Cloud Object Storage

User Tools

Product Introduction

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

Tools Overview

Last updated：2018-07-19 10:24:21

COS provides developers with the following tools:

Tool	Features
COSCMD	This tool allows users to perform operations such as batch upload/download/deletion of objects by using simple command line instructions.
COS Browser	This tool makes it easy for users to perform data upload/download and other operations in a visualized manner.
Migration tool	This tool is used to migrate files from AWS S3, Alibaba Cloud OSS, and Qiniu to COS. It also allows migration of a list of files from given URLs to COS.
Local synchronization tool	This tool is used to synchronize files and sub-directories under a local directory to COS.
FTP Server	This tool allows users to upload/download files to/from COS via FTP server.
COSFS	In Linux, this tool is used to mount buckets to a local file system and operate objects in COS via the local file system.
Hadoop tool	This tool allows users to process objects in COS using Hadoop, such as MapReduce and HiveCOS.
HDFS TO COS	This tool is used to copy the data on HDFS to COS.
WeCOS	This tool automatically uploads image resources in WeChat Mini Programs to COS, replaces image resource reference address with COS online address, and then removes image resources from the project directory. As a result, WeChat Mini Programs are slimmed down.

COSCMD

Last updated：2018-09-10 17:22:41

Feature Description

With COSCMD tool, users can perform operations such as batch upload/download/deletion of Objects by using simple command line instructions.

Operating Environment

System environment

Windows or Linux system
(Local characters should be in utf-8 format, otherwise exceptions will occur when operating on Chinese files.)

Software dependencies

- Python 2.6/2.7/3.5/3.6
- Latest version of pip

Installation and configuration

For more information on the installation and configuration of environment, please see Install and Configure Python.

Download and Installation

1. Check whether it is Windows or Linux system.

2. Check whether Python is installed. For more information on how to install Python, please see Install and Configure Python

3. Check whether the latest version of pip is installed. For more information on how to install pip, please see PyPA pip Document.

4. Download COSCMD installer package.

5. Open the Terminal and install pip by executing the following command:

   ```
   pip install coscmd
   ```

   After installation is completed, you can view the current version information using `pip -v` or `pip --version` command.

6. Update COSCMD by executing the following command:

   ```
   pip install coscmd -U
   ```

> Note: pip can be installed or updated by using the above methods in either Linux or Windows environment.

How to Use

View help

Users can view the tool's help information with `-h` or `--help` command.

```
coscmd -h //View current version information
```

The help information is shown as follows:

```
usage: cos_cmd.py [-h] [-d] [-b BUCKET] [-r REGION] [-c CONFIG_PATH]
[-l LOG_PATH] [-v]
{config,upload,download,delete,copy,list,info,mget,restore,signurl,createbucket,deletebucket,putobjectacl,getobjectacl,putbucketacl,getbucketacl}
...

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,copy,list,info,mget,restore,signurl,createbucket,deletebucket,putobjectacl,getobjectacl,putbucketacl,getbucketacl}
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
copy copy file from COS to COS.
list list files on COS
info get the information of file on COS
mget download file from COS to local.
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

optional arguments:
-h, --help show this help message and exit
-d, --debug debug mode
-b BUCKET, --bucket BUCKET
set bucket
-r REGION, --region REGION
set region
-c CONFIG_PATH, --config_path CONFIG_PATH
set config_path
-l LOG_PATH, --log_path LOG_PATH
set log_path
-v, --version show program's version number and exit
```

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

```
coscmd upload -h //View the usage of upload command
```

Configure parameters

You need to configure parameters before using the COSCMD tool. Run the following command for configuration:

```
coscmd config -a <secret_id> -s <secret_key> -b <bucket> -r <region> [-m <max_thread>] [-p <part
s_size>]
```

In the above example, fields in "<>" are required, and those in "[]" are optional. Parameters are described below:

Name	Description	Valid Value
secret_id	ID of the key corresponding to the APPID (required). It can be obtained on the **Key Management** in the left navigation pane of the COS console, or on the Cloud API Key Console.	String
secret_key	The Key corresponding to the APPID (required). It can be obtained on the **Key Management** in the left navigation pane of the COS console, or on the Cloud API Key Console.	String
bucket	The specified bucket name (required), which is in a format of {name}-{appid}. For more information, please see Create Bucket.	String
region	The region where the bucket resides (required). For more information, please see Available Regions	String
max_thread	The maximum number of threads for multi-threaded upload (optional). Default is 5. Valid value: 1-10.	Numeral
parts_size	Part size in multipart upload (in MB) (optional). Default is 1 MB. Valid value: 1-10.	Numeral

Note:

1. You can directly edit `~/.cos.conf` file (a hidden file located under `My Documents` in Windows environment).
   The following shows an example of the content of the configured `.cos.conf` file:

```
[common]
secret_id = AChT4ThiXAbpBDEFGhT4ThiXAbpHIJK
secret_key = WE54wreefvds3462refgwewerewr
bucket = ABC-1234567890
region = ap-guangzhou
max_thread = 5
```

```
part_size = 1
schema = https
```

2. Add `schema` in the configuration file to select `http/https`. Default is `https`.
3. Bucket must be in a format of `{name}-{appid}`.

Command for specifying a bucket

- You can specify a bucket with `-b <bucket>` command and upload files to it using relevant commands, such as the command for uploading files.
- The bucket entered must be in a format of `{name}-{appid}`.

```
coscmd -b <bucket> method ... //Command format
coscmd -b AAA-12345567 upload a.txt b.txt //Example - Upload files
coscmd -b AAA-12344567 createbucket //Example - Create a bucket
```

Create a bucket

- It should be used together with `-b <bucket>` command.

```
coscmd -b <bucket> createbucket //Command format
coscmd -b AAA-12344567 createbucket //Example
```

Delete a bucket

- It should be used together with `-b <bucket>` command.

```
coscmd -b <bucket> deletebucket //Command format
coscmd -b AAA-12344567 deletebucket //Example
```

Upload files or folder

- Command for file upload is as follows:

```
coscmd upload <localpath> <cospath> //Command format
coscmd upload /home/aaa/123.txt bbb/123.txt //Example
coscmd upload /home/aaa/123.txt bbb/ //Example
```

- Command for folder upload is as follows:

```
coscmd upload -r <localpath> <cospath> //Command format
coscmd upload -r /home/aaa/ bbb/aaa //Example
coscmd upload -r /home/aaa/ bbb/ //Example
coscmd upload -r /home/aaa/ / //Upload to the bucket root directory
coscmd upload -rs /home/aaa/ /home/aaa //Upload files synchronously and skip those with the sa
me md5
coscmd upload -rs /home/aaa/ /home/aaa --ignore *.txt,*.doc //Ignore .txt and .doc files
```

Replace the parameters in "<>" with the path of the local file to be uploaded (localpath) and the storage path on COS (cospath).

Note:

- When uploading a file, you need to provide a complete path including the file (folder) name on COS (refer to the example).
- COSCMD supports resuming upload from breakpoint for large files. When multipart upload of large files failed, only the part that fails to be uploaded is uploaded again, 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 in multipart upload.
- `x-cos-meta-md5` header is carried by default when COSCMD uploads a file, and its value is the `md5` of the file.
- Use -s parameter to upload files synchronously and skip those with the same md5 (only if the source files on COS are uploaded using COSCMD 1.8.3.2 or above, and x-cos-meta-md5 header is carried by default).
- HTTP header that is set with -H parameter must be in json format. For example: `coscmd upload -H '{"Cache-Control":"max-age=31536000","Content-Language":"zh-CN"}' <localpath> <cospath>` .
- You can ignore a certain type of files using `--ignore` parameter when uploading files. Multiple Shell wildcard rules (separated by commas) are supported.
- File size is limited to 40 TB for a single file upload.

Download files or folder

- Command for file download is as follows:

```
coscmd download <cospath> <localpath> //Command format
coscmd download bbb/123.txt /home/aaa/111.txt //Example
coscmd download bbb/123.txt /home/aaa/ //Example
```

- Command for folder download is as follows:

```
coscmd download -r <cospath> <localpath> //Command format
coscmd download -r /home/aaa/ bbb/aaa //Example
coscmd download -r /home/aaa/ bbb/ //Example
coscmd download -rf / bbb/aaa //Download all the files under the current bucket root directory and overwrite local files
coscmd download -rs / bbb/aaa //Download all the files under the current bucket root directory synchronously and skip those with the same md5
coscmd download -rs / bbb/aaa --ignore *.txt,*.doc //Ignore .txt and .doc files
```

Replace the parameters in "<>" with the path of the file to be downloaded on COS (cospath) and the local storage path (localpath).

> **Note:**
>
> - If a file with the same name exists locally, the download will fail. Use the `-f` parameter to overwrite the local file.
> - The API `download` employs multipart download, which should be used since the old version API `mget` has been deprecated.
> - Use the parameter `-s` or `--sync` to skip the files that already exist locally when downloading a folder
>   (only if the folder to be downloaded is uploaded using the API `upload` of `COSCMD`, and `x-cos-meta-md5` header is carried in the files).
> - You can ignore a certain type of files using --ignore parameter when downloading files. Multiple Shell wildcard rules (separated by commas) are supported.

Delete files or folder

- Command for file deletion is as follows:

```
coscmd delete <cospath> //Command format
coscmd delete bbb/123.txt //Example
```

- Command for folder deletion is as follows:

```
coscmd delete -r <cospath> //Command format
coscmd delete -r bbb/ //Example
coscmd delete -r / //Example
```

Replace the parameter in "<>" with the path of the file to be deleted on COS (cospath). You will be prompted to confirm this operation.

> **Note:**

- You can use `-f` parameter to skip the confirmation that is required for batch deletion.

Copy files or folder

- Command for file copying is as follows:

```
coscmd copy <sourcepath> <cospath> //Command format
coscmd copy bucket-appid.cos.ap-guangzhou.myqcloud.com/a.txt aaa/123.txt //Example
```

- Command for folder copying is as follows:

```
coscmd copy -r <sourcepath> <cospath> //Command format
coscmd copy -r bucket-appid.cos.ap-guangzhou.myqcloud.com/coscmd/ aaa //Example
coscmd copy -r bucket-appid.cos.ap-guangzhou.myqcloud.com/coscmd/ aaa/ //Example
```

Replace the parameters in "<>" with the path of the file to be copied on COS (sourcepath) and the path to which the file is copied on COS (cospath).

> **Note:**
> The format of sourcepath: `<bucketname>-<appid>.cos.<region>.myqcloud.com/<cospath>`

Print file list

- Print command is as follows:

```
coscmd list <cospath> //Command format
coscmd list -a //Example
coscmd list bbb/123.txt -r -n 10 //Example
```

  Replace the parameter in "<>" with the path of the file list to be printed on COS (cospath).
- Use `-a` to print all files.
- Use `-r` to print files recursively. The number and total size of files are listed at the end of the returned result.
- Use `-n num` to set the maximum number of files to be printed.

> **Note:**
> If `<cospath>` is left empty, the files under the current Bucket root directory are printed by default.

Display file information

- Command is as follows:

  ```
  coscmd info <cospath> //Command format
  coscmd info bbb/123.txt //Example
  ```

  Replace the parameter in "<>" with the path of the file to be displayed on COS (cospath).

Get signed download URL

- Command is as follows:

  ```
  coscmd signurl <cospath> //Command format
  coscmd signurl bbb/123.txt //Example
  coscmd signurl bbb/123.txt -t 100//Example
  ```

  Replace the parameter in "<>" with the path of the file on COS for which you need to get the download URL (cospath).
  Use `-t time` to set the validity period of the signature to be printed (in sec).

Set Access Control List (ACL)

- Command is as follows:
  Set Bucket ACL using the following command:

  ```
  coscmd putbucketacl [--grant-read GRANT_READ] [--grant-write GRANT_WRITE] [--grant-full-control GRANT_FULL_CONTROL] //Command format
  coscmd putbucketacl --grant-read 12345678,12345678/11111 --grant-write anyone --grant-full-control 12345678/22222 //Example
  ```

  Set Object ACL using the following command:

  ```
  coscmd putobjectacl [--grant-read GRANT_READ] [--grant-write GRANT_WRITE] [--grant-full-control GRANT_FULL_CONTROL] <cospath> //Command format
  coscmd putobjectacl --grant-read 12345678,12345678/11111 --grant-write anyone --grant-full-control 12345678/22222 aaa/aaa.txt //Example
  ```

How to set ACL

- --grant-read represents read permission.
- --grant-write represents write permission.
- --grant-full-control represents read-write permission.
- GRANT_READ / GRANT_WRITE / GRANT_FILL_CONTORL represents the account granted with permission.
- For authorization to a root account, the format of rootid is used.
- For authorization to a sub-account, the format of rootid/subid is used.

- For authorization to any account, the format of anyone is used.
- Accounts that are granted with permissions at the same time are separated with `,`.
- Replace the parameter with the path of the file to be deleted on COS (cospath).
- In case of Object ACL setting, if you want to set an ACL for all files under a folder, add `/` to the end of the folder name.

Get Access Control List (ACL)

- Get the Bucket ACL using the following command:

```
coscmd getbucketacl //Command format
coscmd getbucketacl //Example
```

- Get the Object ACL using the following command:

```
coscmd getobjectacl <cospath> //Command format
coscmd getobjectacl aaa/aaa.txt //Example
```

Restore archived files

- Command is as follows:

```
coscmd restore <cospath> //Command format
coscmd restore a.txt -d 3 -t Expedited//Example
coscmd restore a.txt -d 3 -t Bulk///Example
```

Replace the parameter in "<>" with the path of the file list to be printed on COS (cospath).

- Use `-d day` to set the validity period of the temporary replica. Default value: 7.
- Use `-t tier` to specify restoring mode. Enumerated values: Expedited, Standard, and Bulk. Default value: Standard.

Command for Debug mode

If `-d` or `-debug` is added before each command, the details of operation during command execution are displayed. Here's an example:

```
//Display details of upload operation
coscmd -d upload <localpath> <cospath> //Command format
coscmd -d upload /home/aaa/123.txt bbb/123.txt //Example
```

COSBrowser Tool

Last updated：2018-07-12 18:08:11

COSBrowser (COS for PC)

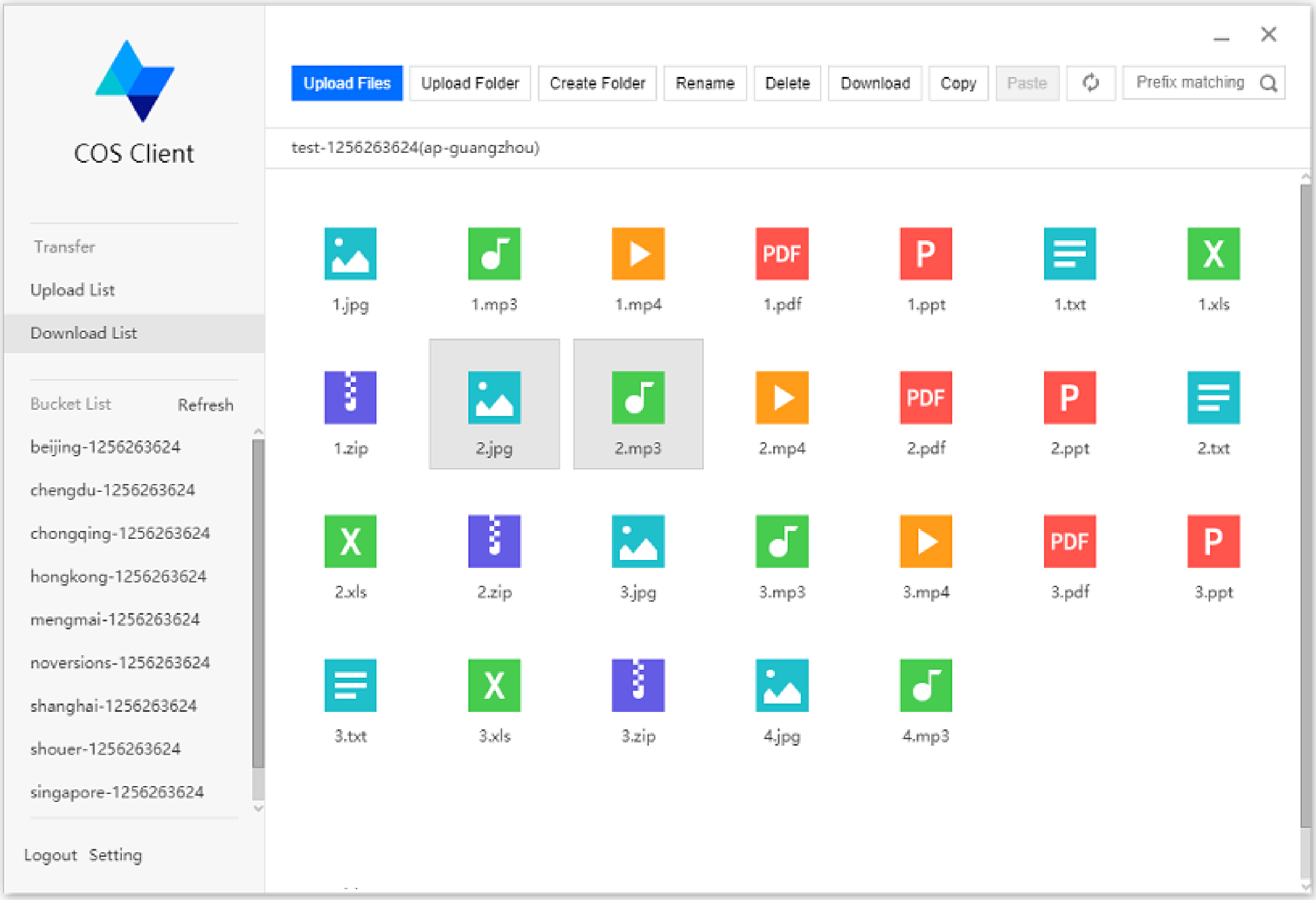
COSBrowser makes it easy for users to upload and download data in a visualized manner.

Download and Installation

Download setup file (double click to install):

- COSBrowser Windows OS Client
- COSBrowser macOS Client

Software Interface



How to Use

Users can log in to COS Browser using Tencent Cloud API key (SecretId and SecretKey), which can be obtained on the Console. Login information will be cached.
Batch upload/download/deletion is supported. Drag and drop upload/download is not allowed.

Change Log

You can get change log here change log

Suggestions and Feedback

If you have any suggestions and feedback, please visit cosbrowser issues

COS Migration

Last updated：2018-09-17 14:29:28

Feature Description

COS Migration is an all-in-one tool integrating COS data migration features. You can migrate data from source address to COS after simple configurations. COS Migration has the following features:

- Various data sources
  - Local data: Supports migrating local data to COS.
  - Other cloud storage services: Supports migrating data from AWS S3, Alibaba Cloud OSS and Qiniu Cloud to COS. More cloud storage services will be supported in the future.
  - URL list: Supports downloading and migrating data from a given URL list to COS.
  - Bucket Replication: Supports replicating data from one bucket to another in COS as well as cross-region and cross-account data replication.
- Resuming upload from breakpoint
- Multipart upload
- Parallel upload
- Migration Verification

Operating Environment

System Environment

Linux/Windows

Software Dependencies

- JDK1.7 or later. For more information on the installation and configuration of JDK, please see Install and Configure Java.

How to Use

1. Obtain Migration Tool

Download link: COS Migration Tool

2. Decompress Toolkit

Windows

Decompress and save it to a directory, such as

```
C:¥Users¥Administrator¥Downloads¥cos_migrate
```

Linux

Decompress and save it to a directory

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

Directory Structure of Migration Tool

After decompression, the directory structure of COS Migration tool is as follows:

```
COS_Migrate_tool
|--conf  #Directory where the configuration file locates
|   |--config.ini  #Configuration file for migration
|--db    #Record of successful migration
|--dep   #Jar package compiled and generated by main program logic
|--log   #Logs generated when using the tool
|--opbin #Scripts used for compiling
|--src   #Source code of the tool
|--tmp   #Temporary file storage directory
|--pom.xml #Project configuration file
|--README  #Instruction documentation
|--start_migrate.sh  #Launch script for migration in Linux
|--start_migrate.bat #Launch script for migration in Windows
```

Notes:

- db directory is used to record the identifiers of the files that have been migrated successfully. Whenever a file is to be migrated, check whether its identifier is recorded in db directory. If yes, skip the file. Otherwise, you can migrate the file.

- log directory is used to record all logs generated during migration. If any error occurs in migration, check error.log in this directory.

3. Modify Configuration File config.ini

Before executing the launch script for migration, you need to modify config.ini (path: `./conf/config.ini`), including the followings:

3.1 Configure Migration Type

type, indicating migration type, is filled in by users according to their needs. For example, to migrate local data to COS, users need to configure `type=migrateLocal` for `[migrateType]` .

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

Supported migration types are as follows:

migrateType	Description
migrateLocal	Migrate local data to COS
migrateAws	Migrate data from AWS S3 to COS
migrateAli	Migrate data from Alibaba Cloud OSS to COS
migrateQiniu	Migrate data from Qiniu to COS
migrateUrl	Download and migrate data from URLs to COS
migrateBucketCopy	Copy data from source bucket to destination bucket

3.2 Configure Migration Task

Users can configure the migration according to their needs, including configuration of migration to destination COS and migration tasks.

```
# Users can configure destination COS account in this common configuration section.
[common]
secretId=AKIDXXXXXXXXXXXXXXXX
secretKey=GYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
bucketName=mybcket-1251668577
region=ap-guangzhou
storageClass=Standard
cosPath=/
https=off
```

```
tmpFolder=./tmp
smallFileThreshold=5242880
smallFileExecutorNum=64
bigFileExecutorNum=8
entireFileMd5Attached=on
daemonMode=off
daemonModeInterVal=60
executeTimeWindow=0,24
```

Name	Description	Default Value
secretId	SecretId for user key. For more information, please see Cloud API Key Console.	-
secretKey	SecretKey for user key. For more information, please see Cloud API Key Console.	-
bucketName	Destination bucket name. Bucket is named in a format of {name}-{appid}, which means a bucket name must contain APPID, such as movie-1251000000	-
region	Region information of the destination bucket. For information on COS region abbreviations, please see Available Regions	-
storageClass	Storage class: Standard - COS Standard, Standard_IA - COS Infrequent Access	Standard
cosPath	COS path to which files are migrated. **/**: Migrate to the root path of the bucket. **/aaa/bbb/**: Migrate to /aaa/bbb/ of the bucket. Path /aaa/bbb/ will be created automatically if it does not exist.	/
https	Whether to transfer via HTTPS. on :Yes, off: No. Transfer via HTTPS is rather slow and is suitable for scenarios that demand high security.	off

Name	Description	Default Value
tmpFolder	The directory to store temporary files generated in the migration from other Cloud Storage to COS, which will be deleted after the migration. The directory should be in the format of an absolute path. The separator used in Linux is a forward slash, such as /a/b/c. Double backslashes are used in Windows, such as E:\a\b\c. Default is the tmp directory under tool path	./tmp
smallFileThreshold	Byte threshold for small files. Default is 5 MB. Files greater than or equal to this threshold are uploaded via multipart upload. Otherwise, simple upload is used.	5242880
smallFileExecutorNum	Concurrency for uploading small files (smaller than smallFileThreshold) via simple upload. Decrease the concurrency if the files are uploaded to COS via public network with small bandwidth.	64
bigFileExecutorNum	Concurrency for uploading large files (greater than or equal to smallFileThreshold) via multipart upload. Decrease the concurrency if the files are uploaded to COS via public network with small bandwidth.	8
entireFileMd5Attached	MD5 for the entire file, which is calculated by the migration tool and stored in the custom header x-cos-meta-md5 for follow-up verification. Because the etag of the large file uploaded to COS via multipart upload is different from the MD5 for the entire file.	on

Name	Description	Default Value
daemonMode	Whether to enable damon mode. on: Yes, off: No. Synchronization is executed repeatedly in damon mode. The interval between two synchronizations is set by damonModeInterVal parameter.	off
daemonModeInterVal	The interval between two synchronizations (in sec)	60
executeTimeWindow	Execution time window, which describes the time period when the migration tool is executed. The time granularity is hour. For example, parameter 3,21 indicates that the migration is executed between 3:00 to 21:00. During other time, migration goes into sleep mode, and it is suspended with migration progress retained.	0,24

3.3 Configure Data Source Information

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 `[migrateLocal]` .

3.3.1 Configure local data source - migrateLocal

To migrate local data to COS, users should configure this section. Configuration items are as follows:

```
# Configuration section for migrating local data to COS
[migrateLocal]
localPath=E:¥¥code¥¥java¥¥workspace¥¥cos_migrate_tool¥¥test_data
exeludes=
```

Configuration Item	Description
localPath	Local path, which should be in the format of an absolute path. The separator used in Linux is a forward slash, such as /a/b/c. Double backslashes are used in Windows, such as E:\a\b\c.
exeludes	Absolute path of the directory or file to be excluded, which means some directories or files under localPath will not be migrated. Multiple absolute paths are separated by semicolons. If the item is left empty, it means to migrate all files under localPath.

3.3.2 Configure Alibaba Cloud OSS data source - migrateAli

To migrate data from Alibaba Cloud OSS to COS, users should configure this section. Configuration items are as follows:

```
# Configuration section for migrating data from Alibaba Cloud OSS to COS
[migrateAli]
bucket=mybucket-test
accessKeyId=xxxxxxxxxx
accessKeySecret=yyyyyyyyyyy
endPoint= OSS -cn-shenzhen.aliyuncs.com
prefix=
proxyHost=
proxyPort=
```

Configuration Item	Description
bucket	Name of Alibaba Cloud OSS bucket
accessKeyId	accessKeyId for user key
accessKeySecret	accessKeySecret for user key
endPoint	Endpoint in Alibaba Cloud
prefix	Prefix of the path from which files are migrated. Prefix should be left empty if all data under bucket is to be migrated.
proxyHost	Proxy IP address that should be entered to access via proxy.
proxyPort	Proxy port

3.3.3 Configure AWS data source - migrateAws

To migrate data from AWS to COS, users should configure this section. Configuration items are as follows:

```
# Configuration section for migrating data from AWS to COS
[migrateAws]
bucket=aws-emr-test
accessKeyId=xxxxxxxxxx
accessKeySecret=yyyyyyyyyyyyyyyy
endPoint=s3.us-east-1.amazonaws.com
prefix=
proxyHost=
proxyPort=
```

Configuration Item	Description
bucket	Name of AWS COS bucket
accessKeyId	accessKeyId for user key
accessKeySecret	accessKeySecret for user key
endPoint	Endpoint in AWS
prefix	Prefix of the path from which files are migrated. Prefix should be left empty if all data under bucket is to be migrated.
proxyHost	Proxy IP address that should be entered to access via proxy.
proxyPort	Proxy port

3.3.4 Configure Qiniu data source - migrateQiniu

To migrate data from Qiniu to COS, users should configure this section. Configuration items are as follows:

```
# Configuration section for migrating data from Qiniu to COS
[migrateQiniu]
bucket=mybuckettest
accessKeyId=xxxxxxxxxx
accessKeySecret=yyyyyyyyyyyyyyy
endPoint=wwww.bkt.clouddn.com
prefix=
proxyHost=
proxyPort=
```

Configuration Item	Description
bucket	Name of Qiniu COS bucket
accessKeyId	accessKeyId for user key
accessKeySecret	accessKeySecret for user key
endPoint	Download address for Qiniu data, corresponding to downloadDomain
prefix	Prefix of the path from which files are migrated. Prefix should be left empty if all data under bucket is to be migrated.
proxyHost	Proxy IP address that should be entered to access via proxy.
proxyPort	Proxy port

3.3.5 Configure URL list data source - migrateUrl

To migrate data from URL list to COS, users should configure this section. Configuration items are as follows:

```
# Configuration section for downloading and migrating data from URL list to COS
[migrateUrl]
```

Configuration Item	Description
urllistPath	URL list path, which should be in the format of an absolute path. The separator used in Linux is a forward slash, such as /a/b/c. Double backslashes are used in Windows, such as E:\a\b\c. If a directory is entered, all files under the directory will be scanned and migrated as urllist files.

3.3.6 Configure bucket replication - migrateBucketCopy

To migrate data from URL list to COS, users should configure this section. Configuration items are as follows:

```
# Configuration section for migrating data from source bucket to destination bucket
[migrateBucketCopy]
srcRegion=ap-shanghai
srcBucketName=mysrcbucket-1251668555
srcSecretId=xxxxxxxxxxx
srcSecretKey=yyyyyyyyyyyyyyy
srcCosPath=/
```

Configuration Item	Description
srcRegion	Region information of the source bucket. For more information, please see Available Regions
srcBucketName	Source bucket name. Bucket is named in a format of {name}-{appid}, which means a bucket name must contain APPID, such as movie-1251000000.

Configuration Item	Description	
srcSecretId	SecretId for the key of the source bucket user. For more information, please see Cloud API Key. For data migration under the same user account, SecretId described in srcSecretId should be the same as that in common. Otherwise, the migration is cross-account bucket replication.	
srcSecretKey		secret_key for the key of the source bucket user. For more information, please see Cloud API Key. For data migration under the same user account, secretId described in srcSecretId should be the same as that in common. Otherwise, the migration is cross-account bucket replication.
srcCosPath	COS path under which files are to be migrated to the destination bucket	

4. Run Migration Tool

Windows

Double-click **start_migrate.bat** to run the migration tool.

Linux

1. Read configurations from config.ini by execute the following command:

```
sh start_migrate.sh
```

2. Configurations of some parameters need to be read from the command line. Execute the command as follows:

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

Notes

- Configuration items can be read by two methods: reading from the command line or reading from the configuration file.
- The command line has higher priority than the configuration file, which means that the use of parameters in the command line is preferred if the command line and configuration file have the same configuration items.
- Reading configuration items from the command line makes it easy for users to run two different migration tasks at the same time, provided that the key configuration items (such as bucket name, COS path, and source path) in the two tasks are not exactly the same. Concurrent migration can be achieved because different migration tasks are written to different db directories. For more information on db, please see the directory structure of COS Migration Tool above.
- Configuration items are in a 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 should be in a format of **-Dcommon.cosPath=/bbb/ddd**.

Migration Mechanism and Process

Migration Mechanism

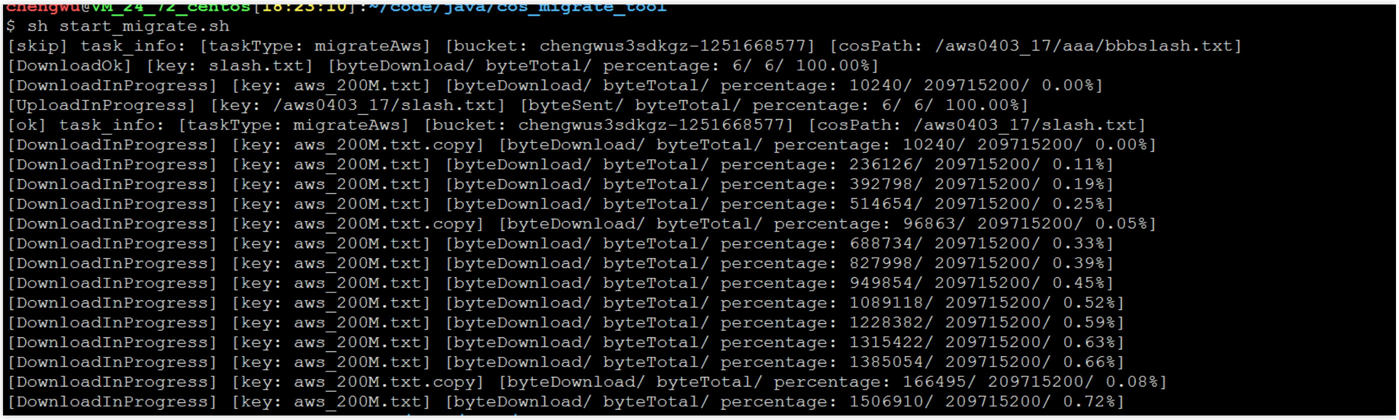
COS migration tool is stateful. Successful migrations will be recorded in the format of KV in leveldb file under db directory. Before each migration, check whether the path to which data is migrated has been recorded in db directory. If yes and its attribute is the same as that in db, this migration will be skipped. Otherwise, the migration will be executed. The attribute for determining whether to migrate varies depending on the type of migration. For local migration, mtime determines whether to migrate. For migration from other cloud storage services and bucket replication, etag and length of the source file determines whether to migrate. That's the reason why we search records of successful migration in db instead of COS. If a file is deleted or modified via COSCMD or console rather than the migration tool, the file will not be re-migrated because it is not changed in the perspective of the migration tool.

Migration Process

1. Read configuration files. Read each configuration section according to the type of migration, and check the parameters.

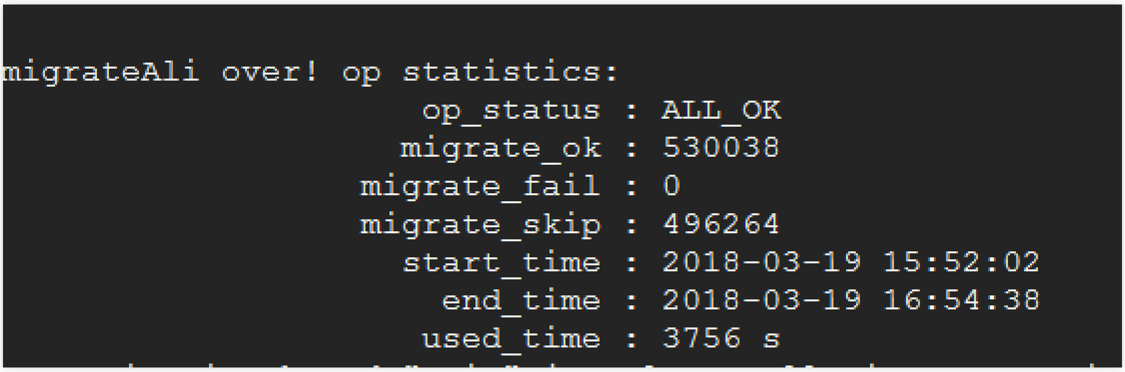
2. Scan and compare with the identifier of the file to be migrated under db based on the migration type, and then determine whether the upload is allowed.

3. Execution result is printed in the process of migration. inprogress: in migration, skip: migration skipped, fail: migration failed, ok: migration succeeded. You can check detailed failure information in error logs in log directory. The execution process is as shown below:



```
chengwu@VM_24_72_centos[18:23:10]:~/code/java/cos_migrate_tool
$ 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. Migration data such as cumulative success count, failure count, skip count and the time used for migration are printed after the migration. The migration tool will skip the migrated files and leave those failed to be migrated. You can check the error logs of the files that failed to be migrated or execute the migration again. Migration results are as follows:



```
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

1. What to do if COS Migration exits abnormally during migration?

COS Migration supports resuming upload from breakpoint. If the upload for large files is suspended due to abnormal exit or service failure, you can execute the tool again and resume the upload from the breakpoint.

2. If I delete the files migrated to COS via console or other methods, will these files be uploaded again by COS Migration?

No. All the migrated files are recorded in db. COS Migration scans db directory before each migration and files recorded in db will not be uploaded again. For more information, please see Migration Mechanism and Process.

3. What to do if the migration failed and 403 Access Deny displays in the log?

Check whether the key, bucket, and region are correct and ensure that you have operation permissions. Sub-account needs to be authorized by its parent account. Write-read permission to bucket is required to migrate data locally or from other cloud storage services. Write permission to source bucket is required for bucket replication.

4. What to do if the migration from other cloud storage services to COS failed and Read timed out displays?

This error occurs when downloading from other cloud storage services timed out due to insufficient bandwidth. For example, when you migrate overseas data from AWS to COS, if the bandwidth is insufficient and causes high latency, the prompt of "Read timed out" may appear. To solve this problem, please increase network bandwidth and test download speed with wget before migration.

5. What to do if the migration failed and 503 Slow Down displays in the log?

This error occurs when frequency control is triggered. The upper limit for a single account is 800 QPS in COS. You are recommended to decrease the concurrency for small files in configuration. Run the tool again and the migration will be executed.

6. What to do if the migration failed and 404 NoSuchBucket displays in the log?

Check whether the key, bucket and region are correct.

7. Others

Run the migration tool again. If the failure persists, please package the configuration information (with key information hidden) and the log directory, and submit a ticket.

FTP Server Tool

Last updated：2018-06-12 10:01:06

The COS FTP Server tool can be used to directly operate objects and directories in COS via the FTP protocol, including uploading/downloading/deleting files and creating folders. This tool is developed with Python, which makes the installation easier.

Operating Environment

System Environment

OS: Linux. It is recommended to use the CVM of Tencent Cloud CentOS 7 series. Windows systems are not supported for now.

Python interpreter version: Python 2.7. For more information on how to install and configure it, please see Python Installation and Configuration.

Dependent libraries:

- requests
- argparse

Download and Installation

GitHub link: COS FTP Server Tool.

After the download is completed, simply run `setup.py` under the `cos ftp server` directory. You need to install dependent libraries online.

```
python setup.py install # sudo or root permissions may be required here.
```

Special Notes

This tool is developed with the COS XML API.

Feature Description

Upload Mechanism

The stream upload is adopted and the uploaded file is not saved locally. It works only if the working directory is configured according to the standard FTP protocol, and no disk storage space is occupied actually.

Download Mechanism

The downloaded file is directly returned to the client in the stream download mode.

Directory Mechanism

Bucket serves as the root directory of the entire FTP Server, and multiple subdirectories can be created under Bucket.

Notes

- Now, only one Bucket can be operated each time. But simultaneous operation on multiple Buckets may be supported in the future.
- The FTP Server tool does not support resuming upload from the breakpoint for now.
- An empty file (0B) cannot be uploaded. The maximum file size is 200 GB.

Supported FTP Commands

- put
- mput
- get
- rename
- delete
- mkdir
- ls
- cd
- bye
- quite
- size

Unsupported FTP Commands

- append
- mget (The native mget command is not supported, but on certain Windows clients, such as FileZilla, the files can still be downloaded in batches.)

Configuration File

`conf/vsftpd.conf` is the configuration file of the FTP Server tool. The relevant configuration items are described as follows:

```
[COS_ACCOUNT]
cos_secretid = XXXXXX
cos_secretkey = XXXXXX
# SecretId and SecretKey can be obtained at https://console.cloud.tencent.com/cam/capi.
cos_bucket = BucketName-appid
# The bucket to be operated. Its format is bucektname-appid, such as cos_bucket = mybucket-125888888888.
cos_region = ap-xxx
# Bucket's region. For more information on supported regions, please see [Available Regions - Applicable to the XML API Section]:https://cloud.tencent.com/document/product/436/6224
cos_user_home_dir = /home/cos_ftp/data
# The working directory of the FTP Server.
[FTP_ACCOUNT]
login_users = user1:pass1:RW;user2:pass2:RW
# FTP account configuration. The configuration format is <User name: Password: Read and write permissions>, and multiple accounts are separated by a semicolon.

[NETWORK]
masquerade_address = XXX.XXX.XXX.XXX
# For FTP server located behind a gateway or NAT, you can assign the gateway's IP or domain name to the FTP Server through this configuration item. In general, this configuration item needs not to be configured.
listen_port = 2121
# The listening port (default: 2121) of the Ftp Server. Please note that the firewall needs this port opened.
passive_ports = 60000,65535
# passive_port can be used to set the port range for the passive mode. Default is (60000, 65535).

[FILE_OPTION]
single_file_max_size = 21474836480
# By default, the maximum size of a single file is 200 GB. Too large files are not recommended.

[OPTIONAL]
# For the following settings, take the defaults if there is no special requirements, or 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 recommended not to set it too big. Otherwise, high delay of ls command may occur.
log_level = INFO # Set the log output level
```

```
log_dir = log # Set the directory to store logs. Default is the log directory under the ftp server director
y.
```

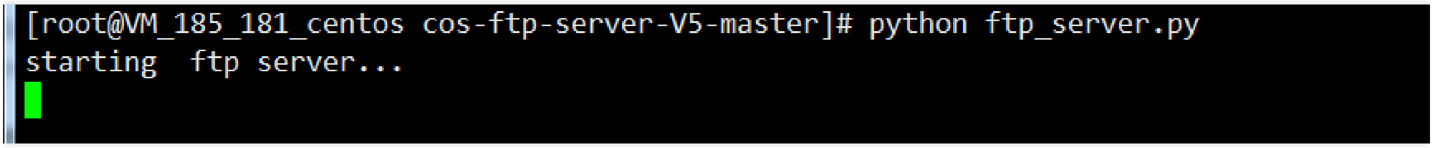
The OPTIONAL part in the configuration is used to adjust the upload performance. In general, it will be fine to take the default values. You can obtain an optimal uploading speed by reasonably adjusting the multipart size and the number of concurrent upload threads based on the server performance. max_connection_num is the limit option for the maximum number of connections, which can be adjusted based on the server condition. If it is set to 0, there is no limit on the maximum number of connections.

Running FTP Server

After the configuration is completed, you can directly run `ftp_server.py` in the root directory via Python to start the FTP Server. You can also use the screen command to make the FTP Server run in the backend.

```
python ftp_server.py
```

After the command is executed, if the result is shown as below, it indicates that the FTP Server service is successfully enabled. And you can access the configured IP and port through the FTP client.



```
[root@VM_185_181_centos cos-ftp-server-V5-master]# python ftp_server.py
starting  ftp server...
```

Stopping FTP Server

The FTP Server (running directly or in the backend with the screen command) can be easily stopped via `Ctrl + C`.

FAQ

What does the masquerade_address option in the configuration file do and when to configure it?

If the FTP Server runs in PASSIVE mode (PASSIVE mode is generally enabled for the FTP client located behind a NAT gateway) on a Server with multiple ENIs, you need to use the masquerade_address option to bind an IP for reply in PASSIVE mode. For example, the FTP Server has multiple IPs (private IP: 10.XXX.XXX.XXX, public IP: 123.XXX.XXX.XXX), and the client adopts PASSIVE mode for transmission. If the public IP is not bound via the masquerade_address option, the private IP may be used when the Server makes a reply in PASSIVE mode. That is, the client can connect to the FTP Server but cannot get any reply data from the Server.

If necessary, it is recommended to set masquerade_address to the IP used by the client to connect with the Server.

If a large file in uploading is cancelled, why does it remain on COS as an uploaded file?

The FTP Server applicable to COS V5 provides the full stream upload capability, and cancelling of file upload or connection break can trigger the completion of large file upload. Then, COS will regard the user's data stream as uploaded completely, and then generate a complete file using the uploaded data. If you want to upload the file again, you can directly upload it with the original file name and overwrite the previous one, or delete the incomplete file and then upload the file again.

Why does the maximum size of uploaded file need to be set in FTP Server configuration?

The maximum number of multiparts to be uploaded to COS is 10,000, and the size of each multipart is limited to 1 MB-5 G. The upper limit on uploaded files is set to calculate the size of a multipart.
The FTP Server supports uploading a single file less than 200 GB by default. However, it is recommended not to set the upper limit to a too big value, because the bigger the set file size, the larger the multipart buffer for uploading. This may increase the consumption of your memory resource. Therefore, it is recommended to set the upper limit on the size of a single file appropriately according to the actual situation.

What will happen if the size of an uploaded file exceeds the upper limit?

If the size of a single uploaded file exceeds the upper limit specified in the configuration file, the system will return an IOError exception and record the error message in the log.

If you have any other questions, please submit a ticket and attach the complete `cos_v5.log` log for troubleshooting.

COSFS Tool

Last updated：2018-09-07 10:02:43

Feature Description

COSFS tool supports mounting COS buckets locally and allows you to operate Tencent Cloud COS directly in the same way as you do with a local file. COSFS has following features:

- Support most of the features of POSIX file system, such as file reading/writing, directory operation, link operation, permission management, uid/gid management.
- Large file transfer.
- Data verification with MD5.

Service Limits

This tool supports access to COS V4 and V5 using COS V5 domain name.

Operating Environment

System Environment

Mainstream Linux system

Software Environment

This tool is compiled in C++ compiling environment, and dependent on such software as automake, git, libcurl-devel, libxml2-devel, fuse-devel, make, openssl-devel. For more information on installation method, please see Environment Installation.

Environment Installation

How to install environment dependency on Ubuntu system

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

How to install environment dependency on CentOS system

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

Note: For CentOS 6.5 or below, a message indicating that the fuse version is too low may be returned when you perform "configure" operation during installation.

```
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 the fuse version by following the procedure below.

```
# yum remove -y fuse-devel
# wget https://github.com/libfuse/libfuse/releases/download/fuse_2_9_4/fuse-2.8.4.tar.gz
# tar -zxvf fuse-2.8.4.tar.gz
# cd fuse-2.8.4
# ./configure
# make
# make install
# export PKG_CONFIG_PATH=/usr/lib/pkgconfig:/usr/lib64/pkgconfig/:/usr/local/lib/pkgconfig
# modprobe fuse
# echo "/usr/local/lib" >> /etc/ld.so.conf
# ldconfig
# pkg-config --modversion fuse
2.8.4 //The display of version indicates successful installation.
```

How to Use

Obtaining Tool

Link on Github: COSFS Tool

Installing Tool

You can directly upload the downloaded source code to the specified directory, or download the code to the specified directory using GitHub. The following example shows how to download source code to `/usr/cosfs` with GitHub:

```
git clone https://github.com/tencentyun/cosfs-v4.2.1 /usr/cosfs
```

Go to the directory to compile and install the tool:

```
cd /usr/cosfs
./autogen.sh
./configure
make
sudo make install
```

Configuration File

In `/etc/passwd-cosfs` file, configure the name of your bucket and the corresponding SecretId and SecretKey. For relevant concepts, please see Concepts. Parameters are separated with a colon. In addition, set read permission for `/etc/passwd-cosfs` . The command format is as follows:

```
echo <bucketname>:<SecretId>:<SecretKey> > /etc/passwd-cosfs
chmod 640 /etc/passwd-cosfs
```

Where:
The bucketname/SecretId/SecretKey should be replaced with your true information.

Example:

```
echo buckettest:AKID8ILGzYjHMG8zhGtnlX7Vi4KOGxRqg1aa:LWVJqIagbFm8IG4sNlrkeSn5DLI3dCYi >
/etc/passwd-cosfs
chmod 640 /etc/passwd-cosfs
```

Running Tool

Execute the following command to mount the configured bucket to the specified directory:

```
cosfs your-APPID:your-bucketname your mount-point -ourl=cos-domain-name -odbglevel=info
```

Where:

- your-APPID/your-bucketname is replaced with your true information.
- your-mount-point is replaced with the local directory to which the bucket needs to be mounted (such as /mnt).
- cos-domain-name is the domain name of the region to which the bucket belongs in a format of `http://cos.<Region>.myqcloud.com` . Region is the region abbreviation for XML API in Available Regions, such as `http://cos.ap-guangzhou.myqcloud.com` , `http://cos.eu-frankfurt.myqcloud.com` .
- The -odbglevel parameter indicates the level of information.

  Example:

```
cosfs 1253972369:buckettest /mnt -ourl=http://cos.ap-guangzhou.myqcloud.com -odbglevel=info
-onoxattr
```

Moreover, if you have a high requirement for the performance, you can use a local disk for file caching. Add -ouse_cache parameter in the command, as shown below:

```
mkdir /local_cache_dir
cosfs 1253972369:buckettest /mnt -ourl=http://cos.ap-guangzhou.myqcloud.com -odbglevel=info
-onoxattr -ouse_cache=/local_cache_dir
```

`/local_cache_dir` is local cache directory, which is not specified if local caching of files is not required, or the capacity of local disk is limited.

Unmount the bucket:

```
fusermount -u /mnt
```

Or

```
umount -l /mnt
```

Common Mounting Options

1. -omultipart_size=[size]
   `multipart_size` is the part size in multipart upload (in MB). Default is 10 MB. Due to a limited number of parts (10,000), for large files larger than 10 MB * 10,000 (100 GB), this parameter should be adjusted as necessary.

2. -oallow_other
   Specify the `allow_other` parameter when you run COSFS to allow other users to access the folder to which the bucket is mounted.

3. -odel_cache
   By default, COSFS will not clear the data cached locally after the bucket is unmounted for optimal performance. If auto removal of cached data is required when COSFS exits, you can add this option when mounting the bucket.

4. -noxattr
   Disable get/setxattr feature which is not supported for the current version of COSFS. If use_xattr is used when you mount the bucket to the disk of the local file, failure may occur when you use mv command to move file to bucket.

Notes

- The capabilities and performance provided by COSFS are limited compared to the local file system. For example, randomly writing data or appending data to the file may cause rewrite of the entire file.
- When a COS bucket is mounted to multiple clients, you need to adjust the behaviors of these clients by yourself. For example, to avoid multiple clients from writing data into the same file.
- The hard link is not supported, which is not applicable to scenarios demanding frequent concurrent read/write.
- Do not mount and unmount files in the current directory. You can use cd command to switch to another directory to mount or unmount file.

FAQs

- How to mount a file to a directory
  You can specify a directory when executing mount command, for example:

  `cosfs appid:my-bucket:/my-dir /tmp/cosfs -ourl=http://cn-south.myqcloud.com -odbglevel=info -ouse_cache=/path/to/local_cache`
  Note: my-dir must begin with /.

- Why can't I write files using the previous version?

  Due to security updates, we highly recommend upgrading to the latest version of COSFS, and try to re-mount.

- For CentOS 6.5 or below, what should I do when I receive a message indicating that the fuse version is too low?

  If the following message is returned when you perform "configure" operation:

  ```
  hecking 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 the fuse version by following the procedure below.

```
# yum remove -y fuse-devel
# wget https://github.com/libfuse/libfuse/releases/download/fuse_2_9_4/fuse-2.8.4.tar.gz
# tar -zxvf fuse-2.8.4.tar.gz
# cd fuse-2.8.4
# ./configure
# make
# make install
# export PKG_CONFIG_PATH=/usr/lib/pkgconfig:/usr/lib64/pkgconfig/:/usr/local/lib/pkgconfig
# modprobe fuse
# echo "/usr/local/lib" >> /etc/ld.so.conf
# ldconfig
# pkg-config --modversion fuse
2.8.4 //The display of version indicates successful installation.
```

- Why did COSFS exit during its normal operation, and "unable to access MOUNTPOINT /path/to/mountpoint: Transport endpoint is not connected" shows when I remount the tool?

  If COSFS is not killed forcibly, check whether the fuse version is older than 2.9.4. The libfuse below 2.9.4 may cause COSFS to exit abnormally.
  It is recommended to update the fuse version or download COSFS V1.0.2 or above. Download link: https://github.com/tencentyun/cosfs/releases

- Why is the Content-Type of files uploaded via COSFS always "application/octet-stream"?

COSFS automatically sets the Content-Type according to /etc/mime.types and the extension of uploaded file. You should check if the file exists on the machine.

For Ubuntu, add files through sudo apt-get install mime-support

For Centos, add files through sudo yum install mailcap

Or you can add files manually with each file in a separate row, such as image/png png.

HDFS TO COS Tools

Last updated：2018-07-19 10:31:00

Feature Description

HDFS TO COS is used to copy the data on HDFS to Tencent Cloud COS.

Operating Environment

System Environment

Linux/Windows system

Software Dependencies

JDK 1.7/1.8

Installation and Configuration

For more information on the installation and configuration of environment, please see Install and Configure Java.

Configuration and Usage

How to Configure

1. For more information on how to install Hadoop-2.7.2 or later, please see Install and Test Hadoop.
2. Download HDFS TO COS at here and decompress it.
3. Copy core-site.xml in the HDFS cluster to be synchronized to the conf folder. core-site.xml contains the configuration information of NameNode.
4. Edit the configuration file `cos_info.conf` , bucket, region, and API key. The provided name of "bucket" should be the full name including the appid provided by Tencent. E.g. "mybucket-125000000".
5. Specify a location for the configuration file in the command line parameter. By default, it is located at `conf/cos_info.conf` .

> **Note:**
> If the command line parameter conflicts with the configuration file, the command line parameter prevails.

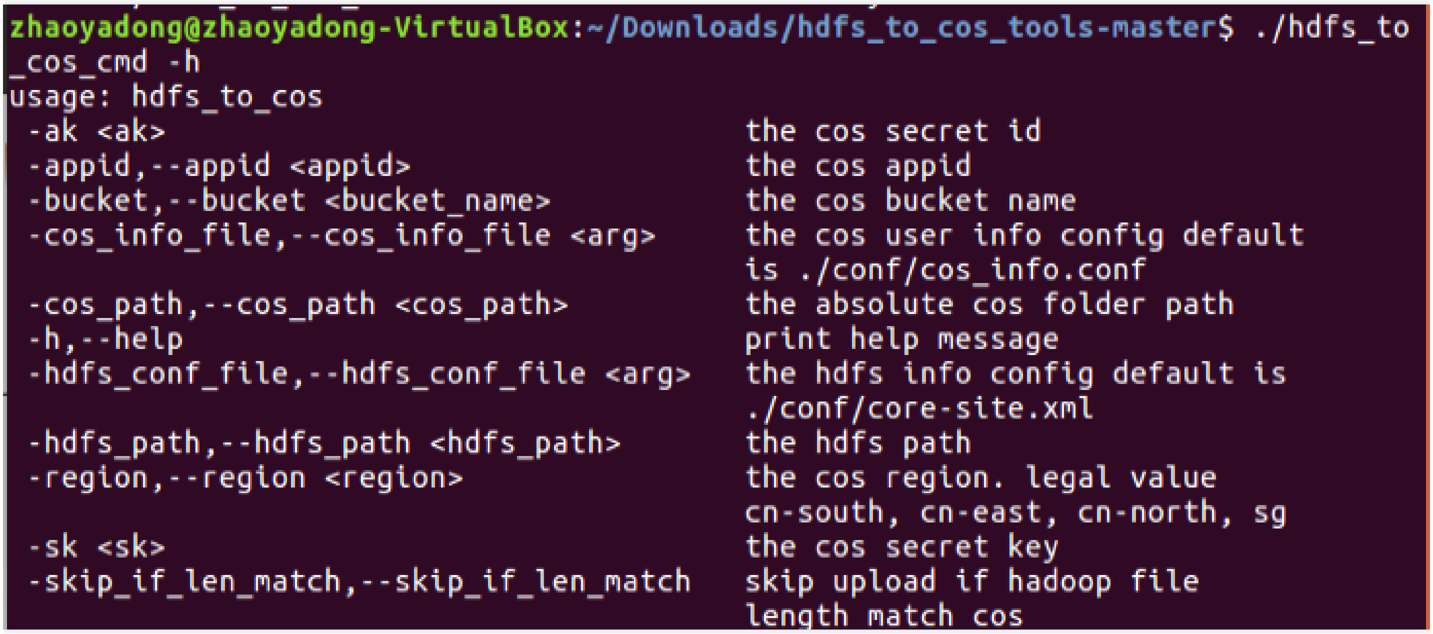
How to Use

(Take Linux as an example)

View Help

```
./hdfs_to_cos_cmd -h
```

The execution result is as follows:



```
zhaoyadong@zhaoyadong-VirtualBox:~/Downloads/hdfs_to_cos_tools-master$ ./hdfs_to
_cos_cmd -h
usage: hdfs_to_cos
 -ak <ak>                                  the cos secret id
 -appid,--appid <appid>                    the cos appid
 -bucket,--bucket <bucket_name>            the cos bucket name
 -cos_info_file,--cos_info_file <arg>      the cos user info config default
                                           is ./conf/cos_info.conf
 -cos_path,--cos_path <cos_path>           the absolute cos folder path
 -h,--help                                 print help message
 -hdfs_conf_file,--hdfs_conf_file <arg>    the hdfs info config default is
                                           ./conf/core-site.xml
 -hdfs_path,--hdfs_path <hdfs_path>        the hdfs path
 -region,--region <region>                 the cos region. legal value
                                           cn-south, cn-east, cn-north, sg
 -sk <sk>                                  the cos secret key
 -skip_if_len_match,--skip_if_len_match    skip upload if hadoop file
                                           length match cos
```

Copying File

- If the file you copy from HDFS to COS has the same name with a file originally stored in COS, the latter will be overwritten.

  ```
  ./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/
  ```

- If the file you copy from HDFS to COS has the same name and length with a file originally stored in COS, the copy will be ignored.

  ```
  ./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/ -skip_if_len_match
  ```

  Only file length is used as a metric, because calculating file summaries of Hadoop takes too much work.

Directories

```
conf: Configuration file, which stores core-site.xml and cos_info.conf
log: log directory
src: Java source program
dep: Runnable JAR package generated after compilation
```

Q&A

About Configuration Information

Make sure you enter the correct configuration information, including bucket, region and API key. The provided name of "bucket" should be the full name including the appid provided by Tencent. E.g. "mybucket-125000000". Make sure the time difference between the server and Beijing time does not exceed 1 minute. Otherwise, reset the server time.

About DateNode

Make sure DateNode can be connected with the server that contains the replication program. NameNode can be connected because it has a public IP. But the DateNode server that contains the obtained block only has a private IP, it cannot be connected directly. It is recommended to execute the synchronization program at a Hadoop node, so that both NameNode and DateNode can be accessed.

About Permissions

Use Hadoop command to download and check files, and then use synchronization tools to synchronize data on Hadoop.

About File Overwriting

If the file you copy from HDFS to COS has the same name with a file originally stored in COS, the latter will be overwritten by default. If a user specifies `-skip_if_len_match` , the copy will be skipped when the file you copy from HDFS to COS has the same name and length with a file originally stored in COS.

About COS path

COS path is a directory by default, and all the files copied from HDFS are stored in the directory.