

Cloud Object Storage User Tools Product Documentation





Copyright Notice

©2013-2022 Tencent Cloud. All rights reserved.

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

Trademark Notice

🔗 Tencent Cloud

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

Service Statement

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

Contents

User Tools
Tool Overview
Installation and Configuration of Environment
Java
Python
Hadoop
COSBrowser
Overview
User Guide for Desktop Version
User Guide for Mobile Version
Installation and Login
Mobile Version Features
Bucket Management and Operations
File Management and Operations
Data Monitoring
COSCLI (Beta)
COSCLI Overview
Download and Installation Configuration
Common Options
Common Commands
Generating and Modifying Configuration Files - config
Creating a Bucket - mb
Deleting a Bucket - rb
Tagging Bucket - bucket-tagging
Querying Bucket/Object List - Is
Obtaining Statistics on Different Types of Objects - du
Uploading/Downloading/Copying Objects - cp
Syncing Upload/Download/Copy - sync
Deleting Objects - rm
Getting File Hash Value - hash
Listing Incomplete Multipart Uploads - Isparts
Clearing Incomplete Multipart Uploads - abort
Restoring Archived Objects - restore
Getting Pre-signed URL - signurl
FAQs

COSCMD

COS Migration

FTP Server

COSFS

Hadoop

COSDistCp

Hadoop-cos-DistChecker

HDFS TO COS

Online Auxiliary Tools

COS Request Tool

Diagnostic Tool

User Tools Tool Overview

Last updated : 2022-10-12 14:05:53

Tool	Features		
COSBrowser	Provides a visualized UI for users to easily upload/download data, generate access URL, and perform other operations.		
COSCLI	Enables users to upload, download, delete, and perform other operations on COS objects with simple commands.		
COSCMD	Enables users to perform operations (such as upload, download, and delete) in batches with simple commands.		
COS Migration	Migrates data from multiple data sources (such as an on-premises server, and other cloud storage services) to COS.		
FTP Server	Enables users to upload/download files to/from COS by using an FTP client.		
COSFS	In Linux, this tool is used to mount buckets to a local file system and to perform operations on objects in COS via the local file system.		
Hadoop Tool	Helps integrate COS with big data computing frameworks such as Hadoop, Spark, and Tez, so that they can read and write COS data.		
COSDistCp	Implements data copy between HDFS and COS. It is a distributed copy tool based on the MapReduce framework.		
Hadoop-cos-DistChecker	Verifies the directory integrity after you use the hadoop distop command to migrate data from HDFS to COS.		
HDFS TO COS	Copies data from HDFS to COS.		
Diagnostic Tool	COS's web-based diagnostic tool that allows you to troubleshoot error requests.		

If you need to use any other tools, please inform us by submitting a tool request, and we will promptly and carefully consider your request.

Installation and Configuration of Environment Java

Last updated : 2022-10-12 14:05:53

Java Development Kit (JDK) is the SDK for Java. This document takes JDK 1.7 and 1.8 as examples to describe how to install and configure JDK under Windows and Linux systems.

Windows

1. Downloading a JDK

Go to the Oracle website to download the desired JDK version.

2. Installation

Install the JDK as instructed. You can specify the installation paths (C drive by default), for example, as

D:\Program Files\Java\jdk1.8.0_31 and D:\Program Files\Java\jre1.8.0_31 .

3. Configuration

After the installation is completed, right-click **Computer**, and then click **Properties** > **Advanced system settings** > **Environment Variables** > **System variables** > **New** to configure the software.

Variable name (N): JAVA_HOME

Variable value (V): D:\Program Files\Java\jdk1.8.0_31 (Configure according to your actual installation path).

Variable name (N): CLASSPATH

Variable value (V): .;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar; (Note that the variable value begins with .).

Variable name (N): **Path**

Variable value (V): %JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;

4. Testing

Test whether the configuration is successful: click **Start** (or shortcut: Win+R) > **Run** (enter cmd) > **OK** (or press Enter), then enter the command javac and press Enter. If messages such as command parameters and syntax are displayed, the environment variables are configured successfully.

Linux

If openjdk is installed by using yum or apt-get command, the class library may be incomplete, thus leading to errors when you run relevant tools after the installation. Therefore, we recommend that you manually decompress and install JDK. Specific steps are as follows:

1. Download a JDK

Go to the Oracle website to download the desired JDK version to install.

Note :

```
The following uses jdk-8u151-linux-x64.tar.gz as an example. If you are using other versions, ensure that the extension is .tar.gz .
```

2. Create a directory

Run the following command to create the java directory in /usr/:

```
mkdir /usr/java
cd /usr/java
```

Copy the downloaded jdk-8u151-linux-x64.tar.gz to the /usr/java/ directory.

3. Decompress the JDK

Run the following command to decompress the JDK:

```
tar -zxvf jdk-8u151-linux-x64.tar.gz
```

4. Set environment variables

Edit the /etc/profile file. Add the following content to the profile file and save it:

```
set java environment
JAVA_HOME=/usr/java/jdk1.8.0_151
JRE_HOME=/usr/java/jdk1.8.0_151/jre
CLASS_PATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/lib
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
export JAVA_HOME JRE_HOME CLASS_PATH PATH
```

Note:

JAVA_HOME and JRE_HOME should be configured according to the actual installation paths and JDK version.



Run the following command for the modifications to take effect:

source /etc/profile

5. Test

Run the following command to test the JDK installation:

java -version

If information about the Java version is displayed, the JDK is installed successfully.

```
java version "1.8.0_151"
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)
```

Python

Last updated : 2022-10-12 14:05:53

This document describes how to install Python for different operating systems.

Using an Installation Package

1. Download a package

Go to the Python website to download an installation package according to your OS.

Note:

Python has dropped support for Python 2 since January 1, 2020. Therefore, you are advised to install Python 3.

2. Install the package

Install the downloaded package as instructed.

Note :

If you use Windows, check Add Python to environment variables.

Python 3.9.5 (64-bit) Setu	p —	
	Advanced Options	
	✓ Install for <u>a</u> ll users	
	Associate files with Python (requires the py launch of	
	Create shortcuts for installed applications	
	Add Python to environment variables	
	Precompile standard library	
	Download debugging symbols	
	Download debug binaries (requires VS 2017 or later)	
	Customize install location	
	C:\Program Files\Python39\	B <u>r</u> owse
python	You will require write permissions for the selected location.	
windows	<u>B</u> ack <u>I</u> nstall	<u>C</u> ancel

3. Verify the installation

Run the following command in Terminal to view the Python version:

```
python -V
```

If the Python version is displayed, Python has been installed successfully.

Note :

If you use Windows, you may need to restart your computer after the installation.

4. Configure environment variables

In Windows, if "not recognized as an internal or external command" is reported in Terminal after the command above is run, right-click the **Computer** icon, click **Properties** > **Advanced system settings** > **Environment Variables**, and in the **System variables** area, click **New** to add the Python installation path:

Using a Package Manager

macOS

Install HomeBrew and use it to install Python:

```
brew install python
```

Ubuntu

Use the built-in Advanced Packaging Tool (APT) to install Python:

sudo apt-get install python

CentOS

Use the built-in Yellowdog Updater, Modified (YUM) to install Python:

```
sudo yum install -y python
```

Hadoop

Last updated : 2022-10-12 14:05:53

Hadoop (2.7.2 or above) tool provides the capability to run computing tasks using Tencent Cloud COS as the underlying file storage system. The Hadoop cluster can be launched in three modes: stand-alone, pseudo-distributed, and fully-distributed. This document uses Hadoop-2.7.4 as an example to describe how to build a fully-distributed Hadoop environment and how to use wordcount to execute a simple test.

Preparation

- 1. Prepare several servers.
- 2. Install and configure the system: CentOS-7-x86_64-DVD-1611.iso。.
- 3. Install the Java environment. For more information, see Installing and Configuring Java.
- 4. Install the available Hadoop package: Apache Hadoop Releases Download。 .

Network Configuration

Use ifconfig -a to check the IP of each server, then use the ping command to check if they can ping each other, and record the IP of each server.

Configuring CentOS

Configure hosts

vi /etc/hosts

Edit the content:

```
202.xxx.xxx master
202.xxx.xxx.xxx slave1
202.xxx.xxx.xxx slave2
202.xxx.xxx.xxx slave3
//Replace IPs with the real ones
```

Turn off firewall

```
systemctl status firewalld.service //Check firewall status
systemctl stop firewalld.service //Turn off firewall
```



systemctl disable firewalld.service //Disable firewall to start on boot

Time synchronization

```
yum install -y ntp //Install ntp service
ntpdate cn.pool.ntp.org //Sync network time
```

Install and configure JDK

Upload JDK installer package (such as jdk-8u144-linux-x64.tar.gz) to the root directory.

```
mkdir /usr/java
tar -zxvf jdk-8u144-linux-x64.tar.gz -C /usr/java/
rm -rf jdk-8u144-linux-x64.tar.gz
```

Copy JDKs among hosts

```
scp -r /usr/java slave1:/usr
scp -r /usr/java slave2:/usr
scp -r /usr/java slave3:/usr
.....
```

Configure environment variables for JDK of each host

```
vi /etc/profile
```

Edit the content:

```
export JAVA_HOME=/usr/java/jdk1.8.0_144
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
source/etc/profile //Make the configuration file take effect
java -version //View Java version
```

Configuring Keyless Access via SSH

Check the SSH service status on each host:

```
systemctl status sshd.service //Check the SSH service status.
yum install openssh-server openssh-clients //Install the SSH service. Ignore this
step if it is already installed.
systemctl start sshd.service //Enable the SSH service. Ignore this step if it is
already enabled.
```



Generate a key on each host:

ssh-keygen -t rsa //Generate Keys

On slave1:

```
cp ~/.ssh/id_rsa.pub ~/.ssh/slave1.id_rsa.pub
scp ~/.ssh/slave1.id_rsa.pub master:~/.ssh
```

On slave2:

```
cp ~/.ssh/id_rsa.pub ~/.ssh/slave2.id_rsa.pub
scp ~/.ssh/slave2.id_rsa.pub master:~/.ssh
```

And so on...

On master:

```
cd ~/.ssh
cat id_rsa.pub >> authorized_keys
cat slave1.id_rsa.pub >>authorized_keys
cat slave2.id_rsa.pub >>authorized_keys
scp authorized_keys slave1:~/.ssh
scp authorized_keys slave2:~/.ssh
```

Installing and Configuring Hadoop

Installing Hadoop

Upload the Hadoop installer package (such as hadoop-2.7.4.tar.gz) to the root directory.

```
tar -zxvf hadoop-2.7.4.tar.gz -C /usr
rm -rf hadoop-2.7.4.tar.gz
mkdir /usr/hadoop-2.7.4/tmp
mkdir /usr/hadoop-2.7.4/logs
mkdir /usr/hadoop-2.7.4/hdf
mkdir /usr/hadoop-2.7.4/hdf/data
mkdir /usr/hadoop-2.7.4/hdf/name
```

Go to the hadoop-2.7.4/etc/hadoop directory and proceed to the next step.

Configure Hadoop

1. Add the following to the hadoop-env.sh file.

```
export JAVA_HOME=/usr/java/jdk1.8.0_144
```

If the SSH port is not 22 (default value), modify it in the hadoop-env.sh file:

export HADOOP_SSH_OPTS="-p 1234"

2. Modify yarn-env.sh

```
export JAVA_HOME=/usr/java/jdk1.8.0_144
```

3. Modify slaves

Configure the content:

Delete: localhost Add: slave1 slave2 slave3

4. Modify core-site.xml

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://master:9000</value>
</property>
<name>hadoop.tmp.dir</name>
<value>file:/usr/hadoop-2.7.4/tmp</value>
</property>
</configuration>
```

5. Modify hdfs-site.xml

```
<configuration>
<property>
<name>dfs.datanode.data.dir</name>
<value>/usr/hadoop-2.7.4/hdf/data</value>
<final>true</final>
```

```
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>/usr/hadoop-2.7.4/hdf/name</value>
<final>true</final>
</property>
</configuration>
```

6. Modify mapred-site.xml

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
<name>mapreduce.jobhistory.address</name>
<value>master:10020</value>
</property>
<property>
<name>mapreduce.jobhistory.webapp.address</name>
<value>master:19888</value>
</property>
</configuration>
```

7. Modify yarn-site.xml

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.mapred.ShuffleHandler</value>
</property>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>master:8032</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>master:8030</value>
</property>
<property>
```

```
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>master:8031</value>
</property>
<name>yarn.resourcemanager.admin.address</name>
<value>master:8033</value>
</property>
<name>yarn.resourcemanager.webapp.address</name>
<value>master:8088</value>
</property>
</configuration>
```

8. Copy Hadoop among hosts

scp -r /usr/ hadoop-2.7.4 slave1:/usr scp -r /usr/ hadoop-2.7.4 slave2:/usr scp -r /usr/ hadoop-2.7.4 slave3:/usr

9. Configure environment variables for Hadoop of each host

Open the configuration file:

vi /etc/profile

Edit the content:

```
export HADOOP_HOME=/usr/hadoop-2.7.4
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export HADOOP_LOG_DIR=/usr/hadoop-2.7.4/logs
export YARN_LOG_DIR=$HADOOP_LOG_DIR
```

Implement the configuration file:

source /etc/profile

Start Hadoop

1. Format namenode

```
cd /usr/hadoop-2.7.4/sbin
hdfs namenode -format
```

2. Start

```
cd /usr/hadoop-2.7.4/sbin
start-all.sh
```

3. Check processes

If processes on master contain ResourceManager, SecondaryNameNode and NameNode, Hadoop starts successfully. For example:

2212 ResourceManager2484 Jps1917 NameNode2078 SecondaryNameNode

If processes on each slave contain DataNode and NodeManager, Hadoop starts successfully. For example:

17153 DataNode 17334 Jps 17241 NodeManager

Running wordcount

The wordcount built in Hadoop can be called directly. After Hadoop starts, use the following command to work with files in HDFS:

```
hadoop fs -mkdir input
hadoop fs -put input.txt /input
hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar wordcount /
input /output/
```

[root@VM_96_24_centos /usr/hadoop-2.7.4]# hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar wordcount /input /output/
17/07/18 23:04:51 INFO client.RMProxy: Connecting to ResourceManager at master/10.104.96.24:8032
17/07/18 23:04:53 INFO input.FileInputFormat: Total input paths to process : 1
17/07/18 23:04:53 INFO mapreduce.JobSubmitter: number of splits:1
17/07/18 23:04:54 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1500344813707_0002
17/07/18 23:04:54 INFO impl.YarnClientImpl: Submitted application application_1500344813707_0002
17/07/18 23:04:54 INFO mapreduce.Job: The url to track the job: http://master:8088/proxy/application_1500344813707_0002/
17/07/18 23:04:54 INFO mapreduce.Job: Running job: job_1500344813707_0002
17/07/18 23:05:01 INFO mapreduce.Job: Job job 1500344813707_0002 running in uber mode : false
17/07/18 23:05:01 INFO mapreduce.Job: map 0% reduce 0%
17/07/18 23:05:05 INFO mapreduce.Job: map 100% reduce 0%
17/07/18 23:05:11 INFO mapreduce.Job: map 100% reduce 100%
17/07/18 23:05:12 INFO mapreduce.Job: Job job 1500344813707_0002 completed successfully
17/07/18 23:05:12 INFO mapreduce.Job: Counters: 49

The above result shows that Hadoop is installed successfully.



View output directory

hadoop fs -ls /output

View output result

hadoop fs -cat /output/part-r-00000

[root@V	96_24_centos	/usr/hadoop-	2.7.4]#	hadoop	fs -	cat	/output/p	art-r-00	0000
a	5								
dasdada	1								
ds	2								
qwe	1								
ret	1								
S	1								
V	2								
vdfd	1								
wqere	1								

Note:

For more information on how to run Hadoop in stand-alone and pseudo-distributed modes, see Get Started with Hadoop on the official website.

COSBrowser Overview

Last updated : 2022-12-28 11:28:59

COSBrowser is a visualization interface tool provided by Tencent Cloud COS. You can use it to view, transfer, and manage COS resources easily. Currently, it is available for desktop and mobile clients as described below. If you need to migrate or batch upload data, you can use Migration Service Platform (MSP).

- User Guide for Desktop Version
- User Guide for Mobile Version

Download Address

COSBrowser	OS	System Requirements	Download Address
	Windows	Windows 7 32/64-bit or later, Windows Server 2008 R2 64- bit or later	Windows
	macOS	macOS 10.13 or later	macOS
Desktop	Linux	Includes a GUI that supports the AppImage format Note: To launch a client that runs CentOS, you need to run ./cosbrowser.AppImage no-sandbox in the terminal.	Linux
Mobile Version	Android	Android 4.4 or later	Android
	iOS	iOS 11 or above	iOS
Web	Web	Browsers such as Chrome, Firefox, Safari, and Internet Explorer 10+	Web
Uploader Plugin	Web	Chrome browsers	Web Store/Offline Download

COSBrowser Desktop Version

COSBrowser Desktop Version focuses on resource management and uploading and downloading data in batches.

Note :

COSBrowser Desktop Version uses the system-configured proxy to connect to the internet. Make sure that your proxy is set up properly, or you can disable the proxy configuration if it fails to connect to the internet.

- For queries on Windows, go to Internet Options.
- For queries on macOS, go to Network Preferences.
- For queries on Linux, go to System Settings > Network > Network Proxy.

COSBrowser Desktop Version has the following features:

Feature	Description
Creating/Deleting a bucket	Creates or deletes a bucket
Viewing bucket details	Views the basic information of your bucket
Viewing statistics	Views the current storage capacity and number of objects in your bucket
Permission management	Modifies the permissions on your buckets and objects
Setting versioning	Enables/Suspends bucket versioning
Adding an access path	Adds an access path
Uploading files/folders	Uploads files/folders to a bucket separately, in batches, or incrementally.
Downloading files/folders	Downloads files/folders to the local file system separately, in batches, or incrementally.
Deleting a file/folder	Deletes files/folders from a bucket separately or in batches
Synchronizing files	Synchronizes local files to your bucket in real time
Copying and pasting files	Copies files/folders separately or in batches from one directory to another
Renaming files	Renames files in a bucket
Creating a folder	Creates a folder in a bucket
Viewing file details	Views the basic information of the files in a bucket



Feature	Description	
Generating a file link	Generates a file access link with a certain validity period by requesting a temporary signature	
Sharing a file/folder	Shares a file/folder and sets a validity period for the sharing	
Exporting file URLs	Exports file URLs in batches	
Previewing a file	Previews media files (images, video, and audio) in your bucket	
Searching a file	Searches files in a bucket through prefix search	
Searching buckets	Searches existing buckets	
Viewing file versions/incomplete multipart uploads	 Views multiple versions of a file in a versioning-enabled bucket Views the incomplete multipart uploads in your bucket 	
Comparing files	Compares files in a local folder to those in a bucket	
Transcoding a video	Transcodes videos with the media processing feature enabled in a bucket	
Generating an authorization code	Generates an authorization code for logging in to the COSBrowser client	
Processing an image	Scales, crops, or rotates an image, or adds text or image watermarks, and generates a URL of the output image	
Setting up a proxy	Sets up a proxy to access COS	
Setting the number of concurrent uploads/downloads	Sets the number of concurrent transfers for file upload or download	
Setting the number of parts to upload/download	Sets the number of parts for multipart upload or download	
Setting the number of retries upon upload/download failure	Sets the number of retries upon upload or download failure	
Limiting single-thread upload/download speed	Limits the upload and download speeds for a single thread	
Setting upload check	Double-checks files uploaded to a bucket	
Viewing a local log	Saves the record of operations on COSBrowser in the form of a local log	

COSBrowser Mobile Version

The COSBrowser mobile version is mainly used to monitor COS resources (such as storage usage and traffic) at any time you want. For more supported features, see basic features in User Guide for Mobile Version.

Changelog

- Desktop Version changelog: changelog.
- Mobile Version changelog: changelog_mobile.

Feedback and Suggestions

If you have any questions or suggestions during your use of COSBrowser, feel free to give us your feedback:

- Feedback on Desktop Version: issues.
- Feedback on Mobile Version: issues_mobile.

User Guide for Desktop Version

Last updated : 2022-08-01 16:23:58

Download and Installation

Downloads

OS	OS Requirement	Download Link
Windows	Windows 7 32/64-bit or later, Windows Server 2008 R2 64-bit or later	Windows
macOS	macOS 10.13 or later	macOS
Linux	Includes GUI and supports the AppImage format	Linux
Web	Browsers such as Chrome, Firefox, Safari, and Internet Explorer 10+	Web
Uploader extension	Chrome	Chrome Web Store/Download

Installations

You can install the tool by using the installer or decompressing the installation package for your platform. You can also directly use it in browsers without installation.

Note:

```
To launch the client running CentOS, you need to run ./cosbrowser.AppImage --no-sandbox in the terminal.
```

Login

You can log in to the COSBrowser desktop version using a permanent key, Tencent Cloud account, or a shared link. You can log in to an account on multiple devices at the same time.

Login with a permanent key

You can log in using your Tencent Cloud API key (SecretID and SecretKey), which can be created and obtained at Manage API Key in the CAM console. After you logged in successfully, the key will be saved to **Historical Sessions**. The login page and configuration items are as follows:

- Bucket/Access Path: If the root account allows you to access only a specific bucket or directory with the currently used key, then this configuration item is required. After setting it, you can quickly enter the corresponding file path. The path format is Bucket or Bucket/Object-prefix; for example, if you are allowed to access only the doc folder in the examplebucket-125000000 bucket with the current key, enter examplebucket-1250000000 bucket with the current key, enter examplebucket-
- **Description**: Description of the permanent key entered, such as the operator and the usage. The description can be used to distinguish different SecretID when you manage the historical sessions on the historical key page.
- Remember Session:
 - If this box is not checked, the Tencent Cloud API key entered will be cleared when you log out (if the key has been saved to the historical sessions, it will be removed).
 - If this box is checked, the Tencent Cloud API key entered will be remembered and can be managed in the historical sessions.

Note :

You cannot log in to COSBrowser with a project key.



COSBrowser	- ×
	Key Login Advanced Setting > SecretID AKID7wXsRnJIAl8G7hvzMSHsDH4v
	SecretKey Buckets/Access Path (i) examplebucket-125000000/test/ Region ap-chengdu
	Remark Not required, Add remark Login Get SecretKey History key Local Logs

Login with a Tencent Cloud account

Click **Login with Tencent Cloud Account** and use your Tencent Cloud account to log in to COSBrowser Desktop Version in the pop-up window. You can log in to your Tencent Cloud account via WeChat, email, QQ, WeChat Official Account, WeCom, or sub-user. For detailed directions, see Signing Up.

Login with a shared link

You can log in to COSBrowser Desktop Version temporarily through the **shared link** and **password** forwarded or shared by another user. For more information, see **Sharing file/folder**.

Basic Features

Note:

COSBrowser for Windows v2.8.4 is used as an example here. For other versions, see Changelog.

1. Creating/Deleting bucket



Feature	Description	Directions
Creating bucket	You can create a bucket directly via the client	 In the bucket list, click Add Bucket in the top-left corner Enter the bucket name, and select the region and access permission Click OK
Deleting bucket	Before you delete a bucket, make sure that all data in the bucket has been cleared	 In the bucket list, click Delete on the right of the target bucket Click OK

2. Viewing bucket details

You can view bucket details by clicking **Details** on the right of the bucket list. Details include bucket name, region, access permissions, and domain name, as well as versioning and global acceleration status.

3. Viewing statistics

You can click ... > Statistics to view the bucket statistics, including the storage usage and the number of objects.

4. Managing permissions

You can use COSBrowser to manage permissions for buckets and objects.

- Bucket permissions: Click Permission Management in the Operation column on the right of the bucket list.
- Object permissions: Click **Permission Management** in the **Operation** column on the right of the object.

Note : For more information about COS permissions, see ACL.

5. Setting versioning

You can use COSBrowser to enable/disable versioning for a bucket.

Click **Details** in the **Operation** column on the right of the bucket list. In the bucket details pop-up window, click the versioning icon.

Note :

For more information on versioning, see Overview.



6. Adding an access path

If you log in with a sub-account that does not have permission to access the bucket list, you can initiate an access via **Add Access Path** in the following two ways:

(1) Add an access path directly on the login page and select the corresponding bucket region. Once you log in, you can manage your resources.

COSBrowser	- ×
	Key Login Advanced Setting > SecretID AKID7wXsRnJIAl8G7hvzMSHsDH4v
	SecretKey
	Buckets/Access Path () examplebucket-1250000000/test/ Region
	ap-chengdu Remark Not required, Add remark
	Login Get SecretKey History key Local Logs

(2) Log in with your sub-account, click Add Path in the top-left corner of the bucket list page, and enter a specified





path to enter the bucket and manage its resources.

COSBrowser	
Bucket List	
Add Bucket Add Path	
Basic Info	

7. Uploading file/folder

Upload Feature	Remarks	Directions
Uploading files	COSBrowser allows you to upload a single file or multiple files in batches in different ways.	You can upload files in the following ways. In the spe 1. Click Upload Files to upload files directly. 2. Right-click in the blank space of the file list and se 3. Drag a file to the file list pane.
Uploading a folder and the files contained	If the name of the file/folder to upload already exists in the bucket or path, the old file/folder will be	You can upload a folder in the following ways. In the 1. Click Upload Folder to upload a folder directly. 2. Right-click in the blank space of the file list and se 3. Drag a folder to the file list pane.

	overwritten by default.	
Incremental upload	Incremental upload compares the files to upload with existing files in the bucket before the upload. An existing file with the same name in the bucket will be skipped.	You can perform the following two steps to use incre or path: 1. Upload as you do with a folder and click Next . 2. In Storage method , select Skip . Then, click Upl

Note:

If you need to upload files in batches, we recommend you use a computer with a 4-core CPU and 16 GB RAM, which allows you to upload up to 300,000 files at a time.

8. Downloading file/folder

Download Feature	Remarks	Directions
Downloading files	COSBrowser allows you to download a single file or multiple files in batches in different ways.	You can download a file in the following ways: 1. Select the desired file and click Download in the L 2. Right-click the file and select Download . 3. Drag the file to the local file system.
Downloading a folder and files contained	If the name of the file/folder to download already exists in the local file	 You can download a folder and the files contained in the following ways: 1. Select the desired folder and click Download in th UI. 2. Right-click the folder and select Download. 3. Drag a folder to the local file system.

	system, it will be renamed by default.	
Incremental download	Incremental download compares the files to download with local files before the download. An existing file with the same name will be skipped.	You can perform the following steps to use incremental download: 1. Right-click the desired file or folder. 2. Click Advanced Download and select Skip in the pop-up window. 3. Click Download Now to download the incrementa unique file/folder.

Note:

If you need to download files in batches, we recommend you use a computer with a 4-core CPU and 16 GB RAM, which allows you to upload up to 300,000 files at a time.

9. Deleting file/folder

Select the desired file/folder and click **More** > **Delete** at the top of the page or right-click the file/folder and select **Delete** to delete it. Batch deletion is supported.

10. Synchronizing file

The file sync feature allows you to upload specified files in your local folders to a bucket in real time or as scheduled. Detailed directions are as follows:

- 1. Click **Toolbox** > **File Sync** in the top-right corner of the page.
- 2. In the pop-up window, specify the local folder to be synced and the destination COS bucket directory.
- 3. You can select one-time, automatic, or scheduled sync as the sync type and click **Start Sync** to enable file sync.



4. You can view the file sync history in sync logs.

COSBrowser		1 0 4	×
Bucket List	Sync Setting		
Add Bucket Add Path	Local Folder * Select Local Folder Change	C	
Basic Info	Bucket Path * Select bucket Bucket Path		
examplebucket-1254 examplebucket-1254	Auto Sync		
	Support incremental upload only. See the documentation for details. Start Sync Clear Local Cache		

Note :

- Synchronization means that when the file is uploaded, the system automatically identifies whether the same file exists in the bucket. Only files that do not exist in the bucket are synchronously uploaded.
- Currently, only synchronizing local files to the bucket is supported. Reverse operation is not supported.
- The file sync feature supports manual, automatic, and scheduled sync.

11. Copying and pasting file

To copy a file/folder, select the desired file/folder in the specified bucket/path and click **Copy** in **More** at the top of the UI. Alternatively, you can right-click it and select **Copy**. After the file/folder is successfully copied, you can paste it to **another bucket or path**. You can copy and paste multiple files/folders in batches.

Note:

If the name of the file/folder to paste already exists in the destination path, the old file/folder will be overwritten by default.

12. Renaming file



Select a file you want to rename, right-click, and select **Rename**. Alternatively, you can click **Rename** in **More Actions** on the right of the file. Then, enter your file name, and click **OK**.

Note : Folders cannot be renamed.

13. Creating folder

To create a folder in the specified bucket or path, click **Create Folder** in the UI, or right-click in the UI and select **Create Folder**, enter the folder name, and click **OK**.

Note:

- The folder name can contain up to 255 characters, including digits, letters, and visible characters.
- ... cannot be used as the folder name.
- The folder cannot be renamed.

14. Viewing file details

To view the details of a file, click its filename or right-click it and select **Detail**. File details include filename, file size, modification time, access permission, storage class, ETag, headers, specified domain name, object location, and an option to create a temporary object URL.

Detail		×
Name	exampleobject.txt 📭	
Size	500KB (512,000B)	
Last Modified	2019-09-29 12:11:38	
ACL	Inherit 🧨	
Storage Class	Standard Storage 🎤	
ETag	"c939165a4566ac3eba011f641e94c519"	
Headers	content-type: text/plain; charset=utf-8 x-cos-metadata-directive: Replaced	
Object location	https://examplebucket-125cos.ap- chengdu.myqcloud.com/exampleobject.txt 🗗	
Create temporary object url	Create temporary object url	
	Close	•

15. Generating file URL

Each file stored in COS can be accessed through a specific URL. If a file is private-read, you can request a temporary signature to generate a temporary access URL with a certain validity period.

You can generate file links in the following ways:

- In the list view, click the Share icon on the right of the object to generate a URL and copy it. If the file is public-read, the URL will not carry a signature and be valid permanently. If the file is private-read, the URL will carry a signature and be valid for 2 hours.
- Right-click a file and select **Copy Link** to generate a URL and copy it. If the file is public-read, the URL will not carry a signature and be valid permanently. If the file is private-read, the URL will carry a signature and be valid for two hours.
- On the file detail page, click **Create link with expires** to set the temporary URL, URL type, and validity period for the specified domain name (available only when a CDN acceleration domain name is enabled).

16. Sharing file/folder

Click **Share** in the **Operation** column or the context menu to share a COS folder. You can also set a validity period for the link.

🔗 Tencent Cloud

Note:

- You can only share a single folder but not multiple files.
- If multiple users share a folder, the file content may be hard to manage. In this case, we recommend you enable versioning for your bucket so that you can roll back to the desired historical version.

Share Folder	s/Files ×
Name	doc 1 folders/files.
Permissions	O Read only O Read & write
Valid time	2 hours 🔻
	The longest valid time is 2 hours (If log in with a sub-ac count, the maximum validity period can be extended to 1.5 days)
Extraction code	Auto generate Custom ee9b31
	Please enter the 6-digit extraction code
	OK Close

Parameter	Description
Permission	 You can set the access permission for the shared folder. Read only: Pulls the folder list and downloads files in the folder from the access URL. Can edit: Pulls the folder list and downloads files in the folder, uploads files to the folder, and creates folders using the access URL.
Validity Period	 In minutes, hours, or days. If you log in to the client using a key, the validity period can be 1 minute to 2 hours for the root account, and 1 minute to 1.5 days for sub-accounts. If you log in to the client using a Tencent Cloud account, the longest validity period is 2 hours. The default value is the longest validity period allowed for the current user.



Parameter	Description
Password	A 6-character password automatically generated by the system. You can customize one as needed (numbers, letters, and symbols are supported).

Note:

When the link is valid, users receiving the link and password can access the folder.

17. Exporting file link

COSBrowser allows you to export file links. In the top-right corner of the UI, click the toolbox icon, select **Export File** Link in the toolbox pop-up window, select the bucket, folder path (for example, to export the folder folder under the root path, enter folder/), and export destination path of the target files, and click **Export**.

18. Previewing file

COSBrowser allows you to preview media files, including images, videos, and audio. To preview a media file, doubleclick it or right-click it and select **Preview** or **Playback** in the context menu. On the file preview or playback page, you can click:

- Copy Link to generate a file access URL and copy it.
- **Download** to download the file to the local file system. If a file with the same name already exists in the local file system, it will be overwritten by default.
- View on phone to generate a QR code for the file, which can be scanned on a mobile phone for direct view.

Note :

- Preview is available for images in most formats, .mp4 and .webm videos, and .mp3. and .wav audios.
- Note that file preview will incur downstream traffic.




19. Searching for File

To search for a file, enter the filename in the search box at the top right of the bucket. COSBrowser supports prefix search and fuzzy search.

20. Searching for Bucket

To quickly locate a bucket, enter the bucket name in the search box above the bucket list on the left.

21. Viewing historical versions or incomplete multipart uploads

If versioning is enabled for your bucket, you can click View > Versions above the file list to view the historical versions. Prefix search and clearing all historical versions (retaining the latest version only) are supported.

Prefix matching				
Name	Size	Last Modified	VersionId	Action
• index.html	169.71KB	2019-09-19 15:47	-	上 前
		No history files		
video.mp4	169.71KB	2019-09-19 17:40		上 前
		No history files		
exampleobject.txt	500KB	2019-09-29 12:11	-	上 前
		No history files		
🔹 image.png	9.72KB	2019-11-06 16:01	-	上 前
		No history files		
 manifest.csv 	169.71KB	2019-11-27 18:27	-	上 前

If you pause or cancel an ongoing upload, incomplete multipart uploads may be generated. You can click View >
 Incomplete Multipart Uploads above the file list to view them. Prefix search and clearing all incomplete multipart

uploads are supported.

Clear Incomplete Multipart Uploads			Q Prefix n	natching
Multipart Upload Name	Upload Task ID	Storag Cre	eate time	Action
	This list is em	pty		
	There is no incomplete	upload.		
	Cancel			

22. Comparing files

In the top-right corner of the UI, click the toolbox icon, select **File Comparison** in the toolbox pop-up window, select the local folder and the bucket (you need to select the region, bucket, and directory) for comparison, and click **Start Comparison**.

23. Generating authorization code

You can generate authorization codes to temporarily authorize the specified buckets, resources in a bucket, and operations. It is more flexible than folder sharing, as it can grant custom operation permissions for specified directories. You can generate temporary <code>SecretId</code>, <code>SecretKey</code>, token, and authorization code for other users to log in to the client temporarily.

The operations are as detailed below:

In the top-right corner of the UI, click the toolbox icon, select **Generate Authorization Code** in the toolbox pop-up window, select the bucket and scope of resources for authorization, select the authorized operations such as read/write in the policy permission settings, set the authorization code validity period, and click **OK**.

24. Processing image

The image processing feature of COSBrowser supports basic image processing operations, such as scaling, cropping, rotation, and text and image watermarking. It can also generate processed image links.

Select the target bucket, click the toolbox icon in the top-right corner of the UI, and select **Image Processing** in the toolbox pop-up window. In the image processing pop-up window, select the target image file, configure the feature parameters, and click **Image Preview** to generate the link of the output image.

Settings

System Feature	Remarks	Directions
Setting up proxy	COSBrowser uses the system-configured proxy to connect to the Internet. Make sure that your proxy is set up properly or disable the proxy configuration if it fails to connect to the internet.	1. Select Settings > Pro 2. Set up a proxy to conn
Setting the maximum number of concurrently uploaded/downloaded files	COSBrowser allows you to set the maximum number of concurrently uploaded and downloaded files.	 Select Settings > Dov Set the maximum num transferred files, which is
Setting the maximum number of concurrently uploaded/downloaded parts	COSBrowser supports uploading/downloading a file in multiple parts. When the file to be transferred exceeds a certain size, multipart transfer will be performed by default.	 Select Settings > Dov Set the maximum num transferred parts, which i
Setting the number of retries upon upload/download failure	COSBrowser will retry failed jobs by default when transferring files.	 Select Settings > Dov Set the number of retriwhich is 5 by default.
Setting the single-thread upload/download speed limit	COSBrowser supports limiting the upload and download speeds for a single thread.	 Select Settings > Upl Set the single-thread u limit in MB/s.



	Total upload (download) speed limit = single-thread upload (download) speed limit x number of concurrent files x number of concurrent parts	
Setting upload check	COSBrowser supports checking files online after upload to verify whether their size and status are correct.	 Select Settings > Upl Select Perform secon upload completion.
Viewing local log	COSBrowser will record all the performed operations in the `cosbrowser.log` local file.	1. Select Settings > Abc 2. Click Local Logs to en

User Guide for Mobile Version Installation and Login

Last updated : 2022-12-19 12:17:36

Download and Installation

Downloading software

OS	System Requirements	Download Address
Android	Android 4.4 or Later	Android
iOS	iOS 11 and later	iOS

Installation

COSBrowser Mobile Version is currently available in most app platforms such as MyApp and App Store. You can download it from the above download address or in an app platform.

Login Options

COSBrowser Mobile Version supports the following login options:

- Login with email: If your Tencent Cloud account was created through email or associated with a specific email address, you can log in to COSBrowser by entering the email address and password.
- Login with permanent key: You can log in using your TencentCloud API key (SecretId and SecretKey; project key is not supported), which can be created or obtained on the API Key Management page in the CAM console. After successful login, the account will be kept logged in permanently.

Note:

• Sub-accounts can log in with a key.

Mobile Version Features

Last updated : 2022-06-09 12:27:50

COSBrowser mobile app allows you to easily view and manage COS resources anytime you want. Supported operations are as described below.

Data Monitoring

Operation	Description
Usage overview	You can view the recent data usage.
Bucket monitoring	You can view the recent data usage by bucket.
Widget	You can view the data usage without opening the app.

Bucket Management and Operations

Operation	Description
Viewing bucket list	You can view buckets by region.
Adding access path	If you log in with a sub-account that does not have permission to access the bucket list, you can initiate an access request via Add Access Path .
Bucket creation	You can create buckets.
Bucket search	You can fuzzy search for buckets by keyword in the bucket list.
Viewing bucket's basic information	You can view the basic information of buckets, such as name, region, and creation time on Mobile Version.
Bucket permission management	You can modify the public and user permissions of buckets.
Enabling global acceleration	You can enable the global acceleration feature for buckets.
Bucket transfer configuration	You can set the domain names for uploads and downloads.

Object Management and Operations

Operation	Description
Folder creation	You can create folders in buckets.
Folder deletion	You can delete all files in the current directory and all sub-directories.
File upload	You can use COSBrowser Mobile Version to upload local and remote files as well as files from other apps and file managers to COS. You can also set file information such as storage class, access permissions, encryption method, object tags, and metadata during upload.
File backup	COSBrowser provides the automatic backup feature. After you enable backup, it will automatically back up files in your album to the specified bucket.
File download	You can download files in buckets to the app or save them to the local album.
Batch operation	You can batch upload, download, delete, copy, and move files in buckets.
File preview	You can preview images, videos, audios, documents, and files in many different formats in buckets.
Online file decompression	You can decompress .zip, .tar, and .gz packages online.
File sharing	COSBrowser provides the folder and file sharing feature for you to quickly collect or share data in buckets to other users.
File renaming	You can rename files in buckets.
File search	You can fuzzy search for files by file type, such as folder, image, video, document, and audio.
Object sorting or filtering	You can sort files in buckets by filename, size, and modification time and filter them by storage class.
File permission management	You can set file access permissions, which have a higher priority than those for buckets.

Bucket Management and Operations

Last updated : 2022-05-31 15:12:34

Viewing Bucket List

COSBrowser Mobile Version displays bucket lists by region. You can view buckets by region and click **Resources** at the bottom to view created buckets.

Adding Access Path

If you log in with a sub-account that has no access to the bucket list, you can add the specified path to a bucket or directory to manage resources by clicking **Add Access Path** in the top-right corner of the bucket list page.

Note :

This feature is supported only for sub-accounts.

Directions

- 1. Go to the bucket list page and click + in the top-right corner.
- 2. In the pop-up operation list, click Add Access Path and enter an access path authorized by the root account.

Creating Bucket

You can create a bucket by specifying the bucket name, region, and access permissions on COSBrowser Mobile Version.

Directions

- 1. In the bucket list, tap + in the top-right corner.
- 2. In the pop-up operation list, tap Create Bucket.
- 3. On the bucket creation page, configure the following information:

🕗 Tencent Cloud

- Name: Enter a custom bucket name, which cannot be modified once configured. For naming instructions, see Bucket Overview.
- Region: Select a COS region corresponding to the physical location where your business or users are distributed. This parameter cannot be modified once configured. For more information on regions, see Regions and Access Endpoints.
- Access Permission: Three access permissions are available for buckets by default: "Private Read/Write", "Public Read/Private Write", and "Public Read/Write". You can modify this parameter if needed. For more information, see Setting Access Permission.
- 4. After confirming that everything is correct, tap OK.On the bucket list page, you can view the newly created bucket.

Searching for Bucket

If you have many buckets, you can enter a bucket name in the search box at the top of the bucket list page to fuzzy search for buckets.

Viewing Bucket's Basic Information

- 1. On the bucket list page, tap ... on the right of the target bucket.
- 2. In the pop-up operation list, tap **Details** to view the bucket's basic information. The basic information includes bucket name, region, and creation time.

Managing Bucket Permission

You can set or modify bucket access permissions of the following two types on COSBrowser Mobile Version:

- **Public permissions**: Include private read/write, public read/private write and public read/write. For more information, see **Types of Permission** in **Bucket Overview**.
- User ACLs: The root account has all bucket permissions (full control) by default. You can add sub-accounts and grant them permissions including data read/write, ACL read/write, and even full control.
- 1. Go to the bucket list page and tap ... on the right of the target bucket.
- 2. In the pop-up operation list, tap **Permission** to enter the bucket permission page.

Modifying public permission

On the bucket permission page, tap a public permission configuration item to modify it.

Setting user permission

On the bucket permission page, tap **Add User** to set bucket access permissions.

Editing or deleting bucket permission

Select the target user permission, swipe left, and click the displayed **Edit** or **Delete** button to edit or delete the user permission.

Enabling Global Acceleration

You can enable the global acceleration feature for your bucket on COSBrowser Mobile Version. It allows users across the world to quickly access the bucket and improves your business access success rate and stability. It can accelerate both uploads and downloads.

Note:

Enabling global acceleration will not affect the existing default bucket domain name, which can still be used.

Directions

- 1. Go to the bucket list page and tap ... on the right of the target bucket.
- 2. In the pop-up operation list, tap **Transfer** to enter the transfer list page.
- 3. Tap the global acceleration configuration item to enable or disable it as needed.

Configuring Bucket Transfer

You can enable data transfer via a custom domain name for a bucket on the Transfer list page.

Setting upload domain name

You can specify the domain name for file upload to a bucket, which will be used first for file uploads once configured.

Setting download domain name

You can specify the domain name for file download from a bucket, which will be used first for file downloads once configured.

File Management and Operations

Last updated : 2022-05-31 15:12:34

Creating Folder

COSBrowser Mobile Version allows you to create a folder in a bucket as follows:

- 1. On the bucket list or file list page, click + in the top-right corner.
- 2. In the displayed operation list, click Create Folder.
- 3. On the Create Folder page, enter a folder name and click OK.

Deleting Folder

Note:

Deleting a folder will delete all files under it and its sub-directories.

Directions

- 1. Click ... on the right of the target folder.
- 2. In the displayed operation list, click **Delete**.

Uploading File

You can use COSBrowser Mobile Version to upload local and remote files and files from other apps or the file manager to COS and set information such as file storage class, access permission, encryption method, object tags, and metadata during upload. You can enter the operation list in the following two methods:

- On the bucket list page, click ... in the top-right corner.
- On the file list page, click ... in the top-right corner.

Uploading image/video

COSBrowser allows you to batch upload images or videos from your album to COS.

Directions

- 1. In the operation list, click **Upload Images**.
- 2. In the file list of the displayed album, select the target files and click Next.
- 3. (Optional) Configure the upload parameters. COSBrowser allows you to set the following file attributes during upload:
- Storage Class: Select a storage class for your object based on your business scenario. It is set to STANDARD by default. For more information, see Overview.
- Access Permission: Select the access permission for your object as needed. This field is set to Inherit by default (i.e., inheriting permissions of the bucket). For more information, see Basic Concepts of Access Control.
- Server-Side Encryption: Configure server-side encryption for the object you want to upload. COS will
 automatically encrypt your data as it is written and decrypt it when you access it. Currently, COS offers two
 encryption methods: SSE-KMS (only available in the Beijing, Shanghai, and Guangzhou regions) and SSE-COS.
 For more information, see Server-Side Encryption Overview.
- Object Tag: An object tag is composed of a tag key, equal sign (=), and a tag value, for example, group =
 IT . You can set, query, and delete tags of a specified object.
- Metadata: Object metadata, or HTTP header, is a string sent by the server over HTTP before it sends HTML data to the browser. By modifying HTTP headers, you can modify how the webpage responds as well as certain configurations, such as caching time. Modifying an object's HTTP headers does not modify the object itself. For more information, see Custom Headers.

4. Click Upload.

Uploading file through link

COSBrowser allows you to upload a file through the file link. Every time you enter the app, it will check the current clipboard. If there is any valid file link on the clipboard, a message will pop up asking you whether to upload the file through the link. In this case, you simply need to click **Upload Now**.

You can also upload a file through the link as follows:

- 1. On the bucket list or file list page, click + in the top-right corner to display the operation list.
- 2. Click **Upload Link**, paste the link for file upload into the text box, select an upload path, and click **Upload**.

Uploading file shared by third-party app

🕗 Tencent Cloud

You can also share a file from another app to COSBrowser to upload it.

Note :

This feature requires that the third-party app supports file sharing to other apps.

Directions

Take QQ as an example:

1. In QQ, click a file to preview it, click ... in the top-right corner, and select Other Apps.

2. Click COSBrowser in the app list.

Uploading file from file manager

COSBrowser can upload files from the system file manager ("Files" on iOS and the corresponding system file manager app on Android) to COS.

Directions

- 1. On the bucket list or file list page, click +.
- 2. In the displayed operation list, click Upload Files.
- 3. On the displayed file manager page, click the target file.

Backing up File

COSBrowser provides the auto backup feature. After this feature is enabled, COSBrowser will automatically back up the files in your album to the specified bucket. To help you manage backup files, COSBrowser displays the backup data as a separate module on the homepage.

Note :

- The album backup feature is only supported for the root account.
- The album module displays only image and video files. To view all files, go to the bucket list and search for the backup bucket.

Setting backup

Go to Personal > Album Backup and toggle on Automatic Photo Backup or Automatic Video Backup.

- Automatic Photo Backup: After it is enabled, all images in the album will be backed up.
- Automatic Video Backup: After it is enabled, all videos in the album will be backed up.
- Back up over Wi-Fi Only: After it is enabled, files will be backed up only over a Wi-Fi network.
- **Region**: After the backup region is selected, a bucket named from-phone-date-APPID will be created in the region by default.

Note :

The region can be set only before backup is enabled and cannot be modified after the settings are saved.

 Enable Smart Storage: After it is enabled, files in the album will be uploaded to COS in INTELLIGENT TIERING. COS will automatically switch between the STANDARD and STANDARD_IA classes based on the access frequency of INTELLIGENT TIERING objects with no data retrieval fees incurred, which reduces your storage costs. For more information, see INTELLIGENT TIERING Overview.

Managing backup file

COSBrowser supports batch file upload and download.

Directions

- 1. On the **Album** page, click the batch operation icon in the top-right corner to display operation buttons.
- 2. Select the target files and click **Download** or **Delete** to download or delete the files.

Adding file to backup bucket

You can also add image or video files from other buckets to the backup bucket to quickly preview them in the **Album** module.

Directions

- 1. Click ... on the right of the target image or video file.
- 2. In the displayed operation list, click Add to Album.
- 3. Enter the Album module again to view the file.

Downloading File

COSBrowser allows you to download files from COS to your local file system. It also enables you to save images to the system album.

Downloading to app

COSBrowser provides multiple download entries for you to download files anytime, anywhere.

• Method 1:

1. On the file list page, click ... on the right of the target file.

- 2. In the displayed operation list, click **Add to Download List**.
- Method 2:

On the File Details page, click In the displayed operation list, click Download.

• Method 2:

On the file preview page, click ... in the top-right corner. In the displayed operation list, click **Download**.

Saving to album

COSBrowser allows you to save images to the local album as follows:

- 1. On the file list page, click ... on the right of the target file to display the operation list.
- 2. Click Download and select Save to Album.

Batch Operation

COSBrowser allows you to batch download, delete, copy, and move files from a bucket.

Batch download

- 1. Click a bucket to enter the file list page and click ... in the top-right corner.
- 2. In the displayed operation list, click **Batch Operation**.
- Batch select files and click **Download** on the bottom operation bar.
 You can also click the **Transfer** button in the top-right corner of the file list page to view the jobs on the transfer list page.

Deleting domain names by batches

1. Click a bucket to enter the file list page and click ... in the top-right corner.

- 2. In the displayed operation list, click **Batch Operation**.
- 3. Batch select files and click **More** > **Delete** on the bottom operation bar.

Batch copy

Note:

When you use COSBrowser to copy a source file, its information such as ACLs, policies, and tags will also be copied.

- 1. Click a bucket to enter the file list page and click ... in the top-right corner.
- 2. In the displayed operation list, click **Batch Operation**.
- 3. Batch select files and click **More** > **Copy** on the bottom operation bar.

Batch moving

Note:

When you use COSBrowser to move a source file, its information such as ACLs, policies, and tags will also be moved.

- 1. Click a bucket to enter the file list page and click ... in the top-right corner.
- 2. In the displayed operation list, click **Batch Operation**.
- 3. Batch select files and click **More** > **Move** on the bottom operation bar.

Previewing File

COSBrowser allows you to preview various file formats such as image, audio, video, and document. It also supports online file decompression.

Online image preview

COSBrowser allows you to preview images online without downloading them.

©2013-2022 Tencent Cloud. All rights reserved.

Enabling image preview

You can preview images in the following three methods:

1. Enable the feature globally

Select **Personal** > **Settings** and toggle on **Image Preview**.

The effect after this feature is enabled is as shown on the right:

2. Enable the feature temporarily

When you launch the app for the first time and enter the file list page, the app will check whether the current list page contains images, and if yes, a pop-up window will appear asking you whether to enable image preview. You can click **Enable Preview** to quickly enable the **image preview** feature. If you don't need this feature, you can click **Cancel**. If you want to use it in the future, you can manually enable it in **Personal** > **Settings**.

3. Enable the feature for one image

If you have disabled image preview in **Personal** > **Settings**, and it's not the first time that you open the app, when you enter the image details page, the app will ask you whether to enable preview. In this case, you can click **Click to Preview** to preview a file.

Note :

This switch doesn't change the status of the global **image preview** toggle.

Big image mode

COSBrowser provides the big image mode feature for you to view image details more clearly.

1. On the file list page, click the target image to enter the **File Details** page.

2. Click the image to enter the big image mode, in which you can zoom in the image with two fingers to view its details.



Playing back audio

COSBrowser can play back audio files in MP3, OGG, AAC, WMA, WAV, APE, or FLAG format.

Directions

1. On the file list page, click an audio file to enter the **File Details** page.

2. Click the playback button.

You can speed up or pause the playback.

Playing back video

COSBrowser provides a simple online video playback feature, which supports various formats such as AVI, WMV, MPEG, RM, RMVB, MKV, MOV, QT, and MP4. You can speed up or pause the video playback during watch. You can also play back the video in a floating window after the app enters the background.

Directions

- 1. On the file list page, click the target video file to enter the **File Details** page.
- 2. Click the playback button.

You can speed up or pause the playback.

Note:

After the app enters the background, you can still watch the video in a floating window.

Previewing document

COSBrowser allows you to preview files in multiple formats such as PDF.

Directions

- 1. Click the target document to enter the **File Details** page. If you haven't enabled the document preview feature, click **Click to Enable**.
- 2. On the document preview feature configuration page, click **Enable Now**.
- 3. Return to the previous page and click **Click to Preview**.
- 4. On the document preview page, swipe left to view the next page.

Decompressing File

COSBrowser allows you to decompress files in ZIP, TAR, or GZ format online. The extracted files will be stored in the current directory.

1. On the file list page, click ... on the right of a compressed file.

- 2. In the displayed operation list, click **Online Decompression**.
- 3. After the file is successfully decompressed, the extracted files will be displayed in the current directory.

Sharing File

COSBrowser provides the folder and file sharing feature for you to quickly collect data or share the data in a bucket with other users.

Sharing folder

Note:

- You can share only one folder but not multiple files at a time.
- A sharing duration of up to two hours and up to 1.5 days can be set for the root account and a sub-account respectively.
- If multiple users share a folder, the file content may be hard to manage. In this case, we recommend you enable versioning for your bucket, so that you can rewind to the desired historical version.

Generating sharing QR code/link

The detailed steps are as follows:

- 1. Click ... on the right of a folder and click **Share** in the displayed operation list.
- 2. On the displayed sharing page, you can select QR code or link for sharing.
- 3. (Optional) Configure the sharing parameters, which are as detailed below. You can skip this step to keep the parameters as default.
- **Permission**: You can set the access permission for the folder to be shared.
 - Read-only: Pull the file list through the access URL and download files in the folder.
 - Read/Write: Pull the file list through the access URL, download files in the folder, upload files to the folder, and create a folder.
- Validity Period: It is in minutes, hours, or days. The validity period for the root account and a sub-account is two hours and 24 hours respectively by default, which cannot be changed.
- **Password**: It can contain six digits, letters, and special symbols. It is automatically generated by the system by default but is customizable.

4. Click Generate Link or QR Code Sharing to generate the sharing link or QR code.

Viewing folder shared by another user

You can open a folder shared by another user from a mobile or desktop client.

Method 1. View on mobile device

- 1. On the **Login** or **Personal** page, click **Scan** to scan the QR code.
- 2. Enter the password and click OK to view the shared folder.

Method 2. View on PC

- 1. Click Log in with Shared Link on the Login page.
- 2. Enter the obtained URL address and password and log in.

Method 3. View in browser

- 1. Open the browser and enter the shared URL to open it.
- 2. Enter the password and click **Extract** to enter the shared folder.

Sharing file

Each file stored in COS can be accessed through a specific URL. You can generate a file URL with the specified domain name if you have set other domain names (such as CDN acceleration domain name and custom origin domain name) for the bucket. If a file is private-read, you can request a temporary signature to generate a temporary access URL with a certain validity period.

Note :

- If you log in with a temporary key, you cannot configure the validity period of the file URL, which is one hour by default.
- If the file is public-read, the URL will not carry a signature and will be valid permanently. If the file is privateread, the URL will carry a temporary signature and will be valid for one hour.

Directions

1. Click a file to enter the **File Details** page.

The configuration items are described as follows:

- Specify Domain: Set the domain name for the URL (optional).
- Validity Period: Set the validity period for the URL (optional).

2. After confirming that the configuration is correct, click **Generate Link**.

3. Send the generated file URL.

Renaming File

Note:

Folders cannot be renamed.

Directions

- 1. Click ... on the right of a file.
- 2. In the file operation list, click Rename.
- 3. In the pop-up window, enter the new filename and click **OK**.

If you select **Overwrite Duplicate File**, the original file will be overwritten; otherwise, a new file will be generated. You can also enable overwriting globally in **Personal** > **Settings** > **Default Upload Options** > **Rename Duplicate File**.

Searching for File

COSBrowser Mobile Version provides the fuzzy search and search by type features. You can use a prefix to search for files with that filename prefix in the current folder and its sub-folders. You can also specify the file type first to search for files in that type.

Searching by keyword

You can enter a search keyword to filter out all filenames containing the keyword in the current folder and all its subfolders.



Search by type

COSBrowser allows you to search for files by type, such as video, folder, audio, document, image, and others.

Directions

Taking folder search as an example:

- 1. Click the search box and then Folder.
- 2. All files (which are folders here) in the folder type in the current directory will be listed.

Sorting or Filtering Objects

COSBrowser allows you to sort and filter files in a bucket.

Note :

Currently, sorting by file name/file size/modification time, and filtering by storage class are supported.

Directions

1. Enter the file list page and click the filter and sort button on the right of the search box.

2. In the operation list, sort or filter files in the bucket.

Managing File Permissions

COSBrowser allows you to set access permissions for files, which have a higher priority than those for buckets.

Note:

- Object access permissions take effect only when the access is made via the default endpoint. For any access made via a CDN acceleration domain name or a custom endpoint, bucket access permissions will take effect.
- There is a limit on the number of ACLs. For more information, see Specifications and Limits.

Modifying public permission

- 1. Click the target file to enter the **File Details** page.
- 2. Click **Permission Info** at the top to enter the permission list page.
- 3. Click **Public Permission** to modify the access permissions of the file.

Setting user permission

- 1. Click the target file to enter the **File Details** page.
- 2. Click **Permission Info** at the top to enter the permission list page.
- Click Public Permission to modify the user permissions of the file.
 You can click Add User to add a user permission. You can also swipe left on a user permission to edit or delete it.

Data Monitoring

Last updated : 2022-05-31 15:12:34

The data overview feature of COSBrowser Mobile Version displays your COS data usage in the past 30 days, including total traffic, total read and write requests, storage trend, traffic trends (for public network downstream traffic, private network traffic, and CDN origin-pull traffic), trend in the number of requests, trend in the ratio of effective requests, amount of STANDARD_IA data retrieved, and amount of archive data retrieved.

Note:

- Current, you cannot view data with a sub-account on Mobile Version. You can log in to a sub-account to view data in the console as instructed in Viewing Statistics.
- The data in this feature isn't real-time but has a delay of about two hours, so it is for reference only. You can view more accurate billing and usage data in the console as instructed in Viewing Statistics.

Usage Overview

COSBrowser provides the storage data overview page, where you can view the number of objects, storage usage, number of requests, and traffic.

Bucket Monitoring

COSBrowser provides data overview by bucket. You can view the bucket monitoring data in the following two ways:

- 1. Tap Home at the bottom and tap User Overview to switch between buckets to view their monitoring data.
- Tap Resources at the bottom and tap More on the right of the target bucket. In the pop-up operation list, tap Monitoring to enter the bucket page.

Widget

COSBrowser can be added to the Home Screen as a widget, so you can view the monitoring data anywhere, anytime with no need to open COSBrowser.

Adding widget

- 1. Download and install COSBrowser for iOS on your iPhone. Then, touch and hold an empty area on the Home Screen until apps jiggle and tap + in the top-left corner.
- 2. Find and tap COSBrowser.
- 3. Select a layout and tap Add Widget at the bottom.

Customizing displayed data

The widget allows you to customize the scope of data to be displayed. You can specify a certain bucket or storage class for tracking and monitoring. The following storage classes can be displayed: STANDARD, STANDARD_IA, and ARCHIVE.

The detailed steps are as follows:

1. Open COSBroswer for iOS and tap **Me** > **Settings** > **Widget Configuration** to enter the widget settings page.

- 2. Set the configuration items as follows:
- Select Bucket: If no bucket is selected, the overview data of all buckets under the current account will be displayed. If a bucket is selected, its overview data will be displayed.
- Select Storage Class: The widget displays the monitoring data of the STANDARD storage class by default. You can manually select another storage class such as STANDARD_IA and ARCHIVE.

Note:

If you want to quickly reset the configured bucket and storage class, simply tap **Reset**. Then, the monitoring data of all buckets in the STANDARD storage class under the current account will be displayed.

Removing widget

Touch and hold the widget and tap Remove Widget.

COSCLI (Beta) COSCLI Overview

Last updated : 2022-05-09 14:18:31

COS provides the command-line client COSCLI to allow you to upload, download, delete, and perform other operations on COS objects by using simple commands.

COSCLI is written in Go and adopts the Cobra framework. It supports multi-bucket configurations and cross-bucket operations. You can run ./coscli [command] --help to see how to use COSCLI.

Features

- Generating and Modifying Configuration Files config
- Creating Buckets mb
- Deleting Buckets rb
- Tagging Bucket bucket-tagging
- Querying Bucket/Object List Is
- Obtaining Statistics on Different Types of Objects du
- Uploading/Downloading/Copying Objects cp
- Syncing Upload/Download/Copy sync
- Deleting Objects rm
- Obtaining Object Hash hash
- Listing Incomplete Multipart Uploads Generated Upon Upload Isparts
- Aborting Incomplete Multipart Uploads abort
- Restoring Archive Objects restore
- Getting Pre-Signed URLs signurl

Download and Installation Configuration

Last updated : 2022-07-20 11:34:18

COSCLI provides binary packages for Windows, macOS, and Linux, which can be used after simple installation and configuration.

Download Address

GitHub Address	Tencent Cloud Address
Windows	Windows
Mac	Mac
Linux	Linux

Note :

The current version is v0.11.1-beta. To get the latest and earlier versions and changelogs of the tool, see Releases.

Installation

Windows

- 1. Download COSCLI for Windows.
- 2. Move the downloaded COSCLI executable file to the C:\Users\<username> directory.
- 3. Rename coscli-windows.exe to coscli.exe .
- 4. Press Windows + R to launch the Run program.
- 5. In the dialog box, enter cmd and press Enter to open the command line window.
- 6. Enter coscli --version in the command line window. If the following information is printed out, the installation is successful:

Note :

On Windows, the method for using COSCLI may vary slightly by command line client. If COSCLI does not work properly after coscli [command] is entered, try ./coscli [command].



coscli version v0.11.1-beta

macOS

1. Run the following command to download COSCLI:

```
wget https://github.com/tencentyun/coscli/releases/download/v0.11.1-beta/coscli
-mac
```

2. Run the following command to rename the file:

```
mv coscli-mac coscli
```

3. Run the following command to modify the file execution permission:

chmod 755 coscli

4. Enter the command line ./coscli --version . If the following information is printed out, the installation is successful:

coscli version v0.11.1-beta

Note :

```
When you use COSCLI on macOS, if the Unable to open "coscli" because the developer cannot be verified prompt is displayed, go to Settings > Security & Privacy > General and select Open COSCLI anyway, and then COSCLI can be used properly.
```

Linux

1. Download COSCLI for Linux or run the following command to download COSCLI:

```
wget https://github.com/tencentyun/coscli/releases/download/v0.11.1-beta/coscli
-linux
```

2. Run the following command to rename the file:

```
mv coscli-linux coscli
```

3. Run the following command to modify the file execution permission:

```
chmod 755 coscli
```

4. Enter ./coscli --version in the command line window. If the following information is printed out, the installation is successful:

coscli version v0.11.1-beta

Parameter Configuration

You can quickly check how to use COSCLI by running the command ./coscli --help .

COSCLI will generate a configuration file in ~/.cos.yaml by default when used for the first time. You can also run the ./coscli config init command later to interactively generate a configuration file in another location.

The configuration items in the configuration file are as described below:

Configuration Item	Description
Secret ID	Key ID, which can be created and obtained from the CAM console.
Secret Key	Key, which can be created and obtained from the CAM console.
Session Token	Temporary key token, which should be specified if a temporary key is used; otherwise, press Enter to skip it.
APP ID	APP ID is the account you get after successfully registering your Tencent Cloud account. It is automatically assigned by the system and can be obtained from Account Information. The full name of a bucket consists of Bucket Name and APP ID in the format of <bucketname-appid> . For more information about bucket naming conventions, see Bucket Overview.</bucketname-appid>

Configuration Item	Description
Bucket Name	Bucket name, which forms the full name of the bucket together with the APP ID in the format of <bucketname-appid> . For more information about bucket naming conventions, see Bucket Overview.</bucketname-appid>
Bucket Region	Bucket region. For details, see Regions and Access Endpoints.
Bucket Alias	Bucket alias, which can be used to replace BucketName-APPID after configuration to shorten required commands. If it is not configured, its value will be the value of BucketName-APPID.

Other configuration methods

In addition to interactively generating a configuration file by running ./coscli config init , you can also manually write a YAML configuration file for COSCLI as shown below:

Note:

```
By default, COSCLI reads configuration items from ~/.cos.yaml . If you want to use the custom configuration file, select -c (--config-path) after running commands. The strings (secretid / secretkey / sessiontoken ) stored in the configuration file are encrypted.
```

Configuring multiple buckets



COSCLI supports multiple buckets; however, it will ask you to configure the information of only one bucket during the initial configuration. You can run the ./coscli config add command later to add the information of more buckets.

Note :

For more configuration file operations, see Generating and Modifying Configuration Files - config.

Common Options

Last updated : 2022-05-09 14:18:31

You can view the common options supported by COSCLI with the ./coscli --help or ./coscli -h command.

Option Description

The following are common options for COSCLI, which can be used in all its commands:

Option	Description
-h,help	Outputs help information. You can view the help information and usage of the tool with the $-h$ or $help$ command. You can also enter $-h$ after each command (with no parameter appended) to see how to use the command. For example, to view the specific usage of the bucket creation command, enter coscli mb $-h$.
-c,config-path	Configuration file path, which is $\sim /.cos.yaml$ for COSCLI by default. You can also specify a custom configuration file by adding $-c$ after a command.
-e,endpoint	In addition to configuring the region of a bucket in advance in the configuration file, you can also use <u>e</u> in COSCLI to specify the bucket endpoint in the format of <u>cos.<region>.myqcloud.com</region></u> , where <u><region></region></u> represents the bucket region, such as <u>ap-guangzhou</u> and <u>ap-beijing</u> . For the list of regions supported by COS, see Regions and Access Endpoints.
-i,secret-id	Specifies the SecretId used to access COS.
-k,secret-key	Specifies the SecretKey used to access COS.
-t,session-token	If you access COS with a temporary key, you can use $-t$ to specify its token.
-v,version	Displays the COSCLI version.

Examples

Example 1: Switching bucket to upload an object

When you need to switch to a bucket in another region through COSCLI, you can use the endpoint of the bucket.

For example, to upload the local file test.txt to the bucket examplebucket-1250000000 in the Chengdu region with the endpoint cos.ap-chengdu.myqcloud.com , run the following command:

```
./coscli cp test.txt cos://examplebucket-125000000/test.txt -e cos.ap-chengdu.my
qcloud.com
```

Example 2: Switching user account to view the file list

When you need to use the identity of another account, you can use the -i and -k options to specify the SecretId and SecretKey of your key respectively.

For example, to use the identity of another account to list the files in the bucket examplebucket-1250000000 in the Chengdu region, run the following command:

```
./coscli ls cos://examplebucket-1250000000 -e cos.ap-chengdu.myqcloud.com -i AKID
Yv3vWrwkHXVDfqkNjoc9PP8anjOm**** -k 4rNbYF1XmmVw67rKWTBernUu66u****
```

Common Commands Generating and Modifying Configuration Files - config

Last updated : 2022-05-09 14:18:31

Command Syntax

The config command is used to generate and modify the configuration file.

./coscli config [command] [flag]

Note :

- After setting the configuration items correctly, you can run ./coscli config show to view the configuration.
- For other common options of this command (such as switching bucket and user account), see Common Options.

config includes the following sub-commands:

Command	Description
add	Adds a new bucket configuration.
delete	Deletes an existing bucket configuration.
init	Generates the configuration file interactively.
set	Modifies one or more configuration items in the base group of the configuration file.
show	Prints information in a specific configuration file.

config and its sub-commands include the following optional flags:

Flag Abbreviation	Flag Name	Description
-------------------	-----------	-------------


Flag Abbreviation	Flag Name	Description
-C	config-path	Path of the configuration file to use

The config add sub-command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-a	alias	Bucket alias
-b	bucket	Bucket name
-r	region	Region of the bucket

The config delete sub-command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-a	alias	Bucket alias

The config set sub-command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-i	secret_id	Secret ID
-k	secret_key	Secret key
-t	token	Token

Examples

Adding a new bucket configuration

./coscli config add -b examplebucket3-1250000000 -r ap-chengdu -a bucket3

Deleting an existing bucket configuration

```
./coscli config delete -a bucket3
```

Modifying session-token in the default configuration file

```
./coscli config set -t test-token123
```

Printing information in a specific configuration file

```
./coscli config show -c /your/config/path.yaml
```

Creating a Bucket - mb

Last updated : 2022-05-09 14:18:32

The mb command is used to create a bucket.

Command Syntax

./coscli mb cos://<BucketName-APPID> -r <Region> [flag]

mb includes the following optional flags:

Flag Abbreviation	Flag Name	Description
None	BucketName-APPID	Customizes the bucket name, such as examplebucket- 125000000
-r	region	Indicates the region of the bucket

Note :

- To use the md command to create a bucket in COSCLI, you need to run the config add command to update the bucket configuration in the configuration file after running the md command successfully.
- For other common options of this command (such as switching bucket and user account), see Common Options.

Examples

```
// Create the bucket3 bucket.
./coscli mb cos://bucket3-1250000000 -r ap-chengdu
// Update the configuration file.
./coscli config add -b bucket3-1250000000 -r ap-chengdu -a bucket3
// After the update, you can access the bucket at cos://bucket3.
```

Deleting a Bucket - rb

Last updated : 2022-05-09 14:18:32

The rb command is used to delete a bucket.

Command Syntax

./coscli rb cos://<BucketName-APPID> -r <Region> [flag]

rb includes the following optional flags:

Flag Abbreviation	Flag Name	Description
None	BucketName-APPID	Specifies the name of the bucket to be deleted, such as examplebucket- 125000000
-r	region	Indicates the region of the bucket

Note:

For other common options of this command (such as switching bucket and user account), see Common Options.

Examples

```
// Delete the bucket3 bucket.
./coscli rb cos://bucket3-1250000000 -r ap-chengdu
// Update the configuration file.
./coscli config delete -a bucket3
```

Tagging Bucket - bucket-tagging

Last updated : 2022-05-09 14:18:32

The bucket-tagging command is used to create (modify), get, and delete bucket tags.

Command Options

The bucket-tagging command contains the following optional flags:

Flag Abbreviation	Flag Name	Description
-m	method	Specifies the operation to be performed, including PUT, GET, and DELETE

Note :

For other common options of this command (such as switching bucket and user account), see Common Options.

Adding or Modifying Bucket Tag

A bucket tag is represented by a key-value pair. Only the bucket owner and users with the PutBucketTagging permission can add or modify bucket tags. Error message "403 AccessDenied" will be returned for other users.

Command format

```
./coscli bucket-tagging --method put cos://<BucketName-APPID> key1#value1 key2#va
lue2
```

Here, key#value represents the tag key-value pair separated by # . If the bucket does not have a tag, this command will add the specified tag to the bucket; otherwise, it will overwrite the original tag.

Example

To configure two tags for the bucket exmaplebucket-1250000000 with the keys of 1 and 2 as well as values of 111 and 222 respectively, run the following command:

./coscli bucket-tagging --method put cos://exmaplebucket-1250000000 1#111 2#222

Querying Bucket Tags

Command format

./coscli bucket-tagging --method get cos://<BucketName-APPID>

Example

```
./coscli bucket-tagging --method get cos://exmaplebucket-125000000
```

The following output shows that exmaplebucket-125000000 has two tags with the keys of 1 and 2 as well as values of 111 and 222 respectively.

KEY | VALUE -----+-----1 | 111 2 | 222

Deleting Bucket Tags

Command format

./coscli bucket-tagging --method delete cos://<BucketName-APPID>

Example

./coscli bucket-tagging --method delete cos://exmaplebucket-125000000

Querying Bucket/Object List - Is

Last updated : 2022-05-09 14:18:32

The 1s command is used to query the list of buckets, objects in a bucket, and objects in a directory.

Command Syntax

./coscli ls [cos://bucketAlias[/prefix/]] [flag]

Note:

- For more information on bucketAlias , see Download and Installation Configuration.
- For other common options of this command (such as switching bucket and user account), see Common Options.
- ls includes the following optional parameters:

Parameter Format	Description	Example
cos://bucketAlias	Specifies a bucket.	cos://bucket1
/prefix/	Specifies a directory.	/picture/

ls includes the following optional flags:

Flag Abbreviation	Flag Name	Description
None	include	Includes specific objects.
None	exclude	Excludes specific objects.
-r	recursive	Specifies whether to traverse directories recursively and list all objects.

Note:

- --include and --exclude support standard regular expressions. You can use regular expressions to filter objects that meet your requirements.
- When using zsh, you may need to add double quotes at both ends of the pattern string.

./coscli ls cos://bucket1 -r --include ".*.mp4"

Examples

Listing all buckets of the current account

./coscli ls

Listing objects

Listing all objects in the bucket1 bucket

```
./coscli ls cos://bucket1
```

Listing all objects and subdirectories in the picture directory in the bucket1 bucket

```
./coscli ls cos://bucket1/picture/
```

Listing all objects in the picture directory in the bucket1 bucket recursively

```
./coscli ls cos://bucket1/picture/ -r
```

Listing all MP4 objects in the bucket1 bucket recursively

./coscli ls cos://bucket1 -r --include .*.mp4

Listing all non-MP4 objects in the bucket1 bucket recursively

```
./coscli ls cos://bucket1 -r --exclude .*.mp4
```

Listing all non-JPG objects prefixed with test in the picture directory in the bucket1 bucket



./coscli ls cos://bucket1/picture -r --include ^picture/test.* --exclude .*.jpg

Obtaining Statistics on Different Types of Objects - du

Last updated : 2022-05-09 14:18:33

The du command is used to obtain statistics on each storage class for a bucket or a directory, including the size and number of objects in each storage class.

Command Syntax

./coscli du cos://<bucketAlias>[/prefix/] [flag]

Note:

- For more information on bucketAlias , see Download and Installation Configuration.
- For other common options of this command (such as switching bucket and user account), see Common Options.

1s includes the following optional parameters:

Parameter Format	Description	Example
/prefix/	Specifies a directory.	cos://bucket1/picture/

du includes the following optional flags:

Flag Abbreviation	Flag Name	Description
None	include	Includes specific objects.
None	exclude	Excludes specific objects.

Note:

• --include supports standard regular expressions. You can use regular expressions to filter objects that meet your requirements.

• When using zsh , you may need to enclose the pattern string with double quotation marks.

./coscli du cos://bucket1/picture/ --include ".*.mp4"

Examples

Listing statistics on objects in the bucket1 bucket

```
./coscli du cos://bucket1
```

Listing statistics on objects in the picture directory in the bucket1 bucket

```
./coscli du cos://bucket1/picture/
```

Listing statistics on all MP4 objects in the picture directory in the bucket1 bucket

./coscli du cos://bucket1/picture/ --include .*.mp4

Listing statistics on all non-MD objects in the picture directory in the bucket1 bucket

./coscli du cos://bucket1/picture/ --exclude .*.md

Uploading/Downloading/Copying Objects - cp

Last updated : 2022-12-19 12:17:36

The cp command is used to upload, download, or copy objects.

Command Syntax

./coscli cp <source_path> <destination_path> [flags]

Note:

- For more information on bucketAlias , see Download and Installation Configuration.
- For other common options of this command (such as switching bucket and user account), see Common Options.

cp includes the following optional flags:

Flag Abbreviation	Flag Name	Description
None	include	Includes specific objects.
None	exclude	Excludes specific objects.
-r	recursive	Specifies whether to traverse all objects in the directory recursively
None	storage-class	Specifies the storage class for the object to upload. Default value: STANDARD
None	part-size	Part size in MB. Default value: 32 MB
None	thread-num	Number of concurrent threads. Default value: 5
None	rate-limiting	Speed limit for a single URL in MB/s. Value range: 0.1–100 MB/s



Flag Abbreviation	Flag Name	Description
None	meta	<pre>Metadata of the uploaded file, including certain HTTP standard attributes (HTTP Header) and custom metadata prefixed with x-cos- meta- (User Meta). The file metadata is in the format of header:value#header:value , such as Expires:2022-10- 12T00:00:00.000Z#Cache- Control:no-cache#Content- Encoding:gzip#x-cos-meta- x:x .</pre>

Note:

- cp automatically uses concurrent upload/download for large objects.
- If an object is larger than --part-size , COSCLI will split the object into multiple parts according to -part-size and use --thread-num threads to concurrently upload/download the object.
- Each thread maintains a URL. For each URL, you can use the --rate-limiting parameter to limit the speed of a single URL. When concurrent upload/download is enabled, the total rate is --thread-num * --rate-limiting.
- If an object is uploaded/downloaded in parts, checkpoint restart will be enabled by default.
- --include and --exclude support standard regular expression syntax, so you can use them to filter out objects that meet specific criteria.
- When using zsh, you may need to add double quotes at both ends of the pattern string.

```
./coscli cp ~/test/ cos://bucket1/example/ -r --include ".*.txt" --meta=x-cos-met
a-a:a#ContentType:text#Expires:2022-10-12T00:00:00.000Z
```

Samples

Upload

Uploading a single object

./coscli cp ~/example.txt cos://bucket1/example.txt

Uploading all objects in the local directory test to the example directory in the bucket1 bucket

```
./coscli cp ~/test/ cos://bucket1/example/ -r
```

Uploading all MP4 objects in the local directory test to the example directory in the bucket1 bucket

```
./coscli cp ~/test/ cos://bucket1/example/ -r --include .*.mp4
```

Uploading all non-MD objects in the local directory test to the example directory in the bucket1 bucket

./coscli cp ~/test/ cos://bucket1/example/ -r --exclude .*.md

Uploading all objects in the dir directory (containing the dirA, dirB, dirC, and dirD subdirectories) except the dirD directory

```
./coscli cp dir/ cos://bucket1/example/ -r --exclude dirD/.*
```

Uploading all objects in the local directory test to the example directory in the bucket1 bucket using the ARCHIVE storage class

```
./coscli cp ~/test/ cos://bucket1/example/ -r --storage-class ARCHIVE
```

Uploading the local file.txt file to the bucket1 bucket and setting the single-URL speed limit to 1.3 MB/s

./coscli cp ~/file.txt cos://bucket1/file.txt --rate-limiting 1.3

Download

Downloading a single object

```
./coscli cp cos://bucket1/example.txt ~/example.txt
```

Downloading all objects in the example directory in the bucket1 bucket to the test directory

./coscli cp cos://bucket1/example/ ~/test/ -r

Downloading all MP4 objects in the example directory in the bucket1 bucket to the test directory

```
./coscli cp cos://bucket1/example/ ~/test/ -r --include .*.mp4
```

Downloading all non-MD objects in the example directory in the bucket1 bucket to the test directory

```
./coscli cp cos://bucket1/example/ ~/test/ -r --exclude .*.md
```

Сору

Copying a single object within a bucket

./coscli cp cos://bucket1/example.txt cos://bucket1/example_copy.txt

Copying a single object across buckets

./coscli cp cos://bucket1/example.txt cos://bucket2/example_copy.txt

Copying all objects in the example1 directory in bucket1 to the example2 directory in bucket2

./coscli cp cos://bucket1/example1/ cos://bucket2/example2/ -r

Copying all MP4 objects in the example1 directory in bucket1 to the example2 directory in bucket2

./coscli cp cos://bucket1/example1/ cos://bucket2/example2/ -r --include .*.mp4

Copying all non-MD objects in the example1 directory in bucket1 to the example2 directory in bucket2

./coscli cp cos://bucket1/example1/ cos://bucket2/example2/ -r --exclude .*.md

Syncing Upload/Download/Copy - sync

Last updated : 2022-05-03 10:52:28

Command Syntax

The sync command is used to sync object upload, download, and copy. The difference between sync and cp is that sync first compares the CRC64 value of an object with the same name that already exists, and if the value is the same, the object will not be transferred.

./coscli sync <source_path> <destination_path> [flag]

Note:

- For more information on bucketAlias , see Download and Installation Configuration.
- For more general options for this command (such as switching buckets or user accounts), see General Options.

sync includes the following optional flags:

Flag Abbreviation	Flag Full Name	Description
None	include	Includes specific objects.
None	exclude	Excludes specific objects.
-r	recursive	Specifies whether to traverse all objects in the directory recursively
None	storage-class	Specifies the storage class for the object to upload. Default value: STANDARD
None	part-size	Part size. Default value: 32 MB; maximum value: 5 GB
None	thread-num	Number of concurrent threads. Default value: 5



Flag Abbreviation	Flag Full Name	Description
None	rate-limiting	Speed limit for a single URL. Value range: 0.1–100 MB/s

Note:

- sync automatically uses concurrent upload/download for large objects.
- If an object is larger than --part-size , COSCLI will split the object into multiple parts according to -part-size and use --thread-num threads to concurrently upload/download the object.
- Each thread maintains a URL. For each URL, you can use the _-rate-limiting parameter to limit the speed of a single URL. When concurrent upload/download is enabled, the total rate is _-thread-num * _-rate-limiting .
- If an object is uploaded/downloaded in parts, checkpoint restart will be enabled by default.
- --include and --exclude support standard regular expression syntax, so you can use them to filter files that meet specific criteria.
- When using <code>zsh</code> , you may need to add double quotes at both ends of the <code>pattern</code> string.

./coscli sync ~/test/ cos://bucket1/example/ -r --include ".*.mp4"

Examples

Syncing object upload

./coscli sync ~/example.txt cos://bucket1/example.txt

Syncing object download

./coscli sync cos://bucket1/example.txt ~/example.txt

Syncing intra-bucket replication

./coscli sync cos://bucket1/example.txt cos://bucket1/example_copy.txt

Syncing cross-bucket replication



./coscli sync cos://bucket1/example.txt cos://bucket2/example_copy.txt

Deleting Objects - rm

Last updated : 2022-05-03 11:01:50

The rm command is used to delete an object.

Command Syntax

./coscli rm cos://<bucketAlias>[/prefix/] [cos://<bucket-name>[/prefix/]...] [fla
g]

Note:

- For more information on bucketAlias , see Download and Installation Configuration.
- For more general options for this command (such as switching buckets or user accounts), see General Options.

rm includes the following optional flags:

Flag Abbreviation	Flag Full Name	Description
None	include	Includes specific objects.
None	exclude	Excludes specific objects.
-ř	recursive	Specifies whether to traverse all objects in the directory recursively.
-f	force	Forces deletion (no prompt before the deletion).

Note:

- --include and --exclude support standard regular expression syntax, so you can use them to filter files that meet specific criteria.
- When using <code>zsh</code> , you may need to add double quotes at both ends of the <code>pattern</code> string.

./coscli rm cos://bucket1/example/ -r --**include** ".*.mp4"

Examples

Deleting an object

```
./coscli rm cos://bucket1/fig1.png
```

Deleting all objects in the picture directory

```
./coscli rm cos://bucket1/pictrue/ -r
```

Getting File Hash Value - hash

Last updated : 2022-05-03 11:04:52

The hash command is used to calculate the hash value of a local file or get the hash value of a file in COS.

Command Syntax

```
./coscli hash <object-name> [flag]
```

Note:

- For more information on bucketAlias , see Download and Installation Configuration.
- For more general options for this command (such as switching buckets or user accounts), see General Options.

The hash command contains the following optional flags:

Flag Abbreviation	Flag Full Name	Description
None	type	Hash type, which can be MD5 or CRC64 (default)

Examples

Calculating the CRC64 value of local file

./coscli hash ~/test.txt

Getting the MD5 value of COS file

./coscli hash cos://bucket1/example.txt --type=md5

Listing Incomplete Multipart Uploads - Isparts

Last updated : 2022-05-03 11:08:19

The lsparts command is used to list the generated incomplete multipart uploads.

Command Syntax

./coscli lsparts cos://<bucketAlias>[/prefix/] [flag]

Note:

- For more information on bucketAlias , see Download and Installation Configuration.
- For more general options for this command (such as switching buckets or user accounts), see General Options.

The lsparts command contains the following optional flags:

Flag Abbreviation	Flag Full Name	Description
None	include	Includes specific objects.
None	exclude	Excludes specific objects.
None	limit	Specifies the maximum quantity (0-1000) to be listed.

Examples

Listing all incomplete multipart uploads in bucket1 bucket

./coscli lsparts cos://bucket1

Listing all incomplete multipart uploads in picture folder in bucket1 bucket

```
./coscli lsparts cos://bucket1/picture/
```



Clearing Incomplete Multipart Uploads - abort

Last updated : 2022-05-03 11:10:23

The abort command is used to clear the generated incomplete multipart uploads.

Command Syntax

./coscli abort cos://<bucketAlias>[/prefix/] [flag]

Note:

- For more information on bucketAlias , see Download and Installation Configuration.
- For more general options for this command (such as switching buckets or user accounts), see General Options.

The abort command contains the following optional flags:

Flag Abbreviation	Flag Full Name	Description
None	include	Includes specific objects.
None	exclude	Excludes specific objects.

Examples

Clearing all incomplete multipart uploads in bucket1 bucket

```
./coscli abort cos://bucket1
```

Clearing all incomplete multipart uploads in picture folder in bucket1 bucket

```
./coscli abort cos://bucket1/picture/
```

Restoring Archived Objects - restore

Last updated : 2022-05-03 11:19:36

The restore command is used to restore archived objects.

Command Syntax

./coscli restore cos://<bucketAlias>[/prefix/] [flag]

Note:

- For more information on bucketAlias , see Download and Installation Configuration.
- For more general options for this command (such as switching buckets or user accounts), see General Options.

The	restore	command	contains	the	following	optional	flags:
-----	---------	---------	----------	-----	-----------	----------	--------

Flag Abbreviation	Flag Full Name	Description
None	include	Includes specific objects
None	exclude	Excludes specific objects
-d	days	Specifies the expiration time of temporary objects, which is 3 days by default
-m	mode	Specifies the restoration mode, which is Standard by default
-r	recursive	Traverses the folder recursively

Note :

- --include and --exclude support standard regular expression syntax, so you can use them to filter out objects that meet specific criteria.
- When using zsh, you may need to add double quotes at both ends of the pattern string.

./coscli restore cos://bucket1/example/ -r --include ".*.mp4"

Examples

Restoring archived objects in bucket1 bucket in Standard mode

```
./coscli restore cos://bucket1/picture.jpg
```

Restoring all archived objects in picture folder in bucket1 bucket in Expedited mode

./coscli restore cos://bucket1/picture/ -r --mode Expedited

Note:

Before running this command, you need to make sure that all objects in the folder are of the same storage class (such as ARCHIVE). If there are objects in different classes, use --include or --exclude to filter out objects in the same class.

Getting Pre-signed URL - signurl

Last updated : 2022-05-03 11:21:54

The signurl command is used to get the pre-signed URL of an object, through which the object can be accessed anonymously.

Command Syntax

./coscli signurl cos://<bucketAlias>/<key> [flag]

Note :

- For more information on bucketAlias , see Download and Installation Configuration.
- For more general options for this command (such as switching buckets or user accounts), see General Options.

The signurl command contains the following optional flags:

Flag Abbreviation	Flag Full Name	Description
-t	time	Sets the URL expiration time, which is 1000s by default

Examples

Getting the pre-signed URL of picture.jpg in bucket1 bucket

./coscli signurl cos://bucket1/picture.jpg

Getting the pre-signed URL of picture.jpg in bucket2 bucket and setting the URL expiration time to 1314s

```
./coscli signurl cos://bucket2/picture.jpg --time 1314
```

FAQs

Last updated : 2022-01-17 14:11:06

What's the difference between COSCLI and COSCMD?

- 1. COSCLI is a binary package compiled using Golang. It is ready-to-use and does not require installing any dependency before the installation and deployment. However, COSCMD is compiled in Python and requires installing the Python environment and dependency packages.
- 2. COSCLI supports setting an alias for a bucket, meaning that you can use a short string to represent <bucketname-appid> for convenience. However, COSCMD does not support setting aliases and requires using <bucketname-appid> to specify a bucket, making the commands harder to use and read.
- COSCLI supports configuring multiple buckets in the configuration file and supports cross-bucket operations. However, with COSCMD, you can only configure one bucket in the configuration file and the commands for crossbucket operations are long.

COSCMD

Last updated : 2022-11-22 17:23:17

Overview

COSCMD enables you to use simple command lines to batch-operate objects, such as upload, download, and delete.

Operating Environments

Operating system

Windows, Linux, and macOS

Note :

- Local characters should use UTF-8 encoding. Otherwise, exceptions will occur when you operate on Chinese files.
- Ensure that the local time is in sync with UTC. If there is a large deviation between the two, COSCMD might not function properly.

Software dependency

- Python 2.7/3.5/3.6/3.9
- Latest version of pip

Note :

You are advised to install the latest version of Python (3.9.0) that is integrated with pip.

Installation and configuration

- For more information about the installation and configuration of the environment, see Python.
- For more information about the installation and configuration of pip, see Installation.

Download and Installation



You can install COSCMD in the following three ways:

1.1 Installing with pip

Run the pip command to install COSCMD:

```
pip install coscmd
```

After the installation is complete, you can run the -v or --version command to view the version information.

Note :

```
After installing COSCMD in Windows, you need to add the C:\python_install_dir and C:\python_install_dir\Scripts paths to environment variables.
```

1.2 Upgrading with pip

After the installation is complete, you can run the following command to upgrade COSCMD:

pip install coscmd -U

2. Installing with the source code (not recommended)

You can click here to download the source code.

```
git clone https://github.com/tencentyun/coscmd.git
cd coscmd
python setup.py install
```

Note :

If your Python version is 2.6, installing the dependent libraries with pip may fail. Therefore, this installation method is recommended.

3. Installing from offline

Note:

Ensure that the two devices are installed with the same version of Python. Otherwise, the installation might fail.



```
# Run the following commands on a device that is connected to the Internet
mkdir coscmd-packages
pip download coscmd -d coscmd-packages
tar -czvf coscmd-packages.tar.gz coscmd-packages
# Copy the installation package to a device that is not connected to the Internet
and run the following commands
tar -xzvf coscmd-packages.tar.gz
pip install coscmd --no-index -f coscmd-packages
```

Parameter Configuration

Viewing the help option

Run the -h or --help command to view the information and usage of COSCMD.

coscmd -h

The help information is as follows:

```
usage: coscmd [-h] [-d] [-s] [-b BUCKET] [-r REGION] [-c CONFIG_PATH]
[-1 LOG_PATH] [--log_size LOG_SIZE]
[--log_backup_count LOG_BACKUP_COUNT] [-v]
{config,upload,download,delete,abort,copy,move,list,listparts,info,restore,signur
1, createbucket, deletebucket, putobjectacl, getobjectacl, putbucketacl, getbucketacl, p
utbucketversioning, getbucketversioning, probe}
. . .
an easy-to-use but powerful command-line tool. try 'coscmd -h' to get more
informations. try 'coscmd sub-command -h' to learn all command usage, likes
'coscmd upload -h'
positional arguments:
{config,upload,download,delete,abort,copy,move,list,listparts,info,restore,signur
1, createbucket, deletebucket, putobjectacl, getobjectacl, putbucketacl, getbucketacl, p
utbucketversioning, getbucketversioning, probe}
config Config your information at first
upload Upload file or directory to COS
download Download file from COS to local
delete Delete file or files on COS
abort Aborts upload parts on COS
copy Copy file from COS to COS
move move file from COS to COS
list List files on COS
listparts List upload parts
```

```
info Get the information of file on COS
restore Restore
signurl Get download url
createbucket Create bucket
deletebucket Delete bucket
putobjectacl Set object acl
getobjectacl Get object acl
putbucketacl Set bucket acl
getbucketacl Get bucket acl
putbucketversioning
Set the versioning state
getbucketversioning
Get the versioning state
probe Connection test
optional arguments:
-h, --help show this help message and exit
-d, --debug Debug mode
-s, --silence Silence mode
-b BUCKET, --bucket BUCKET
Specify bucket
-r REGION, --region REGION
Specify region
-c CONFIG_PATH, --config_path CONFIG_PATH
Specify config_path
-1 LOG_PATH, --log_path LOG_PATH
Specify log_path
--log_size LOG_SIZE specify max log size in MB (default 1MB)
--log_backup_count LOG_BACKUP_COUNT
specify log backup num
-v, --version show program's version number and exit
```

In addition, you can enter h after each command (with no parameter appended) to see how to use the command. For example:

coscmd upload -h // View how to use the upload command

Generating a configuration file

COSCMD will first read the necessary information from the configuration file before running. By default, COSCMD will read configuration items from ~/.cos.conf .

Note:

Before the configuration, you need to go to the COS console to create a bucket (e.g., configure-bucket-1250000000) for parameter configuration and create a key.



The following is a sample configuration file:

```
[common]
secret_id = AKIDA6wUmImTMzvXZNbGLCgtusZ2E8mG****
secret_key = TghWBCyf5LIyTcXCoBdw1oRpytWk****
bucket = configure-bucket-125000000
region = ap-chengdu
max_thread = 5
part_size = 1
retry = 5
timeout = 60
schema = https
verify = md5
anonymous = False
```

Note:

- schema can be set to http or https (default).
- anonymous can be set to True or False. If it is set to True, the anonymous mode is used (the signature is empty).
- For more information about the parameter description, run the coscmd config -h command.

Generating a configuration file via the config command

You can run the config command to automatically generate a configuration file in ~/.cos.conf . The command format is as follows:

```
cosemd config [OPTION]...<FILE>...
[-h] --help
[-a] <SECRET_ID>
[-s] <SECRET_KEY>
[-t] <TOKEN>
[-b] <BucketName-APPID>
[-r] <REGION> | [-e] <ENDPOINT>
[-m] <MAX_THREAD>
[-p] <PART_SIZE>
[--do-not-use-ssl]
[--anonymous]
```

Note:

Fields enclosed in "[]" are optional, and those in "<>" are required.

The parameters are described as follows:

Option	Parameter Description	Valid Value	Required
-a	Key ID, which can be obtained at Manage API Key	String	Yes
-S	Key, which can be obtained at Manage API Key	String	Yes
-t	Temporary key token, which needs to be specified in the x-cos-security- token header when a temporary key is used.	String	No
-b	Name of the specified bucket, formatted as BucketName-APPID. For more information, see Naming Conventions. If this is your first time using COSCMD, you need to create a bucket in the COS console to configure COSCMD.	String	Yes
-r	Region of the bucket. For more information, see Regions and Access Endpoints.	String	Yes
-e	Endpoint of the request. Once you configure this parameter, the region parameter will be invalidated. If you use the default endpoint, this parameter is formatted as cos. <region>.myqcloud.com ; if you use a global acceleration endpoint, the format is cos.accelerate.myqcloud.com .</region>	String	No
-m	Maximum number of threads in a multi- thread operation (default: 5; value range: 1-30)	Number	No



Option	Parameter Description	Valid Value	Required
-р	Size of the multipart upload part, in MB (default: 1; value range: 1-1000)	Number	No
do-not-use-ssl	Uses the HTTP protocol instead of HTTPS	String	No
anonymous	Anonymous operation (without carrying a signature)	String	No

The following sample shows how to use the config command:

```
coscmd config -a AChT4ThiXAbpBDEFGhT4ThiXAbp**** -s WE54wreefvds3462refgwewe****
-b configure-bucket-1250000000 -r ap-chengdu
```

Common Commands

Specifying a bucket and its region

If you do not specify the bucket and region in the commands, the commands take effect for the bucket that is used to configure COSCMD. To perform operations on another bucket, you need to specify the bucket and the region where the bucket resides.

```
Note:
```

- Use the -b <bucketname-appid> parameter to specify the bucket name, which must be formatted as BucketName-APPID .
- Use the -r <region> parameter to specify the region where the bucket resides.
- · Command syntax

```
coscmd -b <BucketName-APPID> -r <region> <action> ...
```

• Sample: creating a bucket named examplebucket that resides in the Beijing region

```
coscmd -b examplebucket-1250000000 -r ap-beijing createbucket
```

• Sample: uploading "picture.jpg" from D drive to the examplebucket bucket

```
coscmd -b examplebucket-1250000000 -r ap-beijing upload D:/picture.jpg /
```

Specifying the configuration file and log file paths

If you do not specify the configuration file path, ~/.cos.conf will be used by default. Similarly, if you do not specify the log file path, ~/.cos.log will be used by default.

Note:

- Use the <u>-c <conf_path></u> parameter to specify the configuration file path. COSCMD will read configuration information from the path when running.
- Use the <u>-1 <log_conf></u> parameter to specify the log file path. COSCMD will output the logs generated at runtime to the log files in the path.

Command syntax

coscmd -c <conf_path> -l <log_conf> <action> ...

• Sample: setting the configuration file's path to /data/home/cos_conf and the log path to /data/home/cos_log , and creating a bucket named examplebucket in the Beijing region

```
coscmd -c /data/home/cos_conf -l /data/home/cos_log -b examplebucket-1250000000
-r ap-beijing createbucket
```

Running commands in debugging mode

If -d or -debug is added before a command, detailed operation information will be displayed when executing the command, as shown in the example below:

· Command syntax

coscmd -d upload <localpath> <cospath>

· Sample: outputting detailed information upon the upload

```
coscmd -d upload -rs D:/folder/ /
```


Running commands in silence mode

You can prefix -s or --silence in each command so that no message will be output.

Note:

To run this command, the version should be at least 1.8.6.24.

Command syntax

coscmd -s upload <localpath> <cospath>

• Sample:

```
coscmd -s upload D:/picture.jpg /
```

Common Bucket Commands

Creating a bucket

Note :

Specify -b <bucketname-appid> and -r <region> when you run the coscmd createbucket command; otherwise, an error may be reported. This is because if the bucket and region are not specified, this command takes effect for the bucket that is used to configure COSCMD.

Command syntax

coscmd -b <BucketName-APPID> createbucket

• Sample: creating a bucket named examplebucket that resides in the Beijing region

coscmd -b examplebucket-1250000000 -r ap-beijing createbucket



Deleting a bucket

Note :

coscmd deletebucket takes effect only for the bucket that is used to configure COSCMD. To delete another bucket, run the command with the -b <bucketname-appid> and -r <region> parameters specified.

· Command syntax

coscmd -b <BucketName-APPID> deletebucket

• Sample: deleting empty buckets

coscmd -b examplebucket-1250000000 -r ap-beijing deletebucket

Sample: forcibly deleting a non-empty bucket

```
coscmd -b examplebucket-1250000000 -r ap-beijing deletebucket -f
```

Note :

The -f parameter will forcibly delete the bucket, including all files, noncurrent folders (if versioning is enabled), and incomplete multipart uploads.

Common Object Commands

Uploading files

· Command syntax for uploading a file

coscmd upload <localpath> <cospath>

Note :

Replace "localpath" and "cospath" enclosed in "<>" with the path of the local file to upload and the COS storage path, respectively.

• Sample: uploading "picture.jpg" in D drive to the "doc" folder of COS

```
coscmd upload D:/picture.jpg doc/
```

· Sample: uploading "picture.jpg" in the "doc" folder in D drive to the "doc" folder of COS

coscmd upload D:/doc/picture.jpg doc/

· Sample: uploading a file to the ARCHIVE storage class to the "doc" directory of COS

coscmd upload D:/picture.jpg doc/ -H "{'x-cos-storage-class':'Archive'}"

Note:

```
When you set the HTTP header with the _H parameter, please use the JSON format, for example, coscmd upload _H "{'x-cos-storage-class': 'Archive', 'Content-Language': 'zh-CN'}" <localpath> <cospath> . For more information about the headers, see PUT Object.
```

• Sample: setting meta attributes and uploading a file to the "doc" folder of COS

coscmd upload D:/picture.jpg doc/ -H "{'x-cos-meta-example':'example'}"

Uploading a folder

• Command syntax for uploading a folder

```
coscmd upload -r <localpath> <cospath>
```

Note:

Windows users are advised to use the upload command in cmd or PowerShell that comes with the system. Other tools, such as Git Bash, have a different command path resolution strategy than PowerShell and can cause users' files to be uploaded to an incorrect path. Sample: uploading the "doc" folder in D drive to the root directory of COS

```
coscmd upload -r D:/doc /
```

· Sample: uploading the "doc" folder in D drive to the "doc" folder of COS

```
coscmd upload -r D:/doc doc
```

• Sample: uploading files synchronously (files with the same name, MD5 checksum, and file size will be skipped)

coscmd upload -rs D:/doc doc

Note:

Use the -s parameter to upload files synchronously while skipping those with the same MD5 value (please note that the source files in COS must have been uploaded using COSCMD v1.8.3.2 or above; the x-cos-meta-md5 header is included by default).

• Sample: uploading files synchronously (files with the same name and file size will be skipped)

coscmd upload -rs --skipmd5 D:/doc doc

Note:

The -s parameter allows synchronous upload, and the --skipmd5 parameter can be used to skip files with the same name and same file size.

• Sample: uploading the folder synchronously and deleting files that are deleted in the "doc" folder in D drive

coscmd upload -rs --delete D:/doc /

• Sample: ignoring .txt and .doc files in the "doc" folder in D drive



coscmd upload -rs D:/doc / --ignore *.txt,*.doc

Note:

When uploading folders, you can ignore certain types of files by using the <u>--ignore</u> parameter, or filter certain types of files by using <u>--include</u>. Multiple shell wildcard rules (separated by commas ,) are supported. To ignore a specified extension, , must be added at the end, or <u>""</u> must be used to enclose the extension.

• Sample: uploading .txt and .doc files in the "doc" folder in D drive

```
coscmd upload -rs D:/doc / --include *.txt,*.doc
```

Note:

- If the file to upload is larger than 10 MB, COSCMD will upload with multipart upload. The command is coscmd upload <localpath> <cospath> (same as simple upload).
- COSCMD supports checkpoint restart to resume the upload of large files. When the multipart upload of a large file fails, only the failed parts will be uploaded when the operation is resumed instead of starting over from scratch (please ensure that the directory and content of the re-uploaded file are consistent with the uploaded directory).
- COSCMD performs MD5 verification on each part during multipart upload.
- When COSCMD uploads a file, the x-cos-meta-md5 header is carried by default, whose value is the file's MD5 checksum. If the command has carried the --skipmd5 parameter, this header will not be carried.

Querying a file list

The query command is as follows:

Command syntax

coscmd list <cospath>

• Sample: recursively querying the list of all files prefixed with "doc/" in this bucket

coscmd list doc/

· Sample: recursively querying the file list, number of files, and the file sizes of a bucket

```
coscmd list -ar
```

· Sample: recursively querying the list of all files prefixed with "examplefolder"

```
coscmd list examplefolder/ -ar
```

· Sample: querying the historical versions of all files in a bucket

```
coscmd list -v
```

Note :

- Replace "cospath" enclosed in "<>" with the COS path of the file list to query. If <cospath> is empty, the root directory of the current bucket is queried.
- Use -a to query all files.
- Use -r to query files recursively. The number and total size of the files are listed at the end of the returned result.
- Use -n num to set the maximum number of files to query.

Viewing the file information

The command is as follows:

Command syntax

coscmd info <cospath>

• Sample: viewing the metadata of "doc/picture.jpg"

```
coscmd info doc/picture.jpg
```



Note :

Replace "cospath" enclosed in "<>" with the COS path of the file to query.

Downloading a file or folder

Command syntax for downloading a file

```
coscmd download <cospath> <localpath>
```

Note :

Replace "cospath" and "localpath" enclosed in "<>" with the COS path of the file to download and the local storage path, respectively.

• Sample: downloading the "doc/picture.jpg" file in COS to "D:/picture.jpg"

coscmd download doc/picture.jpg D:/picture.jpg

• Sample: downloading the "doc/picture.jpg" file in COS to drive D

coscmd download doc/picture.jpg D:/

Sample: downloading a specified version of "picture.jpg" to drive D

coscmd download picture.jpg --versionId MTg0NDUxMzc2OTM4NTExNTg7Tjg D:/

Command syntax for downloading a folder

coscmd download -r <cospath> <localpath>

• Sample: downloading the "doc" folder to "D:/folder/doc"

```
coscmd download -r doc D:/folder/
```

Sample: downloading files in the root directory while ignoring those in the doc directory that is under the root directory

```
coscmd download -r / D:/ --ignore doc/*
```

· Sample: downloading all files in the root directory of the current bucket and overwriting local files

```
coscmd download -rf / D:/examplefolder/
```

Note:

If a file with the same name exists locally, the download will fail. In this case, you need to use the -f parameter to overwrite the local file.

• Sample: synchronously downloading all files in the root directory of the current bucket while skipping those with the same filename and MD5 checksum

coscmd download -rs / D:/examplefolder

Note:

Use the <u>-s</u> or <u>--sync</u> parameter to skip identical files that already exist locally when downloading a folder (provided that the downloaded files were uploaded via the COSCMD upload API and the <u>x- cos-</u> meta-md5 header was included).

• Sample: synchronously downloading all files in the root directory of the current bucket while skipping those with the same filename and file size

coscmd download -rs --skipmd5 / D:/examplefolder

• Sample: synchronously downloading all files in the root directory of the current bucket and deleting local files that have been deleted in cloud

```
coscmd download -rs --delete / D:/examplefolder
```

• Sample: ignoring .txt or .doc files

```
coscmd download -rs / D:/examplefolder --ignore *.txt,*.doc
```

Note :

When downloading folders, you can ignore certain types of files by using the __ignore parameter, or filter certain types of files by using __include . Multiple shell wildcard rules (separated by commas ,) are supported. To ignore a specified extension, , must be added at the end, or "" must be used to enclose the extension.

• Sample: filtering .txt and .doc files

```
coscmd download -rs / D:/examplefolder --include *.txt,*.doc
```

Note :

```
Since the old mget API is disused, please use the download API for multipart downloads.
```

Getting signed download URLs

Command syntax

coscmd signurl <cospath>

• Sample: generating a signed URL for "doc/picture.jpg"

```
coscmd signurl doc/picture.jpg
```

• Sample: generating a signed URL that is effective for 100 seconds for "doc/picture.jpg"

```
coscmd signurl doc/picture.jpg -t 100
```

Note:

- Replace "cospath" enclosed in "<>" with the COS path of the file for which you need to get the download URL.
- The -t time parameter sets the effective time (in seconds) of the URL signature. The default value is 10000.

Deleting a file or folder

Command syntax for deleting a file

```
coscmd delete <cospath>
```

Note:

Replace "cospath" enclosed in "<>" with the COS path of the file to delete. You will be prompted to confirm this operation.

• Sample: deleting the "doc/exampleobject.txt" file

coscmd delete doc/exampleobject.txt

• Sample: deleting files with version IDs

coscmd delete doc/exampleobject.txt --versionId MTg0NDUxMzc4ODA3NTgyMTErEWN

Command syntax for deleting a folder

coscmd delete -r <cospath>

• Sample: deleting the doc folder

coscmd delete -r doc

• Sample: deleting the folder/doc folder

```
coscmd delete -r folder/doc
```

• Sample: deleting all files with version IDs in the doc directory

```
coscmd delete -r doc/ --versions
```

Note :

- You need to enter y to confirm a batch delete operation. You can skip this step if the ____ parameter is used.
- Note that the delete folder command will delete the current folder as well as the files in it. To delete a versioning-enabled file, you need to specify a version ID.

Viewing incomplete multipart uploads

Command syntax

coscmd listparts <cospath>

Sample: listing incomplete multipart uploads prefixed with "doc/"

```
coscmd listparts doc/
```

Aborting incomplete multipart uploads

Command syntax

coscmd abort

· Sample: aborting all incomplete multipart uploads

coscmd abort

Copying a file or folder

Command syntax for copying a file

```
coscmd copy <sourcepath> <cospath>
```



• Sample (intra-bucket replication): copying "picture.jpg" in the examplebucket-1250000000 bucket to the "doc" folder

```
coscmd -b examplebucket-1250000000 -r ap-chengdu copy examplebucket-1250000000.
ap-chengdu.myqcloud.com/picture.jpg doc/
```

• Sample (cross-bucket replication): copying "doc/picture.jpg" in the examplebucket2-125000000 bucket to "doc/examplefolder/" in the examplebucket1-125000000 bucket

coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy examplebucket2-1250000
000.ap-beijing.myqcloud.com/doc/picture.jpg doc/examplefolder/

• Change the storage class of the file to STANDARD_IA.

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy examplebucket2-1250000
000.ap-beijing.myqcloud.com/doc/picture.jpg doc/examplefolder/ -H "{'x-cos-stor
age-class':'STANDARD_IA'}"
```

• Change the storage class of the file to ARCHIVE and rename it "photo.jpg".

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy examplebucket2-1250000
000.ap-beijing.myqcloud.com/doc/picture.jpg doc/examplefolder/photo.jpg -H "{'x
-cos-storage-class':'Archive'}"
```

Command syntax for copying a folder

coscmd copy -r <sourcepath> <cospath>

• Sample: copying the examplefolder directory in the examplebucket2-1250000000 bucket to the doc directory in the examplebucket1-1250000000 bucket

coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy -r examplebucket2-1250
000000.cos.ap-guangzhou.myqcloud.com/examplefolder doc/

Note :

- Replace "sourcepath" and "cospath" enclosed in "<>" with the path of the COS file to copy and the COS destination path, respectively.
- The source path is formatted as <bucketname-appid>.cos. <region>.myqcloud.com/<cospath> .
- Use the -d parameter to set the x-cos-metadata-directive header. Valid values are Copy (default) and Replaced .
- When setting the HTTP header with the -H parameter, use the JSON format, for example, coscmd copy -H -d Replaced "{'x-cos-storage-class':'Archive', 'Content-Language':'zh-CN'}" <localpath> <cospath> . For more information about the headers, see PUT Object Copy.

Moving a file or folder

Note:

In a move command, <sourcepath> and <cospath> cannot be the same. Otherwise, files will be deleted. The reason is that the move command copies files first and then deletes them, and the files in the <sourcepath> path are eventually deleted.

Command syntax for moving a file

```
coscmd move <sourcepath> <cospath>
```

• Sample (intra-bucket movement): moving "picture.jpg" in the examplebucket-125000000 bucket to the "doc" folder

```
coscmd -b examplebucket-1250000000 -r ap-chengdu move examplebucket-1250000000.
ap-chengdu.myqcloud.com/picture.jpg doc/
```

• Sample (cross-bucket movement): moving "picture.jpg" in the examplebucket2-125000000 bucket to "doc/folder/" in the examplebucket1-125000000 bucket

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou move examplebucket2-1250000
000.ap-beijing.myqcloud.com/picture.jpg doc/folder/
```

• Sample: changing the storage class of the file to STANDARD_IA

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou move examplebucket2-1250000
000.ap-beijing.myqcloud.com/picture.jpg doc/folder/ -H "{'x-cos-storage-clas
s':'STANDARD_IA'}"
```

· Sample: changing the storage class of the file to ARCHIVE

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou move examplebucket2-1250000
000.ap-beijing.myqcloud.com/data/exampleobject data/examplefolder/exampleobject
-H "{'x-cos-storage-class':'Archive'}"
```

Command syntax for moving a folder

coscmd move -r <sourcepath> <cospath>

• Sample: moving the "examplefolder" directory in the examplebucket2-125000000 bucket to the "doc" directory in the examplebucket1-125000000 bucket

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou move -r examplebucket2-1250
000000.cos.ap-guangzhou.myqcloud.com/examplefolder doc/
```

Note:

- Replace "sourcepath" and "cospath" enclosed in "<>" with the path of the COS file to move and the COS destination path, respectively.
- The source path is formatted as <bucketname-appid>.cos. <region>.myqcloud.com/<cospath> .
- Use the -d parameter to set the x-cos-metadata-directive header. Valid values are Copy (default) and Replaced .
- When setting the HTTP header with the _H parameter, use the JSON format, for example, coscmd move _H _d Replaced "{'x-cos-storage-class':'Archive', 'Content-Language':'zh-CN'}" <localpath> <cospath> . For more information about the headers, see PUT Object - Copy.

Setting object access permission

Command syntax

```
coscmd putobjectacl --grant-<permissions> <UIN> <cospath>
```

• Sample: granting 10000000001 permission to read "picture.jpg"

coscmd putobjectacl --grant-read 10000000001 picture.jpg

• Sample: querying the file's access permission

coscmd getobjectacl picture.jpg

Enabling/Suspending versioning

Command syntax

coscmd putbucketversioning <status>

• Sample: enabling versioning

coscmd putbucketversioning Enabled

• Sample: suspending versioning

coscmd putbucketversioning Suspended

• Sample: querying versioning

coscmd getbucketversioning

Note:

- Replace "status" enclosed in "<>" with the desired versioning status.
- Once versioning is enabled for the bucket, it cannot return to the prior status (initial status). However, you can suspend versioning for the bucket so that subsequent uploads of objects will not generate multiple versions.

Restoring an ARCHIVED file

· Command syntax for restoring an archived file

```
coscmd restore <cospath>
```

• Sample: restoring "picture.jpg" using the expedited mode (effective for 3 days)

```
coscmd restore -d 3 -t Expedited picture.jpg
```

Command syntax for restoring archived files

```
coscmd restore -r <cospath>
```

• Sample: restoring the "examplefolder/" folder using the expedited mode (effective for 3 days)

```
coscmd restore -r -d 3 -t Expedited examplefolder/
```

Note:

- Replace "cospath" enclosed in "<>" with the COS path of the file list to query.
- Use -d < day > to set the validity period of the temporary copy. Default value: 7.
- Use -t <tier> to specify the restoration mode. Enumerated values: Expedited , Standard (default), and Bulk .

FAQs

If you have any questions about COSCMD, see COSCMD.

COS Migration

Last updated : 2022-12-19 12:20:02

Feature Overview

COS Migration is an all-in-one tool that integrates the COS data migration feature. You can migrate local data to COS through simple configurations and steps. It has the following features:

- Checkpoint restart: Restarting uploads from checkpoints is supported. For large files, if the upload exits halfway or service failure occurs, you can run the tool again to restart the upload.
- Multipart upload: An object can be uploaded to COS by parts.
- Parallel upload: Multiple objects can be uploaded at the same time.
- Migration verification: Migrated objects can be verified.

Note:

- COS Migration only supports UTF-8 encoding.
- If you use this tool to upload a file that already has the same name, the existing file will be overwritten. You need to configure the tool to skip files with the same name.
- Use the migration service platform preferably for scenarios other than local data migration.
- COS Migration is used for **one-time** migration but is not suitable for continuous sync. For example, if files are added locally every day and need to be continuously synced to COS, then in order to avoid repeated migration tasks, COS Migration will save the records of successful migrations. In case of continuous sync, the record scanning time will keep increasing. We recommend you use COSBrowser as described in User Guide for Desktop Version for this scenario.

Operating Environments

Operating system

Windows, Linux, and macOS.

Software dependency

- JDK 1.8 X64 or above. For more information, see Java Installation and Configuration.
- IFUNC needs to be supported on Linux and the binutils version should be later than 2.20.

How to Use

1. Get the tool

Download COS Migration here.

2. Decompress the package

Windows

Decompress the package and save it to a directory, for example:

C:\Users\Administrator\Downloads\cos_migrate

Linux

Decompress the package and save it to a directory, for example:

unzip cos_migrate_tool_v5-master.zip && cd cos_migrate_tool_v5-master

Migration tool structure

The structure of the properly decompressed COS Migration tool is as follows:

Note:

• The db directory mainly records the IDs of files successfully migrated by the tool. Each migration job will first compare the records in the db directory. If the ID of the current file has already been recorded, the current file will be skipped, otherwise it will be migrated.

• The log directory keeps all the logs generated during tool migration. If an error occurs during migration, first check error.log in this directory.

3. Modify the config.ini file

Before running the migration startup script, modify the config.ini file (path: ./conf/config.ini) first. This file contains the following parts:

3.1 Configure the migration type

type indicates the migration type, which is filled in by users based on their migration needs. For example, to migrate local data to COS, users need to configure type=migrateLocal for [migrateType].

```
[migrateType]
type=migrateLocal
```

Currently, the following migration types are supported:

Migration Type	Description
migrateLocal	From local system to COS

3.2 Configure the migration job

You can configure a migration job based on your actual needs, including information configuration for the destination COS and job-related configurations.

```
# The common configuration section of the migration tool includes account informa
tion to be migrated to the destination COS.
[common]
secretId=COS_SECRETID
secretKey=COS_SECRETKEY
bucketName=examplebucket-125000000
region=ap-guangzhou
storageClass=Standard
cosPath=/
https=off
tmpFolder=./tmp
smallFileThreshold=5242880
smallFileExecutorNum=64
bigFileExecutorNum=8
entireFileMd5Attached=on
daemonMode=off
daemonModeInterVal=60
```

executeTimeWindow=00:00,24:00
outputFinishedFileFolder=./result
resume=false
skipSamePath=false

Name	Description	Default Value
secretId	SecretId of your key. Replace COS_SECRETID with your real key information, which can be obtained on the TencentCloud API key page in the CAM console	-
secretKey	SecretKey of your key. Replace COS_SECRETKEY with your real key information, which can be obtained on the TencentCloud API key page in the CAM console	-
bucketName	Name of the destination bucket in the format of <bucketname-appid> . The bucket name must include the APPID such as examplebucket-1250000000</bucketname-appid>	-
region	Region information of the destination bucket. For the region abbreviations in COS, see Regions and Access Domain Names	-
storageClass	Storage class for the migrated data. Valid values: Standard , Standard_IA , Archive . For more information, see Storage Class Overview.	Standard
cosPath	COS path to migrate to. / indicates to migrate to the root path of the bucket, /folder/doc/ indicates to migrate to in the bucket. If does not exist, a path will be created automatically	/
https	Whether to transfer via HTTPS. on: Yes, off: No. It takes time to enable transfer via HTTPS, which is suitable for scenarios that demand high security.	off



Name	Description	Default Value
tmpFolder	The directory used to store temporary files when data is migrated from another cloud storage service to COS, which will be deleted after the migration is completed. The format must be an absolute path: The separator on Linux is /, such as /a/b/c The separator on Windows is such as $E: \\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $./tmp
smallFileThreshold	Number of bytes as the threshold for small files. If the size is greater than or equal to this threshold, multipart upload is used; otherwise, simple upload is used. The default value is 5 MB	5242880
smallFileExecutorNum	Concurrency for uploading small files (smaller than smallFileThreshold) via simple upload. Decrease the concurrency if files are uploaded to COS via public network with low bandwidth	64
bigFileExecutorNum	Concurrency for uploading large files (greater than or equal to smallFileThreshold) via multipart upload. Decrease the concurrency if files are uploaded to COS via public network with low bandwidth	8
entireFileMd5Attached	The migration tool calculates the MD5 of the entire file and stores it in the custom header "x- cos-meta-md5" of the file for subsequent verification, because the ETag of a large file uploaded to COS via multipart upload is not the MD5 of the entire file	on
daemonMode	Whether to enable daemon mode. on: yes; off: no. In daemon mode, the program will keep performing synchronization. The synchronization interval is configured by the daemonModeInterVal parameter	off
daemonModeInterVal	Time interval in seconds between two rounds of synchronization	60



Name	Description	Default Value
executeTimeWindow	Execution time window with a granularity in minute, which defines the time period when the migration tool runs jobs. For example: 03:30, 21:00 means that jobs will be run between 03:30 and 21:00, and the tool is in sleep mode at other times, when the migration will be paused and the progress will be retained until the next time window when the migration will resume automatically	00:00,24:00
outputFinishedFileFolder	This directory stores the results of successful migration tasks, and result files are named by date, for example, ./result/2021-05- 27.out, where ./result is the directory that is created. Each line in the result files is in the format of "Absolute path"\t"File size"\t"Last modified time" . If outputFinishedFileFolder is left empty, no results will be output.	./result
resume	Whether to continue with the result of the last run and traverse through the list of files from the source. The tool starts from scratch by default.	false
skipSamePath	Whether to skip the current file if a file with the same name already exists in COS. By default, the tool does not skip the current file: it overwrites the existing file.	false

3.3 Configure the data source

```
Configure each section according to the migration type described in [migrateType] . For example, if the configuration item of [migrateType] is type=migrateLocal , users only need to configure the [migrateLocal] section.
```

3.3.1 Configure a local data source migrateLocal

If you migrate from a local system to COS, configure this section. The specific configuration items and descriptions are as follows:

```
# Configuration section for migration from a local system to COS
[migrateLocal]
```

```
localPath=E:\\code\\java\\workspace\\cos_migrate_tool\\test_data
excludes=
```

ignoreModifiedTimeLessThanSeconds=

Configuration Item	Description
localPath	 Absolute path of the local directory Linux uses a slash (/) as the delimiter, for example, /a/b/c. Windows uses two backlashes (\) as the delimiter, for example, E:\\a\\b\\c. Note: You can enter only a directory path but not file path for this parameter; otherwise, an error will occur while parsing the target object name. In the case of cosPath=/, the request will be incorrectly parsed into a bucket creation request.
excludes	Absolute path of the directory or file to be excluded, meaning some directories or files under localPath are not to be migrated. Multiple absolute paths are separated by semicolons. If this is left blank, all files in localPath will be migrated
ignoreModifiedTimeLessThanSeconds	Exclude files that have an update time less than a certain period of time from the current time (in seconds). This item is left blank by default, indicating files are not to be filtered by the time specified by lastmodified. It is suitable for scenarios where you run the migration tool while updating files and don't want files being updated to be migrated to COS. For example, if it is configured as 300, only files updated at least 5 minutes ago will be uploaded.

4. Run the migration tool

Windows

Double-click start_migrate.bat to run the tool

Linux

1. Read the configuration from the config.ini file by running the following command:

sh start_migrate.sh

2. Read the configuration from command lines for some parameters by running the following command:

```
sh start_migrate.sh -Dcommon.cosPath=/savepoint0403_10/
```

Note :

- The tool supports reading configuration items in two ways: command line or configuration file.
- The command line takes priority over the configuration file, i.e., for the same configuration item, parameters in command lines take priority.
- Reading configuration items from command lines allows users to run different migration jobs at the same time, provided that key configuration items (such as bucket name, COS path, source path to be migrated, etc.) in the two jobs are not exactly the same. Concurrent migration can be achieved because different migration jobs are written into different db directories. Refer to db information in the tool structure above.
- Configuration items are in the format of -D{sectionName}.{sectionKey}={sectionValue}.
 sectionName is the section name of the configuration file. sectionKey is the name of the configuration item in the section. sectionValue is the value of the configuration item in the section.
 COS path to which data is migrated to should be in the format of -Dcommon.cosPath=/bbb/ddd.

Migration mechanism and process

Migration mechanism

COS Migration has a status. Successful migrations will be recorded in the format of KV in the leveldb file under the db directory. Before each migration, check whether the path to which data is migrated has been recorded in the db directory. If yes, and its attribute is the same as that in db, the migration will be skipped; otherwise, the migration will be executed. The attribute varies by migration type. For local migration, mtime determines whether to migrate. For migration from other cloud storage services and bucket replication, the etag and length of the source file determine whether to migrate. Therefore, we search for records of successful migrations in the db directory rather than in COS. If a file is deleted or modified via COSCMD or the console rather than the migration tool, the migration tool cannot detect this change, and the file will not be re-migrated.

Migration process

- 1. The configuration file is read, the corresponding configuration section is read according to the migration type, and parameters are checked.
- 2. The IDs of the files to be migrated are scanned and compared in the db directory according to the specified migration type to determine whether upload is allowed.
- 3. The execution results are printed out during migration, where inprogress indicates migration is in progress, skip indicates skipped, fail indicates failed, ok indicates succeeded, and condition_not_match indicates file fails to meet migration conditions (such as lastmodified and excludes) and is skipped.

Details about the failure can be viewed in the error log. The execution process is as shown below:

s sh start migrate.sh
[skip] task info: [taskType: migrateAws] [bucket: chengwus3sdkgz-1251668577] [cosPath: /aws0403 17/aaa/bbbslash.txt]
[DownloadOk] [key: slash.txt] [byteDownload/ byteTotal/ percentage: 6/ 6/ 100.00%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 10240/ 209715200/ 0.00%]
[UploadInProgress] [key: /aws0403_17/slash.txt] [byteSent/ byteTotal/ percentage: 6/ 6/ 100.00%]
[ok] task_info: [taskType: migrateAws] [bucket: chengwus3sdkgz-1251668577] [cosPath: /aws0403_17/slash.txt]
[DownloadInProgress] [key: aws_200M.txt.copy] [byteDownload/ byteTotal/ percentage: 10240/ 209715200/ 0.00%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 236126/ 209715200/ 0.11%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 392798/ 209715200/ 0.19%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 514654/ 209715200/ 0.25%]
[DownloadInProgress] [key: aws_200M.txt.copy] [byteDownload/ byteTotal/ percentage: 96863/ 209715200/ 0.05%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 688734/ 209715200/ 0.33%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 827998/ 209715200/ 0.39%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 949854/ 209715200/ 0.45%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1089118/ 209715200/ 0.52%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1228382/ 209715200/ 0.59%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1315422/ 209715200/ 0.63%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1385054/ 209715200/ 0.66%]
[DownloadInProgress] [key: aws_200M.txt.copy] [byteDownload/ byteTotal/ percentage: 166495/ 209715200/ 0.08%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1506910/ 209715200/ 0.72%]

4. Statistics are printed out after the migration is completed, which include the total number of migrated, failed, and skipped files as well as the amount of time consumed. For failures, check the error log, or rerun the migration job as the migration tool will skip successfully migrated files and retry migrating failed ones. The execution result of a migration job is shown below:

migrateAli over!	op statistics:			
	op_status	:	ALL_OK	
	migrate_ok	:	530038	
	migrate_fail	:	0	
	migrate_skip	:	496264	
	start_time	:	2018-03-19	15:52:02
	end_time	:	2018-03-19	16:54:38
	used_time	:	3756 s	

FAQs

If an exception such as migration failure or execution error occurs when you use the COS Migration, troubleshoot as instructed in COS Migration.

FTP Server

Last updated : 2022-05-06 17:12:58

Overview

COS FTP Server allows you to directly operate on COS objects and directories over FTP protocol, including uploading/downloading/deleting files and creating folders. This tool is developed with Python, which makes the installation easier.

Features

Upload mechanism: streaming without saving uploaded files locally. It works as long as the working directory is configured using the standard FTP protocol, and no actual disk storage capacity is occupied.

Download mechanism: the downloaded file is directly streamed and returned to the client.

Directory mechanism: bucket serves as the root directory of the entire FTP Server, and multiple subdirectories can be created under a bucket.

Binding to multi-buckets: multiple buckets can be bound at the same time.

Note :

Binding to multi-buckets: multiple buckets can be bound via different FTP Server working paths

(home_dir). Therefore, ensure a unique home_dir is assigned to each bucket and user.

Restriction on deletion: the delete_enable option can be configured for each FTP user in the new FTP Server to identify whether the FTP user is allowed to delete files.

```
Supported FTP commands: put , mput , get , rename , delete , mkdir , ls , cd , bye , quite , and size .
```

Unsupported FTP commands: append and mget (The native mget command is not supported, but batch download is allowed on certain Windows clients, such as the FileZilla client.)

Note:

The FTP Server tool does not support checkpoint restart.

Getting Started

Operating system

- OS: Linux. The CVM of Tencent CentOS series is recommended. Windows systems are not supported for now.
- psutil-dependent Linux package: python-devel or python-dev, depending on the names of different Linux distributions. It is added using Linux package manager, such as yum install python-devel or aptitude install python-dev .
- Python interpreter version: Python 2.7. For more information on the installation and configuration, see Python.

Note : FTP Server does not support Python 3.

- Dependent packages:
 - cos-python-sdk-v5 (v1.6.5 or later)
 - pyftpdlib (v1.5.2 or later)
 - psutil (v5.6.1 or later)

Use limits

Applicable to COS XML version.

Installation and Operation

Download the FTP Server tool at cos-ftp-server. The installation steps are as follows:

1. Enter the FTP Server directory, and run setup.py to install FTP Server and its dependent libraries (network required).

```
python setup.py install # Your account should use sudo or have the root permiss
ion.
```

- 2. Copy the configuration sample file conf/vsftpd.conf.example and name it conf/vsftpd.conf . See Configuration File of this document to correctly configure bucket and user information.
- 3. Run ftp_server.py to start FTP Server.

```
python ftp_server.py
```

FTP Server can also be started in the following two ways:

• Execute the nohup command to start it in the backend process:

```
nohup python ftp_server.py >> /dev/null 2>&1 &
```

• Execute the screen command to run it at the backend (you need to install the screen tool):

```
screen -dmS ftp
screen -r ftp
python ftp_server.py
#Use the keyboard shortcut Ctrl + A + D to go back to the main screen.
```

Stop Operation

- If FTP Server is running directly or running at the backend with the screen command, you can stop it with the shortcut key combination Ctrl+C.
- If FTP Server is started with the nohup command, you can stop it by the following way:

```
ps -ef | grep python | grep ftp_server.py | grep -v grep | awk '{print $2}' | x
args -I{} kill {}
```

Configuration File

The configuration sample file for FTP Server is conf/vsftpd.conf.example . Copy and name it vsftpd.conf, and then configure it as follows:

```
[COS_ACCOUNT_0]
cos_secretid = COS_SECRETID # Replaced with your SECRETID
cos_secretkey = COS_SECRETKEY # Replaced with your SECRETKEY
cos_bucket = examplebucket-125000000
cos_region = region # Replaced with your bucket region
cos_protocol = https
#cos_endpoint = region.myqcloud.com
home_dir = /home/user0
ftp_login_user_name=user0 #Replaced with a custom user name
ftp_login_user_password=pass0 #Replaced with a custom password
authority=RW # The user's read and write permissions. R: read; W: write; RW: bot
h.
delete_enable=true # true allows the FTP user to delete files by default; false p
```



rohibits the user to delete files. [COS_ACCOUNT_1] cos_secretid = COS_SECRETID # Replaced with your SECRETID cos_secretkey = COS_SECRETKEY # Replaced with your SECRETKEY cos_bucket = examplebucket-125000000 cos_region = region # Replaced with your bucket region cos_protocol = https #cos_endpoint = region.myqcloud.com home_dir = /home/user1 ftp login user name=user1 #Replaced with a custom user name ftp_login_user_password=pass1 #Replaced with a custom password authority=RW # The user's read and write permissions. R: read; W: write; RW: bot h delete_enable=false # true allows the FTP user to delete files by default; false prohibits the user to delete files. [NETWORK] # If the FTP Server is behind a gateway or NAT, you can use this section to speci fy the gateway's IP address or domain name as the FTP Server's IP address. masquerade_address = XXX.XXX.XXX.XXX # The listening port for FTP Server is 2121 by default. Please note that your WAF needs to ALLOW this port (for example, if you deploy FTP Server on a Tencent Clou d CVM, you need to ALLOW this port in your CVM security group.) listen_port = 2121 # `passive_port` sets the available port range in Passive mode, with a default of [60000, 65535]. Note that your WAF (such as CVM security group) needs to ALLOW th is range. passive_port = 60000,65535 [FILE OPTION] # By default, the maximum size of a single file is 200 G. We do not recommend goi

By derault, the maximum size of a single file is 200 G. We do not recommend going beyond the limit.

single_file_max_size = 21474836480

[OPTIONAL]

For the following settings, take the default settings unless otherwise needed.
Fill in an appropriate integer if necessary.

min_part_size = default

upload_thread_num = default

```
max_connection_num = 512
```

max_list_file = 10000 # The maximum number of files to be listed by `ls` command. It is not recommended to go beyond this limit. Otherwise, high latency of `ls` co mmand will occur.

log_level = INFO # Set the log output level.

log_dir = log # Set the directory to store logs. Default: `log` under the `FTP Se
rver` directory.

Note:

- To bind each user to a unique bucket, simply add the [COS_ACCOUNT_X] section. The section for each COS_ACCOUNT_X is described as follows:
- The username (ftp_login_user_name) and the home directory (home_dir) under each account must be unique, and the home directory must be a directory that exists in the system.
- The number of users logging in to each COS FTP Server simultaneously cannot exceed 100.
- endpoint and region will not take effect at the same time. To use the public cloud COS service, enter the region field correctly. The endpoint is commonly used in the privatized deployment environment. When both 'region and endpoint are entered, endpoint` will take precedence.
- The OPTIONAL part in the configuration file is used to adjust the upload performance for advanced users. You can obtain an optimal uploading speed by reasonably adjusting the part size and the number of concurrent upload threads based on the server performance. General users can keep the default settings without adjustment.

Meanwhile, the limit option for the maximum number of connections is provided. If you do not want to set a limit to it, enter 0, meaning no limit to the maximum number of connections (a reasonable evaluation is required based on your server performance).

- Generally, for the masquerade_address section in your configuration file, we recommend you specify the IP address that your client is using to connect to the COS FTP Server. If you have any questions, please see the FAQs about FTP Server.
- Assume that the FTP Server has more than one IP address, and after running the ifconfig command, you get a private ENI IP 10.xxx.xxx , which is mapped to the public IP 119.xxx.xxx . At this time, if the FTP Server does not explicitly set masquerade_address to the public IP (119.xxx.xxx) that the client uses to access the server, the FTP Server in Passive mode may use the private IP (10.xxx.xxx) to return packets to the client. As a result, the client is able to connect to the FTP Server, but cannot return data packets to the client properly. Therefore, generally speaking, we recommend you to set masquerade_address to the IP address that your client is using to connect to the Server.
- In the configuration file, <u>listen_port</u> sets the listening port for the COS FTP Server, and is defaulted to 2121. <u>passive_port</u> sets the range of data channel listening ports for the COS FTP Server, and is defaulted to [60000, 65535]. When your client connects to the COS FTP Server, ensure that your WAF allows the ports configured in these two sections.

Quick Practice

Accessing COS FTP Server using Linux ftp command

1. Open the Linux command line, and use the command ftp [ip address] [port No.] to connect to the COS FTP Server. Example:

ftp 192.xxx.103 2121

- In the ftp command, the IP field corresponds to the masquerade_address section in the sample configuration file conf/vsftpd.conf.example . In this example, the IP is set to 192.xxx.xx.103 .
- In the ftp command, the port field corresponds to the **listen_port** section in the sample configuration file conf/vsftpd.conf.example . In this example, the port is set to 2121 .
- 2. Run the above command, and you can see **Name** and **Password** to be entered. Copy and paste the contents of the ftp_login_user_name and ftp_login_user_passwordthe sections for COS FTP Server, and the connection will succeed.
- Name: corresponds to ftp_login_user_name (requires configuration) in the sample configuration file conf/vsftpd.conf.example .
- **Password**: corresponds to **ftp_login_user_password** (requires configuration) in the sample configuration file conf/vsftpd.conf.example .

Accessing COS FTP Server using FileZilla

- 1. Download and install FileZilla client.
- 2. After configuring the access information for COS FTP Server on your FileZilla client, click Quick Connect.
- Host (H): corresponds to masquerade_address in the sample configuration file

conf/vsftpd.conf.example . In this example, the IP is set to 192.xxx.xx.103 .

Note :

If the COS FTP Server is behind a gateway or NAT, you can use this section to specify the gateway's IP address or domain name as the Server's IP address.

- Username (U): corresponds to ftp_login_user_name (requires configuration) in the sample configuration file conf/vsftpd.conf.example .
- **Password (W):** corresponds to **ftp_login_user_password** (requires configuration) in the sample configuration file conf/vsftpd.conf.example .
- Port (P): corresponds to listen_port in the sample configuration file conf/vsftpd.conf.example . In this
 example, the port is set to 2121 .



FAQs

If any error occurs or you have any question on the upload limit while using FTP Server, see FTP Server FAQs.

COSFS

Last updated : 2022-11-30 14:59:51

Feature Overview

COSFS allows you to mount COS buckets to local and work with the COS objects as you do with a local file system. COSFS supports the following features:

- Most features of the POSIX file system, such as reading/writing files, operations on directories/links, permission management, and uid/gid management.
- Multipart upload of large files.
- Data verification with MD5.
- Data upload to COS using COS Migration or COSCMD.

Limitations

COSFS is built on S3FS. As disks are required for COSFS's read and write operations, COSFS is only suitable for simple management of the mounted files and does not support all features of a local file system. Therefore, you are advised to access COS through Tencent Cloud Storage Gateway (CSG). Tencent CSG can mount COS buckets to multiple servers as network file systems. Users can use the POSIX file protocol to read and write objects in COS through mount points. The COSFS tool is not applicable to some scenarios, for example:

- Randomly writing data or appending data to a file may lead to the re-download/re-upload of the entire file. To avoid this, you can use a CVM in the same region as the bucket to accelerate the upload and download.
- When a COS bucket is mounted to multiple clients, you need to coordinate the behaviors of these clients, for example, to prevent the clients from simultaneously writing data to the same file.
- Rename operation on a file/folder is not atomic.
- For metadata operations such as list directory, COSFS performs unsatisfactorily as it requires remote access to the COS server.
- COSFS does not support hard links and is inapplicable to high-concurrency reads/writes.
- Mounting and unmounting files cannot be performed on the same mount point at the same time. You can use the
 cd command to switch to another directory and then mount and unmount the files at the mount point.

Operating Environments

Mainstream Ubuntu, CentOS, SUSE, and macOS

Installation

You can install COSFS with an installation package or by compiling the source code.

Method 1: Install with an installation package

Note :

This installation method supports only mainstream Ubuntu and CentOS.

Ubuntu

 Download the appropriate installation package according to your system version. Currently, Ubuntu 14.04, 16.04, 18.04, and 20.04 are supported.

Download from GitHub:

```
#Ubuntu14.04
sudo wget https://github.com/tencentyun/cosfs/releases/download/v1.0.20/cosfs_
1.0.20-ubuntu14.04_amd64.deb
#Ubuntu16.04
sudo wget https://github.com/tencentyun/cosfs/releases/download/v1.0.20/cosfs_
1.0.20-ubuntu16.04_amd64.deb
#Ubuntu18.04
sudo wget https://github.com/tencentyun/cosfs/releases/download/v1.0.20/cosfs_
1.0.20-ubuntu18.04_amd64.deb
#Ubuntu20.04
sudo wget https://github.com/tencentyun/cosfs/releases/download/v1.0.20/cosfs_
1.0.20-ubuntu20.04
```

Download from CDN:

```
cosfs_1.0.20-ubuntu14.04_amd64.deb
cosfs_1.0.20-ubuntu16.04_amd64.deb
cosfs_1.0.20-ubuntu18.04_amd64.deb
cosfs_1.0.20-ubuntu20.04_amd64.deb
```

2. Install the package. The following takes Ubuntu 16.04 as an example.

```
sudo dpkg -i cosfs_1.0.20-ubuntu16.04_amd64.deb
```



CentOS

1. Install dependencies.

```
sudo yum install libxml2-devel libcurl-devel -y
```

2. Download the appropriate installation package according to your system version. Currently, CentOS 6.5 and 7.0 are supported.

Download from GitHub:

```
#CentOS6.5
sudo wget https://github.com/tencentyun/cosfs/releases/download/v1.0.20/cosfs-
1.0.20-centos6.5.x86_64.rpm
#CentOS7.0
sudo wget https://github.com/tencentyun/cosfs/releases/download/v1.0.20/cosfs-
1.0.20-centos7.0.x86_64.rpm
```

Download from CDN:

cosfs-1.0.20-centos6.5.x86_64.rpm

cosfs-1.0.20-centos7.0.x86_64.rpm

3. Install the package. The following takes CentOS 7.0 as an example.

sudo rpm -ivh cosfs-1.0.20-centos7.0.x86_64.rpm

Note :

If the system reports the error conflicts with file from package fuse-libs-* during installation, add the --force parameter and install the package again.

Method 2: Install by compiling the source code

Note :

This installation method supports mainstream Ubuntu, CentOS, SUSE, and macOS.

1. Install the dependency software

The compilation and installation of COSFS depend on software packages such as automake , git ,

libcurl-devel, libxml2-devel, fuse-devel, make, and openssl-devel. The following describes how to install dependency software on Ubuntu, CentOS, SUSE, and macOS:

• Install dependency software on the Ubuntu system:

```
sudo apt-get install automake autotools-dev g++ git libcurl4-gnutls-dev libfuse
-dev libssl-dev libxml2-dev make pkg-config fuse
```

Install dependency software on the CentOS system:

```
sudo yum install automake gcc-c++ git libcurl-devel libxml2-devel fuse-devel ma
ke openssl-devel fuse
```

• Install dependency software on the SUSE system:

```
sudo zypper install gcc-c++ automake make libcurl-devel libxml2-devel openssl-d
evel pkg-config
```

• Install dependency software on the macOS system:

```
brew install automake git curl libxml2 make pkg-config openssl brew install cask osxfuse
```

2. Obtain the source code

Download the COSFS Source Code from GitHub to a specified directory. The following uses /usr/cosfs as an example. You can use another directory as needed.

sudo git clone https://github.com/tencentyun/cosfs /usr/cosfs

3. Compile and install COSFS

Open the installation directory, and execute the following command to compile and install COSFS:

```
cd /usr/cosfs
sudo ./autogen.sh
sudo ./configure
sudo make
```
🔗 Tencent Cloud

sudo make install
cosfs --version #View the COSFS version number

4. Troubleshoot configure issues

Messages displayed during the configure operation vary depending on the OS. If your FUSE version is earlier than 2.8.4, the following error message will be displayed:

```
checking for common_lib_checking... configure: error: Package requirements (fuse
>= 2.8.4 libcurl >= 7.0 libxml-2.0 >= 2.6) were not met:
Requested 'fuse >= 2.8.4' but version of fuse is 2.8.3
```

In this case, you need to manually install fuse 2.8.4 or later as shown below:

```
sudo yum -y remove fuse-devel
yum -y remove fuse-devel84%97sudo wget https://github.com/libfuse/libfuse/release
s/download/fuse_2_9_4/fuse-2.9.4.tar.gz
tar -zxvf fuse-2.9.4.tar.gz
cd fuse-2.9.4
sudo ./configure
sudo make
sudo make
sudo make install
export PKG_CONFIG_PATH=/usr/lib/pkgconfig:/usr/lib64/pkgconfig/:/usr/local/lib/pk
gconfig
modprobe fuse # Mount FUSE's kernel module.
echo "/usr/local/lib" >> /etc/ld.so.conf
ldconfig # Update the dynamic-link library.
pkg-config --modversion fuse #View the fuse version number. If "2.9.4" is display
ed, fuse 2.9.4 is installed successfully.
```

Install fuse 2.8.4 or later on the SUSE system manually, as shown below:

```
Note :
During installation, you need to comment out the content of line 222 in <code>example/fusermp.c</code> by using /*content*/ . Otherwise, an error will be reported when you use Make.
```

```
zypper remove fuse libfuse2
yum -y remove fuse-devel84%97sudo wget https://github.com/libfuse/libfuse/relea
ses/download/fuse_2_9_4/fuse-2.9.4.tar.gz
tar -zxvf fuse-2.9.4.tar.gz
cd fuse-2.9.4
```

sudo ./configure
sudo make
sudo make install
export PKG_CONFIG_PATH=/usr/lib/pkgconfig:/usr/lib64/pkgconfig/:/usr/local/lib/
pkgconfig
modprobe fuse # Mount FUSE's kernel module.
echo "/usr/local/lib" >> /etc/ld.so.conf
ldconfig # Update the dynamic-link library.
pkg-config --modversion fuse #View the fuse version number. If "2.9.4" is displ
ayed, fuse 2.9.4 is installed successfully.

• When the "configure" operation is performed on macOS, the following may be displayed:

```
configure: error: Package requirements (fuse >= 2.7.3 libcurl >= 7.0 libxml-2.0
>2.6 libcrypto >= 0.9) were not met
No package 'libcrypto' found
```

In this case, you need to set the variable PKG_CONFIG_PATH, so that the pkg-config tool can find openssl. The command is as follows:

brew info openssl export PKG_CONFIG_PATH=/usr/local/opt/openssl/lib/pkgconfig #You may need to mo dify this command based on the message displayed for the previous command.

How to Use

1. Configure the key file

Write the bucket information in the /etc/passwd-cosfs file, including the bucket name (in BucketName-APPID format) <SecretId>, as well as <SecretKey>, and use colons (:) to separate them. To avoid compromising your key, set permissions for the key file to 640. Run the following command to configure the /etc/passwdcosfs key file:

```
sudo su # Switch to the root account to modify the /etc/passwd-cosfs file. Skip t
his step if you have already logged in with the root account
echo <BucketName-APPID>:<SecretId>:<SecretKey> > /etc/passwd-cosfs
chmod 640 /etc/passwd-cosfs
```

Note :

You need to replace the content enclosed in <> with the actual information.

- <BucketName-APPID> indicates the name of the bucket. For more information, see Bucket Naming Conventions.
- <SecretId> and <SecretKey> are information about the key, which can be obtained and created at Manage API Key in the CAM console.
- You can configure the key in <code>\$HOME/.passwd-cosfs</code> . Alternatively, you can run <code>-opasswd_file=</code> [path] to specify the directory of the key file and then set permissions of the key file to 600.

Example:

```
echo examplebucket-1250000000:AKIDHTVVaVR6e3****:PdkhT9e2rZCfy6**** > /etc/passwd
-cosfs
chmod 640 /etc/passwd-cosfs
```

Note:

If your COSFS version is v1.0.5 or earlier, the configuration file format is <BucketName>:<SecretId>: <SecretKey>.

2. Run the tool

Run the following command to mount the bucket configured in the key file to a specified directory:

```
cosfs <BucketName-APPID> <MountPoint> -ourl=http://cos.<Region>.myqcloud.com -odb
glevel=info -oallow_other
```

On the Stream Interruption Records page:

- <MountPoint> is the mount point, for example, /mnt .
- <Region> is the abbreviation for the region, such as ap-guangzhou and eu-frankfurt . For more information about region abbreviations, see Regions and Access Endpoints.
- -odbglevel specifies the log level. The default value is crit . Available options are crit , error , warn , info , and debug .
- -oallow_other allows other users to access the mount folder.

Example:

```
mkdir -p /mnt/cosfs
cosfs examplebucket-1250000000 /mnt/cosfs -ourl=http://cos.ap-guangzhou.myqcloud.
com -odbglevel=info -onoxattr -oallow_other
```

Note :

- To improve performance, COSFS uses the system disk by default for the temporary cache of uploaded and downloaded files and releases space after files are closed. When a large number of concurrent files are opened or large files are read or written, COSFS uses hard disk space as much as possible to improve performance. By default, only 100 MB of free hard disk space is reserved for other applications. You can use the oensure_diskfree=[size] option to set the size of available hard disk space in MB reserved by COSFS. For example, __oensure_diskfree=1024 indicates that COSFS will reserve 1024 MB of free space.
- If your COSFS is v1.0.5 or earlier, use the following mount command: cosfs
 <APPID>:<BucketName> <MountPoint> -ourl=<CosDomainName> oallow_other .

3. Unmount a bucket

Unmount a bucket using the following commands:

Method 1: Use `fusermount -u /mnt, fusermount` to unmount a FUSE file system Method 2: Use `umount -l /mnt`. The unmount operation will be performed when no p rogram is using any file in the file system. Method 3: Use `umount /mnt`. If any program is using a file in the file system du ring the unmount, an error will be reported.

Common Mounting Options

-omultipart_size=[size]

Specifies the size (in MB) of each part for the multipart upload. It is 10 MB by default. Up to 10,000 parts are allowed for a file in a multipart upload. If the file is larger than 100 GB (10 MB * 10000), you need to adjust this parameter accordingly.

-oallow_other

Allows other users to access the folder to which the bucket is mounted.



-odel_cache

By default, to ensure optimal performance, the COSFS does not clear local cached data after a bucket is unmounted. To enable the COSFS to automatically clear cached data upon its exit, you can add this option during mounting.

-onoxattr

Disables getattr/setxattr. For the COSFS earlier than 1.0.9, you cannot set or obtain extended attributes. If the use_xattr option is used during mounting, the files may fail to be copied to the bucket.

-opasswd_file=[path]

Specifies the path for the COSFS key file. You need to set the permission for the key file to 600.

-odbglevel=[dbg|info|warn|err|crit]

Sets the log level for COSFS. Valid values: info , dbg , warn , err , crit . You are advised to set it to info in the production environment, and dbg for debugging. If you do not clear system logs regularly, or numerous logs will be generated due to a huge access volume, you can set it to err or crit .

-oumask=[perm]

Removes the permission of a specified type of users to operate files in the mounting destination directory. For example, when -oumask=755, the permission for the mounting destination directory is changed to 022.

-ouid=[uid]

Allows the user whose ID is [uid] to access all the files in the mounting destination directory without being restricted by the file permission bits.

You can obtain the uid of a user using the ID command id -u username. For example, you can run id -u user_00 to obtain the uid of user_00.

-oensure_diskfree=[size]

To improve performance, COSFS uses the system disk by default for the temporary cache of uploaded and downloaded files and releases space after files are closed. When a large number of concurrent files are opened or large files are read or written, COSFS uses hard disk space as much as possible to improve performance. By default, only 100 MB of free hard disk space is reserved for other applications. You can use the <code>oensure_diskfree=</code> [size] option to set the size of available hard disk space in MB reserved by COSFS. For example, - <code>oensure_diskfree=1024</code> indicates that COSFS will reserve 1024 MB of free space.

FAQs

If you have any questions about COSFS, see COSFS FAQs.

Hadoop

Last updated : 2022-08-04 15:47:47

Feature Description

Hadoop-COS implements a standard Hadoop file system on the Tencent Cloud COS platform. It helps integrate COS with big data computing frameworks such as Hadoop, Spark, and Tez, so that they can read and write COS data as they do with HDFS file systems.

Since Hadoop-COS uses COSN (a Tencent Cloud big data tool) as its URI scheme, it can also be referred to as a COSN-based file system.

Operating Environment

Operating system

Windows, Linux, or macOS.

Software dependency

Hadoop 2.6.0 or later.

Note :

- 1. Hadoop-COS has been integrated into Apache Hadoop 3.3.0. For more information, see Integration of Tencent COS in Hadoop.
- 2. If your version is earlier than Hadoop 3.3.0, or CDH has integrated the Hadoop-COS JAR package, you need to restart NodeManager to load the JAR package.
- 3. To build a JAR package of a specified Hadoop version, modify hadoop.version in the pom file.

Download and Installation

Getting Hadoop-COS JAR package and dependencies

Download Hadoop-COS from GitHub.

Installing Hadoop-COS plugin



1. Copy hadoop-cos-{hadoop.version}-{version}.jar and cos_api-bundle-{version}.jar to
 \$HADOOP_HOME/share/hadoop/tools/lib .

Note :

Select the JAR package that corresponds to your Hadoop version. If you cannot find the desired JAR package in the release, manually build and generate one by modifying the Hadoop version number in the pom file.

2. Modify the hadoop-env.sh file under the \$HADOOP_HOME/etc/hadoop directory by adding the COSN JAR package to your Hadoop environment variables as follows:

```
for f in $HADOOP_HOME/share/hadoop/tools/lib/*.jar; do
if [ "$HADOOP_CLASSPATH" ]; then
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
else
export HADOOP_CLASSPATH=$f
fi
done
```

Configuration Method

Configuration item description

Attribute Key	Description	Default Va
fs.cosn.userinfo. secretId/secretKey	The API key for your account. Log in to the CAM console to view the key.	None

Attribute Key	Description	Default Va
fs.cosn. credentials.provider	 The way to get SecretId and SecretKey . Currently, eight ways are supported: 1. org.apache.hadoop.fs.auth.SessionCredentialProvider: Gets them from the request URI in the format of cosn://{secretId}: {secretKey}@examplebucket-125000000/. 2. org.apache.hadoop.fs.auth.SimpleCredentialProvider: Gets them by reading fs.cosn.userinfo.secretId and fs.cosn.userinfo.secretKey in the core-site.xml configuration file. 3. org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider: Gets them from system variables Cos_SECRET_ID and COS_SECRET_KEY. 4. org.apache.hadoop.fs.auth.SessionTokenCredentialProvider: Accesses by using a temporary key as described in Generating and Using Temporary Keys. 5. org.apache.hadoop.fs.auth.CVMInstanceCredentialProvider: Gets a temporary key to access COS by using the role bound to CVM. 6. org.apache.hadoop.fs.auth.EMRInstanceCredentialProvider: Gets a temporary key to access COS by using the role bound to CPM. 7. org.apache.hadoop.fs.auth.EMRInstanceCredentialProvider: Gets a temporary key to access COS by using the role bound to CPM. 8. org.apache.hadoop.fs.auth.RangerCredentialsProvider: Gets a temporary key to access COS by using the role bound to EMR. 8. org.apache.hadoop.fs.auth.RangerCredentialsProvider: Gets a temporary key to access COS by using the role bound to EMR. 	If this para 1. org.apa 2. org.apa 3. org.apa 4. org.apa 5. org.apa 7. org.apa
fs.cosn.useHttps	Whether to use HTTPS as the transfer protocol for the COS backend.	true
fs.cosn.impl	The COSN implementation class for FileSystem , which is fixed at org.apache.hadoop.fs.CosFileSystem .	None
fs.AbstractFileSystem. cosn.impl	The COSN implementation class for AbstractFileSystem , which is fixed at org.apache.hadoop.fs.CosN .	None
fs.cosn.bucket.region	The region of the bucket to access. For enumerated values, see the region abbreviations in Regions and Access Endpoints, such as apbeijing and ap-guangzhou. This parameter is compatible with the old parameter fs.cosn.userinfo.region.	None



Attribute Key	Description	Default Va
fs.cosn.bucket. endpoint_suffix	The COS endpoint to connect (optional). Public cloud COS users only need to provide the correct region in the parameter above. This parameter is compatible with the old parameter fs.cosn.userinfo.endpoint_suffix . To make endpoint take effect, you should delete the fs.cosn.bucket.region parameter first.	None
fs.cosn.tmp.dir	An existing local directory where temporary files generated at runtime are stored.	/tmp/hadc
fs.cosn.upload. part.size	The size of each part for multipart upload through the COSN file system. A COS multipart upload supports a maximum of 10,000 parts to be uploaded for a single object. You need to estimate the desired part size as needed. For example, if the part size is set to 8388608 (8 MB), you can upload an object of up to 78 GB in size. The size of a part can be up to 2 GB, that is, the size of a single object can be up to 19 TB.	8388608
fs.cosn. upload.buffer	The type of buffer used when files are uploaded through COSN. Currently, there are three types of buffers: non_direct_memory, direct_memory, and mapped_disk. The non-direct memory buffer uses JVM on-heap memory, the direct_memory buffer uses off-heap memory, and the mapped_disk buffer works based on memory file mapping.	mapped_
fs.cosn. upload.buffer.size	The size of buffer used during upload through COSN. A value of -1 means unlimited. You can specify this value only if you set the buffer type to mapped_disk . If you specify a value greater than 0, it cannot be smaller than the block size. This parameter is compatible with the old parameter fs.cosn.buffer.size .	-1
fs.cosn.block.size	The size of a block in the COSN file system.	13421772
fs.cosn. upload_thread_pool	The number of concurrent threads when files are uploaded to COS through streams.	10
fs.cosn. copy_thread_pool	The number of threads used to copy and delete concurrent files during directory replication.	3
fs.cosn. read.ahead.block.size	The size of a read-ahead block.	1048576



Attribute Key	Description	Default Va
fs.cosn. read.ahead.queue.size	The length of the read-ahead queue.	8
fs.cosn.maxRetries	The maximum number of retries if an error occurs when accessing COS.	200
fs.cosn.retry. interval.seconds	The time interval between retries in seconds.	3
fs.cosn. server-side- encryption.algorithm	The COS server-side encryption algorithm. Valid values: SSE-C, SSE-COS. If this parameter is left empty (default value), no encryption algorithm will be used.	None
fs.cosn. server-side- encryption.key	The required SSE-C key if the SSE-C server encryption algorithm is used. This parameter is a Base64-encoded AES-256 key. If this parameter is left empty (default value), no encryption key will be used.	None
fs.cosn. crc64.checksum.enabled	Whether to enable CRC-64 checksum. It is disabled by default, meaning that you can't run the hadoop fs -checksum command to obtain the CRC-64 checksum of a file.	false
fs.cosn. crc32c.checksum.enabled	Whether to enable CRC32C checksum. It is disabled by default, meaning that you cannot run the hadoop fs -checksum command to obtain the CRC32C checksum of a file. CRC-64 and CRC32C cannot be both enabled.	false
fs.cosn.traffic.limit	The limit on the upload bandwidth in bits/s. Value range: 819200– 838860800. Default value: -1 (unlimited).	None

Hadoop configuration

Modify \$HADOOP_HOME/etc/hadoop/core-site.xml by adding the information of COS users and implementation classes as shown below:

```
<configuration>
<property>
<name>fs.cosn.credentials.provider</name>
<value>org.apache.hadoop.fs.auth.SimpleCredentialProvider</value>
<description>
This option allows the user to specify how to get the credentials.
Comma-separated class names of credential provider classes which implement
com.qcloud.cos.auth.COSCredentialsProvider:
1.org.apache.hadoop.fs.auth.SessionCredentialProvider: Obtain the secret id and s
```



```
ecret key from the URI: cosn://secretId:secretKey@examplebucket-1250000000/;
2.org.apache.hadoop.fs.auth.SimpleCredentialProvider: Obtain the secret id and se
cret key
from fs.cosn.userinfo.secretId and fs.cosn.userinfo.secretKey in core-site.xml;
3.org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider: Obtain the sec
ret id and secret key
from system environment variables named COS_SECRET_ID and COS_SECRET_KEY.
If unspecified, the default order of credential providers is:
1. org.apache.hadoop.fs.auth.SessionCredentialProvider
2. org.apache.hadoop.fs.auth.SimpleCredentialProvider
3. org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider
4. org.apache.hadoop.fs.auth.SessionTokenCredentialProvider
5. org.apache.hadoop.fs.auth.CVMInstanceCredentialsProvider
6. org.apache.hadoop.fs.auth.CPMInstanceCredentialsProvider
7. org.apache.hadoop.fs.auth.EMRInstanceCredentialsProvider
</description>
</property>
<property>
<name>fs.cosn.userinfo.secretId</name>
<value>xxxxxxxxxxxxxxxxxxxxxxxxxxxx/value>
<description>Tencent Cloud Secret Id</description>
</property>
<property>
<name>fs.cosn.userinfo.secretKey</name>
<value>xxxxxxxxxxxxxxxxxxxxxx/value>
<description>Tencent Cloud Secret Key</description>
</property>
<property>
<name>fs.cosn.bucket.region</name>
<value>ap-xxx</value>
<description>The region where the bucket is located.</description>
</property>
<property>
<name>fs.cosn.bucket.endpoint_suffix</name>
<value>cos.ap-xxx.myqcloud.com</value>
<description>
COS endpoint to connect to.
For public cloud users, it is recommended not to set this option, and only the co
rrect area field is required.
</description>
</property>
```

```
<property>
```



<name>fs.cosn.impl</name> <value>org.apache.hadoop.fs.CosFileSystem</value> <description>The implementation class of the CosN Filesystem.</description> </property> <property> <name>fs.AbstractFileSystem.cosn.impl</name> <value>org.apache.hadoop.fs.CosN</value> <description>The implementation class of the CosN AbstractFileSystem.</descriptio</pre> n> </property> <property> <name>fs.cosn.tmp.dir</name> <value>/tmp/hadoop_cos</value> <description>Temporary files will be placed here.</description> </property> <property> <name>fs.cosn.upload.buffer</name> <value>mapped_disk</value> <description>The type of upload buffer. Available values: non_direct_memory, dire ct_memory, mapped_disk</description> </property> <property> <name>fs.cosn.upload.buffer.size</name> <value>134217728</value> <description>The total size of the upload buffer pool. -1 means unlimited.</descr</pre> iption> </property> <property> <name>fs.cosn.upload.part.size</name> <value>8388608</value> <description>Block size to use cosn filesysten, which is the part size for Multip artUpload. Considering the COS supports up to 10000 blocks, user should estimate the maximum size of a single file. For example, 8MB part size can allow writing a 78GB single file.</description> </property> <property> <name>fs.cosn.maxRetries</name> <value>3</value> <description> The maximum number of retries for reading or writing files to COS, before we signal failure to the application. </description> </property>



```
<property>
<name>fs.cosn.retry.interval.seconds</name>
<value>3</value>
<description>The number of seconds to sleep between each COS retry.</description>
</property>
<name>fs.cosn.server-side-encryption.algorithm</name>
<value></value>
<description>The server side encryption algorithm.</description>
</property>
<name>fs.cosn.server-side-encryption.key</name>
<value></value>
<description>The SE-C server side encryption key.</description>
</property>
```

```
</configuration>
```

We recommend you not configure fs.defaultFS in the production environment. To use it for certain test cases such as hive-testbench , add the following configurations:

```
<property>
<name>fs.defaultFS</name>
<value>cosn://examplebucket-125000000</value>
<description>
This option is not advice to config, this only used for some special test cases.
</description>
</property>
```

Server-side encryption

Hadoop-COS supports server-side encryption through either COS-managed key (SSE-COS) or user-defined key (SSE-C). This feature is disabled by default, and you can enable it by configuring as follows.

SSE-COS encryption

SSE-COS encryption refers to server-side encryption with a COS-managed key. In this mode, COS manages the master key and its data. When using Hadoop-COS, you can add the following configuration in the

\$HADOOP_HOME/etc/hadoop/core-site.xml file to implement SSE-COS encryption.

```
<property>
<name>fs.cosn.server-side-encryption.algorithm</name>
<value>SSE-COS</value>
```



<description>The server side encryption algorithm.</description>
</property>

SSE-C encryption

SSE-C encryption refers to server-side encryption with a user-defined key. In this mode, the encryption key is provided by you. When you upload an object, COS will use the encryption key to apply AES-256 encryption to the data. When using Hadoop-COS, you can add the following configuration in the \$HADOOP_HOME/etc/hadoop/coresite.xml
file to implement SSE-C encryption.

```
<property>
<name>fs.cosn.server-side-encryption.algorithm</name>
<value>SSE-C</value>
<description>The server side encryption algorithm.</description>
</property>
<property>
<name>fs.cosn.server-side-encryption.key</name>
<value>MDEyMzQ1Njc4OUFCQ0RFRjAxMjMONTY3OD1BQkNERUY=</value> # You need to configu
re the SSE-C key in the format of a Base64-encoded AES-256 key.
<description>The SSE-C server side encryption key.</description>
</property>
```

Note :

- The SSE-C encryption feature of Hadoop-COS relies on the SSE-C server-side encryption of COS. This
 means Hadoop-COS does not store any user-defined encryption keys just like COS. Instead, COS stores
 HMAC values with random data added to the encryption keys to authenticate access requests. COS cannot
 use the HMAC values to derive the value of an encryption key or decrypt the content of an object. Therefore,
 if you lose your encryption key, you will not be able to access the object again.
- If you configure an SSE-C server-side encryption algorithm in Hadoop-COS, you must also configure an SSE-C key by using the fs.cosn.server-side-encryption.key parameter in the format of a Base64-encoded AES-256 key.

Directions

Samples

Run a command in the format of hadoop fs -ls -R cosn://<bucketname-appid>/<path> or hadoop fs -ls -R /<path> (you need to set fs.defaultFS to cosn://BucketName-APPID). The following



example uses a bucket named examplebucket-1250000000, to which you can append a specific path.

```
hadoop fs -ls -R cosn://examplebucket-125000000/
-rw-rw-rw- 1 root root 1087 2018-06-11 07:49 cosn://examplebucket-1250000000/LICE
NSE
drwxrwxrwx - root root 0 1970-01-01 00:00 cosn://examplebucket-1250000000/hdfs
drwxrwxrwx - root root 0 1970-01-01 00:00 cosn://examplebucket-1250000000/hdfs/20
18
-rw-rw-rw- 1 root root 1087 2018-06-12 03:26 cosn://examplebucket-1250000000/hdf
s/2018/LICENSE
-rw-rw-rw- 1 root root 2386 2018-06-12 03:26 cosn://examplebucket-1250000000/hdf
s/2018/ReadMe
drwxrwxrwx - root root 0 1970-01-01 00:00 cosn://examplebucket-1250000000/hdfs/te
st
-rw-rw-rw- 1 root root 1087 2018-06-11 07:32 cosn://examplebucket-1250000000/hdf
s/test/LICENSE
-rw-rw-rw- 1 root root 2386 2018-06-11 07:29 cosn://examplebucket-1250000000/hdf
s/test/ReadMe
```

Run the WordCount program provided by MapReduce and run the following command.

Note:

This example uses hadoop-mapreduce-examples-2.7.2.jar . To use a different version of the JAR file, modify the version number.

bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcou
nt cosn://example/mr/input cosn://example/mr/output3

If the command is successful, the following information will be returned:

```
File System Counters

COSN: Number of bytes read=72

COSN: Number of bytes written=40

COSN: Number of read operations=0

COSN: Number of large read operations=0

FILE: Number of bytes read=547350

FILE: Number of bytes written=1155616

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=0
```

```
HDFS: Number of bytes written=0
HDFS: Number of read operations=0
HDFS: Number of large read operations=0
HDFS: Number of write operations=0
Map-Reduce Framework
Map input records=5
Map output records=7
Map output bytes=59
Map output materialized bytes=70
Input split bytes=99
Combine input records=7
Combine output records=6
Reduce input groups=6
Reduce shuffle bytes=70
Reduce input records=6
Reduce output records=6
Spilled Records=12
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=0
Total committed heap usage (bytes)=653262848
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=36
File Output Format Counters
Bytes Written=40
```

Accessing COSN through Java code

package com.qcloud.chdfs.demo; import org.apache.commons.io.IOUtils; import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.fs.FSDataInputStream; import org.apache.hadoop.fs.FSDataOutputStream; import org.apache.hadoop.fs.FileChecksum; import org.apache.hadoop.fs.FileStatus; import org.apache.hadoop.fs.FileStatus; import org.apache.hadoop.fs.FileSystem; import org.apache.hadoop.fs.Path; import java.io.IOException; 🔗 Tencent Cloud

```
import java.net.URI;
import java.nio.ByteBuffer;
public class Demo {
private static FileSystem initFS() throws IOException {
Configuration conf = new Configuration();
// For more information on COSN configuration items, visit https://cloud.tencent.
com/document/product/436/6884#hadoop-.E9.85.8D.E7.BD.AE.
// The following configuration items are required.
conf.set("fs.cosn.impl", "org.apache.hadoop.fs.CosFileSystem");
conf.set("fs.AbstractFileSystem.cosn.impl", "org.apache.hadoop.fs.CosN");
conf.set("fs.cosn.tmp.dir", "/tmp/hadoop_cos");
conf.set("fs.cosn.bucket.region", "ap-guangzhou");
conf.set("fs.cosn.userinfo.secretId", "AKXXXXXXXXXXXXXXXXXXXXXX);
conf.set("fs.cosn.userinfo.secretKey", "XXXXXXXXXXXXXXXXXXXXXXX);
conf.set("fs.ofs.user.appid", "XXXXXXXXXX");
// For more information on other configuration items, visit https://cloud.tencen
t.com/document/product/436/6884#hadoop-.E9.85.8D.E7.BD.AE.
// Whether to enable CRC-64 checksum. It is disabled by default, meaning that you
can't run the `hadoop fs -checksum` command to obtain the CRC-64 checksum of a fi
le.
conf.set("fs.cosn.crc64.checksum.enabled", "true");
String cosnUrl = "cosn://f4mxxxxxx-125xxxxxx";
return FileSystem.get(URI.create(cosnUrl), conf);
}
private static void mkdir (FileSystem fs, Path filePath) throws IOException {
fs.mkdirs(filePath);
}
private static void createFile (FileSystem fs, Path filePath) throws IOException {
// Create a file (if it already exists, it will be overwritten)
// if the parent dir does not exist, fs will create it!
FSDataOutputStream out = fs.create(filePath, true);
try {
// Write a file
String content = "test write file";
out.write(content.getBytes());
} finally {
IOUtils.closeQuietly(out);
}
}
private static void readFile (FileSystem fs, Path filePath) throws IOException {
FSDataInputStream in = fs.open(filePath);
try {
byte[] buf = new byte[4096];
int readLen = -1;
do {
readLen = in.read(buf);
} while (readLen >= 0);
```

```
} finally {
IOUtils.closeQuietly(in);
}
private static void queryFileOrDirStatus (FileSystem fs, Path path) throws IOExcep
tion {
FileStatus fileStatus = fs.getFileStatus(path);
if (fileStatus.isDirectory()) {
System.out.printf("path %s is dir\n", path);
return;
}
long fileLen = fileStatus.getLen();
long accessTime = fileStatus.getAccessTime();
long modifyTime = fileStatus.getModificationTime();
String owner = fileStatus.getOwner();
String group = fileStatus.getGroup();
System.out.printf("path %s is file, fileLen: %d, accessTime: %d, modifyTime: %d,
owner: %s, group: %s\n",
path, fileLen, accessTime, modifyTime, owner, group);
}
private static void getFileCheckSum(FileSystem fs, Path path) throws IOException
FileChecksum checksum = fs.getFileChecksum(path);
System.out.printf("path %s, checkSumType: %s, checkSumCrcVal: %d\n",
path, checksum.getAlgorithmName(), ByteBuffer.wrap(checksum.getBytes()).getInt()
);
}
private static void copyFileFromLocal (FileSystem fs, Path cosnPath, Path localPat
h) throws IOException {
fs.copyFromLocalFile(localPath, cosnPath);
}
private static void copyFileToLocal (FileSystem fs, Path cosnPath, Path localPath)
throws IOException {
fs.copyToLocalFile(cosnPath, localPath);
private static void renamePath (FileSystem fs, Path oldPath, Path newPath) throws
IOException {
fs.rename(oldPath, newPath);
private static void listDirPath(FileSystem fs, Path dirPath) throws IOException {
FileStatus[] dirMemberArray = fs.listStatus(dirPath);
for (FileStatus dirMember : dirMemberArray) {
System.out.printf("dirMember path %s, fileLen: %d\n", dirMember.getPath(), dirMem
ber.getLen());
}
```



```
// The recursive deletion flag is used to delete directories.
// If recursion is `false` and `dir` is not empty, the operation will fail.
private static void deleteFileOrDir (FileSystem fs, Path path, boolean recursive)
throws IOException {
fs.delete(path, recursive);
}
private static void closeFileSystem(FileSystem fs) throws IOException {
fs.close();
}
public static void main(String[] args) throws IOException {
// Initialize a file
FileSystem fs = initFS();
// Create a file
Path cosnFilePath = new Path("/folder/exampleobject.txt");
createFile(fs, cosnFilePath);
// Read a file
readFile(fs, cosnFilePath);
// Query a file or directory
queryFileOrDirStatus(fs, cosnFilePath);
// Get the checksum of a file
getFileCheckSum(fs, cosnFilePath);
// Copy a file from the local system
Path localFilePath = new Path("file:///home/hadoop/ofs_demo/data/exampleobject.tx
t");
copyFileFromLocal(fs, cosnFilePath, localFilePath);
// Download a file to the local file system
Path localDownFilePath = new Path("file:///home/hadoop/ofs_demo/data/exampleobjec
t.txt");
copyFileToLocal(fs, cosnFilePath, localDownFilePath);
listDirPath(fs, cosnFilePath);
// Rename a file
mkdir(fs, new Path("/doc"));
Path newPath = new Path("/doc/example.txt");
renamePath(fs, cosnFilePath, newPath);
// Delete a file
deleteFileOrDir(fs, newPath, false);
// Create a directory
Path dirPath = new Path("/folder");
mkdir(fs, dirPath);
// Create a file in a directory
Path subFilePath = new Path("/folder/exampleobject.txt");
createFile(fs, subFilePath);
// List directories
listDirPath(fs, dirPath);
// Delete a directory
deleteFileOrDir(fs, dirPath, true);
deleteFileOrDir(fs, new Path("/doc"), true);
```

```
// Close a file system
closeFileSystem(fs);
}
```

FAQs

If you have any questions about Hadoop-COS, see Hadoop.

COSDistCp

Last updated : 2022-05-04 17:30:50

Overview

COSDistCp is a MapReduce-based distributed file copy tool mainly used for data copy between HDFS and COS. It introduces the following features:

- Performs incremental file migration and data verification based on length and CRC checksum.
- Filters files in the source directory with regular expression.
- Decompresses files in the source directory and compresses them to the target compression format.
- Aggregates text files based on a regular expression.
- Preserves user/user group, extension attributes, and time of the source file and directory.
- Configures alarms and Prometheus monitoring.
- Collects the statistics of file size distribution.
- Limits the read bandwidth.

Operating Environments

Operating system

Linux

Software dependency

Hadoop 2.6.0 or above; Hadoop-COS 5.9.3 or above

Download and Installation

Obtaining the COSDistCp JAR package

- If your Hadoop version is 2.x, you can download cos-distcp-1.10-2.8.5.jar and verify the integrity of the downloaded JAR package according to the MD5 checksum of the package.
- If your Hadoop version is 3.x, you can download cos-distcp-1.10-3.1.0.jar and verify the integrity of the downloaded JAR package according to the MD5 checksum of the package.

Installation notes

In the Hadoop environment, install Hadoop-COS and then run the COSDistCp tool.

How It Works

COSDistCp uses the MapReduce framework. The multi-process and multi-thread tool performs operations such as file copy, data verification, compression, file attribute preservation, and copy retries. COSDistCp will overwrite files with the same name in the destination location. If data copy or verification fails, the corresponding file may fail to be copied and information about these files will be written in a temporary directory. If new files are added to your source file system or the file content changes, you can use the ___skipMode or ___diffMode parameter to compare the length or CRC checksum of the files to implement data verification and incremental file migration.

Parameters

You can run the hadoop jar cos-distcp-\${version}.jar --help (\${version} is the version number) command to view the COSDistCp-supported parameters. The following table describes the parameters of the COSDistCp of the current version:

Attribute Key	Description	Default Value
help	Outputs parameters supported by COSDistCp. Example:help	None
src=LOCATION	Location of the data to copy. This can be either an HDFS or COS location. Example:src=hdfs://user/logs/	None
dest=LOCATION	Destination for the data. This can be either an HDFS or COS location. Example:dest=cosn://examplebucket- 125000000/user/logs	None
srcPattern=PATTERN	A regular expression that filters files in the source location. Example:srcPattern='.*\.log\$' *Note: Enclose your parameter in single quotation marks (') in case asterisks () are parsed by the shell.**	None
taskNumber=VALUE	Number of copy threads Example:taskNumber=10	10

Attribute Key	Description	Default Value
workerNumber=VALUE	Number of copy threads. COSDistCp will create a copy thread pool for each copy process based on this value set. Example: workerNumber=4	4
filesPerMapper=VALUE	The number of files input to each mapper. Example:filesPerMapper=10000	500000
groupBy=PATTERN	A regular expression to concatenate text files that match the regular expression. Example:groupBy='.*group-input/(\d+)- (\d+).*'	None
targetSize=VALUE	The size (in MB) of the files to create. This parameter is used together with	None
outputCodec=VALUE	Compression method of output file. Valid values: gzip , lzo , snappy , none , keep . Here: 1. keep indicates to keep the compression method of the original file. 2. none indicates to decompress the file based on the file extension. Example:outputCodec=gzip Note: if the /dir/test.gzip and /dir/test.gz files exist, and you specify the output format as lzo , only /dir/test.lzo will be retained.	keep
deleteOnSuccess	Deletes the source file immediately after it is successfully copied to the destination directory. Example:deleteOnSuccess Note: v1.7 and above no longer provide this parameter. We recommend you delete the data in the source file system after migrating the data successfully and usingdiffMode for verification.	false



Attribute Key	Description	Default Value
multipartUploadChunkSize=VALUE	The size (in MB) of the multipart upload part transferred to COS using the Hadoop-COS plugin. COS supports up to 10,000 parts. You can set the value based on the file size. Example:multipartUploadChunkSize=20	8
cosServerSideEncryption	Specifies whether to use SSE-COS for encryption on the COS server side. Example:cosServerSideEncryption	false
outputManifest=VALUE	Creates a file (Gzip compressed) that contains a list of all files copied to the destination location. Example:outputManifest=manifest.gz	None
requirePreviousManifest	If this parameter is set to true, previousManifest=VALUE must be specified for incremental copy. Example:requirePreviousManifest	false
previousManifest=LOCATION	A manifest file that was created during the previous copy operation. Example: previousManifest=cosn://examplebucket- 125000000/big-data/manifest.gz	None
copyFromManifest	Copies files specified in previousManifest to the destination file system. This is used together with previousManifest=LOCATION . Example:copyFromManifest	false
storageClass=VALUE	The storage class to use. Valid values are STANDARD, STANDARD_IA, ARCHIVE, DEEP_ARCHIVE, and INTELLIGENT_TIERING. For more information, please see Storage Class Overview.	None
srcPrefixesFile=LOCATION	A local file that contains a list of source directories, one directory per line. Example:srcPrefixesFile=file:///data/migrate- folders.txt	None



Attribute Key	Description	Default Value
skipMode=MODE	Verifies whether the source and destination files are the same before the copy. If they are the same, the file will be skipped. Valid values are none (no verification), length , checksum , length-mtime , and length-checksum . Example:skipMode=length	length-checksum
checkMode=MODE	Verifies whether the source and destination files are the same when the copy is completed. Valid values are none (no verification), length, checksum, length- mtime, and length-checksum. Example:checkMode=length-checksum	length-checksum
diffMode=MODE	Specifies the rule for obtaining the list of different files in the source and destination directories. Valid values are length , checksum , length-mtime , and length-checksum . Example:diffMode=length-checksum	None
diffOutput=LOCATION	Specifies the HDFS output directory in diffMode. This directory must be empty. Example:diffOutput=/diff-output	None
cosChecksumType=TYPE	Specifies the CRC algorithm used by the Hadoop-COS plugin. Valid values are CRC32C and CRC64. Example:cosChecksumType=CRC32C	CRC32C
preserveStatus=VALUE	Specifies whether to copy the user , group , permission , xattr , and timestamps metadata of the source file to the destination file. Valid values are any combinations of letters u, g, p, x, and t (initials of user , group , permission , xattr , and timestamps , respectively). Example:preserveStatus=ugpt	None
ignoreSrcMiss	Ignores files that exist in the manifest file but cannot be found during the copy.	false



Attribute Key	Description	Default Value
promGatewayAddress=VALUE	Specifies the Prometheus PushGateway address and port for pushing the counter data of MapReduce jobs.	None
 promGatewayDeleteOnFinish=VALUE	Whether to delete JobName metrics from Prometheus PushGateway when the specified job is completed. Example:promGatewayDeleteOnFinish=true	true
promGatewayJobName=VALUE	JobName to report to Prometheus PushGateway Example:promGatewayJobName=cos- distcp-hive-backup	None
promCollectInterval=VALUE	Interval to collect MapReduce jobs, in ms Example:promCollectInterval=5000	5000
promPort=VALUE	Server port to expose Prometheus metrics Example:promPort=9028	None
enableDynamicStrategy	Enables the dynamic task assignment policy to make tasks with quicker migration migrate more files. Note: this mode has certain limits; for example, the task counter may be inaccurate if the process is exceptional. Therefore, usediffMode to verify the data after migration. Example:enableDynamicStrategy	false
splitRatio=VALUE	Split ratio of the dynamic strategy. A higher splitRatio indicates a smaller job granularity. Example:splitRatio=8	8
localTemp=VALUE	Local folder to store the job files generated by the dynamic strategy Example:localTemp=/tmp	/tmp
taskFilesCopyThreadNum=VALUE	Number of concurrency to copy the job files generated by the dynamic strategy to the HDFS Example:taskFilesCopyThreadNum=32	32

Attribute Key	Description	Default Value
statsRange=VALUE	Statistics range Example: statsRange=0,1mb,10mb,100mb,1gb,10gb,inf	0,1mb,10mb,100mb,
printStatsOnly	Collects only statistics on the file size distribution without copying the data. Example:printStatsOnly	None
bandWidth	Maximum bandwidth for reading each migrated file (in MB/s). Default value: -1, which indicates no limit on the read bandwidth. Example:bandWidth=10	None
jobName	Migration task name. Example:jobName=cosdistcp-to-warehouse	None
compareWithCompatibleSuffix	Whether to change the source file extension gzip to gz and lzop to lzo when using the skipMode anddiffMode parameters. Example:compareWithCompatibleSuffix	None

Examples

Viewing the help option

Run the following command with _-help to view the parameters supported by COSDistCp:

```
hadoop jar cos-distcp-${version}.jar --help
```

In the command above, \${version} is the version ID of the COSDistCp. For example, the name of the COSDistCp JAR package (version 1.0) is cos-distcp-1.0.jar.

File size distribution of the files to copy

Run the following command with the --printStatsOnly and --statsRange=VALUE parameters to output the file size distribution of the files to copy:

```
hadoop jar cos-distcp-${version}.jar --src /wookie/data --dest cosn://examplebuck
et-1250000000/wookie/data --printStatsOnly --statsRange=0,1mb,10mb,100mb,1gb,10g
b,inf
Copy File Distribution Statistics:
```

```
Total File Count: 4

Total File Size: 1190133760

| SizeRange | TotalCount | TotalSize |

| OMB ~ 1MB | 0(0.00%) | 0(0.00%) |

| 1MB ~ 10MB | 1(25.00%) | 10485760(0.09%) |

| 10MB ~ 100MB | 1(25.00%) | 104857600(0.88%) |

| 100MB ~ 1024MB | 1(25.00%) | 104857600(8.81%) |

| 1024MB ~ 10240MB | 1(25.00%) | 1073741824(90.22%) |

| 10240MB ~ LONG_MAX| 0(0.00%) | 0(0.00%) |
```

Specifying the source and destination locations for the files to copy

Run the following command with the --src and --dest parameters:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-1250000000/data/warehouse
```

COSDistCp will retry 5 times for files that failed to be copied. If the copy still fails, these files will be written to the /tmp/\${randomUUID}/output/failed/ directory, where \${randomUUID} is a random string. After recording the failed file information, COSDistCp will continue to migrate the remaining files, and the migration task will not fail due to the migration failure of some files. When the migration task is completed, COSDistCp will output counter information (ensure that your task submitting machine is configured with INFO log output for MapReduce jobs on the submission end) and determine whether there are files that failed to be migrated, and if yes, it will throw an exception on the client that submitted the task.

The following information about a source file might be contained in the output:

1. SRC_MISS: The copy fails because the source file contained in the manifest is not found.

2. COPY_FAILED: The copy fails due to other reasons.

You can run the copy command again to implement incremental migration. Run the following command to obtain the log of the MapReduce job. In this way, you can find out the cause of the copy failure. Note that application_1610615435237_0021 is the application ID.

```
yarn logs -applicationId application_1610615435237_0021 > application_16106154352
37_0021.log
```

Querying counters

When the copy operation ends, statistics on the copy will be output. The counters are as follows:

```
CosDistCp Counters
BYTES_EXPECTED=10198247
```



```
BYTES_SKIPPED=10196880
FILES_COPIED=1
FILES_EXPECTED=7
FILES_FAILED=1
FILES_SKIPPED=5
```

The statistics are described as follows:

Statistics Item	Description
BYTES_EXPECTED	Total size (in bytes) to copy according to the source directory
FILES_EXPECTED	Number of files to copy according to the source directory, including the directory itself
BYTES_SKIPPED	Total size (in bytes) of files that can be skipped (same length or checksum value)
FILES_SKIPPED	Number of source files that can be skipped (same length or checksum value)
FILES_COPIED	Number of source files that are successfully copied
FILES_FAILED	Number of source files that failed to be copied
FOLDERS_COPIED	Number of directories that are successfully copied
FOLDERS_SKIPPED	Number of directories that are skipped

Specifying the number of copy processes and the number of threads in each process

Run the following command with the --taskNumber and --workersNumber parameters. COSDistCp adopts a multi-process, multi-thread framework for the copy operation. You can:

- Use --taskNumber to specify the number of processes.
- Use --workerNumber to specify the number of threads in each copy process.

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/ --dest cosn://example
bucket-1250000000/data/warehouse --taskNumber=10 --workerNumber=5
```

Skipping files with the same check value for incremental migration

Run the following command with the <u>--skipMode</u> parameter to skip copying source files with the same length and checksum as those of destination files. The default value is <u>length-checksum</u> :



```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-125000000/data/warehouse --skipMode=length-checksum
```

--skipMode is used to verify whether the source and destination files are the same before the copy. If they are the same, the file will be skipped. Valid values are none (no verification), length , checksum , and length-checksum (length + CRC checksum).

If the checksum algorithms of the source and destination file systems are different, the source file will be read for calculating a new checksum. If your source is HDFS, you can identify whether the HDFS source supports the COMPOSITE-CRC32C algorithm as follows:

hadoop fs -Ddfs.checksum.combine.mode=COMPOSITE_CRC -checksum /data/test.txt /data/test.txt COMPOSITE-CRC32C 6a732798

Verifying data after migration and migrating incremental data

Run the command with the --diffMode and --diffOutput parameters:

- --diffMode can be set to length or length-checksum .
- --diffMode=length obtains the list of different files based on whether the file sizes are the same.
- --diffMode=length-checksum obtains the list of different files based on whether the file size and CRC checksum are the same.
- --diffOutput specifies the output directory for the diff operation.

If the destination file system is COS and the CRC algorithm of the source file system is different from that of COS, COSDistCp will pull the source file to calculate the CRC checksum of the destination file system and compare the CRC checksums to check whether they are the same. In the following sample code, the --diffMode parameter is used to check whether the source and destination files are the same based on the file size and CRC checksum after migration.

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-1250000000/data/warehouse/ --diffMode=length-checksum --diffOutput=/tmp/dif
f-output
```

After the above command is executed successfully, the counter information based on the file list of the source file system will be output (ensure that your task submitting machine is configured with INFO log output for MapReduce jobs on the submission end). You can analyze whether the source and destination files are the same based on the counter information as detailed below:

- 1. SUCCESS: The source and destination files are the same.
- 2. DEST_MISS: The destination file does not exist.

3. SRC_MISS: The source file contained in the source file manifest is not found during the verification.

- 4. LENGTH_DIFF: Sizes of the source and destination files are different.
- 5. CHECKSUM_DIFF: CRC checksums of the source and destination files are different.
- 6. DIFF_FAILED: The diff operation fails due to insufficient permissions or other reasons.
- 7. TYPE_DIFF: The source is a directory but the destination is a file.

In addition, COSDistCp will generate a list of different files in the /tmp/diff-output/failed directory in HDFS (or /tmp/diff-output for v1.0.5 or earlier versions). You can run the following command to obtain the list of different files except for those recorded as SRC_MISS:

```
hadoop fs -getmerge /tmp/diff-output/failed diff-manifest
grep -v '"comment":"SRC_MISS"' diff-manifest |gzip > diff-manifest.gz
```

Run the following command to implement incremental copy based on the list of different files:

```
hadoop jar cos-distcp-${version}.jar --taskNumber=20 --src /data/warehouse --dest
cosn://examplebucket-125000000/data/warehouse/ --previousManifest=file:///usr/lo
cal/service/hadoop/diff-manifest.gz --copyFromManifest
```

After incremental migration is completed, run the command with the <u>--diffMode</u> parameter again to check whether the files are completely identical.

Verifying whether the source and destination files have the same CRC checksum

Run the command with the <u>--checkMode</u> parameter to check whether the source and destination files have the same length and checksum after file copy is completed. The default value is <u>length-checksum</u>.

When you are copying files from a non-COS file system to COS, if the CRC algorithms of the source and Hadoop-COS are different, the CRC checksum will be calculated during the copy operation. When the copy operation is completed, the CRC checksum of the destination file will be obtained and compared with the calculated CRC checksum of the source file.

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-1250000000/data/warehouse --checkMode=length-checksum
```

Note:

It takes effect if --groupBy is not specified and --outputCodec is the default value.

Restricting the read bandwidth for a single file

Run the command with the <u>--bandWidth</u> parameter (in MB). The following command example restricts the read bandwidth of each copied file to 10 MB/s:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-1250000000/data/warehouse --bandWidth=10
```

Copying multiple directories

You can create a local file (for example, srcPrefixes.txt) and add the absolute paths of multiple directories to copy to the file (the directories cannot be in parent-child relationships). After this, you can run the cat command to view the directories as follows:

```
cat srcPrefixes.txt
/data/warehouse/20181121/
/data/warehouse/20181122/
```

Then, you can use --srcPrefixesFile to specify this file. The command is as follows:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --srcPrefixesFile fil
e:///usr/local/service/hadoop/srcPrefixes.txt --dest cosn://examplebucket-1250000
000/data/warehouse/ --taskNumber=20
```

Filtering source files with a regular expression

Run the following command with the --srcPattern parameter. In this example, only files whose extension is ".log" in the /data/warehouse/ directory are copied.

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/ --dest cosn://example
bucket-1250000000/data/warehouse --srcPattern='.*\.log$'
```

Do not copy files whose extension is ".temp" or ".tmp":

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/ --dest cosn://example
bucket-1250000000/data/warehouse/ --srcPattern='.*(?<!\.temp|\.tmp)$'</pre>
```

Specifying the checksum type of Hadoop-COS

Run the following command with the --cosChecksumType parameter. Valid values are CRC32C (default) and CRC64 .

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-1250000000/data/warehouse --cosChecksumType=CRC32C
```

Specifying the storage class for COS objects

```
Run the following command with the --storageClass parameter:
```

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-1250000000/data/warehouse/ --outputManifest=manifest-2020-01-10.gz --storag
eClass=STANDARD_IA
```

Specifying the output compression codec

Run the command with the --outputCodec parameter, which allows you to compress HDFS data to COS in real time to reduce storage costs. Valid values are keep, none, gzip, lzop, and snappy. If the parameter is set to none, the files will be copied uncompressed. If it is set to keep, the files will be copied with no change in their compression. The following is an example:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/logs --dest cosn://exa
mplebucket-125000000/data/warehouse/logs-gzip --outputCodec=gzip
```

Note:

If the parameter is not set to keep, the files will be decompressed and converted to the target compression format. Due to the difference in compression parameters, the content of the destination files might be different from that of the source files, but the files will be the same after decompression. If --groupBy is not specified and --outputCodec is the default value, you can use --skipMode to perform incremental migration and --checkMode to perform data verification.

Deleting the source files

Run the command with the --deleteOnSuccess parameter. The following example deletes the corresponding source files in the /data/warehouse directory immediately after they are copied from HDFS to COS:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-125000000/data/warehouse --deleteOnSuccess
```

Note:

If <u>--deleteOnSuccess</u> is specified, each source file is deleted immediately after the file is copied, but not after all source files are copied. The parameter is not provided in version 1.7 or later.

Generating the target manifest and specifying the previous manifest

Run the command with the --outputManifest and --previousManifest parameters.

- --outputManifest generates a local manifest.gz (Gzip compressed) file. When the copy operation is successful, the file is moved to the directory specified in --dest .
- --previousManifest specifies the destination files that are copied during the previous copy operation (-- outputManifest). COSDistCp will skip files of the same size.

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-125000000/data/warehouse/ --outputManifest=manifest.gz --previousManifest=
cosn://examplebucket-125000000/data/warehouse/manifest-2020-01-10.gz
```

Note:

The command above performs incremental copy only. Only files with size changes can be copied. If the file content is changed, you can refer to the example of --diffMode and determine the changed manifest files based on the CRC checksum.

Using dynamic strategy for migration job distribution

If your files differ greatly in size, (e.g., there are a few large files, causing imbalanced loads of a large number of small files and machines), you can use --enableDynamicStrategy to enable the dynamic strategy, which allows faster-speed jobs to copy more files to speed up the whole copy process.

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-125000000/data/warehouse --enableDynamicStrategy
```

Verify the migrated data after migration is completed:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-1250000000/data/warehouse/ --diffMode=length-checksum --diffOutput=/tmp/dif
f-output
```

Note:

This mode has certain limits; for example, the task counter may be inaccurate if the process is exceptional. Therefore, use --diffMode to verify the data after migration.

Copying metadata of the source file

Run the following command with the --preserveStatus parameter to copy the user, group,

permission, and timestamps (modification time and access time) metadata of the source file/directory to the destination file/directory. The parameter takes effect when files are copied from HDFS to CHDFS.

Sample:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-125000000/data/warehouse/ --preserveStatus=ugpt
```

Configuring Prometheus

You can go to YARN to view the COSDistCp job counter, including the number of files/bytes that have been copied. To easily view the graph of the COSDistCp jobs, you can display the data using Prometheus and Grafana with easy configurations. The following example configures prometheus.yml to add the jobs to grab:

```
- job_name: 'cos-distcp-hive-backup'
static_configs:
- targets: ['172.16.16.139:9028']
```

Run the command with the --promPort=VALUE parameter to expose the counter of the current MapReduce job:

hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://exampleb ucket-1250000000/data/warehouse --promPort=9028

Download the sample Grafana Dashboard and import it. The Grafana dashboard will be as follows:



Alarms for copy failures

Run the command with the --completionCallbackClass parameter to specify the path of the callback class. When the task is completed, COSDistCp will use the collected task information as parameters to execute the callback function. For user-defined callback functions, the following APIs need to be implemented. You can download the callback sample code.

```
package com.qcloud.cos.distcp;
import java.util.Map;
public interface TaskCompletionCallback {
/**
* @description: When the task is completed, the callback function is executed
* @param jobType Copy or Diff
* @param jobStartTime the job start time
* @param errorMsg the exception error msg
* @param applicationId the MapReduce application id
* @param: cosDistCpCounters the job
*/
void doTaskCompletionCallback(String jobType, long jobStartTime, String errorMsg,
String applicationId, Map<String, Long> cosDistCpCounters);
/**
* @description: init callback config before execute
*/
void init() throws Exception;
}
```

COSDistCp has integrated the alarms of Cloud Monitor. When the task runs abnormally or some files fail to be copied, alarming will be performed.

```
export alarmSecretId=SECRET-ID
export alarmSecretKey=SECRET-KEY
export alarmRegion=ap-guangzhou
export alarmModule=module
export alarmPolicyId=cm-xxx
hadoop jar cos-distcp-1.4-2.8.5.jar \
-Dfs.cosn.credentials.provider=org.apache.hadoop.fs.auth.SimpleCredentialProvider
\backslash
-Dfs.cosn.userinfo.secretId=SECRET-ID \
-Dfs.cosn.userinfo.secretKey=SECRET-KEY \
-Dfs.cosn.bucket.region=ap-guangzhou \
-Dfs.cosn.impl=org.apache.hadoop.fs.CosFileSystem \
-Dfs.AbstractFileSystem.cosn.impl=org.apache.hadoop.fs.CosN \
--src /data/warehouse \
--dest cosn://examplebucket-1250000000/data/warehouse/ \
--checkMode=checksum \
--completionCallbackClass=com.qcloud.cos.distcp.DefaultTaskCompletionCallback
```

alarmPolicyId in the command above is an alarm policy created in Cloud Monitor. You can go to the Cloud Monitor console (Alarm Management > Alarm Configuration > Custom Messages) to create and configure one.
FAQs

What stages are involved in migration of HDFS data with COSDistCp? How do I adjust the migration performance and ensure the data correctness?

COSDistCp verifies each migrated file upon migration completion according to checkMode :

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-1250000000/data/warehouse --taskNumber=20
```

After migration is completed, you can also run the following command to view the list of different source and destination files:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://exampleb
ucket-1250000000/data/warehouse/ --diffMode=length-checksum --diffOutput=/tmp/dif
f-output
```

How can I run COSDistCp if Hadoop-COS is not configured in the environment?

You can download a specific version of the COSDistCp JAR package according to the Hadoop version and specify the Hadoop-COS-related parameters to perform the copy operation.

```
hadoop jar cos-distcp-${version}.jar \
-Dfs.cosn.credentials.provider=org.apache.hadoop.fs.auth.SimpleCredentialProvider
\
-Dfs.cosn.userinfo.secretId=COS_SECRETID \
-Dfs.cosn.userinfo.secretKey=COS_SECRETKEY \
-Dfs.cosn.bucket.region=ap-guangzhou \
-Dfs.cosn.impl=org.apache.hadoop.fs.CosFileSystem \
-Dfs.AbstractFileSystem.cosn.impl=org.apache.hadoop.fs.CosN \
--src /data/warehouse \
--dest cosn://examplebucket-125000000/warehouse
```

What do I do if the result shows that some files failed to be copied?

COSDistCp will retry 5 times for IOException occurred during the copy process. If the copy still fails, information about the failed files will be written to the /tmp/\${randomUUID}/output/failed/ directory, where

\${randomUUID} is a random string. Common reasons for the copy failure are as follows:

- 1. The source file contained in the copy manifest is not found during the copy (recorded as SRC_MISS).
- The job initiator does not have permission to read the source file or write the destination file, or the copy failed due to other reasons (recorded as COPY_FAILED).

If the log message indicates that the source file does not exist, and the source file is ignorable, you can run the following command to obtain the list of different files except for those recorded as SRC_MISS:

```
hadoop fs -getmerge /tmp/${randomUUID}/output/failed/ failed-manifest
grep -v '"comment":"SRC_MISS"' failed-manifest |gzip > failed-manifest.gz
```

Except for those recorded as SRC_MISS, if there are other failed files, you can locate the failure reasons by referring to the error log messages in the /tmp/\${randomUUID}/output/logs/ directory and pulling the application logs. The following command example pulls the logs of the YARN application:

```
yarn logs -applicationId application_1610615435237_0021 > application_16106154352
37_0021.log
```

In the command above, application_1610615435237_0021 is the application ID.

Will COSDistCp generate incomplete files due to network or other exceptions?

If the network is abnormal, the source file is missing, or the permissions are insufficient, COSDistCp cannot generate a destination file with the same size as the source file.

- For versions earlier than COSDistCp 1.5, COSDistCp will attempt to delete the destination files generated. If the deletion fails, you need to re-execute the copy task to overwrite the incomplete files, or manually delete them.
- If your COSDistCp version is 1.5 or later and the version of the Hadoop-COS plugin in the running environment is 5.9.3 or later, when files fail to be copied to COS, COSDistCp will call the abort API to terminate the ongoing upload request. Therefore, no incomplete file will be generated even if an exception occurs.
- If your COSDistCp version is 1.5 or later but the version of Hadoop-COS in the running environment is earlier than 5.9.3, you are advised to upgrade it to 5.9.3 or later.
- If the destination location is not COS, COSDistCp will attempt to delete the destination files.

There are some invisible incomplete multipart uploads in COS buckets, which occupy storage space. How do I deal with them?

COS buckets may have some incomplete multipart uploads occupying storage space due to events such as server exception and process kill. You can configure an incomplete multipart upload deletion rule as instructed in Setting Lifecycle to clear them.

A memory overflow and task timeout occurred during migration. How do I adjust parameters?

During migration, both COSDistCp and the tools used to access COS and CHDFS, based on their own logic, occupy some memory. To avoid memory overflow and task timeout, you can adjust parameters of some MapReduce jobs, for example:

```
hadoop jar cos-distcp-${version}.jar -Dmapreduce.task.timeout=18000 -Dmapreduce.r
educe.memory.mb=8192 --src /data/warehouse --dest cosn://examplebucket-1250000000
/data/warehouse
```

As shown in the example above, the value of mapreduce.task.timeout is changed to 18,000 seconds to avoid job timeout when large files are copied, and the value of mapreduce.reduce.memory.mb (memory size of the Reduce process) is changed to 8 GB to avoid memory overflow.

How do I control the migration bandwidth of the migration task through migration over Direct Connect?

The formula for calculating the total bandwidth limit of COSDistCp migration is: taskNumber x workerNumber x bandWidth. You can set workerNumber to 1, use the taskNumber parameter to control the number of concurrent migrations, and use the bandWidth parameter to control the bandwidth of a single concurrent migration.

Hadoop-cos-DistChecker

Last updated : 2022-05-04 12:35:56

Overview

After migrating data from HDFS to COS by using the hadoop distcp command, you can use the Hadoop-cos-DistChecker tool to verify the integrity of the migrated directory. Based on the parallel processing capabilities of MapReduce, it can quickly check the **source directory** against the **destination directory**.

Operating Environment

- Hadoop-cos above v5.8.2 (for details, see hadoop-cos release.)
- Hadoop MapReduce runtime environment

Note :

- If you are using a self-built Hadoop cluster, the Hadoop-cos dependency should be of the latest version (with GitHub release as 5.8.2 or above) to obtain the CRC64 checksum.
- If you are using Tencent Cloud EMR, only clusters created after May 8, 2020 contain the Hadoop-cos version above. To deal with earlier clusters, please contact us.

Directions

Since Hadoop-cos-distchecker needs to get CRC64 checksum for files from Hadoop-cos (CosN file system) before running, you should first configure fs.cosn.crc64.checksum.enabled to true to do so. Once this tool finishes, configure the value back to false to stop getting CRC64 checksum.

Note :

The CRC64 checksum in Hadoop-COS is not compatible with the CRC32C checksum in HDFS. Therefore, after using this tool, be sure to set the above parameter to false. Otherwise, Hadoop-COS may fail to run in some scenarios where the file system getFileChecksum API is called.



Parameter description

• Source file list: a list of subdirectories and files obtained by running the following command:

```
hadoop fs -ls -R hdfs://host:port/{source_dir} | awk '{print $8}' > check_list.
txt
```

Its format is as follows:

```
/benchmarks/TestDFSIO
/benchmarks/TestDFSIO/io_control
/benchmarks/TestDFSIO/io_control/in_file_test_io_0
/benchmarks/TestDFSIO/io_data
/benchmarks/TestDFSIO/io_data/test_io_0
/benchmarks/TestDFSIO/io_data/test_io_1
/benchmarks/TestDFSIO/io_write
/benchmarks/TestDFSIO/io_write/_SUCCESS
/benchmarks/TestDFSIO/io_write/part-00000
```

• Source directory: the directory where the source files are stored; it usually serves as the source path for data migration through the distcp command. For example, hdfs://host:port/source_dir is the source directory in the following sample:

```
hadoop distcp hdfs://host:port/source_dir cosn://examplebucket-appid/dest_dir
```

This is also the common parent directory in the **source file path list**, such as / benchmarks in the sample above.

• Destination directory: the destination directory to check against.

Command line syntax

Hadoop-cos-DistChecker is a MapReduce job-based program, and can be submitted just like a MapReduce job:

hadoop jar hadoop-cos-distchecker-2.8.5-1.0-SNAPSHOT.jar com.qcloud.cos.hadoop.di stchecker.App <Absolute path of the source file list> <Absolute path representati on of the source directory> <Absolute path representation of the destination dire ctory> [optional parameters]



Note:

The "optional parameters" are for Hadoop.

Directions

The example below describes how to use this tool by checking hdfs://10.0.0.3:9000/benchmarks against cosn://hdfs-test-1250000000/benchmarks .

First, run the following command:

```
hadoop fs -ls -R hdfs://10.0.0.3:9000/benchmarks | awk '{print $8}' > check_list.
txt
```

#@VM_38_97_centos ~]\$ hadoop fs -ls -R hdfs://10.0.0.3:9000/benchmarks | awk '{print \$8}' > check_list.txt

Export all the source paths to be checked to a check_list.txt file which stores the list of source file paths, as shown below:

Then, run the following command to put check_list.txt to the HDFS:

hadoop fs -put check_list.txt hdfs://10.0.0.3:9000/

[put check_list.txt hdfs://10.0.0.3:9000/
[/ 🛶 ,u@VM_38_97_centos ~]\$ hadoop fs -	ls hdfs://10.0.0.3:9000/
Found 2 items	
drwxr-xr-x - ia 🚛 supergroup	0 2020-02-20 22:21 hdfs://10.0.0.3:9000/benchmarks
-rw-rr 2 ia supergroup	580 2020-02-21 17:59 hdfs://10.0.0.3:9000/check_list.txt

Run the Hadoop-cos-DistChecker to check hdfs://10.0.0.3:9000/benchmarks against cosn://hdfs-test-test-1250000000/benchmarks , and output the result to the cosn://hdfs-test-



125000000/check_result path by using the following command:

hadoop jar hadoop-cos-distchecker-2.8.5-1.0-SNAPSHOT.jar com.qcloud.cos.hadoop.di stchecker.App hdfs://10.0.0.3:9000/check_list.txt hdfs://10.0.0.3:9000/benchmarks cosn://hdfs-test-1250000000/benchmarks cosn://hdfs-test-1250000000/check_result



Hadoop-cos-DistChecker will read the source file list and source directory, and run the MapReduce job to perform a distributed check. The final check result will be output to cosn://examplebucket-appid/check_result .



The check report is as follows:

hdfs://10.0.0.3:9000/benchmarks/TestDFSIO hdfs://10.0.0.3:9000/benchmarks/TestDFS IO,cosn://hdfs-test-1250000000/benchmarks/TestDFSIO,None,None,None,SUCCESS,'The s ource file and the target file are the same.'

hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_control hdfs://10.0.0.3:9000/benchma rks/TestDFSIO/io_control,cosn://hdfs-test-1250000000/benchmarks/TestDFSIO/io_cont rol,None,None,SUCCESS,'The source file and the target file are the same.' hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_control/in_file_test_io_0 hdfs://10. 0.0.3:9000/benchmarks/TestDFSIO/io_control/in_file_test_io_0,cosn://hdfs-test-125 0000000/benchmarks/TestDFSIO/io_control/in_file_test_io_0,CRC64,15663109861765878 38,1566310986176587838,SUCCESS,'The source file and the target file are the sam e.'

hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_control/in_file_test_io_1 hdfs://10. 0.0.3:9000/benchmarks/TestDFSIO/io_control/in_file_test_io_1,cosn://hdfs-test-125

0000000/benchmarks/TestDFSIO/io_control/in_file_test_io_1,CRC64,-6584441696534676 125,-6584441696534676125,SUCCESS,'The source file and the target file are the sam e.' hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_data hdfs://10.0.0.3:9000/benchmark s/TestDFSIO/io data, cosn://hdfs-test-1250000000/benchmarks/TestDFSIO/io data, Non e, None, None, SUCCESS, 'The source file and the target file are the same.' hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_data/test_io_0 hdfs://10.0.0.3:9000/ benchmarks/TestDFSIO/io_data/test_io_0, cosn://hdfs-test-1250000000/benchmarks/Tes tDFSIO/io_data/test_io_0,CRC64,3534425600523290380,3534425600523290380,SUCCESS,'T he source file and the target file are the same.' hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_data/test_io_1 hdfs://10.0.0.3:9000/ benchmarks/TestDFSIO/io_data/test_io_1,cosn://hdfs-test-1250000000/benchmarks/Tes tDFSIO/io_data/test_io_1,CRC64,3534425600523290380,3534425600523290380,SUCCESS,'T he source file and the target file are the same.' hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_write hdfs://10.0.0.3:9000/benchmark s/TestDFSIO/io_write,cosn://hdfs-test-1250000000/benchmarks/TestDFSIO/io_write,No ne, None, SUCCESS, 'The source file and the target file are the same.' hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_write/_SUCCESS hdfs://10.0.0.3:9000/ benchmarks/TestDFSIO/io_write/_SUCCESS,cosn://hdfs-test-1250000000/benchmarks/Tes tDFSIO/io_write/_SUCCESS,CRC64,0,0,SUCCESS,'The source file and the target file a re the same.' hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_write/part-00000 hdfs://10.0.0.3:900 0/benchmarks/TestDFSIO/io_write/part-00000,cosn://hdfs-test-1250000000/benchmark s/TestDFSIO/io_write/part-00000,CRC64,-4804567387993776854,-4804567387993776854,S

UCCESS, 'The source file and the target file are the same.'

Check Report Format

The check report is in the following format:

Source file path in `check_list.txt`, absolute path of the source file, absolute path of the destination file, checksum algorithm, checksum of the source file, ch ecksum of the destination file, check result, result description

There are 7 check results:

- SUCCESS: The source and destination files exist and are the same.
- MISMATCH: The source and destination files exist but are different.
- UNCONFIRM: The system cannot determine whether the source and destination files are the same. This may be because the destination file already existed in COS before the CRC64 feature was launched, and thus its CRC64 checksum cannot be obtained.
- UNCHECKED: The check is not performed. This is mainly because the source file cannot be read, or its checksum cannot be obtained.

- SOURCE_FILE_MISSING: The source file does not exist.
- TARGET_FILE_MISSING: The destination file does not exist.
- TARGET_FILESYSTEM_ERROR: The destination file system is not CosN.

FAQs

Why is there a negative CRC64 checksum in the check report?

A CRC64 checksum may contain 20 digits, which exceeds the range of the Java long type. However, they have the same underlying bytes. Therefore, when the long value is printed, it may be negative.

HDFS TO COS

Last updated : 2022-12-28 16:17:16

Overview

The HDFS TO COS tool is used to copy data from HDFS to Tencent Cloud COS (Cloud Object Storage) .

Operating Environments

Operating system

Linux or Windows

Software dependency

JDK v1.7 or v1.8

Installation and configuration

For more information on environment installation and configuration, see Java.

Configuration Method

- 1. Install Hadoop v2.7.2 or higher. For detailed directions, see Hadoop.
- 2. Download the HDFS TO COS tool from GitHub and decompress it.
- 3. Copy the core-site.xml file of the HDFS cluster to be synced to the conf folder. The coresite.xml file contains the configuration information of NameNode.
- 4. In the cos_info.conf configuration file, configure the bucket, region, and API key information. The bucket name is formed by connecting a user-defined string and the system-generated APPID with a hyphen, for example, examplebucket-1250000000 .
- 5. Specify the configuration file location in the command line parameter. The default location is conf/cos_info.conf .

Note :

If the command line parameter conflicts with the configuration file, the command line parameter shall apply.



Usage

Note :

Linux is used as an example below.

Viewing help

./hdfs_to_cos_cmd -h

Copying a file

• Copy from HDFS to COS. If a file with the same name as the file to be copied already exists in COS, the former will be overwritten.

./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/

• Copy from HDFS to COS. If a file with the same name and length as the file to be copied already exists in COS, the latter will be skipped (this is suitable for repeated copy).

```
./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/ -skip_if_len
_match
```

Only the length is checked here, as the overheads would be very high if the digests of files in Hadoop are calculated.

• Copy from HDFS to COS. If the Har directory (Hadoop archive file) exists in HDFS, the .har files can be automatically decompressed by specifying the --decompress_har parameter:

```
./hdfs_to_cos_cmd --decompress_har --hdfs_path=/tmp/hive --cos_path=/hdfs/20170
224/
```

If the --decompress_har parameter is not specified, the directory will be copied as an ordinary HDFS directory, that is, the files in the Har directory such as index and masterindex will be copied as-is.

Directory information

🔗 Tencent Cloud

```
log: log directory
src: Java source program
dep: compiled executable JAR package
```

FAQs and Help

Configuration information

Please make sure that the entered configuration information is correct, including bucket, region, and API key information. The bucket name is formed by connecting the user-defined string and system-generated APPID with a hyphen, such as examplebucket-125000000 . Please also make sure that the time on the server is in sync with the local time (if there is a difference of about 1 minute, it is okay, but if the difference is large, please set the server time correctly).

DataNode

Please make sure that the server where the copy program is located can also access the DataNode. The NameNode uses a public IP address and can be accessed, but the DataNode where the obtained block is located uses a private IP address and cannot be accessed directly; therefore, it is recommended that the copy program be placed in a Hadoop node for execution, so that both the NameNode and DataNode can be accessed.

Permissions

Please use the current account to download a file with the Hadoop command, check whether everything is correct, and then use the synchronization tool to sync the data in Hadoop.

File overwriting

Files that already exist in COS will be overwritten by default in case of repeated upload, unless you explicitly specify the -skip_if_len_match parameter, which indicates to skip files during upload if they have the same length as existing files.

cos path

cos path is considered as a directory by default, and files that are eventually copied from HDFS will be stored in this directory.

Copying data from Tencent Cloud EMR HDFS

To copy data from Tencent Cloud EMR HDFS to COS, you are advised to use the high-performance DistCp tool. For more information, see Migrating Data Between HDFS and COS.

Online Auxiliary Tools COS Request Tool

Last updated : 2022-10-25 14:47:40

Feature Overview

The COS request tool is a web-based debugging tool provided by COS. It is integrated on the TencentCloud API 3.0 Explorer platform for API debugging.

Note:

Requests sent by the COS request tool will be sent to the real COS server. As all operations are real, be careful when performing operations such as DELETE .

The COS request tool supports XML APIs but not JSON APIs.

- JSON APIs were provided by COS for you to access COS before XML APIs were released. Both types of APIs
 have the same underlying architecture where data is interconnected; however, they are incompatible with each
 other.
- XML APIs have a richer set of features and strengths over JSON APIs. We strongly recommend you upgrade to XML APIs for COS.

Tool URL

Click here to enter the COS request tool.

Directions

Select the **Cloud Object Storage** product, select the required API, enter parameters for the API, and click **Send Request** to get the corresponding response.

The COS request tool page shows the sections of product, API, parameter, and result from left to right. You can perform operations in different sections and send the request in the result section to get the response and process



parameters.

Search Tencent Cloud services and		Q,	GetService GetServiceAPI Document 🗹	Code Generating Online Call COS self-service diagnostic tool 🖾
Cloud Virtual Machine	Service API	×	Private Key	Nate: Disco anto their conting contexts on the ADI is contributed to cost according.
lighthouse	GET Service			Note: Prease note that senting requests waither APT is equivalent to real operations.
Auto Scaling	Bucket Basic Operations	~	The platform will provide the logged-in user with an Access Key for debugging; the tool will automatically generate a signature without	Online Call
Batch Compute	Bucket ACL	\sim	additional input from you.	Click the "Send Request" button below, the system will send your request to the server, operate your bucket or o bject according to your request. This operation is equivalent to the real operation. The system will show you the i
Edge Computing Machine	Bucket CORS	×	Input Parameters View Only Required Parameters	nformation like request result, header, etc. for your debugging and other reference.
Cloud Object Storage	Bucket Lifecycle	\sim	Perion (1)	Sand Panuart
Tencent Kubernetes Engine	Bucket Policy	\sim		
Tencent Container Registry	Bucket Referer	\sim	Region, Example, ap-guangzhou	
lencent container keystry	Bucket Tagging	\sim	Headers (j)	
Cloud File Storage	Bucket Website	\sim	key value +	
Cloud HDFS	Bucket Inventory	~		
Cloud Block Storage	Bucket Cross-region	\sim		
Cloud Load Balancer	Replication			
Virtual Private Cloud	Bucket Versioning	~		
Direct Connect	Object Basic Operations	×		
Content Delivery Network	Object ACL	\sim		
Global Application Acceleration F	Object Multipart Upload	×		

The detailed steps to use the COS request tool are as shown below:

1. Select the COS product.

Click **Cloud Object Storage** in the product section on the far left, and then you can see COS APIs in the API section.

Note :

The COS request tool is integrated on the TencentCloud API 3.0 platform that provides API debugging tools for many Tencent Cloud products. You can also select other products to debug their APIs as needed.

2. Select the API to be debugged.

You can select the API for debugging as needed. Three types of COS APIs are shown in the API section: service APIs, bucket APIs, and object APIs.

- Take GET Service as an example for service APIs. This API can list the information of all buckets under your account. Your API key is required. To get the bucket information of your account in a specified region, select the corresponding region in the parameter section. For more information on this API, see GET Service (List Buckets).
- Bucket APIs are used to manipulate buckets, such as PUT Bucket lifecycle . For more information, see Bucket APIs.
- Object APIs are used to manipulate objects, such as PUT Object . For more information, see Object APIs.

3. Enter parameters for the API.

The parameter section lists the corresponding parameters for the selected API. For more information on the parameters of COS APIs, see API Documentation.

API key is a required parameter for API calling. When using an API to manipulate resources such as buckets or objects, you need to enter your API key to authorize the API request, which can be obtained on the Manage API Key page in the CAM console.

Note :

For each API, the COS request tool displays a red asterisk behind each required parameter. You can also select **Only Required Parameters** to view required parameters only in the parameter section.

4. Send a request and view the response.

After selecting an API and entering parameters, click **Send Request** on the **Online Call** tab. Your request will be sent to the server, and the server will manipulate your buckets or objects accordingly.

Note :

Requests sent by the COS request tool will be sent to the real COS server. As all operations are real, be careful when performing operations such as **DELETE**.

After the request is sent, the returned result and the request parameters will be displayed in the result section. The **Request Parameters** part lists your HTTP request body; the **Response Result** part lists the response body of the request; the **Signature Process** part lists the signature involved in the request and its generation process; and the **Curl** part lists the statement called by Curl.

Sample

For example, a GET Objectrequest is sent to get a file named0001.txtas shown below. The RequestParameters part lists the corresponding parameters of the request.

```
GET https://bucketname-appid.cos.ap-region.myqcloud.com/0001.txt
Host: bucketname-appid.cos.ap-region.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDwqaGoCIWIG4hDWdJUTL5e3hn04xi***&q-
sign-time=1543398166;1543405366&q-key-time=1543398166;1543405366&q-header-list=ho
st&q-url-param-list=&q-signature=f50ddd3e0b54a92df9d4efe2d0c3734a8c90****
```

The first line shows your HTTP Verb and the link to be accessed; the second line shows the domain name to be accessed; and the last line shows the signature information of the request. For requests of the PUT type, request headers are complicated, but there are some common request headers. For more information, see Common Request Headers.

The **Signature Process** part shows the signature involved in this request and its generation process. For more information on signature algorithms, see Request Signature. If you need to generate and debug request signatures, we recommend you use the COS signature tool.

The response result returned by COS is as follows:

```
200 OK
content-type: text/plain
content-length: 6
connection: close
accept-ranges: bytes
date: Wed, 28 Nov 2018 09:42:49 GMT
etag: "5a8dd3ad0756a93ded72b823b19dd877"
last-modified: Tue, 27 Nov 2018 20:05:26 GMT
server: tencent-cos
x-cos-request-id: NWJmZTYzMTlfOWUxYzBiMDlfOTA4NF8yMWI2****
x-cos-version-id: MTg0NDY3NDI1MzAzODkyMjU****
hello!
```

The 200 OK in the first line is the status code returned for the request. If the request fails, the corresponding error code will be returned. For more information, see Error Codes. Other lines are response headers, which vary by API, but there are some common response headers. For more information, see Common Response Headers.

Notes

After you click **Send Request** to send the request with its required parameters entered to the COS server, COS will perform the corresponding operation on your buckets or objects. The operation cannot be undone or reverted; therefore, proceed with caution.

Diagnostic Tool

Last updated : 2022-05-05 07:50:50

Overview

COS's web-based diagnostic tool allows you to troubleshoot error requests. You can enter the error RequestId (see Obtaining RequestId) on the tool page and click **Diagnose**. The tool will check the request and provide basic information about the request as well as suggestions so that you can quickly troubleshoot COS API errors.

Tool URL

Diagnosis Tool

Directions

- 1. Click Diagnosis Tool.
- 2. Enter the RequestId and click Diagnose.
- 3. Wait and view the diagnostic result.

The result includes the suggestions and the request information.

- The suggestions help you quickly locate the COS API errors.
- The request information is the information about the request corresponding to RequestId .
- 4. Send your feedback on the suggestions.

Click Helpful or Not Helpful below the suggestions so that we can further improve the tool.

FAQs

On the diagnostic tool page, you can also find the FAQs. If you have any queries, please contact us.



Notes

ACOS RequestId must:

- 1. Start with N.
- 2. Contain at least 30 characters.
- 3. Comply with the Base64 standard.