

# **Cloud Object Storage**

# ツールガイド

# 製品ドキュメント





#### **Copyright Notice**

©2013-2024 Tencent Cloud. All rights reserved.

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

Trademark Notice

### STencent 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.

### カタログ:

ツールガイド ツール概要 環境のインストールと設定 Javaのインストールと設定 Pythonのインストールと設定 Hadoopのインストールとテスト COSBrowserツール COSBrowser概要 デスクトップでの使い方 COSCLIツール COSCLI概要 ダウンロードとインストール設定 よく使用するコマンド 設定ファイルの発行と変更 - config バケットの作成 - mb バケットの削除 - rb バケットまたはファイルリストの照会 - ls さまざまなファイルの統計情報を取得 - du ファイルのアップロード・ダウンロードまたはコピー - cp ファイルの同時アップロード・ダウンロードまたはコピー - sync ファイルの削除 - rm ファイルハッシュ値の取得 - hash マルチパートアップロード中に発生したフラグメントを一覧表示 - Isparts フラグメントのクリーンアップ - abort アーカイブファイルの取得 - restore よくある質問 COSCMDツール COS Migrationツール FTP Serverツール Hadoopツール COSDistCpツール Hadoop-cos-DistChecker $\mathcal{V} - \mathcal{V}$ HDFS TO COSツール

オンライン補助ツール

COSリクエストツール



セルフ診断ツール

# ツールガイド

ツール概要

最終更新日:::2024-06-26 09:59:21

ツール	機能の説明
<b>COSBrowser</b>	このツールは、ユーザーがビジュアルインターフェースを介して、データのアップロード、
ツール	ダウンロードおよびアクセスリンクの発行といった操作を簡単に実行できるようにします。
COSCLIツー ル	COSCLIは、Tencent CloudのCloud Object Storageが提供するクライアントサイドのコマン ドラインツールです。COSCLIツールを使用すれば、シンプルなコマンドラインの指示で、 COS内のオブジェクト(Object)に対し、一括アップロード・ダウンロード・削除などの操作 を実行することができます。
COSCMD	このツールは、ユーザーが簡単なコマンドラインを使用して、オブジェクトの一括アップ
ツール	ロード・ダウンロード・削除などの操作を実行できるようサポートします。
COS Migration ツール	このツールは、ユーザーのローカルや他のクラウドストレージなど、さまざまなソースデー タアドレスから <b>COS</b> への移行をサポートします。
<b>FTP Server</b>	このツールは、ユーザーがFTPクライアントを使用してCOSから行うファイルのアップロー
ツール	ド・ダウンロードをサポートします。
COSFSツー	Linuxシステムでは、このツールを使用してバケットをローカルファイルシステムにマウン
ル	トし、ローカルファイルシステム経由でCOS上のオブジェクトを操作することができます。
Hadoopツー ル	Hadoopツールは、Hadoop、SparkおよびTezなどのビッグデータコンピューティングフレー ムワークをCOSに統合してサポートを提供することで、COSに保存されたデータの読み書 きを容易に行うことができます。
COSDistcp	COSDistcpは、MapReduceをベースとする分散ファイルコピーツールであり、主にHDFSと
ツール	COS間でデータをコピーするときに使います。
Hadoop-cos- DistChecker ツール	Hadoop-cos-DistCheckerは、hadoop distcpコマンドを使用してHDFSからCOSにデータを移 行した後、移行ディレクトリの整合性を検証するときに使います。
HDFS TO COSツール	このツールは、HDFSからCOSにデータをコピーするときに使います。
自己診断	自己診断ツールは、Tencent Cloud COSがユーザー向けに提供するWebツールで、エラーリ
ツール	クエストのセルフチェックとトラブルシューティングができます。



他のツールの要件がある場合は、ツール要件をお知らせください。速やかに要件の評価を行います。

# 環境のインストールと設定 Javaのインストールと設定

最終更新日:::2024-06-26 09:59:21

JDKはJavaソフトウェア開発ツールキットです。ここでは、JDK 1.8バージョンを例として、WindowsとLinuxシステムにおけるJDKのインストールと環境設定の手順についてそれぞれご紹介します。

### Windows

### 1.JDKのダウンロード

Oracle公式サイトに進み、適切なJDKのバージョンをダウンロードします。

### 2. インストール

プロンプトに従って、ステップバイステップでインストールします。インストール中にインストールディレクトリ をカスタマイズすることができます(デフォルトではCドライブにインストールします)。例えば、選択するイン ストールディレクトリは、 D:\\Program Files\\Java\\jdk1.8.0\_31 と D:\\Program Files\\Java\\jre1.8.0\_31 です。

### 3. 設定

インストールが完了したら、コンピュータ>プロパティ>高度なシステム設定>環境変数>システム変数>新規作成を 右クリックして、ソフトウェアをそれぞれ設定します。

変数名(N):**JAVA\_HOME**変数値(V): D:\\Program Files\\Java\\jdk1.8.0\_31 (実際のインストールパ スに従って設定してください)。

#### 変数名(N): CLASSPATH変数值

(V): .;%JAVA\_HOME%\\lib\\dt.jar;%JAVA\_HOME%\\lib\\tools.jar; (変数の値は . から始まりま すので、ご注意ください)。

変数名(N):Path

変数値(V): %JAVA\_HOME%\\bin;%JAVA\_HOME%\\jre\\bin;

### 4. テスト

設定が成功したかどうかをテストするには、スタート(またはショートカットキー:Win+R)>実行(cmdを入 力)>OK(またはEnterキーを押す)をクリックし、コマンドjavacを入力してEnterキーを押します。下図のよう な情報が表示されれば、環境変数の設定は成功です。

### Linux

yumまたはapt-getコマンドを使用してopenjdkをインストールすると、クラスライブラリが不完全になり、ユー ザーがインストール後に関連ツールを実行する際、エラーが報告される場合がありますので、ここでは手動で解 凍してJDKをインストールする方法をお勧めします。具体的な手順は、次のとおりです。

### 1.JDKのダウンロード

Oracle公式サイトに進み、適切なJDKのバージョンをダウンロードし、インストールの準備を行います。 注意:

以下は、例としてjdk-8u151-linux-x64.tar.gzを取り上げています。他のバージョンをダウンロードする場合は、 ファイルの拡張子が.tar.gzであることに注意してください。

### 2. ディレクトリの作成

以下のコマンドを実行し、/usr/ディレクトリにjavaディレクトリを作成します。





mkdir /usr/java cd /usr/java

ダウンロードしたファイルjdk-8u151-linux-x64.tar.gzを/usr/java/ディレクトリにコピーします。

### 3.JDKの解凍

次のコマンドを実行して、ファイルを解凍します。





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

### 4. 環境変数の設定

/etc/profileファイルを編集し、profileファイルに次の内容を追加して保存します。





# 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

### 注意:

ここでJAVA\_HOME、JRE\_HOMEは、実際のインストールパスとJDKのバージョンに応じて設定してください。 変更を有効にするには、以下を実行します。





source /etc/profile

### 5. テスト

次のコマンドを実行して、テストを行います。





java -version

Javaのバージョン情報が表示されれば、JDKのインストールは成功です。





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のインストールと設定

最終更新日:::2024-06-26 09:59:21

ここでは、さまざまなオペレーティングシステムにPython開発環境をインストールする方法を簡単にご紹介します。

インストールパッケージによるインストール

### 1. ダウンロード

Python公式サイトにアクセスし、使用しているOSに応じて適切なインストールパッケージを選択し、ダウンロー ドします。

### 注意:

Pythonは、2020年1月1日をもってPython2のメンテナンスを終了しており、Python3のインストールが推奨されています。

### 2. インストール

インストールパッケージをダウンロードしたら、インストールパッケージの指示に従って、Python開発環境のインストールを完了します。

### 説明:

Windowsシステムユーザーは、インストール時にオプションの「Add Python to environment variables」にチェックを入れるようご注意ください。



### 3. 検証

端末で以下のコマンドを実行し、Pythonのバージョンを確認します。





python -V

端末からPythonのバージョン番号が出力されれば、インストールは成功です。

説明:

Windowsシステムユーザーはインストール完了後、コンピュータの再起動が必要になる場合があります。

### 4. 環境変数の設定

Windowsシステムで上記のコマンドを実行した際、端末に「内部コマンドでも外部コマンドでもありません」と 表示された場合は、【コンピュータ】>【プロパティ】>【高度なシステム設定】>【環境変数】>【システム変数 (S)】で「Path」を編集し、Pythonのインストールパスを追加してください。

パッケージマネージャーによるインストール

Mac OS

Mac OSをお使いのユーザーは、あらかじめHomeBrewをインストールした上で、HomeBrew経由でPythonをイン ストールすることができます。



brew install python

### Ubuntu

Ubuntuをお使いのユーザーは、Ubuntuに付属しているapt(Advanced Packaging Tool)パッケージマネージャーを使用すると、Pythonをインストールすることができます。



sudo apt-get install python

### CentOS

CentOSをお使いのユーザーは、CentOSに付属しているyum(Yellow dog Updater, Modified)パッケージマネー ジャーを使用すると、Pythonをインストールすることができます。



sudo yum install -y python

# Hadoopのインストールとテスト

最終更新日:::2024-06-26 09:59:21

Hadoopツールは、Hadoop-2.7.2以降のバージョンに依存して、基盤となるファイルストレージシステムとして Tencent Cloud COSを使用し、上位層のコンピューティングタスクを実行する機能を実装しています。Hadoopク ラスターを起動するには、主にスタンドアロン、疑似分散、完全分散という3つの主なモードがあります。ここで は主に、Hadoop-2.7.4バージョンを例として、完全分散Hadoop環境をビルド、およびwordcountの簡単なテストを ご紹介します。

### 環境の準備

1. 複数のマシンを準備します。

2. システム: CentOS-7-x86\_64-DVD-1611.isoをインストールして設定します。

Java環境をインストールします。操作の詳細については、Javaのインストールと設定をご参照ください。
 Hadoopの利用可能なパッケージApache Hadoop Releases Downloadをインストールします。

### ネットワークの設定

ifconfig -a を使用して各マシンのIPを確認し、pingコマンドを使用して相互にpingを送信できるかどうかを 確認し、各マシンのIPを記録します。

## **CentOS**の設定

hostsの設定





vi /etc/hosts

コンテンツの編集:





202.xxx.xxx.xxx master 202.xxx.xxx.xxx slave1 202.xxx.xxx.xxx slave2 202.xxx.xxx.xxx slave3 //IPアドレスを実際のIPに置き換えます

ファイアウォールの無効化





```
systemctl status firewalld.service //ファイアウォールのステータスをチェックします
systemctl stop firewalld.service //ファイアウォールを無効にします
systemctl disable firewalld.service //ファイアウォールの起動を無効にします
```

### 時刻同期





yum install -y ntp //ntpサービスをインストールします ntpdate cn.pool.ntp.org //ネットワークの時刻を同期させます

### JDKのインストールと設定

JDKインストールパッケージ(jdk-8u144-linux-x64.tar.gzなど)を root ルートディレクトリにアップロードしま す。





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

各ホスト間でのJDKのコピー





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

各ホストのJDK環境変数の設定





vi /etc/profile

コンテンツの編集:





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 //設定ファイルを有効にします java -version //Javaのバージョンを確認します

### SSHキーレスアクセスの設定

各ホストのSSHサービスステータスを個別にチェックします。





systemctl status sshd.service //SSHサービスのステータスをチェックします yum install openssh-server openssh-clients //SSHサービスをインストールします。すでにインス systemctl start sshd.service //SSHサービスを開始します。すでにインストールされている場合は、

各ホストで個別にキーを発行します





ssh-keygen -t rsa //キーの発行

slave1の場合:





cp ~/.ssh/id\_rsa.pub ~/.ssh/slave1.id\_rsa.pub scp ~/.ssh/slave1.id\_rsa.pub master:~/.ssh

slave2の場合:





cp ~/.ssh/id\_rsa.pub ~/.ssh/slave2.id\_rsa.pub scp ~/.ssh/slave2.id\_rsa.pub master:~/.ssh

という感じで.. masterの場合:





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

## Hadoopのインストールと設定

### Hadoopのインストール

hadoop1ンストールパッケージ (hadoop-2.7.4.tar.gzなど) を root ルートディレクトリにアップロードしま す。



```
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
```

hadoop-2.7.4/etc/hadoop ディレクトリに移動し、次の操作に進みます。

### Hadoopの設定

1. hadoop-env.sh ファイルを変更して、以下を追加します



export JAVA\_HOME=/usr/java/jdk1.8.0\_144
SSHポートがデフォルトの22でない場合は、 hadoop-env.sh ファイルで以下のように変更します。



export HADOOP\_SSH\_OPTS="-p 1234"

2. yarn-env.sh を変更します





export JAVA\_HOME=/usr/java/jdk1.8.0\_144

# 3. slaves を変更します

設定内容:





削除: localhost 追加: slave1 slave2 slave3

4. core-site.xml を変更します





```
<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. hdfs-site.xml を変更します



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



Cloud Object Storage

```
<final>true</final>
</property>
</configuration>
```

6. mapred-site.xml を変更します



<configuration> <property> <name>mapreduce.framework.name</name> <value>yarn</value>

```
</property>
<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. yarn-site.xml を変更します





```
<configuration>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.mapred.ShuffleHandler</value>
</property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</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>
 <property>
    <name>yarn.resourcemanager.admin.address</name>
    <value>master:8033</value>
 </property>
 <property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>master:8088</value>
  </property>
</configuration>
```

#### 8. ホスト間でHadoopを複製します





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. 各ホストのHadoop環境変数を設定します

設定ファイルを開きます。





vi /etc/profile

コンテンツの編集:





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

設定ファイルを有効にします。





source /etc/profile

# Hadoopの起動

1. namenodeのフォーマット





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

# 2. 起動





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

# 3. プロセスのチェック

masterホストにResourceManager、SecondaryNameNode、NameNodeなどが含まれている場合は、次のような表示があれば起動に成功しています。





2212 ResourceManager2484 Jps1917 NameNode2078 SecondaryNameNode

各slaveホストにDataNode、NodeManagerなどが含まれている場合は、次のような表示があれば起動に成功しています。





17153 DataNode 17334 Jps 17241 NodeManager

# wordcountの実行

Hadoopにはwordcountルーチンが付属されているため、直接呼び出すことができます。Hadoopを起動した後、以下のコマンドを使用すれば、HDFS内のファイルを操作することができます。





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

not a fatta onth faorfinadoop Erritzionare/nadoop/mapreadee enampted Erritzijan
[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

上図に示されている結果は、Hadoopのインストールが成功したことを示しています。

# 出力ディレクトリの確認





hadoop fs -ls /output

出力結果の確認





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

[root@VM	1_96_24_centos	/usr/hadoop-2	2.7.4]#	hadoop	fs	-cat	/output/pa	rt-r-0000	90
a	5								
dasdada	1								
ds	2								
qwe	1								
ret	1								
S	1								
V	2								
vdfd	1								
wqere	1								

# 説明:

スタンドアロンモードと疑似分散モードの詳細な操作方法については、公式サイトのHadoop入門をご参照ください。

# COSBrowserツール COSBrowser概要

最終更新日:::2024-06-26 09:59:21

COSBrowserはTencent Cloud Object Storage (COS) がリリースした視覚化インターフェースツールです。より 簡単なインタラクションの使用を可能にし、COSリソースの確認、転送および管理を手軽に実現できます。デー タ移行またはデータの一括アップロードを行う場合は、Migration Service Platform (MSP) をご利用ください。現 在COSBrowserはデスクトップ端末およびモバイル端末向けにご提供しています。詳細については以下をご参照く ださい。

デスクトップの使用説明

モバイル端末の使用説明

# ダウンロードアドレス

COSBrowser カテゴリー	サポート プラット フォーム	システム要件	
	Windows	Windows 7 32/64ビット以上、Windows Server 2008 R2 64ビット 以上	Windows
デスクトッ	macOS	macOS 10.13以上	macOS
プ	Linux	グラフィカルインターフェースとサポートが必要ですAppImage 形式 注意:CentOSでクライアントを起動するに は、/cosbrowser.AppImageno-sandboxを実行する必要がありま す	Linux
モバイル端	Android	Android 4.4以上	Android
末	iOS	iOS 11以上	iOS
Web版	Web	Chrome/FireFox/Safari/IE10以上のブラウザ	

デスクトップ機能リスト

COSBrowserデスクトップはリソース管理に重点を置いており、ユーザーはCOSBrowserからデータを一括でアッ プロード・ダウンロードすることができます。

# 注意:

COSBrowserデスクトップは、システムによって設定されたプロキシを使用してインターネットへの接続を試みま す。プロキシ設定が正常であることを確認するか、インターネットに接続できないプロキシ設定を無効にしてくだ さい。

Windowsユーザーは、OSの「インターネットオプション」で確認することができます。

macOSユーザーは、「ネットワーク環境設定」で確認することができます。

Linuxユーザーは、システム設定>ネットワーク>ネットワークプロキシで確認することができます。

COSBrowserデスクトップは、以下の機能をサポートします。

機能	機能説明
バケットの作成/削除	バケットの作成と削除をサポートします
バケットの詳細情報の表 示	バケットの基本情報の表示をサポートします
統計データの表示	バケットの現在のストレージ容量とオブジェクト総数の表示をサポートします
権限管理	バケットやオブジェクトに関する権限の変更をサポートします
バージョン管理の設定	バケットのバージョン管理の有効化と一時停止をサポートします
アクセスパスの追加	アクセスパスの追加をサポートします
ファイル/フォルダのアッ プロード	ファイルまたはフォルダのバケットへの単一アップロード、一括アップロー ド、増分アップロードをサポートします 注意: 一括アップロードするファイルは10万個以内とします 中断からの再開はサポートしていません データ移行またはデータの一括アップロードを行う場合は、Migration Service Platform (MSP)をご利用ください
ファイル/フォルダのダウ ンロード	ファイルまたはフォルダのローカルへの個別ダウンロード、一括ダウンロー ド、差分ダウンロードをサポートします 注意: 一括ダウンロードするファイルは10万個以内とします 中断からの再開はサポートしていません
ファイル/フォルダの削除	バケット内のファイルやフォルダの個別削除、一括削除をサポートします
ファイルの同期	ローカルファイルのバケットへのリアルタイム同期をサポートします
ファイルのコピーと貼り 付け	1つのディレクトリから別のディレクトリへのファイルまたはフォルダへの個別 コピー、一括コピーをサポートします



ファイルのリネーム	バケット内のファイルのリネームをサポートします
フォルダの新規作成	バケット内でのフォルダの新規作成をサポートします
ファイルの詳細を表示	バケット内のファイルの基本情報の表示をサポートします
ファイルリンクの発行	一時的な署名をリクエストすることで、制限時間のあるファイルへのアクセス リンクの発行をサポートします
ファイル/フォルダの共有	ファイルとフォルダの共有と、共有の有効時間の設定をサポートします
ファイルリンクのエクス ポート	ファイルリンクの一括エクスポートをサポートします
ファイルプレビュー	バケット内のメディアファイル(画像・ビデオ・オーディオ)のプレビューを サポートします
ファイル検索	プレフィックス検索によるバケット内のファイル検索をサポートします
バケットの検索	作成済みのバケットの検索をサポートします
バージョンの履歴または ファイルフラグメントの 表示	バージョン管理が有効になっているバケット内のファイルのバージョン履歴の 表示をサポートします バケット内のファイルフラグメントの詳細の表示をサポートします
ファイル比較	ローカルフォルダとバケット内のファイルの比較をサポートします
ビデオトランスコード	MPS機能を有効にしているバケット内のファイルのトランスコードをサポート します
権限承認コードの生成	佐田丞知一 じにし てつつつし いちご ノマンした 吐めに一ビノンでナナ
	椎限承認コートによってCOSBrowserクライアントに一時的にロクインできます
画像処理	権限承認コートによってCOSBrowserクライアントに一時的にロクインできます ズーム、トリミング、回転などの基本的な画像処理およびテキストウォーター マーク、画像ウォーターマーク、処理後の画像リンクの生成をサポートします
画像処理 ネットワークプロキシの 設定	権限承認コートによってCOSBrowserクライアントに一時的にロクインできます ズーム、トリミング、回転などの基本的な画像処理およびテキストウォーター マーク、画像ウォーターマーク、処理後の画像リンクの生成をサポートします COSにアクセスするためのネットワークプロキシの設定をサポートします
画像処理 ネットワークプロキシの 設定 ファイルのアップロード/ ダウンロード同時実行数 の設定	権限承認コートによってCOSBrowserクライアントに一時的にロクインできます ズーム、トリミング、回転などの基本的な画像処理およびテキストウォーター マーク、画像ウォーターマーク、処理後の画像リンクの生成をサポートします COSにアクセスするためのネットワークプロキシの設定をサポートします ファイルのアップロード・ダウンロードの伝送同時実行数の設定をサポートし ます
画像処理 ネットワークプロキシの 設定 ファイルのアップロード/ ダウンロード同時実行数 の設定 アップロード/ダウンロー ドマルチパート同時実行 数の設定	権限承認コートによってCOSBrowserクライアントに一時的にログインできま す ズーム、トリミング、回転などの基本的な画像処理およびテキストウォーター マーク、画像ウォーターマーク、処理後の画像リンクの生成をサポートします COSにアクセスするためのネットワークプロキシの設定をサポートします ファイルのアップロード・ダウンロードの伝送同時実行数の設定をサポートし ます ファイルのマルチパートアップロード・ダウンロードのパート数の設定をサ ポートします

ド失敗時の再試行回数の 設定	サポートします
シングルスレッドアップ ロード/ダウンロード速度 制限の設定	シングルスレッドのアップロードとダウンロードの速度制限の設定をサポート します
アップロードの二次 チェックの設定	バケットにアップロードされたファイルの二次チェックをサポートします
ローカルログの表示	COSBrowserへのユーザー操作記録のローカルログ形式による保存をサポート します

# モバイル端末機能リスト

COSBrowserモバイル端末は、リソースの表示とモニタリングに重点を置いており、ユーザーはCOSのストレージ 量やトラフィックなどのデータをいつでもどこでもモニタリングすることができます。COSBrowserモバイル端末 でサポートされている機能については、モバイル端末機能リストをご参照ください。

# ログの更新

デスクトップ更新ログ: changelog。 モバイル端末更新ログ: changelog\_mobile。

# フィードバックと提案

COSBrowserの使用に関してご質問やご提案がございましたら、フィードバックをお寄せください。 デスクトップのフィードバック:issues。 モバイル端末のフィードバック:issues\_mobile。

# デスクトップでの使い方

最終更新日:::2024-06-26 09:59:21

# ダウンロードとインストール

# ソフトウェアのダウンロード

サポートプラット フォーム	システム要件	ダウンロードアドレス
Windows	Windows 7 32/64ビット以上、Windows Server 2008 R2 64ビット以上	Windows
macOS	macOS 10.13以上	macOS
Linux	グラフィカルインターフェースとAppImage形式 をサポートしている必要があります	Linux
Web版	Chrome/FireFox/Safari/IE10以上などのブラウザ	Web

# ソフトウェアのインストール

ユーザーはプラットフォームに応じて、プログラムパッケージによるインストール、解凍してインストール、イン ストールせずブラウザで直接使用する方法のいずれかを選択できます。

#### 注意:

**CentOS**でクライアントを起動するには、ターミナルで ./cosbrowser.AppImage --no-sandbox を実行します。

# ログインソフトウェア

COSBrowserデスクトップは、パーマネントキーログイン、Tencent Cloudアカウントログイン、共有リンクログ インという3つのログイン方法をサポートしています。同一アカウントでのマルチデバイス・マルチポイントログ インもサポートしています。

# パーマネントキーログイン

Tencent Cloud APIキー(SecretID、SecretKey)でログインして使用します。このキーはCAMコンソールのAPI キー管理のページから作成、取得することができます。ログインに成功すると、キーは**履歴セッション**に保存さ れ、次回から続けて使用できるようになります。ソフトウェアのログインインターフェースを以下に示します。そ の他の設定項目の説明は以下のとおりです。 バケット/アクセスパス:現在使用しているキーがそのルートアカウントからアクセスを許可されている範囲が、 バケットまたはバケット内のいずれかのディレクトリの権限のみである場合は、入力必須となります。入力後すぐ に対応するファイルパスに入ることができます。形式はBucketまたはBucket/Object-prefixとなります。例えば、現 在使用しているキーがバケットexamplebucket-1250000000内のdocフォルダへのアクセスのみ許可されている場 合は、examplebucket-1250000000/docと入力します。

備考:オペレーターや用途など、現在入力されているパーマネントキーの説明を記述することができます。履歴 キーインターフェースで履歴セッションを管理する際、さまざまなSecretIDを区別することが可能です。

#### セッションの記憶:

チェックが入っていない場合、現在のログインのみが有効となり、ログアウトすると入力されたTencent Cloud APIキーがクリアされます(現在のキーが履歴セッションに保存されている場合は、履歴セッションから削除され ます)。

チェックが入っている場合、入力したTencent Cloud APIキーは記憶され、履歴セッションで管理できるようにな ります。

#### 注意:

COSBrowserは、プロジェクトキーでのログインをサポートしていません。





# Tencent Cloud アカウントでのログイン

**Tencent Cloudアカウントでのログイン**をクリックして、ポップアップウィンドウからTencent Cloudアカウント でCOSBrowserデスクトップにログインします。Tencent Cloudアカウントでサポートされているログイン方法に は、WeChat、電子メール、QQ、WeChat公式アカウント、WeComおよびサブユーザーログインがあります。具 体的なガイドについては、アカウント関連ドキュメントをご参照ください。

# 共有リンクログイン

他の人から転送または共有された**共有リンク**と**抽出コード**によって、COSBrowserデスクトップに一時的にログインすることができます。機能の説明についてはフォルダの共有をご参照ください。

# 基本機能

# 注意:

この機能説明はWindowsのバージョンv 2.8.4の例です。その他のバージョンでは違いがありますので、更新ログをご参照ください。

# 1. バケットの作成/削除

機能	説明	操作方法
バ ケッ トの 作成	クライアントから直接バケッ トを作成できます	1. バケットリストで左上隅の <b>バケットの追加</b> をクリックします  2. バケット名を正しく入力し、リージョンやアクセス権限 を選択します 3. OKをクリックすると、作成が完了します
バ ケッ トの 削除	バケットを削除する前に、バ ケット内のデータがクリアさ れていることを確認してくだ さい	1. バケットリストで、対応するバケットの右側にある <b>削除</b> をク リックします <b> 2</b> . バケット内のデータがすべてクリアされて いることを確認し、 <b>OK</b> をクリックして削除します

### 2. バケット詳細の表示

バケットリストの右側にある**詳細**アイコンをクリックすると、バケット詳細を表示できます。バケットの詳細情報 には、バケット名、リージョン、アクセス権限、マルチAZ特性、バージョン管理、グローバルアクセラレーショ ンなどのステータスおよびバケットドメイン名が含まれます。

#### 注意:

マルチAZ特性は現在、北京、広州、上海、シンガポールリージョンなど、一部のリージョンでのみサポートされています。マルチAZ特性に関する説明は、マルチAZの特徴の概要をご参照ください。

### 3. 統計データの表示

バケットリストの右側にある …>統計 をクリックすると、バケットの統計データを表示できます。バケットデータ には、ストレージの使用量とオブジェクト数が含まれます。

### 4. 権限管理

COSBrowserは、バケットとファイルの権限管理をサポートしています。 バケット権限管理:バケットリストの右側にある操作バーの**権限管理**をクリックすると操作できます。 オブジェクト権限管理:ファイルの右側にある操作バーの**権限管理**をクリックすると操作できます。

#### 説明:

COSの権限に関する説明については、ACLの概要をご参照ください。

### 5. バージョン管理の設定

COSBrowserは、バケットのバージョン管理の有効化/無効化をサポートしています。 バケットリストの右側にある操作バーの**詳細**をクリックし、バケット詳細のポップアップウィンドウで、バー ジョン管理編集アイコンをクリックして操作できます。

説明:

バージョン管理に関する説明については、バージョン管理の概要をご参照ください。

### 6. アクセスパスの追加

バケットリストにアクセスできないサブアカウントでログインしている場合、**アクセスパスを追加**することでア クセスできるようになります。COSBrowserでは、アクセスパスの追加方法として、次の2つの方法をご提供して います。

(1) ログインインターフェースに直接アクセスパスを追加し、対応するバケットのリージョン情報を選択しま す。ログインが完了すると、リソースを管理することができます。





(2) サブアカウントにログインした後、バケットリスト画面の左上隅にあるパスの追加をクリックし、指定した パスを入力してバケット管理リソースに進みます。

COSBrowser		
Bucket List		
Add Bucket	Add Path	
Basic Info		

# 7. ファイル/フォルダのアップロード

アッ プロ ド機 能	説明	操作方法
ファルのアップロド	COSBrowserは、単一または一括 アップロードといった複数のアップ ロード方法をサポートしています	指定したバケットまたはパスにファイルをアップロードす るには、次のような方法があります。1. アップロードをク リックしてファイルを選択し、ファイルを直接アップロー ドします。2. ファイルリストの空白部分にあるファイルの アップロードを右クリックしてアップロードします。3. ファイルをマウスでファイルリストウィンドウにドラッグ してアップロードします。
フルとのフイのアプロド	バケットまたはパス内に同名のファ イルやフォルダが存在する場合、デ フォルトで上書きされます	指定したバケットまたはパスにフォルダをアップロードす るには、次のような方法があります。1. アップロードをク リックしてフォルダを選択し、フォルダを直接アップロー ドします。2. ファイルリストの空白部分にあるフォルダの アップロードを右クリックで選択してアップロードしま す。3. フォルダをマウスでファイルリストウィンドウにド ラッグすればアップロード完了です。
イン クリ メン タル	インクリメンタルアップロードと は、アップロード操作を実行する前 に、アップロードされたファイルと バケット内の既存のオブジェクトを	インクリメンタルアップロード操作は、指定したバケット またはパス内で、次のように実行します。1.フォルダの アップロード方法を使用するには、「次へ」をクリックし ます。2.ストレージ方法でスキップを選択し、アップロー

アッ	比較し、同名のオブジェクトがある	ドをクリックすると、インクリメンタルアップロードが完
プ	場合、そのファイルをスキップして	了します。
<u>п</u> –	アップロードしないことをいいます	
ド		

# 注意:

大容量のファイルをアップロードする必要がある場合は、4コアでRAMが16GBのスペックのコンピュータをお勧めします(一度に最大30万ファイルのアップロードが可能です)。

### 8. ファイル/フォルダのダウンロード

ダウ ン ロ ド 機 能	説明	操作方法
ファイのクロド	COSBrowserは、単一または一括ダ ウンロードといった複数のダウン ロード方法をサポートしています	ファイルをダウンロードするには、次のような方法があり ます。1. ダウンロードしたいファイルを選択し、画面内の ダウンロードをクリックすると、ファイルがダウンロード されます。2. ファイルを選択し、ダウンロードを右クリッ クします。3. ファイルをマウスでローカルにドラッグして ダウンロードします。
フルとのフイのウロドオダそのアルダンー	ローカルに同名のファイルやフォル ダが存在する場合、デフォルトでリ ネームが行われます	フォルダとそのファイルをダウンロードするには、次のよ うな方法があります。1. ダウンロードしたいフォルダを選 択し、画面内のダウンロードをクリックするとダウンロー ドできます。2. ダウンロードを右クリックして、フォルダ を直接ダウンロードします。3. フォルダをマウスでローカ ルにドラッグしてダウンロードします。
イフノメタクンロド	インクリメンタルダウンロードと は、ダウンロード操作を実行する前 に、ダウンロードされたオブジェク トとローカルファイルを比較し、同 名のオブジェクトがある場合、その オブジェクトをスキップしてダウン ロードしないことをいいます	インクリメンタルダウンロード操作の手順は以下のとおり です。1. ダウンロードしたいファイル/フォルダを選択し、 マウスで右クリックします。2. メニューの高度なダウン ロードをクリックし、ポップアップウィンドウでスキップ を選択します。3. 次に、今すぐダウンロードをクリックす ると、異なる名前のファイル/フォルダのインクリメンタル ダウンロードが完了します。

# 注意:

多数のファイルをダウンロードする必要がある場合は、4コアでRAMが16GBのスペックのコンピュータをお勧め します(一度に最大30万ファイルのダウンロードが可能です)。

### 9. ファイル/フォルダの削除

削除したいファイル/フォルダを選択し、画面上の**その他**にある**削除**をクリックするか、**削除**を右クリックすると ファイル/フォルダが削除されます。一括削除が可能です。

### 10. ファイル同期

ユーザーは、ファイル同期機能を使用することで、指定したローカルフォルダ内のファイルを自動でかつリアル タイムにバケット内にアップロードするか、または一定の時刻にバケット内にアップロードすることができます。 具体的な操作手順は以下のとおりです。

1. 画面右上のツールボックス>ファイル同期をクリックします。

2. ポップアップウィンドウで同期アップロードしたいローカルフォルダと、COSにアップロードするバケット ディレクトリを指定します。

3. 同期タイプから1回のみの同期、自動同期、定時同期を選択できます。その後、**同期開始**をクリックすると、 ファイル同期機能が有効になります。

4. ファイルの同期履歴は同期ログで確認できます。

COSBrowser		
Bucket List	Sync Setting	
Add Bucket Add Path	Local Folder * Select Local Folder	Change
Basic Info	Bucket Path *	Rucket Path
examplebucket-1254	Auto Sync	
examplebucket-1254	Support incremental upload only. See the documen Start Sync Clear Local Cache	tation for details.

#### 注意:

同期とは、ファイルをアップロードする際に、バケットに同じファイルが存在するかどうかをシステムが自動認 識し、同期機能によってバケットに存在しないファイルのみをアップロードすることをいいます。

現在、ローカルファイルのストレージバケットへの同期アップロードのみサポートされており、逆方向の操作は サポートされていません。

ファイル同期機能は、手動同期、自動同期、定時同期の設定をサポートしています。

### 11. ファイルのコピー&ペースト

指定したバケットまたはパス内で、コピーしたいファイル/フォルダを選択し、画面上のその他にあるコピーをク リックするか、コピーを右クリックすると、ファイル/フォルダのコピーが完了します。コピーに成功すると、そ の他のバケットまたはパスに貼り付けることができます。一括コピー&ペーストに対応しています。

#### 注意:

貼り付け先のパスに、コピーしたファイルまたはフォルダと同名のファイルがある場合、デフォルトで上書きされます。

12. ファイルのリネーム
リネームしたいファイルを選択し、右クリックで**リネーム**を選択するか、ファイル右側の**その他の操作**の**リネーム**をクリックし、ファイル名を入力して確定すると、ファイルのリネームが完了します。

#### 説明:

フォルダのリネーム操作は行えません。

#### 13. フォルダの新規作成

指定したバケットまたはパス内で、画面内の**フォルダの新規作成**をクリックするか、**フォルダの新規作成**を右ク リックし、フォルダ名を入力して確定すると、フォルダの新規作成が完了します。

#### 注意:

フォルダ名は**255**文字までに制限されており、数字、英語およびこれらの文字を組み合わせて使用できます。 フォルダ名に、 \\ / : \* ? " | < > などの特殊文字を含めることはできません。

...をフォルダ名にすることは許可されません。

フォルダのリネーム操作は行えませんので、慎重に命名してください。

#### 14.ファイル詳細の表示

ファイル名をクリックするか、メニューの**詳細**を右クリックすると、ファイルの詳細を表示できます。ファイルの 詳細情報には、ファイル名、ファイルサイズ、変更時刻、アクセス権限、ストレージタイプ、ETag、Headers、 指定ドメイン名、オブジェクトアドレス、一時リンクの作成などが含まれます。

Detail		×
Name	exampleobject.txt I	
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-125.cos.ap- chengdu.myqcloud.com/exampleobject.txt	
Create temporary object url	Create temporary object url	
	Close	)

#### 15. ファイルリンクの発行

COS内の各ファイルは、すべて特定のリンクを介してアクセスすることができます。ファイルにプライベート読 み取り権限がある場合は、一時的な署名をリクエストすることで、時間制限付きの一時的なアクセスリンクを発 行することができます。

ファイルリンクを発行するには、次のような方法があります。

テーブルビューで、ファイルの右側にある共有アイコンをクリックすると、ワンクリックでリンクが発行され、 コピーすることができます。ファイルの権限がパブリック読み取り権限の場合、リンクは署名なしで永続的に有効 になります。ファイルの権限がプライベート読み取り権限の場合、リンクは署名付きで2時間有効になります。

ファイルを選択し、**リンクのコピー**を右クリックすると、ワンクリックでリンクが発行され、コピーすることがで きます。ファイルの権限がパブリック読み取り権限の場合、リンクは署名なしで永続的に有効になります。ファイ ルの権限がプライベート読み取り権限の場合、リンクは署名付きで2時間有効になります。

ファイルの詳細で、**一時リンクの作成**をクリックすると、指定したドメイン名(CDNアクセラレーションドメイン名が有効になっている場合は設定可能など)の一時リンク、リンクタイプ、有効期間を設定することができます。

16. ファイル/フォルダの共有

操作バーの**共有**アイコンまたは右クリックメニューの**共有**オプションによって、COS内のいずれかのフォルダを 共有でき、共有の有効期間を設定することも可能です。

#### 説明:

単一フォルダの共有のみサポートされており、ファイルの一括共有はサポートされていません。

複数人でファイルを共有すると、ファイルのバージョンが混乱しやすくなります。バケットのバージョン管理機能 を有効にして、ファイルのバージョン履歴を遡れるようにすることをお勧めします。

Share Folders/Files ×		
Name	doc 1 folders/files.	
Permissions	O Read only C 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	

説明
共有フォルダのアクセス権限を設定することができます。 読み取り専用:アクセスリンクからフォルダリストをプルし、フォルダ内のファイルをダウン ロードすることができます。 読み書き アクセスリンクからフォルダリストのプル、フォルダからのファイルのダウンロー ド、共有フォルダへのファイルのアップロード、フォルダの新規作成ができます。
単位は分、時間または日です。 キーを使用してクライアントにログインする場合、ルートアカウントの有効期間の範囲は1分 ~2時間、サブアカウントの有効期間の範囲は1分~1.5日です。 Tencent Cloudアカウントを使用してクライアントにログインする場合、設定できる共有時間 はすべて最大2時間までとなります。デフォルト値は、現在のアカウントで許可されている最 大有効期間です。

抽出コー6桁で、デフォルトでシステムによって自動的に発行され、ユーザー定義が可能です。数字、ドアルファベット、記号の入力をサポートしています

#### 注意:

リンクの有効期間中、共有リンクと抽出コードを受け取ったユーザーは、そのリンクを経由してフォルダにアク セスすることができます。

#### 17. ファイルリンクのエクスポート

COSBrowserはファイルリンクのエクスポートをサポートしています。ツール画面右上のツールボックスのアイコ ンをクリックし、ツールボックスのポップアップウィンドウで**ファイルリンクのエクスポート**を選択します。その 後、ファイルのあるバケット、エクスポートしたいフォルダパス(例えばルートパス下のfolderフォルダをエクス ポートしたい場合は、 folder/ と入力します)および保存パスを入力し、最後に**エクスポート**をクリックすれ ば完了です。

#### 18. フォルダのプレビュー

COSBrowserはメディア系ファイルのプレビューをサポートしており、現在は画像、ビデオ、オーディオに対応し ています。メディア形式のファイルをダブルクリックするか、右クリックメニューの**プレビュー**または**再生**オプ ションをクリックすると、ファイルプレビュー画面を開くことができます。ファイルプレビューまたは再生画面で は、次を選択することができます。

**コピーリンク**:ファイルアクセスリンクを発行してコピーします。

**ダウンロード**:ファイルをローカルにダウンロードします。ローカルに同名ファイルが存在する場合、デフォルト で上書きされます。

スマホでの表示:プレビューインターフェースにファイルアクセスの2次元コードが発行されますので、スマホからこのコードを読み取れば、スマホから直接このファイルを表示することができます。

#### 注意:

プレビューはほとんどの画像形式をサポートしています。ビデオ形式はmp4、webmのみ、オーディオ形式はmp3、wavのみをサポートしています。

ファイルプレビューはダウンリンクのトラフィックを発生させるので、必要に応じて使用してください。





#### 19. ファイル検索

バケットの右上にある検索ボックスにファイル名を入力すると検索できます。COSBrowserは、ファイル名のプレ フィックス検索とファイルあいまい検索をサポートしています。

#### 20. バケットの検索

左側バケットリストの上にある検索ボックスにバケット名を入力すると、バケットをすばやく見つけることがで きます。

#### 21. バージョン履歴またはファイルフラグメントの表示

バケットのバージョン管理が有効になっている場合は、ファイルリストの上にある**表示>バージョン履歴**をクリッ クすると、ファイルのバージョン履歴を表示できます。バージョン履歴のリストでは、プレフィックス検索とすべ てのバージョン履歴のクリア(最新バージョンのみ保持)をサポートしています。

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

オブジェクトのアップロード中に、アップロードの一時停止やキャンセルを行うと、ファイルフラグメントが発 生する場合があります。ファイルリストの上にある**表示>ファイルフラグメント**をクリックすると、ファイルフラ グメントを表示できます。プレフィックス検索とすべてのファイルフラグメントのクリアに対応しています。

Clear Incomplete Multipart Uploads			Q Pre
Multipart Upload Name	Upload Task ID	Storag	Create time
	This list is en There is no incomple	mpty ete upload.	
	Cancel		

#### 22. ファイルの比較

ツール画面右上のツールボックスのアイコンをクリックし、ツールボックスのポップアップウィンドウで**ファイ** ルの比較を選択します。その後、ローカルフォルダと、比較したいバケットを選択し、指定するリージョン、バ ケット、ディレクトリを選択し、最後に**比較を開始**をクリックします。

#### 23. ビデオトランスコード

ツール画面右上のツールボックスのアイコンをクリックし、ツールボックスのポップアップウィンドウでビデオ トランスコードを選択します。その後、メディア処理サービスを有効化しているバケットを選択し、トランスコー ドタスクの作成をクリックします。トランスコードを行いたいメディアファイルとトランスコードテンプレートを 選択し、トランスコード後のファイル名とストレージのパスを入力します。リージョン、バケット、ディレクトリ を指定し、最後にトランスコードをクリックします。

#### 24. 権限承認コードの生成

権限承認コードの生成機能によって、指定のバケット、バケット下のリソースおよび操作に対し、一時的な権限を 承認することができます。フォルダ共有機能に比べてより柔軟に、カスタマイズした操作権限や特定のディレクト リに入る権限を付与することができます。一時的に使用可能なSecretId、SecretKey、Tokenおよび権限承認コード を生成することで、クライアントへの一時的なログインが可能になります。

操作に関する説明は次のとおりです。

ツール画面右上のツールボックスのアイコンをクリックし、ツールボックスのポップアップウィンドウで **権限承** 認コードの生成を選択します。その後、権限承認コードを生成したいバケットと権限を承認するリソースの範囲を 選択し、policy権限設定で権限を承認する操作(例えば読み取り書き込み権限など)を選択し、権限承認コードの 有効期間を設定し、最後に**OK**をクリックします。

#### 25. 画像処理

ツールの画像処理機能はズーム、トリミング、回転などの基本的な画像処理およびテキストウォーターマーク、画 像ウォーターマーク、処理後の画像リンクの生成をサポートしています。

画像処理機能を使用したいバケットを選択した後、ツール画面右上のツールボックスのアイコンをクリックし、 ツールボックスのポップアップウィンドウで**画像処理**を選択します。画像処理ポップアップウィンドウで、処理し たい画像ファイルを選択し、機能オプションを設定した後、最後に**画像のプレビュー**をクリックすれば、処理後 の画像リンクを生成できます。

システム 機能	説明	操作方法
ネット ワークプ ロキシの 設定	COSBrowserはデフォルトで、システムによって設定されたプロキ シを使用してネットワークへの接続を試みます。プロキシ設定が 正常であることを確認するか、インターネットに接続できないプ ロキシ設定を無効にしてください	高度な設定 > プロキシ を選択します。 ネットワークプロキシ を設定して、ネット ワークに接続します。
アップ ロード/ダ ウンロー ドファイ ル同時実 行数の設 定	COSBrowserは、アップロード、ダウンロードファイル同時実行数 の設定をサポートしています	高度な設定 > ダウン ロード/アップロードを 選択します。 一括転送の同時実行数 を設定します。
アップ	COSBrowserは、マルチパートアップロードとマルチパートダウン	高度な設定>ダウン

### ソフトウェアの設定

ロード/ダ ウンロー ドマルチ パート同 時実行数 の設定	ロードをサポートしています。一定サイズ以上のファイル転送を 行った場合、デフォルトでチャンクファイルによる転送になりま す	ロード/アップロードを 選択します。 マルチパート転送の同 時実行数を設定しま す。
アップ ロード/ダ ウンロー ドライ回 数の設定	COSBrowserは、ファイル転送時に失敗したタスクをデフォルトで 再試行します	高度な設定 > ダウン ロード/アップロードを 選択します。 転送失敗のリトライ回 数を設定します。
シングル スップ マード/ダ ウン度 制 限の設定	COSBrowserは、シングルスレッドのアップロード速度制限とシン グルスレッドのダウンロード速度制限の設定をサポートしていま す。総アップロード(またはダウンロード)速度制限 = シングル スレッドアップロード(またはダウンロード)速度制限×ファイル 同時実行数×マルチパート同時実行数	高度な設定 > アップ ロード(またはダウン ロード)を選択しま す。 シングルスレッドの アップロード(または ダウンロード)速度制 限を設定します。単位 はMB/sです。
アップ ロード時 の <b>2</b> 次 チェック の設定	COSBrowserは、アップロード後の2次チェックをサポートしてお り、オンラインファイルのサイズとステータスが正しいかどうか を確認します	高度な設定 > アップ ロードを選択します。 アップロード完了後の 2次チェックにチェッ クを入れます。
ローカル ログの表 示	COSBrowserはユーザーの操作を記録し、cosbrowser.logログの形 式でローカルに保存します	高度な設定 > 関連を選 択します。 ローカルログをクリッ クすると、システムに よってローカルログが 配置されているディレ クトリが開かれます。

# COSCLIツール COSCLI概要

最終更新日:::2024-06-26 10:27:35

COSCLIは、Tencent CloudのCloud Object Storage(COS)が提供するクライアントサイドのコマンドラインツール です。COSCLIツールを使用すれば、シンプルなコマンドラインの指示で、COS内のオブジェクト(Object)に対 し、一括アップロード、ダウンロード、削除などの操作を実行することができます。 COSCLIは、CobraフレームワークをベースにGoで記述されており、複数のバケットの設定やバケットをまたいだ 操作をサポートしています。 ./coscli [command] --help でCOSCLIの使用方法を確認することができま す。

### 機能リスト

設定ファイルの発行と変更 - config バケットの作成 - mb バケットの削除 - rb バケットまたはファイルリストの照会 - ls さまざまなタイプのファイルの統計情報の取得 - du ファイルのアップロード・ダウンロード・コピー - cp ファイルの同時アップロード・ダウンロード・コピー - sync ファイルの削除 - rm ファイルハッシュ値の取得 - hash マルチパートアップロード中に発生したフラグメントのリストアップ - lsparts フラグメントのクリーンアップ - abort アーカイブファイルの取得 - restore 署名済みURLの取得 - signurl

## ダウンロードとインストール設定

最終更新日:::2024-06-26 10:27:35

COSCLIツールは、Windows、macOS、およびLinuxオペレーティングシステムのバイナリパッケージを提供して いるため、簡単なインストールと設定後、使用可能です。

### 手順1:COSCLIツールをダウンロードする

操作シナリオに基づいてCOSCLIツールのダウンロードアドレスを選択してください。お使いのサーバーが中国に ある場合は、中国国内サイトのダウンロードアドレスを使用することを推奨します。

国内サイトのダウンロードアドレス	Githubダウンロードアドレス (海外サイトでの使用を 推奨)
Windows	Windows
macOS	macOS
macOS-m1	macOS-m1
Linux	Linux

また、国内サイトのmacOSとLinux環境におけるCOSCLIツールファイルをコマンドラインから取得することも可 能です。

macOS: wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-mac

macOS-m1: wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-mac-m1

Linux: wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-linux 説明:

Githubでは、現在のバージョン番号はv0.20.0-betaです。ツールの最新バージョン、過去のバージョンおとび更新 ログを取得しようとする場合は、releaseにアクセスして確認してください。

### 手順2:COSCLIツールをインストールする

#### Windows

1. ダウンロードされたWindowsバージョンのCOSCLIツールをC:\\Users\\<ユーザー名>ディレクトリに移動してください。

2. & coscli-windows.exe にリネームします coscli.exe 。

3. win+rキーを押して実行プログラムを開きます。



4.ダイアログボックスで、 cmd を入力して Enter を押すと、コマンドラインウィンドウが表示されます。
5.コマンドラインウィンドウで coscli --version を入力し、次の情報がプリントアウトされた場合、正常 にインストールされたことを示します。

説明:

シ Windows ステムにおいて、異なるコマンドラインクライアントでCOSCLIを使用する方式がやや異なる可能
 性があります。 coscli [command] を入力した後にCOSCLIが正しく動作できない場合は、 ./coscli
 [command] のフォーマットをお試しください。



coscli version v0.20.0-beta



#### macOS

1. 次のコマンドを実行し、リネームmacOSバージョンのCOSCLIファイルは下記のとおりです。



mv coscli-mac coscli

2. 次のコマンドを実行してファイルの実行権限を修正します。





chmod 755 coscli

3. コマンドラインに ./coscli --version を入力すると、次の情報がプリントアウトされた場合、正常にインストールされたことを示します。





coscli version v0.20.0-beta

#### 説明:

macOSシステムにおいてCOSCLIを使用する場合、「開発者が確認できないため、coscliが開けられない」という プロンプトが表示された場合、設定〉安全性とプライバシー〉通用で「coscliを引き続き開く」を選択した後、 COSCLIが正常に使用可能になります。

#### Linux

1. 次のコマンドを実行し、リネームLinuxバージョンのCOSCLIファイルは下記のとおりです。





mv coscli-linux coscli

2. 次のコマンドを実行してファイルの実行権限を修正します。





chmod 755 coscli

3. コマンドラインウィンドウで ./coscli --version を入力すると、次の情報がプリントアウトされた場 合、正常にインストールされたことを示します。





coscli version v0.20.0-beta

### 手順3:COSCLIツールの構成

#### 注意事項:

一時的な許可の使用方法によってツール使用の安全性がさらに向上できるため、ユーザーは一時的キーでツール を使用することを推奨します。一時的キーを申請するときは、ターゲットストレージバケットやオブジェクト以外 のリソースの漏洩を防止するために、 最小限の権限ガイドライン原則 に準拠してください。

永久キーを使用する必要がある場合は、 最小限の権限ガイドライン原則 に準拠して永久キーの権限範囲を制限す ることを推奨します。

ユーザーはCOSCLIを初めて使用する場合、設定ファイルを初期化する必要があります。設定ファイルには次の2 つの部分が含まれています。

1. COSCLIにユーザーのTencent Cloudアカウントへのアクセスを許可するには、ユーザーはキーID、キーKey、一時的キーTokenを構成する必要があります。

2. 通常バケットにエイリアスを追加するには、通常バケット名、バケットのリージョン情報、およびバケットエ イリアスを構成する必要があります。通常バケットの情報を構成した後、ユーザーはバケット名やリージョン情報 を入力せずに、エイリアスを使用してバケット操作を行うことが可能です。複数の通常バケット構成を追加した 後、クロスバケットまたはクロスドメインの操作をより便利に行うことが可能です。通常バケット情報を構成する 必要がない場合は、 Enter をクリックして次へ進みます。

初めて使用する場合、COSCLIが自動的に ./coscli config init を呼び出して ~/.cos.yaml で設定 ファイルを生成するが、ユーザーはコマンドラインでインタラクティブに構成を完了することが可能です。その 後、ユーザーは ./coscli config init コマンドを使用して他の位置でCOSCLIのために設定ファイルをイ ンタラクティブに生成することも可能です。ユーザーは ./coscli config show を使用して設定ファイルの 場所と構成パラメーター情報を確認することが可能です。

設定ファイルの中の各設定項目の説明は次のとおりです。

設定項目の	説明
Secret ID	キーIDであり、使用上のリスクを軽減するために、サブアカウントキーを使用し、最小限 の権限ガイドラインに準拠するよう許可することを推奨します。サブアカウントキーを取 得するには、 サブアカウントのアクセス キーの管理 を参照してください。
Secret Key	キーKeyであり、使用上のリスクを軽減するために、サブアカウントキーを使用し、最小限 の権限ガイドラインに準拠するよう許可することを推奨します。サブアカウントキーを取 得するには、サブアカウントのアクセス キーの管理 を参照してください。
Session Token	一時的キーtokenであり、一時キーを使用するときに設定する必要があるが、使用しない場合は、直接 Enter を押して次へ進みます。一時的キーの詳細については、一時的キーを使用してCOS にアクセスする を参照してください。
Mode	はIDモードを設定し、列挙値の SecretKey と CvmRole をサポートします。空にする ことが可能です。空である場合、値がデフォルトでは SecretKey になり、キーを使用 してCOSをリクエストすることを示します。Modeが CvmRole になる場合、インスタン スロールの管理を使用してCOSをリクエストすることを示します。
Cvm Role Name	CVMロールのインスタンス名を設定します。 詳細については、 インスタンスロールの管理 を参照してください。
protocol	ネットワーク伝送プロトコルであり、デフォルトではhttpsです。Httpに変更する必要があ る場合は、設定プロファイルに直接入って修正しても可能です。

APPID	APPIDは、ユーザーがTencent Cloudアカウントを申請した後に得られたアカウントであ り、システムに自動的に割り当てられ、アカウント情報から取得できます。ストレージバ ケットのフルネームは Bucket Name と APPID という2つの要素からなり、フォー マットが <bucketname-appid> です。詳細については、ストレージバケットの命名 ルールを参照してください。</bucketname-appid>
Bucket Name	ストレージバケット名であり、APPIDとともにストレージバケットのフルネームを構成し ます。フォーマットが <bucketname-appid> です。詳細については、 ストレージバ ケットの命名ルール を参照してください。</bucketname-appid>
Bucket Endpoint	ストレージバケットが存在するリージョンのドメイン名であり、ドメイン名のフォーマッ トがデフォルトでは cos. <region>.myqcloud.com です。その中、 <region> がス トレージバケットのリージョンを表し、たとえば、ap-guangzhou、ap-beijingなどです。 COSがサポートするリージョンリストは リージョンとアクセスドメイン名 を参照してくだ さい。 ストレージバケットではアクセラレーションが有効になっている場合は、グローバルアク セラレーションドメイン名を構成することことが可能です。たとえば、グローバルアクセ ラレーションドメイン名を cos.accelerate.myqcloud.com に構成し、イントラネッ トグローバルアクセラレーションドメイン名を cos- internal.accelerate.tencentcos.cn に構成します。</region></region>
Bucket Alias	ストレージバケットのエイリアスであり、入力する必要があるコマンドの長さを減らすた めに、設定後に、使用時に BucketName-APPID の代わりに BucketAlias を使用す ることが可能です。この項目を構成しない場合は、 BucketAlias の値が BucketName-APPID の値になります。
OFS Bucket	メタデータアクセラレーションバケットのマークであり、該当のバケットではメタデータ アクセラレーション機能が有効になっているかどうかを識別するために使用されます。現 在、メタデータアクセラレーション機能はパブリックベータ版であり、使用しようとする 場合は、お問い合わせをクリックして申請してください。パブリックベータ期間中に料金 が発生しません。ユーザーのサイトのメッセージまたは費用計算概要から最新の費用発 生状況を把握することが可能です。

初回構成時には、COSCLIは1つのストレージバケットの構成のみを要求するが、複数のストレージバケットを構成しようとする場合は、後で ./coscli config add コマンドを使用してストレージバケットの構成を追加することが可能です。設定ファイルを変更する必要がある場合、または設定ファイルに関する操作をさらに取得しようとする場合、config コマンドを参照するか、または ./coscli config --help コマンドを使用して設定ファイルに関連するコマンドを迅速に確認してください。

コマンドを正式に使用し開始する前に、 ./coscli --help コマンドを使用してCOSCLIの使用方法を迅速に 確認することが可能です。

#### その他の構成方法

./coscli config init を使用して設定ファイルをインタラクティブに生成するほか、COSCLIの設定ファ イルを直接手動で作成することも可能です。COSCLIの設定ファイルのフォーマットは yaml です。設定ファイ ルのフォーマットの例は次のとおりです。



```
name: examplebucket1-1250000000
alias: bucket1
region: ap-shanghai
endpoint: cos.ap-shanghai.myqcloud.com
ofs: false
name: examplebucket2-1250000000
alias: bucket2
region: ap-guangzhou
endpoint: cos.ap-guangzhou.myqcloud.com
ofs: false
name: examplebucket3-1250000000
alias: bucket3
region: ap-chengdu
endpoint: cos.ap-chengdu.myqcloud.com
ofs: False
```

#### 注意事項:

COSCLI は、デフォルトでは~/.cos.yaml から構成項目を読み取ります。ユーザーがカスタム設定ファイルを使用 しようとする場合は、コマンドの後に -c (--config-path)オプションを使用してください。設定ファイルに格納さ れている secretid/secretkey/sessiont secretid/secretkey/sessiontoken はすべて暗号化された文字列です。

# よく使用するコマンド 設定ファイルの発行と変更 - config

最終更新日:::2024-06-26 10:27:35

### コマンド形式

以下のconfigコマンドは、設定ファイルを発行・変更するときに使います。



```
./coscli config [command] [flag]
```

#### 説明:

各設定項目を正しく入力すると、 ./coscli config show を使用して設定情報を確認することができます。 configコマンドには、以下のサブコマンドが含まれます。

command名	commandの用途
add	新しいバケット設定を追加します。
delete	既存のバケット設定を削除します。
init	設定ファイルをインタラクティブに作成します。
set	設定ファイルのbaseグループの1つまたは複数の設定項目を変更します。
show	指定された設定ファイルの情報を印刷します。

#### configとそのサブコマンドには、以下のオプションflagが含まれます。

flagの略称	flagの正式名称	flagの用途
-h	help	ヘルプ情報を出力します。
-C	config-path	使用する設定ファイルパスを指定します。

#### config addサブコマンドには、以下のオプションflagが含まれます。

flagの略称	flagの正式名称	flagの用途
-h	help	ヘルプ情報を出力します。
-a	alias	バケットエイリアス。
-b	bucket	バケット名。
-r	region	バケットリージョン。
-0	ofs	メタデータ加速バケトタグ。詳細については メタデータ加速概要 を参照してください。

#### config deleteサブコマンドには、以下のオプションflagが含まれます。

flagの略称	flagの正式名称	flagの用途
-h	help	ヘルプ情報を出力します。

-a	alias	バケットエイリアス。

config setサブコマンドには、以下のオプションflagが含まれます。

flagの略称	flagの正式名称	flagの用途
-h	help	ヘルプ情報を出力します。
なし	secret_id	<b>secret ID</b> は、コンソールへのアクセスから作成 して取得することができます。
なし	secret_key	<b>secret key</b> は、コンソールへのアクセス から作 成して取得することができます。
-t	session_token	一時的なキー token の設定。一時的なキーの詳 細については、一時的なキーで COS にアクセ スするを参照してください。
なし	mode	身分設定モードは、列挙値 SecretKey と CvmRole をサポートします。ヌルにすること もでき、ヌル値のデフォルトは SecretKey で、キーを使用して COS をリクエストするこ とを示します。mode が CvmRole の場合、イ ンスタンスロールの管理で COS をリクエスト することを示します。
なし	cvm_role_name	<b>CVM</b> ロールのインスタンス名を設定します。 詳細については、インスタンスロールの管理 を参照してください。
なし	close_auto_switch_host	バックアップドメイン名の自動切り替えを無 効にするかどうかを設定します。値は true ま たは false から選択でき、ヌルにすることもで きます。 設定しない場合、または値が False の場合は、 バックアップドメイン名の切り替えが実行さ れます。 true に設定すると、バックアップドメイン名 の切り替えは実行されません。

### 操作事例

新しいバケット設定の追加





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

#### 既存のバケット設定の削除





./coscli config delete -a bucket3

#### デフォルトの設定ファイルのsession-tokenの変更





./coscli config set -t test-token123

#### 指定された設定ファイルの情報の印刷





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

デフォルト設定ファイルの mode と cvmrolename を変更します





./coscli config set --mode CvmRole --cvm\_role\_name testName

# バケットの作成 - mb

最終更新日:::2024-06-26 10:27:35

mbコマンドは、バケットを作成するときに使います。

### コマンド形式



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

#### mbコマンドには、以下のオプションflagが含まれます。

flagの略称	flagの正式名称	flagの用途
-h	help	ヘルプ情報を出力
-C	config-path	使用する設定ファイルパスを指定
-r	region	バケットリージョン

#### 注意:

COSCLIにおいてmbコマンドで作成したバケットを操作する場合は、mbコマンドが成功した後、config addコマンドで設定ファイル内のバケット設定を更新する必要があります。

操作事例





// バケットbucket3の作成
./coscli mb cos://bucket3-125000000 -r ap-chengdu
// 設定ファイルの更新
./coscli config add -b bucket3-1250000000 -r ap-chengdu -a bucket3
// cos://bucket3を更新すると、このバケットにアクセスできるようになります

# バケットの削除 - rb

最終更新日:::2024-06-26 10:27:35

rbコマンドは、バケットを削除するときに使います。

### コマンド形式



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

#### rbコマンドには、以下のオプションflagが含まれます。

flagの略称	flagの正式名称	flagの用途
-h	help	ヘルプ情報を出力
-C	config-path	使用する設定ファイルパスを指定
-r	region	バケットリージョン

操作事例





// bucket3の削除
./coscli rb cos://bucket3-1250000000 -r ap-chengdu
// 設定ファイルの更新
./coscli config delete -a bucket3
## バケットまたはファイルリストの照会 - Is

最終更新日:::2024-06-26 10:27:35

lsコマンドは、すべてのバケットリスト、バケット内のファイルリストおよびフォルダ内のファイルリストを照会 するために使います。

コマンド形式





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

説明:

bucketAliasについては、ダウンロードとインストール設定をご参照ください。

このコマンドのその他の共通オプション(例えばバケットの切り替え、ユーザーアカウントの切り替えなど)に 関しては、共通オプションのドキュメントをご参照ください。

lsコマンドには、次のオプションのパラメータが含まれます。

パラメータ形式	パラメータ用途	事例

cos://bucketAlias	バケットの指定	cos://bucket1
/prefix/	いずれかのフォルダを指定します	/picture/

操作事例

### 現在のアカウント下のすべてのバケットを一覧表示します



./coscli ls

### ファイルの一覧表示

bucket1にあるすべてのファイルを一覧表示します



./coscli ls cos://bucket1

bucket1内のpictureフォルダにあるすべてのファイルとフォルダを一覧表示します





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

## さまざまなファイルの統計情報を取得-du

最終更新日:::2024-06-26 10:27:35

duコマンドは、バケットまたはフォルダ内の各バケットタイプのファイルの統計情報を一覧表示するために使い ます。この統計情報には、異なるストレージタイプのファイルの総数や各タイプのファイルの合計サイズが含まれ ます。

コマンド形式





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

説明:

bucketAliasについては、設定をご参照ください。

lsコマンドには、次のオプションのパラメータが含まれます。

パラメータ形式	パラメータ用途	事例
/prefix/	いずれかのフォルダを指定します	cos://bucket1/picture/

### duコマンドには、以下のオプションflagが含まれます。

flagの略称	flagの正式名称	flagの用途
-h	help	ヘルプ情報を出力
-C	config-path	使用する設定ファイルパスを指定
なし	include	特定のモードを含むファイル
なし	exclude	特定のモードを除外したファイル

#### 説明:

--include は標準的な正規表現の構文をサポートしており、これを使えば特定の条件を満たすファイルをフィ ルタリングすることができます。

zshを使用する際、pattern文字列の両端に二重引用符を付ける必要がある場合があります。





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

## 操作事例

bucket1にあるファイルの統計情報を一覧表示します





./coscli du cos://bucket1

bucket1のpictrueフォルダにあるファイルの統計情報を一覧表示します





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

bucket1のpictrueフォルダにあるすべての.mp4タイプのファイルの統計情報を一覧表示します





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

bucket1のpictrueフォルダにあるすべての.mdタイプのファイルの統計情報を一覧表示します





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

# ファイルのアップロード・ダウンロードまた はコピー - cp

最終更新日:::2024-06-26 10:27:35

cpコマンドは、ファイルをアップロード、ダウンロードまたはコピーするときに使います。

## コマンド形式





./coscli cp <source\_path> <destination\_path> [flags]

説明:

bucketAliasについては、ダウンロードとインストール設定をご参照ください。

このコマンドのその他の共通オプション(例えばバケットの切り替え、ユーザーアカウントの切り替えなど)に 関しては、共通オプションのドキュメントをご参照ください。

cpコマンドには、以下のオプションflagが含まれます。

flagの略 称	flagの正式名 称	flagの用途

なし	include	特定のモードを含むファイル
なし	exclude	特定のモードを除外したファイル
-r	recursive	フォルダ内のすべてのファイルを再帰的にトラバーサル処理するかどうか
なし	storage- class	アップロードするファイルのタイプを指定(デフォルトは <b>STANDARD</b> )
なし	part-size	ファイルチャンクサイズ(デフォルトは32MB)、単位:MB
なし	thread-num	同時実行スレッド数(デフォルトの同時実行は5)
なし	rate- limiting	シングルリンクレート制限(0.1~100MB/s)、単位:MB/s
なし	meta	アップロードファイルのメタ情報。一部のHTTP標準属性(HTTP Header) お よび x-cos-meta- で始まるユーザーカスタムメタデータ(User Meta) が含 まれます。ファイルメタ情報の形式は header:value#header:value であ り、例えば Expires:2022-10-12T00:00:00.000Z#Cache-Control:no- cache#Content-Encoding:gzip#x-cos-meta-x:x のようになります。

#### 説明:

cpコマンドは、大容量ファイルのアップロードやダウンロードを行う場合、アップロード/ダウンロードの同時実 行を自動的に有効にします。

ファイルが --part-size よりも大きい場合、COSCLIはまず --part-size に従ってファイルをチャンクに し、次に --thread-num 個のスレッドを使用してアップロード/ダウンロードのタスクを同時に実行します。

各スレッドは1つのリンクを維持します。各リンクに対して --rate-limiting パラメータを使用すると、シン グルリンクのレート制限ができます。同時アップロード/ダウンロードが有効な場合、合計レートは、 --

thread-num \* --rate-limiting となります。

ファイルをチャンクでアップロード/ダウンロードする場合、デフォルトで中断からの再開が有効になります。

--include と --exclude は標準的な正規表現の構文をサポートしており、これを使えば特定の条件を満た すファイルをフィルタリングすることができます。

zshを使用する場合、pattern文字列の両端に二重引用符を付ける必要がある場合があります。





./coscli cp ~/test/ cos://bucket1/example/ -r --include ".\*.txt" --meta=x-cos-meta-

操作事例

### アップロード操作

単一ファイルのアップロード





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

ローカルのtestフォルダにあるすべてのファイルをbucket1内のexampleフォルダにアップロードします





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

ローカルのtestフォルダにあるすべての.mp4タイプのファイルをbucket1内のexampleフォルダにアップロード します





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

ローカルのtestフォルダにあるすべての.mdタイプではないファイルをbucket1内のexampleフォルダにアップ ロードします





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

ローカルのdirフォルダにはdirA、dirB、dirC、dirDという4つのフォルダがあり、dirフォルダにあるdirDフォル ダを除くすべての内容をアップロードします





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

ローカルのtestフォルダにあるすべてのファイルをbucket1内のexampleフォルダにアップロードし、アーカイブ タイプのファイルとして保存します





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

ローカルのfile.txtファイルをバケットbucket1にアップロードし、シングルリンクレート制限を1.3MB/sに設定します





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

## ダウンロード操作

単一ファイルのダウンロード





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

bucket1のexampleフォルダにあるすべてのファイルをローカルのtestフォルダにダウンロードします





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

bucket1のexampleフォルダにあるすべての.mp4タイプのファイルをローカルのtestフォルダにダウンロードします





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

bucket1のexampleフォルダにあるすべての.mdタイプではないファイルをローカルのtestフォルダにダウンロー ドします





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

コピー操作

バケット内の単一ファイルのコピー





./coscli cp cos://bucket1/example.txt cos://bucket1/example\_copy.txt

バケット間の単一ファイルのコピー





./coscli cp cos://bucket1/example.txt cos://bucket2/example\_copy.txt

bucket1のexample1フォルダにあるすべてのファイルをbucket2のexample2フォルダにコピーします





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

bucket1のexample1フォルダにあるすべての.mp4タイプのファイルをbucket2のexample2フォルダにコピーします





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

bucket1のexample1フォルダにあるすべての.mdタイプではないファイルをbucket2のexample2フォルダにコ ピーします





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

# ファイルの同時アップロード・ダウンロード またはコピー - sync

最終更新日:::2024-07-04 16:48:52

## コマンド形式

syncコマンドは、ファイルのアップロード・ダウンロード・コピーを同期させるときに使います。cpコマンドと 異なる点は、syncコマンドはまず同名ファイルのcrc64を比較し、crc64の値が同じ場合は転送を行わない点です。





./coscli sync <source\_path> <destination\_path> [flag]

説明:

**bucketAlias**については、設定をご参照ください。 **sync**コマンドには、以下のオプションflagが含まれます。

flagの略 称	flagの正式名称	flagの用途
-h	help	ヘルプ情報を出力

-C	config-path	使用する設定ファイルパスを指定
なし	include	特定のモードを含むファイル
なし	exclude	特定のモードを除外したファイル
-r	recursive	フォルダ内のすべてのファイルを再帰的にトラバーサル処理するかどうか
なし	storage-class	アップロードするファイルのタイプを指定(デフォルトは <b>STANDARD</b> )
なし	part-size	ファイルチャンクサイズ(デフォルトは <b>32MB</b> )
なし	thread-num	同時実行スレッド数(デフォルトの同時実行は5)
なし	rate-limiting	シングルリンクレート制限(0.1~100MB/s)
なし	routines	파일 간 동시 업로드 또는 다운로드 스레드의 파일 수를 지정하며, 기본값은 3 입니다.
なし	fail-output	이 옵션은 업로드 또는 다운로드가 실패할 때 파일의 오류 출력을 활성화할지 여부를 결정합니다(기본값은 true, 활성화). 활성화되면 실패한 파일 전송은 지정된 목록(지정하지 않은 경우 기본값은 ./coscli_output 임)에 기록됩 니다. 비활성화되면 실패한 파일의 수만 콘솔로 출력됩니다.
なし	fail-output- path	이 옵션은 업로드 또는 다운로드 실패를 기록하는 파일의 오류 출력 폴더를 지 정하는 데 사용됩니다. 사용자 정의 폴더 경로를 제공함으로써 오류 출력 폴더 의 위치 및 이름을 제어할 수 있다. 이 옵션을 설정하지 않은 경우 기본 오류 로 그 폴더 ./coscli_output 를 사용합니다.
なし	retry-num	빈도 제한 재시도 횟수(기본값은 0, 재시도하지 않음). 1-10 회 선택 가능. 여러 대의 컴퓨터가 동시에 동일한 COS 목록에서 다운로드 작업을 실행하는 경우, 이 매개변수를 지정하여 재시도함으로써 빈도 제한 오류를 피할 수 있습 니다.
なし	err-retry-num	오류 재시도 횟수(기본값은 0 임). 1-10 회로 지정하거나 0 으로 설정하 여 재시도하지 않음을 나타냅니다.
なし	err-retry- interval	재시도 간격(err-retry-num 를 1-10 로 지정할 때에만 사용 가능). 1-10 초의 재시도 간격을 지정하며, 지정하지 않거나 0 으로 설정한 경우 매번 재시도 간격은 1-10 초 내에서 무작위로 선택됩니다.
なし	only-current- dir	현재 목록의 파일만 업로드하고 하위 목록 및 그 내용을 무시할지 여부(기본값 은 false , 무시하지 않음).
なし	disable-all- symlink	업로드 시 모든 심볼릭 링크의 하위 파일 및 심볼릭 링크의 하위 목록을 무시할 지 여부(기본값은 true , 업로드하지 않음).
なし	enable-	심볼링 링크의 하위 목록 업로드할지 여부(기본값은 false, 업로드하지 않


	symlink-dir	<u>으</u> ).
なし	disable-crc64	CRC64 데이터 검증 비활성화 여부 (기본값은 false , 검증 활성화).
なし	delete	지정된 타겟 경로에 있는 임의의 기타 파일을 삭제하고 이번 동기화된 파일만 유지(기본값은 false, 삭제하지 않음) 실수로 데이터가 삭제되는 것을 방지 하기 위해delete 옵션을 사용하기 전에 버전 제어를 활성화하는 것이 좋습니 다.
なし	backup-dir	타겟측에서 이미 삭제되었지만 소스측에 존재하지 않는 파일을 저장하기 위해 삭제된 파일의 백업을 동기화합니다(다운로드할 때에만 유효하고 delete=true 시 필히 지정). 업로드 및 버킷 복제는 버전 제어를 사용하여 실 수로 삭제되는 데이터를 복구하시기 바랍니다.
なし	force	확인 알림이 없는 강제 작업 (기본값은 false ).

説明:

syncコマンドは、大容量ファイルのアップロードやダウンロードを行う場合、アップロード/ダウンロードの同時 実行を自動的に有効にします。

ファイルが --part-size よりも大きい場合、COSCLIはまず --part-size に従ってファイルをチャンクに し、次に --thread-num 個のスレッドを使用してアップロード/ダウンロードのタスクを同時に実行します。

各スレッドは1つのリンクを維持します。各リンクに対して --rate-limiting パラメータを使用すると、シン グルリンクのレート制限ができます。同時アップロード/ダウンロードが有効な場合、合計レートは、 --

thread-num \* --rate-limiting となります。

ファイルをチャンクでアップロード/ダウンロードする場合、デフォルトで中断からの再開が有効になります。

--include と --exclude は標準的な正規表現の構文をサポートしており、これを使えば特定の条件を満た すファイルをフィルタリングすることができます。

zshを使用する場合、pattern文字列の両端に二重引用符を付ける必要がある場合があります。

操作事例

ファイルの同時アップロード





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

## ファイルの同時ダウンロード





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

## バケット内でのファイル同期コピー





./coscli sync cos://bucket1/example.txt cos://bucket1/example\_copy.txt

## バケット間でのファイル同期コピー





./coscli sync cos://bucket1/example.txt cos://bucket2/example\_copy.txt

# ファイルの削除 - rm

最終更新日:::2024-07-04 16:52:35

rmコマンドは、ファイルを削除するときに使います。

# コマンド形式



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

### 説明:

bucketAliasについては、設定をご参照ください。

rmコマンドには、以下のオプションflagが含まれます。

flagの略称	flagの正式名称	flagの用途	
-h	help	ヘルプ情報を出力	
-C	config-path	使用する設定ファイルパスを指定	
なし	include	特定のモードを含むファイル	
なし	exclude	特定のモードを除外したファイル	
-r	recursive	フォルダ内のすべてのファイルを再帰的にトラバーサル処理するかどうか	
-f	force	強制削除(ファイルを削除するまで確認情報はポップアップ表示されませ ん)	
なし	fail-output	このオプションは、アップロードまたはダウンロードが失敗した時にファ イルのエラー出力を有効にするかどうかを決定します(デフォルト は true で、有効にする)。有効にすると、ファイル転送の失敗が指定し たディレクトリに記憶されます(指定しない場合はデフォルト で ./coscli_output になる)。無効にすると、エラーファイルの数だ けがコンソールに出力されます。	
なし	fail-output- path	このオプションは、アップロードまたはダウンロードに失敗したファイル のエラー出力を記憶するフォルダを指定するために使用されます。カスタ ムフォルダパスを指定することで、エラー出力フォルダの場所と名前を管 理できます。このオプションが設定されていない場合は、デフォルトのエ ラーログフォルダ ./coscli_output が使用されます。	

#### 説明:

--include と --exclude は標準的な正規表現の構文をサポートしており、これを使えば特定の条件を満た すファイルをフィルタリングすることができます。

zshを使用する場合、pattern文字列の両端に二重引用符を付ける必要がある場合があります。





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

# 操作事例

ファイルの削除





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

pictrueフォルダからすべてのファイルを削除





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

# ファイルハッシュ値の取得 - hash

最終更新日:::2024-06-26 10:27:35

hashコマンドは、ローカルファイルのハッシュ値を計算したり、Cloud Object Storage(COS)ファイルのハッシュ 値を取得したりするために使います。

コマンド形式





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

## 説明:

bucketAliasについては、設定をご参照ください。 hashコマンドには、以下のオプションflagが含まれます。

flagの略称	flagの正式名称	flagの用途
-h	help	ヘルプ情報を出力
-C	config-path	使用する設定ファイルパスを指定



なしtype ハッシュタイプ(md5またはcrc64で、デフォルトは	tcrc64)
------------------------------------	---------

操作事例

## ローカルファイルのcrc64を計算します



./coscli hash ~/test.txt

# COSファイルのmd5を取得します



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

# マルチパートアップロード中に発生したフラ グメントを一覧表示 - Isparts

最終更新日:::2024-06-26 10:27:35

lspartsコマンドは、マルチパートアップロード中に発生したフラグメントを一覧表示します。

# コマンド形式





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

## 説明:

bucketAliasについては、設定をご参照ください。

lspartsコマンドには、以下のオプションflagが含まれます。

flagの略称	flagの正式名称	flagの用途
-h	help	ヘルプ情報を出力
-C	config-path	使用する設定ファイルパスを指定



なし	include	特定のモードを含むファイル
なし	exclude	特定のモードを除外したファイル
なし	limit	一覧表示する最大数(0~1000)を指定

操作事例

bucket1にあるすべてのファイルフラグメントを一覧表示





./coscli lsparts cos://bucket1

# bucket1のpictrueフォルダにあるすべてのフラグメントを一覧表示





./coscli lsparts cos://bucket1/pictrue/

# フラグメントのクリーンアップ - abort

最終更新日:::2024-06-26 10:27:35

abortコマンドは、マルチパートアップロード中に発生したファイルフラグメントをクリーンアップするときに使います。

コマンド形式





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

## 説明:

bucketAliasについては、設定をご参照ください。 abortコマンドには、以下のオプションflagが含まれます。

<b>flag</b> の略称	flagの正式名称	flagの用途
-h	help	ヘルプ情報を出力
-C	config-path	使用する設定ファイルパスを指定



なし	include	特定のモードを含むファイル
なし	exclude	特定のモードを除外したファイル

操作事例

bucket1にあるすべてのファイルフラグメントをクリア



./coscli abort cos://bucket1

# bucket1のpictrueフォルダにあるすべてのフラグメントをクリア



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

# アーカイブファイルの取得 - restore

最終更新日:::2024-06-26 10:27:35

restoreコマンドは、アーカイブファイルを取得するときに使います。

コマンド形式



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

#### 説明:

bucketAliasについては、設定をご参照ください。

restoreコマンドには、以下のオプションflagが含まれます。

flagの略称	flagの正式名称	flagの用途
-h	help	ヘルプ情報を出力
-C	config-path	使用する設定ファイルパスを指定
なし	include	特定のモードを含むファイル
なし	exclude	特定のモードを除外したファイル
-d	days	一時ファイルの有効期限を指定(デフォルトは3日間)
-m	mode	リカバリモードを指定(デフォルトはStandard)
-r	recursive	フォルダの再帰的なトラバーサル処理

#### 説明:

--include と --exclude は標準的な正規表現の構文をサポートしており、これを使えば特定の条件を満た すファイルをフィルタリングすることができます。

zshを使用する場合、pattern文字列の両端に二重引用符を付ける必要がある場合があります。





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

操作事例

標準モードでbucket1のアーカイブファイルを取得します





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

bucket1内のpictureフォルダにあるすべてのアーカイブファイルを超高速モードで取得します





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

#### 説明:

このコマンドを実行する前に、フォルダ内のすべてのファイルが同じタイプ(例:ARCHIVEタイプ)であること を確認する必要があります。異なるタイプのファイルがある場合は、 --include または --exclude を使用 して、同じタイプのファイルをフィルタリングで除外してください。

# よくある質問

最終更新日:::2024-06-26 10:27:35

### COSCLIツールとCOSCMDツールの違いとは何ですか。

1. COSCLIツールはgolangでビルドされており、コンパイルされたバイナリーパッケージを直接配布します。ユー ザーはインストールやデプロイの際に依存関係をインストールする必要がなく、開梱したらすぐに使用可能です。 COSCMDツールはPythonでビルドされており、ユーザーはインストールする前に、Python環境と依存パッケージ をインストールする必要があります。

2. COSCLIツールはバケットエイリアスの設定に対応しており、 <BucketName-APPID> の代わりに短い文字列 を使用できるため、ユーザーにとって使い勝手がよいです。COSCMDツールはバケットエイリアスの設定に対応 していないため、ユーザーは <BucketName-APPID> と入力してバケットを指定し、複雑で読みにくいコマン ドに対応しなければなりません。

3. COSCLIツールは設定ファイル内に複数のバケットの設定やバケット間操作にも対応しています。COSCMD ツールは設定ファイルに1つのバケットしか設定できず、バケット間の操作コマンドは冗長になりすぎます。

# COSCMDツール

最終更新日:::2024-06-26 10:27:35

# 機能説明

ユーザーはCOSCMDツールを使用すれば、簡単なコマンドラインによって、オブジェクト(Object)の一括アップ ロード・ダウンロード・削除などの操作が行えます。

# 使用環境

#### システム環境

Windows、Linux、macOSシステムをサポートします。

#### 説明:

ローカル文字形式がUTF-8であることを確認してください。UTF-8でない場合、中国語版のファイルを操作する と、異常が発生します。

本機の時刻が協定世界時で修正されていることを確認してください。誤差が大きすぎると、正常に使用できなく なります。

#### ソフトウェア依存

Python 2.7/3.5/3.6/3.9 $_{\circ}$ 

最新バージョンのpip。

#### 説明:

pipが統合された最新版のPython(例:バージョン3.9.0)を直接インストールすることをお勧めします。

#### インストールおよび設定

環境のインストールと設定操作の詳細については、Pythonのインストールと設定をご参照ください。 pip環境のインストールと設定操作の詳細については、公式サイトpipのインストールの説明をご参照ください。

#### ダウンロードとインストール

ユーザーがCOSCMDをインストールする方法として、以下の3つの方法が提供されています。

#### 1.1 pipによるインストール

pip コマンドを実行してインストールします。





pip install coscmd

インストールの成功後、ユーザーは -v または --version コマンドを使用して、現在のバージョン情報を確認することができます。

#### 注意:

Windowsでインストールした場合は、環境変数

に C:\\python\_install\_dir; と C:\\python\_install\_dir\\Scripts という2つのパスを追加する必 要があります。



#### 1.2 pipの更新

インストールが完了したら、次のコマンドを実行して更新します。



pip install coscmd -U

#### 2. ソースコードのインストール(非推奨)

ソースコードダウンロードアドレス:ここをクリック。





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

## 注意:

Pythonバージョンが2.6で、pipの依存ライブラリへのインストールが失敗しやすい場合は、この方法でインストールすることをお勧めします。

```
3. オフラインインストール
```

注意:



2台のマシンのPythonのバージョンが同じであることを確認してください。同じでない場合、インストールは失敗 してしまいます。



# パブリックネットワークを備えたマシンで次のコマンドを実行します
mkdir coscmd-packages
pip download coscmd -d coscmd-packages
tar -czvf coscmd-packages.tar.gz coscmd-packages

# インストールパッケージをパブリックネットワークのないマシンにコピーしてから、次のコマンドを実行しま

```
tar -xzvf coscmd-packages.tar.gz
pip install coscmd --no-index -f coscmd-packages
```

パラメータの設定

### helpオプションの確認

ユーザーは、 -h または --help コマンドを使用して、ツールのhelp情報や使用方法を確認することができます。





**Cloud Object Storage** 

coscmd -h

help情報は次のとおりです。



```
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,re
        ...
        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
   config
                        Config your information at first
                        Upload file or directory to COS
   upload
                        Download file from COS to local
    download
                        Delete file or files on COS
    delete
                        Aborts upload parts on COS
    abort
                        Copy file from COS to COS
   сору
                        move file from COS to COS
   move
    list
                        List files on COS
                       List upload parts
   listparts
    info
                        Get the information of file on COS
    restore
                        Restore
    signurl
                        Get download url
    createbucket
                        Create bucket
    deletebucket
                       Delete bucket
   putobjectacl
                       Set object acl
                        Get object acl
    getobjectacl
    putbucketacl
                        Set bucket acl
                        Get bucket acl
    getbucketacl
    putbucketversioning
                        Set the versioning state
    getbucketversioning
                        Get the versioning state
                        Connection test
    probe
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
  -l 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
```

これに加えて、ユーザーは各コマンドの後に(パラメータなしで) -h を入力すると、そのコマンドの具体的な 使用法を確認することができます。次に例を示します。



coscmd upload -h //コマンドの使用方法を確認します

# 設定ファイルの発行

**COSCMD**ツールは、実行前にまず実行時に必要な情報を設定ファイルから読み込みます。**COSCMD**は、デフォル トでは ~/.cos.conf から設定項目を読み込みます。 **説明:**  設定する前に、COSコンソールでパラメータ設定用のバケットを作成し(例:configure-bucket-125000000)、 キー情報を作成する必要があります。 設定ファイルの例を次に示します。



[common] secret\_id = AKIDA6wUmImTMzvXZNbGLCgtusZ2E8mG\*\*\*\* secret\_key = TghWBCyf5LIyTcXCoBdw1oRpytWk\*\*\*\* bucket = configure-bucket-1250000000 region = ap-chengdu max\_thread = 5 part\_size = 1

```
retry = 5
timeout = 60
schema = https
verify = md5
anonymous = False
```

### 説明:

設定ファイルの schema 項目で、オプション値はhttp、https、デフォルトはhttpsです。

設定ファイルの anonymous 項目で、オプション値はTrue、False、匿名モードを使用するかどうか、つまり署 名を空にするかどうかを示します。

パラメータ設定の詳細については、コマンド coscmd config -h を使用して確認してください。

## configコマンドを使用して設定ファイルを発行

### 注意:

ユーザーには一時キーを使用してSDKを呼び出し、一時権限承認方式によってSDK使用の安全性をさらに向上さ せることをお勧めします。一時キーを申請する際は、最小権限の原則についてのガイドに従い、ターゲットバケッ トまたはオブジェクト以外のリソースが漏洩しないようにしてください。

どうしてもパーマネントキーを使用したい場合は、最小権限の原則についてのガイドに従って、パーマネント キーの権限範囲を限定することをお勧めします。

configコマンドは、 ~/.cos.conf に設定ファイルを自動的に発行します。コマンド形式は次のとおりです。





```
coscmd 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]
```

### 説明:

ここで「[]」内のフィールドはオプション、「<>」内のフィールドは入力が必要なパラメータです。 パラメータ設定の説明は次のとおりです。

オプション	パラメータ説明	有効 値	入力必 須かど うか
-a	キーIDはAPIキーコンソールに移動して取得してください	文字 列	はい
-S	KeyはAPIキーコンソールに移動して取得します	文字 列	はい
-t	一時キーtokenは、一時キーを使用するときに設定が必要で、x-cos- security-tokenヘッダーを設定します	文字 列	いいえ
-b	指定されたバケット名。バケットの命名形式はBucketName-APPIDで す。命名ルールをご参照ください。初回設定時に使用する場合、COSコ ンソールでバケットを作成し、設定ツールとして用いる必要があります	文字 列	はい
-r	バケットの所在リージョンです。リージョンとアクセスドメイン名をご 参照ください。	文字 列	はい
-e	リクエストのENDPOINTを設定します。ENDPOINTパラメータを設定す ると、REGIONパラメータは無効になります。デフォルトのドメイン名 を使用している場合、ここでの設定形式は、 cos. <region>.myqcloud.com となります。グローバルアクセラレーショ ンドメイン名を使用する場合、設定 は cos.accelerate.myqcloud.com となります</region>	文字 列	いいえ
-m	マルチスレッド操作の最大スレッド数(デフォルトは5、範囲は1~30)	数值	いいえ
-р	チャンク操作の1チャンクサイズ(MB単位、デフォルトは1MB、範囲は 1~1000)	数值	いいえ
do-not- use-ssl	HTTPSではなく、HTTPプロトコルを使用します	文字 列	いいえ
 anonymous	匿名操作(署名なし)	文字 列	いいえ

configコマンドの使用例は次のとおりです。





coscmd config -a AChT4ThiXAbpBDEFGhT4ThiXAbp\*\*\*\* -s WE54wreefvds3462refgwewe\*\*\*\* -b

# 一般的なコマンド

# BucketとRegionのコマンドを指定

ユーザーがコマンドを実行するバケット名と所属リージョンを指定しない場合、パラメータの設定時に入力され たバケットがデフォルトで有効になります。異なるバケットで操作を実行する必要がある場合は、バケット名と所



属リージョンを指定する必要があります。

### 説明:

-b <BucketName-APPID> パラメータでバケット名を指定します。バケットの命名形式はBucketName-APPID です。ここに入力するバケット名は、必ずこの形式である必要があります。

-r <region> でRegionを指定すると、バケットの所属リージョンを指定することができます。

コマンド形式



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

操作事例 - バケット名がexamplebucket、所属リージョンが北京のバケットを作成します





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

操作事例 - Dドライブのファイルpicture.jpgを examplebucketという名前のバケットにアップロードします





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

## 設定ファイルとログファイルパスを指定するコマンド

ユーザーが設定ファイルのパスを指定しない場合、デフォルトの設定ファイルパス ~/.cos.conf が使用されま す。ログファイルパスが指定されない場合、デフォルトのログファイルパス ~/.cos.log が使用されます。 **説明:** 

-c <conf\_path> パラメータで設定ファイルパスを指定すると、COSCMDは実行時にこのパスから設定情報 を読み込むようになります。



-1 <log\_conf> パラメータでログパスを指定すると、COSCMDは実行中に生成されたログをこのパスのログ ファイルに出力します。 コマンド形式



coscmd -c <conf\_path> -l <log\_conf> <action> ...

操作事例 - 設定ファイルパスを/data/home/cos\_conf、ログ出力パスを/data/home/cos\_logと指定し、バケット名が examplebucket、所属リージョンが北京のバケットを作成します





coscmd -c /data/home/cos\_conf -l /data/home/cos\_log -b examplebucket-1250000000 -r

# Debugモード実行コマンド

各コマンドの前に -d または --debug を追加すると、コマンドの実行中に詳細な操作情報が表示されます。次 に例を示します。

コマンド形式





coscmd -d upload <localpath> <cospath>

操作事例 - アップロード時に詳細情報を出力します





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

# Silenceモード実行コマンド

各コマンドの前に -s または --silence を追加すると、コマンドの実行中にいかなる情報も出力されなくな ります。

### 説明:

このコマンドは、最小バージョン1.8.6.24を満たす必要があります。 コマンド形式





coscmd -s upload <localpath> <cospath>

操作事例





coscmd -s upload D:/picture.jpg /

# 一般的なバケットコマンド

# バケットの作成

説明:

バケットを作成するコマンドを実行するときは、バケット名を指定するパラメータ -b <BucketName-APPID> と所属リージョンを指定するパラメータ -r <Region>`を付けてください。coscmd createbucketを直接実行した場 合、バケット名と所属リージョンを指定しないと、既存のバケット(パラメータ設定時に入力したバケット)を 作成したのと同じことになるため、エラーが発生します。

コマンド形式



coscmd -b <BucketName-APPID> createbucket

操作事例-バケット名がexamplebucket、所属リージョンが北京のバケットを作成します





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

# バケットの削除

### 説明:

coscmd deletebucket の使用法は、パラメータを設定する際にストレージバケットに対してのみ有効で
す。 -b <BucketName-APPID> でBucketを指定し、 -r <region> でRegionを指定することをお勧めしま
す。
コマンド形式





coscmd -b <BucketName-APPID> deletebucket

操作事例 - 空のバケットを削除します





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

操作事例 - 空ではないバケットを強制的に削除します





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

## 注意:

-f パラメータを使用すると、すべてのファイルやバージョン管理を有効にした後の履歴フォルダ、アップロードによって生成されたフラグメントを含むバケットが強制的に削除されますので、操作は慎重に行ってください。

一般的なオブジェクトコマンド

# ファイルのアップロード

ファイルアップロードのコマンド形式



coscmd upload <localpath> <cospath>

### 注意:

「<>」のパラメータ を、アップロードする必要のあるローカルファイルパス(localpath)とCOSのストレージパス (cospath)に置き換えてください。

操作事例 - Dドライブのpicture.jpgファイルをCOSのdocディレクトリにアップロードします





coscmd upload D:/picture.jpg doc/

操作事例 - Dドライブのdocフォルダからpicture.jpgファイルをCOSのdocディレクトリにアップロードします





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

操作事例 - オブジェクトタイプを指定し、アーカイブタイプのファイルをCOSのdocディレクトリにアップロード します





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

### 注意:

-Hパラメータを使用してHTTP headerを設定する場合は、形式がJSONであることを必ず確認してください。 例: coscmd upload -H "{'x-cos-storage-class':'Archive','Content-Language':'zh-CN'}" <localpath> <cospath> 。その他のヘッダーについては、PUT Objectのドキュメントをご参照ください。 操作事例 - metaメタ属性を設定し、COSのdocディレクトリにファイルをアップロードします





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

# フォルダのアップロード

フォルダアップロードのコマンド形式





coscmd upload -r <localpath> <cospath>

#### 注意:

Windowsユーザーは、システムに標準搭載されているcmdツールやPowerShellでCOSCMDのuploadコマンドを使用することをお勧めします。その他のツール(git bashなど)は、コマンドパスの解析ポリシーがPowerShellと異なるため、ユーザーのファイルが誤ったパスにアップロードされる可能性があります。 操作事例 - DドライブのdocフォルダとそのファイルをCOSのルートパスにアップロードします





coscmd upload -r D:/doc /

操作事例 - DドライブのdocフォルダとそのファイルをCOSのdocパスにアップロードします





coscmd upload -r D:/doc doc

操作事例 - 同時アップロード、md5と同名・同サイズのファイルをスキップします





coscmd upload -rs D:/doc doc

注意:

-sパラメータを使用すれば、同時アップロードによってmd5と一致するファイルのアップロードをスキップでき ます(COSの元のファイルは、必ずバージョン1.8.3.2以降のCOSCMDによってアップロードされている必要があ ります。デフォルトではx-cos-meta-md5のheaderが使用されます)。 操作事例 - 同時アップロード、同名・同サイズのファイルをスキップします





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

### 注意:

-sパラメータを使用すれば、同時アップロードを使用することができます。また、--skipmd5パラメータを使用す ると、同名ファイルのサイズのみが比較され、同じサイズであればアップロードがスキップされます。 操作事例 - 同時アップロードを行い、「Dドライブのdocフォルダにある削除されたファイル」を削除します





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

操作事例 - Dドライブのdocフォルダにある.txtおよび.doc拡張子を持つファイルのアップロードを無視することを 選択します





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

操作事例 - Dドライブのdocフォルダにある.txt拡張子を持つファイルのアップロードを無視することを選択します





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

#### 注意:

フォルダをアップロードするときに、 --ignore パラメータを使用すると、いずれかのタイプのファイルを無 視することができます。 --include パラメータを使用すると、いずれかのタイプのファイルをフィルタリング することができます。shellワイルドカードルールや複数のルールをサポートし、カンマ , で区切ります。いずれ かの種類の拡張子を無視する場合は、最後に , を入力するか、 "" を追加する必要があります。 "" の中に複 数のカンマ区切りルールが含まれる場合は、最初のルールに準じます。 --ignore を使用して特定のフォルダ内のすべてのファイルをフィルタリングしたい場合は、絶対パスを使用し、パスの前後に "" を追加する必要があります。例えば、 coscmd upload -rs D:/doc / --ignore
 "D:/doc/ignore\_folder/\*" などとします。

操作事例 - Dドライブのdocフォルダにある.txtと.docという拡張子を持つファイルをアップロードします



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

操作事例 - Dドライブのdocフォルダにある.txt拡張子を持つファイルをアップロードします





coscmd upload -rs D:/doc / --include "\*.txt"

#### 注意:

10MB以上のファイルをアップロードする場合、COSCMDはマルチパートアップロード方式を採用します。コマ ンドの使用法は単純なアップロードと同じで、 coscmd upload <localpath> <cospath> です。 COSCMDは、大きなファイルのブレークポイントアップロード機能をサポートしています。大きなファイルのマ ルチパートアップロードが失敗した場合、このファイルの再アップロードでは失敗したチャンクのみがアップ ロードされ、最初からやり直すことはありません(再アップロードしたファイルのディレクトリとコンテンツが アップロードしたディレクトリと同じであることを確認してください)。
COSCMDのマルチパートアップロードのときは、チャンクごとにMD5チェックが行われます。 COSCMDのアップロードはデフォルトで x-cos-meta-md5 というヘッダーが付きます。これはこのファイル のmd5値となりますが、--skipmd5パラメータがある場合、このヘッダーは付きません。

## ファイルリストの照会

照会コマンドは次のとおりです。 コマンド形式



coscmd list <cospath>



操作事例 - このバケット内のdoc/というプレフィックスを持つすべてのファイルリストを再帰的に照会します



coscmd list doc/

操作事例 - このバケット内のすべてのファイルリスト、ファイル数およびファイルサイズを再帰的に照会します





coscmd list -ar

操作事例 - examplefolderというプレフィックスを持つすべてのファイルリストを再帰的に照会します





coscmd list examplefolder/ -ar

操作事例 - このバケット内のすべてのファイル履歴を再帰的に照会します





coscmd list -v

説明:

「<>」内のパラメータを、ファイルリストを照会する必要のあるCOS上のファイルのパス(cospath)に置き換えて ください。 <cospath> が空の場合は、デフォルトで現在のバケットルートディレクトリを照会します。

- -a を使用してすべてのファイルを照会します。
- -r を使用して再帰的に照会すると、リストアップされたファイルの数とサイズの合計が末尾に返されます。
- -n num を使用して、照会の最大値を設定します。

## ファイル情報の確認

コマンドは次のとおりです。 コマンド形式



coscmd info <cospath>

操作事例 - doc/picture.jpgのメタ情報を確認します





coscmd info doc/picture.jpg

「<>」内のパラメータを、表示させる必要のあるCOS上のファイルのパス(cospath)に置き換えてください。

ファイルまたはフォルダのダウンロード

### ファイルダウンロードのコマンド形式



coscmd download <cospath> <localpath>

## 注意:

「<>」のパラメータ を、ダウンロードする必要のあるCOS上のファイルのパス(cospath)とローカルストレージパ ス(localpath)に置き換えてください。

操作事例 - COS上のdoc/picture.jpgをD:/picture.jpgにダウンロードします





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

操作事例 - COS上のdoc/picture.jpgをDドライブにダウンロードします





coscmd download doc/picture.jpg D:/

操作事例 - バージョンID付きのpicture.jpgファイルをDドライブにダウンロードします





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

フォルダダウンロードのコマンド形式





coscmd download -r <cospath> <localpath>

操作事例 - docディレクトリをD:/folder/docにダウンロードします





coscmd download -r doc D:/folder/

操作事例 - ルートディレクトリファイルをダウンロードしますが、ルートディレクトリ下のdocディレクトリはス キップします





coscmd download -r / D:/ --ignore "doc/\*"

操作事例 - 現在のバケットのルートディレクトリにあるすべてのファイルを上書きしてダウンロードします





coscmd download -rf / D:/examplefolder/

#### 注意:

ローカルに同名ファイルがある場合、ダウンロードは失敗しますので、 -f パラメータを使用してローカルファ イルを上書きする必要があります。

操作事例 - 現在のbucketのルートディレクトリにあるすべてのファイルを同時にダウンロードし、同名ファイルの md5チェックはスキップします





coscmd download -rs / D:/examplefolder

#### 注意:

パラメータ -s または --sync を使用すると、フォルダをダウンロードする際に、すでにローカルに存在する 同一ファイルをスキップすることができます(ただし、ダウンロードするファイルがCOSCMDのuploadインター フェース経由でアップロードされたもので、そのファイルに x-cos-meta-md5 ヘッダーがふくまれることが前 提条件です)。

操作事例 - 現在のbucketのルートディレクトリにあるすべてのファイルを同時にダウンロードし、同サイズ・同名のファイルをスキップします





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

操作事例 - 現在のバケットのルートディレクトリにあるすべてのファイルを同時にダウンロードし、「クラウドで は削除されたがローカルでは削除されていないファイル」を同時に削除します





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

操作事例 - .txtと.docという拡張子を持つファイルを無視します





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

操作事例 - .txt拡張子を持つファイルを無視します





coscmd download -rs / D:/examplefolder --ignore "\*.txt"

#### 注意:

フォルダをアップロードするときに、 --ignore パラメータを使用すると、いずれかのタイプのファイルを無 視することができます。 --include パラメータを使用すると、いずれかのタイプのファイルをフィルタリング することができます。shellワイルドカードルールや複数のルールをサポートし、カンマ , で区切ります。いずれ かの種類の拡張子を無視する場合は、最後に , を入力するか、 "" を追加する必要があります。 "" の中に複 数のカンマ区切りルールが含まれる場合は、最初のルールに準じます。 --ignore を使用して特定のディレクトリ内のすべてのファイルをフィルタリングしたい場合は、絶対パスを 使用し、パスの前後に "" を追加する必要があります。例えば、 coscmd upload -rs D:/doc / -ignore "D:/doc/ignore\_folder/\*" などとします。

操作事例 - .txtと.docという拡張子を持つファイルをフィルタリングします



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

操作事例 - .txt拡張子を持つファイルをフィルタリングします





coscmd download -rs / D:/examplefolder --include "\*.txt"

## 注意:

古いバージョンのmgetインターフェースは廃止されました。downloadインターフェースはチャンク化されたダウ ンロードを使いますので、downloadインターフェースを使用してください。

## 署名入りダウンロードURLの取得

コマンド形式





coscmd signurl <cospath>

操作事例 - doc/picture.jpgパスの署名付きURLを発行します





coscmd signurl doc/picture.jpg

操作事例 - doc/picture.jpgパスの100s署名付きURLを発行します





coscmd signurl doc/picture.jpg -t 100

### 説明:

「<>」のパラメータを、ダウンロードURLを取得する必要のあるCOS上のファイルのパス(cospath)に置き換えて ください。

-t time を使用して、このURLの署名の有効期間(単位は秒)を設定します。デフォルトは10000sです。

### ファイルまたはフォルダの削除

ファイル削除のコマンド形式





coscmd delete <cospath>

「<>」内のパラメータを、削除する必要のあるCOS上のファイルのパス(cospath)に置き換えてください。 ツールは、削除操作を確認するようユーザーに促します。

操作事例 - doc/exampleobject.txtを削除します





coscmd delete doc/exampleobject.txt

操作事例 - バージョンID付きのファイルを削除します





coscmd delete doc/exampleobject.txt --versionId MTg0NDUxMzc4ODA3NTgyMTErEWN

フォルダ削除のコマンド形式





coscmd delete -r <cospath>

操作事例 - docディレクトリを削除します





coscmd delete -r doc

操作事例 - folder/docディレクトリを削除します





coscmd delete -r folder/doc

操作事例 - docフォルダ下のすべてのバージョン管理ファイルを削除します





coscmd delete -r doc/ --versions

説明:

一括削除では y を入力して確定する必要がありますが、 −f パラメータを使用すると確認をスキップして直接 削除することができます。

フォルダを削除するコマンドを実行すると、現在のフォルダとそのファイルが削除されますのでご注意ください。ただし、バージョン管理されたファイルを削除する場合は、バージョンIDを指定して削除する必要があります。



## マルチパートアップロードファイルのフラグメントの照会

コマンド形式



coscmd listparts <cospath>

操作事例 - doc/プレフィックス付きファイルフラグメントを照会します





coscmd listparts doc/

# マルチパートアップロードファイルのフラグメントをクリア

コマンド形式





coscmd abort

操作事例 - アップロードされたすべてのファイルフラグメントを削除します





coscmd abort

ファイルまたはフォルダのコピー

ファイルコピーのコマンド形式




coscmd copy <sourcepath> <cospath>

操作事例 - 同じバケット内でコピーする場合、examplebucket-1250000000バケット内のpicture.jpgファイルをdoc フォルダにコピーします





coscmd -b examplebucket-1250000000 -r ap-chengdu copy examplebucket-1250000000.ap-c

操作事例 - 異なるバケット内でコピーする場合、examplebucket2-1250000000バケットのdoc/picture.jpgオブジェ クトをexamplebucket1-125000000バケットのdoc/examplefolder/にコピーします





coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy examplebucket2-1250000000.

ストレージタイプを変更し、ファイルタイプを低頻度ストレージに変更します





coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy examplebucket2-1250000000. ストレージタイプを変更し、ファイルタイプをCASに変更して、photo.jpgとリネームします





coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy examplebucket2-1250000000.

フォルダコピーのコマンド形式





coscmd copy -r <sourcepath> <cospath>

操作事例 - examplebucket2-1250000000バケット内のexamplefolderディレクトリをexamplebucket1-1250000000 バケットのdocディレクトリにコピーします





coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy -r examplebucket2-12500000

#### 説明:

「<>」内のパラメータを、コピーする必要のあるCOS上のファイルのパス(sourcepath)と、COS上にコピーする必要のあるファイルのパス(cospath)に置き換えてください。

**sourcepath**の形式は、 <BucketName-APPID>.cos.<region>.myqcloud.com/<cospath> です。 -dパラメータを使用して x-cos-metadata-directive パラメータを設定します。オプション値はCopyと Replacedで、デフォルトはCopyです。 -Hパラメータを使用してHTTP headerを設定する場合は、形式がJSONであることを確認してください。 例: coscmd copy -H -d Replaced "{'x-cos-storage-class':'Archive','Content-Language':'zh-CN'}" <localpath> <cospath> 。その他のヘッダーについては、PUT Object - Copyのド キュメントをご参照ください。

## ファイルまたはフォルダの移動

#### 注意:

移動コマンドの はと同一にすることはできません。同一にした場合はファイルが削除されます。moveコマンド は先にコピーを行ってから削除するため、 パスのファイルが最終的に削除されることが原因です。

#### ファイル移動のコマンド形式





coscmd move <sourcepath> <cospath>

操作事例 - 同じバケット内で移動する場合、examplebucket-1250000000バケット内のpicture.jpgファイルをdoc フォルダに移動します





coscmd -b examplebucket-1250000000 -r ap-chengdu move examplebucket-1250000000.ap-c

操作事例 - 異なるバケット内で移動する場合、examplebucket2-1250000000パケット内のdoc/picture.jpgオブジェ クトをexamplebucket1-1250000000パケットのdoc/folder/に移動します





coscmd -b examplebucket1-1250000000 -r ap-guangzhou move examplebucket2-1250000000.

操作事例 - ストレージタイプを変更し、ファイルタイプを低頻度ストレージに変更します





coscmd -b examplebucket1-1250000000 -r ap-guangzhou move examplebucket2-1250000000.

操作事例 - ストレージタイプを変更し、ファイルタイプをCASに変更します





coscmd -b examplebucket1-1250000000 -r ap-guangzhou move examplebucket2-1250000000.

フォルダ移動のコマンド形式





coscmd move -r <sourcepath> <cospath>

操作事例 - examplebucket2-1250000000バケット内のexamplefolderディレクトリをexamplebucket1-1250000000 バケットのdocディレクトリに移動します





coscmd -b examplebucket1-1250000000 -r ap-guangzhou move -r examplebucket2-12500000

#### 説明:

「<>」内のパラメータを、移動させる必要のあるCOS上のファイルのパス(sourcepath)と、COSに移動する必要のあるファイルのパス(cospath)に置き換えてください。

**sourcepath**の形式は、 <BucketName-APPID>.cos.<region>.myqcloud.com/<cospath> です。 -dパラメータを使用して x-cos-metadata-directive パラメータを設定します。オプション値はCopyと Replacedで、デフォルトはCopyです。 -Hパラメータを使用してHTTP headerを設定する場合は、形式がJSONであることを確認してください。 例: coscmd move -H -d Replaced "{'x-cos-storage-class':'Archive','Content-Language':'zh-CN'}" <localpath> <cospath> 。その他のヘッダーについては、PUT Object - copyのド キュメントをご参照ください。

# オブジェクトのアクセス権限の設定

コマンド形式



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

操作事例 - アカウント10000000001にpicture.jpgの読み込み権限を付与します



coscmd putobjectacl --grant-read 10000000001 picture.jpg

操作事例 - ファイルのアクセス権限を照会します





coscmd getobjectacl picture.jpg

# バージョン管理の有効化/一時停止

コマンド形式





coscmd putbucketversioning <status>

操作事例 - バージョン管理を有効化します





coscmd putbucketversioning Enabled

操作事例 - バージョン管理を一時停止します





coscmd putbucketversioning Suspended

操作事例 - バージョン管理を照会します





coscmd getbucketversioning

注意:

「<>」のパラメータを、ご希望のバージョン管理ステータス(status)に置き換えてください。

一度バージョン管理を有効にしたバケットは、バージョン管理を有効にしていない状態(初期ステータス)に戻 すことはできません。ただし、このバケットのバージョン管理を一時停止することはでき、その後にアップロー ドされるオブジェクトでは複数のバージョンが生成されなくなります。

アーカイブファイルのリカバリ



## アーカイブファイルリカバリのコマンド形式



coscmd restore <cospath>

操作事例 - 高速取得モードでpicture.jpgをリトリーブします。有効期間は3日間です





coscmd restore -d 3 -t Expedited picture.jpg

アーカイブファイル一括リカバリのコマンド形式





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

操作事例 - 高速取得モードでexamplefolder/ディレクトリをリトリーブします。有効期間は3日間です





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

説明:

「<>」のパラメータを、ファイルリストを照会する必要のあるCOS上のファイルのパス(cospath)に置き換えてください。

-d <day> を使用して、一時コピーの有効期限を設定します。デフォルト値は7です。

-t <tier> を使用してリカバリモードを指定します。列挙値はExpedited (高速取得モード)、Standard (標 準取得モード)、Bulk (一括取得モード)であり、デフォルト値はStandardです。

# よくあるご質問

COSCMDツールの使用に関するご質問は、COSCMDツールに関するよくあるご質問をご参照ください。

# COS Migrationツール

最終更新日:::2024-06-26 10:27:35

# 機能説明

COS Migrationは、COSのデータ移行機能を統合したオールインワンツールです。簡単な設定操作により、ユー ザーはローカルデータをCOSにすばやく移行することができ、以下のような特徴があります:

中断からの再開:ツールはアップロード時の中断からの再開をサポートしています。大容量ファイルについては、 途中で終了した場合やサービスの障害があった場合、ツールを再実行して、アップロードが完了していないファ イルのアップロードを再開することができます。

マルチパートアップロード:COSにオブジェクトをチャンクでアップロードします。

並列アップロード:複数のオブジェクトの同時アップロードをサポートします。

#### 注意:

COS Migrationのエンコード形式は、UTF-8形式のみをサポートします。

このツールを使って同名ファイルをアップロードすると、デフォルトでは古い方の同名ファイルが上書きされま す。同名ファイルをスキップするには、追加の設定が必要です。

ローカルデータ移行以外の場合、Migration Service Platformを優先して使用してください。

COS Migrationは1回限りの移行に用いるサービスであり、継続的な同期のシナリオには適しません。例えばロー カルに毎日ファイルが追加され、COSに継続的に同期する必要がある場合、COS Migrationは移行タスクの重複を 避けるために移行成功の記録を保存するため、継続的同期後の記録のスキャン時間が増大し続けることになりま す。このようなシナリオではファイルの同期の使用をお勧めします。

# 使用環境

#### システム環境

WindowsとLinuxシステム。

#### ソフトウェア依存

JDK 1.8 X64以降、JDKのインストールと設定の詳細については、Javaのインストールと設定をご参照ください。 Linux環境ではIFUNCのサポートが必要です。環境のbinutilsのバージョンをチェックし、2.20以降のバージョンで あることを確認します。

# 利用方法

### 1. ツールの取得

COS Migrationツールに移動してダウンロードします。

## 2. 解凍ツールキット

### Windows

解凍して、次のようにディレクトリに保存します。



C:\\Users\\Administrator\\Downloads\\cos\_migrate

Linux



解凍して、いずれかのディレクトリに保存します。



unzip cos\_migrate\_tool\_v5-master.zip && cd cos\_migrate\_tool\_v5-master

## マイグレーションツールの構造

COS Migrationツールを正しく解凍すると、次のようなディレクトリ構造になります。





COS_Migrate_tool	
――conf #設定ファイルが配置されているディレクトリ	
—config.ini #設定ファイルの移行	
――db #移行に成功した記録を保存	
——dep #プログラムのメインロジックコンパイルによって発行されたJARパッケージ	
――log #ツールの実行中に発行されたログ	
——opbin #コンパイル用スクリプト	
result #マイグレーション成功記録を保存するためのディレクトリであり、記録のファイル名は 「期日	
ー-src #ツールのソースコード	
——tmp #一時ファイルストレージディレクトリ	
— gitignore #git バージョン管理で無視されたファイルとフォルダ	



|---pom.xml #プロジェクト設定ファイル
 |--README #説明ドキュメント
 |--start\_migrate.sh #Linuxでの移行起動スクリプト
 |--start\_migrate.bat #Windowsでの移行起動スクリプトdbディレクトリには、主にツールの移行に成功

ログディレクトリには、ツールの移行に伴うすべてのログが記録されています。移行中にエラーが発生した場合 は、まずこのディレクトリ内のerror.logをご確認ください。

#### 説明:

dbディレクトリには、主にツールの移行に成功したファイル識別子が記録されています。各移行タスクはdb内の 記録に優先順位をつけ、現在のファイル識別子がすでに記録されている場合は現在のファイルをスキップし、それ 以外の場合はファイルの移行を実行します。

ログディレクトリには、ツールの移行に伴うすべてのログが記録されています。移行中にエラーが発生した場合 は、まずこのディレクトリ内のerror.logをご確認ください。

#### 3. 設定ファイル config.iniの変更

移行起動スクリプトを実行する前に、設定ファイルconfig.iniを変更する必要があります(パ

ス: ./conf/config.ini )。config.iniの内容は、以下のセクションに分けられます。

#### 3.1 移行タイプの設定

type はマイグレーションのタイプを示し、固定入力とします type=migrateLocal 。





[migrateType]
type=migrateLocal

# 3.2 移行タスクの設定

ユーザーは、主にターゲット**COS**情報への移行の設定や移行タスク関連の設定など、実際の移行要件に基づいた 設定を行います。





# マイグレーションツールのパブリック設定セクションには、ターゲットcosに移行する必要のあるアカウント [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
executeTimeWindow=00:00,24:00
<pre>outputFinishedFileFolder=./result</pre>
resume=false
skipSamePath=false
requestTryCount=5

名称	説明	デフォルト 値
secretId	ユーザーキーのSecretIdです。 COS_SECRETID を実際のキー 情報に置き換えてください。CAMコンソールのTencent Cloud APIキー画面に進むと取得できます	-
secretKey	ユーザーキーのSecretKeyです。 COS_SECRETKEY を実際の キー情報に置き換えてください。CAMコンソールのTencent Cloud APIキー画面に進むと取得できます	-
bucketName	ターゲットBucketの名称で、命名形式は <bucketname- APPID&gt; 。Bucket名には必ずAPPIDを含める必要があります。 例: examplebucket-125000000</bucketname- 	-
region	ターゲットBucketのRegion情報です。COSのリージョンの略称 については、リージョンとアクセスドメイン名をご参照ください	-
storageClass	データ移行後のストレージタイプです。オプション値はStandard (標準ストレージ)、Standard_IA(低頻度ストレージ)、 Archive(アーカイブストレージ)、Maz_Standard(複数のAZを備 えた標準ストレージ)、Maz_Standard_IA(複数のAZを備えた 低頻度ストレージ)です。詳細については、ストレージタイプの 概要をご参照ください	Standard
cosPath	移行先の <b>COS</b> パスです。 / は <b>Bucket</b> のルートパスへの移 行、 /folder/doc/ は <b>Bucket</b> の /folder/doc/ への移行を 表します。 /folder/doc/ が存在しない場合は、自動的にパ スが作成されます	/
https	HTTPS通信を使用するかどうかです。onはオン、offはオフを表 します。オンの場合は通信速度が遅くなり、セキュリティ要件の 高いシナリオに適しています	off
tmpFolder	他のクラウドストレージからCOSへの移行中に一時ファイルを 保存するために使用されたディレクトリは、移行が完了すると削 除されます。形式は絶対パスである必要があります。	./tmp



	Linuxの場合、 /a/b/c のようにシングルスラッシュで区切り ます。 Windowsの場合、 E:\\\\a\\\\b\\\\c のようにダブルバッ クスラッシュ区切ります。 デフォルトでは、ツールが配置されているパスのtmpディレクト リです	
smallFileThreshold	小さなファイルの閾値のバイトです。この閾値以上であれば、マ ルチパートアップロードを使用します。それ以外の場合はシンプ ルアップロードを使用します。デフォルトは5MB(5242880 Byte)です	5242880
smallFileExecutorNum	小さなファイル(smallFileThresholdより小さいファイル)の同 時実行性で、シンプルアップロードを使用します。パブリック ネットワーク経由でCOSに接続しており、帯域幅が小さい場合 は、この同時実行性を低下させてください	64
bigFileExecutorNum	大きなファイル(smallFileThreshold以上のファイル)の同時実 行性で、マルチパートアップロードを使用します。パブリック ネットワーク経由でCOSに接続しており、帯域幅が小さい場合 は、この同時実行性を低下させてください	8
entireFileMd5Attached	マイグレーションツールがフルテキストのMD5を計算し、その 後の検証のためにファイルのカスタムヘッダーx-cos-meta-md5 に格納することを表します。COSのマルチパートアップロード された大きなファイルのetagは、フルテキストのMD5ではない からです	on
executeTimeWindow	実行時間ウィンドウ。時刻粒度は分単位です。このパラメータ は、マイグレーションツールが毎日実行される時間帯を定義しま す。例えば: パラメータ 03:30,21:00は、タスクが朝 03:30 から夜 21:00 の間 に実行され、それ以外の時間はスリープ状態に入ることを意味 します。スリープ状態でマイグレーションを一時停止すると、マ イグレーションの進行状況が維持され、次のタイムウィンドウ になると、自動的に続行されます。後の時点が前の時点より大き い必要があることに注意してください。	00:00,24:00
outputFinishedFileFolder	このディレクトリには、移行に成功した結果が保存されます。結 果のファイルは、例えば ./result/2021-05-27.out のよ うに日付に基づいて命名されます。ここで./resultは、作成済みの ディレクトリです。ファイル内容の各行の形式は、絶対パス\\t ファイルサイズ\\t最終更新時刻です。設定を空欄にすると、結果 は出力されません。	./result
resume	ソースファイルのリストを、最後に実行した結果までトラバーサ	false
	ル処理し続けるかどうかです。デフォルトは先頭からです。	
-----------------	---	-------
skipSamePath	COSにすでに同名ファイルがある場合に、そのままスキップす るかどうかです。デフォルトではスキップされず、元のファイル が上書きされます。	false
requestTryCount	各ファイルのアップロードの合計試行回数。	5

### 3.3 データソース情報の設定

### 3.3.1 ローカルのデータソースmigrateLocalの設定

ローカルからCOSに移行する場合は、この部分の設定を行います。具体的な設定項目および説明は以下のとおり です。





```
# ローカルからCOSへの移行の設定セクション
[migrateLocal]
localPath=E:\\\\code\\\\java\\\\workspace\\\\cos_migrate_tool\\\\test_data
excludes=
ignoreModifiedTimeLessThanSeconds=
```

設定項目	説明
localPath	ローカルディレクトリ。形式は絶対パスである必要があります。



	Linuxの場合、 /a/b/c のようにシングルスラッシュで区切りま す。 Windowsの場合、 E:\\\\a\\\b\\\c のようにダブルバック スラッシュで区切ります 注意:このパラメータはディレクトリのパスのみを入力でき、具体 的なファイルのパスは入力できません。具体的なファイルのパスを 入力した場合は目的のオブジェクト名の解析エラーとなり、 cosPath=/の場合、バケット作成リクエストであると誤って解析さ れます
excludes	除外するディレクトリまたはファイルの絶対パスで、localPathより 下のディレクトリまたはファイルは移行されないことを表します。 複数の絶対パスはセミコロンで区切ります。空欄の場合、localPath より下のすべてが移行されることを表します
ignoreModifiedTimeLessThanSeconds	更新時間が現在時刻から一定時間に満たないファイルを除外しま す。単位は秒で、デフォルトでは設定されていません。これは、 lastmodified時刻に基づくフィルタリングを行わないことを表しま す。ファイルの更新と同時にマイグレーションツールを実行し、更 新中のファイルがCOSに移行、アップロードされないようにしたい お客様向けです。例えば300に設定すると、更新から5分以上経過し たファイルのみをアップロードします

### 4. マイグレーションツールの実行

Windows

start\_migrate.batをダブルクリックして実行します。

Linux

1.設定ファイルconfig.iniから設定を読み込み、次のコマンドを実行します。





sh start\_migrate.sh

2.一部のパラメータは、コマンドラインから設定を読み込み、次のコマンドを実行します。





sh start\_migrate.sh -Dcommon.cosPath=/savepoint0403\_10/

説明:

ツールは、コマンドラインからの読み取りと設定ファイルからの読み取りという2種類の設定項目の読み取りをサ ポートしています。

コマンドラインは設定ファイルより優先されます。同じ設定オプションであれば、コマンドラインのパラメータが 優先されるということです。

コマンドラインで設定項目を読み取る形式により、Bucket名、COSパス、移行するソースパスなど、2つのタスクの主要な設定項目が完全に同一でない場合、ユーザーは異なる移行タスクを同時に実行しやすくなります。さまざ

まな移行タスクを異なるdbディレクトリに書き込むことで、同時移行が確保されるからです。上記のツール構造のdb情報をご参照ください。

設定項目の形式は、\*\*-D{sectionName}.{sectionKey}={sectionValue}です。ここで、sectionNameは設定ファイル のセクション名、sectionKeyはセクション内の設定項目名、sectionValueはセクション内の設定項目値を表しま す。移行先のCOSのパスを設定した場合、-Dcommon.cosPath=/bbb/ddd\*\*で表されます。

## 移行のメカニズムとプロセス

### 移行メカニズムの原理

COS マイグレーションツールには状態があり、マイグレーションに成功したものは db ディレクトリに記録さ れ、KV の形として leveldb ファイルに保存されます。毎回マイグレーションする前に、マイグレーションするパ スが db に存在するかどうかをチェックします。もし存在し、且つ mtime が db に存在するものと一致している場 合にはマイグレーションをスキップし、そうしない場合にはマイグレーションします。そのため、COS を探すの ではなく、db の中にマイグレーション成功の記録があるかどうかを確認するわけです。マイグレーションツール の代わりに、他の方法(COSCMD やコンソールなど)を通じてファイルを削除・変更した場合、マイグレーショ ンツールを実行しても、その変更を察知しないため、再度マイグレーションを行わないことになります。

### 移行プロセスの手順

1. 設定ファイルが読み取られ、セクションがマイグレーションtypeに従って読み取られ、パラメータがチェック されます。

**2. db** においてマイグレーションするファイルのマークをスキャンして比較し、アップロードが許可されているか どうかを判断します。

3. マイグレーション実行中、実行結果がプリントアウトされます。そのうち、inprogress はマイグレーション中、 skip はスキップ、fail は失敗、ok は成功、condition\_not\_match はマイグレーション条件を満たさないためにス キップされたファイル(lastmodifed や excludes など)を示します。失敗の詳細情報は log の error ログで確認する ことができます。

4. 移行全体の終了時に、累積した移行の成功数、失敗数、スキップ数、所要時間などの統計情報が出力されま す。マイグレーションツールは成功したものをスキップして失敗したものを再移行するので、失敗した場合は errorログを確認するか、再実行してください。実行完了結果の略図は以下のとおりです。

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	

## よくあるご質問

COS Migrationツールの使用中に移行失敗や実行エラーといった異常が発生した場合は、COS Migrationツールに 関するよくあるご質問をご参照ください。

## 結論

もちろん、COS は上記のアプリケーションとサービスを提供するだけでなく、多くの人気のあるオープンソース アプリケーションをも提供し、Tencent Cloud COS プラグインを統合しております。 こちら をクリックしてワン クリックで起動し、すぐにお使いになることを歓迎します!

# FTP Serverツール

最終更新日:::2024-06-26 10:27:35

## 機能の説明

COS FTP Serverは、FTPプロトコルによるCloud Object Storage(COS)内のオブジェクトやディレクトリの直接 操作をサポートしています。これにはファイルのアップロード、ダウンロード、削除、フォルダ作成などが含まれ ます。FTP ServerツールはPythonで実装されており、インストールがより簡単です。

### 機能説明

**アップロードメカニズム**:ストリーミングアップロードはローカルディスクに記録されることなく、標準的なFTP プロトコルに従って作業ディレクトリを設定するだけで、実際のディスクストレージ容量を占有することはあり ません。

**ダウンロードメカニズム**:ダイレクトストリーミングでクライアントへ戻ります。

**ディレクトリメカニズム**: bucketはFTP Server全体のルートディレクトリであり、bucketの下に複数のサブディレクトリを作成することができます。

複数bucketのバインド:複数のbucketの同時バインドをサポートします。

説明:

複数bucketのバインド:異なるFTP Serverの作業パス(home\_dir)を介して実装されるため、異なるbucketとユー ザー情報を指定する場合は、home\_dirが異なることを確認する必要があります。

**削除操作の制限**:新しいFTP Serverでは、各ftpユーザーに対しdelete\_enableオプションを設定して、そのFTP ユーザーがファイルの削除を許可されているかどうかを識別できるようになっています。

サポートされているFTPコマンド:put、mput、get、rename、delete、mkdir、ls、cd、bye、quite、size。 サポートされていないFTPコマンド:append、mget(ネイティブのmgetコマンドはサポートされていませんが、 FileZillaクライアントなど一部のWindowsクライアントでは、引き続き一括ダウンロードが可能です) 説明:

FTP Serverツールは現在、中断からの再開機能をサポートしていません。

## 使用の開始

### システム環境

オペレーティングシステム:Linux。Tencent CentOSシリーズCVMを推奨します。現時点ではWindowsシステムは サポートしていません。



psutilが依存するLinuxシステムパッケージ: python-devel(またはpython-dev。Linuxディストリビューションに よって名前が異なります)。 yum install python-devel や aptitude install python-dev などの Linuxのパッケージ管理ツールによって追加されます。 Pythonインタープリターバージョン: Python 2.7、インストールと設定は、Pythonのインストールと設定を参照し て行ってください。 **説明:** FTP Serverツールは、Python 3をサポートしていません。 依存パッケージ: cos-python-sdk-v5 (≥1.6.5) pyftpdlib (≥1.5.2)

<mark>psutil</mark>(>=5.6.1)

#### 使用制限

COS XMLバージョンに適用します。

#### インストールと実行

FTP Serverツールのダウンロードアドレスは、cos-ftp-serverです。インストール手順は次のとおりです。
1. FTP Serverのディレクトリに移動してsetup.pyを実行し、FTP Serverとそれに関連する依存ライブラリをインストールします(インターネットへのアクセスが必要です)。





python setup.py install # ここでは、お客様のアカウントのsudoまたはroot権限が必要な場合があ

サンプル設定ファイル conf/vsftpd.conf.example をコピーし、 conf/vsftpd.conf というファイル
 名にします。本ドキュメントの設定ファイルの項を参照し、bucketやユーザー情報を適切に設定します。
 ftp\_server.pyを実行して、FTP Serverを起動します。





python ftp\_server.py

また、FTP Serverの起動方法には、次の2通りがあります。 nohupコマンドを使用して、バックグラウンドプロセスとして起動します。





nohup python ftp\_server.py >> /dev/null 2>&1 &

screenコマンドを使用して、バックグラウンドで実行します(screenツールをインストールする必要があります)。





```
screen -dmS ftp
screen -r ftp
python ftp_server.py
#ショートカットキーCtrl+A+Dを使用して、メインscreenに戻ります。
```

### 実行の停止

FTP Serverを直接起動する場合、またはFTP Serverをバックグラウンドでscreenモードで実行する場合は、ショー トカットキー Ctrl+C を使用すると、FTP Serverの実行を停止することができます。 nohupコマンドで起動した場合は、以下のように停止させることができます。





ps -ef | grep python | grep ftp\_server.py | grep -v grep | awk '{print \$2}' | xargs

<a id="conf"></a>

## 設定ファイル

**FTP Server**ツールのサンプル設定ファイルは、 conf/vsftpd.conf.example ですので、これをコピーして vsftpd.confという名前を付け、次の設定項目に従って設定してください。





```
[COS_ACCOUNT_0]
cos_secretid = COS_SECRETID # お客様のSECRETIDに置き換えてください
cos_secretkey = COS_SECRETKEY # お客様のSECRETKEYに置き換えてください
cos_bucket = examplebucket-125000000
cos_region = region # お客様のバケットリージョンに置き換えてください
cos_protocol = https
#cos_endpoint = region.myqcloud.com
home_dir = /home/user0 # FTPをマウントしたいローカルパスに置き換えてください (マシン上に
ftp_login_user_name=user0 # ユーザー定義のアカウントに置き換えてください
ftp_login_user_password=pass0 # ユーザー定義のパスワードに置き換えてください
authority=RW # このユーザーの読み書き権限を設定します。Rは読み込み、Wは書き込そ
```



# このftpユーザーの削除を許可する場合はtrue(デフォルト)、 delete enable=true [COS ACCOUNT 1] cos\_secretid = COS\_SECRETID # お客様のSECRETIDに置き換えてください cos\_secretkey = COS\_SECRETKEY # お客様のSECRETKEYに置き換えてください cos\_bucket = examplebucket-125000000 cos\_region = region # お客様のバケットリージョンに置き換えてください  $\cos \text{protocol} = \text{https}$ #cos\_endpoint = region.myqcloud.com # FTPをマウントしたいローカルパスに置き換えてください(マシン上に home dir = /home/user1 ftp\_login\_user\_name=user1 # ユーザー定義のアカウントに置き換えてください ftp\_login\_user\_password=pass1 #ユーザー定義のパスワードに置き換えてください # このユーザーの読み書き権限を設定します。Rは読み込み、Wは書き込み。 authority=RW # このftpユーザーの削除を許可する場合はtrue(デフォルト)、 delete\_enable=false [NETWORK] # FTP ServerがゲートウェイまたはNATの背後にある場合は、この設定項目を使用してゲートウェイのIPアト masquerade address = XXX.XXX.XXX.XXX # FTP Serverのリスニングポート、デフォルトは2121です。ファイアウォールがこのポートを開放する必要フ listen port = 2121 # passive\_portは、passiveモードでポートの選択範囲を設定できます。デフォルトでは、[60000, 6553. passive\_port = 60000,65535 [FILE\_OPTION] # デフォルトの単一ファイルサイズは最大200gをサポートしますが、大きすぎるサイズを設定することはお勧。 single\_file\_max\_size = 21474836480 [OPTIONAL] # 以下の設定は、特に必要がなければdefaultのままにしておくことをお勧めします。設定する必要がある場合 min\_part\_size = default upload\_thread\_num = default max\_connection\_num = 512 = 10000# 1sコマンドでリストアップできるファイルの最大数 max\_list\_file # ログ出力レベルを設定します log\_level = INFO # ログをストレージするディレクトリを設定します。

#### 説明:

log\_dir

各ユーザーを異なるbucketにバインドする場合は、[COS\_ACCOUNT\_X]のsectionを追加するだけで可能です。

それぞれのCOS ACCOUNT Xのsectionには、次のような説明があります

= loq

各ACCOUNTのユーザー名(ftp login user name)とユーザーのホームディレクトリ(home dir)は、それぞれ異なっ ていなければなりません。また、ホームディレクトリは、システム上の実際のディレクトリである必要がありま す。

各COS FTP Serverに同時にログインできるユーザー数は100以下とします。

endpointとregionは同時に有効になりません。パブリッククラウドCOSサービスを使用するには、regionフィール ドを正しく入力するだけで可能です。endpointは一般的に、プライベートなデプロイ環境で使用されます。region とendpointの両方が入力された場合、endpointが優先して有効になります。

設定ファイルのOPTIONALオプションは、上級ユーザーがアップロード性能を調整するためのオプションです。 マシンの性能に応じてアップロードスライスのサイズと同時アップロードするスレッド数を適度に調整すること によって、アップロード速度を高めることができます。通常、ユーザーはこれを調整する必要はなく、デフォルト 値のままで問題ありません。

また、最大接続数を制限するオプションも提供しています。最大接続数を制限したくない場合は、ここに0を入力 します。0は、最大接続数が制限されていないことを意味します(ただし、マシンの性能に応じて合理的に評価す る必要があります)。

通常は、設定ファイルのmasquerade\_address設定項目において、クライアントがCOS FTP Serverに接続するために使用するIPアドレスを指定することをお勧めします。この点に関してご質問がある場合は、FTP Serverツールのよくあるご質問のドキュメントをご参照ください。

FTP Serverに複数のIPアドレスがある場合、ifconfigコマンドを実行すると、パブリックネットワークにマッピン グされたネットワークカードIPは10.xxx.xxx.となり、マッピングしたパブリックネットワークIPは 119.xxx.xxxとみなされます。この際、FTP Serverが明示的にmasquerade\_addressをパブリックネットワーク IP(119.xxx.xxx)として設定しない場合、FTP ServerはPassiveモードでプライベートネットワークアドレス (10.xxx.xxx)を使用してクライアントにパケットを返すことがあります。この場合、クライアントはFTP Server に接続できますが、クライアントにパケットを正しく返すことができません。したがって通常は、クライアント がServerに接続する際に使用するIPアドレスをmasquerade\_addressに設定することをお勧めします。 設定ファイルのlisten\_port設定項目は、COS FTP Serverのリスニングポートであり、デフォルトは2121です。 passive\_port設定項目は、COS FTP Serverのデータチャネルリスニングポート範囲であり、デフォルトでは

[60000, 65535]の範囲で選択されています。クライアントがCOS FTP Serverに接続するときは、ファイアウォー ルがlisten\_portとpassive\_portに設定されたポートを開放していることを確認する必要があります。

# クイックプラクティス

### Linuxftpコマンドを使用したCOS FTP Serverへのアクセス

1. Linux ftpクライアントをインストールします。





yum install -y ftp

2. Linuxコマンドラインでコマンド ftp [ipアドレス] [ポート番号] を使用して、COS FTP Serverに接続しま す。例えば、次のコマンドです。





ftp 192.xxx.xx.103 2121

ftpコマンドでは、IP設定はサンプル設定ファイル conf/vsftpd.conf.example の masquerade\_address 設 定項目に対応しています。この例では、IPは192.xxx.xx.103に設定されています。

ftpコマンドでは、ポート設定はサンプル設定ファイル conf/vsftpd.conf.example のlisten\_port</mark>設定項目に 対応しています。この例では、**2121**に設定されています。

3. 上記のコマンドを実行すると、**Name**と**Password**という入力すべき項目が表示されますので、**COS FTP** Serverの設定項目ftp\_login\_user\_nameとftp\_login\_user\_passwordに設定した内容を入力すると、接続が完了しま す。 **Name**:サンプル設定ファイル conf/vsftpd.conf.example の**ftp\_login\_user\_name**設定項目に対応してい ます(設定が必要です)。

**Password**:サンプル設定ファイル conf/vsftpd.conf.example の **ftp\_login\_user\_password**設定項目に対応しています(設定が必要です)。

### FileZillaコマンドを使用したCOS FTP Serverへのアクセス

1. FileZillaクライアントをダウンロードしてインストールします。

2. FileZillaクライアントでCOS FTP Serverのアクセス情報を設定した後、**クイック接続**をクリックします。 ホスト(H):サンプル設定ファイル conf/vsftpd.conf.example の masquerade\_address設定項目に対応し ています。この例では、IPは192.xxx.xx.103に設定されています。

#### 注意:

COS FTP ServerがゲートウェイまたはNATの背後にある場合は、この設定項目を使用してゲートウェイのIPアドレスまたはドメイン名をCOS FTP Serverに割り当てることができます。

**ユーザー名(U):**サンプル設定ファイル conf/vsftpd.conf.example の **ftp\_login\_user\_name**設定項目に対応しています(設定が必要です)。

**パスワード(W)**:サンプル設定ファイル conf/vsftpd.conf.example の **ftp\_login\_user\_password**設定項目 に対応しています(設定が必要です)。

**ポート(P)**:サンプル設定ファイル conf/vsftpd.conf.example の**listen\_port**設定項目に対応しています。 この例では、2121に設定されています。

## よくあるご質問

FTP Serverツールの使用時にエラーが発生した場合やアップロードの制限についてご質問がある場合は、FTP Serverツールのよくあるご質問をご参照ください。

# Hadoopツール

最終更新日:::2024-06-26 10:27:35

# 機能説明

Hadoop-COSは、Tencent Cloud Object Storage(COS)をベースとして標準的なHadoopファイルシステムを実装し ています。これにより、Hadoop、Spark、TezなどのビッグデータコンピューティングフレームワークがCOSと統 合し、HDFSファイルシステムにアクセスする場合と同じように、COSに保存されたデータの読み書きを可能にす るサポートを提供しています。

Hadoop-COSはURIのschemeとしてcosnを使用するため、CosNファイルシステム用のHadoop-COSとも呼ばれます。

### 使用環境

### システム環境

Linux、Windows、macOSシステムがサポートされています。

### ソフトウェア依存

Hadoop-2.6.0およびそれ以降のバージョン。

#### 説明:

1. 現在、Hadoop-COSは、Apache Hadoop-3.3.0に正式に統合されました公式統合。

2. Apache Hadoop-3.3.0以前のバージョンまたはCDHがHadoop-cos jarパッケージを統合した後、NodeManagerを 再起動してjarパッケージをロードする必要があります。

3. 特定のHadoopバージョンのjarパッケージをコンパイルする必要がある場合は、pomファイルのhadoop.version を変更してコンパイルできます。

### ダウンロードとインストール

### Hadoop-COSディストリビューションパッケージとその依存関係の取得

ダウンロードアドレス: Hadoop-COS release。

### Hadoop-COSプラグインのインストール

```
    hadoop-cos-{hadoop.version}-{version}.jar と cos_api-bundle-
{version}.jar を $HADOOP_HOME/share/hadoop/tools/lib にコピーします。
    説明:
```

Hadoopの特定バージョンに応じて対応するjarパッケージを選択します。releaseの中に対応するバージョンのjar パッケージが提供されていない場合は、pomファイルのHadoopバージョン番号を変更することで再コンパイルと 発行ができます。

**2. hadoop-env.sh**ファイルを変更します。 \$HADOOP\_HOME/etc/hadoop ディレクトリに移動し、hadoop-env.sh ファイルを編集して、以下の内容を追加し、cosn関連のjarパッケージをHadoop環境変数に追加します。



```
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
```

# 設定方法

### 設定項目の説明

プロパティキー	説明	デフォルト値
fs.cosn.userinfo. secretId/secretKey	お客様のアカウントのAPIキー情報を入力します。CAMコン ソールにログインすれば、Tencent Cloud APIキーを確認するこ とができます。	なし
fs.cosn. credentials.provider	SecretldとSecretKeyの取得方法を設定します。現在、5つの取 得方法がサポートされています。 1. org.apache.hadoop.fs.auth.SessionCredential Provider : リクエストURIからsecret idとsecret keyを取得しま す。その形式は、次のとおりです。cosn://[secretld]: [secretKey]@examplebucket-125000000/。 2. org.apache.hadoop.fs.auth.SimpleCredentialProvider : core- site.xml設定ファイルからfs.cosn.userinfo.secretldと fs.cosn.userinfo.secretKeyを読み込み、SecretldとSecretKeyを 取得します。 3. org.apache.hadoop.fs.auth.EnvironmentVariableCredential Provider : システム環境変数のCOS_SECRET_IDと COS_SECRET_KEYから取得します。 4. org.apache.hadoop.fs.auth.SessionTokenCredentialProvider : 一 時キー形式を使用してアクセスします。 5. org.apache.hadoop.fs.auth.CVMInstanceCredentialProvider : Tencent CloudのCloud Virtual Machine(CVM)にバインドされた ロールを使用して、COSにアクセスするための一時キーを取得 します 6. org.apache.hadoop.fs.auth.CPMInstanceCredentialProvider : Tencent CloudのCloud Physical Machine(CPM)にバインドされ	この設定項目 以下の順序で 1. org.apache.ł 2. org.apache.ł 3. org.apache.hac 4. org.apache.ł 5. org.apache.ł 6. org.apache.ł 7. org.apache.ł



	<ul> <li>たロールを使用して、COSにアクセスするための一時キーを取得します。</li> <li>7.</li> <li>org.apache.hadoop.fs.auth.EMRInstanceCredentialsProvider:</li> <li>Tencent Cloud EMRインスタンスにバインドされたロールを使用して、COSにアクセスするための一時キーを取得します。</li> <li>8. org.apache.hadoop.fs.auth.RangerCredentialsProviderは、rangerを使用してキーを取得します。</li> </ul>	
fs.cosn.useHttps	COSバックエンドでのトランスポートプロトコルとして HTTPSを使用するかどうかを設定します。	true
fs.cosn.impl	FileSystem用のcosn実装クラス、 org.apache.hadoop.fs.CosFileSystemに固定されます。	なし
fs.AbstractFileSystem. cosn.impl	AbstractFileSystem用のcosn実装クラス、 org.apache.hadoop.fs.CosNに固定されます。	なし
fs.cosn.bucket.region	アクセスするバケットのリージョン情報を入力してください。 列挙値については、リージョンとアクセスドメイン名のリー ジョンの略称をご参照ください。 例えば、ap-beijing、ap-guangzhouなどです。元の設定: fs.cosn.userinfo.regionと互換性があります。	なし
fs.cosn.bucket. endpoint_suffix	接続するCOS endpointを指定します。この項目は入力必須項目 ではありません。パブリッククラウドCOSユーザーの場合、 上記のregionの設定を正しく入力するだけです。元の設定: fs.cosn.userinfo.endpoint_suffixと互換性があります。この項目 を有効にするために、設定時にfs.cosn.bucket.regionの設定項 目のendpointを削除してください。	なし
fs.cosn.tmp.dir	実際に存在するローカルディレクトリを設定してください。プ ロセス実行中に生成された一時ファイルは、一時的にここに置 かれます。	/tmp/hadoop_c
fs.cosn.upload. part.size	CosNファイルシステムの各blockのサイズです。マルチパート アップロードされた各part sizeのサイズでもあります。COSは マルチパートアップロードで最大10,000ブロックまでしかサ ポートできないため、使用できる最大1ファイルサイズを推定 する必要があります。 例えば、part sizeが8MBの場合、最大で78GBの単一ファイル のアップロードに対応します。part sizeは最大2GBまで対応可 能で、1ファイルあたり最大19TBまで対応できます。	8388608(8MI
fs.cosn. upload.buffer	<b>CosN</b> ファイルシステムがアップロード時に依存するバッファ のタイプです。現在、非ダイレクトメモリバッファ	mapped_disk

	<pre>(non_direct_memory) ダイレクトメモリバッファ(direct_memory)、ディスクマップ バッファ(mapped_disk)という3種類のバッファがサポートされ ています。非ダイレクトメモリバッファ エリアではJVMヒープメモリが使用されますが、ダイレクトメ モリバッファエリアではオフヒープメモリが使用されます。 ディスクマップバッファはメモリファイルマッピングをベース とするバッファエリアです。</pre>	
fs.cosn. upload.buffer.size	CosNファイルシステムがアップロードに依存するバッファの サイズです。-1に指定されている場合、バッファに制限がない ことを意味します。 バッファサイズが制限されていない場合、バッファタイプは mapped_diskである必要があります。指定されたサイズが0よ り大きい場合、この値は少なくとも1つのblockのサイズ以上で ある必要があります。元の設定:fs.cosn.buffer.sizeと互換性が あります。	-1
fs.cosn.block.size	CosNファイルシステムのblock sizeです。	134217728(12
fs.cosn. upload_thread_pool	ファイルが <b>COS</b> にストリーミングされるときの同時アップロー ドスレッドの数です。	10
fs.cosn. copy_thread_pool	ディレクトリのコピー操作中にファイルを同時にコピーおよび 削除するために使えるスレッドの数です。	3
fs.cosn. read.ahead.block.size	先読みブロックのサイズです。	1048576(1MB)
fs.cosn. read.ahead.queue.size	先読みキューの長さです。	8
fs.cosn.maxRetries	COSへのアクセス中にエラーが発生した場合の最大再試行回数 です。	200
fs.cosn.retry. interval.seconds	各再試行間の時間間隔です。	3
fs.cosn.	COSサーバー側の暗号化アルゴリズムを設定します。SSE-Cと	なし



server-side- encryption.algorithm	SSE-COSをサポートしており、デフォルトでは空で、暗号化 なしです。	
fs.cosn. server-side- encryption.key	COSのSSE-Cサーバー暗号化アルゴリズムを有効にする場合 は、SSE-Cのキーを設定する必要があります。 キーの形式は、base64でエンコードされたAES-256キーです。 デフォルトでは空で、暗号化なしです。	なし
fs.cosn. crc64.checksum.enabled	CRC64チェックを有効にするかどうかです。デフォルトでは有 効になっていません。現時点では、hadoop fs -checksumコマ ンドを使用してファイルのCRC64チェックサムを取得するこ とはできません。	false
fs.cosn. crc32c.checksum.enabled	CRC32Cチェックを有効にするかどうかです。デフォルトでは 有効になっていません。現時点では、hadoop fs -checksumコ マンドを使用してファイルのCRC32Cチェックサムを取得する ことはできません。有効にできるチェック方法は、crc32cか crc64のうち1つだけです。	false
fs.cosn.traffic.limit	アップロード帯域幅の制御オプション。819200 - 838860800 bits/s、デフォルト値は-1です。これは、デフォルトでは制限が ないことを意味します。	なし

### Hadoopの設定

\$HADOOP\_HOME/etc/hadoop/core-site.xml を変更して、COS関連のユーザーおよび実装クラスの情報を 追加します。次に例を示します。





```
2.org.apache.hadoop.fs.auth.SimpleCredentialProvider: Obtain the secret
        from fs.cosn.userinfo.secretId and fs.cosn.userinfo.secretKey in core-s
        3.org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider: Obta
        from system environment variables named COS_SECRET_ID and COS_SECRET_KE
        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>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/value>
    <description>Tencent Cloud Secret Id</description>
</property>
<property>
    <name>fs.cosn.userinfo.secretKey</name>
    <value>xxxxxxxxxxxxxxxxxxxxxxxxxxxx/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 onl
    </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> <name>fs.AbstractFileSystem.cosn.impl</name> <value>org.apache.hadoop.fs.CosN</value> <description>The implementation class of the CosN AbstractFileSystem.</description></property>
<property> <name>fs.cosn.tmp.dir</name> <value>/tmp/hadoop_cos</value> <description>Temporary files will be placed here.</description> </property>
<property></property>
<name>fs.cosn.upload.buffer</name>
<pre><value>mapped_disk</value> </pre> <pre></pre> <pre></pre> <pre></pre> <pre></pre> <pre>// Construct for a second sec</pre>
<property> <name>fs.cosn.upload.buffer.size</name> <value>134217728</value></property>
<pre><description>The total size of the upload buffer pool1 means unlimited.</description></pre>
<property> <name>fs.cosn.upload.part.size</name> </property>
<pre><description>Block size to use cosn filesysten, which is the part size for Considering the COS supports up to 10000 blocks, user should estimate the m For example, 8MB part size can allow writing a 78GB single file.</description></pre>
<pre><pre>chame&gt;fs cosh maxRetries</pre></pre>
<value>3</value>
<description></description>
The maximum number of retries for reading or writing files to COS, before we signal failure to the application.
<property>     (name) for even not my interval cocords ( / name)</property>
<pre><name>is.cosn.retry.interval.seconds</name> <value>3</value></pre>
<description>The number of seconds to sleep between each COS retry.</description>



<property></property>
<name>fs.cosn.server-side-encryption.algorithm</name>
<value></value>
<description>The server side encryption algorithm.</description>
<property></property>
<name>fs.cosn.server-side-encryption.key</name>
<value></value>
<description>The SSE-C server side encryption key.</description>

### </configuration>

このうちfs.defaultFSは、本番環境で設定することはお勧めしません。一部のテストシナリオ(例えば hivetestbenchなど)で使用する必要がある場合は、次の設定情報を追加できます。





```
<property>
<property>
<property>

<name>fs.defaultFS</name>

<value>cosn://examplebucket-125000000</value>

<description>

This option is not advice to config, this only used for some special t
</description>
```

```
</property>
```

サーバー側の暗号化

Hadoop-COSはサーバー側の暗号化をサポートしており、現在、COSホストキー方式(SSE-COS)とユーザー定義 キー方式(SSE-C)という種類の暗号化方式があります。Hadoop-COSの暗号化機能はデフォルトでは無効状態に なっており、ユーザーは以下の方法で設定して有効にすることができます。

### SSE-COSの暗号化

SSE-COSの暗号化とは、COSホストキーによるサーバー側の暗号化です。Tencent Cloud COSはマスターキーを ホストし、データを管理します。ユーザーはHadoop-COSを使用する場

合、 \$HADOOP\_HOME/etc/hadoop/core-site.xml ファイルに以下の設定を追加すると、SSE-COSの暗号化 を実装することができます。





<property>

#### SSE-Cの暗号化

SSE-Cの暗号化とは、ユーザー定義キーによるサーバー側の暗号化です。暗号化鍵はユーザーが提供し、COSは ユーザーがオブジェクトをアップロードする際に、ユーザーが提供した暗号化鍵を使用して、ユーザーのデータ をAES-256で暗号化します。Hadoop-COSを使用する場合、ユーザーは \$HADOOP\_HOME/etc/hadoop/coresite.xml ファイルに以下の設定を追加して、SSE-Cの暗号化を実装することができます。





```
<property>
<property=</pro>

</property=</pre>
</pro>
</pro>
</pro>
</pro>

</pro>
</pro>
```

#### 注意:

Hadoop-COSのSSE-Cサーバー側の暗号化は、COSのSSE-Cサーバー側の暗号化に依存しています。したがって Hadoop-COSは、ユーザーが提供する暗号化鍵を保存しません。また、COSのSSE-Cサーバー側の暗号化方式で は、ユーザーが提供した暗号化鍵は保存されず、暗号化鍵にランダムデータが追加されたHMAC値が保存され、 この値がオブジェクトにアクセスするためのユーザーのリクエストの認証に使われる点に注意する必要がありま す。COSは、ランダムデータのHMAC値を使用して暗号化鍵の値を推測したり、暗号化されたオブジェクトの内 容を復号したりすることはできません。そのため、ユーザーが暗号化鍵を紛失した場合、このオブジェクトを再 取得できなくなります。

Hadoop-COSがSSE-Cサーバー側暗号化アルゴリズムで設定されている場合、SSE-Cキーはfs.cosn.server-sideencryption.key設定項目で設定しなければなりません。キー形式は、base64でエンコードされたAES-256キーで す。

### 使用方法

### ユースケース

コマンド形式は、 hadoop fs -ls -R cosn://<BucketName-APPID>/<パス> 、または hadoop fs -ls -R /<パス> ( fs.defaultFS オプションは cosn://BucketName-APPID に設定する必要あり)です。次 の例では、examplebucket-1250000000という名のbucketを例として取り上げ、その後に具体的なパスを追加する ことができます。





```
hadoop fs -ls -R cosn://examplebucket-1250000000/
-rw-rw-rw- 1 root root 1087 2018-06-11 07:49 cosn://examplebucket-125000000
                                 0 1970-01-01 00:00 cosn://examplebucket-125000000
drwxrwxrwx - root root
                                 0 1970-01-01 00:00 cosn://examplebucket-125000000
drwxrwxrwx - root root
                             1087 2018-06-12 03:26 cosn://examplebucket-125000000
-rw-rw-rw- 1 root root
                             2386 2018-06-12 03:26 cosn://examplebucket-125000000
-rw-rw-rw- 1 root root
drwxrwxrwx - root root
                                 0 1970-01-01 00:00 cosn://examplebucket-125000000
-rw-rw-rw- 1 root root
                             1087 2018-06-11 07:32 cosn://examplebucket-125000000
-rw-rw-rw- 1 root root
                              2386 2018-06-11 07:29 cosn://examplebucket-125000000
```

MapReduceに付属しているwordcountを実行し、以下のコマンドを実行します。
### 注意:

以下のコマンドのhadoop-mapreduce-examples-2.7.2.jarは、バージョン2.7.2を例としています。バージョンが異なる場合は、対応するバージョン番号に変更してください。



bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount 実行に成功すると、以下のような統計情報が返されます。





```
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
```

## 🕗 Tencent Cloud

```
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
```

JavaコードによるCOSNへのアクセス





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.FileSystem; import org.apache.hadoop.fs.Path;

```
import java.io.IOException;
import java.net.URI;
import java.nio.ByteBuffer;
public class Demo {
   private static FileSystem initFS() throws IOException {
       Configuration conf = new Configuration();
       // COSNの設定項目については、https://cloud.tencent.com/document/product/436/688
       // 以下の設定は、入力必須項目です
       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.ofs.user.appid", "XXXXXXXXXXX");
       // その他の設定については、公式サイトのドキュメントhttps://cloud.tencent.com/docume
       // CRC64チェックを有効にするかどうかです。デフォルトでは有効になっていません。この時点で
       conf.set("fs.cosn.crc64.checksum.enabled", "true");
       String cosnUrl = "cosn://f4mxxxxxxx-125xxxxxxx";
       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
       // ファイルを作成します(存在する場合は上書きします)
       // if the parent dir does not exist, fs will create it!
       FSDataOutputStream out = fs.create(filePath, true);
       try {
          // ファイルに書き込みます
          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 IOExc
    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
            path, fileLen, accessTime, modifyTime, owner, group);
}
private static void getFileCheckSum(FileSystem fs, Path path) throws IOExceptio
    FileChecksum checksum = fs.getFileChecksum(path);
    System.out.printf("path %s, checkSumType: %s, checkSumCrcVal: %d\\n",
            path, checksum.getAlgorithmName(), ByteBuffer.wrap(checksum.getByte
}
private static void copyFileFromLocal (FileSystem fs, Path cosnPath, Path localP
    fs.copyFromLocalFile(localPath, cosnPath);
}
private static void copyFileToLocal (FileSystem fs, Path cosnPath, Path localPat
    fs.copyToLocalFile(cosnPath, localPath);
private static void renamePath (FileSystem fs, Path oldPath, Path newPath) throw
    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.getPat
    }
}
```

```
// 再帰的削除フラグは、ディレクトリを削除するために用いられます
// 再帰がfalseで、dirが空でない場合、操作は失敗します
private static void deleteFileOrDir(FileSystem fs, Path path, boolean recursive
    fs.delete(path, recursive);
}
private static void closeFileSystem(FileSystem fs) throws IOException {
   fs.close();
public static void main(String[] args) throws IOException {
   // ファイルの初期化
   FileSystem fs = initFS();
   // ファイルの作成
   Path cosnFilePath = new Path("/folder/exampleobject.txt");
   createFile(fs, cosnFilePath);
   // ファイルの読み取り
   readFile(fs, cosnFilePath);
   // ファイルまたはディレクトリの照会
   queryFileOrDirStatus(fs, cosnFilePath);
   // ファイルのチェックサムの取得
   getFileCheckSum(fs, cosnFilePath);
   // ローカルからファイルをコピーする
   Path localFilePath = new Path("file:///home/hadoop/ofs_demo/data/exampleobj
   copyFileFromLocal(fs, cosnFilePath, localFilePath);
   // ファイルをローカルで取得する
   Path localDownFilePath = new Path("file:///home/hadoop/ofs_demo/data/exampl
   copyFileToLocal(fs, cosnFilePath, localDownFilePath);
   listDirPath(fs, cosnFilePath);
   // リネーム
   mkdir(fs, new Path("/doc"));
   Path newPath = new Path("/doc/example.txt");
   renamePath(fs, cosnFilePath, newPath);
   // ファイルの削除
   deleteFileOrDir(fs, newPath, false);
   // ディレクトリの作成
   Path dirPath = new Path("/folder");
```



```
mkdir(fs, dirPath);

// ディレクトリにファイルを作成する
Path subFilePath = new Path("/folder/exampleobject.txt");
createFile(fs, subFilePath);

// ディレクトリのリストアップ
listDirPath(fs, dirPath);

// ディレクトリの削除
deleteFileOrDir(fs, dirPath, true);
deleteFileOrDir(fs, new Path("/doc"), true);

// ファイルシステムを閉じる
closeFileSystem(fs);
}
```

# よくあるご質問

}

Hadoopツールの使用に関するご質問は、Hadoopツールに関するよくあるご質問をご参照ください。

# COSDistCpツール

最終更新日:::2024-06-26 10:27:35

# 機能説明

COSDistCpは、MapReduceをベースとした分散ファイルコピーツールで、主にHDFSとCOS間のデータコピーに 使用され、主に次のような機能を備えています。 長さ、CRCチェックサムに応じて、ファイルの増分移行とデータチェックを行います ソースディレクトリ内のファイルに対して正規表現でのフィルタリングを行います ソースディレクトリ内のファイルを解凍し、想定の圧縮形式に変換します 正規表現に基づいてテキストファイルを集約します ソースファイルとソースディレクトリのユーザー、グループ、拡張属性および時間を保持します アラートとPrometheusモニタリングを設定します ファイルサイズ分布の統計 読み込み帯域幅の速度制限

# 使用環境

### システム環境

Linuxシステムをサポートしています。

#### ソフトウェア依存

Hadoop-2.6.0以降、Hadoop-COSプラグイン5.9.3以降。

### ダウンロードとインストール

#### COSDistCpjarパッケージの取得

Hadoop 2.x ユーザーは、cos-distcp-1.12-2.8.5.jarパッケージをダウンロードし、jarパッケージのMD5チェックサ

ムに基づき、ダウンロードしたjarパッケージが完全かどうかを確認します。

Hadoop 3.xユーザーは、cos-distcp-1.12-3.1.0.jarパッケージをダウンロードし、jarパッケージのMD5チェックサム に基づき、ダウンロードしたjarパッケージが完全かどうかを確認します。

### インストール説明

Hadoop環境において、Hadoop-COS をインストールすると、直接COSDistCpツールを実行できます。 Hadoop-COSプラグインがインストールおよび設定されていない環境のユーザーの場合、Hadoopのバージョンに 従って、対応するバージョンのCOSDistCp jar、Hadoop-COS jar、cos\_api-bundle jarパッケージをダウンロードし (jarパッケージのダウンロードアドレスは上記参照)、Hadoop-COS関連のパラメータを指定してコピータスク を実行します。jarパッケージアドレスにはローカルjarの存在するアドレスを入力する必要があります。



```
hadoop jar cos-distcp-${version}.jar \\
-libjars cos_api-bundle-${version}.jar,hadoop-cos-${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.CosN \\
```

## 🕗 Tencent Cloud

--src /data/warehouse \\
--dest cosn://examplebucket-125000000/warehouse

# 原理の説明

COSDistCpはMapReduceフレームワークをベースとして実装されています。これはマルチプロセス+マルチス レッドのアーキテクチャで、ファイルのコピー、データのチェック、圧縮、ファイル属性の保持およびコピーの再 試行を行うことができます。COSDistCpは、デフォルトでターゲットの同名の既存ファイルを上書きします。 ファイルの移行またはチェックに失敗した場合、対応するファイルのコピーが失敗し、移行に失敗したファイル情 報が一時ディレクトリに記録されます。ソースディレクトリにファイルが追加されたり、内容が変更されたりした 場合、skipModeやdiffModeモードを使用して、ファイルの長さまたはCRCチェック値を比較することによって、 データチェックサムとファイルの増分移行を行うことができます。

# パラメータの説明

hadoopユーザーの下でコマンド hadoop jar cos-distcp-\${version}.jar --help を使用すると、 COSDistCpでサポートされているパラメータオプションを確認することができます。ここで \${version} は バージョン番号です。以下は現在のバージョンのCOSDistCpのパラメータの説明です。

プロパティキー	説明	デフォルト値
help	COSDistCpでサポートされているパラメータ オプションを出力します 例:help	なし
src=LOCATION	コピーのソースディレクトリを指定します。 これはHDFSまたはCOSパスにすることがで きます 例:src=hdfs://user/logs/	なし
dest=LOCATION	コピーのターゲットディレクトリを指定しま す。これは、HDFSまたはCOSパスにするこ とができます	なし



	例:dest=cosn://examplebucket- 1250000000/user/logs	
srcPattern=PATTERN	正規表現を指定して、ソースディレクトリに あるファイルをフィルタリングします。 例:srcPattern='.*\\.log\$' 注意:記号 * がshellによって解釈されるの を避けるために、パラメータをシングル クォーテーションで囲む必要があります	なし
taskNumber=VALUE	コピープロセス数を指定します。例: taskNumber=10	10
workerNumber=VALUE	コピースレッド数を指定します。COSDistCp は、各コピープロセスでこのパラメータサイ ズのコピースレッドプールを作成します 例:workerNumber=4	4
filesPerMapper=VALUE	Mapper入力ファイル1ファイルあたりの行数 を指定します 例:filesPerMapper=10000	500000
groupBy=PATTERN	テキストファイルを集約するための正規表現 を指定します 例:groupBy='.*group-input/(\\d+)-(\\d+).*'	なし
targetSize=VALUE	ターゲットファイルのサイズをMB単位で指 定します。groupByとともに使用します 例:targetSize=10	なし
outputCodec=VALUE	<ul> <li>出力ファイルの圧縮方法を指定します。</li> <li>gzip、lzo、snappy、none、keepを選択できます。このうち、</li> <li>keepは、元のファイルの圧縮方法を維持します。</li> <li>noneは、ファイルの拡張子に従ってファイルを解凍します。</li> <li>例:outputCodec=gzip</li> <li>注意:ファイル/dir/test.gzipと/dir/test.gzがあり、出力形式をlzoに指定する場合は、最終的に1つのファイル/dir/test.lzoだけが維持されます</li> </ul>	keep
deleteOnSuccess	ソースファイルのターゲットディレクトリへ のコピーが成功したら、直ちにソースファイ ルを削除するよう指定します	false



	例:deleteOnSuccess, 注意:1.7以降のバージョンではこのパラ メータは提供されませんので、データ移行が 成功し、diffModeを使用してチェックを 行った後、ソースファイルシステムのデータ を削除することをお勧めします	
multipartUploadChunkSize=VALUE	Hadoop-COSプラグインがファイルをCOSに 転送するときのチャンクサイズを指定しま す。COSがサポートするチャンクの最大数は 10000で、ファイルサイズに応じてチャンク サイズをMB単位で調整できます。デフォル トは8MBです 例:multipartUploadChunkSize=20	8MB
cosServerSideEncryption	ファイルがCOSにアップロードされる際に、 暗号化・復号アルゴリズムとしてSSE-COS を使用するように指定します 例:cosServerSideEncryption	false
outputManifest=VALUE	コピーが完了した際、ターゲットディレクト リにそのコピーのターゲットファイルの情報 リスト(GZIP圧縮)が発行されるように指 定します 例:outputManifest=manifest.gz	なし
requirePreviousManifest	増分をコピーする場合は、 previousManifest=VALUEパラメータを指定 する必要があります 例:requirePreviousManifest	false
previousManifest=LOCATION	前回のコピーで生成されたターゲットファイ ルの情報 例: previousManifest=cosn://examplebucket- 125000000/big-data/manifest.gz	なし
copyFromManifest	previousManifest=LOCATIONとともに使用 すると、previousManifest内のファイルを ターゲットファイルシステムにコピーできま す 例:copyFromManifest	false
storageClass=VALUE	COSタイプを指定します。オプション値は、 STANDARD、STANDARD_IA、ARCHIVE、 DEEP_ARCHIVE、INTELLIGENT_TIERING です。サポートされているストレージタイプ	なし



	と概要については、ストレージタイプの概要 をご参照ください	
srcPrefixesFile=LOCATION	ローカルファイルを指定します。ファイルの 各行には、コピーする必要のあるソースディ レクトリが含まれます 例:srcPrefixesFile=file:///data/migrate- folders.txt	なし
skipMode=MODE	ファイルをコピーする前に、ソースファイル とターゲットファイルが同じかどうかを チェックし、同じであればスキップします。 none (チェックしない) 、length (長さ) 、 checksum (CRC値) 、length-mtime(長さ +mtime値)、length-checksum (長さ+CRC 値) が選択可能です 例:skipMode=length	length-checksum
checkMode=MODE	ファイルコピー完了時に、ソースファイルと ターゲットファイルが同じかどうかをチェッ クします。none(チェックしない)、length (長さ)、checksum(CRC値)、length- checksum(長さ+mtime値)、length- checksum(長さ+ CRC値)が選択可能です 例:checkMode=length-checksum	length-checksum
diffMode=MODE	ソースディレクトリとターゲットディレクト リの差分ファイルリストを指定して取得しま す。length(長さ)、checksum(CRC値)、 length-checksum(長さ+mtime値)、length- checksum(長さ+CRC値)が選択可能です 例:diffMode=length-checksum	なし
diffOutput=LOCATION	diffModeのHDFS出力ディレクトリを指定し ます。この出力ディレクトリは、必ず空であ る必要があります 例:diffOutput=/diff-output	なし
cosChecksumType=TYPE	Hadoop-COSプラグインが使用するCRCアル ゴリズムを指定します。オプション値は CRC32CとCRC64です 例:cosChecksumType=CRC32C	CRC32C
preserveStatus=VALUE	ソースファイルのuser、group、permission、 xattr、timestampsのメタ情報をターゲット ファイルにコピーするかどうかを指定しま す。オプション値は、ugpxt(user、group、	なし



	permission、xattr、timestampsの英語頭文 字)です 例:preserveStatus=ugpt	
ignoreSrcMiss	ファイルリストには存在しても、コピー時に は存在しないファイルを無視します	false
promGatewayAddress=VALUE	MapReduceタスクによって実行される CounterデータがプッシュされるPrometheus PushGatewayのアドレスとポートを指定しま す	なし
 promGatewayDeleteOnFinish=VALUE	タスクが完了したら、Prometheus PushGatewayのJobNameのメトリクスコレ クションを削除するよう指定します 例:promGatewayDeleteOnFinish=true	true
promGatewayJobName=VALUE	Prometheus PushGatewayに報告する JobNameを指定します 例:promGatewayJobName=cos-distcp- hive-backup	なし
promCollectInterval=VALUE	MapReduceタスクのCounter情報を収集する 間隔をms単位で指定します 例:promCollectInterval=5000	5000
promPort=VALUE	Prometheusメトリクスを外部に公開する サーバーポートを指定します 例:promPort=9028	なし
enableDynamicStrategy	タスクの動的割り当てポリシーを指定して、 移行速度の速いタスクがより多くのファイル を移行できるようにします。 注意:このモードには一定の制限がありま す。例えば、プロセスに異常が発生した場 合、タスクカウンターは不正確になります。 移行の完了後、diffModeを使用してデータ チェックを行ってください 例:enableDynamicStrategy	false
splitRatio=VALUE	Dynamic Strategyの分割比を指定します。 splitRatio値が大きいほど、タスクの粒度は小 さくなります 例:splitRatio=8	8
localTemp=VALUE	Dynamic Strategyによって生成されたタスク	/tmp



	情報ファイルが保存されているローカルフォ ルダを指定します 例:localTemp=/tmp	
taskFilesCopyThreadNum=VALUE	Dynamic Strategyのタスク情報ファイルを HDFSにコピーする際の同時実行性を指定し ます 例:taskFilesCopyThreadNum=32	32
statsRange=VALUE	統計の区間範囲を指定します 例: statsRange=0,1mb,10mb,100mb,1gb,10gb,inf	0,1mb,10mb,100mb,
printStatsOnly	移行するファイルサイズの分布情報のみ統計 が行われ、データは移行されません 例:printStatsOnly	なし
bandWidth	移行する各ファイルの読み込み帯域幅を制限 します。単位はMB/s、デフォルトは-1で、読 み込み帯域幅は制限しません。 例:bandWidth=10	なし
jobName	移行タスクの名前を指定します。 例:jobName=cosdistcp-to-warehouse	なし
compareWithCompatibleSuffix	skipModeとdiffModeパラメータを使用す る場合、ソースファイルの拡張子gzipをgzに 変換するかどうか、lzopファイルの拡張子を lzoに変換するかどうかを判断します。 例:compareWithCompatibleSuffix	なし
delete	ソースディレクトリとターゲットディレクト リのファイルの一致性を保証します。ソース ディレクトリにはなく、ターゲットディレク トリにはあるファイルを独立したtrashディレ クトリ下に移動させ、同時にファイルリスト を生成します。 注意:diffModeパラメータは同時に使用で きません	なし
deleteOutput	deleteのHDFS出力ディレクトリを指定しま す。このディレクトリは必ず空でなければな りません 例:deleteOutput=/dele-output	なし

# ユースケース

## helpオプションの確認

パラメータ --help によりコマンドを実行し、COSDistCpでサポートされているパラメータを確認します。次に 例を示します。



hadoop jar cos-distcp-\${version}.jar --help

上記コマンドにおいて \${version} はCOSDistCpのバージョン番号です。例えば、バージョン1.0のCOSDistCp jarパッケージ名はcos-distcp-1.0.jarとなります。

### 移行するファイルサイズ分布に関する情報の統計を行います

パラメータ --printStatsOnly と --statsRange=VALUE によりコマンドを実行すると、移行するファイル のサイズ分布に関する情報が出力されます。



hadoop jar cos-distcp-\${version}.jar --src /wookie/data --dest cosn://examplebucket Copy File Distribution Statistics:



Total File Count: 4	1		
Total File Size: 11	90133760		
SizeRange	TotalCount	TotalSize	
OMB ~ 1MB	0(0.00%)	0(0.00%)	
1MB ~ 10MB	1(25.00%)	1048576(0.09%)	
10MB ~ 100MB	1(25.00%)	10485760(0.88%)	
100MB ~ 1024MB	1(25.00%)	104857600(8.81%)	
1024MB ~ 10240MB	1(25.00%)	1073741824(90.22%)	
10240MB ~ LONG_MA	AX  0(0.00%)	0(0.00%)	

# 移行するファイルのソースディレクトリとターゲットディレクトリを指定します

パラメータ --src と --dest によりコマンドを実行します。次に例を示します。





hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebuc

COSDistCpは、コピーが失敗したファイルをデフォルトで5回再試行します。それでも失敗した場合、失敗した ファイル情報を/tmp/\${randomUUID}/output/failed/ディレクトリに書き込みます。ここで\${randomUUID}は、ラン ダム文字列です。失敗したファイル情報を記録した後、COSDistcpは残りのファイルの移行を続行しますが、一部 のファイルが失敗したために移行タスクが失敗することはありません。移行タスクが完了すると、COSDistcpは カウンター情報(タスクがマシンへ送信され、MapReduceタスクの提出側INFOログの出力が設定されたことを確 認してください)を出力し、移行に失敗したファイルが存在するかどうかを判断します。存在した場合は、タスク を送信したクライアントに異常が報告されます。 出力ファイルには、次のようなソースファイル情報が含まれます。

1. ソースファイルのリストに存在しますが、コピー時にソースファイルが存在しないため、SRC\_MISSと記録されます

2. その他の原因によるコピーの失敗は、すべてCOPY\_FAILEDと記録されます

copyコマンドを再実行して、増分移行を行うことができます。次のコマンドによって、MapReduceタスクのログ を取得し、ファイルコピーに失敗した原因を特定します。ここでapplication\_1610615435237\_0021は、アプリ ケーションIDです。



yarn logs -applicationId application\_1610615435237\_0021 > application\_1610615435237



### Countersの確認

コピータスクの終了時に、ファイルのコピーに関する統計情報が出力されます。関連するカウンターは次のとおり です。



CosDistCp Counters BYTES\_EXPECTED=10198247 BYTES\_SKIPPED=10196880 FILES\_COPIED=1 FILES\_EXPECTED=7 FILES\_FAILED=1 FILES\_SKIPPED=5



ファイルコピーの統計情報は次のとおりです。

統計項目	説明
BYTES_EXPECTED	ソースディレクトリの統計に基づいてコピーが必要なファイルの合計サイズ。単 位:バイト
FILES_EXPECTED	ディレクトリファイルを含む、ソースディレクトリの統計に基づいてコピーが必要 なファイル数
BYTES_SKIPPED	長さまたはチェックサム値が等しく、コピーされないファイルサイズの合計。単 位:バイト
FILES_SKIPPED	長さまたはチェックサム値が等しく、コピーされないソースファイル数
FILES_COPIED	コピーに成功したソースファイル数
FILES_FAILED	コピーに失敗したソースファイル数
FOLDERS_COPIED	コピーに成功したディレクトリ数
FOLDERS_SKIPPED	スキップしたディレクトリ数

## コピープロセス数と各コピープロセスのコピースレッド数を指定します

パラメータ --taskNumber と --workersNumber によってコマンドを実行します。COSDistCpはマルチプロ セス+マルチスレッドのコピーアーキテクチャを使用しており、次のことが可能です。

--taskNumber でコピーするプロセス数を指定します

--workerNumber で各コピープロセスにおけるコピースレッド数を指定します





hadoop jar cos-distcp-\${version}.jar --src /data/warehouse/ --dest cosn://examplebu

### 同じチェックサムを持つファイルをスキップし、増分移行を実行します

パラメータ --skipMode によりコマンドを実行します。同じ長さとチェックサムを持つソースとターゲット ファイルのコピーをスキップします。デフォルト値はlength-checksumです。





hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebu

--skipMode オプションは、ファイルをコピーする前に、ソースファイルとターゲットファイルが同じかどう かをチェックして、同じであればスキップします。none(チェックしない)、length(長さ)、checksum(CRC 値)、length-checksum(長さ+CRC値)が選択可能です。

ソースとターゲットのファイルシステムのチェックサムアルゴリズムが異なる場合、ソースファイルを読み込ん で新しいチェックサムを計算します。ソースがHDFSの場合、HDFSソースがCOMPOSITE-CRC32Cチェックサム アルゴリズムをサポートしているかどうかは、次のように確認することができます。





hadoop fs -Ddfs.checksum.combine.mode=COMPOSITE\_CRC -checksum /data/test.txt /data/test.txt COMPOSITE-CRC32C 6a732798

## 移行完了後のデータチェックおよび増分移行

パラメータ --diffMode と --diffOutput によりコマンドを実行します。

--diffMode のオプション値はlengthとlength-checksumです。

--diffMode=length は、ファイルサイズが同じかどうかによって、差分ファイルのリストを取得することを 表します。 --diffMode=length-checksum 、ファイルサイズとCRCチェックサムが同じかどうかによって、差分ファ イルのリストを取得することを表します。

--diffOutput は、diff操作の出力ディレクトリを指定します。

ターゲットファイルシステムがCOSで、ソースファイルシステムのCRCアルゴリズムが異なる場合、COSDistCp はソースファイルをプルしてターゲットファイルシステムのCRCを計算し、同じCRCアルゴリズム値を比較しま す。以下の例では、移行が完了した後、--diffModeパラメータを使用して、ファイルサイズとCRC値に基づきソー スファイルとターゲットファイルが同じであることをチェックしています。



hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebuc

上記のコマンドの実行が成功すると、ソースファイルシステムのファイルリストを基準にカウンター情報(タス クがマシンへ送信され、MapReduceタスクの提出側INFOログの出力が設定されたことを確認してください)が出 力されます。このカウンター情報を基に、ソースとターゲットが同じであるか分析することができます。カウン ター情報の説明は、次のとおりです。

1. ソースファイルとターゲットファイルが同じ場合、SUCCESSと記録されます

2. ターゲットファイルが存在しない場合、DEST\_MISSと記録されます

3. ソースディレクトリのリストに存在しますが、チェック時にソースファイルが存在しないため、SRC\_MISSと 記録されます

4. ソースファイルとターゲットファイルのサイズが異なる場合、LENGTH\_DIFFと記録されます

5. ソースファイルとターゲットファイルのCRCアルゴリズムの値が異なる場合、CHECKSUM\_DIFFと記録されま す

6. 読み込み権限が不十分であるなどの要因により、diff操作が失敗した場合、DIFF\_FAILEDと記録されます

7. ソースがディレクトリ、ターゲットがファイルの場合、TYPE\_DIFFと記録されます

また、COSDistcpは、HDFSの /tmp/diff-output/failed ディレクトリ(1.0.5および以前のバージョンでは/tmp/diff-output)に差分ファイルリストを発行します。次のコマンドを使用して、SRC\_MISS以外の差分ファイルリストを取得することができます。





hadoop fs -getmerge /tmp/diff-output/failed diff-manifest
grep -v '"comment":"SRC\_MISS"' diff-manifest |gzip > diff-manifest.gz

次のコマンドを実行し、差分ファイルリストに基づいて増分移行を実行します。





hadoop jar cos-distcp-\${version}.jar --taskNumber=20 --src /data/warehouse --dest

増分移行が完了したら、再度--diffModeパラメータ付きのコマンドを実行し、ファイルが同じであることをチェックします。

### ソースファイルとターゲットファイルのCRCが同じであるかチェックします

パラメータ --checkMode によりコマンドを実行します。ファイルのコピーが完了したときに、ソースファイル とターゲットファイルの長さとチェックサムが同じであることをチェックします。デフォルト値はlengthchecksumです。 非COSファイルシステムからCOSに同期する場合やソースのCRCアルゴリズムがHadoop-COSのCRCアルゴリズ ムと一致しない場合、コピー中にCRCが計算され、コピー完了時にターゲットCOSファイルのCRCを取得し、算 出されたソースファイルのCRCと照合します。



hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://exampleb

### 注意:

--groupByが指定されておらず、outputCodecがデフォルト値の場合に有効になります。

単一ファイルの読み込み帯域幅を制限します

パラメータ --bandWidth によりコマンドを実行します。単位はMBです。移行した各ファイルの読み込み帯域 幅を10MB/sに制限します。次に例を示します。



hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebu

### 複数ディレクトリの同期

ローカルファイル(srcPrefixes.txtなど)を新規作成し、このファイルに移行が必要なディレクトリの絶対パスを 追加します。これらのディレクトリ間に親子関係は必要なく、追加すれば、次の例に示すように、catコマンドで 確認することができます。





cat srcPrefixes.txt
/data/warehouse/20181121/
/data/warehouse/20181122/

--srcPrefixesFile パラメータを使用してこのファイルを指定し、移行コマンドを実行します。





hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --srcPrefixesFile file

### 入力ファイルの正規表現でのフィルタリング

パラメータ --srcPattern でコマンドを実行すると、 /data/warehouse/ ディレクトリにある.logで終わる ログファイルのみを同期します。次に例を示します。





hadoop jar cos-distcp-\${version}.jar --src /data/warehouse/ --dest cosn://exampleb

.tempまたは.tmpで終わるファイルは移行されません。





hadoop jar cos-distcp-\${version}.jar --src /data/warehouse/ --dest cosn://exampleb

### Hadoop-COSのファイルチェックとタイプの指定

パラメータ --cosChecksumType によりコマンドを実行します。デフォルトはCRC32Cで、オプションは CRC32CとCRC64です。




hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebu

## COSファイルのストレージタイプの指定

パラメータ --storageClass によりコマンドを実行します。次に例を示します。





hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebuc

## ターゲットファイルの圧縮タイプの指定

パラメータ --outputCodec によりコマンドを実行します。このパラメータを使用すると、HDFS内のデータを リアルタイムに圧縮してCOSにバックアップし、ストレージコストを節約することができます。パラメータのオ プション値は、keep、none、gzip、lzop、snappyです。noneオプションはターゲットファイルを非圧縮状態で保 存し、keepはオリジナルファイルの圧縮状態を維持します。次に例を示します。





hadoop jar cos-distcp-\${version}.jar --src /data/warehouse/logs --dest cosn://examp

#### 注意:

keepオプションを除いて、まずファイルを解凍し、それからターゲットの圧縮タイプに変換します。そのため keepオプション以外では、圧縮パラメータなどの不整合により、ターゲットファイルがソースファイルと一致し ないことがありますが、解凍後のファイルは同じになります。--groupByが指定されておらず、--outputCodecがデ フォルトの場合は、--skipModeで増分の移行を行い、--checkModeでデータチェックを行うことができます。

#### ソースファイルの削除

パラメータ --deleteOnSuccess によりコマンドを実行し、 /data/warehouse ディレクトリ内のファイル をHDFSからCOSに同期してから、直ちにソースディレクトリ内の対応するファイルを削除します。



hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebuc

#### 注意:

このオプションを指定すると、ファイルが移行されるたびに対応するソースファイルが直ちに削除されます。全体の移行が完了した後、ソースファイルを削除する必要はありませんので、ご注意ください。1.7以降のバージョンでは、このパラメータは提供されません。

## ターゲットリストファイルの発行と最終リスト出力ファイルの指定

パラメータ --outputManifest と --previousManifest によりコマンドを実行します。 --outputManifest このオプションは、まずローカルでgzip圧縮されたmanifest.gzを発行し、移行が成功した ときに --dest で指定されたディレクトリに移動させます。

--previousManifest は前回の --outputManifest 出力ファイルを指定します。COSDistCpは、同じサイズのファイルをスキップします。



hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebu

注意:

上記のコマンドの増分移行では、サイズが変更されたファイルのみを同期できますが、ファイル内容が変更され たファイルは同期できません。ファイルの内容が変更される可能性がある場合は、--diffModeの使用例を参照し て、ファイルのCRCに基づいて変更されたファイルのリストを決定してください。

## 移行タスクの割り当てポリシーを動的割り当てに指定

ごく少量の非常に大きなファイル、多数の小さなファイル、移行用マシンの負荷が変化する場合などのように、 ファイルサイズが極端に分かれている場合は、enableDynamicStrategy`を使用して、タスクの動的割り当てポリ シーを有効にすると、実行速度の速いタスクがより多くのファイルを移行できるようになり、タスクの実行時間 が短縮されます。



hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://exampl

#### 移行完了後の移行データのチェックを行います。



hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebuc

#### 注意:

このモードには一定の制限があります。例えば、プロセス例外が発生した場合、タスクカウンターはカウントが 不正確になる可能性があります。移行完了後に--diffModeを使用してデータをチェックしてください。

ファイルのメタ情報のコピー

パラメータ --preserveStatus によりコマンドを実行し、ソースファイルまたはソースディレクトリのuser、 group、permission、timestamps (modification timeとaccess time)をターゲットファイルまたはターゲットディレ クトリにコピーします。このパラメータは、HDFSからCHDFSへファイルをコピーするときに有効になります。 例:



hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebuc

Prometheusモニタリングの設定

Yarnの管理画面では、すでに移行が完了したファイル数、バイト数などの情報を含む、COSDistcp移行タスクの カウンターを確認することができます。移行タスクカウンターの曲線の推移をより見やすくするためには、 prometheus.ymlを設定してクロールタスクを追加するといった簡単な設定だけで、Prometheus + Grafanaモニタリ ングシステムでCOSDistcp移行タスクのカウンターを表示させることができます。



```
- job_name: 'cos-distcp-hive-backup'
    static_configs:
        - targets: ['172.16.16.139:9028']
```



パラメータ --promPort=VALUE によりコマンドを実行し、現在のMapReduceタスクのカウンターを外部に公 開します



hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebu 事例のGrafana Dashboardをダウンロードしてインポートすると、Grafanaは次のように表示されます。



## ファイルコピー失敗時のアラート

パラメータ --completionCallbackClass でコールバッククラスパスを指定してコマンドを実行します。 COSDistCpはコピータスクが完了すると、収集したタスク情報をパラメータとしてコールバック関数を実行しま す。ユーザー定義のコールバック関数については、以下のインターフェースを実装する必要がありますので、コー ルバックのサンプルコードのダウンロードに移動してください。





```
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, S
/**
 * @description: init callback config before execute
 */
void init() throws Exception;
}
```

COSDistCpは、内部でBCMのアラートを統合し、タスクに異常が発生したり、ファイルコピーに失敗したりした 場合にアラートを実行します。



```
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 \
-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-125000000/data/warehouse/ \\
--checkMode=checksum \\
--completionCallbackClass=com.qcloud.cos.distcp.DefaultTaskCompletionCallback
```

上記コマンドのalarmPolicyIdは、BCMアラートポリシーであり、BCMコンソールで作成と設定ができます(ア ラート管理>アラート設定>カスタムメッセージ)。

# よくあるご質問

COSDistcpを使用したHDFSデータの移行にはどのような段階があり、どのように移行パフォーマンスを調整してデータの正確性を確保すればよいですか。

COSDistcpでは各ファイルの移行完了ごとに、checkModeに基づいて、移行したファイルに対してチェックが行われます。





hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebuc

また、移行が完了したら、以下のコマンドを実行し、ソースとターゲットの差分ファイルリストを確認すること ができます。





hadoop jar cos-distcp-\${version}.jar --src /data/warehouse --dest cosn://examplebuc

## Hadoop-COSが設定されていない環境で、COSDistCpを実行するにはどうすればいいですか。

Hadoop-COSプラグインが設定されていない環境のユーザーの場合、Hadoopのバージョンに従って、対応する バージョンのCOSDistCp jarパッケージをダウンロードし、Hadoop-COS関連のパラメータを指定してコピータス クを実行します。





```
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
```

## コピー結果に、一部のファイルのコピーに失敗したと表示されます。どうすればよいですか。

COSDistCpは、ファイルコピー中に発生した IOExceptionに対して5回再試行します。5回目のコピーも失敗した場合は、失敗したファイル情報を /tmp/\${randomUUID}/output/failed/ ディレクトリに書き込みます。ここで、\${randomUUID}はランダムな文字列です。コピーが失敗する一般的な原因としては、次のようなものがあります。

1. コピーリストにソースファイルは存在しますが、コピー時にソースファイルが存在しないため、SRC\_MISSと 記録されます。

2. タスクを開始したユーザーが、ソースファイルの読み込みやターゲットファイルへの書き込みの権限を持って いないことなどが原因で、COPY\_FAILEDと記録されます。

ログ情報にソースファイルが存在しないことが記録されており、ソースファイルを実際に無視できる場合は、次のコマンドを使用して、SRC\_MISS以外の差分ファイルのリストを取得することができます。





hadoop fs -getmerge /tmp/\${randomUUID}/output/failed/ failed-manifest
grep -v '"comment":"SRC\_MISS"' failed-manifest |gzip > failed-manifest.gz

SRC\_MISS以外に失敗したファイルがある場合は、 /tmp/\${randomUUID}/output/logs/ ディレクトリに まとめられた例外ログ情報と、yarnアプリケーションなどのプルアプリケーションログに基づいて原因を診断する ことができます。次のコマンドを使用できます。





yarn logs -applicationId application\_1610615435237\_0021 > application\_1610615435237

ここでapplication\_1610615435237\_0021は、アプリケーションIDです。

## COSDistCpは、ネットワークの異常などが発生した場合、不完全なファイルのコピーは生成されま すか。

COSDistCpは、ネットワークの異常、ソースファイルの欠落、権限不足などの場合、ターゲット側とソース側で 同じサイズのファイルを発行することはできません。 COSDistCpのバージョンが1.5以下の場合、COSDistCpはターゲット側で生成されたファイルの削除を試みます。 削除に失敗した場合は、コピータスクを再実行してこれらのファイルを上書きするか、不完全なファイルを手動で 削除する必要があります。

COSDistCp 1.5以降のバージョンで、実行環境のHadoop COSプラグインバージョンが5.9.3以降の場合、

COSDistCpはCOSへのコピーが失敗すると、abortインターフェースを呼び出してアップロード中のリクエストを 終了させるようにします。そのため異常が発生しても、不完全なファイルは発行されません。

COSDistCp 1.5以降のバージョンで、実行環境のHadoop COSプラグインバージョンが5.9.3未満の場合は、5.9.3以降のバージョンにアップグレードすることをお勧めします。

COS以外のターゲット側では、COSDistCpはターゲットファイルの削除を試みます。

## COSバケットに、目に見えず完了していないアップロードファイルがあり、ストレージ容量を占有 しています。どうすればよいですか。

マシンの異常やプロセスのKillなどの要因により、COSバケット内でフラグメントファイルがストレージ容量を占 有する場合があります。この場合、公式ウェブサイトのライフサイクルドキュメントを参照して、フラグメントの 削除ルールを設定し、クリーンアップを行うことができます。

## 移行プロセスでは、メモリオーバーフローとタスクタイムアウトが発生した場合、どのようにパラ メータチューニングを行いますか。

移行プロセスでは、COSDistcpとCOSへのアクセスとCHDFSのツールは、 自身のロジックに基づき、メモリの一 部を占有します。メモリオーバーフローとタスクタイムアウトを防ぐため、下の例のようにしてMapReduceタス クのパラメータ調整をすることができます。





hadoop jar cos-distcp-\${version}.jar -Dmapreduce.task.timeout=18000 -Dmapreduce.red

そのうち、タスクのタイムアウト時間mapreduce.task.timeoutを18000秒に調整することで、大容量のファイルを コピーするとき、タスクタイムアウトの発生を防ぐことができます。Reduceプロセスのメモリ容量 mapreduce.reduce.memory.mbサイズを8GBに調整することで、メモリオーバーフローを防ぐことができます。

## 専用回線の移行で、どのように移行タスクの移行帯域幅をコントロールしますか。

COSDistcpでは移行の総帯域幅制限計算式は、taskNumber \* workerNumber \* bandWidthです。workerNumberを1 に設定し、パラメータtaskNumberによる移行の同時実行数のコントロール、およびパラメータbandWidthによる



単一の同時実行帯域幅のコントロールができます。

# Hadoop-cos-DistChecker $\mathcal{V} - \mathcal{V}$

最終更新日:::2024-06-26 10:27:35

# 機能説明

Hadoop-cos-DistCheckerは、移行したディレクトリの整合性を検証するためのツールです。ユーザーが hadoop distcp コマンドを使用してHDFSからCOSにデータを移行した後、MapReduceの並列機能に基づいて、 Hadoop-cos-DistCheckerツールは**ソースディレクトリ**と**ターゲットディレクトリ**間の検証比較をすばやく実行す ることができます。

## 使用環境

Hadoop-cos v5.8.2以降。詳細については、hadoop-cos releaseをご参照ください。

Hadooop MapReduceの実行環境。

### 注意:

自分でHadoopクラスターをビルドする場合、CRC64チェックデジットの取得をサポートするため、Hadoop-cos 依存関係を最新バージョン(GitHub releaseバージョンは5.8.2以降)にする必要があります。

Tencent Cloud EMRスイートを使用している場合、2020年5月8日以降に作成されたクラスターには、このHadoopcosバージョンのみが含まれます。それ以前に作成されたクラスターについては、チケットを提出してお問い合わ せください。

## 利用説明

Hadoop-cos-distcheckerは、Hadoop-cos(CosNファイルシステム)内のファイルのCRC64チェックサムを取得す る必要があるため、ツールを実行する前に、設定項目fs.cosn.crc64.checksum.enabledをtrueに設定して、HadoopcosファイルのCRC64チェックサムの取得をサポートする必要があります。ツールの実行が完了したら、このオプ ションをfalseに戻し、CRC64チェックサムの取得を無効にします。

#### 注意:

Hadoop-COSでサポートされているCRC64チェックサムはHDFSファイルシステムのCRC32Cチェックサムと互換 性がないため、このツールを使用した後は、必ず上記の設定項目を無効状態に戻してください。戻さない場合、 Hadoop-cosがファイルシステムのgetFileChecksumインターフェースが呼び出されるシナリオによっては、実行が 失敗する恐れがあります。

## パラメータの概念

**ソースファイルリスト**:ソースファイルパスリストとは、ユーザーが以下のコマンドを実行してエクスポートしたチェック対象のサブディレクトリとファイルのリストです。



hadoop fs -ls -R hdfs://host:port/{source\_dir} | awk '{print \$8}' > check\_list.txt 形式の例は次のとおりです。





/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

**ソースディレクトリ**:ソースファイルのリストがあるディレクトリを指します。このディレクトリは通常、データ を移行する際の distcp コマンドのソースパスにもなります。次に示すよう

に、 hdfs://host:port/source\_dir はソースディレクトリです。



hadoop distcp hdfs://host:port/source\_dir cosn://examplebucket-appid/dest\_dir

さらに、このパスは**ソースファイルパスリスト**の共通の親ディレクトリでもあります。例えば、上記のソース ファイルリスト共通の親ディレクトリは、 /benchmarks です。 **ターゲットディレクトリ**:比較するターゲットディレクトリです。

## コマンドライン形式

Hadoop-cos-DistCheckerは、MapReduceジョブプログラムです。MapReduceジョブの送信プロセスに従えばOK です。



hadoop jar hadoop-cos-distchecker-2.8.5-1.0-SNAPSHOT.jar com.qcloud.cos.hadoop.dist

## 説明:

Optional parametersは、Hadoopのオプションパラメータを示します。

使用手順

以下では、ツールの使用手順を紹介する例として、

hdfs://10.0.0.3:9000/benchmarks と cosn://hdfs-test-1250000000/benchmarks のチェックを 例として取り上げます。

まず、次のコマンドを実行します。



hadoop fs -ls -R hdfs://10.0.0.3:9000/benchmarks | awk '{print \$8}' > check\_list.tx

@VM\_38\_97\_centos ~]\$ hadoop fs -ls -R hdfs://10.0.0.3:9000/benchmarks | awk '{print \$8}' > check\_list.txt

次のように、チェックするソースパスをcheck\_list.txtファイルにエクスポートします。このファイルには、ソース ファイルのパスリストが格納されています。

次に、check\_list.txtをHDFSに入れて、次のように実行します。





hadoop fs -put check\_list.txt hdfs://10.0.0.3:9000/

[imin_ju@VM_38_97_centos ~]\$ hadoop fs [imin_ju@VM_38_97_centos ~]\$ hadoop fs	-put check_list.txt hdfs://10.0.0.3:9000/ -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



最後に、Hadoop-cos-DistCheckerを実行し、 hdfs://10.0.0.3:9000/benchmarks と cosn://hdfstest-125000000/benchmarks を比較して、出力結果を cosn://hdfs-test-125000000/check\_result のパスに保存します。コマンド形式は次のとおりです。



hadoop jar hadoop-cos-distchecker-2.8.5-1.0-SNAPSHOT.jar com.qcloud.cos.hadoop.dist



Hadoop-cos-DistCheckerは、ソースファイルリストとソースディレクトリを読み込み、MapReduceジョブを実行 して分散チェックを行い、最終的なチェックレポートを cosn://examplebucket-appid/check\_result パ スに出力します。

🖕 🖕 j@VM_38_97_centos ~]\$ hadoop fs -ls -R cosn://hdfs-test-125🚛 🚛/check_result	
20/02/21 17:37:54 INFO fs.BufferPool: The type of the upload buffer pool is [MAPPED_DISK]. Buffer size:[134217728]	
20/02/21 17:37:54 INFO fs.BufferPool: Initialize the MAPPED_DISK buffer pool. size: 16	
-rw-rw-rw- 1 ia ia 0 2020-02-21 17:36 cosn://hdfs-test-1252******/check_result/_SUCCESS	
-rw-rw-rw- l ia 💶 ia 👥 1690 2020-02-21 17:36 cosn://hdfs-test-125 🛄 灯 / check_result/part-m-00000	
20/02/21 17:37:55 INFO fs.BufferPool: Close a buffer pool instance.	
20/02/21 17:37:55 INFO fs.BufferPool: Begin to release the buffers.	

チェックレポートは次のとおりです。





```
hdfs://10.0.0.3:9000/benchmarks/TestDFSI0
                                             hdfs://10.0.0.3:9000/benchmarks/Tes
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_control hdfs://10.0.0.3:9000/benchm
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_control/in_file_test_io_0 hdfs://10.0
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_control/in_file_test_io_1 hdfs://10.0
hdfs://10.0.0.3:9000/benchmarks/TestDFSI0/io_data
                                                 hdfs://10.0.0.3:9000/benchm
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_data/test_io_0
                                                            hdfs://10.0.0.3:900
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_data/test_io_1
                                                             hdfs://10.0.0.3:900
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_write hdfs://10.0.0.3:9000/benchm
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_write/_SUCCESS
                                                             hdfs://10.0.3:900
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_write/part-00000
                                                             hdfs://10.0.0.3:900
```

# チェックレポート形式

チェックレポートは、次のような形式で表示されます。



check\_list.txtのソースファイルパス、ソースファイルの絶対パス、ターゲットファイルの絶対パス、Chec

これらのチェック結果は、以下7つのカテゴリーに分類されます。

SUCCESS:ソースファイルとターゲットファイルの両方が存在し、整合性が取れていることを示します。 MISMATCH:ソースファイルとターゲットファイルの両方が存在しますが、整合性が取れていないことを示しま す。 UNCONFIRM:ソースファイルとターゲットファイルの整合性を確認できません。この状態は、COS上のファイルがCRC64チェックデジット機能がオンラインになる前に存在していたファイルである可能性があり、そのCRC64チェックサムを取得できないことが主な原因です。

UNCHECKED:チェックされていません。この状態は、ソースファイルが読み込めないか、ソースファイルの checksum値を取得できないことが主な原因です。

SOURCE\_FILE\_MISSING:ソースファイルが存在しません。

TARGET\_FILE\_MISSING:ターゲットファイルが存在しません。

TARGET\_FILESYSTEM\_ERROR:ターゲットファイルシステムがCosNファイルシステムではありません。

# よくあるご質問

### チェックレポートのCRC64値が負の値になるのはなぜですか。

CRC64の値が20ビットの値である可能性があり、Java long型の表現できる範囲を超えていますが、基になるバイトは同じであるため、long型をダンプすると負の数が表示されます。
# HDFS TO COSツール

最終更新日:::2024-06-26 10:27:35

# 機能説明

HDFS TO COSツールは、HDFSからTencent Cloud COSにデータをコピーするときに使います。|

### 使用環境

### システム環境

LinuxまたはWindowsシステム。

### ソフトウェア依存

JDK 1.7または1.8。

### インストールと設定

環境のインストールと設定の詳細については、Javaのインストールと設定をご参照ください。

# 設定方法

1. Hadoop-2.7.2以降のバージョンをインストールします。具体的なインストール手順については、Hadoopのイン ストールとテストをご参照ください。

2. GitHubからHDFS TO COSツールをダウンロードし、解凍してください。

3. 同期するHDFSクラスターのcore-site.xmlをconfフォルダにコピーします。このうちcore-site.xmlには、

### NameNodeの設定情報が含まれています。

4. 設定ファイルcos\_info.confを編集して、バケット(Bucket)、リージョン(Region)およびAPIキー情報を保存しま す。このうちバケット名は、ユーザー定義の文字列と、システムが発行したAPPID文字列をハイフンで連結するこ とで構成されます(例:examplebucket-125000000)。

5. コマンドラインパラメータで設定ファイルの場所を指定します。デフォルトの場所はconf/cos\_info.confです。 注意:

コマンドラインパラメータのパラメータが設定ファイルと重複している場合は、コマンドラインが優先されます。

# 利用方法



### 説明:

以下は、Linuxを例とした使用方法です。

### ヘルプの確認



./hdfs\_to\_cos\_cmd -h

### ファイルのコピー

HDFSからCOSにコピーします。COSにすでに同名のファイルが存在する場合、元のファイルは上書きされます。





./hdfs\_to\_cos\_cmd --hdfs\_path=/tmp/hive --cos\_path=/hdfs/20170224/

HDFSからCOSにコピーします。COSにすでに同名で同じ長さのファイルが存在する場合、アップロードは無視されます(1回目のコピー後、再度コピーする場合に適用されます)。





./hdfs\_to\_cos\_cmd --hdfs\_path=/tmp/hive --cos\_path=/hdfs/20170224/ -skip\_if\_len\_mat

Hadoop上でファイルサマリーを計算するとオーバーヘッドが大きくなるため、ここでは長さのみを判断しています。

HDFSからCOSにコピーします。HDFSにHarディレクトリ(Hadoop Archiveアーカイブファイル)が存在する場合、--decompress\_harパラメータを指定することでharファイルを自動的に解凍できます。





./hdfs\_to\_cos\_cmd --decompress\_har --hdfs\_path=/tmp/hive --cos\_path=/hdfs/20170224/

--decompress\_harパラメータを指定しない場合、デフォルトで通常のHDFSディレクトリがコピーされます。すな わち、.harディレクトリ内のindexやmasterindexなどのファイルがそのままコピーされるということです。

ディレクトリ情報





```
    conf : 設定ファイル。core-site.xmlとcos_info.confを保存するときに使います
    log : ログディレクトリ
    src : Javaソースプログラム
    dep : 発行した実行可能なJARパッケージをコンパイルします
```

質問とヘルプ

設定情報について

バケット(Bucket)、リージョン(Region)、APIキー情報など、入力された設定情報が正しいことを確認してください。このうちバケット名は、ユーザー定義の文字列と、システムが発行したAPPID文字列をハイフンが連結することで構成されます(例:examplebucket-1250000000)。また、マシンの時刻が北京の時刻と一致していることを確認してください(1分程度の差は正常です)。差が大きい場合は、マシンの時刻をリセットしてください。

### DataNodeについて

DataNodeについては、コピープログラムが配置されているマシンも接続できることを確認してください。 NameNodeには接続するパブリックネットワークIPがありますが、取得したblockが配置されているDateNodeマシ ンはプライベートネットワークIPであり、直接接続することはできません。したがって、NameNodeとDataNode の両方にアクセスできるように、Hadoopのいずれかのノードで同期プログラムを実行することをお勧めします。

### 権限について

Hadoopコマンドを使用してファイルをダウンロードし、正常かどうかチェックしてから、同期ツールを使用して Hadoopのデータサポートを同期してください。

### ファイルの上書きについて

COSにすでに存在するファイルについては、デフォルトで再送信と上書きが行われます。ユーザーが明示的にskip\_if\_len\_matchを指定しない限り、ファイルの長さが同じである場合、アップロードはスキップされます。

#### cos pathについて

cos pathのデフォルトはディレクトリであり、最終的にHDFSからコピーされるファイルはこのディレクトリに保存されます。

### Tencent Cloud EMR HDFSからのデータのコピーについて

Tencent Cloud EMR HDFSからCOSにデータをコピーする場合、高性能のDistcpツールを使用することをお勧めし ます。HadoopファイルシステムとCOS間のデータ移行をご参照ください。

# オンライン補助ツール COSリクエストツール

最終更新日:::2024-06-26 10:27:35

# 機能の説明

COSリクエストツールはTencent Cloud Object Storage (COS)が提供するWeb端末デバッグツールです。 Tencent Cloud API 3.0 Explorerプラットフォームに統合され、APIインターフェースのデバッグ作業に使用できま す。

### 注意:

COSリクエストツールが送信したリクエストは実際にCOSの業務サーバーに送信されます。すべての操作は実際の操作に等しいため、DELETE系の操作を選択する場合は慎重に操作してください。

現在COSリクエストツールがサポートしているAPIバージョンはXML APIです。JSON APIバージョンはサポート していません。

JSON APIはTencent Cloud COSサービスがXML APIを打ち出す前にユーザーにCOSにアクセスして使用するため に提供したAPIインターフェースです。インターフェースと標準XMLのAPIの基礎となるアーキテクチャは同じ で、データは相互接続し、互いに利用することができますが、インターフェースに互換性はありません。 XML APIにはより豊富な機能と特徴があります。Tencent Cloud COSはXML APIバージョンにアップグレードする ことを強くお勧めします。

### ツールアドレス

COSリクエストツールをクリックして進みます。

# 使用方法

**COS**製品を選択し、必要なAPIインターフェースを選択し、そのインターフェース下で対応するパラメータを入力 し、リクエストの送信をクリックして対応するリクエストのレスポンス結果を取得します。

COSリクエストツールのページ全体は、左から右へ順に製品欄,インターフェース欄,パラメータ欄および結果 欄です。下図のようにそれぞれのカテゴリーで対応する操作を実行でき、最後に結果欄でリクエストを送信してレ スポンス結果および関連するプロセスパラメータ情報を取得できます。

Search Tencent Cloud services and APIs		Q	GetService GetServiceAPI Document 🗷	Code Generating Online Call COS self-service diagnostic tool 🖾
Cloud Virtual Machine	Service API	~	Deline Ken	
lighthouse	GET Service		Private Key	Note: Please note that sending requests via the API 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	~	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	Ý		nformation like request result, header, etc. for your debugging and other reference.
Cloud Object Storage	Bucket Lifecycle	~	view Only Required Parameters	
Tencent Kubernetes Engine	Bucket Policy	×	Region ()	Send kequest
Tencent Container Registry	Bucket Referer	~	regrons compret up-guongenou -	
Cloud File Storage	Bucket Tagging	×	Headers 🚯	
6	Bucket Website	~	key value +	
Cloud HDFS	Bucket Inventory	~		
Cloud Block Storage	Bucket Cross-region	~		
Cloud Load Balancer	Replicatori			
Virtual Private Cloud	Bucket Versioning	Ť		
Direct Connect	Object Basic Operations	č		
Content Delivery Network	Object ACL	Ý		
Global Application Acceleration F	Object Multipart Upload	Ý		

COSリクエストツールに関する詳細な操作は以下の手順をご確認ください。

### 1. COS製品の選択

一番左側の製品欄内で**COS**をクリックすると、APIインターフェース欄内のCOSに関連するAPIインターフェース を確認することができます。

#### 説明:

COSリクエストツールはTencent Cloud API3.0プラットフォームに統合され、このプラットフォームにはその他の Tencent Cloud製品のAPI デバッグツールが搭載されています。同じようにニーズに応じてこのプラットフォーム 上でその他の製品を選択し、対応するインターフェースをデバッグすることができます。

#### 2. デバッグする必要があるAPIインターフェースを選択します

ニーズに応じて対応するAPIインターフェースを選択してデバッグを行うことができます。APIインターフェース 欄には3つのクラスとCOSに関連するAPIインターフェースが表示されます: Serviceインターフェース、Bucketイ ンターフェースおよびObjectインターフェース。

Serviceクラスのインターフェースは、例えばGET Serviceです。このAPIアカウント下の全てのバケット情報を一 覧表示し、APIキー情報を入力する必要があります。アカウントの指定領域内のバケット情報を取得する必要があ る場合は、パラメータ欄内で対応するリージョンを選択できます。このAPIの詳細情報について知りたい場合は、 GET Service ドキュメントをご参照ください。

Bucketクラスのインターフェースには、バケットに操作を行うAPIインターフェース、例えばPUT Bucket lifecycle が含まれます。より詳しくBucketクラスのインターフェースについて知りたい場合は、Bucketインターフェース をご参照ください。

Objectクラスのインターフェースには、各オブジェクトに操作を行うことができるAPIインターフェース、例えば PUT Objectが含まれます。より詳しくObjectクラスのインターフェースについて知りたい場合は、Objectインター フェースをご参照ください。

### 3. APIに必要なパラメータ情報の入力

パラメータ欄には、選択したAPIインターフェースの入力必須パラメータが表示されます。COSに関する各APIイ ンターフェースのパラメータ情報については、APIドキュメントを参照し、ご確認ください。

APIキー情報はAPIインターフェースを呼び出す段階で入力必須のパラメータです。APIインターフェースを使用し てバケットまたはオブジェクトなどのリソースを操作する際は、APIキー情報を入力し、今回のAPI リクエスト操 作を許可する必要があります。CAMコンソールの APIキー管理 ページにアクセスしてAPIキー情報を取得すること ができます。

#### 説明:

各APIインターフェースに対して、COSリクエストツールは各入力必須のパラメータ項目の後にすべて赤い星マー クを追加してそのパラメータが入力必須項目であることを通知します。入力必須パラメータのみを表示にチェック を入れて、パラメータ欄の入力必須ではないパラメータの表示を減らすこともできます。

### 4. リクエストの送信およびレスポンス結果の確認

APIを選択し、対応するパラメータを入力した後、オンラインコールのオプションタブでリクエストの送信をク リックします。この時システムはリクエストをサーバーに送信し、リクエストに応じてバケットまたはオブジェク トリソースを操作します。

#### 注意:

COSリクエストツールが送信したリクエストを実際にCOSの業務サーバーに送信します。すべての操作は実際の 操作に等しいため、DELETE系の操作を選択する場合は慎重に操作してください。

リクエストの送信後に、返された結果およびリクエストパラメータは結果欄の下部に表示されます。結果欄下部で は、リクエストパラメータでHTTPリクエストボディ情報を確認できます。レスポンス結果で今回のリクエストの レスポンスボディ情報が確認できます。署名プロセスで今回のリクエストに関する署名およびその生成プロセスが 確認でき、同時にシステムはCurlでCurl呼び出しのステートメントを提供します。

#### 事例

GET Objectを例に取ると、リクエストの送信によって名前が0001.txtのファイルを取得したい場合、リクエストパ ラメータ内に今回のリクエストのリクエストパラメータ情報が表示されます。今回のリクエストのリクエスト例は 次のとおりです。





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-si

最初の行に表示されるのはHTTP Verbおよびアクセスのリンクです。次の行に表示されるのはアクセスするドメイン名です。最後の行に表示されるのは今回のリクエストの署名情報です。PUTクラスのリクエストは、そのリクエストヘッダー情報が複雑ですが、同様にいくつかのパブリックリクエストヘッダーが存在します。パブリックリクエストヘッダーに関する情報は、パブリックリクエストヘッダーをご参照ください。

**署名プロセス**では今回のリクエストに関する署名およびその生成プロセスを確認することができます。署名アルゴ リズムに関する詳細の紹介はリクエスト署名を参照し、詳細をご確認ください。リクエスト署名に生成およびデ バッグを行う場合は、COS署名ツールを使用することをお勧めします。 COSが返したレスポンス結果は以下のとおりです。



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!

最初の行の200 OKはそのリクエストが返された状態コード情報を表し、リクエストが失敗した場合は、対応する エラーコードを返します。エラーコードに関する詳細情報は、エラーコード ドキュメントをご参照ください。そ の後はレスポンスヘッダー情報で、異なるAPIのレスポンスボディにはいずれも違いがありますが、いくつかのパ ブリックレスポンスヘッダー情報が存在します。パブリックレスポンスヘッダーに関する詳細情報は、パブリッ クレスポンスヘッダーをご参照ください。

### 注意事項

リクエストの送信をクリックして、入力必須のパラメータを入力したリクエストがCOSサーバーに送信されたことを確認すると、COSはバケットとオブジェクトに対応する操作を行います。この操作は取り消し、ロールバックする事ができません。慎重にご検討ください。

# セルフ診断ツール

最終更新日:::2024-06-26 10:27:35

### 機能の説明

Cloud Object Storage(COS)自己診断ツールは、Tencent Cloud COSがユーザー向けに提供するWebツールで、エ ラーリクエストのセルフチェックとトラブルシューティングができます。ツール画面でリクエストのRequestIdを 入力し(取得方法はRequestId取得の操作ガイドをご参照ください)、診断をクリックすると、ツールはリクエス トのインテリジェント診断を行います。また、このリクエストの基本情報やヘルプガイドおよび診断プロンプトを 表示して、COSAPIを使用した際のエラーをすばやく見つけるのに役立ちます。

### ツールアドレス

ここをクリックしてCOS自己診断ツールに進みます。

### 利用方法

1. COS自己診断ツールをクリックして、「COS自己診断ツール」画面に進みます。

2. 上部のRequestId入力ボックスに診断するRequestIdを入力し、【診断の開始】をクリックします。

3. しばらくすると、対応するインテリジェント診断の結果が表示されます。

診断結果には、「診断結果」と「リクエスト情報」という2つの部分があります。

診断結果は、診断後のアドバイスやヘルプをリクエストするためのものです。アドバイスとヘルプに従って、 COS APIのエラーをすばやく特定することができます。

リクエスト情報は、この RequestIdに対応する基本情報です。

4. 診断結果をフィードバックします。

診断結果の下にある【役に立った】または【役に立たなかった】をクリックして、診断結果に関するフィード バックをお送りください。いただいたご意見は、ツールの最適化を引き続き行うために役立ててまいります。

# よくあるご質問

このツールは、RequestIdを使用した診断トラブルシューティングに加え、COS APIの使用と、それに関連する問題をチェックするために、よくあるご質問のサンプルと対応するソリューションも豊富にご提供しています。また、ご不明な点がございましたら、お問い合わせをご利用ください。

# 注意事項

標準的なCOSリクエストRequestIdには、次のような特徴があります。

- 1. 大文字のNから始まる。
- 2. 長さは30文字以上でなければならない。

3. base64の文字規格に準拠している。