Product Strengths

Use Strengths

Integrated Management

- CFS can provide POSIX file system access semantics (such as strong data consistency and file locking). CVM instances of the cloud platform can use standard operating system mounting commands to mount CFS through the NFS v3.0/v4.0 protocol.
- CFS offers a console interface, allowing you to create and configure file systems quickly and easily, and saving the complexity of deploying and maintaining file systems.

Automatic Scaling

CFS can automatically scale out the storage capacity of a file system according to the size of the file capacity without interrupting requests and applications, ensuring exclusive access to the required storage resources, while reducing the time cost and workload of management.

Security and Reliability

- CFS adopts a three-replica distributed storage mechanism, delivering extremely high reliability. The system returns a response of successful writing only after confirming that the data has been written to all the three replicas. The backend data replication mechanism can quickly create a replica through data migration and other methods when any one of the replicas fails, so as to ensure that three replicas are always available and provide you with secure and reliable data storage services. CFS stores data across racks, with a reliability of 99.9999999% (nine 9s).
- CFS strictly controls the access permissions for a file system by combining security groups of basic networks or VPC networks with permission groups.

Low Costs

- CFS can dynamically adjust the required capacity, so that you do not need to adjust it in advance but pay by usage, with no minimum consumption threshold, deployment fees, or Ops fees involved.
- Multiple compute nodes can share the same storage space through the NFS protocol, so that you do not need to purchase other storage services again or consider a cache.

Storage Types, Performance, and Specifications

CFS provides the following two types of file systems. The following table describes the characteristics, strengths, and use cases of each type of file system.

File System Type	Characteristic	Strength	Use Case
Standard	- The storage space scales out elastically according to the amount of data written, up to 160 TiB. - The throughput scales out linearly according to the file system capacity, up to 300 MB/s. - The maximum IOPS is 4,000 (4 KB file size for random read/write).	- Low cost - Large capacity	Cost-effective, suitable for most file sharing scenarios, such as data backup, office OA, log storage, and small file sharing.
High-performance	- The storage space scales out elastically according to the amount of data written, up to 160 TiB. - The throughput scales out linearly according to the file system capacity, up to 500 MB/s. - The maximum IOPS is 26,000 (4 KB file size for random read/write).	- High throughput - Low latency	More suitable for medium to large workloads and low latency scenarios.

> Note:
>
> The file system type is changed from "Extreme" to "High-Performance". Only the name is changed, and features are not affected.

Performance Formulas

The throughput of standard and high-performance single file systems follows the definition of performance specifications. The specifications include storage capacity, which can be calculated based on either

allocated capacity or used capacity. Additionally, the step size for the performance calculation formula can be customized.

File System Type	Performance Formula
Standard	Throughput (MB/s) = Storage volume (GiB) x Custom step size of throughput + Custom lower specification limit
High-Performance	Throughput (MB/s) = Storage volume (GiB) x Custom step size of throughput + Custom lower specification limit

> Note:
>
> The above performance is the capability provided by the file system. To reach the performance limit of the file system, perform multi-threaded, high-load read-write operations on multiple compute nodes.

Use Cases

Enterprise File Sharing

Storage services provided by CFS are suitable for organizations which have a large number of employees and need to access and share the same data sets. Administrators can use CFS to create file systems and set read and write permissions for clients in the organization.

Big Data and Analysis

CFS provides the scale and performance necessary for big data applications. Its high throughput for computing nodes, read-after-write consistency, and low-delay file operations are particularly suitable for the centralized processing and analysis of server logsets.

Streaming Media Processing Service

Media workflows such as video editing, audio-visual production, broadcast processing, sound design and rendering often rely on shared storage to operate large files. A powerful data consistency model together with high throughput and shared file access can reduce the time required to complete above work.

Content Management and Web Services

CFS can serve as a highly durable and high-throughput file system, which is used for various content management systems, storing and providing information for websites, online distribution, archiving, and other applications.

Dedicated Software Environment

CFS provides the foundation for migrating traditional service architectures to the Clouds for sectors like retailing and finance. In general, dedicated software requires the sharing of the same CFS system, and it only supports POSIX standard protocol operations.

System Limits

Limit Description

Limits

- File system protocols supported by CFS: NFS v3.0/v4.0, CIFS, SMB 2.0, SMB 2.5, and SMB 3.0.
- Maximum capacity of a single standard file system: 160 TiB.
- A single standard file system allows no more than 1,000 compute nodes to be mounted.

Related Instructions

UID and GID Descriptions

- When NFS v3.0 is used: If the local account does not contain a UID or GID to which the file belongs, the UID and GID are displayed. If the local Linux account contains a UID or GID to which the file belongs, the corresponding username and group name are displayed according to the mapping relationship between the UID and GID in the local account.
- When NFS v4.0 is used: If the Linux kernel version is later than 3.0, the UID and GID rules are the same as those in NFS v3.0. If the local kernel version is earlier than 3.0, the UIDs and GIDs of all files are displayed as nobody.

Support for CIFS or SMB Protocols

- Protocol version support: CIFS and SMB 1.0 and later SMB protocol versions are supported. However, it is not recommended to use the SMB 1.0 protocol for mounting, as SMB 1.0 has significant deficiencies in performance and features compared to SMB 2.0 and later versions, mainly due to substantial differences in protocol design. In addition, Windows products that support SMB 1.0 or earlier versions have completely exited Microsoft's support lifecycle. Therefore, to ensure system security and efficiency, it is strongly recommended to use later protocol versions.
- Users are not allowed to access the same file system using NFS and SMB, or directly access the SMB file system through the WAN.
- It only provides read and write permission control at the file system level, but it will not provide ACL permission control at the file or directory level.
- IOCTL/FSCTL operations such as sparse files, file compression, NIC status query, and reparse point are not supported.
- Alternate data streams are not supported.
- Protocol features of SMB 3.0 or later, such as SMB Direct, SMB Multichannel, SMB Directory Leasing, and Persistent File Handle, are not supported.

> Note:
>
> When using NFS v4.0 to mount the file system under Linux kernel versions earlier than 3.0, it is not recommended to perform a change owner or change group operation on a file or directory. Otherwise, the UID and GID of the file or directory will change to nobody.

Purchase Guide
Billing Overview

Billing description

Billing Item	Billing Mode	Billing cycle	Billing cycle description
CFS Volume	Pay-as-you-go (postpaid)	Hour	The fee is deducted every hour, and the storage volume is calculated according to the maximum value of the actually used storage space per unit hour (peak)

Storage Space

When the file system is created, it occupies 32 MB of storage space by default. The storage capacity will not be counted in the actually used storage space.

Viewing Bills

Overview

This document describes how to view CFS bills in the console.

Operation Steps

1. Log in to the console, click Expenses in the upper right corner to enter the Billing Management page.
2. In the left sidebar, choose Bill Management > Bill Details > Bill by Instance.

Getting Started
Creating File Systems and Mount Targets

Overview

You can create the file system and mount point on the file system page through the CFS console. How to create the file system and mount point is described in detail below.

Operation step

1. Enter the file system page

Log in to the CFS console and click File System List in the left sidebar to enter the file system list page.

2. Create a file system

Click Create to pop up the window for creating a file system. Configure the following information in the pop-up window. After the operation is correct, click OK.

Field	Meaning
Region	Select the region where you want to create a CFS file system.
AZ	Select the availability zone for which you want to create a CFS file system.
Storage Class	Select the required storage type, general standard storage, or general performance storage.
File Service Protocol	Select the protocol type of file system, NFS or SMB. NFS protocol is more suitable for Linux/Unix Client, and CIFS/SMB protocol is more suitable for Windows Client.
Network Type	VPC - To share the file system with CVM under the virtual private cloud (VPC), you need to select the VPC when creating the file system. When the file system is a virtual private cloud, only CVM instances within the same virtual private cloud can be mounted if no special network settings are made. - If there is a need to share file systems across multiple networks, check Access File Systems Across Networks.
Permission Group	Each file system must be bound with a permission group, which specifies a set of accessible allowlist and read/write operation permissions.

Field	Meaning
Tag	If you already have a tag, you can add a tag for the new file system here; If you do not already have a tag, go to the tag console and create the required tag before binding it to the file system. Alternatively, you can add tags to the file system after it has been created.

3. Access Mount Point Information

1. After the file system is created, return to the file system list.
2. Click the file system name that has been created. The Basic File System Information page is displayed.
3. Click Mount Point Info to view and obtain the mount commands under Linux and Windows. It is recommended to copy the mount command provided by the console for mounting.

Using CFS on a Linux Client

This document describes how to use CFS on a Linux client.

Prerequisites

- Before creating a file system, ensure that the VPC and subnet are available. It is recommended that the subnet of the Linux-based CVM instance and the mount target of the CFS should belong to the same VPC and be deployed in the same availability zone to reduce network latency and achieve a better user experience. If no VPC or subnet is available, configure them.
- Before creating a file system, ensure that a Linux-based CVM instance is available; otherwise, purchase and configure it.

  > Note:
  >
  > Logging in to the CVM requires an administrator account (root account; for Ubuntu system users, ubuntu account) and the password set during the purchase process.

Operation Steps

Step 1: Creating a File System and Its Mount Point

For detailed steps, see Creating File Systems and Mount Points.

Step 2: Connecting to an Instance

This section describes how to log in to a Linux-based CVM instance. The login method varies depending on scenarios. Logging in to the instance through the console is described here.

1. In the operation column of the CVM Instances list, click Log In to connect to the Linux-based CVM instance through VNC.
2. Enter the account and password for login.

   > Note:
   >
   > This instance is exclusive, meaning that only one user can log in through the console at a time.

Step 3: Verifying Network Communication

Before mounting, check the network connectivity between the client and the file system. You can use the telnet command for verification, for example, telnet 192.168.1.1 111. The specific protocols and open ports are as follows:

File System Protocol	Open Port	Check the Network Connectivity of a Specific Port
NFS 3.0	111, 892, and 2049	telnet 111, 892, and 2049

File System Protocol	Open Port	Check the Network Connectivity of a Specific Port
NFS 4.0	2049	telnet 2049
CIFS/SMB	445	telnet 445

> Note:
>
> If the connection fails, check the security group rules of the client to ensure that access to the above ports is allowed, and ensure that the IP address of the cloud file system is included within the source and destination IP ranges.

Step 4: Mounting a File System

Method 1: Mounting via the Console

You can install the NFS client via the console and mount the corresponding file system. For detailed operation steps, see Mounting a File System via the Console or CFS Client Assistant.

Method 2: Mounting via the Command Line

Mounting an NFS File System

1. Install and launch the NFS client. Before mounting, make sure that nfs-utils or nfs-common has already been installed in the system. The installation method is as follows:
   - CentOS:

     sudo yum install nfs-utils

   - Ubuntu or Debian:

     sudo apt-get install nfs-common

2. Create a destination mount directory with the following command:

   mkdir <destination mount directory>

   Example:

   mkdir /localfolder/
   mkdir /localfolder/test

3. Mount the file system.

Mount NFS v4.0 with the following command:

```
//Go to the CFS console > File System > Mount Target Info to obtain the following command. As some old file system versions do not support the noresvport parameter, it is recommended to use the version-specific commands provided in the console. After norevsport is configured, a new TCP port can be used for reconnection, which guarantees that the client and file system can stay connected during network recovery. Therefore, the norevsport parameter is recommended.
//Some old versions of Linux kernels need to be mounted using vers=4. If an exception occurs when vers=4.0 is used, change it to vers=4.
sudo mount -t nfs -o vers=4.0,noresvport <mount point IP>:/ <destination mount directory>
```

- Mount point IP: It is automatically generated when the file system is created.
- By default, NFS v4.0 is mounted under the root directory / of the file system. It can also be mounted to a subdirectory created in the file system.
- Destination mount directory: This refers to the target directory to be mounted to on the current server. It needs to be created in advance.

> Note:
>
> Note that there is a space between <mount point IP>:/ and <destination mount directory> .

Example:

- Mount to the root directory of CFS:

```
//Go to the CFS console > File System > Mount Target Info to obtain the following command. As some old file system versions do not support the noresvport parameter, it is recommended to use the version-specific commands provided in the console. After norevsport is configured, a new TCP port can be used for reconnection, which guarantees that the client and file system can stay connected during network recovery. Therefore, the norevsport parameter is recommended.
//Some old versions of Linux kernels need to be mounted using vers=4. If an exception occurs when vers=4.0 is used, change it to vers=4.
sudo mount -t nfs -o vers=4.0,noresvport 10.0.24.4:/ /localfolder
```

- Mount to the subdirectory of CFS:

```
//Go to the CFS console > File System > Mount Target Info to obtain the following command. As some old file system versions do not support the noresvport parameter, it is recommended to use the version-specific commands provided in the console. After norevsport is configured, a new TCP port can be used for reconnection, which guarantees that the client and file system can stay connected during network recovery. Therefore, the norevsport parameter is recommended.
//Some old versions of Linux kernels need to be mounted using vers=4. If an exception occurs when vers=4.0 is used, change it to vers=4.
sudo mount -t nfs -o vers=4.0,noresvport 10.0.24.4:/subfolder /localfolder
```

Mount NFS v3.0 with the following command:

```
sudo mount -t nfs -o vers=3,nolock,proto=tcp,noresvport <mount point IP>:/<fsid> <destination mount directory>
```

- Mount point IP: It is automatically generated when the file system is created.
- Destination mount directory: This refers to the target directory to be mounted to on the current server. It needs to be created in advance.

> Note:
>
> Note that there is a space between <mount point IP>:/<FSID> and <destination mount directory> .

Below is an example of mounting to a CFS subdirectory:

```
sudo mount -t nfs -o vers=3,nolock,proto=tcp,noresvport 10.0.24.4:/z3r6k95r /localfolder
```

1. View the mount point information.
   After the mount is completed, run the following command to view the mounted file system:

   ```
   mount -l
   ```

   If there is a lot of information on the page, for easy queries, you can use grep to filter out the desired file system type.

   ```
   mount -l | grep nfs //Used to filter out all NFS file systems.
   mount -l | grep lustre  //Used to filter out all Turbo file systems.
   ```

   Alternatively, run the following command to view the capacity information about the file system.

   ```
   df -h
   ```

> Note:
>
> - To avoid misoperations, do not repeatedly mount a mounted CFS file system or file system directory unless it is necessary.
> - If an overlapping or inclusion relationship exists in the directory to be mounted, it is impossible to display all file system mount results by using df -h only. It is recommended to use it in combination with mount -l to view all mount point information.

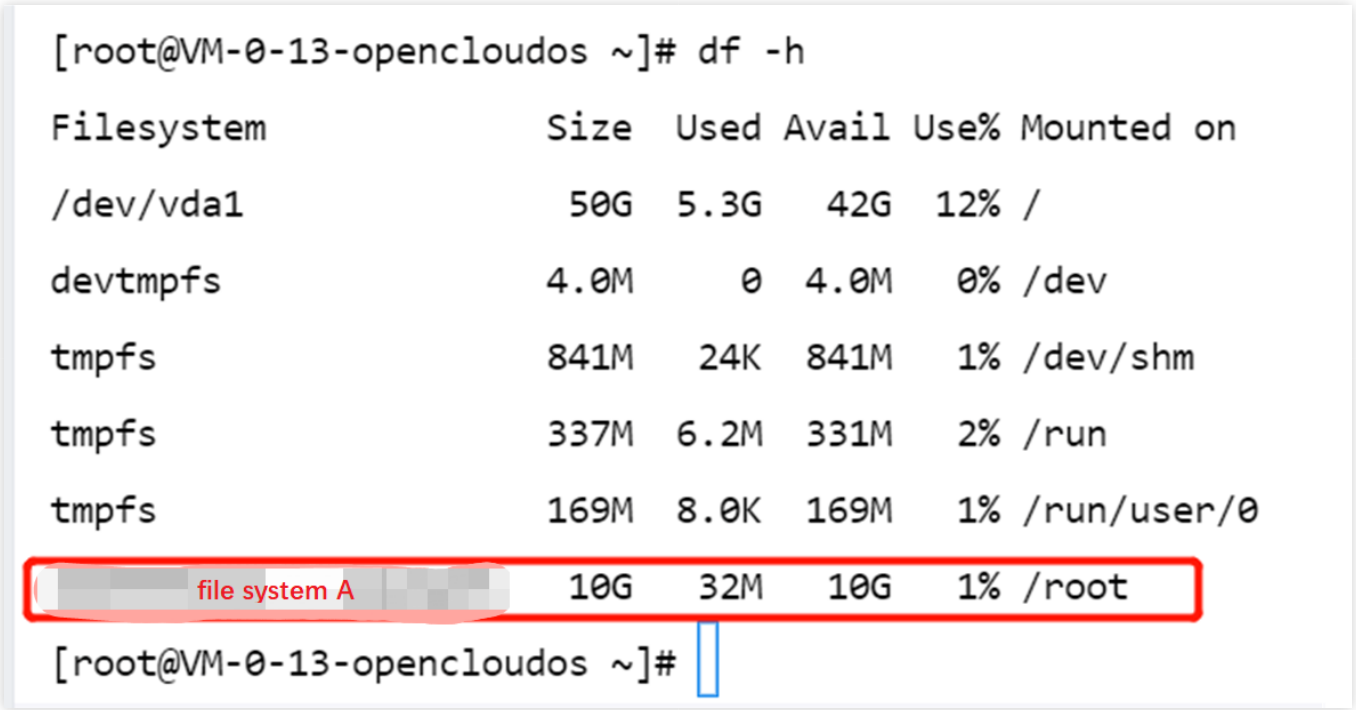
For example, when file system A is mounted to both /root/ and /root/localfolder/ , using mount -l can query all mount target information, while using df -h can only query the file systems under /root/ . After the file systems in the root directory are unmounted, executing df -h can query the file systems in the /root/

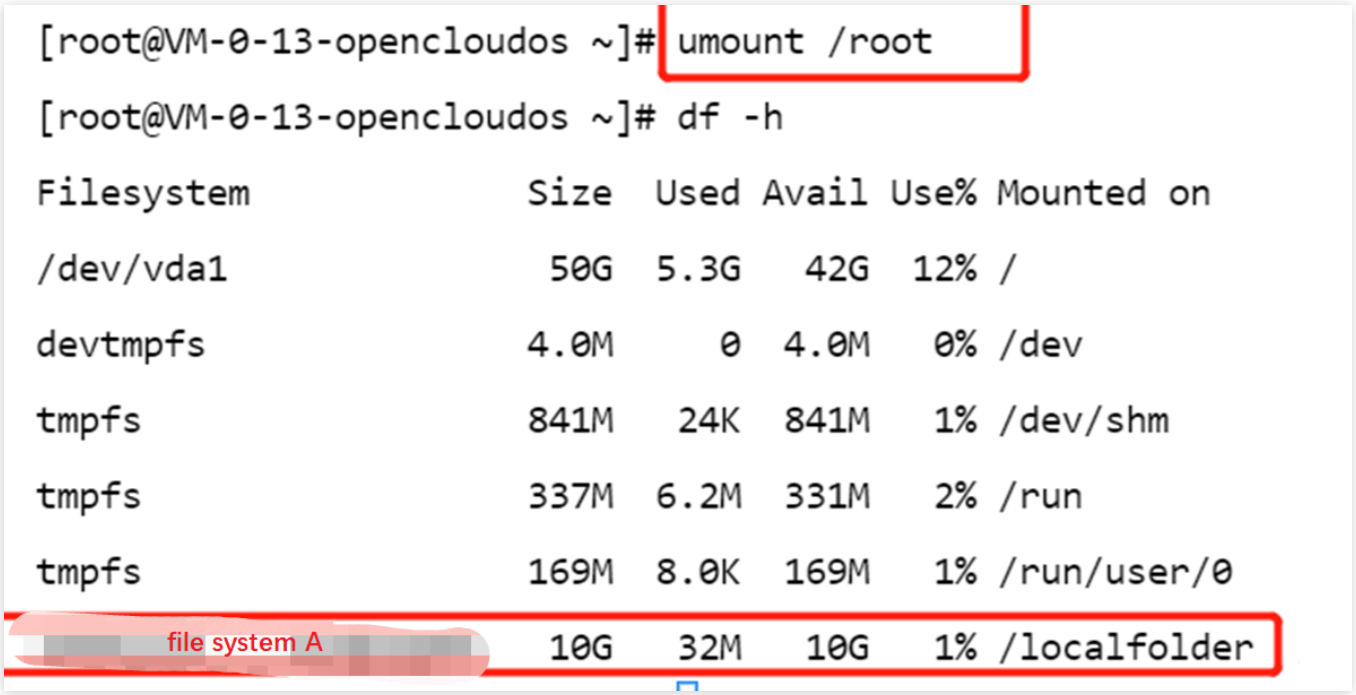
localfolder/ directory.

- mount -l execution result:



- df -h execution result:





Mounting a CIFS/SMB File System

1. Install and launch the CIFS client. Before mounting, make sure that cifs-utils has already been installed in the system. The installation method is as follows:

   sudo yum install cifs-utils.x86_64 -y //CentOS

2. Create a destination mount directory using the following command:

   mkdir <destination mount directory>

   Example:

   mkdir /local/
   mkdir /local/test

3. The command used to mount file systems is as follows:

   # Parameter description:
   # vers: supports protocol versions 2.1 and 3.0. Version 1.0 is used by default.
   # uid: owner of the file when the mount is completed. If not specified, uid=0 is used by default.
   # gid: user group that owns the file when the mount is completed. If not specified, gid=0 is used by de fault.
   # The uid and gid parameters are for apps that need to check the file owner during execution. In this c

ase, you need to set the uid and gid to the app account.
noperm: The client does not perform permission verification. If a permission is denied, add this para meter.
actimeo: file attribute (metadata timestamps) cached by the client.
nocase: By default, using version 1.0 on mount is case-sensitive. However, Windows clients are case-i nsensitive. If you create a file on the Linux CIFS client and the file name already exists on the Windows client, an exception may occur when you access the file through the Windows client.
Example:
mount -t cifs -o guest,vers=1.0,uid=1000,gid=100,noperm,actimeo=1,nocase //<mount point IP>/<FS ID> /<destination mount directory>

- Mount point IP: It is automatically generated when the file system is created.
- By default, the FSID of the file system is used for the mount.
- Destination mount directory: This refers to the target directory to be mounted to on the current server. It needs to be created in advance.

  > Note:
  >
  > Note that there is a space between <FSID>/ and <destination mount directory> .

Example:

```
mount -t cifs -o guest //10.66.168.75/vj3i1135  /local/test
```

4. View the mount point information.
   After the mount is completed, run the following command to view the mounted file system:

```
mount -l
```

If there is a lot of information on the page, for easy queries, you can use grep to filter out the desired file system type:

```
mount -l | grep cifs //Used to filter out cifs file systems.
```

Alternatively, run the following command to view the capacity information about the file system.

```
df -h
```

> Note:
>
> - To avoid misoperations, do not repeatedly mount a mounted CFS file system or file system directory unless it is necessary.
> - If an overlapping or inclusion relationship exists in the directory to be mounted, it is impossible to display all file system mount results by using df -h only. It is recommended to use it in combination

> with mount -l to view all mount point information.

Step 5: Unmounting a Shared Directory

If you need to unmount a shared directory in some cases, run the following command, where "directory name" is the root directory or the full path of the file system.

```
sudo umount <directory name>
```

Example:

```
sudo umount /local/test
```

> Note:
>
> If an overlapping or inclusion relationship exists in the directory to be mounted, it is impossible to display all file system mount results by using df -h only. It is recommended to use it in combination with mount -l to view all mount target information.

Step 6: Terminating Resources

You can terminate a file system in the console. That is, you can go to Manage File Systems, locate the file system to be terminated, and then click Delete and Confirm.

> Note:
>
> Resources cannot be recovered from a deleted file system. Therefore, it is recommended to use the snapshot feature to back up all resources before deleting the file system.

Using CFS on a Windows Client

Overview

This document describes how to use CFS on a Windows client. This guide uses Windows Server 2012 R2 as an example. The operating methods on other versions of operating systems, such as Windows Server 2008 and Windows Server 2016, are the same.

Creating a File System and Its Mount Point

For detailed steps, see Creating File Systems and Mount Targets.

Connecting to an Instance

This section describes how to log in to a Windows-based CVM instance.

Prerequisites

You have logged in to the CVM instance using an administrator account and its corresponding password.

- Administrator account: Administrator for all Windows-based instances.
- Password: the password specified when purchasing the CVM instance.

Logging In to a CVM Instance in the Console

1. In the operation column of the CVM instance list, click Log In to connect to the Windows-based CVM instance through VNC.
2. Click in the upper left corner and send the Ctrl-Alt-Delete command to go to the System Login page.
3. Enter the account (Administrator) and password to log in.

> Note:
>
> This instance is exclusive, meaning that only one user can log in through the console at a time.

Verifying Network Communication

Before mounting, check the network connectivity between the client and the file system. You can use the telnet command on the Windows client for verification. The specific protocols and open ports are as follows:

File System Protocol	Open Port	Check Network Connectivity
NFS 3.0	111, 892, and 2049	telnet 111, 892, or 2049
NFS 4.0	2049	telnet 2049
CIFS/SMB	445	telnet 445

> Note:

> Currently, CFS does not support ping.

Mounting a CIFS/SMB File System

You can mount a CIFS/SMB file system via the command line or graphical interface.

Mounting a File System via the Command Line

Use FSID to mount the file system. The mount command is as follows:

```
net use
```

Example:

```
net use X: \\10.10.11.12\fjie120
```

> Note:
>
> You can obtain FSID by clicking CFS Console > File System Details > Mount Point Info.

Mounting a File System via the Graphical Interface

1. Click  to go to the start menu interface.
2. If you are using Windows 10 or Windows Server 2019 or later, modify the registry and allow Guest access to SMB.

   > Note:
   >
   > Microsoft officially explains the changes related to this permission as follows:
   >
   > Guest access in SMB2 and SMB3 disabled by default in Windows

   Run the following instruction via CMD with the administrator command:

   ```
   reg add "HKLM\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters" /v "AllowInsecu
   reGuestAuth" /t REG_DWORD /d 1 /f && net stop lanmanworkstation /y && net start lanmanworkstati
   on && net start SessionEnv
   ```

3. Right-click This PC and select Map Network Drive.
4. In the pop-up window, set the "Drive" letter name and folder (that is, the mount directory seen in the CIFS/SMB file system) and click Finish.
5. Enter the mounted file system and right-click to create a file to verify the correctness of reading and

writing.

Mounting an NFS File System

1. Enabling NFS Service

Before mounting, ensure the system has started the NFS service.



1. Click , choose Control Panel > Programs > Turn Windows features on or off.
2. In the opened Add Role and Features Wizard window, keep the default configuration and click Next for five times continuously.
3. On the feature interface, select NFS Client and click Next.
4. Click Install.
5. Restart the CVM to finish enabling the Windows NFS client service.

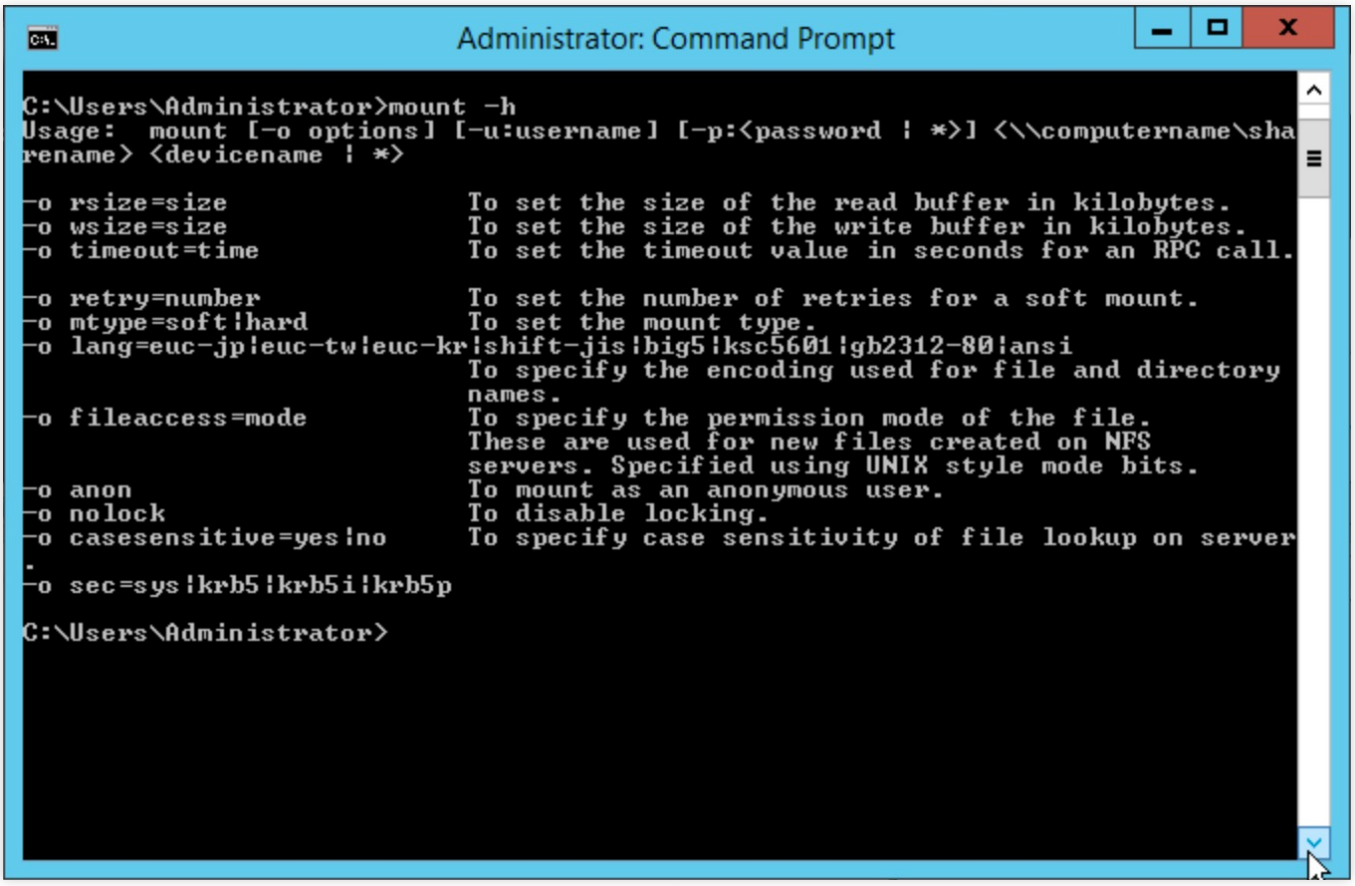
> Note:
>
> Windows Home does not include an NFS client option, so you can use the Windows Server operating system or Windows Professional.

2. Verifying Whether the NFS Service Is Started

1. Open the CMD Command Line Interface and execute the following command:

```
mount -h
```

2. If NFS-related information is returned, it indicates that the NFS client is running normally.



3. Adding Anonymous Access Users and User Groups

1. Run the CMD Command Line Interface as an administrator, execute the following instructions to modify the registry, allow anonymous user/user group access, and set the Mount's Lock parameter to No (Windows only supports NFS v3.0, which does not support the file lock service).

   ```
   reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ClientForNFS\CurrentVersion\Default" /v Anon
   ymousUid /t REG_QWORD /d 0 /f && reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ClientF
   orNFS\CurrentVersion\Default" /v AnonymousGid /t REG_QWORD /d 0 /f && reg add "HKEY_LOCAL_
   MACHINE\SOFTWARE\Microsoft\ClientForNFS\CurrentVersion\User\Default\Mount" /v Locking /t REG
   _DWORD /d 0 /f
   ```

2. To make the modified registry take effect, run the following command to restart the NFS client service or directly restart the Windows system.

   ```
   net stop nfsclnt && net stop nfsrdr && net start nfsrdr && net start nfsclnt
   ```

4. Mounting an NFS File System

You can mount a file system via the graphic interface or the CMD command line.

- Mount a file system via the CMD command line
  In the CMD Command Line Interface, enter the following command to mount the file system. Among them, the default subdirectory of the system is FSID.

  ```
  mount -o nolock mtype=hard
  ```

  Example:

  ```
  mount -o nolock mtype=hard 10.10.0.12:/z3r6k95r X:
  ```

> Note:
>
> You can obtain the FSID mount command by clicking File System Console > File System Details > Mount Point Info.

Uninstalling a File System

Uninstalling a Shared Directory via the Graphical Interface

To disconnect the mounted file system, right-click the disk and click Disconnect in the menu that appears. You can just do this to disconnect the file system.

Uninstalling a Shared NFS Directory via the CMD Command

To uninstall a shared directory in some cases, start the command line terminal and run the following command, where Directory Name is the full path of the root directory or the file system.
Example of NFS:

```
umount X:
```

Example of SMB:

```
net use x: /del
```

Terminating Resources

> Note:
>
> After the file system is deleted, the resources cannot be recovered. It is recommended to back up the resources before deleting the file system.

You can terminate a file system from the console. Go to the CFS console, select the file system to be terminated, and then click Delete and Confirm to delete it.

Installing CFS Service Monitoring Plugin on Clients

Overview

The CFS service monitoring plugin is a plugin integrated on the cloud monitoring component to monitor the performance of the CFS file system and client connections. It is strongly recommended to install the CFS service monitoring plugin on clients such as Cloud Virtual Machine or containers that adopt CFS for better management of CFS services.

Operating System Support

The CFS service monitoring plugin currently supports client monitoring over NFS (CIFS/SMB not supported) and supports the following platforms. Other Linux kernel version systems may have compatibility problems.

Operating System	Version
CentOS	CentOS 8.2 X86_64 Architecture CentOS 8.0 X86_64 Architecture CentOS 7.9 X86_64 Architecture CentOS 7.8 X86_64 Architecture CentOS 7.7 X86_64 Architecture CentOS 7.6 X86_64 Architecture CentOS 7.5 X86_64 Architecture CentOS 7.4 X86_64 Architecture CentOS 6.7 X86_64 Architecture
Ubuntu	Ubuntu Server 20.04 64-bit Ubuntu Server 18.04.1 64-bit Ubuntu Server 16.04.1 LTS 64-bit

Description

The principle and main features of the plugin are as follows:

- Principle: The plugin reads data from NFS related logs in `/proc/self/mountstats` and `/var/log/messages` , and the data read is related to performance, status, and mounted parameters. It does not involve actual user business data.
- Main features: Read the status of client mount points using CFS, detect the availability of mount

points and collect performance-related data.

Linux Installation Guide

For more information, see Cloud Monitor Component Installation Guide.

Windows Installation Guide

Windows clients are not supported yet.

Monitoring Incoming Clients

In the CFS console, click the file system name to enter the details of the file system. Select Mounted Client to view the client information on which the file system has been mounted. If no plugin is installed in the client, the information cannot be obtained.

> Note:
>
> There will be a delay of 1-3 minutes for client information display.

Using the Console or CFS Client Assistant to Mount File Systems

Overview

This document details how to use the console or CFS client assistant to install the NFS client and mount file systems.

Prerequisites

- You have succeeded in Creating File Systems and Mount Targets.
- To install the NFS client and mount general-purpose file systems, see Using CFS File Systems on Linux Clients for details.
- The CVM has the TAT agent installed before using the console or the CFS client assistant to mount file systems.

Operation Steps

Mounting File Systems Through the Console List Page

It is recommended to use the console to mount file systems. This method is easier and faster.

1. Log in to the CFS console > File System.
2. In the Action column of the CFS list, click Mount.
3. Switch the number of items displayed per page, and select the CVMs to perform the mounting operation in batches. After confirming that everything is correct, click Next. You can scroll through the list to view the configuration status of the Restart automatic mount feature. For details, see Automatically Mounting File Systems.

- File protocol client installation status and description:

Status	Description
Not installed	The file protocol client is not installed. Click Next to install it automatically.

Status	Description
Installed	The file protocol client is installed. Clicking Next will not affect the existing configuration.
Automatic installation unsupported	The TAT Agent is not installed.

4. Go to Install/Update Client. After successful installation, click Next.
5. Specify mount parameters, click Next. You can configure the automatic mounting of file systems upon server restart as needed. For details, see Automatically Mounting File Systems.
6. After successful mounting, click Complete.

Mounting File Systems Through the CFS Client Assistant

The main difference between operations through the CFS client assistant and operations through the console is that the CFS client assistant supports the installation of NFS clients in batches, and mount parameters need to specify a specific file system.
The following are the operation steps. You can execute them as needed.

1. Log in to the CFS console > File System.
2. Click CFS Client Assistant above the CFS list.
3. Select CVM, and click Next.
4. Go to Install/Update Client. After successful installation, click Next.
5. Select the file system to be mounted, specify mount parameters, and click Next. You can configure the automatic mounting of file systems upon server restart as needed. For details, see Automatically Mounting File Systems.
6. After successful mounting, click Complete.

Operation Guide
Cloud Access Management(CAM)

Scenarios

Cloud File Storage (CFS) already supports resource-level access permission management, that is, the root account can grant specified operation permissions to specified resources for specified users and user groups. After authorization, the CFS console and API will allow or prohibit user operations according to the user's authorization.

This guide describes how to authorize read-only, read-write, and custom policies for CFS for users. For more information on the principles and guidelines of cloud access management, see CAM.

Creating Access Control Policy

Log in to the policy management page of the cloud access management console.

- If you need to grant user permissions quickly, you can search for CFS in the search box on the right side of the policy management interface, select preset CFS read-only or read/write permissions and associate user groups to complete authorization.
- If you need to grant permissions to users for specific actions, you can create a custom policy and associate user groups to complete authorization.

CFS Full Access Policy

If you want the user to have permissions for all operations such as query, creation, modification, and deletion, you can grant the user QcloudCFSFullAccess permission. The policy syntax of using preset QcloudCFSFullAccess to grant collaborators or sub-users read/write all CFS resources and query permissions for VPC and subnet is as follows:

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

```
        {
            "action": [
                "vpc:DescribeVpcEx",
                "vpc:DescribeSubnetEx"
            ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

CFS Read-Only Policy

If you want the user to have query permission but do not have creation, modification, and deletion permissions, you can grant the user QcloudCFSReadOnlyAccess permission. The policy syntax of using preset QcloudCFSReadOnlyAccess to grant collaborators or sub-users read/write all CFS resources and query permissions for VPC and subnet is as follows:

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cfs:Describe*"
            ],
            "resource": "*",
            "effect": "allow"
        },
        {
            "action": [
                "vpc:DescribeVpcEx",
                "vpc:DescribeSubnetEx"
            ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

Custom Policy

Custom policy is a more flexible way to authorize users, and the cloud access management console provides multiple ways to generate policies. This case describes how to create a custom policy by using Create by Policy Builder (for other methods, see Policy document).

The policy builder page provides visual policy configuration. You can automatically generate the policy

code by selecting parameters, which is suitable for users with CAM authorization for the first time.

On the policy management console policy page, choose Create Custom Policy >Created by Policy Generator. On the create policy page, use the policy builder to add multiple statements to a custom policy. The configuration instructions are as follows:

Parameter	Corresponding policy parameters	Options and Effects
Effect	Effect	Allow or Deny
Service	Service	Here, select CFS
Action	Action	All the operation types supported by CFS
Resource	Resource	Operable resources: - The writing format of all resources for file storage is * - The writing format of all resources in the specified region is qcs::cfs:ap-city::* - The writing format of all resources in all regions under the specified user is qcs::cfs::uin/27700000:* - The writing format of all file systems in the specified region under the specified user is qcs::cfs:ap-city:uin/27700000:filesystem/* - The system writing of the specified user group under the specified user is qcs::cfs::uin/27700000:pgroup/pgroup-doxpcqh Note: The UIN in the policy must be the root account UIN (the subsequent file system or permission group resources must belong to this root account)
Condition	Condition	See Effective Conditions for the conditions under which this policy takes effect, and how to set it up

Among them, APIs of CFS, API features and authorization precautions are listed below. Resource options can be configured according to the following instructions.

- Semantic consistency: Accurately corresponds to the cloud computing context of the original "interface category".	API Name	API Description	Permission type	Note
Service interface	SignUpCfsService	Activate CFS service	Write permission	Authorize this API without specifying resources
Service interface	DescribeCfsServiceStatus	Query CFS service provisioning status	Read Permission	Authorize this API without specifying resources
File system API	DescribeCfsFileSystems	List the file system	Read Permission	To authorize this API, the resource should be specified as *
File system API	CreateCfsFileSystem	Creating a file system	Write permission	To authorize this API, specifying file system resources is not required
File system API	UpdateCfsFileSystemName	Update file system name	Write permission	Authorization of this API requires specifying file system resources
File system API	UpdateCfsFileSystemPGroup	Updating the permission group of the file system	Write permission	Authorization of this API requires specifying file system resources

- Semantic consistency: Accurately corresponds to the cloud computing context of the original "interface category".	API Name	API Description	Permission type	Note
File system API	UpdateCfsFileSystemSizeLimit	Update file system quota	Write permission	Authorization of this API requires specifying file system resources
File system API	DeleteCfsFileSystem	Deleting a file system	Write permission	Authorization of this API requires specifying file system resources
File system API	DescribeMountTargets	Query mount point	Read Permission	Authorization of this API requires specifying file system resources
Permission group API	DescribeCfsPGroups	List permission groups	Read Permission	To authorize this API, the resource should be specified as *
Permission group API	CreateCfsPGroup	Create permission group	Write permission	Authorize this API without specifying resources
Permission group API	UpdateCfsPGroup	Update permission group information	Write permission	Authorization of this API requires specifying permission

- Semantic consistency: Accurately corresponds to the cloud computing context of the original "interface category".	API Name	API Description	Permission type	Note
				group resources
Permission group API	DeleteCfsPGroup	Deleting permission group	Write permission	Authorization of this API requires specifying permission group resources
Permission group API	DescribeCfsRules	List permission groups rules	Read Permission	Authorization of this API requires specifying permission group resources
Permission group API	CreateCfsRule	Create permission group rules	Write permission	Authorization of this API requires specifying permission group resources
Permission group API	UpdateCfsRule	Update permission group rule information	Write permission	Authorization of this API requires specifying permission group resources
Permission group API	DeleteCfsRule	Delete permission	Write permission	Authorization of this API

- Semantic consistency: Accurately corresponds to the cloud computing context of the original "interface category".	API Name	API Description	Permission type	Note
		group rules		requires specifying permission group resources
Key-related APIs	DescribeKmsKeys	Query KMS key	Read Permission	To authorize this API, the resource should be specified as *

> Note:
>
> In addition, since the CFS file system uses the vpc IP, you need to obtain permissions for the "vpc: DescribeVpcEx" and "vpc: DescribeSubnetEx" APIs on pages such as creating file systems, listing CFS file systems, and querying file system details (you cannot query and create without granting them). It is strongly recommended that you add the authorization of all VPC resources by these two APIs to all authorized CFS policies. For detailed policy writing, see the QcloudCFSReadOnlyAccess policy statement.

After the above parameters are set, click Add Claim to add a statement for this custom policy. You can repeat the above steps to add multiple statements. If there are duplicate or conflicting policies, see Syntax Structure for their relationship and effectiveness.

The writing format of the policy is as follows. Each policy can have multiple statements.

```
{
    "version": "2.0",
    "statement": [{
        "effect": "Effect",
        "action": [
            "Action"
        ],
```

```
        "resource": "Resource"

    }]
}
```

For example, the permission policy syntax for prohibiting users from deleting and updating quotas on several file systems is as follows.

```
{
    "version": "2.0",
    "statement": [{
        "effect": "deny",
        "action": [
            "name/cfs:DeleteCfsFileSystem",
            "name/cfs:UpdateCfsFileSystemSizeLimit"
        ],
        "resource": [
            "qcs::cfs::uin/2779643970:filesystem/cfs-11111111",
            "qcs::cfs::uin/2779643970:filesystem/cfs-22222222",
            "qcs::cfs::uin/2779643970:filesystem/cfs-33333333"
        ]
    }]
}
```

Authorizing Users/User Groups/Roles

If the permission provided by the system is selected, you can directly search for QcloudCFSFullAccess or QcloudCFSReadOnlyAccess or other custom policies in the policy details list. Click Associate User/Group/Role in the operation column on the right side of the list, and find and check the user or user group to be authorized in the pop-up window. Finally, click OK to complete the authorization.

Canceling User/User Groups/Role Authorization

To cancel the permission of an authorized user, check the user/user group to be canceled in the Associate User/Group/Role list on the corresponding policy details page, and then click Release User/User Group/Role. After confirming the cancellation, the user/user group will lose permission to operate CFS resources for CFS.

Managing File System

Overview

Users can view the currently created file system through the Cloud File Storage (CFS) console, and manage the created file system, such as viewing the file system status and usage, file system details and mount point information.

> Note:
>
> When the file system is in the Creating status, it will be unable to view the details of the file system and perform deletion operations.

Prerequisites

Log in to the CFS console. The file system list page is displayed.

Viewing File System Status and Usage

On the file system List page, you can view the current file system usage and status. At the same time, CFS supports searching for table contents by file system name, ID, VPCID and IP.

View File System Information

On the file system list page, click a file system name to go to the file system details page. On the file system details page, you can view the basic information of the file system, mount point information, and mounted client information.

Basic Information of File System

The basic information of the file system shows the region, file system ID, name, file service protocol, file system status and creation time. On this page, you can set the file system name.

Mount Point Information

The mount point information of NFS file system includes network information, permission group and recommended mount commands. On this page, you can modify the file system permission group.

Mounted Client Information

Mounted Client tab will display the client information of the mounted file system, but it is necessary to install CFS monitoring plugins on the client. If no plugin is installed, the client information cannot be obtained. For installation guidance, see CFS monitoring plugin usage guide.

> Note:
>
> There will be a delay of 1-3 minutes for client information display.

Modify File System Name

On the file system list page, click the file system whose name needs to be modified. The file system details page is displayed. Click  on the right side of the instance name to modify it.

Deleting a file system

When you no longer need to use a file system, you can find the file system to be deleted in the file system list and click Delete on the right side to delete it.

> Note:
>
> To avoid client system exceptions, disconnect the file system from each client before deleting the file system.

Permission Management

Overview

The visiting client must be in the same network as the file system, and the created file system needs to configure a permission group to further manage the access permissions and read/write permissions of these visiting clients. The operation methods related to permissions are described below.

Directions

Create permission group

1. Log in to the CFS console and click Permission Group in the left sidebar.
2. On the Permission Group page, click Create to create a permission group. Configure the permission group name and remarks in the pop-up Create window.

Creating a permission group rule

In the rule list, you can add, edit, or delete rules. If no rules are added to the permission group, all are allowed. Rules are described as follows.

Field	Description
Visiting address	You can write a single IP or IP range, such as 10.1.10.11 or 10.10.1.0/24. The default incoming address is *, which means all addresses are allowed. At the same time, it should be noted that the private IP address of CVM needs to be filled in here.
Read/Write Permission	Read-only or read/write.
User Permission	There are 4 options for imposing permission restrictions on visiting users. - all_squash: any visiting user will be mapped to an anonymous user or user group. - no_all_squash: A visiting user will be first matched with a local user, and if the match fails, it will be mapped to an anonymous user or user group. - root_squash: A visiting root user will be mapped to an anonymous user or user group. - no_root_squash: A visiting root user will be allowed to maintain root account permissions. Note: Each file system defaults to 755 permissions, and nfsnobody does not have the write permission. Therefore, if there is no special requirement here, it is recommended to configure no_root_squash. If the root user is used to create a file directory and mount the file system, when the visiting IP is set as all_squash or

Field	Description
	root_squash, the visiting IP can only read files. (Since the mount path is root permission and the visiting IP has been mapped to an anonymous user).
Priority	The priority is a positive integer from 1 to 100, where 1 is the highest and 100 is the lowest. Regardless of priority setting, the priority of a single IP will be higher than that of this IP in the IP range. In other cases, the rules with higher priorities will be matched first and take effect.

Configuring a permission group for a file system

The configuration of a permission group can be modified after the file system is created. You can choose to create a permission group first and select it when creating a file system. You can also select the default permission group when creating a file system and then go to the File System Details page to change the permission group.

> Note:
>
> If the file system is mounted with the NFSv4 protocol, after modifying the permission group rules for the file system, you need to wait for 2 minutes for the configuration to take effect.

Modifying the information and rule of a permission group

You can enter the Permission Group Details page to modify the name, remarks, and rule of a permission group.

Monitoring File Systems

Overview

Before using the CFS monitoring feature, install the CFS service monitoring plugin on the CVM client that uses CFS to better manage the CFS service.

Installing the Monitoring Plugin

For specific operations, see Installing the CFS Service Monitoring Plugin on the Client.

Performance Monitoring

Go to the Cloud Monitor console, choose Cloud Product Monitoring > CFS. On the CFS Monitoring page, you can view the health status of each file system and the alarm information.

Viewing the Performance Monitoring Metrics of a File System

Click the ID/name of any file system to go to the Details page.
Users can select the corresponding time window to view the usage of the monitored file system.

The monitoring metrics supported by CFS are as follows:

Metric Name	Description	Unit	Level
Storage capacity	File system storage capacity at the current time	GB	Single file system
Read bandwidth	Average amount of data read from a file system per second	KB/s	Single file system
Read IOPS	Average number of reads from a file system per second	count	Single file system
Write bandwidth	Average amount of data written to a file system per second	KB/s	Single file system
Write IOPS	Average number of write operations on a file system per second	count	Single file system
Read latency	Latency in reading data from a file system	ms	Single file system

Metric Name	Description	Unit	Level
Write latency	Latency in writing data to a file system	ms	Single file system
Number of connected clients	Number of clients connected to the file system	%	Single file system
Success rate of client health monitoring	Success rate of client health status monitoring	%	Single file system
Capacity utilization	Ratio of used storage capacity to total available capacity	%	Single file system
inode utilization	Ratio of assigned inodes (index nodes) to total inodes	%	Single file system

> Note:
>
> Due to the local cache on the client, the read bandwidth and read IOPS of the file system obtained by Cloud Monitor may be slightly less than the actual usage.

Viewing the Alarm Policies of a File System

On the Alarm Policy page, users can view the alarm policies bound to each file system. Click the policy name to view policy details or perform unbinding operations.

Configuring an Alarm

The cloud platform allows users to configure alarm and notification policies for the file system based on monitoring metrics. If you want to create a policy or bind a policy to the file system, go to the Cloud Monitor Alarm Policy page.

Choose Cloud Monitor > Alarm Configuration > Alarm Policy to enter the Alarm Policy page for operations. For detailed operations, see the document titled Cloud Monitor Alarm Service Guide.

Managing Snapshots
Creating Snapshot

Both Standard and High-Performance Type NAS support snapshot creation. This document mainly introduces the operation of creating snapshots.

Overview

A snapshot can be created through the file system to save the file system data at a specified time. Snapshots are created incrementally. That is to say, only newly changed data compared with the previous snapshot will be created. Therefore, snapshot creation can be completed in a shorter time when data size change is not significant. Although the snapshot is created incrementally, deleting it does not prevent you from using any of the snapshot data, and unremoved snapshots restore the file system to that snapshot status.

Prerequisites

- You have successfully created a file system.

Note

- The file system must be in available status, and snapshots cannot be created temporarily in other status;
- The snapshot only retains the data written at the time of creation, and does not retain the data that has not been written to the hard disk in the memory at that time. It is strongly recommended that you shut down before creating a snapshot or ensure the data has been written and pause reading/ writing.
- When the snapshot has not been created completely, you cannot create another snapshot of the file system.
- Creating snapshots may slightly degrade the performance of the file system. It is recommended that you avoid peak business hours when creating snapshots.
- Manually created snapshots will be permanently retained and deleted if no longer needed.

Directions

1. Log in to the CFS Console.
2. On the file system list page, click Create Snapshot on the right side of the target file system.
3. Enter the snapshot name, then click OK.

Scheduled Snapshot

Feature Overview

CFS provides a scheduled snapshot feature, enabling developers to flexibly set backup task policies.
It is recommended to adopt different scheduled snapshot policies for different business scenarios. The recommended settings are as follows:

Business Scenario	Snapshot Frequency	Snapshot Retention Time
Core businesses	Use scheduled snapshots and set the policy to once a day.	7 days to 30 days
Non-core and non-data businesses	Use scheduled snapshots and set the policy to once a week.	7 days
Archiving businesses	Manually create snapshots based on actual business needs, without the need to set a fixed frequency.	One month to several months
Testing businesses	Manually create snapshots based on actual business needs, without the need to set a fixed frequency.	Promptly deleting after use

Operation Steps

Creating a Scheduled Snapshot Policy

1. Log in to the file system console.
2. In the left navigation tree, choose Snapshot > Scheduled Snapshot Policy.
3. On the Scheduled Snapshot Policy page, click Create.
4. Enter a snapshot name, select the backup date and time point, set the snapshot retention period, and click OK.

Associating with a File System

1. In the scheduled snapshot policy list, click Associate File System on the right side of the policy.
2. Select the file system to be associated and click OK.

Enabling/Disabling a Scheduled Snapshot Policy

In the scheduled snapshot policy list, toggle the switch for scheduled snapshots on the right of the policy to enable or disable the scheduled snapshot policy.

Modifying a Scheduled Snapshot Policy

1. In the scheduled snapshot policy list, click Modify Policy on the right side of the policy.
2. On the Modify Snapshot Policy page, modify the parameters, and click OK.

Deleting a Scheduled Snapshot Policy

1. In the scheduled snapshot policy list, click Delete on the right of the policy.
2. In the pop-up snapshot policy deletion dialog box, check whether the policy is associated with a file system, and click OK after confirmation.

> Note:
>
> Snapshots created by a snapshot policy will be retained and will not be synchronously deleted after the policy itself is deleted.

Creating a File System Using a Snapshot

Scenarios

A file system created using a snapshot contains all the data in the snapshot, and the used capacity of the file system is consistent with the data of the source file system at the current snapshot creation time. Snapshot creation enables cross-AZ replication and sharing of file systems.
For a file system created using a snapshot, a file service protocol can be freely selected, which is not limited by the type of source file system protocol, and the network can also be reselected.

Operation Steps

1. Log in to the CFS console.
2. On the Snapshot List page, click Create File System in the Action column on the right side of the target snapshot.
3. In the pop-up dialog box, set the following parameters:

Field	Description
Name	Enter a custom name for the file system.
AZ	Select the AZ for which you want to create a CFS file system.
Project	Optional. When a file system is created, its associated project can be set.
Storage Class	Select the desired storage type, Standard Storage or High-Performance Storage.
File Service Protocol	Select the protocol type of the file system, NFS or SMB. The NFS protocol is more suitable for Linux/Unix clients, and the CIFS/SMB protocol is more suitable for Windows clients.
Snapshot	Optional. When creating a file system using a snapshot, you should select Use Snapshot to Create File System and select the snapshot you need to use. If you choose to create a file system using a snapshot, the initial data volume in the file system will be consistent with the snapshot size.
Network Type	- VPC. To achieve file system sharing by CVM under a VPC, you need to select a VPC when creating a file system. When the file system uses a VPC, only CVM instances within the same VPC can be mounted if no special network settings are made. - If you have multi-network file system sharing needs, see Cross-Network Access to File Systems.
Permission Group	Each file system should be bound to a permission group, which defines a group of allowlists with access permissions and read and write operation permissions.

Field	Description
Tag	- If you already have a tag, you can add the tag for the new file system here. - If you do not have a tag yet, go to the tag console to create a desired tag and bind it to the file system. You can also add a tag to the file system after the file system is created.

4. Click OK.

   You can view the created file system on the File System List page. See Quick Start to mount the file system to a client in the same AZ.

Deleting Snapshots

Overview

When a snapshot is no longer needed, you can delete it to release virtual resources.

Note

- When a snapshot is deleted, only the data exclusive to that snapshot will be deleted without any impact on the file system where it was created.
- All snapshots have been incremented to retain data, and deleting one of the snapshots will not affect other inventory snapshots.
- When a snapshot is deleted, all data in the snapshot will be deleted, and deleted data is irrecoverable. As the deleted snapshot is irrecoverable, act with caution.

Directions

1. Log in to the CFS console.
2. On the snapshot list page, click Delete on the right side of the target snapshot.
3. In the pop-up dialog box, click OK.

Using Tags

Overview

Tag is a marker used to mark cloud resources, and is a key-value pair. Tags can help you conveniently manage cloud resources from multiple dimensions (such as business, purpose, and owner). Note that we will not use the tags you set, and tags are only used for your management of resources. This document describes how to edit tags for CFS resources.

Use Limits

Note the following limits when editing tags:

- Quantity limit: The maximum number of tags allowed by each file system is 50.
- Tag key limit: Only digits , letters , and +=.@- are allowed, and the length of the tag key cannot exceed 255 characters.
- Tag value limit: Only empty strings, digits , letters , and +=.@- are allowed, and the length of the tag value cannot exceed 127 characters.

Operating Methods and Use Cases

Use Case

A company purchased 6 file systems. The information regarding the business group, scope, and owners for these 6 file systems is as follows:

File System ID	Business Group	Business Scope	Owner
cfs-abcdef1	E-commerce	Marketing campaigns	Tom
cfs-abcdef2	E-commerce	Marketing campaigns	Harry
cfs-abcdef3	Gaming	Game A	Jerry
cfs-abcdef4	Gaming	Game B	Harry
cfs-abcdef5	Entertainment	Post-production	Harry
cfs-abcdef6	Entertainment	Post-production	Tom

Taking cfs-abcdef1 as an example, we can add the following three sets of tags to this file system:

Tag Key	Tag Value
dept	ecommerce
business	mkt
owner	zhangsan

Similarly, we can add the corresponding tags to other file systems based on the business group, scope, and owners.

Operation Steps

Tagging a New File System

Prerequisites

Confirm that you have the required tags before creating a file system. Otherwise, you should create them in the Tag console.

1. Log in to the CFS console.
2. Click Create on the File System List page.
3. In the pop-up Create File System window, find Tag below and click Add to add a tag to this file system. Only existing tags can be added in this step.
4. Click OK, and the file system will be bound to the corresponding tags after it is successfully created.

> Note:
>
> For the description of configuration items for creating a file system, see the document titled Creating File Systems and Mount Targets.

Adding, Modifying, or Deleting a Tag on an Existing File System

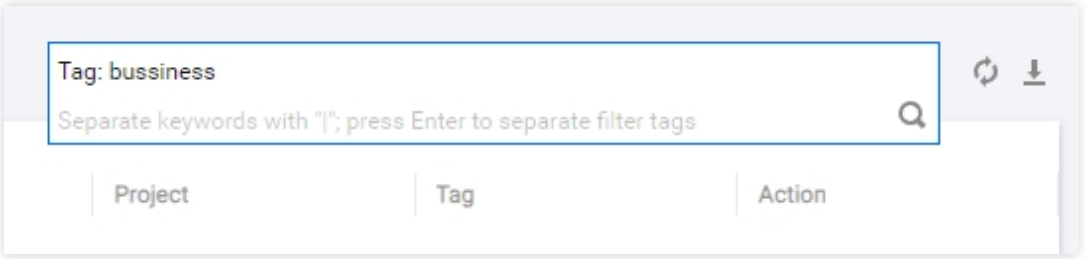
1. On the File System List page, locate the file system and click Edit Tag in the operation column.
2. Add, modify, or delete tags based on actual needs in the pop-up You have selected 1 resource window.
3. Click OK to complete the addition, modification, or deletion of tags.

Filtering File Systems by Tag

You can filter file systems with a certain type of tag through the following operations.

1. Select Tag in the search box.

2. Enter a tag key and tag value after Tag:, and click or press the Enter key to search. As shown in the figure below:
   For example, if you want to filter out file systems with the tag key being business, you can enter Tag:business ; if you want to filter out file systems with the tag key being business and the tag value being mkt, you can enter Tag:business:mkt .



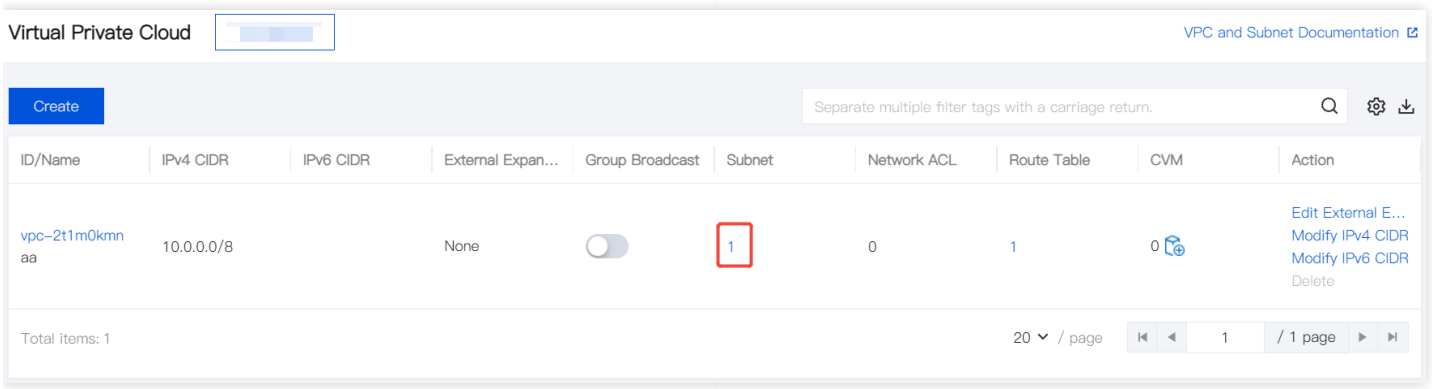
Guide for Cross-AZ and Cross-Network Access

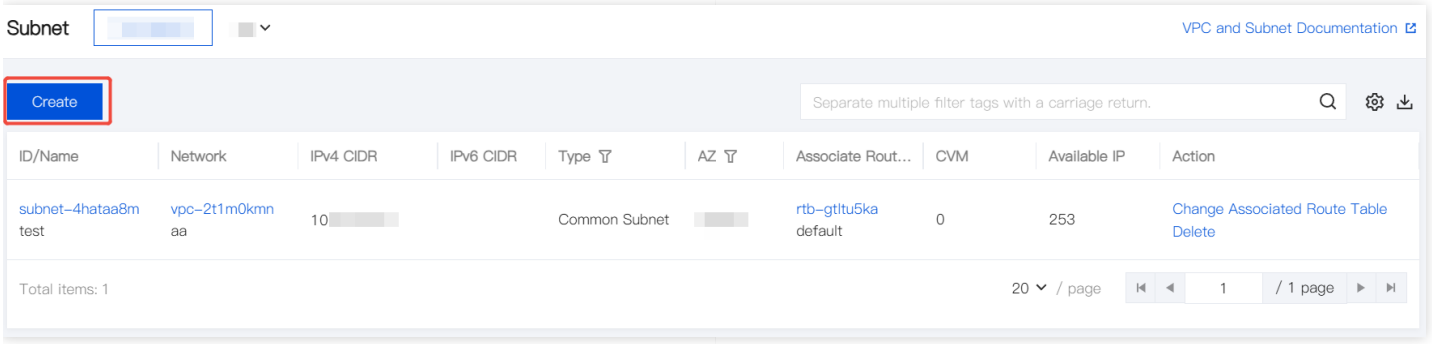
Cross-AZ Access in a VPC

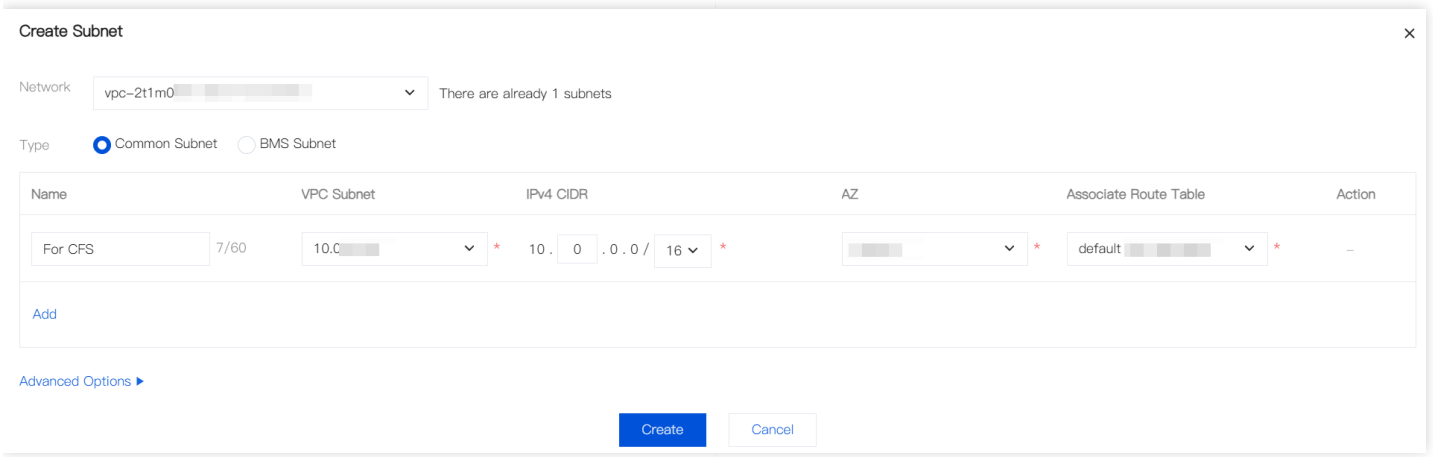
If you need to have a CFS instance shared by multiple CVM instances in different AZs in the same region, configure the CVM and CFS instances into the same VPC to achieve cross-AZ access to resources.

Assume that there is a CVM instance in City Zone 1 that needs to use CFS, but file systems cannot be directly created in City Zone 1 as the resources are sold out there.

If the CVM is in the City1 Zone subnet of the VPC, you can log in to the VPC console to create a subnet with the availability zone being City2 Zone for the VPC.







After the subnet is successfully created, go back to the CFS console and select this VPC and the subnet created to create resources in City Zone 2. In this case, the CFS file system can be directly mounted to the CVM instance under the subnet of City Zone 1 in this VPC. For more information, see File System Mounting Assistance.

Cross-VPC and Cross-Region Access

CFS supports the following scenarios for resource access.

- A CFS instance needs to be shared by multiple CVM instances distributed in different VPCs.
- The CVM and CFS instances are in different VPCs.
- The CVM and CFS instances are in different regions (to achieve better access performance, it is recommended that the CVM and CFS instances be in the same region).

You can use a "peering connection" to achieve resource access across the CVM instances distributed in VPC-A/VPC-B and CFS instances distributed in VPC-C. For more information, see Peering Connection.

Automatically Mounting File Systems

Automatically Mounting NFS File Systems on Linux

Method 1: Using CFS Client Assistant for Setting (Recommended)

> Note:
>
> If you have used fstab or other methods, to avoid conflicts, it is recommended to disable the existing method for restarting automatic mount before using this method.

1. Log in to the CFS console.
2. In the left navigation tree, select File System List to go to the File System List page.
3. On the right side of the file system to be mounted, click Action > Mount.
4. In the pop-up window, view the automatic mounting configuration status of the selected file systems on the CVM (only supported on CVMs with Automation Tools installed). Select the instances to be mounted and click Next step.

Status	Description
Configured	On the CVM, the automatic mounting of selected file systems upon server restart has been configured by using the console or the client assistant.
Not configured	On the CVM, the automatic mounting of selected file systems upon restart has not been configured by using the console or the client assistant.
Not supported	For Windows clients, automatic mounting of file systems upon server restart by using the console or client assistant is not supported currently.

> Note:
>
> This query method only applies to the automatic mounting implemented by using the client assistant and cannot detect the automatic mounting configured through fstab or other methods.

5. Fill in the target local directory and other parameters on this page, select Restart Automatic Mount, and click Next.
6. When the execution status is Completed, the mounting is successful, and automatic mounting upon

restart has been configured.
If you have mounted the selected file system in this directory, the execution status will be Failed, because repeatedly mounting the same file system in the same directory is not supported. However, you can hover the mouse pointer over the execution status. When the execution details display the following content, the restart automatic mounting feature has been configured and enabled.

> Note:
>
> - The restart automatic mounting feature setting does not affect running CVMs and only takes effect when the server is restarted.
> - This process concurrently attempts to mount the specified file systems within 5 minutes after the server is restarted, and exits upon timeout without affecting the normal operation of the Linux system.
> - When a mounting failure occurs, you can view the automatic mounting logs saved in /usr/local/reboot_mount_cfs/logs to quickly locate the issue.
> - To disable automatic mounting, modify the reboot_mount_cfs.sh file in /usr/local/reboot_mount_cfs/bin . Change "true" in the conditional expression to "false" and save the changes to disable automatic mounting upon restart.

```
vim /usr/local/reboot_mount_cfs/bin/reboot_mount_cfs.sh

# ... The above content is omitted...
  if false; then  # Change "true" in the conditional expression to "false".
      vDoMount="sudo mount.lustre x.x.x.x@tcp:/xxxxxxxx/cfs /path/to/mountpoint_x"  # Comm
and to disable the restart automatic mounting feature for the specified file systems
# ... The following content is omitted...
```

Method 2: Using Systemd for Configuration

> Note:
>
> Essentially, both Method 2 and Method 1 implement file system mounting based on systemd. The difference is that Method 1 implements configuration automatically, while Method 2 implements configuration manually.

1. Create a folder in /usr/local/ to store execution scripts and running logs.

> Note:
>
> To avoid failure in querying the status in the console, it is recommended not to change the file name.

```
cd /usr/local/
mkdir -p reboot_mount_cfs/logs
mkdir -p reboot_mount_cfs/bin
```

2. Create the reboot_mount_cfs.sh script in reboot_mount_cfs/bin , modify the mount command where prompted as needed, and grant executable permissions.

```
cd /usr/local/reboot_mount_cfs/bin
file="reboot_mount_cfs.sh"
if [ -e "$file" ]; then
echo "File already exists, skipping creation."
else
cat > "$file"  << 'EOF'
#!/bin/bash
####Logs during file system mounting in the startup process####
readonly LOG_FILE=/usr/local/reboot_mount_cfs/logs/reboot_mount_cfs-$(date +%Y%m%d
_%H%M%S).log

function main() {
    local vDryRun=false
    local vDoMount
    ####Usage Instructions####
    # 0) Mount a file system.
    # 1) Change the true in the conditional expression to false to disable automatic mounting upo
n server restart.
    # 2) Replace vDoMount with the actual mount command for the file system.
    ####Usage Instructions####
    if true; then
        vDoMount="mount.lustre x.x.x.x@tcp:/xxxxxxxx/cfs /path/to/mountpoint_x"
        if ${vDryRun}; then
            echo "${vDoMount}"
        else
            echo "[$(date +%Y-%m-%d\ %H:%M:%S)]Try to run: ${vDoMount}" >> "${LOG_FILE}"
            local vTimesRetried=0
            while [[ ${vTimesRetried} -le ${MAX_MOUNTS2RETRY} ]]; do
                timeout 300 ${vDoMount}&
                pid=$!
                wait $pid
                vRc=$?
                if [[ ${vRc} -ne 0 ]]; then
                    echo "[$(date +%Y-%m-%d\ %H:%M:%S)][${vRc}]Failed to run: ${vDoMount}" >>
"${LOG_FILE}"
                    sleep 120 # Delay for 2 minutes before retry to avoid network slow start.
                    vTimesRetried=$((vTimesRetried + 1))
                    continue
```

```
            else
                echo "[$(date +%Y-%m-%d\ %H:%M:%S)]Succeed to run: ${vDoMount}" >> "${LOG
_FILE}"
                break
            fi
        done
    fi
  fi

  echo "====All mountings tried====" >> "${LOG_FILE}"  # This is a positioning line, do not d
elete or modify it.
  echo >> "${LOG_FILE}"
  echo "Current mountpoints of CFS:" >> "${LOG_FILE}"
  mount -t nfs >> "${LOG_FILE}"
  mount -t nfs4 >> "${LOG_FILE}"
  mount -t lustre >> "${LOG_FILE}"
  wait
  ####Tool Usage Constraints###
  # Only for assisting the systemd service in completing the initial mounting during system start
up.
  # Do not use it in non-startup scenarios.
  ###Tool Usage Constraints###
}
main
EOF
fi
chmod a+x reboot_mount_cfs.sh
```

3. Create reboot_mount_cfs.service in /etc/systemd/system .

```
cd /etc/systemd/system
echo '[Unit]
Description=Used to automatically mount Turbo filesystems.
After=default.target
[Service]
ExecStart=/usr/local/reboot_mount_cfs/bin/reboot_mount_cfs.sh
[Install]
WantedBy=default.target
' > reboot_mount_cfs.service
```

Set the reboot_mount_cfs.service service to start automatically at boot. When the system starts, the mount command with the status of True in reboot_mount_cfs.sh will be executed automatically.

```
systemctl daemon-reload && systemctl enable reboot_mount_cfs.service
```

Method 3: Using fstab for Configuration

> Note:
>
> - After the automatic mounting is configured by using this method, if a file system cannot be mounted properly (for example, it has been deleted), the Linux system cannot start normally. In this case, enter the rescue mode during system startup, delete the automatic mount command in fstab, and then restart the host.
> - Method 1 or Method 2 is recommended to avoid the above situation.

1. First, connect to the CVM instance that needs to automatically mount the file system by logging in through the CVM console or by remote login. Open the `/etc/fstab` file (ensure the account you logged in has root permissions).

   ```
   //Use the following command to open the fstab file.
   vi /etc/fstab
   ```

2. After opening the fstab file, enter i (insert), and add the following command line in `/etc/fstab` . The following mounting methods are supported:

   Using NFS v4.0 to mount file systems
   <Mount target IP address>:/ <target mount directory> nfs vers=4,minorversion=0,hard,timeo=600,retrans=2,_netdev,noresvport 0 0
   For example: 10.10.19.12:/ /local/test nfs vers=4,minorversion=0,hard,timeo=600,retrans=2,_netdev,noresvport 0 0

   Using NFS v3.0 to mount file systems
   <Mount target IP address>:/<fsid> <target mount directory> nfs vers=3,nolock,proto=tcp,hard,timeo=600,retrans=2,_netdev,noresvport 0 0
   For example: 10.10.19.12:/djoajeo4 /local/test nfs vers=3,nolock,proto=tcp,hard,timeo=600,retrans=2,_netdev,noresvport 0 0

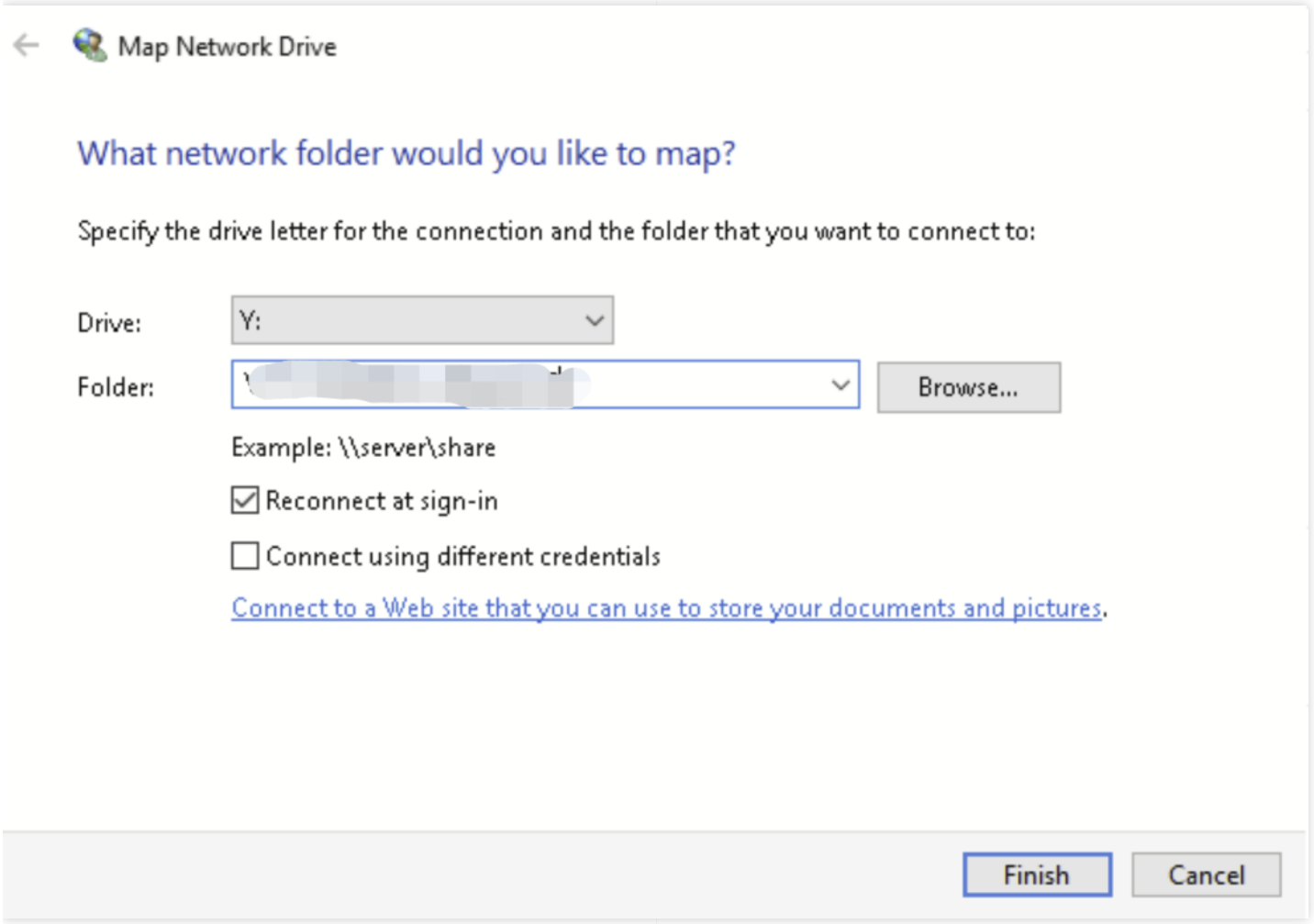
   Mounting with Turbo
   <Mount target IP address>@tcp0:/<fsid>/cfs <target mount directory> lustre defaults,_netdev 0 0
   Example: 172.16.0.7@tcp0:/01184207/cfs /root/turbo lustre defaults,_netdev 0 0

3. Press the Esc key on the keyboard and enter ":wq" to save the above modifications. After restarting

the client, you can see that the file system is automatically mounted.

Automatically Mounting File Systems on Windows

When mounting a file system, select Reconnect at sign-in, as shown in the following figure. For more help information on mounting, see Using CFS File Systems on Windows Clients.



Best Practices
Using CFS on Containers

Overview

Kubernetes Engine (TKE) is a highly scalable high-performance container management service that provides multiple application release methods and continuous delivery capabilities, supports the micro-service architecture, solves environmental problems in user development, testing and Ops processes, and helps users lower the cost and improve efficiency.

For businesses using containers, such as business application deployment, DevOps, machine learning, auto-scaling and other scenes, there are usually a large number of configuration files, model files, log data, document attachments, etc. that need to be shared and accessed by multiple containers. Especially in the scenes of machine learning, intelligent recommendation and log data processing, in addition to basic data sharing, it is required that shared storage can provide services with high concurrent access, high throughput, high IOPS and low latency. Cloud File Storage (CFS) can provide the above-mentioned shared storage characteristics just by easy configuration and mounting on containers, which is especially suitable for use with container businesses. This document will introduce how to use CFS on TKE.

Prerequisites

The premise of this guide is that you have already created a container cluster. If you have not created the TKE, see Deploying Container Service Operation Guide to create a container first.

Apply for CFS Resources and Obtain A Mount Point IP

- If you do not have a file system yet, please follow the Creating CFS File System guide to create a file system. When it is created, please note that the VPC network should be selected under the same VPC as your container host to ensure network intercommunication.
- If you already have a file system under the same VPC as the TKE, you can go to the File System Details page for the mount point IP.

Configure A CFS File Mounting System

Step 1: Start the NFS Client on Node

Before mounting, make sure that nfs-utils or nfs-common has been installed in the system. The installation method is as follows:

- CentOS

  sudo yum install nfs-utils

- Ubuntu

  sudo apt-get install nfs-common

Step 2: Create a PV

Create a PesistentVolume of type CFS by executing the following command.

> Note:
>
> - nfs.server: It is the mount point IP of the CFS file system that has been obtained above. In this example, it is assumed that the file system IP is 10.0.1.41.
> - nfs.path: It is a root directory or subdirectory of the CFS file system. This case takes the root directory as an example.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: cfs
spec:
  capacity:
    storage: 10Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  mountOptions:
    - hard
    - nfsvers=4
  nfs:
    path: /
    server: 10.0.1.41
```

Step 3: Create a PVC

Next, create a PersistentVolumeClaim to request the binding of the created PersistentVolume.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: cfsclaim
spec:
  accessModes:
    - ReadWriteMany
  volumeMode: Filesystem
  storageClassName: ""
  resources:
    requests:
      storage: 10Gi
```

Step 4: Create a Pod

Create a Pod application to declare that the data volume is mounted and used.

```
kind: Pod
apiVersion: v1
metadata:
  name: mypod
spec:
  containers:
    - name: myfrontend
      image: nginx
      volumeMounts:
      - mountPath: "/var/www/html"
        name: mypd
  volumes:
    - name: mypd
      persistentVolumeClaim:
        claimName: cfsclaim
```

After completing the above steps, you can use this file system in the created Pod.

API Documentation
NAS(cfs)
version(2019-07-19)
API overview

API version

V3

APIs

Interface name	Interface function
BindAutoSnapshotPolicy	BindAutoSnapshotPolicy
DescribeAutoSnapshotPolicies	DescribeAutoSnapshotPolicies
DescribeCfsTags	DescribeCfsTags
UpdateCfsDrStatus	UpdateCfsDrStatus

File system APIs

Interface name	Interface function
CreateCfsFileSystem	CreateCfsFileSystem
DeleteCfsFileSystem	DeleteCfsFileSystem
DescribeCfsFileSystemClients	DescribeCfsFileSystemClients
DescribeCfsFileSystems	DescribeCfsFileSystems
DescribeCfsFileSystemsByVpc	DescribeCfsFileSystemsByVpc
DescribeMountTargets	DescribeMountTargets
UpdateAutoSnapshotPolicy	UpdateAutoSnapshotPolicy

Interface name	Interface function
UpdateCfsFileSystemName	UpdateCfsFileSystemName
UpdateCfsFileSystemPGroup	UpdateCfsFileSystemPGroup
UpdateCfsFileSystemSizeLimit	UpdateCfsFileSystemSizeLimit

Other APIs

Interface name	Interface function
CopyFileSystemSnapshotCrossRegion	CopyFileSystemSnapshotCrossRegion
CreateAutoSnapshotPolicy	CreateAutoSnapshotPolicy
DeleteAutoSnapshotPolicy	DeleteAutoSnapshotPolicy
UnbindAutoSnapshotPolicy	UnbindAutoSnapshotPolicy

Permission Group APIs

Interface name	Interface function
BatchAddCfsRules	BatchAddCfsRules
BatchDeleteCfsRule	BatchDeleteCfsRule
CreateCfsPGroup	CreateCfsPGroup
CreateCfsRule	CreateCfsRule
DeleteCfsPGroup	DeleteCfsPGroup
DeleteCfsRule	DeleteCfsRule
DescribeCfsPGroups	DescribeCfsPGroups
DescribeCfsRules	DescribeCfsRules
UpdateCfsPGroup	UpdateCfsPGroup
UpdateCfsRule	UpdateCfsRule

Service APIs

Interface name	Interface function
DescribeAvailableZoneInfo	DescribeAvailableZoneInfo
DescribeCfsServiceStatus	DescribeCfsServiceStatus
SignUpCfsService	SignUpCfsService

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.

APIs
BindAutoSnapshotPolicy

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

Snapshot policy bound to the file system

Default interface request rate limit: 20 times per second.

Interface update time: 2021-08-26 02:37:04.

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: BindAutoSnapshotPolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
AutoSnapshotPolicyId	Yes	No	String	Snapshot Policy ID Example: asp-xcweqdx
FileSystemIds	Yes	No	String	File System ID Example: cfs-yfleili

3. Output Parameters

Parameter name	Type	Description
AutoSnapshotPolicyId	String	Snapshot Policy ID Example: asp-xcweqdx
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
InvalidParameterValue.AutoPolicyNotFound	The snapshot policy does not exist.

DescribeAutoSnapshotPolicies

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

Queries the scheduled snapshot policy list for the file system

Default interface request rate limit: 20 times per second.

Interface update time: 2022-04-01 04:39:47.

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: DescribeAutoSnapshotPolicies
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
AutoSnapshotPolicyId	No	No	String	Snapshot Policy ID Example: asp-xcweqdx
Offset	No	No	Uint64	Paging Example: 2
Limit	No	No	Uint64	Page size Example: 2
Filters	No	No	Array of Filter	Filter Criteria Example:View
Order	No	No	String	Sort rule: ASC, DESC Example: ASC,DESC
OrderField	No	No	String	Sort field: CreationTime, FileSystemNums, TotalSnapshotSize, SizeByte Example: CreationTime

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
TotalCount	Uint64	Total data volume Example: 3
AutoSnapshotPolicies	Array of AutoSnapshotPolicyInfo	Snapshot policy 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

There is no error code related to business logic for this interface, For other error codes, see public ErrorCode.

DescribeCfsTags

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

Queries the CFS resource pool list

Default interface request rate limit: 20 times per second.

Interface update time: 2020-06-11 21:41:20.

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: DescribeCfsTags
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
InternationalFlag	No	No	Int64	Internationalization Example: 1
ZoneId	Yes	No	Uint64	Availability zone ID Example: 80520001

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
Tags	Array of CfsTagUser	Tag 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
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
InternalError	Internal error
FailedOperation	Operation failed

UpdateCfsDrStatus

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API is used to modify the status of the disaster recovery file system.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-10-23 10:18:55.

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: UpdateCfsDrStatus
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
FsId	Yes	No	String	File System ID Example: cfs-s2xcx23q
DrcType	Yes	No	String	drc_source

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
FsId	String	File System ID Example: cfs-xwec22sa
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
InvalidParameterValue.UnavailableZone	This region cannot provide service.
FailedOperation	Operation failed
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
InvalidParameter	Wrong parameter.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.
InvalidParameterValue.InvalidFsStatus	Invalid file system status.
ResourceNotFound.FsNotExist	File system does not exist.
ResourceNotFound.FileSystemNotFound	The file system does not exist.
InternalError.Timeout	Timed out.
InternalError	Internal error
InvalidParameterValue	Invalid parameter value
InvalidParameterValue.InvalidProtocol	Protocol error.

File system APIs
CreateCfsFileSystem

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

Adds file systems

Default interface request rate limit: 10 times per second.

Interface update time: 2020-12-17 07:39:57.

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: CreateCfsFileSystem
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
ZoneId	No	No	Uint64	Availability zone ID Example: 800001
Zone	Yes	No	String	AZ name. For example, ap-zone-1. Example: ap-zone-1
CreationToken	No	No	String	User-defined file system name, with a lower priority than FSNAME Example: fs-name-business
Protocol	No	No	String	File system protocol type. Values: NFS, CIFS. If not specified, this field will be NFS by default. Example: NFS
StorageType	No	No	String	File system storage type. Value: SD (standard type) Example: SD
NetInterface	Yes	No	String	Network type. VPC: Virtual Private Cloud; BASIC: Basic network Example: VPC
PGroupId	Yes	No	String	Permission group ID Example: pgroupbasic
VpcId	No	No	String	Virtual Private Cloud (VPC) ID. Either this field or UnVpcId must be specified when the network type is VPC. Example: vpc-ad1dd
UnVpcId	No	No	String	Unified VPC ID assigned by the system Example: vpc-cxasdad
SubnetId	No	No	String	Subnet. Either this field or UnSubnetId must be specified when the network type is VPC. Example: sbnet-id-business
UnSubnetId	No	No	String	Unified subnet ID assigned by the system Example: unsbnet-id-business
MountIP	No	No	String	Designated IP address, only supported in VPC. If this field is not specified, an IP address within the subnet will be randomly assigned. Example: 10.0.0.1
StorageResourcePkgId	No	No	String	Storage package bound to the file system. Only one storage package can be bound to each file system. Example: no support current
BandwidthResourcePkgId	No	No	String	Bandwidth package bound to the file system. Only one bandwidth package can be bound to each file system. Example: no support current
FsName	No	No	String	User-defined file system name. Either this field or CreationToken must be specified. Example: fs-name-business
Encrypted	No	No	Bool	Indicates whether the file system is encrypted. If not specified, this field will be no by default. Example: False
KmsKeyId	No	No	String	Encryption key ID Example: 23e80852-1e38-11e9-b129-5cb9019b4b02
TagId	No	No	Uint64	CFS resource pool ID Example: 8
SnapId	No	No	String	Snapshot ID Example: d51a195248-ab41-13be-c0e7-9924d5c4c6
ResourceTags	No	No	Array of ResourceTags	Tag Example:View
ProjectId	No	No	String	Project ID Example: pr-aea4e215

Parameter name	Required	Allow NULL	Type	Description
BackupSize	No	No	Uint64	File system size during backup Example: 100000
BackupId	No	No	String	Backup ID Example: 234

3. Output Parameters

Parameter name	Type	Description
CreationTime	String	File system creation time Example: 2017-02-24 11:28:12
CreationToken	String	User-defined file system name Example: fs-name-business
FileSystemId	String	File system ID Example: cfs-xcweqdx
LifeCycleState	String	File system status Example: AVAILABLE
SizeByte	Uint64	File system capacity that is used Example: 1000
ZoneId	Uint64	Availability zone ID Example: 800001
FsName	String	User-defined file system name Example: fs-name-business
Encrypted	Bool	Indicates whether the file system is encrypted Example: False
CfsId	String	File System ID Example: iriaio0h
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
InvalidParameterValue.FsNameLimitExceeded	User-defined name too long (exceeds 64 bytes).
InvalidParameterValue.InvalidVpcParameter	VPC parameter error.
InvalidParameterValue.MissingSubnetidOrUnsubnetid	SUBNETID and UNSUBNETID cannot both be empty.
InternalError.CreateFsFailed	Failed to create the file system.
ResourceInsufficient.RegionSoldOut	Region resources sold out.
InvalidParameterValue.InvalidEncrypted	Encryption parameter error.
InvalidParameterValue.InvalidFsName	Invalid user-defined name.
InvalidParameterValue.InvalidNetInterface	Invalid network type.
InvalidParameterValue.InvalidZoneOrZoneId	Invalid availability zone or availability zone ID.
InvalidParameterValue.MissingStorageResourcePkg	No storage package is bound.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
InvalidParameterValue.InvalidZoneId	Invalid availability zone.
InvalidParameterValue.MissingZoneId	ZoneID missing.
ResourceInsufficient.FileSystemLimitExceeded	The number of file systems has reached the limit.
ResourceNotFound.PgroupNotFound	Permission group does not exist.
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
InvalidParameterValue.UnavailableRegion	This availability zone cannot provide service.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.
InvalidParameterValue.InvalidStorageType	Storage type error.
InvalidParameterValue.MissingKmsKeyId	Key ID or ARN missing.
InvalidParameterValue.InvalidRegionZoneInfo	Region error. Either (ZoneName) or (ZoneId, Region) must be selected.
InvalidParameterValue.InvalidVip	The specified VIP is unavailable.
InvalidParameterValue.ZoneIdRegionNotMatch	ZoneId and Region do not match.

Error Code	Description
InternalError	Internal error
InvalidParameterValue.InvalidPgroupId	Invalid permission group ID.
InvalidParameterValue.MissingVpcidOrUnvpcid	VPCID and UNVPCID cannot both be empty.
AuthFailure.UnauthorizedOperation	Request lacks CAM authorization.
FailedOperation	Operation failed
InvalidParameterValue.MissingVpcParameter	VPC parameter missing.
InvalidParameterValue.MissingZoneOrZoneId	Region error. Either (Zone) or (Zone_id) must be selected.
InvalidParameterValue.InvalidMountTargetIp	Invalid mount target IP.
InvalidParameterValue.UnavailableZone	This region cannot provide service.
InvalidParameter	Wrong parameter.
InvalidParameterValue.InvalidSubnetId	Invalid subnet ID.
InvalidParameterValue.InvalidProtocol	Protocol error.
ResourceInsufficient.SubnetIpAllOccupied	No available IP under this subnet.
InvalidParameterValue.InvalidVpcId	Invalid VPCID.
InvalidParameterValue.SnapSrcFsBusy	The original FS of the snapshot is busy.
InternalError.BillingJipingInterfaceQuotaFail	The measurement quota is insufficient.
InvalidParameter.InvalidTagKey	Parameter value error: Tag key cannot be empty.
ResourceInsufficient.TagLimitExceeded	Insufficient resources: The number of tags for this resource has reached the limit.
AuthFailure	CAM signature/authentication error

DeleteCfsFileSystem

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

Deletes file systems

Default interface request rate limit: 20 times per second.

Interface update time: 2020-12-17 07:45:17.

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: DeleteCfsFileSystem
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
FileSystemId	Yes	No	String	File system ID Example: cfs-2sczq3das
Force	No	No	Int64	Forced deletion of file system: 1 for forced, 0 or empty for default Example: 1

3. Output Parameters

Parameter name	Type	Description
componentName	String	Component name Example: billing
data	Array of String	Returned data. Example: success
eventId	Uint64	Event ID Example: 1097864659

Parameter name	Type	Description
returnCode	Uint64	Returned values Example: 0
returnValue	Uint64	Returned values Example: 0
returnMessage	String	Return Message Example: ok. OK
returnMsg	String	Return Message Example: ok. OK
timestamp	Uint64	Timestamp Example: 1741786693
version	Uint64	API version 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
InvalidParameterValue.InvalidFileSystemId	Invalid FileSystemId.
InvalidParameterValue.InvalidFsStatus	Invalid file system status.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
FailedOperation.PgroupInUse	Permission group already bound to the file system.
ResourceNotFound.FileSystemNotFound	The file system does not exist.
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.
FailedOperation.PgroupIsUpdating	The permission group is being updated.
InvalidParameterValue.InvalidRegionZoneInfo	Region error. Either (ZoneName) or (ZoneId, Region) must be selected.
FailedOperation.MountTargetExists	Mount target exists in the file system.
InternalError	Internal error

Error Code	Description
AuthFailure.UnauthorizedOperation	Request lacks CAM authorization.
InvalidParameter	Wrong parameter.
UnsupportedOperation	Operation unsupported.
InternalError.Timeout	Timed out.
InvalidParameterValue.MissingFsParameter	FileSystem missing.
FailedOperation.UntagResourceFailed	Operation failure: Failed to unbind the resource tag.
FailedOperation.BindResourcePkgFailed	Failed to bind the resource package to the file system.
FailedOperation.ClientTokenInUse	Operation failure: Resources are being created.
InvalidParameter.InvalidTagKey	Parameter value error: Tag key cannot be empty.
AuthFailure	CAM signature/authentication error

DescribeCfsFileSystemClients

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

Queries clients mounted to the file system. This feature requires the client to install the CFS monitoring plugin.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-10-16 06:28:54.

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: DescribeCfsFileSystemClients
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
FileSystemId	Yes	No	String	File system ID Example: cfs-2sczq3das

3. Output Parameters

Parameter name	Type	Description
ClientList	Array of	Client List

Parameter name	Type	Description
	FileSystemClient	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	Internal error
InvalidParameterValue.InvalidFileSystemId	Invalid FileSystemId.
InvalidParameterValue.MissingFileSystemId	FileSystemId missing.
ResourceNotFound.FileSystemNotFound	The file system does not exist.
ResourceNotFound.MountTargetNotFound	Mount target does not exist.
UnsupportedOperation	Operation unsupported.

DescribeCfsFileSystems

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (DescribeCfsFileSystems) is used to query file systems.

Default interface request rate limit: 20 times per second.

Interface update time: 2020-12-17 07:43:42.

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: DescribeCfsFileSystems
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
FileSystemId	No	No	String	File system ID Example: cfs-2sczq3das
InternationalFlag	No	No	Uint64	When the value of this field is 1, unnamed file systems will use English names. Example: 1
VpcId	No	No	String	Virtual Private Cloud (VPC) ID Example: vpc-sxcase4
SubnetId	No	No	String	Subnet ID Example: sbnet-id-business
Offset	No	No	Uint64	Offset Example: 10
Limit	No	No	Uint64	Limit Example: 10
CreationToken	No	No	String	User-defined name Example: fs-name-business
FileSystemIds	No	No	Array of String	File system ID list Example: cfs-sdcxdads,cfs-21esda32
LifeCycleState	No	No	String	File system status Example: creating

Parameter name	Required	Allow NULL	Type	Description
ZoneId	No	No	Uint64	Availability zone ID Example: 50010001

3. Output Parameters

Parameter name	Type	Description
FileSystems	Array of FileSystemInfo	File System Name Example:View
TotalCount	Uint64	Total Number of File Systems 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
InvalidParameterValue.InvalidVpcParameter	VPC parameter error.
InvalidParameterValue.InvalidFileSystemId	Invalid FileSystemId.
ResourceInsufficient.RegionSoldOut	Region resources sold out.
InvalidParameterValue.InvalidZoneOrZoneId	Invalid availability zone or availability zone ID.
InvalidParameterValue.MissingFileSystemId	FileSystemId missing.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
InvalidParameterValue.InvalidZoneId	Invalid availability zone.
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
InvalidParameterValue.UnavailableRegion	This availability zone cannot provide service.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.
InvalidParameterValue	Invalid parameter value
InvalidParameterValue.InvalidStorageType	Storage type error.
InvalidParameterValue.InvalidVip	The specified VIP is unavailable.
InvalidParameterValue.ZoneIdRegionNotMatch	ZoneId and Region do not match.
InternalError	Internal error
AuthFailure.UnauthorizedOperation	Request lacks CAM authorization.
FailedOperation	Operation failed

Error Code	Description
MissingParameter	Missing parameter error.
InvalidParameter	Wrong parameter.
InvalidParameterValue.InvalidPerMode	Invalid storage type.
InvalidParameterValue.MissingAppidOrUin	APPID and UIN cannot both be empty.
InvalidParameterValue.InvalidSubnetId	Invalid subnet ID.
InvalidParameterValue.InvalidVpcId	Invalid VPCID.
InvalidParameterValue.MissingFsParameter	FileSystem missing.
ResourceNotFound.SnapNotExit	Snapshot does not exist.
InvalidParameterValue.InvalidSnapStatus	Invalid snapshot status
InvalidParameterValue.SnapSrcFsBusy	The original FS of the snapshot is busy.
FailedOperation.UntagResourceFailed	Operation failure: Failed to unbind the resource tag.
FailedOperation.BindResourcePkgFailed	Failed to bind the resource package to the file system.
FailedOperation.ClientTokenInUse	Operation failure: Resources are being created.
InvalidParameter.InvalidTagKey	Parameter value error: Tag key cannot be empty.
InvalidParameter.ClientTokenLimitExceeded	Parameter value error: The string for ensuring request idempotency exceeds the length limit (64 bytes).
InvalidParameter.InvalidClientToken	Parameter value error: The string for ensuring request idempotency is incorrect.
AuthFailure	CAM signature/authentication error

DescribeCfsFileSystemsByVpc

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

Queries file system instances by using VPC information

Default interface request rate limit: 20 times per second.

Interface update time: 2022-04-01 05:02:31.

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: DescribeCfsFileSystemsByVpc
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
VpcId	Yes	No	String	Virtual Private Cloud (VPC) ID Example: vpc-hy521x6t
SubnetId	No	No	String	VPC subnet ID Example: sbnet-id-business

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
MountTargets	Array of FileSystemByVpcInfo	File System Name Example:View
TotalCount	Uint64	Total Number of File Systems Example: 0
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
InvalidParameterValue.InvalidVpcParameter	VPC parameter error.
InvalidParameterValue.InvalidFileSystemId	Invalid FileSystemId.
ResourceInsufficient.RegionSoldOut	Region resources sold out.
InvalidParameterValue.InvalidZoneOrZoneId	Invalid availability zone or availability zone ID.
InvalidParameterValue.MissingFileSystemId	FileSystemId missing.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
InvalidParameterValue.InvalidZoneId	Invalid availability zone.
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
InvalidParameterValue.UnavailableRegion	This availability zone cannot provide service.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.
InvalidParameterValue	Invalid parameter value
InvalidParameterValue.InvalidStorageType	Storage type error.
InvalidParameterValue.InvalidVip	The specified VIP is unavailable.

Error Code	Description
InvalidParameterValue.ZoneIdRegionNotMatch	ZoneId and Region do not match.
InternalError	Internal error
AuthFailure.UnauthorizedOperation	Request lacks CAM authorization.
FailedOperation	Operation failed
MissingParameter	Missing parameter error.
InvalidParameter	Wrong parameter.
InvalidParameterValue.InvalidPerMode	Invalid storage type.
InvalidParameterValue.MissingAppidOrUin	APPID and UIN cannot both be empty.
InvalidParameterValue.InvalidSubnetId	Invalid subnet ID.
InvalidParameterValue.InvalidVpcId	Invalid VPCID.
InvalidParameterValue.MissingFsParameter	FileSystem missing.
ResourceNotFound.SnapNotExit	Snapshot does not exist.
InvalidParameterValue.InvalidSnapStatus	Invalid snapshot status
InvalidParameterValue.SnapSrcFsBusy	The original FS of the snapshot is busy.
FailedOperation.UntagResourceFailed	Operation failure: Failed to unbind the resource tag.
FailedOperation.BindResourcePkgFailed	Failed to bind the resource package to the file system.
FailedOperation.ClientTokenInUse	Operation failure: Resources are being created.
InvalidParameter.InvalidTagKey	Parameter value error: Tag key cannot be empty.
InvalidParameter.ClientTokenLimitExceeded	Parameter value error: The string for ensuring request idempotency exceeds the length limit (64 bytes).
InvalidParameter.InvalidClientToken	Parameter value error: The string for ensuring request idempotency is incorrect.
AuthFailure	CAM signature/authentication error

DescribeMountTargets

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (DescribeMountTargets) is used to query mount target information of the file system.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-21 03:41:11.

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: DescribeMountTargets
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
FileSystemId	Yes	No	String	File system ID Example: cfs-2sczq3das

3. Output Parameters

Parameter name	Type	Description
MountTargets	Array of MountInfo	Mount target details Example:View

Parameter name	Type	Description
NumberOfMountTargets	Int64	Number of mount targets 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
InvalidParameter	Wrong parameter.
InternalError	Internal error
InvalidParameterValue.InvalidFileSystemId	Invalid FileSystemId.
InvalidParameterValue.MissingFileSystemId	FileSystemId missing.
InvalidParameterValue.MissingFsParameter	FileSystem missing.
ResourceNotFound.FileSystemNotFound	The file system does not exist.
ResourceNotFound.MountTargetNotFound	Mount target does not exist.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

UpdateAutoSnapshotPolicy

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

Updates the automatic scheduled snapshot policy

Default interface request rate limit: 20 times per second.

Interface update time: 2021-09-09 06:07:01.

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: UpdateAutoSnapshotPolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
AutoSnapshotPolicyId	Yes	No	String	Snapshot Policy ID Example: asp-xcweqdx
PolicyName	No	No	String	Snapshot policy name Example: cfspolicygz
DayOfWeek	Yes	No	String	Day of the week for scheduled snapshot backups Example: '1,2'
Hour	Yes	No	String	Hour for scheduled snapshot backups Example: '1,3,4'
AliveDays	No	No	Uint64	Snapshot retention date Example: 20
IsActivated	No	No	Uint64	Indicates whether to enable the scheduled snapshot feature Example: 0

3. Output Parameters

Parameter name	Type	Description
AutoSnapshotPolicyId	String	Snapshot Policy ID Example: asp-xcweqdx
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
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.
InternalError.Timeout	Timed out.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
AuthFailure	CAM signature/authentication error
UnsupportedOperation.UnauthorizedCfsQcsRole	CFS role not authorized.
InvalidParameter	Wrong parameter.
InvalidParameterValue.InvalidParamDayOfWeek	The value of DayOfWeek is invalid.
AuthFailure.UnauthorizedOperation	Request lacks CAM authorization.

UpdateCfsFileSystemName

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (UpdateCfsFileSystemName) is used to update the file system name.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-21 03:42:06.

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: UpdateCfsFileSystemName
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
CreationToken	No	No	String	Legacy version user- defined file system name, with a lower priority than FsName Example: fs-name-business
FileSystemId	Yes	No	String	File system ID Example: cfs-2sczq3das
FsName	No	No	String	User- defined file system name. Either this field or CreationToken must be specified. Example: fs-name-business

3. Output Parameters

Parameter name	Type	Description
CreationToken	String	User-defined file system name Example: fs-name-business
FileSystemId	String	File System ID Example: cfs-2sczq3das
FsName	String	User-defined file system name Example: fs-name-business

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	Wrong parameter.
InternalError	Internal error
ResourceNotFound	The resource does not exist
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
InvalidParameterValue.FsNameLimitExceeded	User-defined name too long (exceeds 64 bytes).
InvalidParameterValue.InvalidFsName	Invalid user-defined name.
InvalidParameterValue.InvalidFileSystemId	Invalid FileSystemId.
InvalidParameterValue.InvalidRegionZoneInfo	Region error. Either (ZoneName) or (ZoneId, Region) must be selected.
InvalidParameterValue.MissingFsParameter	FileSystem missing.
ResourceNotFound.FileSystemNotFound	The file system does not exist.
FailedOperation	Operation failed
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

UpdateCfsFileSystemPGroup

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (UpdateCfsFileSystemPGroup) is used to update the permission group used by the file system.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-21 03:42:18.

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: UpdateCfsFileSystemPGroup
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
PGroupId	Yes	No	String	Permission group ID Example: pg-xqwqewe
FileSystemId	Yes	No	String	File system ID Example: cfs-2sczq3das

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
PGroupId	String	Permission group ID Example: pg-xqwqewe
FileSystemId	String	File system ID Example: cfs-2sczq3das
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	Wrong parameter.
InternalError	Internal error
ResourceNotFound	The resource does not exist
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
FailedOperation.PgroupInUse	Permission group already bound to the file system.
InvalidParameterValue.InvalidFileSystemId	Invalid FileSystemId.
InvalidParameterValue.InvalidPgroup	The permission group does not belong to the user.
InvalidParameterValue.InvalidPgroupId	Invalid permission group ID.
InvalidParameterValue.InvalidRegionZoneInfo	Region error. Either (ZoneName) or (ZoneId, Region) must be selected.
InvalidParameterValue.MissingFsParameter	FileSystem missing.
ResourceNotFound.FileSystemNotFound	The file system does not exist.
ResourceNotFound.PgroupNotFound	Permission group does not exist.
FailedOperation	Operation failed
UnsupportedOperation	Operation unsupported.

Error Code	Description
FailedOperation.PgroupIsUpdating	The permission group is being updated.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

UpdateCfsFileSystemSizeLimit

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (UpdateCfsFileSystemSizeLimit) is used to update the file system storage capacity limit.

Default interface request rate limit: 60 times per second.

Interface update time: 2019-09-21 03:42:29.

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: UpdateCfsFileSystemSizeLimit
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
FsLimit	Yes	No	Uint64	File system capacity limit, in GB. Value range: 0-1073741824, where 0 indicates no limit on file system capacity. Ensure it is not less than the current total capacity and does not exceed the maximum specification value. Example: 1000
FileSystemId	Yes	No	String	File System ID Example: cfs-2sczq3das

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	Wrong parameter.
InternalError	Internal error
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
InvalidParameterValue.FsSizeLimitExceeded	The file system quota exceeds the limit.
InvalidParameterValue.InvalidFileSystemId	Invalid FileSystemId.
InvalidParameterValue.InvalidFsSizeLimit	Invalid file system quota.
InvalidParameterValue.InvalidRegionZoneInfo	Region error. Either (ZoneName) or (ZoneId, Region) must be selected.
InvalidParameterValue.MissingFsParameter	FileSystem missing.
ResourceNotFound.FileSystemNotFound	The file system does not exist.
FailedOperation	Operation failed
UnsupportedOperation	Operation unsupported.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

Other APIs
CopyFileSystemSnapshotCrossRegion

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

Cross-region replication of file system snapshots

Default interface request rate limit: 20 times per second.

Interface update time: 2021-08-26 02:55:27.

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: CopyFileSystemSnapshotCrossRegion
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
DestinationRegion	Yes	No	String	Example: cfs-copy-file-system-snapshot-cross-region-hsoa
SnapshotId	Yes	No	String	Snapshot ID Example: snapcfs-ailgaets
SnapshotName	No	No	String	Snapshot Name Example: snapshot-yf

3. Output Parameters

Parameter name	Type	Description
Snapshots	Array of SnapList	Snapshot 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

There is no error code related to business logic for this interface, For other error codes, see public ErrorCode.

CreateAutoSnapshotPolicy

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

Creates a scheduled snapshot policy

Default interface request rate limit: 20 times per second.

Interface update time: 2021-08-26 02:17:25.

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: CreateAutoSnapshotPolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
PolicyName	No	No	String	Policy Name Example: cfs-policy
DayOfWeek	No	No	String	Snapshot repeat date. Value range: Monday to Sunday Example: '2,3'
Hour	No	No	String	Snapshot repeat time point Example: '1,3,5'
AliveDays	No	No	Uint64	Snapshot retention duration Example: 3

3. Output Parameters

Parameter name	Type	Description
AutoSnapshotPolicyId	String	Snapshot Policy ID Example: asp-xcweqdx
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.SnapshotPolicyNameLimitExceeded	The length of the snapshot policy name exceeds the limit.

DeleteAutoSnapshotPolicy

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

Deleting snapshot policy

Default interface request rate limit: 20 times per second.

Interface update time: 2021-08-26 02:57:52.

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: DeleteAutoSnapshotPolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
AutoSnapshotPolicyId	Yes	No	String	File system policy ID Example: asp-xcweqdx

3. Output Parameters

Parameter name	Type	Description
AutoSnapshotPolicyId	String	Snapshot Policy ID Example: asp-xcweqdx

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
InvalidParameterValue.AutoPolicyNotFound	The snapshot policy does not exist.

UnbindAutoSnapshotPolicy

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

Unbinds a snapshot policy

Default interface request rate limit: 20 times per second.

Interface update time: 2021-08-26 02:39:03.

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: UnbindAutoSnapshotPolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
FileSystemIds	Yes	No	String	File system ID list Example: cfs-sdcxdads,cfs-21esda32
AutoSnapshotPolicyId	Yes	No	String	File system snapshot policy ID Example: asp-xcweqdx

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
AutoSnapshotPolicyId	String	File system snapshot policy ID Example: asp-xcweqdx
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
InvalidParameterValue.AutoPolicyNotFound	The snapshot policy does not exist.

Permission Group APIs
BatchAddCfsRules

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API is used to create permission group rules in batches.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-10 05:15:03.

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: BatchAddCfsRules
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
PermissionGroupId	Yes	No	String	Permission group ID Example: pgroup-dbne3lnb

Parameter name	Required	Allow NULL	Type	Description
RuleList	Yes	No	Array of InputPermissionGroupRules	Permission group rule list Example:View

3. Output Parameters

Parameter name	Type	Description
RuleList	Array of PGroupRuleInfo	Permission group rule 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
UnsupportedOperation	Operation unsupported.
FailedOperation	Operation failed
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.
InvalidParameterValue.InvalidPgroup	The permission group does not belong to the user.
InvalidParameterValue.InvalidPriority	Priority setting error.
InvalidParameter	Wrong parameter.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
FailedOperation.PgroupIsUpdating	The permission group is being updated.
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.

Error Code	Description
InvalidParameterValue.DuplicatedRuleAuthClientIp	Duplicate rule IP.
FailedOperation.PgroupInUse	Permission group already bound to the file system.
InvalidParameterValue.InvalidUserPermission	User permissions setting error.
InvalidParameterValue.InvalidRwPermission	Read/Write permission setting error.
InvalidParameterValue.InvalidAuthClientIp	Rule IP error.
ResourceInsufficient.RuleLimitExceeded	The number of rules exceeds the limit.
ResourceNotFound.PgroupNotFound	Permission group does not exist.
InternalError	Internal error

BatchDeleteCfsRule

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API is used to delete permission group rules in batches.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-10 06:07:35.

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: BatchDeleteCfsRule
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
PGroupId	Yes	No	String	Permission group ID Example: pgroup-dbne3lnb
RuleIds	Yes	No	Array of String	Rule ID Example: ["rule-r8dt3mph", "rule-hnp6c1wd"]

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
RuleIds	Array of String	Rule ID Example: ["rule-r8dt3mph", "rule-hnp6c1wd"]
PGroupId	String	Permission group ID Example: pg-1czq4zcdf
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	Wrong parameter.
InternalError	Internal error
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
FailedOperation.PgroupInUse	Permission group already bound to the file system.
InvalidParameterValue.DuplicatedRuleAuthClientIp	Duplicate rule IP.
InvalidParameterValue.InvalidAuthClientIp	Rule IP error.
InvalidParameterValue.InvalidPgroup	The permission group does not belong to the user.
InvalidParameterValue.InvalidPriority	Priority setting error.
InvalidParameterValue.InvalidRwPermission	Read/Write permission setting error.
InvalidParameterValue.InvalidUserPermission	User permissions setting error.
ResourceNotFound.PgroupNotFound	Permission group does not exist.
ResourceNotFound.RuleNotFound	Permission rule does not exist.
FailedOperation	Operation failed

Error Code	Description
UnsupportedOperation	Operation unsupported.
FailedOperation.PgroupIsUpdating	The permission group is being updated.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

CreateCfsPGroup

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (CreateCfsPGroup) is used to create permission groups.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-26 06:39:58.

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: CreateCfsPGroup
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
Name	Yes	No	String	Permission group name, 1-64 characters and can only include letters, digits, underscores, or hyphens Example: pg-zxcwsdasd
DescInfo	No	No	String	Permission group description, containing 1-255 characters Example: default_pgroup

3. Output Parameters

Parameter name	Type	Description
PGroupId	String	Permission group ID Example: pg-xqwqewe
Name	String	Permission group name Example: pg-yflei
DescInfo	String	Permission group description Example: pg-for-yf
BindCfsNum	Int64	Number of file systems bound to the permission group Example: 0
CDate	String	Permission group creation time Example: 2019-7-20 10:25:33
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
InvalidParameterValue.DuplicatedPgroupName	Duplicate permission group name.
InvalidParameterValue.InvalidPgroupName	Invalid permission group name.
InvalidParameterValue.PgroupDescinfoLimitExceeded	Permission group description exceeds the length limit (255 bytes).
InvalidParameterValue.PgroupNameLimitExceeded	Permission group name exceeds the length limit (64 bytes).
ResourceInsufficient.PgroupNumberLimitExceeded	The number of permission groups has reached the limit.
FailedOperation	Operation failed
InvalidParameterValue.MissingPgroupName	Permission group name cannot be empty.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

CreateCfsRule

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (CreateCfsRule) is used to create permission group rules.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-21 03:38:18.

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: CreateCfsRule
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
PGroupId	Yes	No	String	Permission group ID Example: pgroup-sdas12cxa
RWPermission	No	No	String	Read-write permission. Values: RO (read-only), RW (read-write). If not specified, this field will be ro by default. Example: rw
UserPermission	No	No	String	User permissions. all_squash: Map all accessing users to anonymous users or user groups. no_all_squash: First match accessing users with local users and then, if the match fails, map them to anonymous users or user groups. root_squash: Map accessing root users to anonymous users or user groups. no_root_squash: Allow accessing root users to keep their root account permissions. If not specified, this field will be root_squash by default. Example: root_squash
AuthClientIp	Yes	No	String	You can specify an IP address or an IP address range, such as 10.1.10.11 or 10.10.1.0/24. The default visitor address is *, indicating that all IP addresses are allowed. Note that you need to specify the CVM instance's private IP here. Example: 10.1.1.10
Priority	Yes	No	Int64	Rule priority. Value range: 1-100, where 1 means the highest, and 100 the lowest Example: 9

3. Output Parameters

Parameter name	Type	Description
RuleId	String	Rule ID Example: rule-xc2cxzcsd
PGroupId	String	Permission group ID Example: pg-s32xczs
AuthClientIp	String	Client IP Example: 10.1.1.10
RWPermission	String	R/W permission Example: rw
UserPermission	String	User permissions Example: root_squash
Priority	Int64	Priority Example: 9
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	Wrong parameter.
InternalError	Internal error
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
FailedOperation.PgroupInUse	Permission group already bound to the file system.
InvalidParameterValue.DuplicatedRuleAuthClientIp	Duplicate rule IP.
InvalidParameterValue.InvalidAuthClientIp	Rule IP error.
InvalidParameterValue.InvalidPgroup	The permission group does not belong to the user.
InvalidParameterValue.InvalidPriority	Priority setting error.
InvalidParameterValue.InvalidRwPermission	Read/Write permission setting error.
InvalidParameterValue.InvalidUserPermission	User permissions setting error.
ResourceInsufficient.RuleLimitExceeded	The number of rules exceeds the limit.
ResourceNotFound.PgroupNotFound	Permission group does not exist.
FailedOperation	Operation failed
UnsupportedOperation	Operation unsupported.
FailedOperation.PgroupIsUpdating	The permission group is being updated.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

DeleteCfsPGroup

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (DeleteCfsPGroup) is used to delete a permission group.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-21 03:38:47.

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: DeleteCfsPGroup
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
PGroupId	Yes	No	String	Permission group ID Example: pg-xqwqewe

3. Output Parameters

Parameter name	Type	Description
PGroupId	String	Permission group ID Example: pg-xcac2

Parameter name	Type	Description
AppId	Int64	User ID Example: 1255000652
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	Wrong parameter.
InternalError	Internal error
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
FailedOperation.PgroupInUse	Permission group already bound to the file system.
InvalidParameterValue.InvalidPgroup	The permission group does not belong to the user.
ResourceNotFound.PgroupNotFound	Permission group does not exist.
FailedOperation	Operation failed
UnsupportedOperation	Operation unsupported.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

DeleteCfsRule

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (DeleteCfsRule) is used to delete permission group rules.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-21 03:38:59.

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: DeleteCfsRule
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
PGroupId	Yes	No	String	Permission group ID Example: pg-xqwqewe
RuleId	Yes	No	String	Rule ID Example: rule-xc2cxzcsd

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
RuleId	String	Rule ID Example: rule-xc2cxzcsd
PGroupId	String	Permission group ID Example: pg-1czq4zcdf
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	Wrong parameter.
InternalError	Internal error
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
FailedOperation.PgroupInUse	Permission group already bound to the file system.
InvalidParameterValue.DuplicatedRuleAuthClientIp	Duplicate rule IP.
InvalidParameterValue.InvalidAuthClientIp	Rule IP error.
InvalidParameterValue.InvalidPgroup	The permission group does not belong to the user.
InvalidParameterValue.InvalidPriority	Priority setting error.
InvalidParameterValue.InvalidRwPermission	Read/Write permission setting error.
InvalidParameterValue.InvalidUserPermission	User permissions setting error.
ResourceNotFound.PgroupNotFound	Permission group does not exist.
ResourceNotFound.RuleNotFound	Permission rule does not exist.
FailedOperation	Operation failed

Error Code	Description
UnsupportedOperation	Operation unsupported.
FailedOperation.PgroupIsUpdating	The permission group is being updated.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

DescribeCfsPGroups

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (DescribeCfsPGroups) is used to query the permission group list.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-25 06:03:36.

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: DescribeCfsPGroups
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
InternationalFlag	No	No	Int64	When the value of this field is 1, the default permission group will use English. Example: 0
PGroupId	No	No	String	Permission Group ID Example: pgroup-1es6pwjb
Name	No	No	String	Permission Group Name Example: qta-searchpgrouplist-ez5g-1255000219
DescInfo	No	No	String	Permission group description Example: default_pgroup
OrderBy	No	No	String	Sort field: BindCfsNum or CDate Example: BindCfsNum
OrderByType	No	No	String	Sort mode: ASC or DESC Example: desc
Offset	No	No	Uint64	Offset paging number Example: 0
Limit	No	No	Uint64	Limit page size Example: 100

3. Output Parameters

Parameter	Type	Description

name		
PGroupList	Array of PGroupInfo	Permission group information list Example:View
TotalCount	Uint64	Total Number of File Systems Example: 10
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
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.

DescribeCfsRules

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (DescribeCfsRules) is used to query the permission group rules list.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-21 03:40:28.

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: DescribeCfsRules
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
PGroupId	Yes	No	String	Permission group ID Example: pg-xqwqewe

3. Output Parameters

Parameter name	Type	Description
RuleList	Array of PGroupRuleInfo	Permission group rule list Example:View

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.GetAccountStatusFailed	Failed to obtain user fee status.
FailedOperation	Operation failed
InvalidParameterValue.InvalidPgroup	The permission group does not belong to the user.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

UpdateCfsPGroup

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (UpdateCfsPGroup) is used to update permission group information.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-21 03:42:39.

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: UpdateCfsPGroup
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
Name	No	No	String	Permission group name, 1-64 characters and can only include letters, digits, underscores, or hyphens. The permission group name and description cannot both be empty. Example: gzgroup
PGroupId	Yes	No	String	Permission group ID Example: pg-xqwqewe
DescInfo	No	No	String	Permission group description, containing 1-255 characters Example: pg-group-for-yanfa

3. Output Parameters

Parameter name	Type	Description
PGroupId	String	Permission Group ID Example: pg-xqwqewe
Name	String	Permission Group Name Example: default_pgroup_name
DescInfo	String	Description Example: default_pgroup_name
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	Internal error
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
FailedOperation.MountTargetExists	Mount target exists in the file system.
InvalidParameterValue.DuplicatedPgroupName	Duplicate permission group name.
InvalidParameterValue.InvalidPgroupName	Invalid permission group name.
InvalidParameterValue.MissingNameOrDescinfo	Permission group name and description cannot both be empty.
InvalidParameterValue.PgroupDescinfoLimitExceeded	Permission group description exceeds the length limit (255 bytes).
InvalidParameterValue.PgroupNameLimitExceeded	Permission group name exceeds the length limit (64 bytes).
ResourceNotFound.PgroupNotFound	Permission group does not exist.
FailedOperation	Operation failed
UnsupportedOperation	Operation unsupported.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

UpdateCfsRule

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (UpdateCfsRule) is used to update permission rules.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-21 03:42:49.

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: UpdateCfsRule
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface
PGroupId	Yes	No	String	Permission group ID Example: pg-xqwqewe
RuleId	Yes	No	String	Rule ID Example: rule-xc2cxzcsd
AuthClientIp	No	No	String	You can specify an IP address or an IP address range, such as 10.1.10.11 or 10.10.1.0/24. The default visitor address is *, indicating that all IP addresses are allowed. Note that you need to specify the CVM instance's private IP here. Example: 10.0.0.10
RWPermission	No	No	String	Read-write permission. Values: RO (read-only), RW (read-write). If not specified, this field will be ro by default. Example: rw
UserPermission	No	No	String	User permissions. all_squash: Map all accessing users to anonymous users or user groups. no_all_squash: First match accessing users with local users and then, if the match fails, map them to anonymous users or user groups. root_squash: Map accessing root users to anonymous users or user groups. no_root_squash: Allow accessing root users to keep their root account permissions. If not specified, this field will be root_squash by default. Example: no_root_squash
Priority	No	No	Int64	Rule priority. Value range: 1-100, where 1 means the highest, and 100 the lowest Example: 7

3. Output Parameters

Parameter name	Type	Description
PGroupId	String	Permission group ID Example: pg-xqwqewe
RuleId	String	Rule ID Example: rule-xc2cxzcsd
AuthClientIp	String	Client IP address or IP address range allowed for access Example: 10.0.0.10
RWPermission	String	R/W permission Example: rw
UserPermission	String	User permissions Example: no_root_squash
Priority	Int64	Priority Example: 7
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	Wrong parameter.
InternalError	Internal error
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
FailedOperation.PgroupInUse	Permission group already bound to the file system.
InvalidParameterValue.DuplicatedRuleAuthClientIp	Duplicate rule IP.
InvalidParameterValue.InvalidAuthClientIp	Rule IP error.
InvalidParameterValue.InvalidPgroup	The permission group does not belong to the user.
InvalidParameterValue.InvalidPriority	Priority setting error.
InvalidParameterValue.InvalidRwPermission	Read/Write permission setting error.
InvalidParameterValue.InvalidUserPermission	User permissions setting error.
InvalidParameterValue.RuleNotMatchPgroup	Permission group rules and permission group do not match.
ResourceNotFound.PgroupNotFound	Permission group does not exist.
FailedOperation	Operation failed
UnsupportedOperation	Operation unsupported.
FailedOperation.PgroupIsUpdating	The permission group is being updated.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

Service APIs
DescribeAvailableZoneInfo

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (DescribeAvailableZoneInfo) is used to query the availability of regions.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-25 07:13:09.

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: DescribeAvailableZoneInfo
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface

3. Output Parameters

Parameter name	Type	Description
RegionZones	Array of AvailableRegion	Resource sale status and supported storage types, storage protocols, etc., in each availability zone 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	Internal error
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

DescribeCfsServiceStatus

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (DescribeCfsServiceStatus) is used to query the CFS service status of users.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-21 03:40:43.

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: DescribeCfsServiceStatus
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface

3. Output Parameters

Parameter name	Type	Description
CfsServiceStatus	String	The current CFS service status of the user. none: Not activated; creating: Being activated; created: Activated Example: none
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	Wrong parameter.
InternalError	Internal error
UnsupportedOperation	Operation unsupported.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

SignUpCfsService

1. Interface Description

Interface request domain name: cfs.api3.convergecloud.com.

This API (SignUpCfsService) is used to activate the CFS service.

Default interface request rate limit: 20 times per second.

Interface update time: 2019-09-21 03:41:38.

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: SignUpCfsService
Version	Yes	No	String	Public parameter, The value of this interface: 2019-07-19
Region	No	No	String	Public parameter,Regional information A list of supported geographies is available through the DescribeRegions interface

3. Output Parameters

Parameter name	Type	Description
CfsServiceStatus	String	The current CFS service status of the user. none: Not activated; creating: Being activated; created: Activated Example: creating
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	Wrong parameter.
InternalError	Internal error
UnsupportedOperation	Operation unsupported.
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.

Data structure

AvailableType

Version control - Type array

Referenced by the following interfaces: DescribeAvailableZoneInfo

name	required	NULL allowed	type	description
Protocols	yes	no	Array of AvailableProtoStatus	Protocol and sale details Example:View
Type	yes	no	String	Storage type. SD: Standard; HP: High Performance Example: SD

PGroupInfo

Permission group array

Referenced by the following interfaces: DescribeCfsPGroups

name	required	NULL allowed	type	description
PGroupId	yes	no	String	Permission Group ID Example: pg-asc2xasdf
Name	yes	no	String	Permission Group Name Example: default_pgroup
DescInfo	yes	no	String	Description Example: default_pgroup
CDate	yes	no	String	Creation Time Example: 2022-11-12 00:00:01
BindCfsNum	yes	no	Int64	Number of file systems associated Example: 1

AvailableProtoStatus

Version control - Protocol details

Referenced by the following interfaces: DescribeAvailableZoneInfo

name	required	NULL allowed	type	description
SaleStatus	yes	no	String	Sale status. sale_out: Sold out; saling: Available for sale; no_saling: Not for sale Example: sale_out
Protocol	yes	no	String	Protocol type. Values: NFS, CIFS Example: NFS

AvailableRegion

Version control - Region array

Referenced by the following interfaces: DescribeAvailableZoneInfo

name	required	NULL allowed	type	description
Region	yes	no	String	Region name. For example, "ap-region". Example: region
RegionName	yes	no	String	Region name, such as bj Example: ap-region1
RegionStatus	yes	no	String	Region availability. When at least one zone in the region is available for sale, the value is AVAILABLE; otherwise, it is UNAVAILABLE. Example: AVAILABLE
Zones	yes	no	Array of AvailableZone	Availability zone array Example:View
RegionCnName	yes	no	String	Region name, such as region1 Example: region1
RegionDisplayName	yes	no	String	Region display name, depending on the language. Example: region1

MountInfo

Mount Point Information

Referenced by the following interfaces: DescribeMountTargets

name	required	NULL allowed	type	description
FileSystemId	yes	no	String	File system ID Example: cfs-hge16re9
MountTargetId	yes	no	String	Mount target ID Example: mount-0ylt3o7o
IpAddress	yes	no	String	Mount target IP Example: 192.168.0.1
FSID	yes	no	String	Mount root directory Example: fwsuvx0l
LifeCycleState	yes	no	String	Mount target status Example: available
NetworkInterface	yes	no	String	Network Type Example: VPC
VpcId	yes	no	String	VPC ID Example: vpc-hq13cw04
VpcName	yes	no	String	VPC name Example: private-for-yanfa
SubnetId	yes	no	String	Subnet ID Example: sbnet-id-business
SubnetName	yes	no	String	Subnet Name Example: private-for-yanfa
PGroupName	yes	no	String	Permission Group Name Example: pg-home

PGroupRuleInfo

Permission group rule list

Referenced by the following interfaces: BatchAddCfsRules, DescribeCfsRules, OverrideCfsRules

name	required	NULL allowed	type	description
RuleId	yes	no	String	Rule ID Example: rule-gabatga4
AuthClientIp	yes	no	String	Client IP allowed for access Example: 10.0.0.2
RWPermission	yes	no	String	Read-write permissions, ro for read-only, rw for read-write Example: ro
UserPermission	yes	no	String	User permissions. all_squash: Map all accessing users to anonymous users or user groups. no_all_squash: First match accessing users with local users and then, if the match fails, map them to anonymous users or user groups. root_squash: Map accessing root users to anonymous users or user groups. no_root_squash: Allow accessing root users to keep their root account permissions. Example: all_squash
Priority	yes	no	Int64	Rule priority. Value range: 1-100, where 1 means the highest, and 100 the lowest Example: 2

FileSystemByPolicy

Information of the file system bound to the snapshot policy

Referenced by the following interfaces: DescribeAutoSnapshotPolicies

name	required	NULL allowed	type	description
CreationToken	yes	no	String	File System Name Example: fs-name-business
FileSystemId	yes	no	String	File System ID Example: cfs-x1x
SizeByte	yes	no	Uint64	File system size Example: 1
StorageType	yes	no	String	File system storage type Example: HP
TotalSnapshotSize	yes	no	Uint64	Snapshot Size Example: 1
Protocol	yes	no	String	Protocol Type Example: SMB

CfsTagUser

cfs tags

Referenced by the following interfaces: DescribeCfsTags

name	required	NULL allowed	type	description
TagId	yes	no	Uint64	tag id Example: 8
TagName	yes	no	String	Identifier name Example: school
Status	yes	no	Uint64	Open status Example: 1
SdSaleStatus	yes	yes	String	SD saleability Example: saling
HpSaleStatus	yes	yes	String	Saleability Example: saling

name	required	NULL allowed	type	description
ZoneId	yes	yes	Uint64	Availability zone ID Example: 100001
ISSMB	yes	yes	Uint64	SMB support Example: 1
ISNFS	yes	yes	Uint64	NFS support Example: 1

FileSystemInfo

Basic file system information

Referenced by the following interfaces: DescribeCfsFileSystems

name	required	NULL allowed	type	description
CreationTime	yes	no	String	Creation Time Example: 2020-08-11 20:57:17
CreationToken	yes	no	String	User-defined name Example: fs-name-business
FileSystemId	yes	no	String	File system ID Example: cfs-gabtqtq
LifeCycleState	yes	no	String	File system status Example: creating
SizeByte	yes	no	Uint64	File system capacity that is used Example: 10
SizeLimit	yes	no	Uint64	Maximum file system space limit Example: 50
ZoneId	yes	no	Uint64	Region ID Example: 100001
Zone	yes	no	String	Region name Example: ap-zone-1
Protocol	yes	no	String	File system protocol type Example: SMB
StorageType	yes	no	String	File system storage type Example: HP
StorageResourcePkg	yes	no	String	Prepaid storage package bound to the file system (not yet supported) Example: no support current
BandwidthResourcePkg	yes	no	String	Prepaid bandwidth package bound to the file system (not yet supported) Example: no support current
PGroup	yes	no	PGroup	Information of the permission group bound to the file system Example:View
FsName	yes	no	String	User-defined name Example: fs-name-business
Encrypted	yes	no	Bool	Indicates whether the file system is encrypted Example: true
KmsKeyId	yes	no	String	Encryption key used, which can be the key ID or ARN Example: gaglkjgfpnggagalkinebn14tw
SizeLimitMax	yes	no	Uint64	Maximum capacity limit Example: 160000000
VpcId	yes	no	String	VPC ID Example: vpc-hq13cw04
IpAddress	yes	no	String	Mount target IP address Example: 192.168.0.1
AllocedSpace	yes	no	Uint64	Allocated space Example: 200000
TagId	yes	no	Uint64	CFS resource pool tag ID Example: 8
TagName	yes	no	String	CFS resource pool tag name Example: school
SnapId	yes	no	String	Snapshot ID Example: 4f3041c8e5-a94a-10ba-4f57-8a8643b957
SnapStatus	yes	no	String	Snapshot status Example: snapping
BandwidthLimit	yes	no	Float	Bandwidth Cap Example: 100
AppId	yes	no	Uint64	appid Example: 1255000652
DrcType	no	no	String	DRC type. Example: normal
BrcStatus	no	no	String	BRC status. Example: normal
SubnetId	no	no	String	VPC subnet ID. Example: subnet-eob81puc

CfsSnapJobHistory

History of file system snapshot operations

Referenced by the following interfaces: CopyFileSystemSnapshotCrossRegion, DescribeCfsSnaps

name	required	NULL allowed	type	description
RegionName	yes	no	String	Region Example: ap-region1
CreateTime	yes	no	String	Creation Time Example: 2021-09-15T19:04:45+08:00
UpdateTime	yes	no	String	Update time Example: 2021-09-15T19:04:45+08:00
JobType	yes	no	String	Task Type Example: Snapshot
Operater	yes	no	String	Operator Example: yflei
Status	yes	no	Uint64	Task Status Example: 1
Message	yes	no	String	Task details Example: success

FileSystemClient

File system client information

Referenced by the following interfaces: DescribeCfsFileSystemClients

name	required	NULL allowed	type	description
CfsVip	yes	no	String	File system IP address Example: 10.0.2.3
ClientIp	yes	no	String	Client IP address Example: 10.0.2.2
VpcId	yes	no	String	File system VPCID Example: vpc-agabral
Zone	yes	no	String	AZ name. For example, ap-zone-1. Example: ap-zone-1
ZoneName	yes	no	String	Name of Availability Zone Example: Region 1
MountDirectory	yes	no	String	Information of the path through which the file system is mounted on the client Example: /home/mnt
ZoneId	yes	no	Uint64	Availability zone ID Example: 100001
LinkStatus	yes	no	Uint64	Link status. 1: Connected; 2: Disconnected Example: 2
MonitorStatus	yes	no	Uint64	Monitor status. 1: Monitor installed; 2: Monitor not installed Example: 1
Protocol	yes	no	String	Protocol. v3: NFSv3 mount; v4: NFSv4 mount Example: SMB
UpdateTime	no	no	String	Link update time. Example: 2021-09-15T19:04:45+08:00

AutoSnapshotPolicyInfo

Snapshot Policy

Referenced by the following interfaces: DescribeAutoSnapshotPolicies

name	required	NULL allowed	type	description
AutoSnapshotPolicyId	yes	no	String	Snapshot Policy ID Example: asp-xcweqdx
PolicyName	yes	no	String	Policy Name Example: default
CreationTime	yes	no	String	Creation Time Example: 2020-05-23 19:00:45
FileSystemNums	yes	no	Uint64	Number of file systems associated Example: 2
DayOfWeek	yes	no	String	Week Example: '1,2'
Hour	yes	no	String	Hour Example: '3,4'

name	required	NULL allowed	type	description
IsActivated	yes	no	Uint64	Activation status Example: 1
NextActiveTime	yes	no	String	Next execution Example: 2020-05-25 19:00:45
Status	yes	no	String	Policy status Example: 1
AliveDays	yes	no	Uint64	Retention Period Example: 3
RegionName	yes	no	String	Region Example: ap-region1
FileSystems	yes	no	Array of FileSystemByPolicy	Associated file system information Example:View
AppId	yes	no	Uint64	appid Example: 1255000652

InputPermissionGroupRules

Input parameter permission group rule list

Referenced by the following interfaces: BatchAddCfsRules, OverrideCfsRules

name	required	NULL allowed	type	description
AuthClientIp	yes	no	String	Client IP allowed for access Example: 10.0.0.2
RWPermission	yes	no	String	Read-write permissions, ro for read-only, rw for read-write Example: ro
UserPermission	yes	no	String	User permissions. all_squash: Map all accessing users to anonymous users or user groups. no_all_squash: First match accessing users with local users and then, if the match fails, map them to anonymous users or user groups. root_squash: Map accessing root users to anonymous users or user groups. no_root_squash: Allow accessing root users to keep their root account permissions. Example: all_squash
Priority	yes	no	Uint64	Rule priority. Value range: 1-100, where 1 means the highest, and 100 the lowest Example: 2

PeriodStatisticsSet

Global statistics information

Referenced by the following interfaces: DescribeStatistics

name	required	NULL allowed	type	description
StandardFilesystemCount	yes	no	Uint64	Number of Standard file systems Example: 0
HighPerformanceFilesystemCount	yes	no	Uint64	Number of High Performance file systems Example: 0
StandardFilesystemStorage	yes	no	Float	Storage capacity of Standard file system Example: 0
HighPerformanceFilesystemStorage	yes	no	Float	Storage capacity of High Performance file system Example: 0
StatisticalPeriod	yes	no	String	Statistical Period Example: 0

Filter

Filter parameter

Referenced by the following interfaces: DescribeAutoSnapshotPolicies

name	required	NULL allowed	type	description
Name	yes	no	String	Name Example: cfsld
Values	yes	no	Array of String	Variable Example: cfs-2sczq3das

PGroup

Information of the permission group bound to the file system

Referenced by the following interfaces: DescribeCfsFileSystems

name	required	NULL allowed	type	description
PGroupId	yes	no	String	Permission Group ID Example: pg-asc2xasdf
Name	yes	no	String	Permission Group Name Example: defaultname

SnapList

File system snapshot list

Referenced by the following interfaces: CopyFileSystemSnapshotCrossRegion, DescribeCfsSnaps

name	required	NULL allowed	type	description
RegionName	yes	no	String	Region name Example: ap-region1
SnapId	yes	no	String	Snapshot ID Example: 4f3041c8e5-a94a-10ba-4f57-8a8643b957
SnapName	yes	no	String	Snapshot Name Example: qta-snap-yf-01
FileSystemId	yes	no	String	File system ID Example: cfs-asd2ada
FsName	yes	no	String	File system name Example: fs-name-business
Size	yes	no	Uint64	Snapshot size in bytes Example: 150000000
StorageType	yes	no	String	Storage Type Example: HP
AliveDay	yes	no	Uint64	Retention period in days Example: 7
CreateTime	yes	no	String	Creation Time Example: 2021-09-15T19:04:45+08:00
UpdateTime	yes	no	String	Update time Example: 2021-09-15T19:04:45+08:00
Status	yes	no	Uint64	Snapshot status Example: 1
Percent	yes	no	Uint64	Snapshot progress Example: 20
JobHistorys	yes	no	Array of CfsSnapJobHistory	Operation history Example:View
JobId	yes	no	String	Snapshot task ID Example: snapcfs-muqq5p9z
BackupId	no	no	String	Backup ID Example: backup-ns2gv8al
BackupCreateTime	no	no	String	Backup creation time Example: 2025-07-28 07:25:57

AvailableZone

Version control - Availability zone array

Referenced by the following interfaces: DescribeAvailableZoneInfo

name	required	NULL allowed	type	description
Zone	yes	no	String	AZ name Example: ap-zone-1
ZoneId	yes	no	Int64	Availability zone ID Example: 800001
ZoneCnName	yes	no	String	Name of Availability Zone Example: zone1
Types	yes	no	Array of AvailableType	Type array Example:View
ZoneEnName	yes	no	String	Availability zone name Example: ap-zone-2
ZoneDisplayName	yes	no	String	Availability zone display name, depending on the language. Example: Region 2

FileSystemByVpcInfo

Queries file systems by VPC

Referenced by the following interfaces: DescribeCfsFileSystemsByVpc

name	required	NULL allowed	type	description
FileSystemId	yes	no	String	File System ID Example: cfs-asd2ada
IpAddress	yes	no	String	IP address Example: 192.168.0.1
FSID	yes	no	String	Mount root directory Example: fwsuv20l

name	required	NULL allowed	type	description
LifeCycleState	yes	no	String	Mount target status Example: available
MountTargetId	yes	no	String	Mount target ID Example: mount-0yit3o7o
NetworkInterface	yes	no	String	Network Type Example: VPC
VpcId	yes	no	String	VPC ID Example: vpc-hq13cw04
SubnetId	yes	no	String	Subnet ID Example: sbnet-id-business
ZoneId	yes	no	Int64	Availability zone ID Example: 100001

ResourceTags

Tag

Referenced by the following interfaces: CreateClsFileSystem

name	required	NULL allowed	type	description
TagKey	yes	no	String	Tag Key Example: Department
TagValue	yes	no	String	Tag Value Example: Product department 1

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
InvalidParameterValue.InvalidParamDayOfWeek	The value of DayOfWeek is invalid.

Error codes	Description
InternalError.BillingJipingInterfaceQuotaFail	The measurement quota is insufficient.
InvalidParameterValue.PgroupNameLimitExceeded	Permission group name exceeds the length limit (64 bytes).
UnsupportedOperation.OutOfService	Out of service due to overdue payment. Top up and try again.
InvalidParameterValue.InvalidEncrypted	Encryption parameter error.
InvalidParameterValue.InvalidPerMode	Invalid storage type.
InternalError.GetAccountStatusFailed	Failed to obtain user fee status.
InvalidParameterValue.InvalidPriority	Priority setting error.
InvalidParameterValue.InvalidFileSystemId	Invalid FileSystemId.
FailedOperation.PgroupIsUpdating	The permission group is being updated.
ResourceInsufficient.RuleLimitExceeded	The number of rules exceeds the limit.
InvalidParameterValue.InvalidRegionZoneInfo	Region error. Either (ZoneName) or (ZoneId, Region) must be selected.
InvalidParameterValue	Invalid parameter value
InvalidParameterValue.InvalidUserPermission	User permissions setting error.
InvalidParameterValue.ZoneIdRegionNotMatch	ZoneId and Region do not match.
ResourceInsufficient.FileSystemLimitExceeded	The number of file systems has reached the limit.
InvalidParameterValue.InvalidProtocol	Protocol error.
InvalidParameterValue.InvalidRwPermission	Read/Write permission setting error.
InvalidParameterValue.FsNameLimitExceeded	User-defined name too long (exceeds 64 bytes).
InvalidParameterValue.InvalidSnapStatus	Invalid snapshot status
InvalidParameterValue.SnapSrcFsBusy	The original FS of the snapshot is busy.
InvalidParameterValue.MissingSubnetidOrUnsubnetid	SUBNETID and UNSUBNETID cannot both be empty.
InvalidParameterValue.MissingVpcParameter	VPC parameter missing.

Error codes	Description
FailedOperation.UntagResourceFailed	Operation failure: Failed to unbind the resource tag.
ResourceNotFound.FileSystemNotFound	The file system does not exist.
ResourceInsufficient.RegionSoldOut	Region resources sold out.
InvalidParameterValue.UnavailableZone	This region cannot provide service.
InvalidParameterValue.MissingZoneId	ZoneID missing.
InvalidParameterValue.MissingKmsKeyId	Key ID or ARN missing.
InvalidParameterValue.AutoPolicyNotFound	The snapshot policy does not exist.
ResourceInsufficient.SubnetIpAllOccupied	No available IP under this subnet.
InvalidParameterValue.InvalidFsSizeLimit	Invalid file system quota.
ResourceNotFound.MountTargetNotFound	Mount target does not exist.
AuthFailure.UnauthorizedOperation	Request lacks CAM authorization.
InvalidParameterValue.MissingVpcidOrUnvpcid	VPCID and UNVPCID cannot both be empty.
InvalidParameterValue.MissingStorageResourcePkg	No storage package is bound.
InvalidParameterValue.InvalidNetInterface	Invalid network type.
FailedOperation.MountTargetExists	Mount target exists in the file system.
InvalidParameterValue.UnavailableRegion	This availability zone cannot provide service.
InvalidParameterValue.MissingAppidOrUin	APPID and UIN cannot both be empty.
InvalidParameterValue.DuplicatedRuleAuthClientIp	Duplicate rule IP.
ResourceNotFound.PgroupNotFound	Permission group does not exist.
InvalidParameterValue.MissingFileSystemId	FileSystemId missing.
InvalidParameterValue.InvalidFsStatus	Invalid file system status.
ResourceNotFound.SnapNotFound	The snapshot is not found.
UnsupportedOperation	Operation unsupported.
InvalidParameterValue.InvalidSubnetId	Invalid subnet ID.

Error codes	Description
InvalidParameterValue.InvalidPgroupId	Invalid permission group ID.
InvalidParameterValue.InvalidVip	The specified VIP is unavailable.
InvalidParameterValue.InvalidMountTargetIp	Invalid mount target IP.
InvalidParameterValue.InvalidFsName	Invalid user-defined name.
ResourceNotFound.FsNotExist	File system does not exist.
InvalidParameterValue.InvalidPgroupName	Invalid permission group name.
InvalidParameter.InvalidClientToken	Parameter value error: The string for ensuring request idempotency is incorrect.
ResourceNotFound.RuleNotFound	Permission rule does not exist.
FailedOperation.PgroupInUse	Permission group already bound to the file system.
InvalidParameter.InvalidTagKey	Parameter value error: Tag key cannot be empty.
InvalidParameterValue.MissingZoneOrZoneId	Region error. Either (Zone) or (Zone_id) must be selected.
InvalidParameterValue.InvalidZoneOrZoneId	Invalid availability zone or availability zone ID.
InvalidParameterValue.MissingFsParameter	FileSystem missing.
InvalidParameterValue.InvalidStorageType	Storage type error.
InternalError	Internal error
InvalidParameterValue.InvalidZoneId	Invalid availability zone.
InvalidParameterValue.InvalidPgroup	The permission group does not belong to the user.
FailedOperation.BindResourcePkgFailed	Failed to bind the resource package to the file system.
ResourceInsufficient	Insufficient resources
InvalidParameterValue.InvalidVpcParameter	VPC parameter error.
InvalidParameterValue.PgroupDescinfoLimitExceeded	Permission group description exceeds the length limit (255 bytes).

Error codes	Description
InvalidParameterValue.FsSizeLimitExceeded	The file system quota exceeds the limit.
ResourceNotFound	The resource does not exist
FailedOperation	Operation failed
UnsupportedOperation.UnverifiedUser	User not authenticated with real name.
InvalidParameterValue.InvalidVpcId	Invalid VPCID.
ResourceInsufficient.TagLimitExceeded	Insufficient resources: The number of tags for this resource has reached the limit.
InvalidParameterValue.MissingNameOrDescinfo	Permission group name and description cannot both be empty.
InvalidParameterValue.RuleNotMatchPgroup	Permission group rules and permission group do not match.
InvalidParameterValue.MissingPgroupName	Permission group name cannot be empty.
AuthFailure	CAM signature/authentication error
InvalidParameterValue.DuplicatedPgroupName	Duplicate permission group name.
FailedOperation.ClientTokenInUse	Operation failure: Resources are being created.
InvalidParameterValue.InvalidAuthClientIp	Rule IP error.
ResourceNotFound.SnapNotExit	Snapshot does not exist.
MissingParameter	Missing parameter error.
UnsupportedOperation.UnauthorizedCfsQcsRole	CFS role not authorized.
InvalidParameter	Wrong parameter.
ResourceInsufficient.PgroupNumberLimitExceeded	The number of permission groups has reached the limit.
InternalError.SnapshotPolicyNameLimitExceeded	The length of the snapshot policy name exceeds the limit.
InternalError.Timeout	Timed out.
InternalError.CreateFsFailed	Failed to create the file system.

Error codes	Description
InvalidParameter.ClientTokenLimitExceeded	Parameter value error: The string for ensuring request idempotency exceeds the length limit (64 bytes).