

对象存储 工具指南



腾讯云

【版权声明】

©2013-2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

工具指南

工具概览

环境安装与配置

Java 安装与配置

Python 安装与配置

Hadoop 安装与测试

COSBrowser 工具

COSBrowser 简介

桌面端使用说明

移动端使用说明

安装与登录

移动端功能列表

存储桶管理与操作

文件管理与操作

资源包管理

COS 终端网络诊断工具使用指南

COSCMD 工具

COSCLI 工具

COSCLI 简介

下载与安装配置

通用选项

常用命令

生成与修改配置文件 - config

创建存储桶 - mb

删除存储桶 - rb

存储桶标签 - bucket-tagging

查询存储桶或文件列表 - ls

获取不同类型文件的统计信息 - du

上传下载或拷贝文件 - cp

同步上传下载或拷贝文件 - sync

删除文件 - rm

获取文件哈希值 - hash

列出分块上传中产生的碎片 - lsparts

清理碎片 - abort

取回归档文件 - restore

获取预签名 URL - signurl

创建/获取软链接 - symlink

查看对象内容 - cat

列出目录下内容并统计 - lsdu

版本控制 - bucket-versioning

常见问题

COS Migration 工具

FTP Server 工具

Hadoop 工具

COSDistCp 工具

HDFS TO COS 工具

GooseFS-Lite 工具

在线辅助工具**COS 签名工具****COS 请求工具****COS 策略生成工具****自助诊断工具****文档预览接入体验工具**

工具指南

工具概览

最近更新时间：2024-12-18 18:04:52

工具	功能说明
COSBrowser 工具	本工具支持多种操作系统，提供可视化界面，可方便用户进行数据上传、下载、生成访问链接等操作。
COSCLI 工具	本工具是一款客户端命令行工具，提供多种常用命令，您可以使用简单的命令行指令实现对象批量上传、同步上传、同步下载、删除等操作。
COSCMD 工具	本工具是另一款客户端命令行工具，可实现对象批量上传、下载、删除等操作。与 COSCLI 功能互补。
COS Migration 工具	本工具是一款数据迁移工具，支持用户将本地数据迁移至 COS。
FTP Server 工具	本工具支持用户使用 FTP 客户端将文件上传到 COS 或者从 COS 下载文件。
Hadoop 工具	Hadoop 工具可以为 Hadoop、Spark 以及 Tez 等大数据计算框架集成 COS 提供支持，便于读写存储在 COS 上的数据。
COSDistCp 工具	COSDistCp 是一款基于 MapReduce 的分布式文件拷贝工具，主要用于 HDFS 和 COS 之间的数据拷贝。
Hadoop-cos-DistChecker 工具	Hadoop-cos-DistChecker 是一款用于校验迁移目录完整性的工具，用户在使用 <code>hadoop distcp</code> 命令从 HDFS 迁移数据到 COS 上后，可使用该工具进行校验。
HDFS TO COS 工具	本工具用于将 HDFS 上的数据拷贝到 COS 上。
GooseFS-Lite 工具	本工具支持将 COS 存储桶挂载到本地，相比于 COSFS 工具，GooseFS-Lite 可提供更高的大文件读写速度，不受本地磁盘的性能限制。
在线辅助工具	本类工具是 COS 为用户提供的 Web 端辅助工具，包括有 COS 签名工具 、 COS 请求工具 。
自助诊断工具	自助诊断工具是腾讯云对象存储为用户提供的 Web 工具，可用于错误请求的自助诊断排查。
文档预览接入体验工具	文档预览接入体验工具是 COS 为用户提供的 Web 端工具，可用于文档转 HTML 预览服务的接入调试。

当然，COS 不仅提供以上应用和服务，还提供多款热门开源应用，并集成腾讯云 COS 插件，欢迎单击 [此处](#) 一键启动，立即使用！

上传能力

各工具对于上传能力的支持情况如下表所示：

工具	简单上传	分块上传	断点上传	高级上传	一致性校验	生成预签名 URL
COSBrowser 工具	支持	支持	支持	支持，默认8MB触发分块上传	支持 md5 校验	支持文件下载
COSCLI 工具	支持	支持	支持	支持，可自定分块触发阈值	支持 crc64 校验	支持
COSCMD 工具	支持	支持	支持	支持，默认10MB触发分块上传	支持 md5 校验	支持
COS Migration 工具	支持	支持	支持	支持，可自定分块触发阈值	支持 md5 校验	不适用
FTP Server 工具	支持	支持	支持	支持，可自定分块触发阈值	不支持	不适用

Hadoop 工具	支持	支持	不支持， HDFS 协议冲突	支持，可自定分块触发阈值	支持 crc64、 crc32 校验	不适用
COSDistCp 工具	支持	支持	不支持， HDFS 协议冲突	支持，可自定分块触发阈值	支持文件大小、 crc64、crc32 校验	不适用
HDFS TO COS 工具	支持	支持	不支持， HDFS 协议冲突	支持，可自定分块触发阈值	支持文件名称和大 小校验	不适用

!

说明：

高级上传对简单上传和分块上传进行封装，支持根据文件大小选择上传方式。

环境安装与配置

Java 安装与配置

最近更新时间：2024-03-25 14:34:51

JDK 是 Java 软件开发工具包，本文以 JDK 1.8 版本为例，分别介绍了在 Windows 和 Linux 系统下，JDK 的安装与环境配置过程。

Windows

1. 下载 JDK

进入 [Oracle 官方网站](#) 下载合适的 JDK 版本。

2. 安装

根据提示一步步安装，安装过程中可以自定义安装目录（默认安装到 C 盘），例如我们选择的安装目录为：

D:\Program Files\Java\jdk1.8.0_31 和 D:\Program Files\Java\jre1.8.0_31。

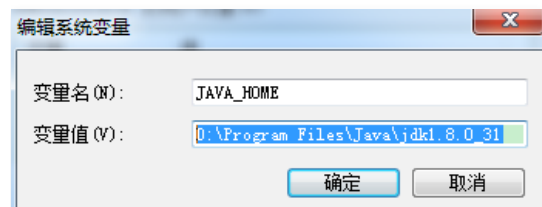


3. 配置

安装完成后，右键单击计算机 > 属性 > 高级系统设置 > 环境变量 > 系统变量 > 新建，分别配置软件。

变量名(N): **JAVA_HOME**

变量值(V): D:\Program Files\Java\jdk1.8.0_31 （请根据您的实际安装路径进行配置）。



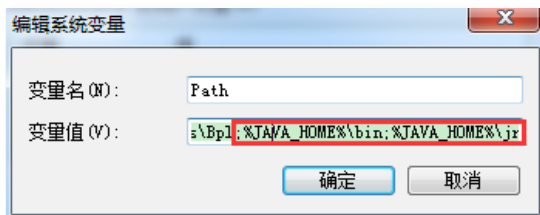
变量名(N): **CLASSPATH**

变量值(V): .;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar; （注意变量值开头有 . ）。



变量名(N): Path

变量值(V): %JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;

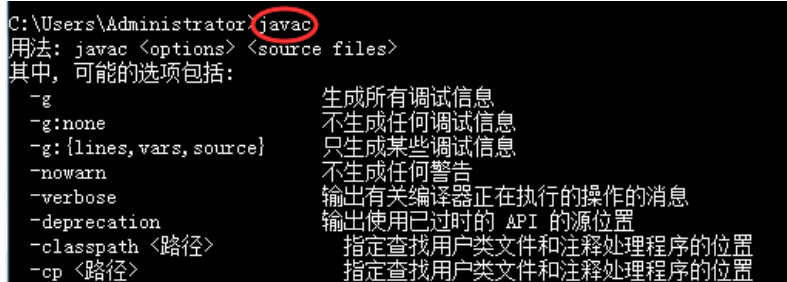


⚠ 注意:

如果 Path 变量已有值，可以在其末尾添加分号 (;)，然后继续添加新的变量值。

4. 测试

测试配置是否成功：单击开始（或快捷键：Win+R）> 运行（输入 cmd）> 确定（或按 Enter 键），输入命令 javac 并回车。若出现如下图所示信息，则说明环境变量配置成功。



Linux

由于使用 yum 或者 apt-get 命令安装 openjdk 可能存在类库不全，从而导致用户在安装后运行相关工具时可能报错的问题，所以此处我们推荐采用手动解压安装的方式来安装 JDK。具体步骤如下：

1. 下载 JDK

进入 [Oracle 官方网站](#) 下载合适的 JDK 版本，准备安装。

⚠ 注意

下面以 jdk-8u151-linux-x64.tar.gz 为例，如果您下载的是其他版本，请注意文件后缀为 .tar.gz 即可。

2. 创建目录

执行如下命令，在 /usr/ 目录下创建 java 目录。

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

将下载的文件 jdk-8u151-linux-x64.tar.gz 复制到 /usr/java/ 目录下。

3. 解压 JDK

执行如下命令，解压文件：

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

4. 设置环境变量

编辑 /etc/profile 文件，在 profile 文件中添加如下内容并保存：

```
# 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

其中 JAVA_HOME, JRE_HOME 请根据自己的实际安装路径及 JDK 版本配置。

使之修改生效，执行如下：

```
source /etc/profile
```

5. 测试

执行如下命令进行测试：

```
java -version
```

若显示 Java 版本信息，则说明 JDK 安装成功：

```
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 安装与配置

最近更新时间：2024-05-13 11:05:31

本文档简要介绍在不同操作系统下，Python 开发环境的安装方法。

通过安装包安装

1. 下载

进入 [Python 官方网站](#)，根据自己使用的操作系统，选择下载合适的安装包。

⚠ 注意：

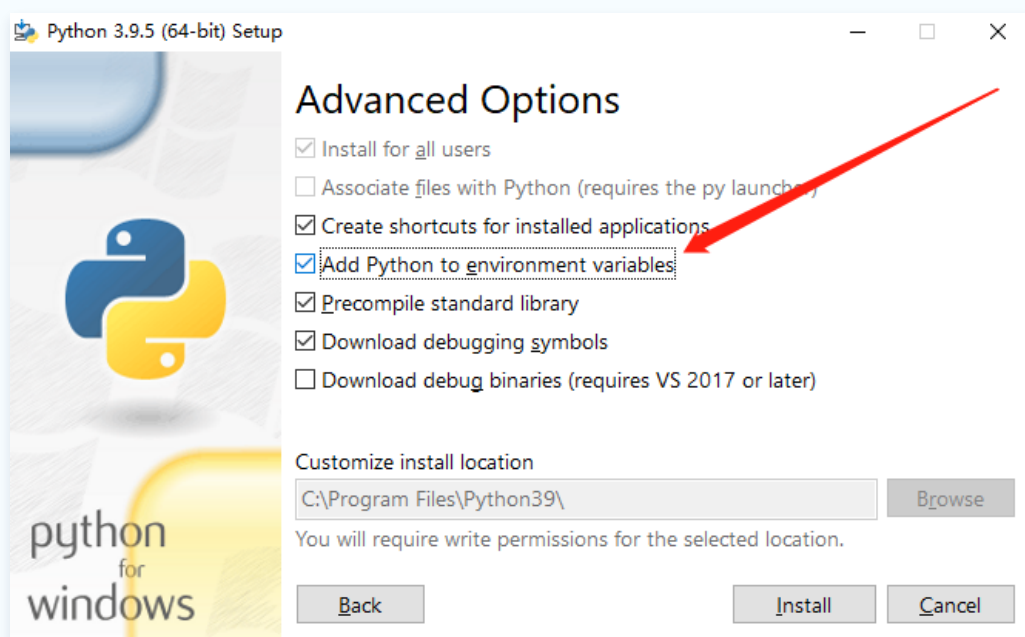
Python 官方已于2020年1月1日停止对 Python2 的维护，建议安装使用 Python3。

2. 安装

下载安装包后，根据安装包的引导即可完成 Python 开发环境的安装。

📌 说明：

Windows 系统用户在安装时要注意勾选"Add Python to environment variables"选项。



3. 验证

在终端中执行以下命令查看 Python 版本：

```
python -V
```

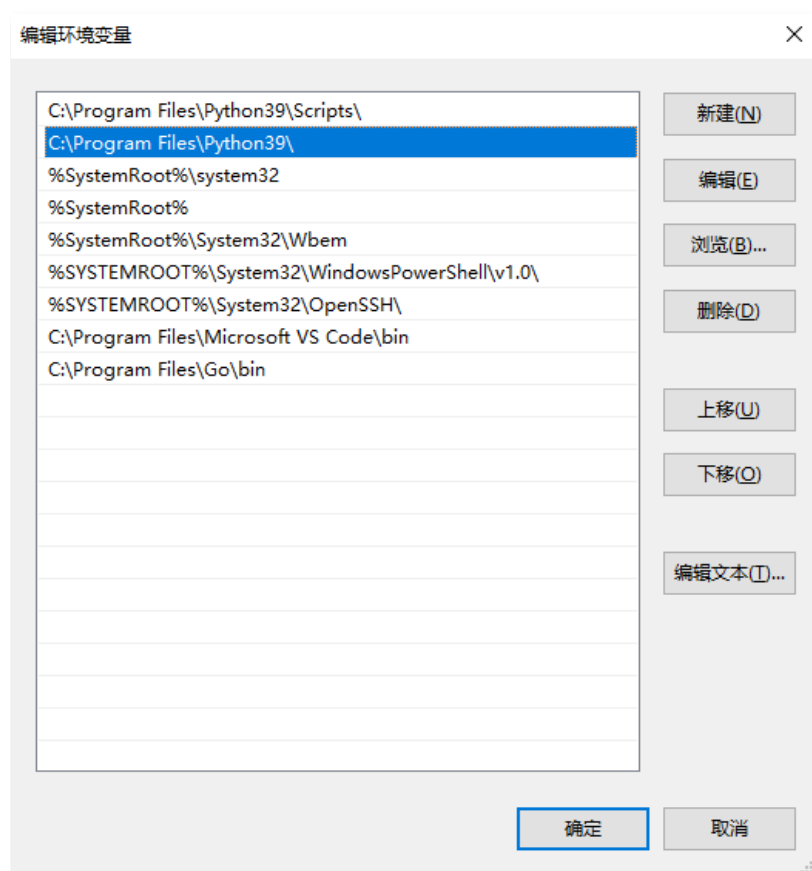
若终端输出 Python 版本号则证明安装成功。

📌 说明：

Windows 系统用户在安装完毕后，可能需要重启计算机。

4. 环境变量配置

在 Windows 系统中，若在执行以上命令时终端提示"不是内部或外部命令"，请在计算机 > 属性 > 高级系统设置 > 环境变量 > 系统变量(S)中编辑 "Path"，增加 Python 的安装路径，如下图所示：



通过包管理器安装

Mac OS

使用 Mac OS 的用户可以先安装 [HomeBrew](#)，再通过 HomeBrew 来安装 Python：

```
brew install python
```

Ubuntu

使用 Ubuntu 的用户可以使用 Ubuntu 自带的 apt (Advanced Packaging Tool) 包管理器来安装 Python：

```
sudo apt-get install python
```

CentOS

使用 CentOS 的用户可以使用 CentOS 自带的 yum (Yellow dog Updater, Modified) 包管理器来安装 Python：

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

Hadoop 安装与测试

最近更新时间：2024-08-07 18:00:51

Hadoop 工具依赖 Hadoop-2.7.2 及以上版本，实现了以腾讯云对象存储（Cloud Object Storage，COS）作为底层存储文件系统运行上层计算任务的功能。启动 Hadoop 集群主要有单机、伪分布式和完全分布式等三种模式，本文主要以 Hadoop-2.7.4 版本为例进行 Hadoop 完全分布式环境搭建及 wordcount 简单测试介绍。

准备环境

1. 准备若干台机器。
2. 安装配置系统，可前往 [CentOS 官网](#) 下载安装。本文使用 CentOS 7.3.1611 系统版本。
3. 安装 Java 环境，具体操作请参见 [Java 安装与配置](#)。
4. 安装 Hadoop 可用包：[Apache Hadoop Releases Download](#)。

网络配置

使用 `ifconfig -a` 查看各台机器的 IP，相互使用 `ping` 命令检查，看是否可以 `ping` 通，同时记录每台机器的 IP。

配置 CentOS

配置 hostname

分别给机器设置相应hostname，如"master"、"slave*"等。

```
hostnamectl set-hostname master
```

配置 hosts

```
vi /etc/hosts
```

编辑内容：

```
202.xxx.xxx.xxx master
202.xxx.xxx.xxx slave1
202.xxx.xxx.xxx slave2
202.xxx.xxx.xxx slave3
# IP 地址替换为真实 IP
```

关闭防火墙

```
systemctl status firewalld.service # 检查防火墙状态
systemctl stop firewalld.service   # 关闭防火墙
systemctl disable firewalld.service # 禁止开机启动防火墙
```

时间同步

```
yum install -y ntp # 安装 ntp 服务
ntpdate cn.pool.ntp.org # 同步网络时间
```

安装配置 JDK

上传 JDK 安装包（如jdk-8u144-linux-x64.tar.gz）到 `root` 根目录。


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

各个主机之间复制 JDK

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

配置各个主机 JDK 环境变量

```
vi /etc/profile
```

编辑内容：

```
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
```

保存文件后，使/etc/profile生效，执行以下命令：

```
source /etc/profile          # 使配置文件生效
java -version                # 查看 java 版本
```

配置 SSH 无密钥访问

分别在各个主机上检查 SSH 服务状态：

```
systemctl status sshd.service      # 检查 SSH 服务状态
yum install openssh-server openssh-clients  # 安装 SSH 服务，如果已安装，则不用执行该步骤
systemctl start sshd.service        # 启动 SSH 服务，如果已安装，则不用执行该步骤
```

分别在各个主机上生成密钥：

```
ssh-keygen -t rsa # 生成密钥
```

在 slave1 上：

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

在 slave2 上：

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

依此类推...

在 master 上：

```
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
scp authorized_keys slave3:~/.ssh
```

安装配置 Hadoop

安装 Hadoop

上传 hadoop 安装包（如hadoop-2.7.4.tar.gz）到 `root` 根目录。

```
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
```

进入 `hadoop-2.7.4/etc/hadoop` 目录下，进行下一步操作。

配置 Hadoop

1. 修改 `hadoop-env.sh` 文件，增加如下内容：

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

若 SSH 端口不是默认的22，可在 `hadoop-env.sh` 文件里修改：

```
export HADOOP_SSH_OPTS="-p 1234"
```

2. 修改 `yarn-env.sh`

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

3. 修改 `slaves`

配置内容：

```
删除：
localhost
添加：
slave1
slave2
slave3
```

4. 修改 `core-site.xml`

```
<configuration>
  <property>
    <name>fs.default.name</name>
```

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

5. 修改 `hdfs-site.xml`

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

6. 将 `mapred-site.xml.template` 拷贝一份出来命名为 `mapred-site.xml`

```
cp mapred-site.xml.template mapred-site.xml
```

7. 修改 `mapred-site.xml`

```
<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>
```

8. 修改 `yarn-site.xml`

```
<configuration>
  <property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.mapred.ShuffleHandler</value>
  </property>
  <property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
  </property>
</configuration>
```

```
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>master:8032</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>master:8030</value>
</property>
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>master:8031</value>
</property>
<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>
```

9. 各个主机之间复制 Hadoop

```
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
```

10. 各个主机配置 Hadoop 环境变量

打开配置文件：

```
vi /etc/profile
```

编辑内容：

```
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
```

使配置文件生效：

```
source /etc/profile
```

启动 Hadoop

1. 格式化 namenode

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

2. 启动

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

3. 检查进程

master 主机包含 ResourceManager、SecondaryNameNode、NameNode 等，则表示启动成功，例如：

```
2212 ResourceManager
2484 Jps
1917 NameNode
2078 SecondaryNameNode
```

各个 slave 主机包含 DataNode、NodeManager 等，则表示启用成功，例如：

```
17153 DataNode
17334 Jps
17241 NodeManager
```

运行 wordcount

由于 Hadoop 自带 wordcount 例程，所以可以直接调用。在启动 Hadoop 之后，我们可以通过以下命令来对 HDFS 中的文件进行操作：

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

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

出现如上图结果就说明 Hadoop 安装已经成功了。

查看输出目录

```
hadoop fs -ls /output
```

查看输出结果

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

```
[root@VM_96_24_centos /usr/hadoop-2.7.4]# hadoop fs -cat /output/part-r-00000
a      5
dasdada 1
ds     2
qwe    1
ret    1
s      1
v      2
vdffd  1
wqere  1
```

❗ 说明：

单机模式与伪分布式模式的操作方法的详细过程，请参见官网文档 [Hadoop入门](#)。

COSBrowser 工具

COSBrowser 简介

最近更新时间：2024-11-05 18:49:31

COSBrowser 是腾讯云对象存储 COS 推出的可视化界面工具，让您可以使用更简单的交互，轻松实现对 COS 资源的查看、传输和管理。若迁移数据，或批量上传数据，请使用 [云迁移（Cloud Migration，CMG）](#)。目前 COSBrowser 有桌面端和移动端两种，详情可参见：

- [桌面端使用说明](#)
- [移动端使用说明](#)

下载地址

COSBrowser 分类	支持平台	系统要求	下载地址
桌面端	Windows	Windows 7 32/64位以上、Windows Server 2008 R2 64位以上	Windows
	macOS	macOS 10.13以上	macOS Intel 芯片版 macOS Apple 芯片版
	Linux	需带有图形界面并支持 ApplImage 格式 ⚠ 注意： CentOS 启动客户端需在终端执行 <code>./cosbrowser.AppImage --no-sandbox</code>	Linux
移动端	Android	Android 5.0 及以上	Android
	iOS	iOS 12 及以上	iOS
网页版	Web	Chrome/FireFox/Safari/IE10+等浏览器	Web

桌面端功能列表

COSBrowser 桌面端注重对资源的管理，用户可以通过 COSBrowser 批量的上传，下载数据。

⚠ 注意：

COSBrowser 桌面端会使用系统配置的代理来尝试网络连接，请确保您的代理配置正常或请停用无法连接互联网的代理配置。

- Windows 用户可在操作系统的 “Internet 选项” 中查询。
- macOS 用户可在 “网络偏好设置” 中查询。
- Linux 用户可在系统设置 > 网络 > 网络代理中查询。

COSBrowser 桌面端支持以下功能：

存储桶管理

功能	功能说明
创建/删除存储桶	支持创建、删除存储桶
查看/配置存储桶	支持查看存储桶基本信息
搜索存储桶	支持按照存储桶名称或者存储桶标签快速搜索存储桶

查看统计数据	支持查看存储桶的当前存储量及对象总数
设置存储桶访问权限	支持修改存储桶访问权限
设置存储桶标签	支持查看、编辑或删除存储桶标签
设置存储桶的上传/下载域名	支持设置上传、下载域名
查看域名	支持查看存储桶的域名配置情况
设置版本控制	支持开启、暂停存储桶版本控制
设置全球加速	支持开启、暂停存储桶全球加速
添加收藏路径	支持添加收藏路径
生成授权码	通过授权码可实现临时登录 COSBrowser 客户端

对象管理

功能	功能说明
上传文件/文件夹	支持单个上传、批量上传、增量上传文件或文件夹至存储桶
下载文件/文件夹	支持单个下载、批量下载、增量下载文件或文件夹至本地
删除文件/文件夹	支持单个删除、批量删除存储桶中的文件或文件夹
文件同步	支持将本地文件实时同步至存储桶中
复制粘贴文件	支持单个复制、批量复制一个目录下的文件或文件夹至另一个目录
文件重命名	支持重命名存储桶中的文件
新建文件夹	支持在存储桶中新建文件夹
查看文件详情	支持查看存储桶中的文件基本信息
生成文件链接	支持通过请求临时签名的方式生成带有时效的文件访问链接
分享文件/文件夹	支持分享文件和文件夹，支持设置分享的有效时间
导出文件链接	支持批量导出文件链接
文件预览	支持预览存储桶中的媒体文件（图片、视频、音频）
搜索文件	支持以前缀搜索的方式对存储桶中的文件进行搜索
查看历史版本或文件碎片	支持在已开启版本控制的存储桶中，查看文件的历史版本支持查看存储桶内的文件碎片详情
文件对比	支持将本地文件夹与存储桶里的文件进行对比
文件列表管理	文件列表支持列表布局、网格布局

数据处理

功能	功能说明
视频转码	支持对已开启媒体处理功能的存储桶中的文件进行转码
图片处理	支持缩放、裁剪、旋转等基本图片处理和文字水印、图片水印，并支持生成处理后的图片链接

软件设置

功能	功能说明
通用配置	支持设置 EndPoint/Service 域名、Bucket域名模板、自动切换备份域名、HTTPS、HTTPS 证书校验等
设置网络代理	支持设置网络代理来访问 COS
设置上传/下载的文件并发数	支持设置文件上传、下载传输并发数
设置上传/下载的分块并发数	支持设置文件分块上传、下载的分块数
设置上传/下载失败重试数	支持设置文件上传、下载失败时的重试次数
设置单线程上传/下载限速	支持设置单线程上传限速和下载限速
设置上传二次校验	支持对上传至存储桶中的文件进行二次校验
查看本地日志	支持将用户对 COSBrowser 的操作记录以本地日志的形式保存

移动端功能列表

COSBrowser 移动端注重对资源的查看及监控，用户可以随时随地监控 COS 的存储量、流量等数据。关于 COSBrowser 移动端所支持的功能，请参见 [移动端功能列表](#)。

更新日志

- 桌面端更新日志：[changelog](#)
- 移动端更新日志：[changelog_mobile](#)

反馈和建议

如您在使用 COSBrowser 时有任何疑问或建议，欢迎反馈给我们：

- 桌面端反馈：[issues](#)
- 移动端反馈：[issues_mobile](#)

结语

当然，COS 不仅提供以上应用和服务，还提供多款热门开源应用，并集成腾讯云 COS 插件，欢迎点击 [此处](#) 一键启动，立即使用！

桌面端使用说明

最近更新时间：2024-08-06 15:24:31

下载与安装

软件下载

支持平台	系统要求	下载地址
Windows	Windows 7 32/64位以上、Windows Server 2008 R2 64位以上	Windows
macOS	macOS 10.13以上	macOS Intel 芯片版 macOS Apple 芯片版
Linux	需带有图形界面并支持 ApplImage 格式	Linux
网页版	Chrome/FireFox/Safari/IE10+等浏览器	Web

软件安装

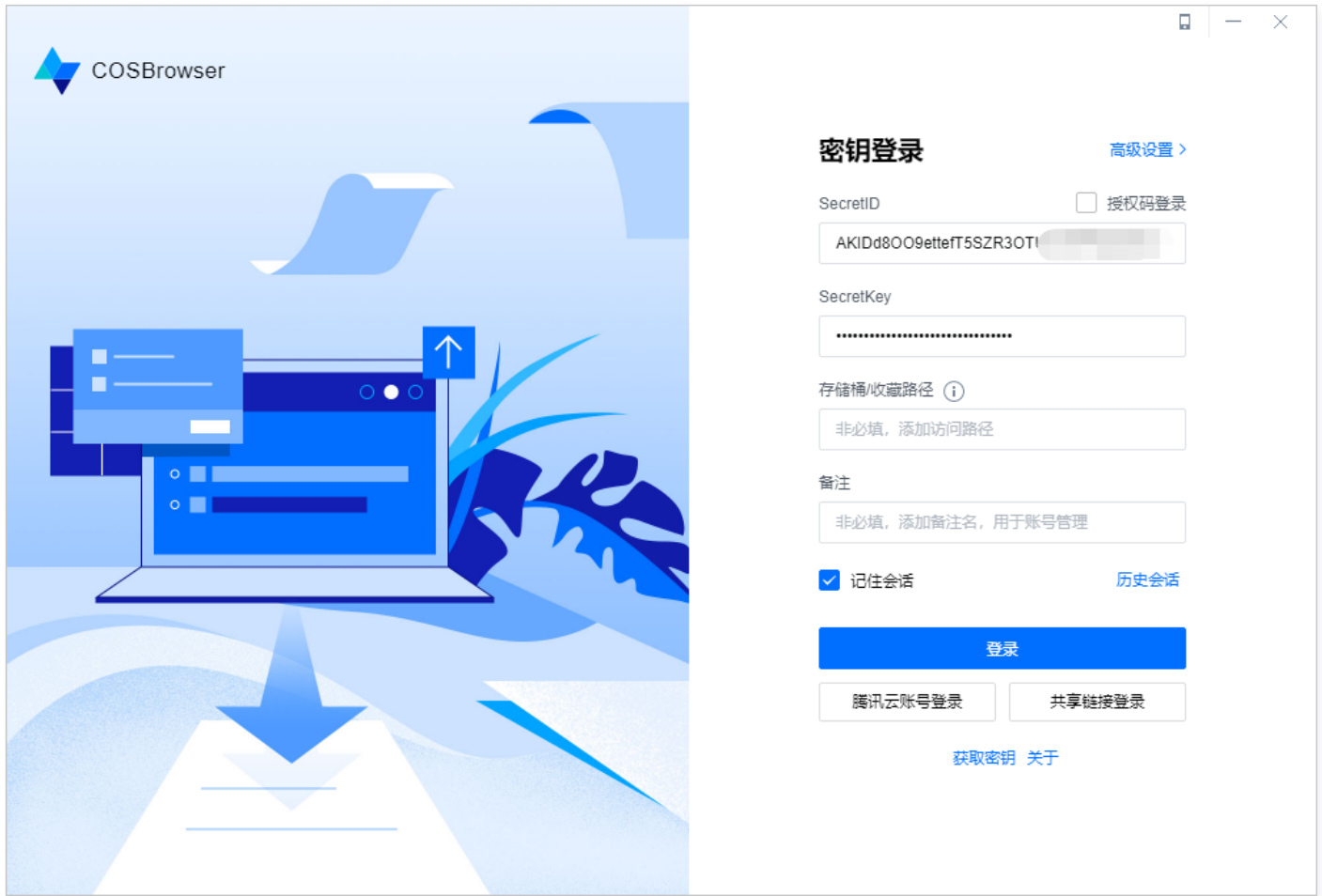
用户可根据不同的平台，选择通过程序包安装、解压安装或者在浏览器中免安装直接使用。

⚠ 注意：

CentOS 启动客户端需在终端执行 `./cosbrowser.AppImage --no-sandbox`。

登录软件

COSBrowser 桌面端支持多种登录方式：密钥登录、腾讯云账号登录、共享链接登录、授权码登录。支持同一账号多设备多点登录。



密钥登录

COSBrowser 支持通过永久密钥（即云 API 密钥）登录，配置项说明如下：

- **SecretID、SecretKey**：云 API 密钥，该密钥可在访问管理控制台的 [API 密钥管理](#) 页面中创建并获取。
- **存储桶/收藏路径**：若您当前使用的密钥仅允许访问指定资源范围且不允许访问存储桶列表，例如指定的存储桶或存储桶中某个路径，则填写后可快速进入对应的存储桶或文件路径。格式为：Bucket 或 Bucket/Object-prefix，例如当前使用的密钥仅被允许访问存储桶 examplebucket-1250000000 下的 doc 文件夹，则填写 examplebucket-1250000000/doc。操作指引请参见 [添加收藏路径](#)。
- **备注**：可对当前填入的永久密钥进行说明。例如操作人员、用途等。在**历史会话**中可用于区分不同的 SecretID。
- **记住会话**：
 - 若不勾选，则仅本次登录有效，退出登录则会清空填入的云 API 密钥（如果当前密钥已保存在历史会话中，则会从历史会话中移除）；
 - 若勾选，在成功登录后将会记住填入的云 API 密钥相关信息，方便下次继续使用，您可在**历史会话**中管理已保存的密钥相关信息。

⚠ 注意：

COSBrowser 不支持使用项目密钥进行登录。

腾讯云账号登录

单击**腾讯云账号登录**，在弹窗中通过腾讯云账号登录 COSBrowser 桌面端。腾讯云账号支持的登录方式有微信、邮箱、QQ、微信公众号、企业微信、子用户登录。具体指引可参见 [账号相关文档](#)。

⚠ 注意：

腾讯云账号登录方式会有会话超时限制，存在登录态失效问题，如果是上传大文件的场景建议使用密钥方式上传。

共享链接登录

通过别人转发或分享的**共享链接**和**提取码**可以临时登录 COSBrowser 桌面端。功能介绍可参见 [分享文件/文件夹](#)。

共享链接登录

共享链接

https://cosbrowser.cloud.tencent.com/share/?id=MeW6Y

提取码

56e58f

请输入 6 位提取码

登录

关闭

授权码登录

通过别人生成的授权码可以临时登录 COSBrowser 桌面端。功能介绍可参见 [生成授权码](#)。

密钥登录

高级设置 >

☒ 授权码登录

请输入授权码

登录

腾讯云账号登录

共享链接登录

获取密钥

关于

存储桶管理

⚠ 注意：

此处功能介绍以 Windows 版本 v 2.11.17 为例，其他版本差异，请参见 [更新日志](#)。

创建/删除存储桶

功能	说明	如何操作
创建存储桶	可以通过客户端直接创建存储桶	1. 在存储桶列表，单击左上角创建桶 2. 正确填写存储桶名称，并选择地域、访问权限，也可根据用户需求设置存储桶标签 3. 单击确定，即可完成创建
删除存储桶	删除存储桶前，请确认存储桶中数据已清空	1. 在存储桶列表，单击对应存储桶右侧的更多按钮 -> 删除 2. 确认存储桶中的数据已全部清空，单击确定进行删除

查看/配置存储桶

您可通过单击存储桶列表右侧的详情图标，可查看存储桶名称、地域、存储桶域名，可对存储桶访问权限、多 AZ 特性、智能分层、图片极智压缩、版本控制、全球加速等进行配置。

⚠ 注意：

多 AZ 特性、智能分层目前仅支持部分地域，详情可参见 [多 AZ 特性概述](#)、[智能分层存储简介](#)。

搜索存储桶

可通过窗口右侧的搜索框，输入存储桶名称或者存储桶标签快速定位存储桶。

查看统计数据

您可通过单击存储桶列表右侧的更多按钮-> **统计**，查看存储桶的统计数据。存储桶数据包含存储用量和对象数量。

设置存储桶访问权限

COSBrowser 支持对存储桶的访问权限进行管理，方法如下：

- 存储桶权限管理：可通过单击存储桶列表右侧操作栏中的更多按钮 -> **权限管理**，设置存储桶的公共权限和用户权限。
- 快速授权：可通过单击存储桶列表右侧操作栏中的**快速授权**，完成用户权限的授权。

❗ 说明：

有关 COS 的权限说明，可查看 [ACL概述](#)。

设置存储桶标签

您可通过单击存储桶列表右侧的更多按钮-> **标签管理**，查看、编辑或删除存储桶标签。

设置存储桶的上传/下载域名

您可通过单击存储桶列表右侧的更多按钮-> **传输管理**，对上传域名、下载域名进行指定，例如在已开启全球加速功能的存储桶中，可将上传、下载域名均设置为全球加速域名，设置后，您在上传或下载对象时将使用全球加速域名进行加速传输。

查看域名

您可通过单击存储桶列表右侧的更多按钮-> **查看域名**，可查看当前存储桶的域名配置情况，例如自定义 CDN 加速域名、自定义源站域名。

设置版本控制

您可通过单击存储桶列表右侧操作栏中的**详情**，在存储桶详情弹窗中，单击版本控制编辑图标，将当前状态修改为开启。开启版本控制后，您可查看同名文件的所有版本并对其进行管理，详情请参见 [查看历史版本或文件碎片](#)。

设置全球加速

您可通过单击存储桶列表右侧操作栏中的**详情**，在存储桶详情弹窗中，单击**全球加速编辑**图标，将当前状态修改为**开启**，并将存储桶的上传/下载域名设置为**全球加速域名**。详情请参见 [设置存储桶的上传/下载域名](#)。

添加收藏路径

若您使用无访问存储桶列表权限的子账号（或密钥）进行登录，可以通过**添加收藏路径**的方式快速访问已被授权的存储桶资源，COSBrowser 提供了两种添加收藏路径的方式：

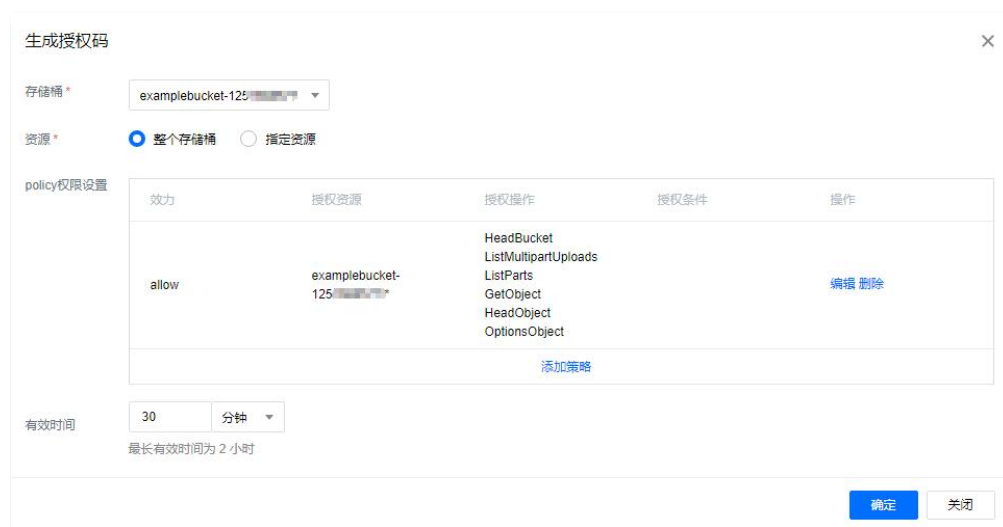
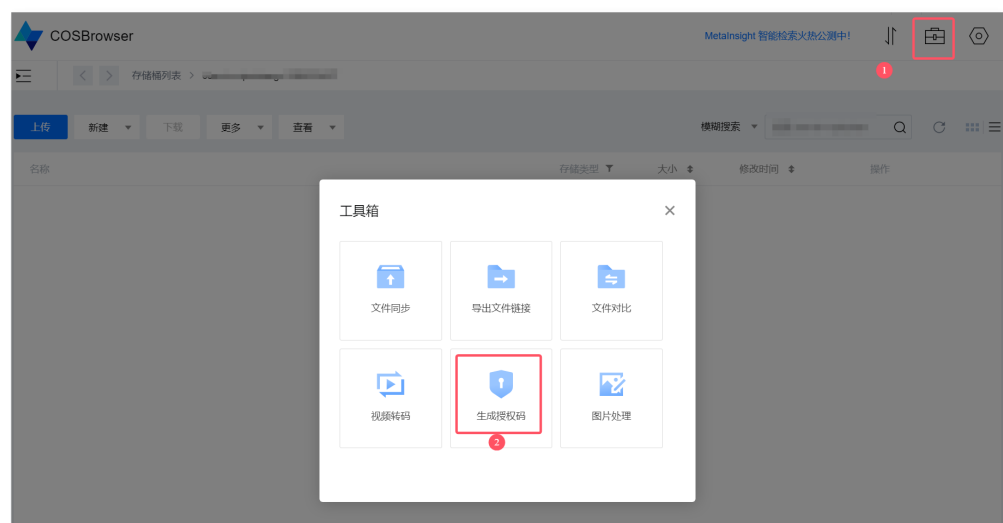
- 在软件登录界面，通过密钥登录方式，直接添加收藏路径，登录完成后，即可管理资源。
- 子账号登录后，在存储桶列表页左上角，单击**收藏的路径 > 添加路径**，并配置指定的路径和地域后进入存储桶管理资源。

生成授权码

通过生成授权码功能可实现对指定存储桶、存储桶下的资源和操作进行临时授权。相比分享文件夹功能更灵活，可以授予自定义操作权限和进入某个目录。通过生成临时使用的 SecretId、SecretKey、Token 和授权码，可实现临时登录客户端。

操作说明如下：

在工具界面右上方单击**工具箱**图标，在工具箱弹窗中选择**生成授权码**，然后选择想要生成授权码的存储桶和授权的资源范围，在 policy 权限设置中选择被授权的操作（例如读写权限），并设置授权码的有效时间（子账号允许设置的最长有效时间为1.5天），最后单击**确定**即可。



对象管理

上传文件/文件夹

上传功能	说明	如何操作
上传文件	COSBrowser 支持多种上传方式，支持单个或批量上传	上传文件的几种方式如下，在指定的存储桶或路径内： 1. 单击上传，选择文件后，直接上传文件。 2. 在文件列表空白处右键单击上传文件，进行上传。 3. 通过鼠标将文件拖拽至文件列表窗口进行上传。
上传文件夹及其文件	若存储桶或路径内存在同名文件或文件夹，则默认覆盖	上传文件夹的几种方式如下，在指定的存储桶或路径内： 1. 单击上传，选择文件夹，直接上传文件夹。 2. 在文件列表空白处右键选择上传文件夹，进行上传。 3. 通过鼠标将文件夹拖拽至文件列表窗口即可完成上传。
增量上传	增量上传指执行上传操作前，对上传文件与存储桶已有对象做比对，若存在同名对象，则跳过该文件不执行上传操作	增量上传操作步骤如下，在指定的存储桶或路径内： 1. 使用上传文件夹的方式，单击下一步。 2. 存储方式选择跳过，单击上传后，即可完成增量上传。

⚠ 注意：

若您需要大批量上传文件，建议电脑配置为4核、运行内存16GB（最大支持一次性上传30万个文件）。

下载文件/文件夹

下载功能	说明	如何操作
下载文件	COSBrowser 支持多种下载方式，支持单个或批量下载文件	下载文件的几种方式如下： 1. 选中需要下载的文件，单击界面内的下载，即可下载该文件。 2. 选中文件，右键单击下载。 3. 通过鼠标将文件拖拽至本地的方式进行下载。
下载文件夹及其文件	若本地已存在同名文件或文件夹，则默认重命名	下载文件夹及其文件的几种方式如下： 1. 选中需要下载的文件夹，通过单击界面内的下载，即可下载。 2. 右键单击下载，直接下载该文件夹。 3. 通过鼠标将文件夹拖拽至本地的方式进行下载。
增量下载	增量下载指执行下载操作前，将下载的对象与本地文件进行比对，若存在同名对象，则跳过该对象不执行下载操作	增量下载操作步骤如下： 1. 选中想要下载的文件/文件夹，然后单击鼠标右键。 2. 在菜单中单击高级下载，在弹窗中选择跳过。 3. 然后单击立即下载即可完成不重名文件/文件夹的增量下载。

⚠ 注意：

若您需要大批量下载文件，建议电脑配置为4核、运行内存16GB（最大支持一次性下载30万个文件）。

删除文件/文件夹

选中想要删除的文件/文件夹，可通过单击界面上方**更多**中的**删除**或右键单击**删除**，即可完成文件/文件夹的删除，支持批量删除。

文件同步

用户可以通过文件同步功能，将指定本地文件夹中的文件同步上传到存储桶中。文件同步功能支持设置手动同步、自动同步和定时同步。

⚠ 注意：

- 文件同步指在上传文件时，系统会自动识别本地目录与存储桶之间是否存在同样的文件，通过同步功能仅会上传存储桶中不存在的文件和变更过的文件。
- 目前同步功能，不会同步删除动作，即删除本地文件，不会同步删除 COS 存储桶中的文件。

- 目前仅支持将本地文件同步上传至存储桶，不支持逆向操作。

具体操作步骤如下：

1. 单击界面右上方的**工具箱>文件同步**。
2. 在弹窗中配置以下信息：
 - **本地文件夹**：指定将要同步上传的本地文件夹。
 - **存储桶目录**：上传到 COS 的存储桶目录。
 - **文件后缀过滤**：需要过滤的文件后缀。在同步时，将忽略带有该后缀的文件，如有多个，以英文 ; 分隔。
 - **同步类型**：可选择单次同步、自动同步或者定时同步，然后单击**开始同步**，即可开启文件同步功能。
 - 单次同步：即手动同步，在单击**开始同步**时，立即进行一次同步操作。
 - 自动同步：当本地目录有文件变更或新文件时，系统会将对应的文件自动同步上传到指定的存储桶中。
 - 定时同步：用户可以预先设置特定的时间点（例如每天或者每周某个时刻），以便系统在预定的时间点执行文件同步操作，确保文件保持同步。

 同步设置

 同步日志

本地文件夹 *

选择本地文件夹

更改

存储桶目录 *

选择存储桶

存储桶目录

文件后缀过滤 ⓘ

请输入需要过滤的文件后缀，在同步时，将忽略带有该后缀的文件，如有多个，以英文;分隔

同步类型

☒ 单次同步

☐ 自动同步

☐ 定时同步

仅支持将本地文件夹同步至存储桶目录，具体规则请查看 [说明文档](#)

开始同步

清除本地缓存

3. 单击**同步日志**，可查看文件的同步历史记录。

复制粘贴文件

在指定的存储桶或路径内，选中想要复制的文件/文件夹，可通过单击界面上方**更多**中的**复制**或右键单击**复制**，完成文件/文件夹的复制，复制成功后可在**其他存储桶或路径**中进行粘贴，支持批量复制粘贴。

⚠ 注意：

对于已复制的文件或文件夹，若其粘贴的目标路径中包含同名文件，则默认覆盖。

文件重命名

选中想要重命名的文件，右键选择**重命名**或单击文件右侧**更多操作**中的**重命名**，输入文件名并保存，即可完成文件的重命名。

❗ 说明：

文件夹无法进行重命名操作。

新建文件夹

在指定的存储桶或路径内，单击界面内的 **新建 > 新建文件夹**或在存储桶内空白处右键单击**新建文件夹**，输入文件夹名并确认，完成文件夹的创建。

⚠ 注意：

- 文件夹名称不能超过255个字符。
- 允许可用数字、英文和可见字符的组合。
- 用 / 分割路径，可快速创建子目录。
- 不允许以 / 开头和连续两个及以上 / 。
- 不允许文件夹名为空。
- 不允许直接以 . 作为文件夹名称。
- 文件夹无法进行重命名操作，请谨慎命名。

查看文件详情

可通过单击文件名或右键单击菜单中的**详情**来查看文件详情，文件详细信息包含文件名、文件大小、修改时间、访问权限、存储类型、ETag、Headers、指定域名、对象地址、创建临时链接。

文件详情

文件名

1.png

大小

0B (0B)

修改时间

2023-03-28 14:54:22

访问权限

继承权限

存储类型

标准存储

ETag

"d41d8cd98f00b204e9800998ecf8427e"

Headers

content-type: image/png
x-cos-hash-crc64ecma: 0
x-cos-version-id: null

指定域名

默认源站域名

对象地址

https://examplebucket-1251700003.cos.ap-guangzhou.myqcloud.com/1.png

创建临时链接

创建临时链接

生成文件链接

存在 COS 中的每个文件均可通过特定的链接来进行访问，若文件是私有读权限，则可通过请求临时签名的方式生成带有时效的临时访问链接。

生成文件链接有以下几种方式：

- 表格视图下，单击文件右侧的分享图标，可一键生成链接并复制。若文件为公有读权限，则链接不带签名永久有效。若文件为私有读权限，则链接带签名，2小时内有效。
- 选择文件后，右键单击**复制链接**，可一键生成链接并复制。若文件为公有读权限，则链接不带签名永久有效。若文件为私有读权限，则链接带签名，2小时内有效。
- 在文件详情中，单击**创建临时链接**，可以设置指定域名（例如开启了自定义 CDN 加速域名，则可设置）的临时链接、链接类型、有效时间。

💡 说明：

如果您使用的是子账号，且需要切换使用其他域名（例如自定义 CDN 加速域名），操作前需确保主账号已为子账号授予 `DescribeDomains` 权限。

自定义复制链接

文件名

picture.png

公共权限

私有读写

指定域名

默认源站域名

链接类型

☐ 复制不带签名的对象地址

☒ 复制带签名的临时链接，指定有效时间

2

小时

最长有效时间为 2 小时

URL

`https://examplebucket-125-...cos.ap-chengdu.myqcloud.com/picture.png?q-sign-algorithm=sha1&q-ak=AKIDeHJXFEG0MDdMRcZ6OumU9bBfjH2...&q-sign-time=1645687964;1645695164&q-key-time=1645687964;1645695164&q-header-list=&q-url-param-list=&q-signature=d557fef587dc0730753bf404bd84bbae97fd8162&x-cos-security-token=017b1f775fd143a23aa523c25f00fbe075e...001`

复制

关闭

分享文件/文件夹

通过单击操作栏中的分享图标可分享单个文件。对文件夹进行单击右键菜单中的分享选项，可分享单个文件夹，并且支持设置分享的有效时间。

说明：

- 只支持单个文件夹的分享，不支持批量文件共享。
- 多人共享文件，很容易造成文件版本混乱。建议开启存储桶的版本控制功能，可以回溯至文件的历史版本。

文件/文件夹分享

文件名

tomp4 等 1 个文件/文件夹

权限

☒ 只读 ☐ 读写

有效时间

1.5

天

子账号支持的最长有效时间为 30 天

提取码

☒ 自动生成 ☐ 自定义

e82548

请输入 6 位提取码

确定

关闭

参数	说明
权限	您可以设置分享文件夹的访问权限。 - 只读：通过访问链接可以拉取文件夹列表及下载文件夹中的文件。 - 读写：通过访问链接可以拉取文件夹列表、下载文件夹中的文件、上传文件至分享文件夹、新建文件夹。

有效时间	单位为分钟、小时或天。 - 使用密钥登录客户端：主账号有效期的取值范围为 1 分钟到 2 小时；子账号有效期的取值范围为 1 分钟到 30 天。 - 使用腾讯云账号登录客户端：支持设置的最长分享时间为 2 小时。默认值为当前账号允许的最长有效期。
提取码	6 位，默认系统自动生成，用户可自定义。支持输入数字、字母、符号。

⚠ 注意：

在链接有效期内，收到分享链接及提取码的用户都可以通过该链接访问该文件夹。

导出文件链接

COSBrowser 支持导出文件链接，在工具界面右上方单击工具箱图标，在工具箱弹窗中选择**导出文件链接**，然后选择文件所在的存储桶、需导出的文件夹路径（例如想要导出根路径下的 folder 文件夹，则填 folder/ ）和保存路径，最后单击**导出**即可。

导出文件链接

选择存储桶

examplebucket-1254287200

选择路径（选填）

folder1/

-

选择路径（选填）

folder2/

-

+

保存路径

D:\

更改

导出格式

☒ CSV

导出

关闭

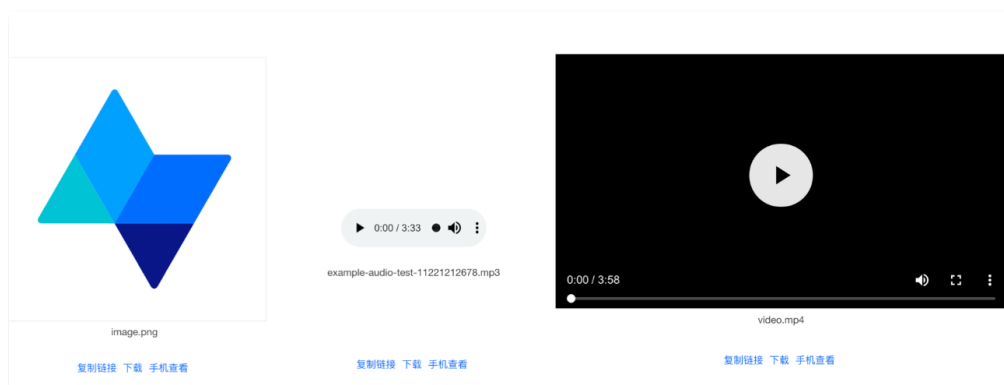
文件预览

COSBrowser 支持预览媒体类文件，目前支持图片、视频、音频。可通过双击媒体格式文件或单击右键菜单中的**预览**或**播放**选项，即可打开文件预览界面。在文件预览或播放界面，您可以选择：

- 复制链接：**生成文件访问链接并复制。
- 下载：**将文件下载至本地，若本地存在同名文件，则默认覆盖。
- 手机查看：**在预览界面会生成文件访问二维码，通过手机扫码可直接在手机上查看此文件。

⚠ 注意：

- 预览支持大多数图片格式，视频格式仅支持 mp4、webm，音频格式仅支持 mp3、wav。
- 文件预览会产生下行流量，请酌情使用。



搜索文件

可通过存储桶内右上方的搜索框，输入文件名进行搜索。COSBrowser 支持文件名前缀搜索和文件模糊搜索。

查看历史版本或文件碎片

- 当您的存储桶开启了版本控制后，可单击文件列表上方的**查看 > 历史版本**，查看文件的历史版本。历史版本列表中，支持按前缀搜索和清空所有历史版本（仅保留最新版本）。
- 当您在上传对象过程中，暂停或取消上传时可能产生文件碎片。您可单击文件列表上方的**查看 > 文件碎片**来查看文件碎片，支持按前缀搜索和清空所有文件碎片。



文件对比

在工具界面右上方单击工具箱图标，在工具箱弹窗中选择**文件对比**，然后选择本地文件夹与需要对比的存储桶，选择指定地域、存储桶、目录，最后单击**开始对比**即可。

对比本地与线上文件夹

本地文件夹

选择本地文件夹

更改

存储桶目录 *

选择存储桶

▼

存储桶目录

地域 *

格式如: ap-guangzhou

▼

注意：文件对比功能目前只支持按文件名比对文件大小。

开始对比

关闭

文件列表管理

单击存储桶名称即可进入文件列表，您可以在文件列表界面中对文件或文件夹进行相关操作，例如上传文件、创建文件夹、管理文件碎片、列出历史版本、搜索或刷新列表、设置文件夹权限、查看文件详情、预览、下载等操作。

文件列表支持两种布局，您可在文件列表的右上角切换所需布局。详情如下：

- 列表布局：文件及文件夹以列表的形式展示，默认布局。

上传

新建 ▼

下载

更多 ▼

查看 ▼

模糊搜索 ▼

搜索 examplebu

Q

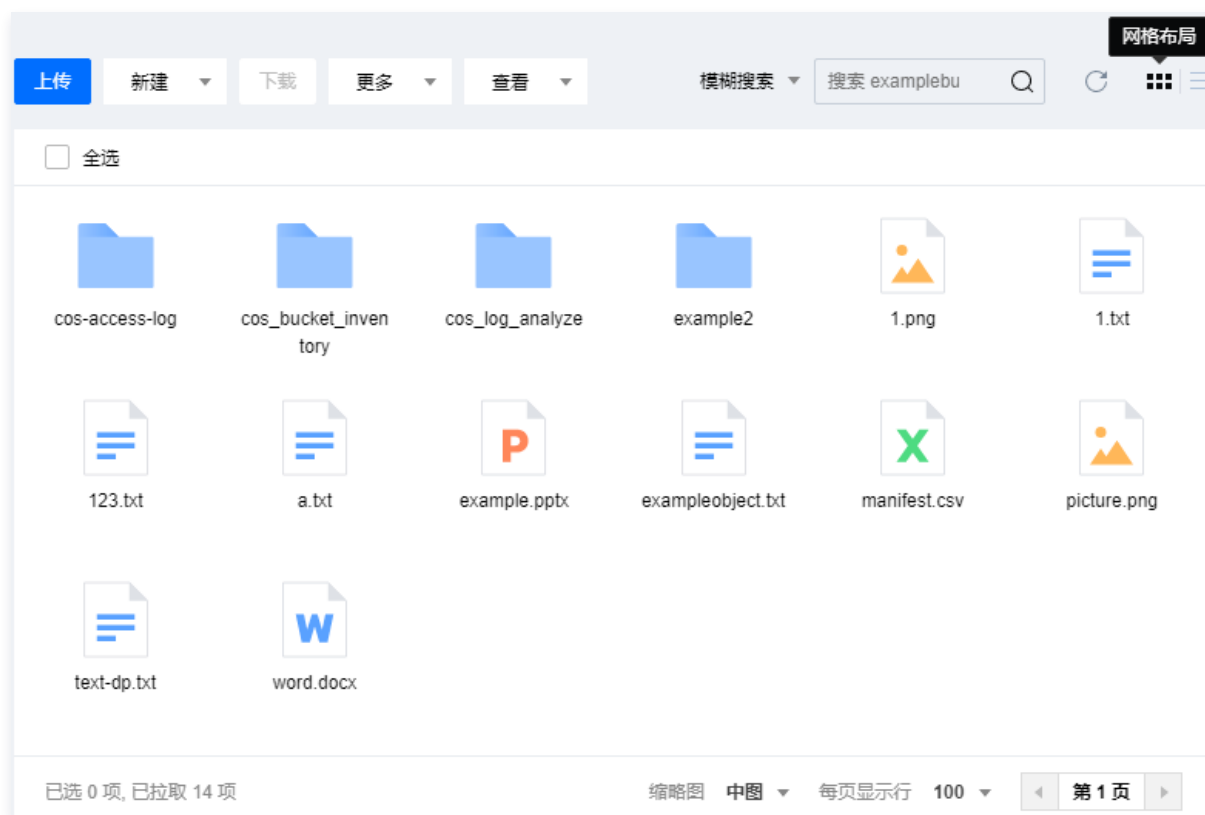
刷新

列表布局

三

名称	存储类型 ▼	大小 ↕	修改时间 ↕	操作
 cos-access-log	-	-	-	★   ...
 cos_bucket_inventory	-	-	-	★   ...
 cos_log_analyze	-	-	-	★   ...
 example2	-	-	-	★   ...
 1.png	标准	0B	2023-03-28 14:54:22	★   ...
 1.txt	归档	0B	2022-12-30 16:45:39	★  ...

- 网格布局：文件及文件夹以网格的形式展示，非默认布局。

**说明：**

在网格布局下，在窗口下方可选择打开缩略图功能，该功能由数据万象 CI 提供技术支持，开启后会产生数据万象 [图片处理费用](#)、[媒体处理费用](#)。

数据处理

视频转码

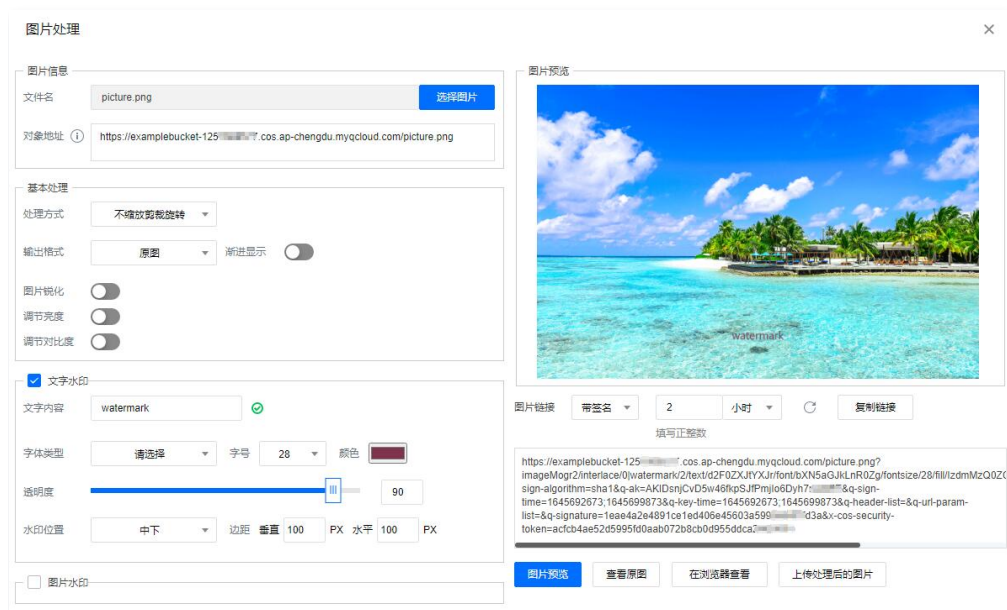
在工具界面右上方单击工具箱图标，在工具箱弹窗中选择**视频转码**，然后选择已开启媒体处理服务的存储桶，然后单击**创建转码任务**。选择需转码的媒体文件和转码模板、输入转码后的文件名和存储的路径。指定地域、存储桶、目录，最后单击**转码**即可。



图片处理

工具中的图片处理功能可支持缩放、裁剪、旋转等基本图片处理和文字水印、图片水印，并支持生成处理后的图片链接。

选择需使用图片处理功能的存储桶，然后在工具界面右上方单击工具箱图标，在工具箱弹窗中选择**图片处理**。在图片处理弹窗中，选择需处理的图片文件，配置功能选项后，最后单击**图片预览**即可生成处理后的图片链接。



软件设置

高级配置	说明	如何操作
通用配置	- EndPoint/Service域名：获取存储桶列表 API 时自定义请求域名。默认是service.cos.myqcloud.com。 - Bucket域名模板：调用操作存储桶和对象的 API 时自定义请求域名。可以使用模板，如"{Bucket}.cos.{Region}.myqcloud.com"，即在调用 API 时会使用参数中传入的Bucket 和 Region 进行替换。 - 自动切换备份域名：如果 Bucket 域名未配置或配置了COS 默认域名，在请求出现网络错误时，会自动走备份域名重试 1 次。 - 使用 HTTPS：若要为您的存储桶域名使用HTTPS，您需要在您的网站上安装SSL 或TLS 证书。选择该选项会使请求更加安全。 - 校验 HTTPS 证书：请求时是否校验https证书。	- 选择高级设置 > 通用。 - 根据配置项说明，按需设置即可。
设置网络代理	COSBrowser 默认会使用系统配置的代理来尝试网络连接，请确保您的代理配置正常或请停用无法连接互联网的代理配置	- 选择高级设置 > 代理。 - 设置网络代理来进行网络连接。
设置上传/下载的文件并发数	COSBrowser 支持设置上传、下载的文件并发数。	- 选择高级设置 > 下载/上传。 - 设定批量传输的并发数。
设置上传/下载的分块并发数	COSBrowser 支持分块上传、下载文件，当传输的文件超过一定大小时，会默认使用分块的方式传输	- 选择高级设置 > 下载/上传。 - 设定分块传输的并发数。
设置上传/下载失败重试数	COSBrowser 在文件传输的时候，会默认重试失败的任务	- 选择高级设置 > 下载/上传。 - 设定传输失败的重试次数。
设置单线程上传/下载限速	COSBrowser 支持设置单线程上传限速和单线程下载限速。总上传（或下载）限速 = 单线程上传（或下载）限速 x 文件并发数 x 分块并发数	- 选择高级设置 > 上传（或下载）。 - 设置单线程上传（或下载）限速，单位为MB/s。
设置上传二次校验	COSBrowser 支持在上传后进行二次校验，检查线上文件大小和状态是否正确	- 选择高级设置 > 上传。 - 勾选上传完二次校验。

查看本地日志	COSBrowser 会记录使用者的操作，以 cosbrowser.log 的日志形式保存在本地	1. 选择高级设置 > 关于。 2. 单击本地日志，系统将打开本地日志所在目录。
--------	--	---

移动端使用说明

安装与登录

最近更新时间：2024-01-29 15:06:01

下载与安装

软件下载

支持平台	系统要求	下载地址
Android	Android 5.0 及以上	Android
iOS 移动端	iOS 12及以上	iOS

软件安装

COSBrowser 移动端目前已覆盖绝大多数应用平台，例如应用宝、App Store 等，可通过单击上述下载地址或自行前往应用平台下载。

登录方式

COSBrowser 移动端支持以下三种登录方式：

- **微信快捷登录：**通过微信创建或关联了指定微信的腾讯云账号，可以使用微信快捷登录方式快速登录 COSBrowser。
- **邮箱登录：**通过邮箱创建或关联了指定邮箱的腾讯云账号，可以通过输入邮箱账号密码进行登录。
- **永久密钥登录：**用户可使用云 API 密钥 SecretId 和 SecretKey（不支持项目密钥）进行登录，该密钥可在访问管理控制台的 [API 密钥管理](#) 页面中创建和获取，登录后将永久保持该账号的登录状态。

ⓘ 说明

- 若用户的腾讯云账号为 QQ 账号所创建的，同样可使用微信快捷登录方式登录，只需在跳转的微信小程序界面选择 QQ 登录即可。
- 子账号用户可以使用密钥或微信快捷登录方式登录，选择微信登录只需在跳转的微信小程序界面选择子账号即可。

移动端功能列表

最近更新时间：2024-11-26 16:32:52

COSBrowser App 是腾讯云对象存储（Cloud Object Storage，COS）官方出品的移动应用，可帮助您更加方便、快捷地对 COS 上的资源进行查看和管理，随时随地掌控云端资源。下面将为您介绍 COSBrowser App 所支持的功能。

数据监控

操作名	操作描述
用量概览	支持展示近期的数据使用情况（需跳转至小程序进行查看）。
存储桶监控	支持从存储桶维度查看近期的数据使用情况（需跳转至小程序进行查看）。

存储桶管理与操作

操作名	操作描述
查看存储桶列表	支持按照地域的分类查看存储桶列表。
添加访问路径	若您使用无访问存储桶列表权限的子账号进行登录，可以通过添加访问路径的方式进行访问。
创建存储桶	支持创建存储桶。
搜索存储桶	支持在存储桶列表页按照关键字进行模糊搜索。
查看存储桶基本信息	您可以在移动端查看存储桶名称、地域、创建时间及是否开启多 AZ 等基本信息。
存储桶权限管理	支持修改存储桶的公共权限和用户权限。
开启全球加速	支持对存储桶开启全球加速功能。
存储桶传输配置	您可以通过该功能设置上传、下载使用的域名。

文件操作与管理

操作名	操作描述
创建文件夹	支持在存储桶内创建文件夹。
删除文件夹	支持删除当前目录以及子目录下的所有文件。
文件上传	您可以通过移动端 COSBrowser 将本地文件、远程文件、其他 App 的文件以及文件管理器上传到 COS 上，也可以在上传时设置文件的存储类型、访问权限、加密方式、对象标签、元数据等信息。
文件备份	COSBrowser 提供了自动备份的功能，在开启备份之后，COSBrowser 会自动将您图库中的文件备份到特定的桶里面。
文件下载	支持将存储桶中的文件下载到 App 或者保存到本地图库。
批量操作	支持批量上传、下载、删除、复制、移动存储桶中的文件。
文件预览	支持预览存储桶中的图片、视频、音频、文档等多种格式的文件。
文件在线解压	支持对 zip、tar、gz 这三种压缩格式的文件进行在线解压。
文件分享	COSBrowser 提供了文件夹和文件分享的功能，方便您在短时间内收集或共享存储桶内数据给其他用户。
文件重命名	支持重命名存储桶中的文件。

文件搜索	支持模糊搜索、按照文件夹、图片、视频、文档、音频等类型进行搜索。
排序或者筛选对象	当前支持对存储桶内的文件按文件名、大小、修改时间进行排序，按存储类型进行筛选。
文件权限管理	支持基于文件维度的访问权限设置，且该权限优先级高于存储桶的访问权限。

资源包管理

操作名	操作描述
查看资源包列表	您可查看您当前账号所拥有的资源包使用情况。
资源包购买	支持在移动端购买资源包。
资源包续费	支持在移动端对资源包续费。
资源包升级	支持在移动端对资源包升级。

存储桶管理与操作

最近更新时间：2024-06-28 10:22:31

说明：

此处功能介绍以移动端 iOS 2.7.6 版本为例，其他版本差异，请参见 [更新日志](#)。

查看存储桶列表

COSBrowser 移动端以地域的维度展示存储桶列表，您可通过地域的分类查看存储桶。点击界面中的资源图标即可查看已创建的存储桶。

添加访问路径

若您使用无访问存储桶列表权限的子账号进行登录，可以通过点击存储桶列表页右上角的**添加访问路径**，添加指定的路径进入存储桶或目录管理资源。

说明：

仅子账号支持该功能。

操作步骤

1. 进入存储桶列表页，点击右上角的+。
2. 在弹出的操作列表中，点击**添加访问路径**，输入已被主账号授权的访问路径即可。

创建存储桶

您可以通过 COSBrowser 移动端来创建存储桶，创建存储桶需要指定名称、地域、访问权限。

操作步骤

1. 在存储桶列表中，点击右上角+。
2. 在弹出的操作列表中，点击**创建存储桶**。
3. 在弹出的创建存储桶页面，配置如下信息：
 - **名称**：请输入自定义的存储桶名称。设置后不可修改。命名说明请参见存储桶的 [命名规范](#)。
 - **所属地域**：请选择与您业务（或用户数量）相对集中的物理区域所对应的 COS 地域，设置后不可修改。地域的更多信息请参见 [地域和访问域名](#)。
 - **访问权限**：存储桶默认提供三种访问权限：私有读写、公有读私有写和公有读写，设置后仍可修改。详细信息请参见 [存储桶访问权限](#)。
4. 确认信息无误后，点击**确定**，即可创建存储桶。

在存储桶列表界面中，您可以看到刚才已创建的存储桶。

搜索存储桶

如果存储桶数量较多，您可以在存储桶列表页面上方的搜索栏，输入存储桶的名称进行查询，目前支持存储桶名称的模糊匹配查询。

查看存储桶基本信息

1. 在存储桶列表页，找到想要操作的桶，点击右侧的…。
2. 在弹出的操作列表中，点击**详情**，即可查看存储桶的基本信息。

基本信息包含存储桶名称、地域、创建时间及是否开启多 AZ。

存储桶权限管理

您可以通过对象存储移动端来设置或修改存储桶的访问权限。COS 支持设置两种权限类型：

- **公共权限**：私有读写、公有读私有写和公有读写。关于公共权限的说明，请参见存储桶概述中的 [权限类别](#)。
 - **用户权限**：主账号默认拥有存储桶所有权限（即完全控制）。另外 COS 支持添加子账号有数据读取、数据写入、权限读取、权限写入，甚至完全控制的最高权限。
1. 进入存储桶列表页，找到您需要设置或修改访问权限的存储桶，点击右侧的…。

2. 在弹出的操作列表中，点击**权限**，即可进入存储桶权限页面。

修改公共权限

在存储桶权限页面，点击公共权限配置项，即可修改公共权限。

设置用户权限

在存储桶权限页面，点击添加用户，即可设置存储桶的访问权限。

编辑或删除用户权限

选择要删除或编辑的用户权限并向左滑动，可弹出编辑、删除按钮，点击对应的按钮即可对该条用户权限进行编辑或者删除。

开启全球加速

您可以通过 COSBrowser 为您的存储桶开启全球加速功能，实现全球各地用户快速访问您的存储桶，提升您的业务访问成功率和业务稳定性。全球加速功能可以加速您的上传和下载。

! 说明：

开启全球加速功能，不会影响原有的存储桶默认域名，您仍然可以正常使用。

操作步骤

1. 进入存储桶列表页，找到需要配置全球加速功能的存储桶，点击存储桶右侧的...
2. 在弹出的操作栏中，点击**传输**，进入传输列表页面。
3. 点击全球加速配置项，根据实际需求，修改全球加速的开启或关闭状态。

存储桶传输配置

如果您想让某个桶通过自定义域名进行传输，那么您可以在该页面进行配置。

设置上传域名

当存储桶开启了全球加速功能，您可以指定该存储桶上传文件时使用的全球加速域名，设置上传域名完成后，存储桶会优先使用该域名进行上传。

设置下载域名

当存储桶设置了自定义源站域名，您可以指定该存储桶下载文件时使用自定义源站域名，设置下载域名完成后，存储桶会优先使用该域名进行下载。

文件管理与操作

最近更新时间：2024-08-29 10:22:31

说明：

此处功能介绍以移动端 iOS 2.8.2 版本为例，其他版本差异，请参见 [更新日志](#)。

新建文件夹

COSBrowser 移动端提供了在存储桶中创建文件夹的功能，操作步骤如下：

1. 在存储桶的文件列表页，点击右上角+。
2. 在弹出的操作列表，点击**新建文件夹**。
3. 在新建文件夹页面，填入文件夹名称，点击**确定**，即可创建成功。

删除文件夹

注意：

删除文件夹会删除该文件夹以及所有子目录下的文件。

操作步骤

1. 找到要删除的文件夹，点击文件夹右侧的...
2. 在弹出的操作列表，点击**删除**，即可删除该文件夹。

上传文件

您可以通过 COSBrowser 移动端将本地文件、远程文件、其他 App 的文件以及文件管理器上传到对象存储（Cloud Object Storage，COS）上，在上传文件时可设置文件的存储类型、访问权限、加密方式、对象标签、元数据等信息。

上传图片和视频

COSBrowser 支持将图库的图片或者视频批量上传到 COS。

操作步骤

1. 进入存储桶列表页或者文件列表页，点击右上角+，弹出操作列表。
 2. 在操作列表中，点击**上传图片**。
 3. 在弹出图库的文件列表，选中要上传的文件，点击**下一步**。
 4. （可选）配置上传参数，COSBrowser 支持在上传时设置文件的属性，目前支持设置以下属性：
 - **存储类型**：根据不同的业务场景，您可以为不同的对象设置不同的存储类型，默认存储类型为标准存储。关于存储类型的介绍，请参见 [存储类型概述](#)。
- ## 注意：

若您的存储桶开启了多 AZ 配置，则**存储类型**只能选择为拥有多 AZ 特性的存储类型，例如标准存储（多 AZ）；若同时还开启了智能分层存储配置，则还可选择智能分层存储（多 AZ）类型。
- **访问权限**：您可以为不同的对象设置不同的访问权限，默认访问权限为继承权限（即继承存储桶权限）。关于访问权限的更多信息，请参见 [访问控制基本概念](#)。
 - **服务端加密**：您在上传对象的同时，配置对象的服务器加密属性。腾讯云 COS 将为上传的对象添加数据加密的保护策略，帮助您在数据写入前自动加密，并在您访问该数据时自动解密。目前支持 SSE-COS、SSE-KMS，其中 SSE-KMS 仅支持部分地域，详情请参见 [服务端加密概述](#)。
 - **对象标签**：对象标签由标签的键（tagKey）、标签的值（tagValue）与“=”相连组成，例如 group = IT。您可以对指定的对象进行标签的设置、查询、删除操作。

- **元数据**：对象元数据是服务器以 HTTP 协议传 HTML 资料到浏览器前所送出的字符串，又称为 HTTP Header。通过修改 HTTP Header，可以改变页面的响应形式，或者传达配置信息，例如修改缓存时间。修改对象的 HTTP Header 不会修改对象本身。具体信息请参见 [自定义对象 Headers](#)。

5. 点击**上传**，即可对图库中的图片和视频进行批量上传。

上传文件链接

COSBrowser 支持以文件链接的方式上传文件，每次进入 App 时会检查当前剪贴板，如果剪贴板上有合法的文件链接将会弹出上传链接的提示，点击**立即上传**即可上传文件。

您也可以通过下面的步骤主动上传文件链接：

1. 进入存储桶列表页或者文件列表页，单击右上角+，弹出操作列表。
2. 单击**上传链接**，将要上传的链接粘贴到文本框里面，然后选择上传路径，最后点击**上传**，即可将该文件上传到 COS。

上传第三方分享

您也可以将其他 App 的文件分享到 COSBrowser 进行上传。

⚠ 注意：

该功能需要第三方应用支持将文件分享到其他 App。

操作步骤

以 QQ App 为例：

1. 在 QQ App 中，先点击文件进行预览，再点击右上角的...，**选择其他应用**。
2. 在应用列表中找到 COSBrowser，点击 COSBrowser 即可进行上传。

上传文件管理器文件

COSBrowser 支持将系统文件管理器的文件上传到 COS（iOS 系统对应的是文件 App，Android 系统对应的是文件管理 App）。

操作步骤

1. 进入存储桶或者文件列表页，点击+。
2. 在弹出的操作列表，点击**上传文件**。
3. 在弹出的文件管理器页面，点击需要上传的文件，即可上传到 COS。

文件备份

COSBrowser 提供了自动备份的功能，在开启备份后，COSBrowser 会自动将您图库中的文件备份到指定的存储桶中。为了方便您管理备份文件，COSBrowser 将备份数据作为一个单独的模块放在了主页面。

⚠ 注意：

- 图库备份功能只支持主账号。
- 相册模块只显示图片和视频文件，如果想要查看完整文件，请前往存储桶列表，搜索备份存储桶进行查看。

设置备份

前往**个人 > 相册备份**，点击开启自动备份照片或视频。

- **自动备份照片**：开启之后会备份图库中的所有图片。
- **自动备份视频**：开启之后会备份图库中的所有视频。
- **仅在 WIFI 环境下备份**：开启之后只会在 Wi-Fi 环境下备份文件。
- **所属地域**：选择备份地域后，我们会在该地域默认创建一个名称格式为 “from-phone-日期-APPID” 的存储桶。

⚠ 注意：

所属地域只有在开启备份前可设置，保存后将不可修改。

- **开启智能存储：**该项适用于开启了智能分层存储的存储桶，开启智能存储后，图库的文件会以智能分层的方式上传到 COS 上，COS 可根据智能分层存储类型对象的访问频次自动在标准存储层和低频存储层两个存储层之间变换，无数据取回费用，可降低用户的存储成本。更多详情请参考 [智能分层存储简介](#)。

管理备份文件

COSBrowser 支持对文件进行批量上传和下载操作。

操作步骤

1. 进入**相册**页面，单击右上角的批量操作图标，弹出操作栏。
2. 选中要批量操作的文件，点击对应的操作按钮，即可对选中的文件进行**下载**或者**删除**操作。

将文件添加到备份存储桶

您也可以将其他存储桶的图片或者视频文件添加到备份存储桶中，方便您在**相册**模块快速预览。

操作步骤

1. 找到要添加的图片或者视频文件，单击右侧的...
2. 在弹出的操作列表，点击**添加至相册**。
3. 添加完成后，再次进入**相册**模块，即可看到该文件。

文件下载

COSBrowser 支持将 COS 上的文件下载到本地，也支持将图片保存到系统图库。

下载到 App

COSBrowser 提供了多种下载入口，方便您随时随地下载文件。

- 方式一：
 1. 进入文件列表页面，找到要下载的文件，单击右侧的...
 2. 在弹出的操作列表，点击**下载**即可。
- 方式二：

进入文件详情页面，单击右上角的...，弹出操作列表，单击**下载**即可。
- 方式三：

进入文件预览页面，单击右上角的...，弹出操作列表，单击**下载**即可。

保存到图库

COSBrowser 支持将图片保存到本地图库，操作步骤如下：

1. 进入文件列表页，找到要下载的文件，单击右侧的...，弹出操作列表。
2. 单击**下载**，弹出下载选项，选择**保存到相册**。

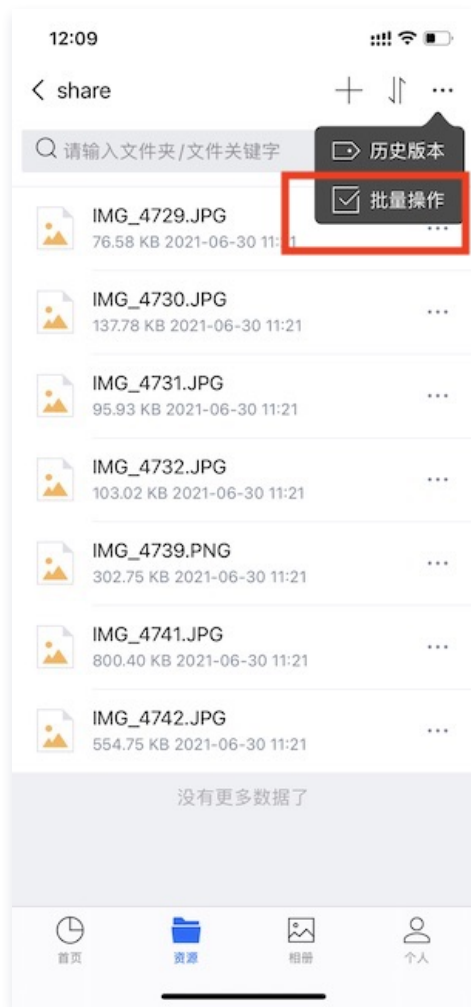
批量操作

COSBrowser 支持批量下载、删除、复制、移动存储桶中的文件。

批量下载

1. 点击存储桶，进入文件列表页，单击右上角...

2. 在弹出的操作菜单，点击**批量操作**。



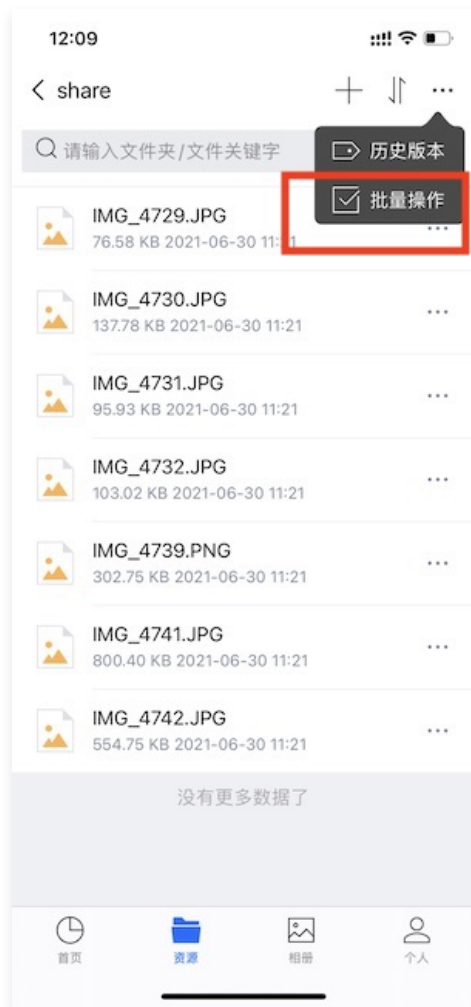
3. 批量选择文件，点击底部操作栏的**下载**，即可将选中的文件批量下载至 App。

您还可以点击文件列表页右上角的传输按钮，进入传输列表页查看相关任务。

批量删除

1. 点击存储桶，进入文件列表页，单击右上角...

2. 在弹出的操作菜单，点击**批量操作**。



3. 批量选择文件，点击底部操作栏的**删除**，即可将选中的文件批量删除。

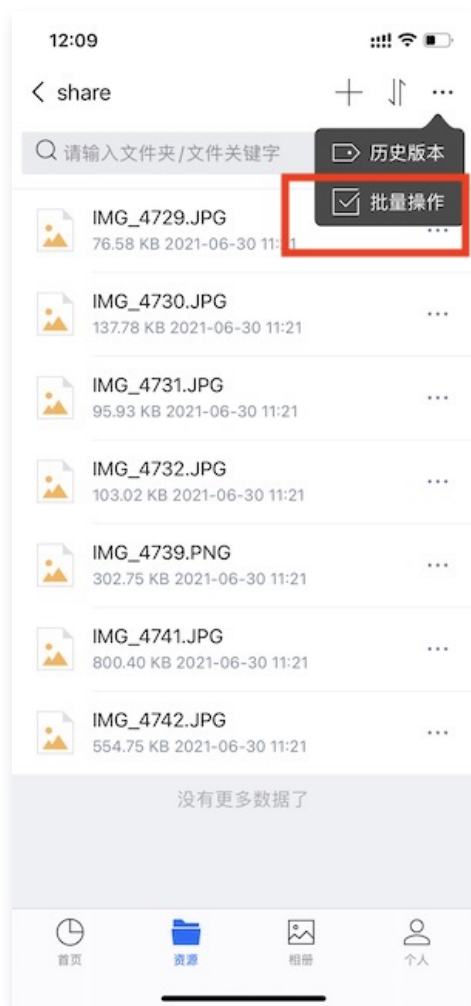
批量复制

说明：

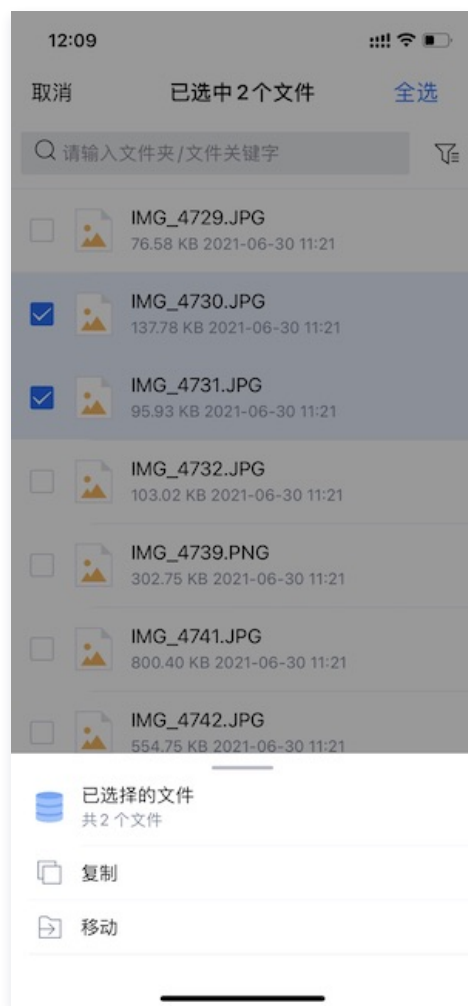
COSBrowser 在复制的过程中，也会将源文件的 ACL、Policy、Tagging 以及元数据等相关信息一同复制。

1. 点击存储桶，进入文件列表页，单击右上角...

2. 在弹出的操作菜单，点击**批量操作**。



3. 批量选择文件，点击底部操作栏的**更多 > 复制**，即可将选中的文件批量复制到新的存储桶或者文件夹。



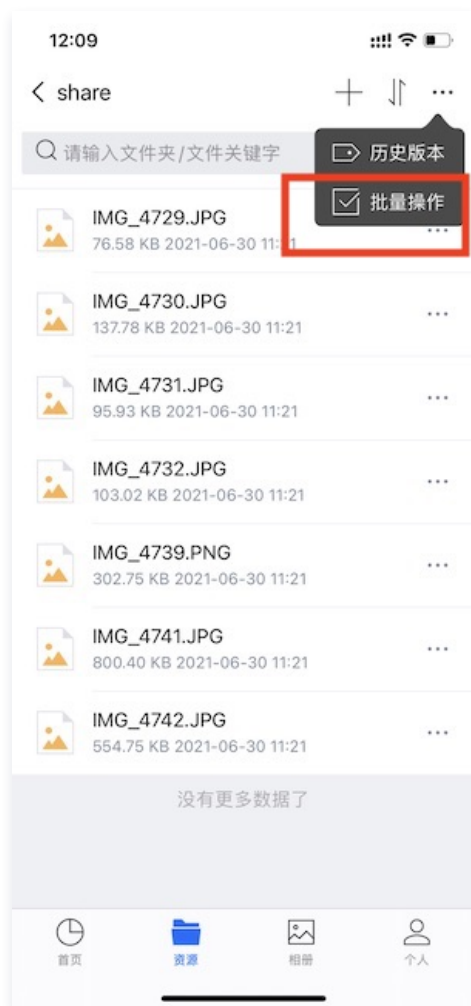
批量移动

说明：

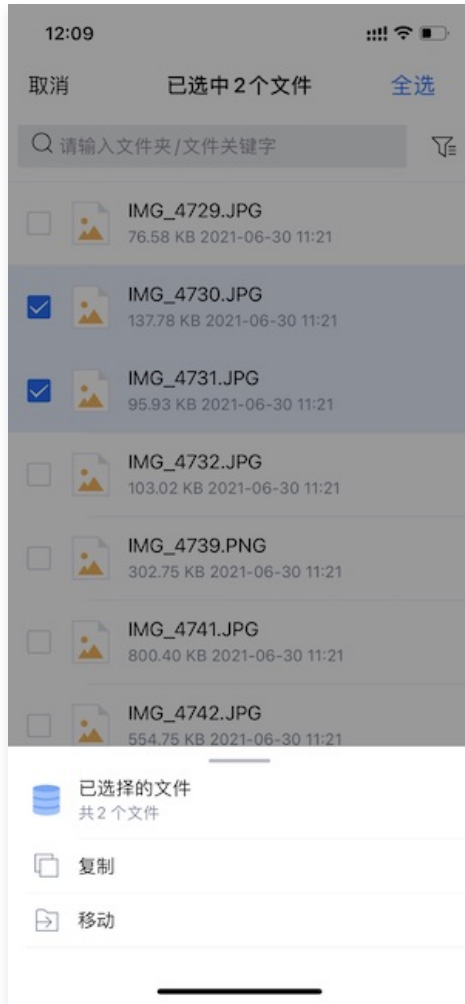
COSBrowser 在移动的过程中，也会将源文件的 ACL、Policy、Tagging 以及元数据等相关信息一起移动过去。

1. 点击存储桶，进入文件列表页，单击右上角...

2. 在弹出的操作菜单，点击**批量操作**。



3. 批量选择文件，点击底部操作栏的**更多 > 移动**，即可将选中的文件批量移动到新的存储桶或者文件夹。



文件预览

COSBrowser 支持图片、音频、视频、文档等多种文件格式的预览，同时还支持文件在线解压的功能。

图片在线预览

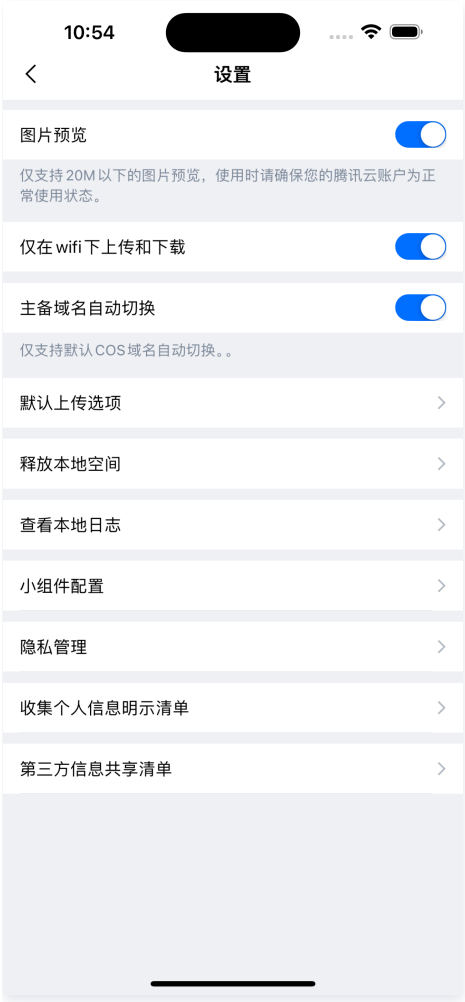
COSBrowser 支持在不下载图片的情况下实现图片的在线预览。

开启图片预览

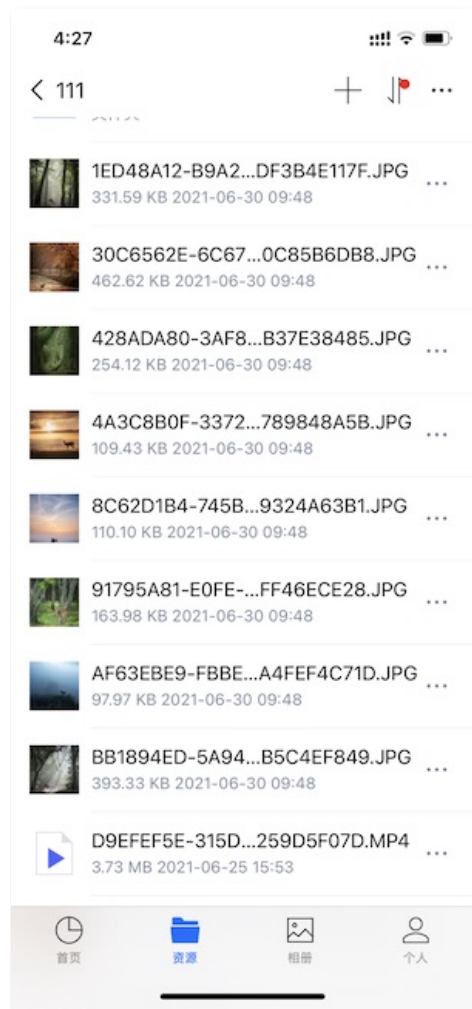
您可以通过以下三种方式实现图片预览：

1. 全局开启

前往**个人 > 设置**，打开图片预览的开启按钮，即可全局开启**图片预览**功能。



开启之后的效果如右图所示：



2. 一次性全局开启

在您首次启动 App 并且进入文件列表页时，App 会检测当前列表页是否有图片。如有，则弹出开启图片预览的提示窗，您可以点击**开启预览**，快速打开**图片预览**功能；如不需要，可点击**取消忽略**。后续如需使用，可前往**个人 > 设置**中手动开启。



3. 单个图片开启预览

直接点击想要查看大图的照片，即进入了大图查看模式、可以双指缩放查看图片的细节，那么进入图片详情的时候，会提示您是否开启预览，点击**预览**，即可实现对单个图片的预览。

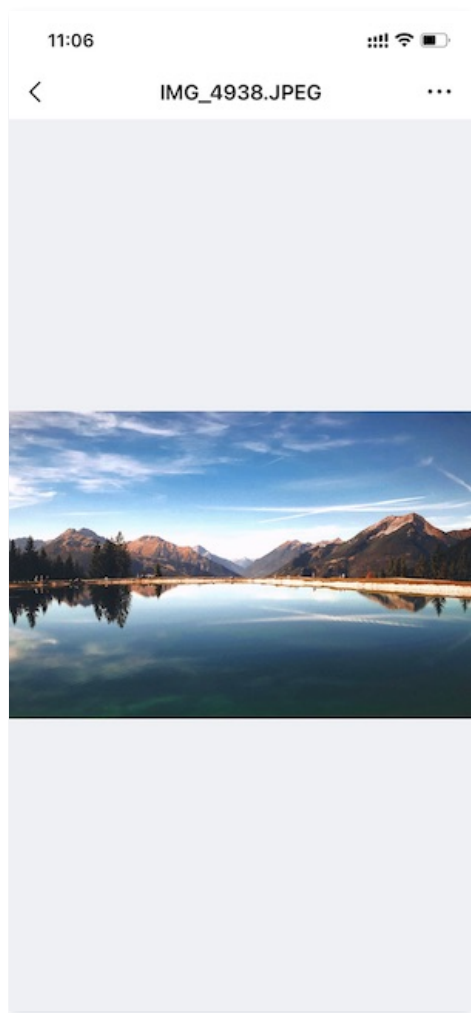
说明：

该开关不会修改全局**图片预览**开关的状态。

大图模式

COSBrowser 提供了大图模式功能，可以更加清晰的查看图片细节。

在文件列表页点击图片，即可进入大图查看模式，在该模式下您可以双指缩放查看图片的细节。

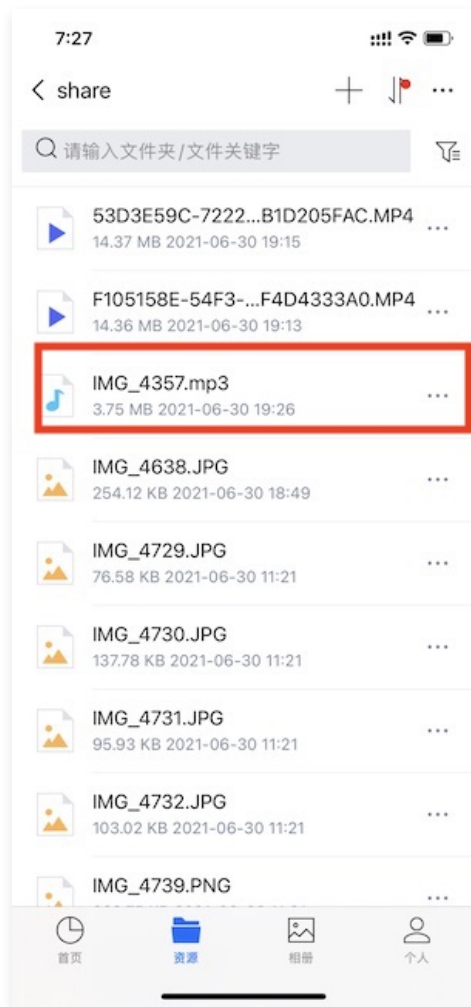


音频播放

COSBrowser 支持 mp3、ogg、aac、wma、wav、ape、flac 这几种音频格式文件的播放。

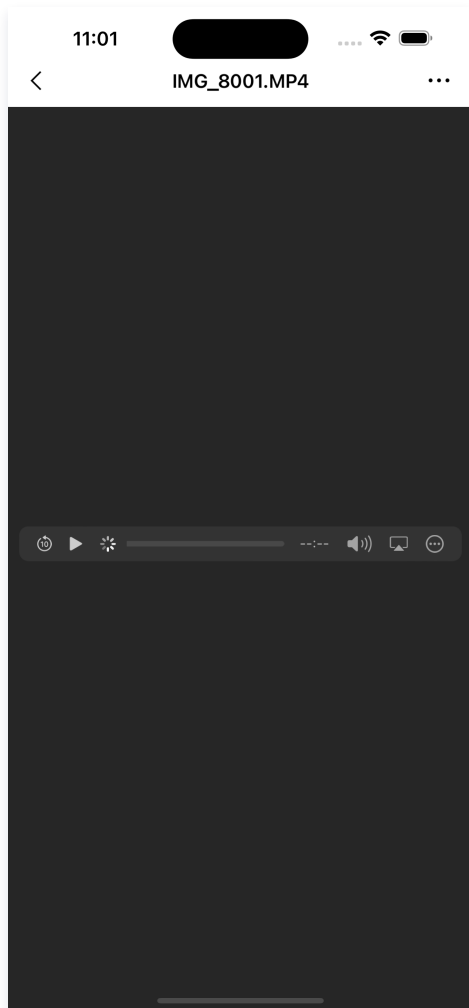
操作步骤

1. 进入文件列表页，点击音频文件，进入预览页面。



2. 点击播放按钮，进行播放。

在播放的过程中，可以进行快进、暂停。

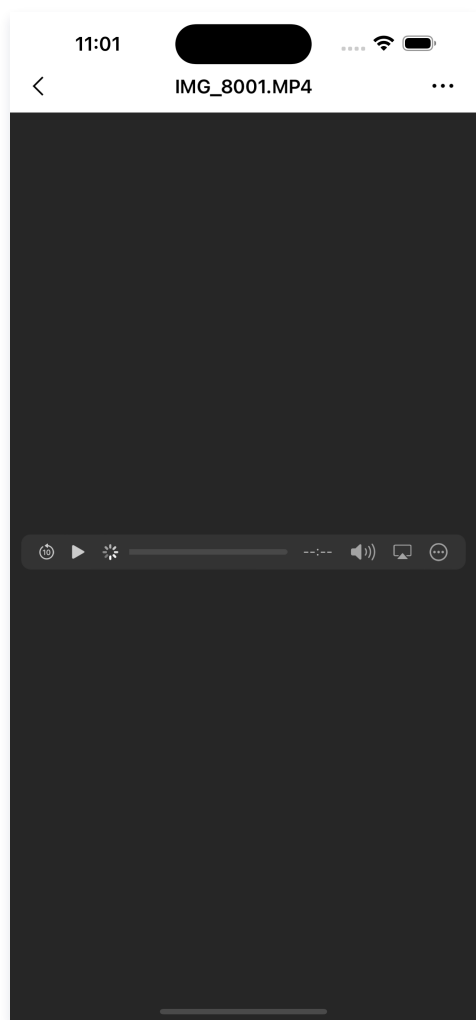


视频播放

COSBrowser 提供了简单的视频在线播放功能，支持 avi、wmv、mpeg、rm、rmvb、mkv、mov、qt、mp4 等多种格式，您可以在观看视频的过程中快进、暂停，同时也支持在 App 进入后台后，以悬浮窗的方式播放。

操作步骤

1. 进入文件列表页，点击需要播放的视频文件，进入文件详情。
2. 点击播放，进行播放。
在播放的过程中，可以进行快进、暂停。



说明：
在 App 进入后台后，仍然可通过悬浮窗的方式观看视频。

文档预览

COSBrowser 支持对 PDF 等多种格式的文件进行预览。

操作步骤

1. 点击要预览的文档，如果尚未开启文档预览功能，则会进入文档预览功能配置页面，点击**立即开启**，即可开启文档预览。



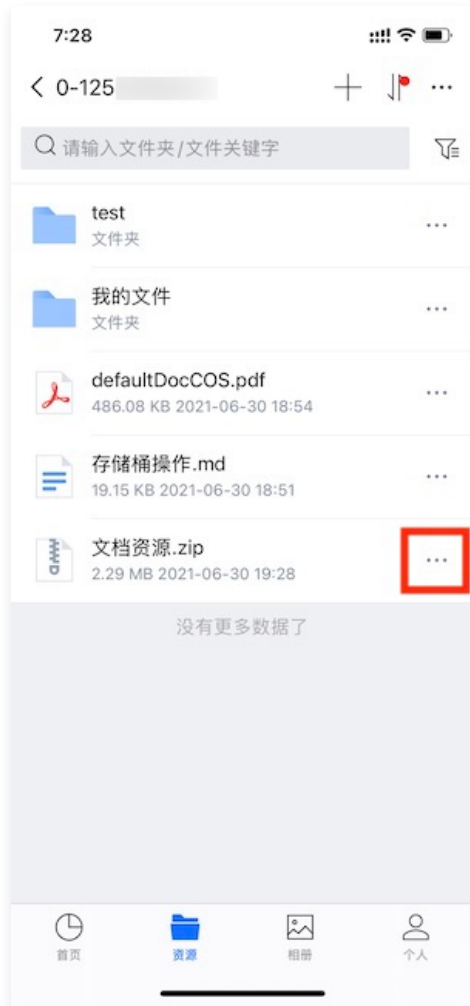
2. 开启成功后，返回文件列表页面，单击文件即可进入预览页面，向上滑动可查看下一页。



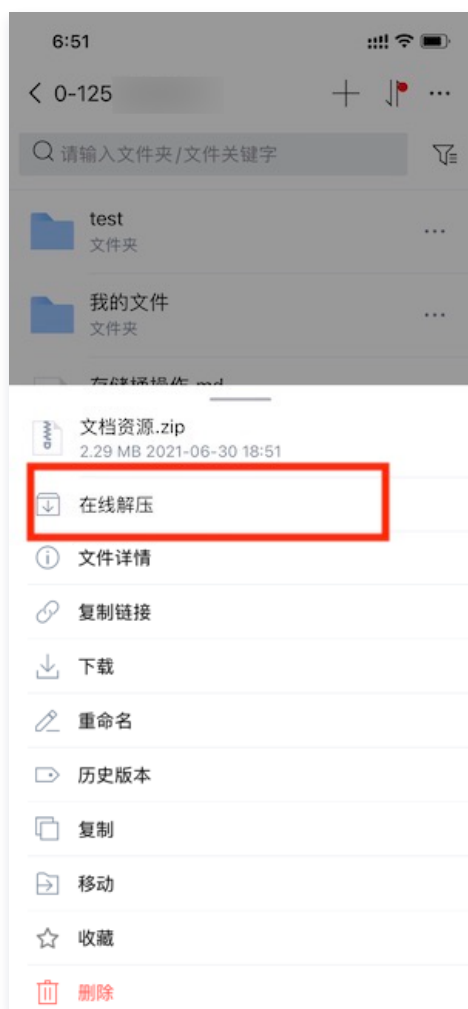
文件解压

COSBrowser 支持对 zip、tar、gz 这三种压缩格式的文件进行在线解压，解压后的文件会存放在当前目录下。

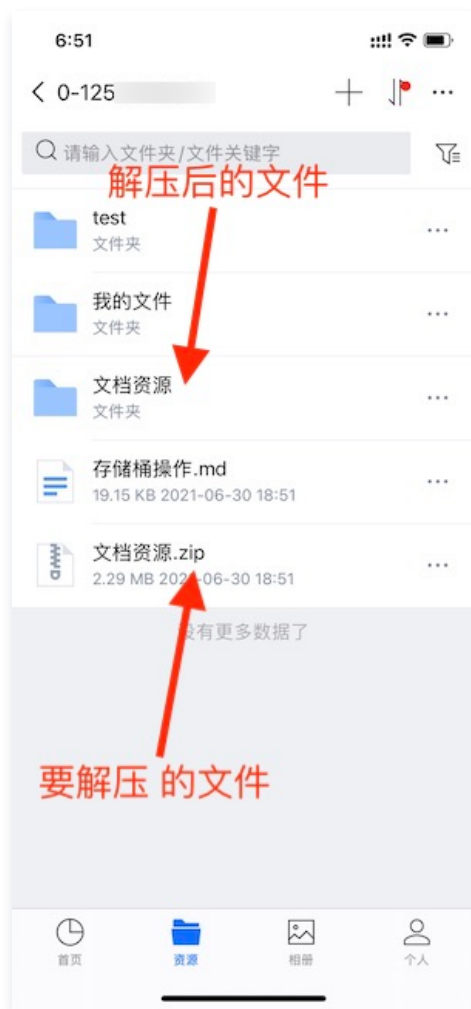
1. 进入文件列表页，点击压缩文件右侧的...



2. 弹出操作列表，点击在线解压。



3. 解压成功后，当前目录下会出现解压后的文件。



文件分享

COSBrowser 提供了文件夹和文件分享的功能，方便您快速收集数据或将存储桶内数据共享给其他用户。

分享文件夹

说明：

- 只支持单个文件夹的分享，不支持批量文件共享。
- 多人共享文件，很容易造成文件版本混乱，建议开启存储桶的版本控制功能，可以回溯至过去的文件版本。

生成分享二维码/链接

具体操作步骤如下：

1. 点击文件夹右侧的...选项，弹出操作列表，点击分享。



2. 在弹出的分享页面中，可选择二维码分享或链接分享。



3.（可选）配置分享参数。配置项说明如下。可不设置，保持默认设置。



- **权限** 您可以设置分享文件夹的访问权限。
 - 只读：通过访问链接可以拉取文件夹列表及下载文件夹中的文件。
 - 读写：通过访问链接可以拉取文件夹列表、下载文件夹中的文件、上传文件至分享文件夹、新建文件夹。
- **有效时间**：单位为分钟、小时或天。主账号和子账号通过**微信快捷登录**方式登录移动端，默认设置有效期为2小时；子账号通过**密钥登录**方式登录移动端，有效期为24小时。移动端的有效时间无法设置。
- **提取码**：6位，默认系统自动生成，用户可自定义。支持输入数字、字母、符号。

4. 点击**生成链接**或者**二维码分享**，即可生成对应的分享链接或者二维码。

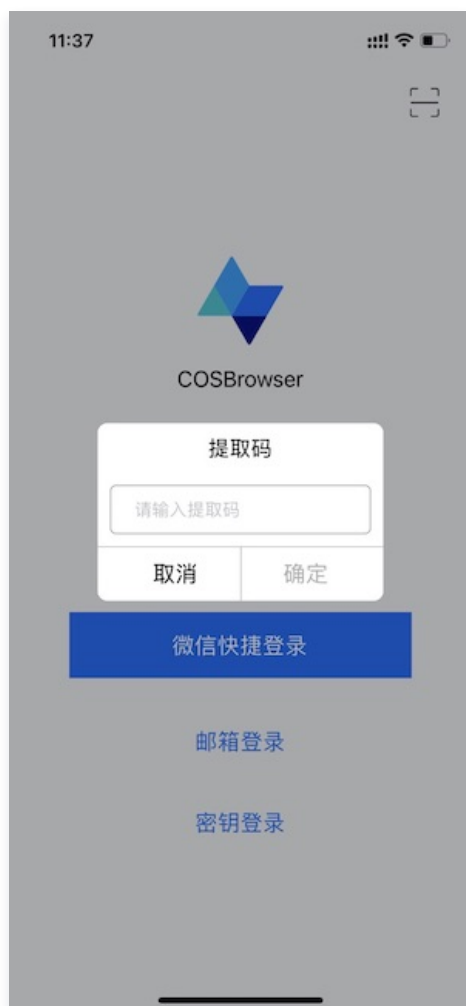


查看他人分享的文件夹

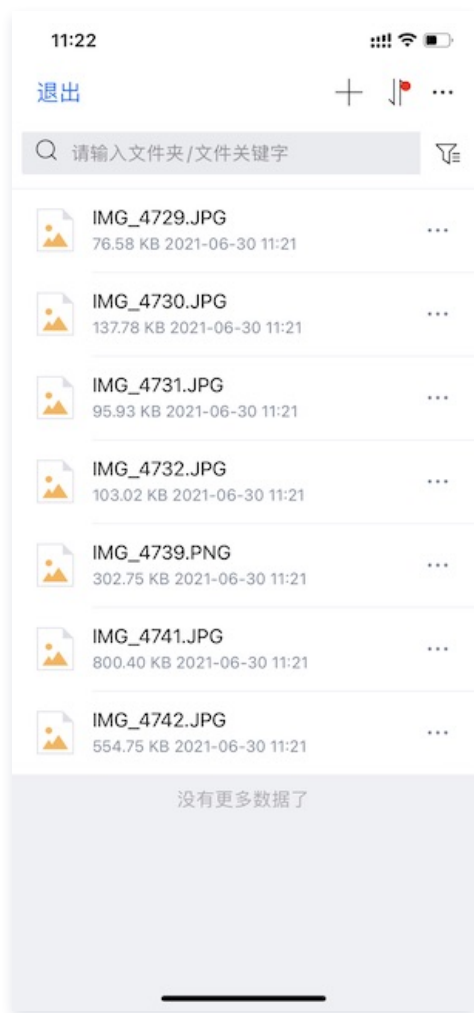
您可以通过移动端、桌面端、浏览器三种方式打开他人分享过来的文件夹。

方式一：在移动端查看

1. 在登录页或个人页，点击扫一扫扫描分享二维码。

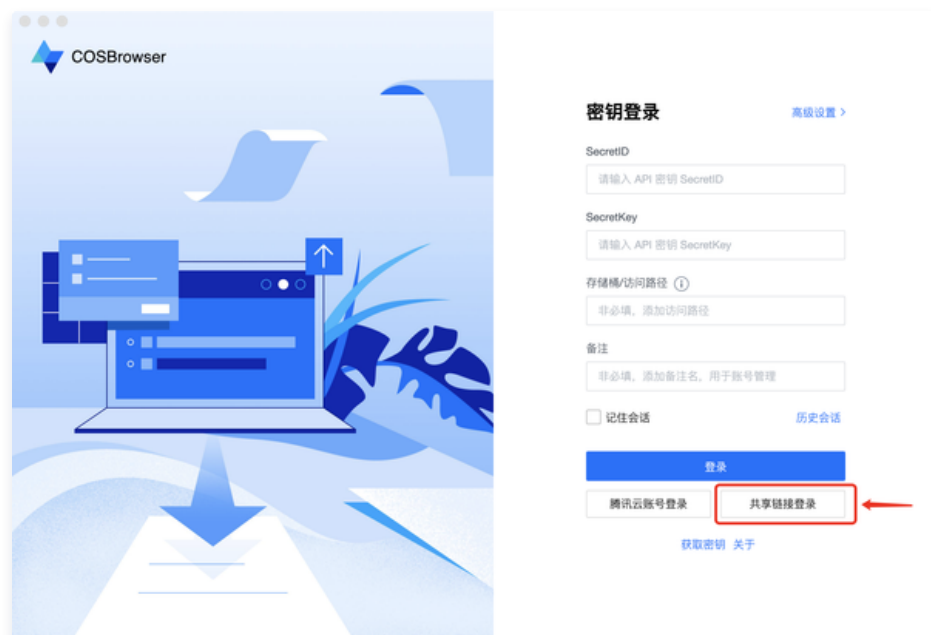


2. 输入提取码，点击**确定**进入分享文件夹。



方式二：在桌面端查看

1. 在登录界面点击**共享链接登录**。



2. 输入已获取的 URL 地址、提取码并登录。

共享链接登录

共享链接

`https://cosbrowser.cloud.tencent.com/share/?id=M3WeYay5O7G214i9dcHcJfq&token=gSXVhkyUmd2aBv4arS6BkIICN%2FNG2YAAAdF0LxnDxPZyxCny5cXExtuPAkSLrO7Of`

提取码

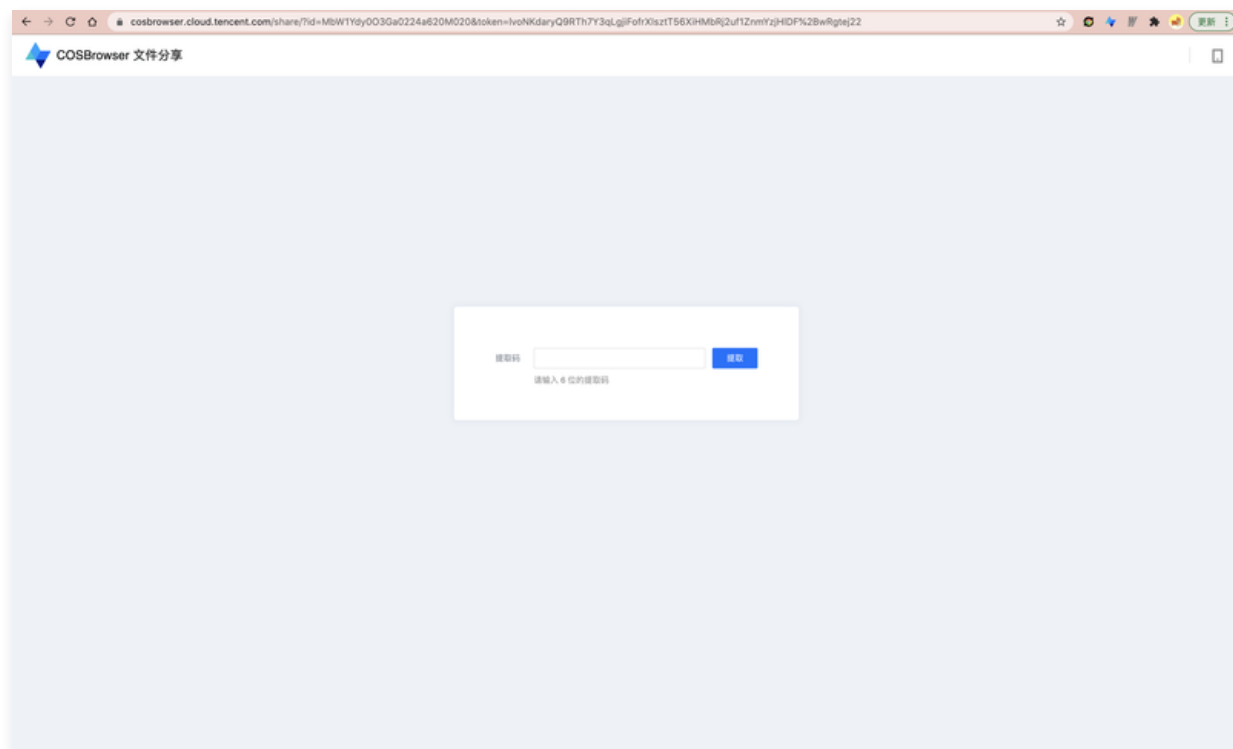
请输入 6 位提取码

登录

关闭

方式三：在浏览器查看

1. 打开浏览器，输入分享的 URL 地址，打开链接。



2. 输入提取码，点击**提取**进入分享文件夹。

分享文件

存在 COS 中的每个文件均可通过特定的链接来进行访问，支持生成指定域名（前提是存储桶设置了其他域名，例如 CDN 加速域名、自定义源站域名等）的文件链接，若文件访问权限为私有读，则可通过请求临时签名的方式生成带有时效的临时访问链接。

⚠ 注意：

- 使用非永久密钥方式登录时，无法配置文件链接的有效时长，默认1小时内有效。
- 若文件为公有读权限，则链接不携带签名，永久有效。若文件为私有读权限，则链接携带临时签名，在1小时内有效。

操作步骤

1. 点击文件，进入文件详情界面。

配置项说明如下：

- 指定域名：可设置链接的域名（可选）。
- 有效时长：设置链接有效时间（可选）。

2. 确认配置无误后，点击生成链接。

3. 发送已生成的文件链接。

文件重命名

❗ 说明：

文件夹无法进行重命名操作。

操作步骤

1. 单击文件右侧的...
2. 在弹出的文件操作列表，点击**重命名**。
3. 在弹出框中，输出新的文件名，点击**确定**即可。

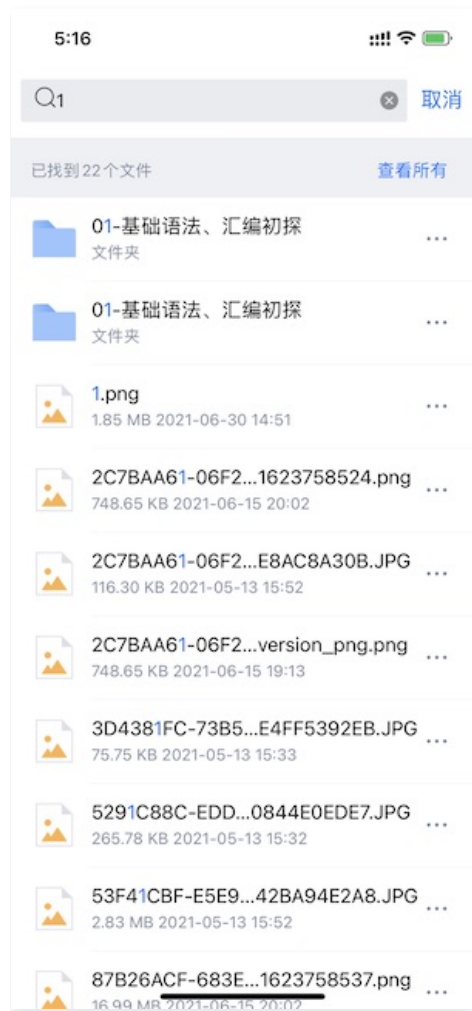
如勾选了**覆盖同名文件**，则会覆盖原来的文件。不勾选会生成一份新的文件，也可以在**个人 > 设置 > 默认上传选项 > 重命名同名文件**中全局配置。

文件搜索

移动端提供了模糊搜索、按照类型搜索的功能，您可以使用搜索前缀关键词，搜索当前文件夹以及子文件夹下指定前缀的文件，也可以先指定要搜索的文件类型，在该类型下搜索。

关键字搜索

输入搜索的关键字，会筛选出当前文件夹以及所有子文件夹下包含该关键字的文件。



类型搜索

COSBrowser 支持按照类型进行搜索，目前支持的类型包括：视频、文件夹、音频、文档、图片、其他。

操作步骤

以搜索文件夹为例：

1. 点击搜索框后，点击文件夹。



2. 该目录下所有的该类型的文件（在该例子中是文件夹）都会被罗列出来。



排序或筛选对象

COSBrowser 支持对存储桶内的文件进行排序、筛选操作。

说明：

当前支持对文件列表按文件名、大小、修改时间进行排序，按存储类型进行筛选。

操作步骤

1. 进入文件列表页，点击搜索栏右边的筛选和排序按钮。
2. 在弹出的操作栏中，即可对存储桶中的文件进行排序或者筛选操作。



文件权限管理

COSBrowser 提供基于文件维度的访问权限设置，且该权限优先级高于存储桶的访问权限。

⚠ 注意：

- 对象的访问权限只在用户通过默认域名访问时有效。通过 CDN 加速域名和自定义域名访问时，以存储桶访问权限为准。
- 访问策略规则存在数量限制，详情请参见 [规格与限制](#)。

修改公共权限

1. 点击想要操作的文件，进入文件详情页面。
2. 点击上边的权限信息，进入权限列表页面。
3. 点击公共权限，即可修改文件的访问权限。

设置用户权限

1. 点击想要操作的文件，进入文件详情页面。
2. 点击上边的权限信息，进入权限列表页面。

3. 点击公共权限，即可修改文件的用户权限。

您可以通过点击**添加用户**新增一条用户权限，新增用户权限后，可以通过左滑，选择编辑或者删除一条用户权限。

资源包管理

最近更新时间：2024-08-30 14:20:11

简介

COSBrowser 移动端提供了查看资源包列表、购买、续费、升级等操作。

说明：

- 关于资源包的类型介绍、购买须知、有效期以及生效范围，请参见 [资源包介绍](#)。
- 通过移动端购买、续费、升级资源包需要跳转到云助手小程序进行操作，在进行相关操作之前请确保 COSBrowser 移动端和腾讯云助手小程序登录的是同一个腾讯云账号。

查看资源包列表

前往 **个人 > 资源包**，跳转至微信小程序，即可查看当前账号下已有的资源包。

资源包购买

操作步骤如下：

- 前往 **个人 > 资源包**，跳转到小程序的资源包购买页面。
- 按照业务需求选择合适的资源包，点击**立即购买**，并按照提示完成支付即可。

资源包续费

若您此前已购买的资源包有效期将要结束，您可按需对已购买的资源包进行手动续期或自动续期，以便将资源包的有效期进行延长。

操作步骤如下：

- 前往 **个人 > 资源包**，跳转到小程序的资源包购买页面...
- 在弹出的操作列表中，点击续费，跳转到腾讯云助手小程序的资源包续费页面。
- 选择要续费的时长，点击**立即购买**，并按照提示完成支付即可。

说明：

如没有续费操作，说明该资源包不支持续费，需要重新购买。

资源包升级

移动端提供资源包的升级功能，若您此前购买的资源包规格不再满足实际业务需求，可点击**升级**，将资源包的规格进行扩容。

操作步骤如下：

- 前往 **个人 > 资源包**，跳转到小程序的资源包购买页面...
- 在弹出的操作列表中，点击升级，跳转到腾讯云助手小程序的资源包升级页面。
- 选择升级后的规格，点击**立即购买**，并按照提示完成支付即可。

说明：

如没有升级操作，说明该资源包不支持升级，需要重新购买。

COS 终端网络诊断工具使用指南

最近更新时间：2025-02-26 16:24:02

工具简介

COS 终端网络诊断工具是集成在 [COSBrowser](#) 中，对终端到 COS 的网络链路进行排障的工具，它可以通过网络探测和传输测速进行真实的网络链路检测，并给出检测的结论和日志。本文将指导您如何使用这款工具进行网络诊断。

使用场景

- 当您的 APP 在移动端使用 COS 服务时遇到错误或者上传、下载慢的问题时，可以用该工具进行网络诊断。
- 移动端的网络情况往往比较复杂，例如手机本身的网络配置以及信号强度、DNS 劫持、运营商链路等等，这些都可能是使用 COS 服务遇到问题的原因，可以用该工具进行问题的确认和原因分析。

下载和安装

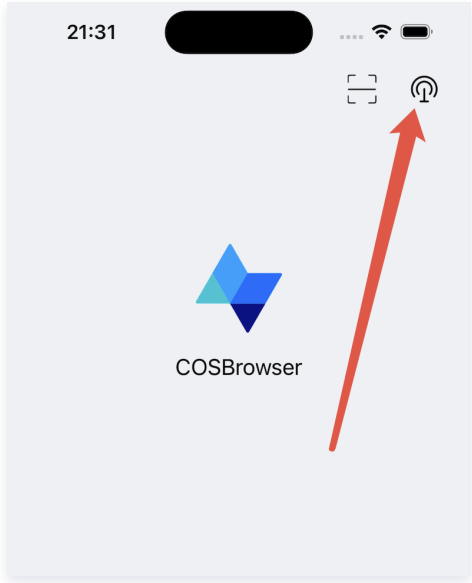
支持平台	系统要求	下载地址
Android	Android 5.0及以上	Android
iOS 移动端	iOS 12及以上	iOS

❗ 说明：

- COSBrowser 移动端目前已覆盖绝大多数应用平台，例如应用宝、App Store 等，可通过单击上述下载地址或自行前往应用平台下载。
- iOS 2.8.7及以上开始支持网络功能，安卓 2.8.6及以上开始支持网络功能。
- 下面以 iOS 移动端为例来介绍工具的具体使用流程。

诊断入口

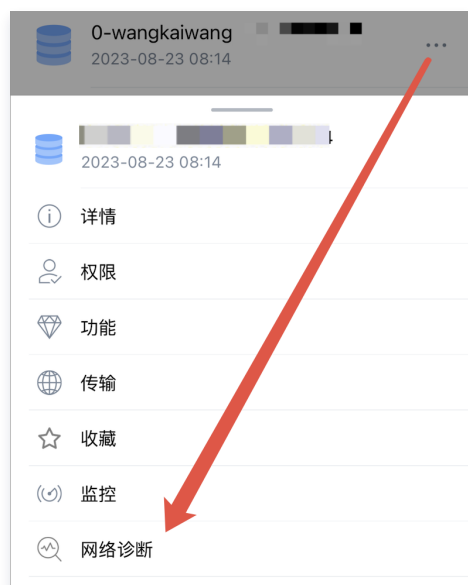
- 登录页面。



- 个人页面。



- 桶列表菜单选项。



网络探测

网络探测可以帮助您检查本地网络开关和信号强度，检测本地代理和 VPN，以及进行网络探测（包括 DNS、Ping、Traceroute、Https），并给出探测结论以及一些解决方法。

功能

本地网络检测

将检测当前设备的网络是否打开以及信号强度、是否配置了代理或使用 VPN，并给出结论和建议。



网络探测

提供了多种网络探测方法，包括 DNS、Ping、Traceroute 和 Https。

- **DNS：**将显示 DNS 解析的结果，包括域名 IP 和 A 记录。



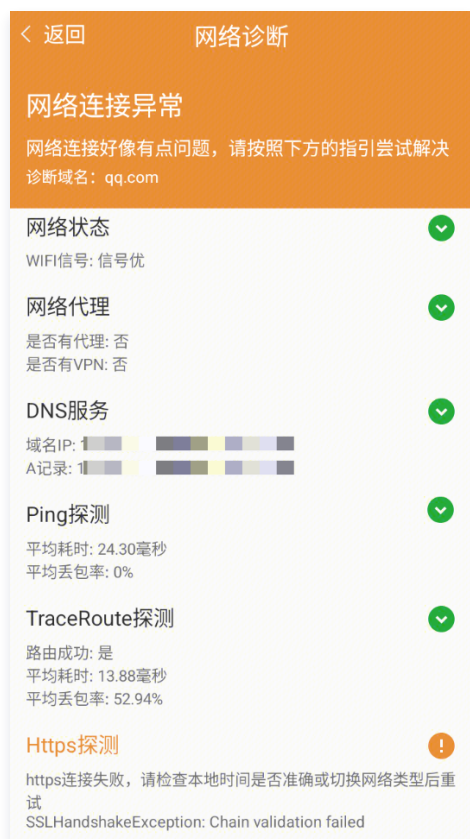
- **Ping：**将显示 ping 的结果，包括平均耗时和丢包率。



- **Traceroute：**将显示路由跟踪的结果，包括最终是否路由成功、平均耗时和平均丢包率。



- **Https：**将尝试建立 https 连接，并显示结果，包括总耗时。



诊断结果和诊断日志

将给出诊断结果，包括可能的问题和建议的解决方案。您也可以查看详细的诊断日志，以获取更多信息或者将日志发送给相关的技术人员进行分析。

注意：

网络诊断工具只能提供网络问题的诊断，不能自动修复问题。如果您在使用工具时遇到任何问题，或者需要进一步的帮助，建议您联系网络服务提供商或专业的网络技术人员。

[< 返回](#)网络诊断

网络连接正常

若仍然无法连接到网络，请下载日志并发送给我们

诊断域名: qq.com

系统版本: 14 (34)

网络状态

WIFI信号: 信号优

网络代理

是否有代理: 否

是否有VPN: 否

DNS服务

域名IP:

A记录:

Ping探测

平均耗时: 32.08毫秒

平均丢包率: 0%

TraceRoute探测

路由成功: 是

平均耗时: 20.53毫秒

平均丢包率: 0%

Https探测

总耗时: 530毫秒

重新诊断

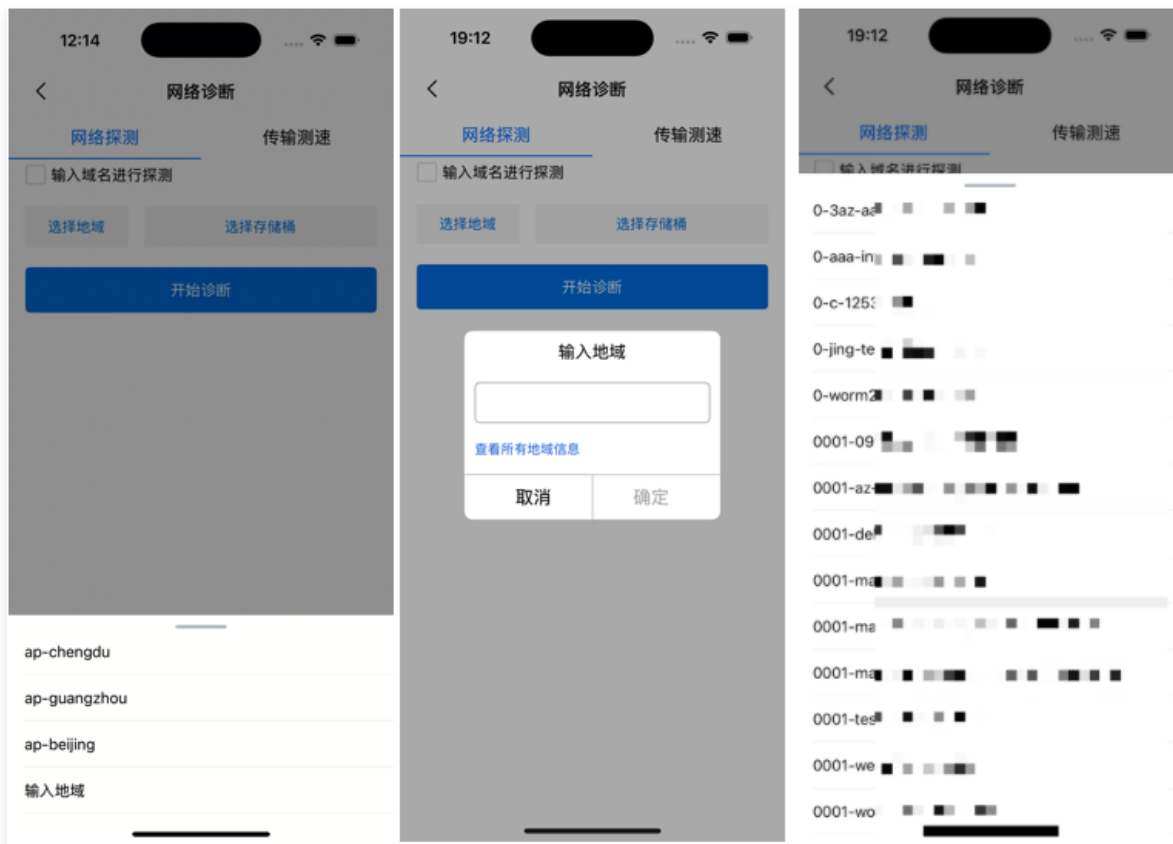
下载日志

使用步骤

1. 进入网络诊断页面后，选择网络探测页签，进行探测。

- 测试账号内存储桶，此种方式要求登录后使用。

选择要测试的地域，若测试的地域不在列表中，则可选择输入地域，但是要确保该地域有已经创建的存储桶。然后选择需要诊断的存储桶，单击开始诊断。

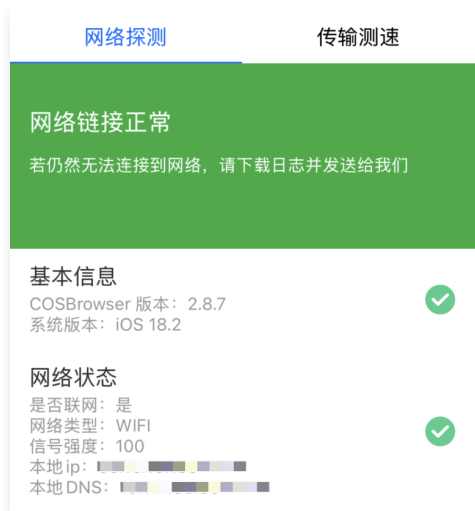


- 输入域名进行探测，此种方式不要求登录。

进入网络诊断工具后，勾选**输入域名进行探测**，在输入框内直接输入需要探测的域名，单击**开始诊断**即可。也可以在右侧选择 COS 域名模板，然后填写对应的<桶名>和<地域>。



- 从桶列表菜单进入，会直接进入探测页面进行默认桶域名的探测。



2. 根据探测结果调整网络配置或将结果及日志发给相关的技术人员进行排查。

传输测速

传输测速可以对多种类型的 COS 域名进行真实的上传、下载操作，检测上传、下载过程中可能遇到的问题，并记录上传、下载的速度。

功能

测试传输速度

针对偏远地区弱网问题，可以快速检测源站域名、全球加速域名、自定义加速域名的传输速度，并根据结果选择最优方案解决。

排查现网问题，获取本地日志

针对现网客户问题，进行点对点排查。生成一个测速链接，让用户尝试复现，并将相关日志抛出，帮助定位问题，缩小排查范围。

使用步骤

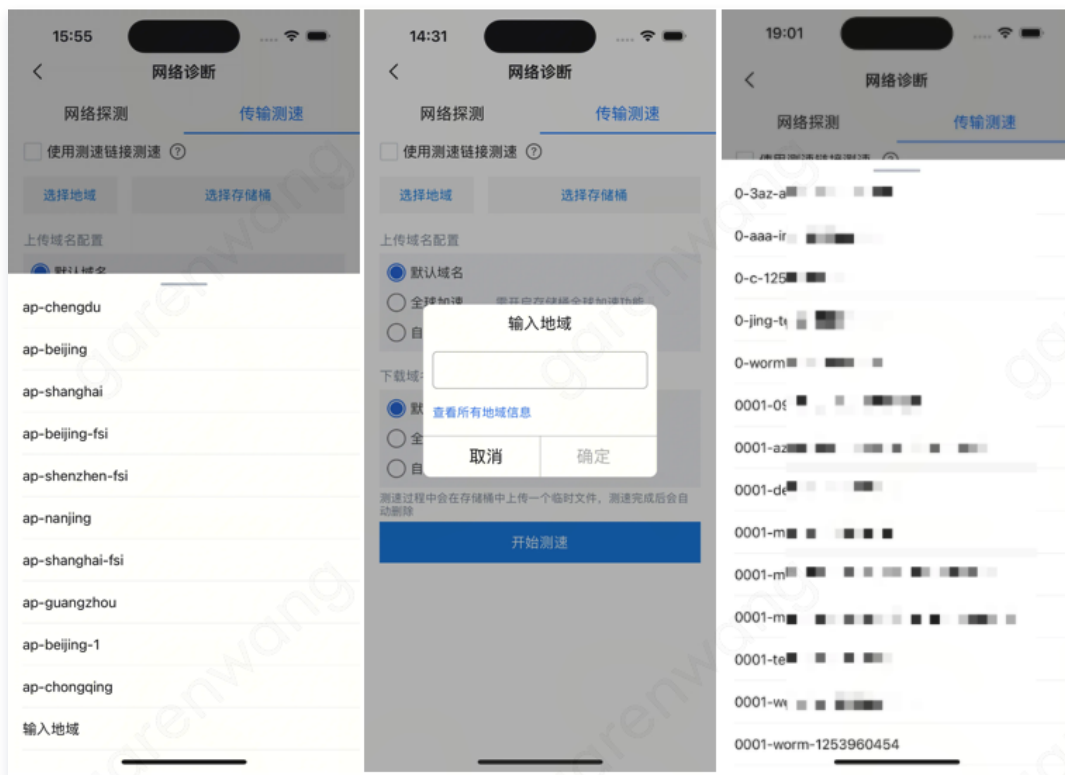
该工具提供以下两种使用方式，下面详细介绍具体使用步骤。

方式一：账号内存储桶测速

1. 安装成功后登录 APP，单击底部 Tab 栏个人，选择网络诊断 > 传输测速。



2. 选择要测试的地域，若测试的地域不在列表中，则可选择**输入地域**，但是要确保该地域有已经创建的存储桶，输入地域后，单击右侧**选择存储桶**按钮，选择要测试的存储桶。



3. 选择存储桶后，开始配置传输所用的域名。默认为源站域名，可以选择**全球加速域名**（选择的存储桶需 [开启全球加速功能](#)）、**自定义域名**（确保域名有效可用，支持手动输入）等。

❗ 说明：

- 选择**全球加速域名**时，选择的存储桶需开全球加速功能。

- 选择自定义域名时，需确认域名有效可用。自定义域名支持手动输入。

4. 开始测速，结果如下图所示。



说明：

测试过程中会给存储桶内上传一个临时文件，并在测试完成后自动删除。

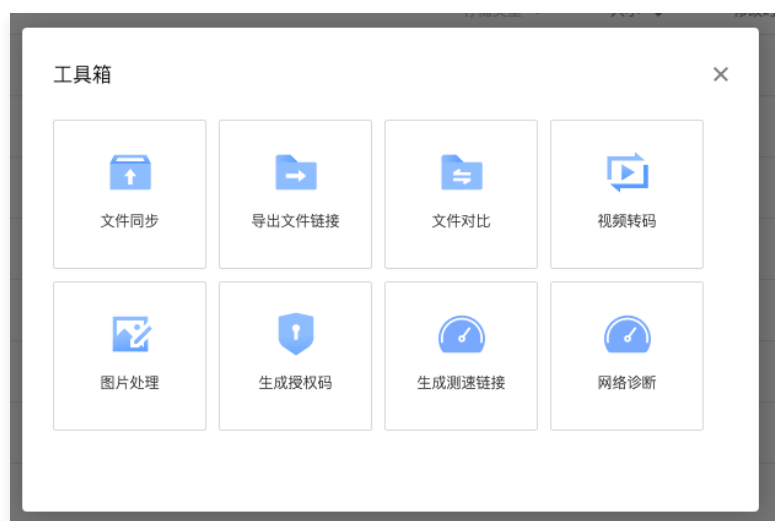
方式二：测速链接测速

方式一只能测试自己的存储桶，但当需要其他人来帮我们测试他所在的网络环境到我们存储桶的传输速度时又不方便让对方直接登录我们的账号。基于这种场景，我们提供了使用测速链接来测速，操作步骤如下：

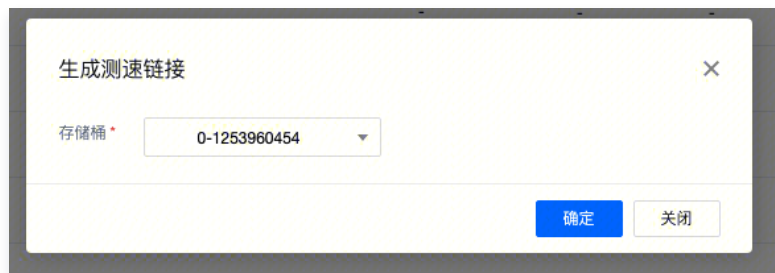
注意：

COSBrowser PC 版本需使用2.11.25及以上。

1. 下载安装 COSBrowser PC 端并登录，单击右上角**工具箱**，选择**生成测速链接**。

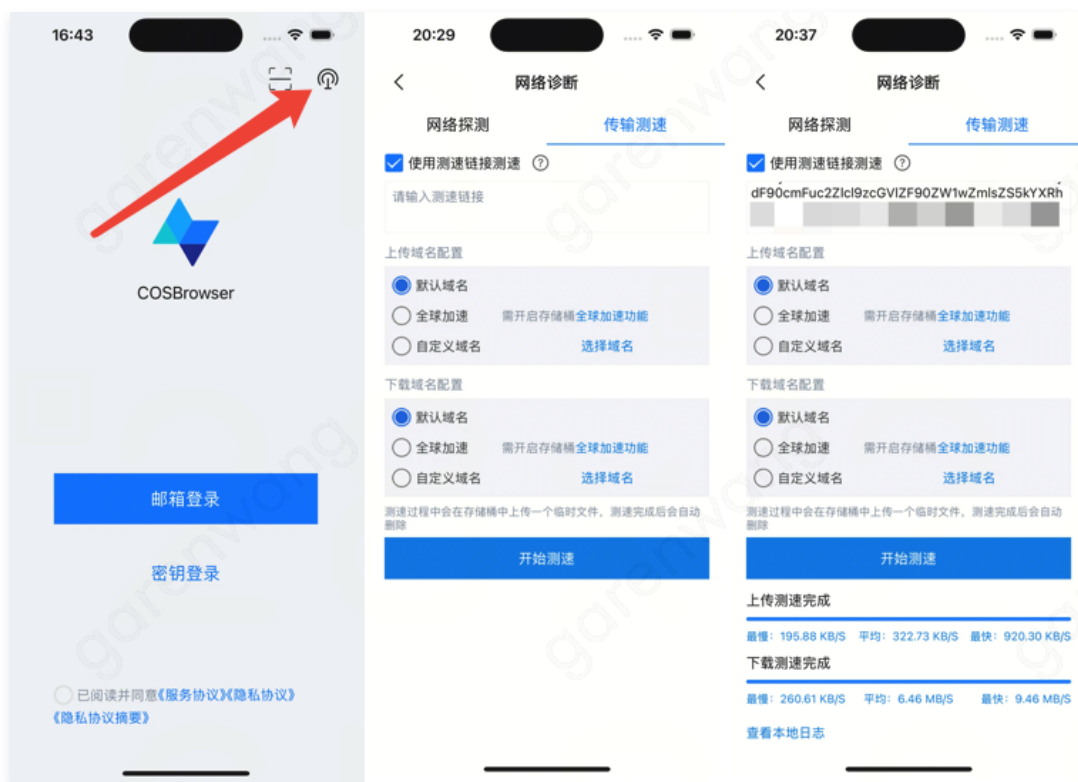


2. 选择存储桶并单击确定，生成测速链接。



3. 将生成的测速链接复制发给对方来帮我们测速。当对方拿到测速链接时，可以通过以下方式进行测速。

- 下载安装 COSBrowser APP 测速，无需登录，安装打开即可。在登录页面右上角单击诊断按钮，进入传输测速页面，粘贴测速链接并单击开始测速。



- 对方在浏览器中（手机、PC 端均可）打开生成的测速链接，单击开始测速。

 COSBrowser 传输测速

上传域名测配置

☒ 默认域名☐ 全球加速域名 (需开启存储桶全球加速功能)☐ 自定义域名:

下载域名测配置

☒ 默认域名☐ 全球加速域名 (需开启存储桶全球加速功能)☐ 自定义域名:

提示: 测速过程中会在存储桶中上传一个临时文件, 测速完成后会自动删除

[开始测速](#)

上传测速完成

最慢: 976.56KB/S 平均: 3.54MB/S 最快: 5.91MB/S

下载测速完成

最慢: 149.60KB/S 平均: 1.75MB/S 最快: 3.11MB/S

结语

希望网络诊断工具以及测速工具能为 COS 的用户排查定位问题带来帮助。如果有更好的想法或者需要工具提供更多的功能, 请 [联系我们](#)。

COSCMD 工具

最近更新时间：2025-06-04 09:55:52

功能说明

使用 COSCMD 工具，用户可通过简单的命令行指令实现对对象（Object）的批量上传、下载、删除等操作。

使用环境

系统环境

支持 Windows、Linux 和 macOS 系统。

❗ 说明：

- 请保证本地字符格式为 UTF-8，否则操作中文版文件会出现异常。
- 请确保本机时间已经与国际标准时间校准，如误差过大，将导致无法正常使用。

软件依赖

- Python 2.7和 Python 3。
- 最新版本的 pip。

❗ 说明：

- 建议用户直接安装集成 pip 的 Python 版本。
- Python 版本为3.12时，pip 安装依赖库时容易失败，建议使用 [源码安装](#)。

安装及配置

- 环境安装与配置详细操作请参见 [Python 安装与配置](#)。
- pip 环境安装与配置详细操作请参见 [官网 pip 安装说明](#)。

下载与安装

下面为用户提供以下三种方式安装 COSCMD。

方式一：通过 pip 安装（推荐）

1. 执行以下 `pip` 命令进行安装。

```
pip install coscmd
```

安装成功之后，用户可以通过 `-v` 或者 `--version` 命令查看当前的版本信息。

⚠ 注意：

使用 Windows 安装后，需要将 `C:\python_install_dir` 和 `C:\python_install_dir\Scripts` 两个路径加入到环境变量中。

2. 安装完成后，执行以下命令进行更新。

```
pip install coscmd -U
```

方式二：源码安装

⚠ 注意:

Python 版本为3.12时, pip 安装依赖库时容易失败, 建议使用该方法安装。

源码下载地址: [单击此处](#)。

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

方式三: 离线安装**⚠ 注意:**

- 如果您的目标机器无法联网安装, 可以先找一台可以联网的机器下载离线安装包, 再将离线安装包拷贝到无法联网的目标机器进行安装。
- 请确保两台机器的 Python 版本保持一致, 否则会出现安装失败的情况。

```
# 在有外网的机器下运行如下命令
mkdir coscmd-packages
pip download coscmd -d coscmd-packages
tar -czvf coscmd-packages.tar.gz coscmd-packages

# 将安装包拷贝到没有外网的机器后运行如下命令
tar -xzvf coscmd-packages.tar.gz
pip install coscmd --no-index -f coscmd-packages
```

配置参数**查看 help 选项**

用户可通过 `-h` 或 `--help` 命令来查看工具的 help 信息和用法。

```
coscmd -h
```

help 信息如下所示:

```
usage: coscmd [-h] [-d] [-s] [-b BUCKET] [-r REGION] [-c CONFIG_PATH]
              [-l LOG_PATH] [--log_size LOG_SIZE]
              [--log_backup_count LOG_BACKUP_COUNT] [-v]

{config,upload,download,delete,abort,copy,move,list,listparts,info,restore,signurl,createbucket,de
letebucket,putobjectacl,getobjectacl,putbucketacl,getbucketacl,putbucketversioning,getbucketversio
ning,probe}
...

an easy-to-use but powerful command-line tool. try 'coscmd -h' to get more
informations. try 'coscmd sub-command -h' to learn all command usage, likes
'coscmd upload -h'

positional arguments:

{config,upload,download,delete,abort,copy,move,list,listparts,info,restore,signurl,createbucket,de
letebucket,putobjectacl,getobjectacl,putbucketacl,getbucketacl,putbucketversioning,getbucketversio
ning,probe}
```

```

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

optional arguments:
-h, --help          show this help message and exit
-d, --debug         Debug mode
-s, --silence       Silence mode
-b BUCKET, --bucket BUCKET
                    Specify bucket
-r REGION, --region REGION
                    Specify region
-c CONFIG_PATH, --config_path CONFIG_PATH
                    Specify config_path
-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

```

除此之外，用户还可以在每个命令后（不加参数）输入 `-h` 查看该命令的具体用法，如下所示。

```

# 查看 upload 命令使用方法
coscmd upload -h

```

生成配置文件

COSCMD 工具在运行时会从配置文件读取所需的必要信息，COSCMD 默认读取的配置文件的路径为 `~/.cos.conf`。如需更改配置文件的路径，请参见 [指定配置文件和日志文件路径的命令](#)。

❗ 说明：

配置前，您需要先在 COS 控制台创建一个用于配置参数的存储桶（例如 `configure-bucket-1250000000`），请参见 [创建存储桶](#)。然后创建密钥信息，请参见 [主账号访问密钥管理](#)。

一个配置文件的示例如下所示。

```
[common]
secret_id = *****
secret_key = *****
bucket = configure-bucket-1250000000
region = ap-chengdu
max_thread = 5
part_size = 1
retry = 5
timeout = 60
schema = https
verify = md5
anonymous = False
```

说明：

- 配置文件中 `timeout` 项，单位为秒，指读写数据的超时时间。
- 配置文件中 `schema` 项，可选值为 `http`、`https`，默认为 `https`。
- 配置文件中 `anonymous` 项，可选值为 `True`、`False`，表示是否使用匿名模式，即签名保持为空。
- 更多配置参数说明，请使用命令 `coscmd config -h` 查看。

使用 config 命令生成配置文件

注意：

- 建议用户通过 [临时密钥](#) 使用工具，通过临时授权的方式进一步提高工具使用的安全性。申请临时密钥时，请遵循 [最小权限指引原则](#)，防止泄露目标存储桶或对象之外的资源。
- 如果您一定要使用永久密钥，建议遵循 [最小权限指引原则](#) 对永久密钥的权限范围进行限制。

`config` 命令可以在 `~/cos.conf` 自动生成配置文件，命令具体格式如下。

```
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]
```

说明：

其中 "[]" 中的字段为选项， "<>" 的字段为需要填写的参数。

参数配置说明如下。

选项	参数说明	有效值	是否必选
-a	密钥 ID 请前往 API 密钥控制台 获取。	字符串	是

-s	密钥 Key 请前往 API 密钥控制台 获取。	字符串	是
-t	临时密钥 token，当使用临时密钥时需要配置，设置 x-cos-security-token 头部。	字符串	否
-b	指定的存储桶名称，存储桶的命名格式为 BucketName-APPID，请参见 命名规范 。初次配置使用时，您需要在 COS 控制台创建一个存储桶，用于配置工具。	字符串	是
-r	存储桶所在地域，请参见 地域和访问域名 。	字符串	是
-e	设置请求的 ENDPOINT，设置 ENDPOINT 参数后，REGION 参数会失效。当您使用的是默认域名，则此处配置格式为 cos.<region>.myqcloud.com；当您使用全球加速域名，则配置为 cos.accelerate.myqcloud.com。	字符串	否
-m	指定多线程操作的最大线程数（默认为5）。如果上传文件缓慢，可适当调高该值。线程数取决于机器性能，假设机器支持最大线程数为30，则将并发线程调整为30，可充分利用机器性能，执行命令为 coscmd config -m 30。	数字	否
-p	分块操作的单块大小（单位MB，默认为1MB，范围为1 - 1000）。如果上传文件缓慢，可适当调高该值。	数字	否
--do-not-use-ssl	使用 HTTP 协议，而不使用 HTTPS。	字符串	否
--anonymous	匿名操作（不携带签名）。	字符串	否

config 命令的使用示例如下：

```
coscmd config -a AChT4ThiXAbpBDEFghT4ThiXAbp**** -s WE54wreefvds3462refgwewe**** -b configure-bucket-1250000000 -r ap-chengdu
```

通用命令

指定 Bucket 和 Region 的命令

用户若不指定存储桶名称和所属地域执行命令，则默认对配置参数时所填的存储桶有效。当需要对不同存储桶执行操作时，需要指定存储桶名称和所属地域。

❗ 说明：

- 通过 -b <BucketName-APPID> 参数指定存储桶名称，存储桶的命名格式为 BucketName-APPID，此处填写的存储桶名称必须为此格式。
- 通过 -r <region> 指定 Region，可以指定存储桶的所属地域。

● 命令格式。

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

● 操作示例：创建一个名称为 examplebucket-1250000000、所属地域为北京的存储桶。

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

● 操作示例：将 D 盘下的文件 picture.jpg 上传到名称为 examplebucket-1250000000的存储桶。

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


指定配置文件和日志文件路径的命令

用户若不指定配置文件的路径，则会使用默认的配置文件路径 `~/cos.conf`。若不指定日志文件的路径，则会使用默认的日志文件路径 `~/cos.log`。

❗ 说明：

- 通过 `-c <conf_path>` 参数指定配置文件路径，COSCND 在运行时会从此路径读取配置信息。
- 通过 `-l <log_conf>` 参数指定日志路径，COSCND 会将运行过程中产生的日志输出到此路径下的日志文件中。

- 命令格式。

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

- 操作示例：指定配置文件路径为 `/data/home/cos_conf`，日志输出路径为 `/data/home/cos_log`，创建一个名称为 `examplebucket-1250000000`、所属地域为北京的存储桶。

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

Debug 模式执行命令

在各命令前加上 `-d` 或者 `--debug`，在命令执行的过程中，会显示详细的操作信息，示例如下。

- 命令格式。

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

- 操作示例：上传时输出详细信息。

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

Silence 模式执行命令

在各命令前加上 `-s` 或者 `--silence`，在命令执行的过程中，将不会再输出任何信息。

❗ 说明：

该命令需满足最低版本为1.8.6.24。

- 命令格式。

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

- 操作示例。

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

常用存储桶命令

创建存储桶

❗ 说明：

执行创建存储桶命令时，请携带参数 `-b <BucketName-APPID>` 指定存储桶名称和 `-r <Region>` 指定所属地域。若直接执行 `coscmd createbucket`，则会报错，原因是不指定存储桶名称和所属地域，则相当于对已存在的存储桶（即配置参数时所填的存储桶）进行创建。

- 命令格式。

```
coscmd -b <BucketName-APPID> createbucket
```

- 操作示例：创建一个名称为 `examplebucket-1250000000`，所属地域为北京的存储桶。

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

删除存储桶

❗ 说明：

`coscmd deletebucket` 的用法仅对配置参数时的存储桶有效。建议配合 `-b <BucketName-APPID>` 指定 Bucket 和 `-r <region>` 指定 Region 使用。

- 命令格式。

```
coscmd -b <BucketName-APPID> deletebucket
```

- 操作示例：删除空的存储桶。

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

- 操作示例：强制删除非空的存储桶。

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

⚠ 注意：

使用 `-f` 参数则会强制删除该存储桶，包括所有文件、开启版本控制之后历史文件夹、上传产生的碎片，请谨慎操作。

常用对象命令

上传文件

- 命令格式。

```
coscmd upload <localpath> <cospath>
```

⚠ 注意：

请将 "<>" 中的参数替换为您需要上传的本地文件路径（`localpath`），以及 COS 上存储的路径（`cospath`）。

- 操作示例：将 D 盘的 `picture.jpg` 文件上传到 COS 的 `doc` 目录下。

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

- 操作示例：将 D 盘的 `doc` 文件夹下的 `picture.jpg` 文件上传到 COS 的 `doc` 目录下。

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

- 操作示例：指定对象类型，上传一个归档类型的文件到 COS 的 doc 目录下。

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

⚠ 注意：

使用 -H 参数设置 HTTP header 时，请务必保证格式为 JSON，示例：

```
coscmd upload -H '{"x-cos-storage-class':'Archive','Content-Language':'zh-CN'}' <localpath>  
<cospath>
```

。更多头部请参见 [PUT Object](#) 文档。

- 操作示例：设置 meta 元属性，上传一个文件到 COS 的 doc 目录下。

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

- 操作示例：指定上传限速为 800Kb/s，将 D 盘的 doc 文件夹下的 file.zip 文件上传到 COS 的 doc 目录下。

```
coscmd upload D:/doc/file.zip doc/ -H '{"x-cos-traffic-limit':'819200'}"
```

❗ 说明：

通过 -H 参数指定请求头部 x-cos-traffic-limit 进行限速，限速值设置范围为 819200 – 838860800，单位默认为 bit/s，即 800Kb/s – 800Mb/s，如果超出该范围将返回 400 错误。

上传文件夹

- 命令格式。

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

⚠ 注意：

Windows 用户推荐在系统自带的 cmd 工具或 PowerShell 上使用 COSCMD 的 upload 命令，其他工具（如 git bash）对命令路径的解析策略与 PowerShell 不同，可能会导致用户的文件被上传到错误的路径上去。

- 操作示例：将 D 盘的 doc 文件夹及其文件上传到 COS 根路径。

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

- 操作示例：将 D 盘的 doc 文件夹中的文件、子文件夹上传到 COS 的 test 路径下，不包括 doc 文件夹本身。

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

- 操作示例：同步上传，跳过 md5、文件大小相同的同名文件。

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

⚠ 注意：

使用 `-s` 参数可以使用同步上传，跳过上传 md5 一致的文件（COS 上的原文件必须是由 1.8.3.2 版本之后的 COSCMD 上传的，默认带有 `x-cos-meta-md5` 的 header）。

- 操作示例：同步上传，跳过文件大小相同的同名文件。

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

⚠ 注意：

使用 `-s` 参数可以使用同步上传，且带上 `--skipmd5` 参数后，将只对比同名文件的大小，如果大小相同则跳上传。

- 操作示例：同步上传，并删除 “D 盘 doc 文件夹中已经删除的文件”。

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

💡 说明：

例如当天 D 盘 doc 文件夹中的文件与 COS 中 doc 路径下的文件已同步上传，源端与目标端一致。第二天用户把 D 盘 doc 文件夹中的 A 文件删除了，此时执行以上命令，会删除 COS 中 doc 路径下的 A 文件，最终实现 D 盘 doc 文件夹中的文件与 COS 中 doc 路径下的文件保持一致。

- 操作示例：D 盘 doc 文件夹中 .txt 和 .doc 后缀的文件选择忽略上传。

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

- 操作示例：D 盘 doc 文件夹中 .txt 后缀的文件选择忽略上传。

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

⚠ 注意：

- 在上传文件夹时，使用 `--ignore` 参数可以忽略某一类文件，使用 `--include` 参数可以过滤某一类文件，支持 shell 通配规则，支持多条规则，用逗号 `,` 分隔。当忽略一类后缀时，必须使用 `"` 包裹。
- 如果您希望使用 `--ignore` 过滤特定文件夹内的所有文件，需要使用绝对路径，并在路径前后加入 `"`。例如

```
coscmd upload -rs D:/doc / --ignore "D:/doc/ignore_folder/*"
```

- 操作示例：D 盘 doc 文件夹中 .txt 和 .doc 的后缀文件上传。

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

- 操作示例：D 盘 doc 文件夹中 .txt 后缀的文件上传。

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

⚠ 注意：

- 当上传大于一定阈值（100MB）的文件，COSCMD 即采用分块上传方式，命令用法和简单上传一致，即

```
coscmd upload <localpath> <cospath>
```

。如果您使用的是 1.8.6.31 以下版本（不包括 1.8.6.31 版本），则该阈值为 10MB。
- COSCMD 支持大文件断点上传功能；当分块上传大文件失败时，重新上传该文件只会上传失败的分块，而不会从头开始（请保证重新上传的文件的目录以及内容和上传的目录保持一致）。
- COSCMD 分块上传时会对每一块进行 MD5 校验。

- COSCMD 上传默认会携带 `x-cos-meta-md5` 的头部，值为该文件的 md5 值，如果带上 `--skipmd5` 参数则不携带该头部。

查询文件列表

查询命令如下：

- 命令格式。

```
coscmd list <cospath>
```

- 操作示例：递归查询该存储桶下 doc/ 前缀下所有的文件列表。

```
coscmd list doc/
```

- 操作示例：递归查询该存储桶下所有的文件列表、文件数量和文件大小。

```
coscmd list -ar
```

- 操作示例：递归查询 examplefolder 前缀的所有文件列表。

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

- 操作示例：查询该存储桶下所有文件的历史版本。

```
coscmd list -v
```

❗ 说明：

- 请将 "<>" 中的参数替换为您需要查询文件列表的 COS 上文件的路径（`cospath`）。`<cospath>` 为空默认查询当前存储桶根目录。
- 使用 `-a` 查询全部文件。
- 使用 `-r` 递归查询，并且会在末尾返回列出文件的数量和大小之和。
- 使用 `-n num` 设置查询数量的最大值。

查看文件信息

命令如下：

- 命令格式。

```
coscmd info <cospath>
```

- 操作示例：查看 doc/picture.jpg 的元信息。

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

❗ 说明：

请将 "<>" 中的参数替换为您需要显示的 COS 上文件的路径（`cospath`）。

下载文件或文件夹

下载文件

- 命令格式。

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

⚠ 注意：

请将 "<>" 中的参数替换为您需要下载 COS 上文件的路径（cospath），以及本地存储路径（localpath）。

- 操作示例：下载 COS 上的 doc/picture.jpg 到 D:/picture.jpg。

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

- 操作示例：下载 COS 上的 doc/picture.jpg 到 D 盘。

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

- 操作示例：下载一个带有版本 ID 的 picture.jpg 文件到 D 盘。

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

- 操作示例：指定下载单链接限速为 800Kb/s，下载 COS 上的 doc/picture.jpg 到 D:/picture.jpg。

```
coscmd download doc/picture.jpg D:/picture.jpg -H '{"x-cos-traffic-limit':'819200'}"
```

ⓘ 说明：

通过 -H 参数指定请求头部 x-cos-traffic-limit 进行限速，限速值设置范围为 819200 – 838860800，单位默认为 bit/s，即 800Kb/s – 800Mb/s，如果超出该范围将返回 400 错误。

下载文件夹

- 命令格式。

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

- 操作示例：下载 doc 目录为 D:/folder/doc。

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

- 操作示例：下载根目录文件，但跳过根目录下的 doc 目录。

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

- 操作示例：覆盖下载当前存储桶根目录下所有的文件。

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

⚠ 注意：

若本地存在同名文件，则会下载失败，需要使用 `-f` 参数覆盖本地文件。

- 操作示例：同步下载当前 bucket 根目录下所有的文件，跳过 md5 校验相同的同名文件。

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

⚠ 注意：

使用 `-s` 或者 `--sync` 参数，可以在下载文件夹时跳过本地已存在的相同文件（前提是下载的文件是通过 COSCMD 的 upload 接口上传的，文件携带有 `x-cos-meta-md5` 头部）。

- 操作示例：同步下载当前 bucket 根目录下所有的文件，跳过文件大小相同的同名文件。

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

- 操作示例：同步下载，并删除 “COS 存储桶中 doc 路径下已删除的文件”。

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

💡 说明：

例如，当天 D 盘 doc 文件夹中的文件与 COS 中 doc 路径下的文件已同步下载，源端与目标端一致。第二天用户把 COS 中 doc 路径下的 A 文件删除了，此时执行以上命令，并按照窗口提示输入 `y`（确定）后，会删除 D 盘 doc 文件夹下的 A 文件，最终实现 D 盘 doc 文件夹中的文件与 COS 中 doc 路径下的文件保持一致。

- 操作示例：忽略 .txt 和 .doc 的后缀文件。

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

- 操作示例：忽略 .txt 后缀的文件。

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

⚠ 注意：

- 在上传文件夹时，使用 `--ignore` 参数可以忽略某一类文件，使用 `--include` 参数可以过滤某一类文件，支持 shell 通配规则，支持多条规则，用逗号 `,` 分隔。当忽略一类后缀时，必须使用 `"` 包裹。
- 如果您希望使用 `--ignore` 过滤特定目录内的所有文件，需要使用绝对路径，并在路径前后加入 `"`。例如

```
coscmd upload -rs D:/doc / --ignore "D:/doc/ignore_folder/*"
```

- 操作示例：过滤 .txt 和 .doc 的后缀文件。

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

- 操作示例：过滤 .txt 后缀的文件。

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

⚠ 注意：

老版本的 `mgget` 接口已经废除，`download` 接口使用分块下载，请使用 `download` 接口。

获取带签名的下载 URL

- 命令格式。

```
coscmd signurl <cospath>
```

- 操作示例：生成 doc/picture.jpg 路径带签名的 URL。

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

- 操作示例：生成 doc/picture.jpg 路径带100s 签名的 URL。

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

说明：

- 请将 "<>" 中的参数替换为您需要获取下载 URL 的 COS 上文件的路径（cospath）。
- 使用 `-t time` 设置该 URL 中签名的有效时间（单位为秒），默认为10000s。

删除文件或文件夹

删除文件

- 命令格式。

```
coscmd delete <cospath>
```

注意：

请将 "<>" 中的参数替换为您需要删除的 COS 上文件的路径（cospath），工具会提示用户是否确认进行删除操作。

- 操作示例：删除 doc/exampleobject.txt。

```
coscmd delete doc/exampleobject.txt
```

- 操作示例：删除带有版本 ID 的文件。

```
coscmd delete doc/exampleobject.txt --versionId MTg0NDUxMzc4ODA3NTgyMTErEWN
```

删除文件夹

- 命令格式。

```
coscmd delete -r <cospath>
```

- 操作示例：删除 doc 目录。

```
coscmd delete -r doc
```

- 操作示例：删除 folder/doc 目录。

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


- 操作示例：删除 doc 文件夹下的所有版本控制文件。

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

说明：

- 批量删除需要输入 `y` 确定，使用 `-f` 参数则可以跳过确认直接删除。
- 需注意，执行删除文件夹命令，将会删除当前文件夹及其文件。但删除版本控制的文件，需要指定版本 ID 进行删除。

查询分块上传文件碎片

- 命令格式。

```
coscmd listparts <cospath>
```

- 操作示例：查询 doc/ 前缀的文件碎片。

```
coscmd listparts doc/
```

清除分块上传文件碎片

- 命令格式。

```
coscmd abort
```

- 操作示例：消除所有上传文件碎片。

```
coscmd abort
```

复制文件或文件夹

复制文件

- 命令格式。

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

- 操作示例：在同个存储桶内进行复制，将 examplebucket-1250000000 存储桶下的 picture.jpg 文件复制到 doc 文件夹下。

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

- 操作示例：在不同存储桶内进行复制，复制 examplebucket2-1250000000 存储桶下的 doc/picture.jpg 对象到 examplebucket1-1250000000 存储桶的 doc/examplefolder/。

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

- 修改存储类型，将复制后的对象所对应的存储类型改为低频存储。

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

- 修改存储类型，将复制后的对象所对应的存储类型改为归档存储，并重命名为 photo.jpg。

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

复制文件夹

- 命令格式。

```
coscmd copy -r <sourcepath> <cospath>
```

- 操作示例：复制 examplebucket2-1250000000 存储桶下的 examplefolder 目录到 examplebucket1-1250000000 存储桶的 doc 目录。

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

❗ 说明：

- 请将 "<>" 中的参数替换为您需要复制的 COS 上文件的路径（sourcepath），和您需要复制到 COS 上文件的路径（cospath）。
- sourcepath 的格式为：<BucketName-APPID>.cos.<region>.myqcloud.com/<cospath>。
- 使用 -d 参数可以设置 x-cos-metadata-directive 参数，可选值为 Copy 和 Replaced，默认为 Copy。
- 使用 -H 参数设置 HTTP header 时，请务必保证格式为 JSON，示例：

```
coscmd copy -H -d Replaced '{"x-cos-storage-class':'Archive','Content-Language':'zh-CN'}"  
<localpath> <cospath>
```

- 更多头部请参见 [PUT Object - Copy](#) 文档。

移动文件或文件夹

⚠ 注意：

移动命令的 <sourcepath> 和 <cospath> 不能相同，否则会导致文件被删除。原因在于 move 命令会先复制，再删除，<sourcepath> 路径下的文件最终会被删除。

移动文件

- 命令格式。

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

- 操作示例：在同个存储桶内进行移动，将 examplebucket-1250000000 存储桶下的 picture.jpg 移动到 doc 文件夹下。

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

- 操作示例：在不同存储桶内进行移动，移动 examplebucket2-1250000000 存储桶下的 picture.jpg 对象到 examplebucket1-1250000000 存储桶的 doc/folder/。

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

- 操作示例：将移动到目标存储桶中的对象所对应的存储类型修改为低频存储。

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

- 操作示例：将移动到目标存储桶中的对象所对应的存储类型修改为归档存储。

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

移动文件夹

- 命令格式。

```
coscmd move -r <sourcepath> <cospath>
```

- 操作示例：移动 examplebucket2-1250000000 存储桶下的 examplefolder 目录到 examplebucket1-1250000000 存储桶的 doc 目录。

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

❗ 说明：

- 请将 "<>" 中的参数替换为您需要移动的 COS 上文件的路径（sourcepath），和您需要移动到 COS 上文件的路径（cospath）。
- sourcepath 的格式为：<BucketName-APPID>.cos.<region>.myqcloud.com/<cospath>。
- 使用 -d 参数可以设置 x-cos-metadata-directive 参数，可选值为 Copy 和 Replaced，默认为 Copy。
- 使用 -H 参数设置 HTTP header 时，请务必保证格式为 JSON，示例：

```
coscmd move -H -d Replaced '{"x-cos-storage-class':'Archive','Content-Language':'zh-CN'}"  
<sourcepath> <cospath>
```

- 更多头部请参见 [PUT Object - copy](#) 文档。

设置对象访问权限

- 命令格式。

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

❗ 说明：

对象访问权限支持设置为 --grant-read（即授予用户读权限）、--grant-full-control（即授予用户完全控制的权限）。对于文件夹则还可以设置为 --grant-write（即授予用户写权限）。

- 操作示例：授予账号 1000000000001 拥有 picture.jpg 的读取权限。

```
coscmd putobjectacl --grant-read 100000000001 picture.jpg
```

- 操作示例：查询文件的访问权限。

```
coscmd getobjectacl picture.jpg
```

开启/暂停版本控制

- 命令格式。

```
coscmd putbucketversioning <status>
```

- 操作示例：开启版本控制。

```
coscmd putbucketversioning Enabled
```

- 操作示例：暂停版本控制。

```
coscmd putbucketversioning Suspended
```

- 操作示例：查询版本控制。

```
coscmd getbucketversioning
```

⚠ 注意：

- 请将 "<>" 中的参数替换为您需要版本控制状态（status）。
- 一旦您对存储桶启用了版本控制，它将无法返回到未启用版本控制状态（初始状态）。但是，您可以对该存储桶暂停版本控制，这样后续上传的对象将不会产生多个版本。

恢复归档文件

- 恢复归档文件命令格式。

```
coscmd restore <cospath>
```

- 操作示例：快速取回模式回热 picture.jpg，有效期3天。

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

- 批量恢复归档文件命令格式。

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

- 操作示例：快速取回模式回热 examplefolder/ 目录，有效期3天。

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

❗ 说明：

- 请将 "<>" 中的参数替换为您需要查询文件列表的 COS 上文件的路径（cospath）。
- 使用 `-d <day>` 设置临时副本的过期时间，默认值：7。
- 使用 `-t <tier>` 指定恢复模式，枚举值：Expedited（快速取回模式），Standard（标准取回模式），Bulk（批量取回模式），默认值：Standard。

常见问题

如您在使用 COSCMD 工具过程中，有相关的疑问，请参见 [COSCMD 工具类常见问题](#)。

结语

当然，COS 不仅提供以上应用和服务，还提供多款热门开源应用，并集成腾讯云 COS 插件，欢迎点击 [此处](#) 一键启动，立即使用！

COSCLI 工具

COSCLI 简介

最近更新时间：2025-03-24 14:31:42

COSCLI 是腾讯云对象存储（Cloud Object Storage，COS）提供的客户端命令行工具。通过 COSCLI 工具，您可以通过简单的命令行指令对您 COS 中的对象（Object）实现批量上传、下载、删除等操作。

COSCLI 使用 Go 编写，基于 Cobra 框架，支持配置多个存储桶和跨桶操作。您可以通过 `./coscli [command] --help` 来查看 COSCLI 的使用方法。

功能列表

- [生成与修改配置文件 - config](#)
- [创建存储桶 - mb](#)
- [删除存储桶 - rb](#)
- [存储桶标签 - bucket-tagging](#)
- [查询存储桶或文件列表 - ls](#)
- [获取不同类型文件的统计信息 - du](#)
- [上传下载或拷贝文件 - cp](#)
- [同步上传下载或拷贝文件 - sync](#)
- [删除文件 - rm](#)
- [获取文件哈希值 - hash](#)
- [列出分块上传中产生的碎片 - lsparts](#)
- [清理碎片 - abort](#)
- [取回归档文件 - restore](#)
- [获取预签名 URL - signurl](#)
- [创建/获取软链接 - symlink](#)
- [查看对象内容 - cat](#)
- [列出目录下内容并统计 - lsdu](#)
- [版本控制 - bucket-versioning](#)

下载地址

COSCLI 工具支持 Windows、MacOS、Linux 操作系统，详情请参见 [下载与安装配置](#)。

下载与安装配置

最近更新时间：2025-04-24 15:48:22

COSCLI 工具提供 Windows、MacOS、Linux 操作系统的二进制包，通过简单的安装和配置后即可使用。

步骤一：下载 COSCLI 工具

您可以根据业务场景选择下载 COSCLI 工具的地址。如果您的服务器在国内，推荐使用国内站点下载地址（此处链接的工具版本均为最新版本，如需使用旧版本，可访问 [release](#) 获取历史版本）。

国内站点下载地址	Github 下载地址（国外站点使用）	SHA256校验
Windows-386	Windows-386	8d415550d4625e0df0b39164326cedd5512d3271c11bebea0dcf9cfc01f6fe46
Windows-amd64	Windows-amd64	a81488cecae7767904e2a111cd8f6d2def0f95f8aba04bb7bcef47b1327bc210
MacOS-amd64	MacOS-amd64	c9c4bb2a021423e3a3ee014c99c8287d484d9a55451f7f75c87d3aaa9417e80f
MacOS-arm64	MacOS-arm64	9253e043985b51219112400a37b39e618ab5c358cf5777193829aa7b9d9ff5f6
Linux-386	Linux-386	e25a856141140c0da262509fa146a424d19f23830cbd4d92e231f7e38d07d8e7
Linux-amd64	Linux-amd64	8a137821c9869d9c07be84e8c2ed69177d9029c1c504e04662e048631112553c
Linux-arm	Linux-arm	07c03d5fc78ef5c714e0e154b8a79fc5850382527e03d7e60b46b26cf9478d59
Linux-arm64	Linux-arm64	34b4d1a82dc5aa23d80c808289d97a03bf101b31871c0a53100de2af2fd25b0e

您也可以通过命令行，获取国内站点 MacOS 和 Linux 环境下的 COSCLI 工具文件：

- **MacOS-amd64:** `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-darwin-amd64`
- **MacOS-arm64:** `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-darwin-arm64`
- **Linux-386:** `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-linux-386`
- **Linux-amd64:** `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-linux-amd64`
- **Linux-arm:** `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-linux-arm`
- **Linux-arm64:** `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-linux-arm64`

 **说明：**
Github 上当前版本号为 v1.0.6，如需获取工具的最新版本、历史版本和更新日志，请前往 [release](#) 进行查看。

步骤二：安装 COSCLI 工具

Windows

1. 以 windows-amd64 版本为例，将已下载的 Windows 版本 COSCLI 工具，移动至 C:\Users\<用户名> 目录下。
2. 将 coscli-windows-amd64.exe 重命名为 coscli.exe。
3. 按 win+r 键打开 运行 程序。

- 在对话框中，输入 `cmd`，按 `Enter` 打开命令行窗口。
- 在命令行窗口中，输入以下命令。

```
coscli --version
```

若输出 `coscli version v1.0.6`，则安装成功。

❗ 说明：

在 Windows 系统下，在不同的命令行客户端中使用 COSCLI 的方式可能略有不同，若输入 `coscli [command]` 后 COSCLI 无法正常工作，请尝试 `./coscli [command]` 的格式。

MacOS

- 以 MacOS-amd64 版本为例，运行以下命令，重命名 MacOS 版本 COSCLI 文件。

```
mv coscli-darwin-amd64 coscli
```

- 运行以下命令修改文件执行权限。

```
chmod 755 coscli
```

- 在命令行中输入以下命令。

```
./coscli --version
```

若输出 `coscli version v1.0.6`，则安装成功。

❗ 说明：

在 MacOS 系统下使用 COSCLI 时，若弹出无法打开“coscli”，因为无法验证开发者的提示，可以前往设置 > 安全性与隐私 > 通用中选择仍要打开 coscli，之后即可正常使用 COSCLI。

Linux

- 以 Linux-amd64 版本为例，运行以下命令，重命名 Linux 版本 COSCLI 文件。

```
mv coscli-linux-amd64 coscli
```

- 运行以下命令修改文件执行权限。

```
chmod 755 coscli
```

- 在命令行窗口中输入以下命令。

```
./coscli --version
```

若输出 `coscli version v1.0.6`，则安装成功。

步骤三：配置 COSCLI 工具

⚠ 注意：

- 建议用户通过 [临时密钥](#) 使用工具，通过临时授权的方式进一步提高工具使用的安全性。申请临时密钥时，请遵循 [最小权限指引原则](#)，防止泄露目标存储桶或对象之外的资源。
- 如果您一定要使用永久密钥，建议遵循 [最小权限指引原则](#) 对永久密钥的权限范围进行限制。

首次使用 COSCLI 时，用户需要初始化配置文件。配置文件包含以下两部分内容：

- 授权 COSCLI 访问您的腾讯云账号，需要您配置密钥 ID、密钥 Key、临时密钥 Token。
- 为常用桶添加别名，需要您配置常用桶名称、桶所在地域信息、桶别名。配置常用桶信息后，用户可以使用别名进行桶操作，无需再填写桶名称和地域信息。添加多个常用桶配置后，还可以更便捷的进行跨桶或跨域的操作。如果无需配置常用桶信息，可以按 `Enter` 跳过。

初次使用时，COSCLI 会自动调用 `./coscli config init` 在 `~/.cos.yaml` 位置生成配置文件，您可以在命令行交互式的完成配置。后续您也可以使用 `./coscli config init` 命令在其他位置为 COSCLI 交互式地生成配置文件。您可以使用 `./coscli config show` 来查看配置文件的所在位置和配置参数信息。

配置文件中各配置项的说明如下：

配置项	说明
Secret ID	密钥 ID，建议使用子账号密钥，授权遵循最小权限指引，降低使用风险。子账号密钥获取请参见 子账号访问密钥管理 。
Secret Key	密钥 Key，建议使用子账号密钥，授权遵循最小权限指引，降低使用风险。子账号密钥获取请参见 子账号访问密钥管理 。
Session Token	临时密钥 token，当使用临时密钥时需要配置，若不使用可以直接按 <code>Enter</code> 跳过。关于临时密钥的更多信息，请参见 使用临时密钥访问 COS 。
Mode	设置身份模式，支持枚举值 <code>SecretKey</code> 和 <code>CvmRole</code> 。可以为空，空值默认为 <code>SecretKey</code> ，表示使用密钥请求 COS。当 Mode 为 <code>CvmRole</code> 时，表示是用 管理实例角色 请求 COS。
Cvm Role Name	设置 CVM 角色实例名称，详情请参见 管理实例角色 。
protocol	网络传输协议，默认为 https。如需更改为 http，可直接进入配置文件进行修改即可。
APPID	APPID 是您在成功申请腾讯云账户后所得到的账号，由系统自动分配，可从 账号信息 中获取。一个存储桶的全称由 <code>Bucket Name</code> 和 <code>APPID</code> 这两个元素组成，格式为 <code><BucketName-APPID></code> ，详情请参见 存储桶命名规范 。
Bucket Name	存储桶名称，和 APPID 一起构成存储桶全称，格式为 <code><BucketName-APPID></code> ，详情请参见 存储桶命名规范 。
Bucket Endpoint	存储桶所在地域的域名，默认域名格式为 <code>cos.<region>.myqcloud.com</code> ，其中 <code><region></code> 代表存储桶地域，例如 <code>ap-guangzhou</code> 、 <code>ap-beijing</code> 等，COS 支持的地域列表请参见 地域与访问域名 。 如果存储桶开启了全球加速，可配置全球加速域名。例如，全球加速域名配置为 <code>cos.accelerate.myqcloud.com</code> ；内网全球加速域名配置为 <code>cos-internal.accelerate.tencentcos.cn</code> 。
Bucket Alias	存储桶别名，配置后可以在使用 <code>BucketAlias</code> 代替 <code>BucketName-APPID</code> ，减少所需输入的命令长度，如果不配置此项， <code>BucketAlias</code> 的值是 <code>BucketName-APPID</code> 的值。
OFS Bucket	元数据加速桶标记，用于标识该桶是否开启 元数据加速 功能。
Auto Switch Host	设置是否关闭自动切换备份域名。可选值 <code>true</code> <code>false</code> ，可以为空。 - 若不设置或值为 <code>false</code>，则会进行备份域名切换。 - 若设置为 <code>true</code>，则不进行备份域名切换。
Disable Encryption	设置是否关闭秘钥加密。可选值 <code>true</code> <code>false</code> ，可以为空。 - 若不设置或值为 <code>false</code>，则会加密配置文件中的秘钥相关信息。 - 若设置为 <code>true</code>，则不加密配置文件中的秘钥相关信息。

在初次配置时，COSCLI 只会要求您配置一个存储桶的信息，如果您希望配置多个存储桶，您后续可以使用 `./coscli config add` 命令添加存储桶配置。如您需要修改配置文件，或获取更多有关配置文件的操作，请参见 [config 命令](#) 或使用 `./coscli config --help` 命令来快速查看和配置文件相关的指令。

在正式开始使用指令之前，您可以使用 `./coscli --help` 命令来快速查看 COSCLI 的使用方法。

其他配置方法

除了使用 `./coscli config init` 交互式地生成配置文件外，您也可以直接手动编写 COSCLI 的配置文件。COSCLI 的配置文件格式是 `yaml` 格式，配置文件示例如下：

```
cos:
  base:
    secretid: XXXXXXXXXXXXXXXX
    secretkey: XXXXXXXXXXXXXXXX
    sessiontoken: ""
    protocol: https
  buckets:
  - 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：

COSCLI 默认从 `~/cos.yaml` 处读取配置项，若用户想要使用自定义的配置文件，请在命令后使用 `-c`（`--config-path`）选项。配置文件存储的 `secretid/secretkey/sessiontoken` 均为加密后的字符串。

通用选项

最近更新时间：2025-06-20 14:07:42

您可以通过 `./coscli --help` 或 `./coscli -h` 命令查看 COSCLI 支持的通用选项。

选项说明

以下为 COSCLI 的通用选项，这些选项可以在工具的所有命令中使用：

⚠ 注意：

- 建议用户通过 [临时密钥](#) 使用工具，通过临时授权的方式进一步提高工具使用的安全性。申请临时密钥时，请遵循 [最小权限指引原则](#)，防止泄露目标存储桶或对象之外的资源。
- 如果您一定要使用永久密钥，建议遵循 [最小权限指引原则](#) 对永久密钥的权限范围进行限制。

选项	说明
-h, --help	输出帮助信息，用户可通过 -h 或 --help 命令来查看工具的 help 信息和用法。用户还可以在每个命令后（不加参数）输入 -h 查看该命令的具体用法，例如查看创建存储桶命令的具体用法，则输入 <code>./coscli mb -h</code>
-c, --config-path	配置文件路径，COSCLI 的默认配置文件路径为 <code>~/.cos.yaml</code> ，同时支持用户自定义配置文件，在命令后使用 <code>-c</code> 可指定配置文件
-e, --endpoint	除了在配置文件中提前配置存储桶的地域外，COSCLI 也支持在命令中通过 <code>-e</code> 指定存储桶的 endpoint，endpoint 形如 <code>cos.<region>.myqcloud.com</code> ，其中 <code><region></code> 代表存储桶地域，如 <code>ap-guangzhou</code> 、 <code>ap-beijing</code> 等，COS 支持的地域列表可参见 地域与访问域名
-i, --secret-id	指定访问 COS 使用的密钥中的 SecretId
-k, --secret-key	指定访问 COS 使用的密钥中的 SecretKey
--token	用户通过临时密钥访问 COS
-v, --version	显示 COSCLI 的版本
-p, --protocol	网络传输协议，默认为 https
--init-skip	默认为 false。设置为 true（--init-skip=true）则跳过 config init 交互式操作，直接使用参数中的 SecretId、SecretKey 及 endpoint 请求 api，使用此参数则 -i、-k、-e 参数必传
--log-path	自定义 coscli.log 文件位置，默认在 COSCLI 的同级目录。可指定文件夹或具体文件（文件需以 .log 结尾）如： <code>/data/</code> 或 <code>/data/coscli.log</code>

操作示例

示例一：切换存储桶并上传对象

当用户需要通过 COSCLI 切换至另一个地域的存储桶时，可以通过 -e 选项指定该存储桶所属的 endpoint。
例如，将本地文件 test.txt 上传至成都地域的存储桶 examplebucket-1250000000，成都读取的 endpoint 为 `cos.ap-chengdu.myqcloud.com`，命令如下：

```
./coscli cp test.txt cos://examplebucket-1250000000/test.txt -e cos.ap-chengdu.myqcloud.com
```

示例二：切换用户账号查看文件列表

当用户需要使用另一个账号的身份时，也可以通过 -i 和 -k 选项分别指定用户密钥的 SecretId 和 SecretKey。

例如，使用另一个账号的身份列出成都地域下的存储桶 examplebucket-1250000000 的文件列表，命令如下：

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

常用命令

生成与修改配置文件 – config

最近更新时间：2025-04-11 10:06:52

命令格式

以下 config 命令用于生成与修改配置文件：

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

❗ 说明

- 正确填写各配置项后，您可以使用 `./coscli config show` 来查看配置信息。
- 关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。
- 生成的配置文件，默认网络传输协议为 https。如需更改为 http，可直接进入配置文件进行修改即可。

config 命令包含以下子命令：

command 名称	command 用途
add	添加一个新的存储桶配置。
delete	删除一个已经存在的存储桶配置。
init	交互式地生成配置文件。
set	修改配置文件 base 组中的一个或多个配置项。base 组包含 <code>secretid</code> 、 <code>secretkey</code> 、 <code>sessiontoken</code> 、 <code>mode</code> 、 <code>cvmrolename</code> 信息。
show	打印指定配置文件中的信息。

config 及其子命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法。
-c	--config-path	指定要使用的配置文件路径。

config add 子命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法。
-a	--alias	存储桶别名。
-b	--bucket	存储桶名称。
-r	--region	存储桶地域。
-o	--ofs	元数据加速桶标记。详情请参见 元数据加速概览 。

⚠ 注意

如果您需要指定存储桶的 endpoint，可以使用通用 flag `-e` 或 `--endpoint`，详见 [通用选项介绍](#)。

config delete 子命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法。
-a	--alias	存储桶别名。

config set 子命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法。
无	--secret_id	设置 secret ID。可从 访问管理控制台 中创建并获取。
无	--secret_key	设置 secret key。可从 访问管理控制台 中创建并获取。
-t	--session_token	设置临时密钥 token。关于临时密钥的更多信息，请参见 使用临时密钥访问 COS 。
无	--mode	设置身份模式，支持枚举值 <code>SecretKey</code> 和 <code>CvmRole</code> 。可以为空，空值默认为 <code>SecretKey</code> ，表示使用密钥请求 COS。当 mode 为 <code>CvmRole</code> 时，表示是用 管理实例角色 请求 COS。
无	--cvm_role_name	设置 CVM 角色实例名称，详情请参见 管理实例角色 。
无	--close_auto_switch_host	设置是否关闭自动切换备份域名。可选值 true false，可以为空。 - 若不设置或值为 false，则会进行备份域名切换。 - 若设置为 true，则不进行备份域名切换。
无	--disable_encryption	设置是否关闭密钥加密。默认为 false，为 true 时即为不加密配置文件中的密钥相关信息。

操作示例

添加一个新的存储桶配置

```
./coscli config add -b examplebucket3-1250000000 -r ap-chengdu -e cos.ap-chengdu.myqcloud.com -a bucket3
```

删除一个已经存在的存储桶配置

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

修改默认配置文件中的 session-token

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

打印指定配置文件中的信息

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

修改默认配置文件中的 mode 和 cvmrolename

```
./coscli config set --mode CvmRole --cvm_role_name testName
```

创建存储桶 – mb

最近更新时间：2025-02-28 18:13:55

mb 命令用于创建存储桶。

注意：

如需使用此命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:PutBucket`，更多授权请参见 [支持 CAM 的业务接口](#)。

命令格式

```
./coscli mb cos://<BucketName-APPID> -e <endpoint> [flag]
```

注意：

使用 mb 命令创建存储桶，需要携带全局 flag `-e` 或 `--endpoint` 指定存储桶所在地域。

mb 命令包含以下参数：

参数格式	参数用途	示例
cos://<BucketName-APPID>	用于自定义存储桶名称	cos://examplebucket-1250000000

mb 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
-r	--region	存储桶地域
-m	--maz	创建多 az 的存储桶
-o	--ofs	创建 ofs 的存储桶

说明：

- 使用 mb 命令成功创建存储桶后，推荐您在配置文件中新增该桶的信息，方便后续使用桶别名快速操作。具体命令用法，可参见下方操作示例。
- 关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

操作示例

```
// 创建存储桶 bucket3
./coscli mb cos://bucket3-1250000000 -e cos.ap-chengdu.myqcloud.com
```

如果您希望为刚创建的存储桶配置别名，需要使用以下命令更新配置文件：

```
// 更新配置文件
./coscli config add -b bucket3-1250000000 -e cos.ap-chengdu.myqcloud.com -a bucket3
// 更新后可用 cos://bucket3 访问此存储桶
```


删除存储桶 – rb

最近更新时间：2025-02-28 18:13:55

rb 命令用于删除存储桶。

⚠ 注意：

如需使用此命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:DeleteBucket`，更多授权请参见 [支持 CAM 的业务接口](#)。

命令格式

```
./coscli rb cos://<bucket-name> [flag]
```

rb 命令包含以下参数：

参数格式	参数用途	示例
cos://<bucket-name>	指定需要访问的存储桶。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag。	使用桶别名访问：cos://example-alias 使用桶名称访问：cos://examplebucket-1250000000

rb 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
-r	--region	存储桶地域

📌 说明：

- rb 命令仅可删除空存储桶，如果您的桶中还有文件请使用 [删除文件](#) 命令和 [清理碎片](#) 命令分别清空存储桶中的文件和文件碎片后再删除存储桶。
- 使用 rb 命令成功删除存储桶后，推荐您在配置文件中删除该桶的信息。具体命令用法，可参见下方操作示例。
- 关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

操作示例

```
// 删除存储桶 bucket3（需要确认操作）
./coscli rb cos://bucket3-1250000000 -e cos.ap-chengdu.myqcloud.com
```

```
// 更新配置文件，删除 bucket3 的存储桶配置
./coscli config delete -a bucket3
```

存储桶标签 – bucket-tagging

最近更新时间：2025-02-28 18:13:55

bucket-tagging 命令用于创建（修改）、获取、删除存储桶标签。每个存储桶下最多支持50组标签。

⚠ 注意：

- 如需获取存储桶标签，在您进行 [授权策略](#) 时，action 需要设置为 `cos:GetBucketTagging`。
- 如需设置存储桶标签，在您进行 [授权策略](#) 时，action 需要设置为 `cos:PutBucketTagging`。
- 如需删除存储桶标签，在您进行 [授权策略](#) 时，action 需要设置为 `cos:DeleteBucketTagging`。

更多授权请参见 [支持 CAM 的业务接口](#)

命令格式

```
./coscli bucket-tagging --method [method] cos://<bucket-name> [tag_key]#[tag_value]
```

bucket-tagging 命令包含以下参数：

参数格式	参数用途	示例
cos://<bucket-name>	指定需要访问的存储桶。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag	使用桶别名访问：cos://example-alias 使用桶名称访问：cos://examplebucket-1250000000

bucket-tagging 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--method	指定需要进行的操作，包括 put（添加标签）、get（查询标签）、delete（删除标签）

❗ 说明：

关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

添加或修改存储桶标签

存储桶标签用一组键值对（Key-Value）来表示，只有存储桶所有者及拥有 PutBucketTagging 权限的用户才可以添加或修改存储桶标签，否则会返回错误码 403 AccessDenied。

命令格式

```
./coscli bucket-tagging --method put cos://bucketAlias key1#value1 key2#value2
```

其中，`key#value` 表示标签键值对 Key-Value，key 和 value 之间以 # 分割。若存储桶未设置标签，此命令将为存储桶添加指定的标签；若存储桶已设置标签，此命令将覆盖原有的标签。

操作示例

为桶别名为 example-alias 的存储桶配置两组标签，其中一组标签的 key 为1，value 为111，一组标签的 key 为2，value 为222。命令如下：

```
./coscli bucket-tagging --method put cos://exmaple-alias 1#111 2#222
```

查询存储桶标签

命令格式

```
./coscli bucket-tagging --method get cos://bucketAlias
```

操作示例

```
./coscli bucket-tagging --method get cos://exmaple-alias
```

以下输出结果表明桶别名为 example-alias 的存储桶配置了两组标签，其中一组标签的 key 为1，value 为111，一组标签的 key 为2，value 为222。

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

删除存储桶标签

命令格式

```
./coscli bucket-tagging --method delete cos://bucketAlias
```

操作示例

```
./coscli bucket-tagging --method delete cos://exmaple-alias
```

查询存储桶或文件列表 – ls

最近更新时间：2025-02-28 18:13:55

ls 命令用于查询所有存储桶列表、查询存储桶下的文件列表和文件夹下的文件列表。

⚠ 注意：

- 如需列出存储桶内文件，在您进行 [授权策略](#) 时，action 需要设置为 `cos:HeadBucket` ， `cos:GetBucket` 。
- 如需列出历史版本信息（传入 `--all-versions` ），在您进行 [授权策略](#) 时，action 需要设置为 `cos:HeadBucket` ， `cos:GetBucketVersioning` ， `cos:GetBucketObjectVersions` 。
- 如需列出账号下的存储桶，在您进行 [授权策略](#) 时，action 需要设置为 `cos:GetService` 。

更多授权请参见 [支持 CAM 的业务接口](#) 。

命令格式

```
./coscli ls [cos://<bucket-name>[/prefix/]] [flag]
```

ls 命令包含以下参数：

参数格式	参数用途	示例
cos://<bucket-name>	可选参数。指定需要访问的存储桶。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag	- 使用桶别名访问：cos://example-alias - 使用桶名称访问：cos://examplebucket-1250000000
/prefix/	可选参数。指定某一文件夹	/picture/

ls 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件
-r	--recursive	是否递归地遍历文件夹，并列出所有文件
无	--limit	指定列出的最大数量（传0或不传则默认为10000）
无	--all-versions	列举对象的所有版本，仅当 bucket 开启版本控制后可用，列出历史版本参数增加显示字段 <code>VersionId</code> ， <code>IsLatest</code> ， <code>Delete Marker</code>

❗ 说明：

- `--include` 和 `--exclude` 支持标准正则表达式的语法，您可以使用它来过滤出符合特定条件的文件。
- 使用 zsh 时，您可能需要在 pattern 串的两端加上双引号。

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

- 关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

操作示例

列出当前账号下所有存储桶

```
./coscli ls
```

返回的信息包括：桶名称，地域，创建时间，存储桶总数。示例如下：

BUCKET NAME	REGION	CREATE DATE
examplebucket-1250000000	ap-nanjing	2022-01-01T00:00:00Z
TOTAL BUCKETS:		2

列出文件

列出 bucket1存储桶中的所有文件

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

返回的信息包括：对象键（对象在存储桶中的唯一标识），存储类型，最近更新时间，对象大小，对象总数。示例如下：

KEY	TYPE	LAST MODIFIED	ETAG
SIZE	RESTORESTATUS		
test.txt	STANDARD	2024-06-05T15:03:37+08:00	"a3bc6c9058109f8da48d41a5ab9abc7c"
OBJECTS:		1	TOTAL

列出 bucket1存储桶中 picture 文件夹下的所有文件和文件夹

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

普通列出仅返回查询路径所在层级的数据，不对子路径进行展开。示例如下：

KEY	TYPE	LAST MODIFIED	ETAG
SIZE	RESTORESTATUS		
picture/a4431470f55662	STANDARD	2024-06-05T15:03:58+08:00	"ed0430c5f27e76605e0555c260478112"
picture/e98c6cefa4abd6	STANDARD	2024-06-05T15:03:58+08:00	"bd5a4bd7248e7dfdb796383bee60470b"
TOTAL OBJECTS:		3	

```
-----+-----
```

递归列出 bucket1 存储桶中 picture 文件夹下的所有文件

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

如果查询路径所在层级有子路径，递归列出会对所有子路径进行扫描，返回查询路径层级下的所有文件。示例如下：

KEY	TYPE	LAST MODIFIED	ETAG
SIZE	RESTORESTATUS		
-----+-----			
-----+-----			
picture/subfolder	DIR		
picture/subfolder/pic2.png	STANDARD	2024-06-05T15:03:58+08:00	
"bd5a4bd7248e7dfdb796383bee60470b"	53.00 B		
-----+-----			
-----+-----			
TOTAL OBJECTS: 3			
-----+-----			

递归列出 bucket1 存储桶中所有的 .mp4 类型文件

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

递归列出 bucket1 存储桶中的所有不是 .mp4 类型的文件

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

递归列出 bucket1 存储桶中 picture 文件夹下所有以 test 开头并且类型不是 .jpg 的文件

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

列出历史版本

列出 bucket1 存储桶中的所有历史版本

```
./coscli ls cos://bucket1/ -r --all-versions
```

如果查询路径所在层级有子路径，递归列出会对所有子路径进行扫描，返回查询路径层级下的所有文件的所有历史版本。示例如下：

KEY	TYPE	VERSIONID	ISLATEST	DELETE
MARKER	LAST MODIFIED	ETAG	SIZE	
-----+-----				
-----+-----				
cmd/cmd/abort.go	STANDARD	MTg0NDUwMDM1MjIxMDM3MDI1MDM	false	false
2025-02-26T14:33:25+08:00	"c9bfc40db6669e9a7aee03abcd8b66e8"	1.89 KB		

```

cmd/cmd/abort_test.go          | STANDARD | MTg0NDUwMDM1MjIxMDM3MjgxODI | false | false
| 2025-02-26T14:33:25+08:00 | "52166b1c60e4089a4652546c0350d2c7" | 4.89 KB
cmd/cmd/bucket_tagging.go      | STANDARD | MTg0NDUwMDM1MjIxMDM3MjY3MDI | false | false
| 2025-02-26T14:33:25+08:00 | "edad62e08bf65a5bff81304e2b40ac1a" | 3.82 KB
cmd/cmd/bucket_versioning.go   | STANDARD | MTg0NDUwMDM1MjIxMDM2MTkzMzU | false | false
| 2025-02-26T14:33:25+08:00 | "50a6b0e2e218c437ccdabe2c762aaef1" | 1.86 KB
cmd/cmd/buket_tagging_test.go  | STANDARD | MTg0NDUwMDM1MjIxMDM2MTY4MDc | false | false
| 2025-02-26T14:33:25+08:00 | "cd11257b22c9816df105da15f3ceb70f" | 8.23 KB
cmd/cmd/abort.go              |          | MTg0NDUwMDM1MjIwNjcyNTcxMDA | true  | true
| 2025-02-26T14:34:02+08:00 |          |
cmd/cmd/abort_test.go         |          | MTg0NDUwMDM1MjIwNjcxNzE5NDY | true  | true
| 2025-02-26T14:34:02+08:00 |          |
cmd/cmd/bucket_tagging.go     |          | MTg0NDUwMDM1MjIwNjcyNjczNzI | true  | true
| 2025-02-26T14:34:02+08:00 |          |
cmd/cmd/bucket_versioning.go   |          | MTg0NDUwMDM1MjIwNjY3ODc4NzM | true  | true
| 2025-02-26T14:34:02+08:00 |          |
cmd/cmd/buket_tagging_test.go |          | MTg0NDUwMDM1MjIwNjY3ODYyMDI | true  | true
| 2025-02-26T14:34:02+08:00 |          |
-----+-----+-----+-----+-----+-----
--+-----+-----+-----+-----+-----+-----

TOTAL OBJECTS:          | 10
-----+-----

```

获取不同类型文件的统计信息 – du

最近更新时间：2025-02-28 18:13:55

du 命令用于列出某存储桶或某文件夹下，每一种存储类型文件的统计信息。统计信息包括不同存储类型的文件总数、每种类型文件的总大小。

⚠ 注意：

- 如需使用此命令统计所有对象信息，在您进行 [授权策略](#) 时，action 需要设置为 `cos:HeadBucket`，`cos:GetBucket`。
- 如需使用此命令统计所有历史版本信息，在您进行 [授权策略](#) 时，action 需要设置为 `cos:HeadBucket`，`cos:GetBucketVersioning`，`cos:GetBucketObjectVersions`。

更多授权请参见 [支持 CAM 的业务接口](#)。

命令格式

```
./coscli du cos://<bucket-name>[/prefix/] [flag]
```

du 命令包含以下参数：

参数格式	参数用途	示例
cos://<bucket-name>	指定需要访问的存储桶。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> <code>flag</code>	使用桶别名访问：cos://example-alias 使用桶名称访问：cos://examplebucket-1250000000
/prefix/	可选参数，指定某一文件夹	/picture/

du 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件
无	--all-versions	统计对象的所有版本数据，仅当 bucket 开启版本控制后可用，统计信息增加显示 DeleteMarker 数量

❗ 说明

- `--include` 和 `--exclude` 支持标准正则表达式的语法，您可以使用它来过滤出符合特定条件的文件。
- 使用 zsh 时，您可能需要在 pattern 串的两端加上双引号。

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

- 关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

操作示例

列出 bucket1 存储桶内文件的统计信息

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


返回的结果示例如下，输出的信息包括：该桶各存储类型下的对象数量和大小、桶内对象总数、桶内对象总容量。

```
STORAGE CLASS      | OBJECTS COUNT | TOTAL SIZE
-----+-----+-----
STANDARD |          2 |      164 B
STANDARD_IA |          0 |          0 B
INTELLIGENT_TIERING |          0 |          0 B
ARCHIVE |          0 |          0 B
DEEP_ARCHIVE |          0 |          0 B
MAZ_STANDARD |          0 |          0 B
MAZ_STANDARD_IA |          0 |          0 B
MAZ_INTELLIGENT_TIERING |          0 |          0 B
MAZ_ARCHIVE |          0 |          0 B
-----+-----+-----
INFO[2022-12-14 17:35:41] Total Objects Count: 2
INFO[2022-12-14 17:35:41] Total Objects Size:  164 B
```

列出 bucket1 存储桶 picture 文件夹下文件的统计信息

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

列出 bucket1 存储桶 picture 文件夹下所有 .mp4 类型文件的统计信息

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

列出 bucket1 存储桶 picture 文件夹下所有非 .md 类型文件的统计信息

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

列出 bucket1 存储桶内文件及所有历史版本的统计信息

```
./coscli du cos://bucket1 --all-versions
```

返回的结果示例如下，输出的信息包括：该桶各存储类型下的对象数量和大小、桶内对象总数、桶内对象总容量。

```
STORAGE CLASS      | OBJECTS COUNT | TOTAL SIZE
-----+-----+-----
STANDARD |        545 |    1.14 MB
STANDARD_IA |          0 |          0 B
INTELLIGENT_TIERING |          0 |          0 B
ARCHIVE |          0 |          0 B
DEEP_ARCHIVE |         13 |    11.74 KB
MAZ_STANDARD |          0 |          0 B
MAZ_STANDARD_IA |          0 |          0 B
MAZ_INTELLIGENT_TIERING |          0 |          0 B
MAZ_ARCHIVE |          0 |          0 B
-----+-----+-----
INFO[2025-02-25 17:36:36] Total Objects Count: 558
INFO[2025-02-25 17:36:36] Total Objects Size:  1.15 MB
INFO[2025-02-25 17:36:36] Total DeleteMarker Count:  501
```

上传下载或拷贝文件 – cp

最近更新时间：2025-07-03 10:57:02

cp 命令用于上传、下载或拷贝文件。

⚠ 注意:

- 如需使用上传文件命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:HeadBucket`，`cos:GetBucket`，`cos:HeadObject`，`cos:InitiateMultipartUpload`，`cos:UploadPart`，`cos:CompleteMultipartUpload`，`cos:ListMultipartUploads`，`cos:ListParts`。
- 如需使用下载文件命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:HeadBucket`，`cos:GetBucket`，`cos:HeadObject`，`cos:GetObject`。
- 如需使用拷贝文件命令，在您进行 [授权策略](#) 时，目标对象action 需要设置为 `cos:GetBucket`，`cos:HeadObject`，`cos:InitiateMultipartUpload`，`cos:PutObject`，`cos:CompleteMultipartUpload`。源对象action 需要设置为 `cos:HeadBucket`，`cos:GetBucket`，`cos:HeadObject`，`cos:GetObject`。

更多授权请参见 [支持 CAM 的业务接口](#)。

命令格式

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

cp 命令包含以下参数：

参数格式	参数用途	示例
source_path	源文件路径。可以为本地路径或 COS 文件路径。COS 路径支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag	- 本地路径：~/example.txt - 使用桶别名指定 COS 文件路径： cos://bucketalias/example.txt - 使用桶名称指定 COS 文件路径： cos://examplebucket-1250000000/example.txt
destination_path	目的文件路径。可以为本地路径或 COS 文件路径。COS 路径支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag	- 本地路径：~/example.txt - 使用桶别名指定 COS 文件路径： cos://bucketalias/example.txt - 使用桶名称指定 COS 文件路径： cos://examplebucket-1250000000/example.txt

cp 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
无	--include	- 包含特定模式的文件（v1.0.4之前的版本在上传时仅筛选传入路径下的本地文件路径，v1.0.4以及之后版本会筛选传入的全路径） - 例如：需要上传./test 下的所有文件到 COS，./test 下包含 aaa 文件夹，aaa 文件夹下包含 1.txt 文件 1.0.4之前的版本会匹配 aaa/1.txt 1.0.4及之后版本会匹配 ./test/aaa/1.txt
无	--exclude	排除特定模式的文件（v1.0.4之前的版本在上传时仅筛选传入路径下的本地文件路径，v1.0.4以及之后版本会筛选传入的全路径）

		- 例如：需要上传 <code>./test</code> 下的所有文件到 COS，<code>./test</code> 下包含 <code>aaa</code> 文件夹，<code>aaa</code> 文件夹下包含 <code>1.txt</code> 文件 1.0.4 之前的版本会匹配 <code>aaa/1.txt</code> 1.0.4 及之后版本会匹配 <code>./test/aaa/1.txt</code>
<code>-r</code>	<code>--recursive</code>	是否递归遍历文件夹下所有文件
无	<code>--storage-class</code>	指定上传文件的存储类型（默认 STANDARD），更多存储类型，请参见 存储类型概述
无	<code>--part-size</code>	文件的分块大小（默认 32MB，最大支持 5GB）。如果需要根据文件大小自适应分块大小，请设置为 0
无	<code>--thread-num</code>	文件分块并发线程数（默认并发 5）
无	<code>--rate-limiting</code>	单链接速率限制（0.1~100MB/s），单位 MB/s
无	<code>--meta</code>	上传文件的元信息。包括部分 HTTP 标准属性（HTTP Header）以及以 <code>x-cos-meta-</code> 开头的用户自定义元数据（User Meta）。文件元信息格式为 <code>header:value#header:value</code> ，示例为 <pre>Expires:2022-10-12T00:00:00.000Z#Cache-Control:no-cache#Content-Encoding:gzip#x-cos-meta-x:x</pre>
无	<code>--routines</code>	指定文件间并发上传或下载的线程数量，默认为 3
无	<code>--fail-output</code>	此选项决定是否启用上传或下载失败时文件的错误输出（默认为 <code>true</code> ，开启）。如果启用，文件传输失败将被记录在指定目录内（如果没有指定，默认为 <code>./coscli_output</code> ）。如果禁用，只会将错误文件的数量输出到控制台
无	<code>--fail-output-path</code>	此选项用于指定记录上传或下载失败文件的错误输出文件夹。通过提供自定义文件夹路径，您可以控制错误输出文件夹的位置和名称。如果未设置此选项，将使用默认的错误日志文件夹 <code>./coscli_output</code>
无	<code>--retry-num</code>	限频重试次数（默认为 0，不重试）。可选 1-10 次。当多台机器同时在一个 COS 目录上执行下载操作时，可以通过指定此参数进行重试来规避限频错误
无	<code>--err-retry-num</code>	错误重试次数（默认为 0）。指定 1-10 次，或者设置为 0 表示不进行重试
无	<code>--err-retry-interval</code>	重试间隔（仅在指定 <code>--err-retry-num</code> 为 1-10 时可用）。指定 1-10 秒的重试间隔，如果不指定或设置为 0，每次重试间隔将在 1-10 秒内的随机
无	<code>--only-current-dir</code>	是否仅上传当前目录中的文件，忽略子目录及其内容（默认为 <code>false</code> ，不忽略）
无	<code>--disable-all-symlink</code>	是否上传时忽略所有软链接子文件和软链接子目录（默认为 <code>true</code> ，不上传）。目前仅支持在 Linux 和 MacOS 系统上使用
无	<code>--enable-symlink-dir</code>	是否上传软链接的子目录（默认为 <code>false</code> ，不上传）。目前仅支持在 Linux 和 MacOS 系统上使用
无	<code>--disable-crc64</code>	是否关闭 CRC64 数据验证（默认为 <code>false</code> ，启用验证）
无	<code>--disable-checksum</code>	默认为 <code>false</code> ，校验整个文件的 crc64。为 <code>true</code> 时仅校验分片 crc64
无	<code>--move</code>	会在文件成功拷贝到目标路径后，删除源文件（仅 cos 路径间可用）
无	<code>--version-id</code>	下载指定版本的文件，仅支持在已开启版本控制的 Bucket 内使用（仅支持单文件）

❗ 说明：

- `cp` 命令在上传和下载大文件时会自动启用并发上传、下载。

- 当文件大于 `--part-size` 时，COSCLI 会先将文件按 `--part-size` 进行切块，之后用 `--thread-num` 个线程并发地执行上传、下载任务。
- 每个线程都会维护一个链接，对于每个链接，您可以使用 `--rate-limiting` 参数对单链接进行限速，当启用并发上传、下载时，总速率为 `--thread-num * --rate-limiting`。
- 当以分块形式上传/下载文件时，会默认开启断点续传。
- `--include` 和 `--exclude` 支持标准正则表达式的语法，您可以使用它来过滤出符合特定条件的文件。
- 使用 `zsh` 时，您可能需要在 `pattern` 串的两端加上双引号。

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

- 在 Windows `cmd` 中使用命令时，需注意“——”字符（中文横杠）粘贴进 `cmd` 会自动变为“--”，需手动输入下。
- 关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

操作示例

上传操作

上传单文件

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

将本地 `test` 文件夹下的所有文件及所有文件夹上传至 `bucket1`桶中的 `example` 文件夹下

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

将本地 `test` 文件夹及其子文件夹下的所有 `.mp4` 类型文件上传至 `bucket1`桶中的 `example` 文件夹下

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

将本地 `test` 文件夹及其子文件夹下的所有非 `.md` 类型文件上传至 `bucket1`桶中的 `example` 文件夹下

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

将本地 `test` 文件夹及其子文件夹下的所有非 `.md` 和非 `.html` 类型文件上传至 `bucket1`桶中的 `example` 文件夹下

```
./coscli cp ~/test/ cos://bucket1/example/ -r --exclude ".*\.html$|.*\.md$"
```

本地 `dir` 文件夹下有 `dirA`、`dirB`、`dirC`、`dirD` 四个文件夹，将 `dir` 文件夹下除 `dirD` 文件夹之外的所有内容上传

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

将本地 `test` 文件夹下的所有文件及所有文件夹上传至 `bucket1`桶中的 `example` 文件夹下，并以归档类型文件存储

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

将本地的 `file.txt` 文件上传至 `bucket1`桶并设置单链接速率限制为1.3MB/s

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

下载操作

下载单文件

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

将 bucket1桶中的 example 文件夹下的所有文件及所有文件夹下载到本地 test 文件夹下

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

将 bucket1桶中的 example 文件夹及其子文件夹下所有 .mp4类型文件下载至本地 test 文件夹下

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

将 bucket1桶中的 example 文件夹及其子文件夹下所有非 .md 类型文件下载至本地 test 文件夹

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

bucket1桶中的 example 文件夹及其子文件夹下所有非 .md 和 非 .html 类型文件下载至本地 test 文件夹

```
./coscli cp cos://bucket1/example/ ~/test/ -r --exclude ".*\.html$|.*\.md$"
```

将bucket1桶中的 example.txt 文件的xxx版本下载至本地 test 目录下

```
./coscli cp cos://bucket1/example.txt ~/test/ --version-id xxx
```

拷贝操作

桶内拷贝单文件

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

跨桶拷贝单文件

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

将 bucket1桶中 example1文件夹下的所有文件及所有文件夹拷贝至 bucket2桶中 example2文件夹下

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

将 bucket1桶中 example1文件夹下及其子文件夹下的所有 .mp4类型文件拷贝至 bucket2桶中 example2文件夹下

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

将 bucket1桶中的 example1文件夹下及其子文件夹下所有非 .md 类型文件拷贝至 bucket2桶中 example2文件夹下

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

将bucket1桶中的 **example.txt** 文件的xxx版本拷贝至 bucket2桶中

```
./coscli cp cos://bucket1/example.txt cos://bucket2/ --version-id xxx
```

将bucket1桶中的 **test** 文件夹移动至 bucket2桶中

```
./coscli cp cos://bucket1/test/ cos://bucket2/test/ --move -r
```

同步上传下载或拷贝文件 – sync

最近更新时间：2025-07-03 10:57:02

sync 命令用于同步上传、下载或拷贝文件，与 cp 命令不同的是：sync 命令首先会对比同名文件的 crc64，如果 crc64 值相同则不进行传输。

⚠ 注意：

- 如需使用上传文件命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:HeadBucket`，`cos:GetBucket`，`cos:HeadObject`，`cos:InitiateMultipartUpload`，`cos:UploadPart`，`cos:CompleteMultipartUpload`，`cos:ListMultipartUploads`，`cos:ListParts`。
- 如需使用下载文件命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:HeadBucket`，`cos:GetBucket`，`cos:HeadObject`，`cos:GetObject`。
- 如需使用拷贝文件命令，在您进行 [授权策略](#) 时，目标对象 action 需要设置为 `cos:GetBucket`，`cos:HeadObject`，`cos:InitiateMultipartUpload`，`cos:PutObject`，`cos:CompleteMultipartUpload`。源对象 action 需要设置为 `cos:HeadBucket`，`cos:GetBucket`，`cos:HeadObject`，`cos:GetObject`。

更多授权请参见 [支持 CAM 的业务接口](#)。

命令格式

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

sync 命令包含以下参数：

参数格式	参数用途	示例
source_path	源文件路径。可以为本地路径或 COS 文件路径。COS 路径支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag。	- 本地路径：~/example.txt - 使用桶别名指定 COS 文件路径： cos://bucketalias/example.txt - 使用桶名称指定 COS 文件路径： cos://examplebucket-1250000000/example.txt
destination_path	目的文件路径。可以为本地路径或 COS 文件路径。COS 路径支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag。	- 本地路径：~/example.txt - 使用桶别名指定 COS 文件路径： cos://bucketalias/example.txt - 使用桶名称指定 COS 文件路径： cos://examplebucket-1250000000/example.txt

sync 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
无	--include	- 包含特定模式的文件（v1.0.4之前的版本在上传时仅筛选传入路径下的本地文件路径，v1.0.4以及之后版本会筛选传入的全路径） - 例如：需要上传./test 下的所有文件到 COS，./test下包含 aaa 文件夹，aaa 文件夹下包含1.txt 文件 1.0.4之前的版本会匹配 aaa/1.txt 1.0.4及之后版本会匹配 ./test/aaa/1.txt
无	--exclude	排除特定模式的文件（v1.0.4之前的版本在上传时仅筛选传入路径下的本地文件路径，v1.0.4以及之后版本会筛选传入的全路径）

		- 例如：需要上传 ./test 下的所有文件到 COS，./test 下包含 aaa 文件夹，aaa 文件夹下包含 1.txt 文件 1.0.4 之前的版本会匹配 aaa/1.txt 1.0.4 及之后版本会匹配 ./test/aaa/1.txt
-r	--recursive	是否递归遍历文件夹下所有文件。
无	--storage-class	指定上传文件的存储类型（默认 STANDARD），更多存储类型，请参见 存储类型概述 。
无	--part-size	文件的分块大小（默认 32MB，最大支持 5GB）。如果需要根据文件大小自适应分块大小，请设置为 0。
无	--thread-num	文件分块并发线程数（默认并发 5）。
无	--rate-limiting	单链接速率限制（0.1 – 100MB/s）。
无	--snapshot-path	指定保存上传或者下载文件时的快照信息所在的目录。在下次上传或者下载文件时，coscli 会读取指定目录下的快照信息进行增量上传或者下载。本选项用来加速目录文件同步。
无	--meta	上传文件的元信息。包括部分 HTTP 标准属性（HTTP Header）以及以 x-cos-meta- 开头的用户自定义元数据（User Meta）。文件元信息格式为 header:value#header:value，示例为 <pre>Expires:2022-10-12T00:00:00.000Z#Cache-Control:no-cache#Content-Encoding:gzip#x-cos-meta-x:x</pre>
无	--routines	指定文件间并发上传或下载的线程数量，默认为 3。
无	--fail-output	此选项决定是否启用上传或下载失败时文件的错误输出（默认为 true，开启）。如果启用，文件传输失败将被记录在指定目录内（如果没有指定，默认为 ./coscli_output）。如果禁用，只会将错误文件的数量输出到控制台。
无	--fail-output-path	此选项用于指定记录上传或下载失败文件的错误输出文件夹。通过提供自定义文件夹路径，您可以控制错误输出文件夹的位置和名称。如果未设置此选项，将使用默认的错误日志文件夹 ./coscli_output。
无	--retry-num	限频重试次数（默认为 0，不重试）。可选 1-10 次。当多台机器同时在一个 COS 目录上执行下载操作时，可以通过指定此参数进行重试来规避限频错误。
无	--err-retry-num	错误重试次数（默认为 0）。指定 1-10 次，或者设置为 0 表示不进行重试。
无	--err-retry-interval	重试间隔（仅在指定 --err-retry-num 为 1-10 时可用）。指定 1-10 秒的重试间隔，如果不指定或设置为 0，每次重试间隔将在 1-10 秒内的随机。
无	--only-current-dir	是否仅上传当前目录中的文件，忽略子目录及其内容（默认为 false，不忽略）。
无	--disable-all-symlink	是否上传时忽略所有软链接子文件和软链接子目录（默认为 true，不上传）。目前仅支持在 Linux 和 MacOS 系统上使用。
无	--enable-symlink-dir	是否上传软链接的子目录（默认为 false，不上传）。目前仅支持在 Linux 和 MacOS 系统上使用。
无	--disable-crc64	是否关闭 CRC64 数据验证（默认为 false，启用验证）。
无	--delete	删除指定目标路径中的任何其他文件，仅保留此次同步的文件（默认为 false，不删除）。建议在使用 --delete 选项之前启用版本控制，以防止意外删除数据。
无	--backup-dir	同步已删除文件的备份，用于保存目标端已删除但源端不存在的文件（仅下载时生效，且在 --delete=true 时必传）。上传及桶拷贝请使用版本控制来恢复意外删除数据。

无	<code>--force</code>	强制操作，不提示确认。（默认为 <code>false</code> ）
无	<code>--disable-checksum</code>	默认为 <code>false</code> ，校验整个文件的 <code>crc64</code> 。为 <code>true</code> 时仅校验分片 <code>crc64</code> 。

❗ 说明：

- `sync` 命令在上传和下载大文件时会自动启用并发上传/下载。
- 当文件大于 `--part-size` 时，COSCLI 会先将文件按 `--part-size` 进行切块，之后用 `--thread-num` 个线程并发地执行上传/下载任务。
- 每个线程都会维护一个链接，对于每个链接，您可以使用 `--rate-limiting` 参数对单链接进行限速，当启用并发上传/下载时，总速率为 `--thread-num * --rate-limiting`。
- 当以分块形式上传/下载文件时，会默认开启断点续传。
- `--include` 和 `--exclude` 支持标准正则表达式的语法，您可以使用它来过滤出符合特定条件的文件。
- 使用 `zsh` 时，您可能需要在 `pattern` 串的两端加上双引号。
- `snapshot-path` 不要设置为待迁移目录或其子目录。

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

- 用户在使用 `sync` 命令时，除了产生 `PUT` 请求费用，还有以下两种情况对云上文件有 `HEAD` 请求，而产生额外的费用：
 - 没有添加指定快照目录的参数（`--snapshot-path`），会产生 `HEAD` 请求费用。
 - 已添加指定快照目录参数（`--snapshot-path`）且属于首次生成快照目录，会产生 `HEAD` 请求费用。若非首次生成的快照目录则不会产生额外的请求费用。
- 关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

操作示例

同步上传文件

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

同步下载文件

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

桶内同步拷贝文件

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

跨桶同步拷贝文件

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

删除文件 – rm

最近更新时间：2025-04-25 14:24:32

rm 命令用于删除文件。

注意：

- 使用 rm 命令，请下载 COSCLI V1.0.1及以上版本，详情请参见 [下载与安装配置](#)。
- 若您当前 COSCLI 的版本为 V1.0.0，请您务必升级版本至 V1.0.1后，再执行 rm 删除命令。V1.0.0版本在执行 rm 命令时，--include 和--exclude 参数不生效，可能会导致非预期的删除情况。
- 如需使用此命令删除对象，在您进行 [授权策略](#) 时，action 需要设置为 cos:HeadBucket ， cos:HeadObject ， cos:GetBucket ， cos>DeleteObject ， cos>DeleteMultipleObjects 。
- 如需使用此命令删除历史版本（如传入--all-versions 或--version-id ），在您进行 [授权策略](#) 时，action 需要设置为 cos:HeadBucket ， cos:HeadObject ， cos:GetBucket ， cos>DeleteObject ， cos>DeleteMultipleObjects ， cos:GetBucketVersioning ， cos:GetBucketObjectVersions 。

更多授权请参见 [支持 CAM 的业务接口](#)，如有任何疑问或需要进一步的帮助，请 [联系我们](#)。

命令格式

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

rm 命令包含以下参数：

参数格式	参数用途	示例
cos://<bucket-name>	指定需要访问的存储桶。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 endpoint flag。	使用桶别名访问：cos://example-alias 使用桶名称访问：cos://examplebucket-1250000000
/prefix/	可选参数。指定某一文件夹。	/picture/

rm 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法。
无	--include	包含特定模式的文件。
无	--exclude	排除特定模式的文件。
-r	--recursive	是否递归地遍历文件夹下所有文件。
-f	--force	强制删除（删除文件前不弹出确认信息）。
无	--fail-output	此选项决定是否启用删除文件时的错误输出（默认为 true ， 开启）。如果启用，文件删除失败将被记录在指定目录内（如果没有指定，默认为 ./coscli_output ）。如果禁用，只会将删除错误文件的数量输出到控制台。
无	--fail-output-path	此选项用于指定记录删除文件时的错误输出文件夹。通过提供自定义文件夹路径，您可以控制错误输出文件夹的位置和名称。如果未设置此选项，将使用默认的错误日志文件夹 ./coscli_output 。
无	--all-versions	仅支持在已开启版本控制的Bucket内且传入 --recursive（-r）参数时可用，遍历删除指定路径下所有版本

无	--version-id	仅支持在已开启版本控制的Bucket内且未传入--recursive (-r) 参数时可用，删除指定object的指定版本
---	--------------	---

说明：

- --include 和 --exclude 支持标准正则表达式的语法，您可以使用它来过滤出符合特定条件的文件。
- 使用 zsh 时，您可能需要在 pattern 串的两端加上双引号。

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

- 关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

操作示例

删除 fig1.png 文件

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

删除 picture 文件夹及包含的所有内容

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

删除 fig1.png 文件的指定版本

```
./coscli rm cos://bucket1/fig1.png --version-id xxx
```

删除 test 前缀的所有版本

```
./coscli rm cos://bucket1/test -r --all-versions
```

获取文件哈希值 – hash

最近更新时间：2025-04-02 19:21:32

hash 命令用于计算本地文件的哈希值或获取对象存储（Cloud Object Storage，COS）文件的哈希值。

⚠ 注意：

如需使用此命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:HeadObject`，更多授权请参见 [支持 CAM 的业务接口](#)。

命令格式

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

hash 命令包含以下参数：

参数格式	参数用途	示例
<object-name>	用于指定访问的文件。可以为本地路径或 COS 文件路径。COS 路径支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag	本地路径：~/example.txt 使用桶别名指定 COS 文件路径： cos://bucketalias/example.txt 使用桶名称指定 COS 文件路径： cos://examplebucket-1250000000/example.txt

hash 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--type	哈希类型（md5或 crc64，默认为 crc64） 注意：md5仅能获取 etag，如需获取文件 md5，可以下载完整文件后计算

❗ 说明：

关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

操作示例

计算本地文件的 crc64

```
./coscli hash ~/test.txt
```

获取 COS 文件的 md5

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

列出分块上传中产生的碎片 – lsparts

最近更新时间：2025-02-28 18:13:55

lsparts 命令列出分块上传中产生的文件碎片。

⚠ 注意：

- 如需用此命令列出分块上传任务，在您进行 [授权策略](#) 时，action 需要设置为 `cos:PutBucket`。
- 如需使用此命令列出分块上传任务对应的文件碎片，在您进行 [授权策略](#) 时，action 需要设置为 `cos:ListMultipartUploads`，`cos:ListParts`。

更多授权请参见 [支持 CAM 的业务接口](#)

命令格式

```
./coscli lsparts cos://<bucket-name>[/prefix/] [flag]
```

lsparts 命令包含以下参数：

参数格式	参数用途	示例
cos://<bucket-name>	指定需要访问的存储桶。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag。	使用桶别名访问：cos://example-alias 使用桶名称访问：cos://examplebucket-1250000000
/prefix/	可选参数。指定某一文件夹	/picture/

lsparts 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件
无	--limit	指定列出的最大数量（0 - 1000）
无	--upload-id	上传记录 id。当传入 uploadid 时，需指定具体文件，仅展示 uploadid 对应的文件碎片信息，未传入 uploadid 时，展示指定 prefix 下正在进行中的分块上传任务

📌 说明：

- 关于分块上传的更多信息，请参见 [分块上传](#)。
- 关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

操作示例

列出 bucket1 桶内的所有正在进行中的分块上传任务

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

返回的结果示例如下，输出的信息包括：对象键（对象在存储桶中的唯一标识）、分块上传的 ID、分块上传的起始时间、桶内碎片总数量。

KEY	UPLOAD ID	INITIATE TIME
test.txt	1671191183635d2b71b1d68a0*****	2022-01-01T00:00:00.000Z
		TOTAL: 1

列出 bucket1桶内 test.txt 文件 upload_id 为 1671191183635d2b71b1d68a0***** 的所有上传碎片

```
./coscli lsparts cos://bucket1/test.txt --upload-id="1671191183635d2b71b1d68a0*****"
```

返回的结果示例如下，输出的信息包括：PARTNUMBER（块的编号）、ETAG（块的 MD-5 算法校验值）、LastModified（分块最后被修改时间）、SIZE（分块大小）。

PARTNUMBER	ETAG	LAST MODIFIED	SIZE
1	"58f06dd588d8ffb3beb46ada6309436b"	2024-12-17T16:34:48+08:00	32.00 MB
2	"58f06dd588d8ffb3beb46ada6309436b"	2024-12-17T16:34:48+08:00	32.00 MB
3	"58f06dd588d8ffb3beb46ada6309436b"	2024-12-17T16:34:48+08:00	32.00 MB
4	"58f06dd588d8ffb3beb46ada6309436b"	2024-12-17T16:34:48+08:00	32.00 MB
			TOTAL: 4

清理碎片 – abort

最近更新时间：2025-02-28 18:13:55

abort 命令用于清理分块上传过程中产生的文件碎片。

 **注意：**

如需使用此命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:ListMultipartUploads`，`cos:AbortMultipartUpload`。更多授权请参见 [支持 CAM 的业务接口](#)。

命令格式

```
./coscli abort cos://<bucket-name>[/prefix/] [flag]
```

abort 命令包含以下参数：

参数格式	参数用途	示例
cos://<bucket-name>/<key>	指定需要存储桶中的对象。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> <code>flag</code>	使用桶别名访问：cos://example-alias 使用桶名称访问： cos://examplebucket-1250000000
/prefix/	可选参数。指定某一文件夹	/picture/

abort 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件
无	--fail-output	此选项决定是否启用清理文件碎片时的错误输出（默认为 <code>true</code> ，开启）。如果启用，文件碎片清理失败将被记录在指定目录内（如果没有指定，默认为 <code>./coscli_output</code> ）。如果禁用，只会将清理失败的数量输出到控制台
无	--fail-output-path	此选项用于指定记录清理文件碎片时的错误输出文件夹。通过提供自定义文件夹路径，您可以控制错误输出文件夹的位置和名称。如果未设置此选项，将使用默认的错误日志文件夹 <code>./coscli_output</code>

 **说明**

关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

操作示例

清除 bucket1桶内的所有文件碎片

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

清除 bucket1桶内 picture 文件夹下的所有碎片

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


取回归档文件 – restore

最近更新时间：2025-02-28 18:13:55

restore 命令用于取回归档文件。

注意：

如需使用此命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:HeadBucket`，`cos:GetBucket`，`cos:PostObjectRestore`，更多授权请参见 [支持 CAM 的业务接口](#)。

命令格式

```
./coscli restore cos://<bucket-name>[/prefix/] [flag]
```

restore 命令包含以下参数：

参数格式	参数用途	示例
cos://<bucket-name>	指定需要访问的存储桶。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag	使用桶别名访问：cos://example-alias 使用桶名称访问：cos://examplebucket-1250000000
/prefix/	可选参数。指定某一文件夹	/picture/

restore 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件
-d	--days	指定解冻对象所生成的临时文件的过期时间（默认为3天）
-m	--mode	指定恢复模式（默认 Standard） - 若恢复的是归档存储类型数据，可选值为 Expedited、Standard、Bulk，分别对应快速取回模式、标准取回模式以及批量取回模式这三种模式 - 若恢复的是深度归档存储类型数据，则可选值为 Standard、Bulk
-r	--recursive	递归遍历文件夹
无	--fail-output	此选项决定是否启用回热失败时文件的错误输出（默认为 <code>true</code> ，开启）。如果启用，文件回热失败将被记录在指定目录内（如果没有指定，默认为 <code>./coscli_output</code> ）。如果禁用，只会将错误文件的数量输出到控制台
无	--fail-output-path	此选项用于指定记录回热失败文件的错误输出文件夹。通过提供自定义文件夹路径，您可以控制错误输出文件夹的位置和名称。如果未设置此选项，将使用默认的错误日志文件夹 <code>./coscli_output</code>

说明

- `--include` 和 `--exclude` 支持标准正则表达式的语法，您可以使用它来过滤出符合特定条件的文件。
- 使用 `zsh` 时，您可能需要在 `pattern` 串的两端加上双引号。

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

- 关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。
- 关于取回归档文件的更多信息，请参见 [POST Object restore](#) 文档。

操作示例

以标准取回模式取回 bucket1桶内的归档文件

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

以快速取回模式取回 bucket1桶内 picture 文件夹下的所有归档文件

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

获取预签名 URL – signurl

最近更新时间：2023-02-03 16:21:55

signurl 命令用于获取某个对象的预签名 URL，可以通过此 URL 匿名访问对象。

命令格式

```
./coscli signurl cos://<bucket-name>/<key> [flag]
```

signurl 命令包含以下参数：

参数格式	参数用途	示例
cos://<bucket-name>/<key>	指定需要存储桶中的对象。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag。	使用桶别名访问：cos://example-alias/test.txt 使用桶名称访问：cos://examplebucket-1250000000/test.txt

signurl 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
-t	--time	设置 URL 过期时间（默认1000s）

说明

关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

操作示例

获取 bucket1 桶内 picture.jpg 的预签名 URL

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

获取 bucket2 桶内 picture.jpg 的预签名 URL，并设置 URL 的过期时间为1314秒

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

创建/获取软链接 – symlink

最近更新时间：2025-02-28 18:13:55

symlink 命令用于创建或获取某个对象的软链接，可以通过此软链接快速找到对象。

⚠ 注意：

- 如需使用创建软链接命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:PutObject`。
- 如需使用获取软链接命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:GetObject`。

更多授权请参见 [支持 CAM 的业务接口](#)。

命令格式

```
./coscli symlink --method create cos://<bucket-name>/<key> --link linkKey [flag]
```

symlink 命令包含以下参数：

参数格式	参数用途	示例
cos://<bucket-name>	需要指定存储桶。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag	使用桶别名访问：cos://example-alias 使用桶名称访问：cos://examplebucket-1250000000
/<key>	创建软链接时需指定对象名	/test

symlink 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--method	可选值 create/get
无	--link	软链接 key 值

操作示例

为 bucket1桶内 picture.jpg 创建软链接symlink

```
./coscli symlink --method create cos://bucket1/picture.jpg --link symlink
```

通过软链接 symlink 获取指向的对象

```
./coscli symlink --method get cos://bucket2 --link symlink
```

查看对象内容 – cat

最近更新时间：2025-04-17 18:19:12

cat 命令用于查看某个对象的内容。

注意：

如需使用此命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:GetObject`，更多授权请参见 [支持 CAM 的业务接口](#)。

命令格式

```
./coscli cat cos://<bucket-name>/<key> [flag]
```

cat 命令包含以下参数：

参数格式	参数用途	示例
cos://<bucket-name>/<key>	指定需要存储桶中的对象。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 endpoint flag	使用桶别名访问：cos://example-alias/test.txt 使用桶名称访问：cos://examplebucket-1250000000/test.txt

cat 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法

操作示例

查看 bucket1桶内 test.txt 的内容

```
./coscli cat cos://bucket1/test.txt
```

列出目录下内容并统计 – lsdu

最近更新时间：2025-04-17 18:19:12

lsdu 命令用于获取指定前缀当前层级下的内容，并统计其中各个 object 及目录中 object 的大小及数量。

 **注意：**

如需使用此命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:HeadBucket` 、 `cos:GetBucket` ，更多授权请参见 [支持 CAM 的业务接口](#)。

命令格式

```
./coscli lsdu cos://<bucket-name>[/prefix/] [flags]
```

lsdu 命令包含以下参数：

参数格式	参数用途	示例
cos://<bucket-name>	指定需要访问的存储桶。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> <code>flag</code>	使用桶别名访问：cos://example-alias 使用桶名称访问：cos://examplebucket-1250000000
/prefix/	可选参数。指定某一文件夹	/picture/

lsdu 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件

操作示例

获取 bucket1桶根目录下的统计信息

```
./coscli lsdu cos://bucket1/
```

返回的信息包括：目录或 object 名，总计对象数量，总计大小。示例如下：

```
NAME | OBJECTS COUNT | TOTAL SIZE
-----+-----+-----
300123/ | 1 | 300.00 MB
300s/ | 1 | 100.00 MB
300u/ | 301 | 8.22 GB
activity/ | 35 | 129.08 KB
test/ | 1 | 3 B
test100/ | 20 | 20.00 GB
test5/ | 6 | 9.00 GB
testrm/ | 1 | 0 B
10GB_file | 1 | 11.72 GB
1GB_file | 1 | 1.00 GB
```

获取 bucket1 桶 picture 目录下的统计信息

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

❗ 说明:

文件较多时此命令耗时会较长，建议后台执行。

版本控制 – bucket-versioning

最近更新时间：2025-04-07 16:56:32

bucket-versioning 命令用于管理存储桶的版本控制功能，可以用来开启、暂停和查看存储桶的版本控制状态。

⚠ 注意：

- 如需使用开启/暂停版本控制命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:PutBucketVersioning`。
 - 如需使用查看桶版本控制状态命令，在您进行 [授权策略](#) 时，action 需要设置为 `cos:GetBucketVersioning`。
- 更多授权请参见 [支持 CAM 的业务接口](#)。

命令格式

以下 bucket-versioning 命令用于开启与暂停版本控制：

```
./coscli bucket-versioning --method [method] cos://<bucket-name> versioning
```

bucket-versioning 及其子命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
-c	--method	可选值 put/get

操作示例

开启存储桶版本控制

```
./coscli bucket-versioning --method put cos://examplebucket Enabled
```

暂停存储桶版本控制

```
./coscli bucket-versioning --method put cos://examplebucket Suspended
```

查看桶版本控制状态

```
./coscli bucket-versioning --method get cos://examplebucket
```


常见问题

最近更新时间：2023-09-20 21:18:32

COSCLI 工具与 COSCMD 工具有什么区别？

1. COSCLI 工具使用 golang 构建，直接发布编译后的二进制包，用户在安装部署时无需预先安装任何依赖，开箱即用；COSCMD 工具使用 Python 构建，用户在安装时需先安装 Python 环境和依赖包。
2. COSCLI 工具支持设置存储桶别名，可以使用一个短字符串来代替 `<BucketName-APPID>`，方便用户使用；COSCMD 工具不支持存储桶别名，用户需要输入 `<BucketName-APPID>` 来指定一个存储桶，命令繁琐且不易阅读。
3. COSCLI 工具支持在配置文件内配置多个存储桶，且支持跨桶操作；COSCMD 工具在配置文件中只能配置一个存储桶，且跨桶操作命令过于冗长。

COS Migration 工具

最近更新时间：2024-11-06 17:30:22

功能说明

COS Migration 是一个集成了 COS 数据迁移功能的一体化工具。通过简单的配置操作，用户可以将本地数据迁移至 COS 中，它具有以下特点：

- 断点续传：工具支持上传时断点续传。对于一些大文件，如果中途退出或者因为服务故障，可重新运行工具，会对未上传完成的文件进行续传。
- 分块上传：将对象按照分块的方式上传到 COS。
- 并行上传：支持多个对象同时上传。

⚠ 注意：

- COS Migration 已经停止迭代更新，不再发布新版本，后续将下线。本地数据迁移场景请使用 [COSCLI](#) 的 cp 或者 sync 命令，其他迁移场景推荐使用 [云迁移](#)。
- COS Migration 的编码格式只支持 UTF-8 格式。
- 若源数据中有空目录、软链接文件或目录，此类数据不会被迁移。
- 使用该工具上传同名文件，默认会覆盖较旧的同名文件，需要额外设置以跳过同名文件。
- COS Migration 是用来做一次性迁移服务的，不适合于持续同步的场景。例如本地每天新增文件，需要持续同步至 COS 中，COS Migration 为了避免重复迁移任务，会保存迁移成功的记录，持续同步后，扫描记录时间会持续增大。此种场景可以尝试使用 [文件同步](#)。

使用环境

系统环境

Windows 和 Linux 系统。

软件依赖

- JDK 1.8 X64或以上，有关 JDK 的安装与配置请参见 [Java 安装与配置](#)。
- Linux 环境需要 IFUNC 支持，确保环境 binutils 版本大于 2.20。

使用方法

1. 获取工具

前往下载 [COS Migration 工具](#)。

2. 解压缩工具包

Windows

解压并保存到某个目录，例如：

```
C:\Users\Administrator\Downloads\cos_migrate
```

Linux

解压并保存到某个目录：

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

迁移工具结构

正确解压后的 COS Migration 工具目录结构如下所示：

```
COS_Migrate_tool
|--conf    #配置文件所在目录
|   |--config.ini  #迁移配置文件
|--db      #存储迁移成功的记录
|--dep     #程序主逻辑编译生成的 JAR 包
|--log     #工具执行中生成的日志
|--opbin   #用于编译的脚本
|--result  #用于保存迁移成功记录的目录，记录文件名为 "日期.out"，格式为 "绝对路径\t文件大小\t最后修改时间"
|--src     #工具的源码
|--tmp     #临时文件存储目录
|--.gitignore  #git版本管理忽略的文件与文件夹
|--pom.xml  #项目配置文件
|--README  #说明文档
|--start_migrate.sh  #Linux 下迁移启动脚本
|--start_migrate.bat  #Windows 下迁移启动脚本
```

❗ 说明：

- db 目录主要记录工具迁移成功的文件标识，每次迁移任务会优先对比 db 中的记录，若当前文件标识已被记录，则会跳过当前文件，否则进行文件迁移。
- log 目录记录着工具迁移时的所有日志，若在迁移过程中出现错误，请先查看该目录下的 error.log。

3. 修改 config.ini 配置文件

在执行迁移启动脚本之前，需先进行 config.ini 配置文件修改（路径：`./conf/config.ini`），config.ini 内容可以分为以下几部分：

3.1 配置迁移类型

type 表示迁移类型，固定填写 `type=migrateLocal`。

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

3.2 配置迁移任务

用户根据实际的迁移需求进行相关配置，主要包括迁移至目标 COS 信息配置及迁移任务相关配置。

```
# 迁移工具的公共配置分节，包含了需要迁移到目标 COS 的账户信息。
[common]
secretId=COS_SECRETID
secretKey=COS_SECRETKEY
bucketName=examplebucket-1250000000
region=ap-guangzhou
storageClass=Standard
cosPath=/
https=off
tmpFolder=./tmp
smallFileThreshold=5242880
smallFileExecutorNum=64
bigFileExecutorNum=8
entireFileMd5Attached=off
executeTimeWindow=00:00,24:00
outputFinishedFileFolder=./result
resume=false
skipSamePath=false
```

requestTryCount=5

名称	描述	默认值
secretId	用户密钥 SecretId, 请将 COS_SECRETID 替换为您的真实密钥信息。可前往 访问管理控制台 中的云 API 密钥页面查看获取	-
secretKey	用户密钥 SecretKey, 请将 COS_SECRETKEY 替换为您的真实密钥信息。可前往 访问管理控制台 中的云 API 密钥页面查看获取	-
bucketName	目的 Bucket 的名称, 命名格式为 <BucketName-APPID>, 即 Bucket 名必须包含 APPID, 例如 examplebucket-1250000000	-
region	目的 Bucket 的 Region 信息。COS 的地域简称请参照 地域和访问域名	-
storageClass	数据迁移后的存储类型, 可选值为 Standard (标准存储)、Standard_IA (低频存储)、Archive (归档存储)、Maz_Standard (标准存储多 AZ)、Maz_Standard_IA (低频存储多 AZ), 相关介绍请参见 存储类型概述	Standard
cosPath	要迁移到的 COS 路径。/ 表示迁移到 Bucket 的根路径下, /folder/doc/ 表示要迁移到 Bucket 的 /folder/doc/ 下, 若 /folder/doc/ 不存在, 则会自动创建路径	/
https	是否使用 HTTPS 传输: on 表示开启, off 表示关闭。开启传输速度较慢, 适用于对传输安全要求高的场景	off
tmpFolder	从其他云存储迁移至 COS 的过程中, 用于存储临时文件的目录, 迁移完成后会删除。要求格式为绝对路径: Linux 下分隔符为单斜杠, 例如 /a/b/c Windows 下分隔符为两个反斜杠, 例如 E:\\a\\b\\c 默认为工具所在路径下的 tmp 目录	./tmp
smallFileThreshold	小文件阈值的字节, 大于等于这个阈值使用分块上传, 否则使用简单上传, 默认5MB (5242880 Byte)	5242880
smallFileExecutorNum	小文件 (文件小于 smallFileThreshold) 的并发度, 使用简单上传。如果是通过外网来连接 COS, 且带宽较小, 请减小该并发度	64
bigFileExecutorNum	大文件 (文件大于等于 smallFileThreshold) 的并发度, 使用分块上传。如果是通过外网来连接 COS, 且带宽较小, 请减小该并发度	8
entireFileMd5Attached	表示迁移工具将全文的 MD5 计算后, 存入文件的自定义头部 x-cos-meta-md5 中, 用于后续的校验, 因为 COS 的分块上传的大文件的 etag 不是全文的 MD5	off
executeTimeWindow	执行时间窗口, 时刻粒度为分钟, 该参数定义迁移工具每天执行的时间段。例如: 参数 03:30,21:00, 表示在凌晨 03:30 到晚上 21:00 之间执行任务, 其他时间则会进入休眠状态, 休眠状态暂停迁移会保留迁移进度, 直到下一个时间窗口自动继续执行。注意后面的时间点必须大于前面的时间点。	00:00,24:00
outputFinishedFileFolder	这个目录保存迁移成功的结果, 结果文件会按照日期命名, 例如 ./result/2021-05-27.out, 其中 ./result 为已创建的目录。文件内容每一行的格式为: 绝对路径\t文件大小\t最后修改时间。设置为空, 则不输出结果。	./result
resume	是否接着最后一次运行的结果, 继续往下遍历源的文件列表。默认从头开始。	false
skipSamePath	如果 COS 上已经有相同的文件名, 是否直接跳过。默认不跳过, 即覆盖原有文件。	false
requestTryCount	每个文件上传总的尝试次数。	5

3.3 配置数据源信息

3.3.1 配置本地数据源 migrateLocal

若从本地迁移至 COS，则进行该部分配置，具体配置项及说明如下：

```
# 从本地迁移到 COS 配置分节
[migrateLocal]
localPath=E:\\code\\java\\workspace\\cos_migrate_tool\\test_data
excludes=
ignoreModifiedTimeLessThanSeconds=
```

配置项	描述
localPath	本地目录，要求格式为绝对路径： - Linux 下分隔符为单斜杠，例如 <code>/a/b/c</code> - Windows 下分隔符为两个反斜杠，例如 <code>E:\\a\\b\\c</code> 注意：此参数只能填目录的路径，不能填具体文件的路径，否则会导致目标对象名解析错误，在 <code>cosPath=/</code> 情况下，还会错误地解析成创桶请求
excludes	要排除的目录或者文件的绝对路径，表示将 localPath 下面某些目录或者文件不进行迁移，多个绝对路径之前用分号分割，不填表示 localPath 下面的全部迁移
ignoreModifiedTimeLessThanSeconds	排除更新时间与当前时间相差不足一定时间段的文件，单位为秒，默认不设置，表示不根据 lastmodified 时间进行筛选，适用于客户在更新文件的同时又在运行迁移工具，并要求不把正在更新的文件迁移上传到 COS，例如设置为300，表示只上传更新了5分钟以上的文件

4. 运行迁移工具

Windows

双击 `start_migrate.bat` 即可运行。

Linux

1.从 config.ini 配置文件读入配置，运行命令为：

```
sh start_migrate.sh
```

2.部分参数从命令行读入配置，运行命令为：

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

❗ 说明：

- 工具支持配置项读取方式有两种：命令行读取或配置文件读取。
- 命令行优先级高于配置文件，即相同配置选项会优先采用命令行里的参数。
- 命令行中读取配置项的形式方便用户同时运行不同的迁移任务，但前提是两次任务中的关键配置项不完全一样，例如 Bucket 名称，COS 路径，要迁移的源路径等。因为不同的迁移任务写入的是不同的 db 目录，可以保证并发迁移。请参照前文中的工具结构中的 db 信息。
- 配置项的形式为 `-D{sectionName}.{sectionKey}={sectionValue}` 的形式。其中 sectionName 是配置文件的分节名称，sectionKey 表示分节中配置项名称，sectionValue 表示分节中配置项值。如设置要迁移到的 COS 路径，则以 `-Dcommon.cosPath=bbb/ddd` 表示。

迁移机制及流程

迁移机制原理

COS 迁移工具是有状态的，已经迁移成功的会记录在 db 目录下，以 KV 的形式存储在 leveldb 文件中。每次迁移前对要迁移的路径，先查找下 db 中是否存在，如果存在，且 mtime 和 db 中存在的一致，则跳过迁移，否则进行迁移。因此，我们参照 db 中是否有过迁移成功的记录，而不是查找 COS，如果绕过了迁移工具，通过别的方式（如 COSCMD 或者控制台）删除修改了文件，那么运行迁移工具由于不会察觉到这种变化，是不会重新迁移的。

迁移流程步骤

1. 读取配置文件，根据迁移 type，读取相应的配置分节，并执行参数的检查。
2. 扫描对比 db 下对所迁移文件的标识，判断是否允许上传。
3. 迁移执行过程中会打印执行结果，其中 inprogress 表示迁移中，skip 表示跳过，fail 表示失败，ok 表示成功，condition_not_match 表示因不满足迁移条件而跳过的文件（如 lastmodified 和 excludes）。失败的详细信息可以在 log 的 error 日志中查看。
4. 整个迁移结束后会打印统计信息，包括累积的迁移成功量，失败量，跳过量，耗时。对于失败的情况，请查看 error 日志，或重新运行，因为迁移工具会跳过已迁移成功的，对未成功的会重新迁移。

常见问题

如您在使用 COS Migration 工具过程中，遇到迁移失败、运行报错等异常情况，请参见 [COS Migration 工具类常见问题](#) 寻求解决。

结语

当然，COS 不仅提供以上应用和服务，还提供多款热门开源应用，并集成腾讯云 COS 插件，欢迎点击 [此处](#) 一键启动，立即使用！

FTP Server 工具

最近更新时间：2024-11-21 16:05:02

功能概述

COS FTP Server 支持通过 FTP 协议直接操作对象存储（Cloud Object Storage，COS）中的对象和目录，包括上传文件、下载文件、删除文件以及创建文件夹等。FTP Server 工具使用 Python 实现，使安装更加简单。

功能说明

上传机制：流式上传，不落本地磁盘，只要按照标准的 FTP 协议配置工作目录即可，不占用实际的磁盘存储空间。

下载机制：直接流式返回给客户端。

目录机制：bucket 作为整个 FTP Server 的根目录，bucket 下面可以建立若干个子目录。

多 bucket 绑定：支持同时绑定多个 bucket。

❗ 说明：

多 bucket 绑定：通过不同的 FTP Server 工作路径（home_dir）来实现，因此，指定不同的 bucket 和用户信息时必须保证 home_dir 不同。

删除操作限制：在新的 FTP Server 中可以针对每个 ftp 用户配置 delete_enable 选项，以标识是否允许该 FTP 用户删除文件。

支持的 FTP 命令：put、mput、get、rename、delete、mkdir、ls、cd、bye、quite、size。

不支持的 FTP 命令：append、mget（不支持原生的 mget 命令，但在某些 Windows 客户端下，仍然可以批量下载，例如 FileZilla 客户端。）

❗ 说明：

FTP Server 工具暂时不支持断点续传功能。

开始使用

系统环境

- 操作系统：Linux，推荐使用腾讯 CentOS 系列 [云服务器](#)，暂时不支持 Windows 系统。其中，setuptools 需要升级至最新版本（>=44.1.1）。
- psutil 依赖的 Linux 系统包：python-devel（或 python-dev，依据不同的 Linux 发行版名字不同），通过 Linux 下的包管理工具添加，例如 `yum install python-devel` 或 `aptitude install python-dev`。
- Python 解释器版本：Python 2.7，请参见 [Python 安装与配置](#) 进行安装与配置。

❗ 说明：

FTP Server 工具不支持 Python 3。

- 依赖包：
- [cos-python-sdk-v5](#)：1.6.5及以上版本。
- [pyftplib](#)：1.5.7及以上版本。

❗ 说明：

如果您的服务器所使用的 pyftplib 版本为 1.5.9，可能出现版本不兼容导致报错421，请尝试使用 pyftplib 1.5.7版本。

- [psutil](#)：5.6.1及以上版本。

使用限制

适用于 COS XML 版本。

安装运行

FTP Server 工具下载地址为：[cos-ftp-server](#)。安装步骤如下：

1. 进入 FTP Server 目录，运行 setup.py 安装 FTP Server 及其相关的依赖库（需要联网）：

```
python setup.py install # 这里可能需要您的账号 sudo 或者拥有 root 权限。
```

2. 将配置示例文件 `conf/vsftpd.conf.example` 复制命名为 `conf/vsftpd.conf`，参考本文 [配置文件](#) 章节，正确配置 bucket 和用户信息。

3. 运行 ftp_server.py 启动 FTP Server：

```
python ftp_server.py
```

此外还有两种方式启动 FTP Server，如下：

- 使用 nohup 命令，以后台进程方式启动：

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

- 使用 screen 命令放入后台运行（需要安装 screen 工具）：

```
screen -dmS ftp
screen -r ftp
python ftp_server.py
#使用快捷键 Ctrl+A+D，切回主 screen 即可。
```

停止运行

- 若您是直接运行，或 screen 方式放在后台运行的 FTP Server，您可以使用快捷键 `Ctrl+C` 停止 FTP Server 运行。
- 若您是通过 nohup 命令启动，可以使用下面方式停止：

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

配置文件

FTP Server 工具的配置示例文件为 `conf/vsftpd.conf.example`，请复制命名为 `vsftpd.conf`，并按照以下的配置项进行配置：

```
[COS_ACCOUNT_0]
cos_secretid = COS_SECRETID # 替换为您的 SECRETID
cos_secretkey = COS_SECRETKEY # 替换为您的 SECRETKEY
cos_bucket = examplebucket-1250000000
cos_region = region # 替换为您的存储桶地域
cos_protocol = https
#cos_endpoint = region.myqcloud.com
home_dir = /home/user0 # 替换为您希望 FTP 挂载到的本地路径（需设置为机器上实际存在的路径，不支持软链接）
ftp_login_user_name=user0 # 替换为用户自定义的账号
ftp_login_user_password=pass0 # 替换为用户自定义的密码
authority=RW # 设置该用户的读写权限，R 表示读权限，W 表示写权限，RW 表示同时具备读写权限
delete_enable=true # true 为允许该 ftp 用户进行删除操作（默认），false 为禁止该用户进行删除操作

[COS_ACCOUNT_1]
cos_secretid = COS_SECRETID # 替换为您的 SECRETID
cos_secretkey = COS_SECRETKEY # 替换为您的 SECRETKEY
```



```
cos_bucket = examplebucket-1250000000
cos_region = region    # 替换为您的存储桶地域
cos_protocol = https
#cos_endpoint = region.myqcloud.com
home_dir = /home/user1    # 替换为您希望 FTP 挂载到的本地路径（需设置为机器上实际存在的路径，不支持软链接）
ftp_login_user_name=user1 # 替换为用户自定义的账号
ftp_login_user_password=pass1 # 替换为用户自定义的密码
authority=RW              # 设置该用户的读写权限，R 表示读权限，W 表示写权限，RW 表示同时具备读写权限
delete_enable=false       # true 为允许该 ftp 用户进行删除操作（默认），false 为禁止该用户进行删除操作

[NETWORK]
# 如果 FTP Server 处于某个网关或 NAT 后，可以通过该配置项将网关的 IP 地址或域名指定给 FTP
masquerade_address = XXX.XXX.XXX.XXX
# FTP Server 的监听端口，默认为2121，注意防火墙需要放行该端口（例如您是将 FTP Server 工具部署在腾讯云 CVM，则需要
# 在 CVM 安全组放行该端口）
listen_port = 2121
# passive_port 可以设置 passive 模式下，端口的选择范围，默认在 [60000, 65535] 区间上选择，注意防火墙（例如
# CVM 安全组）需要放行此区间端口
passive_port = 60000,65535

[FILE_OPTION]
# 默认单文件大小最大支持到200G，不建议设置太大
single_file_max_size = 21474836480

[OPTIONAL]
# 以下设置，如无特殊需要，建议保留 default 设置。如需设置，请合理填写一个整数
min_part_size      = default
upload_thread_num  = default
max_connection_num = 512
max_list_file      = 10000    # ls 命令最大可列出的文件数目，建议不要设置太大，否则 ls 命令延
# 时会很高
log_level          = INFO     # 设置日志输出的级别
log_dir            = log      # 设置日志的存放目录，默认是在 FTP Server 目录下的 log 目录中
```

❗ 说明：

- 如果要将每个用户绑定到不同的 bucket 上，则只需要添加 [COS_ACCOUNT_X] 的 section 即可。
针对每个不同的 COS_ACCOUNT_X 的 section 有如下说明：
 - 每个 ACCOUNT 下的用户名（ftp_login_user_name）和用户的主目录（home_dir）必须各不相同，并且主目录必须是系统中真实存在的目录。
 - 每个 COS FTP Server 允许同时登录的用户数目不能超过100。
 - endpoint 和 region 不会同时生效，使用公有云 COS 服务只需要正确填写 region 字段即可，endpoint 常用于私有化部署环境中。当同时填写了 region 和 endpoint，则 endpoint 会优先生效。
- 配置文件中的 OPTIONAL 选项是提供给高级用户用于调整上传性能的可选项，根据机器的性能合理地调整上传分片的大小和并发上传的线程数，可以获得更好的上传速度，一般用户不需要调整，保持默认值即可。同时，提供最大连接数的限制选项。这里如果不想限制最大连接数，可以填写0，即表示不限制最大连接数目（不过需要根据您机器的性能合理评估）。
- 在配置文件的 masquerade_address 配置项中，一般建议指定为客户端连接 COS FTP Server 所使用的 IP 地址。如您对此有疑问，可参见 [FTP Server 工具 常见问题文档](#)。
 - 当 FTP Server 有多个 IP 地址时，执行 ifconfig 命令后，得到映射到外网的网卡 IP 为10.xxx.xxx.xxx，它映射的外网 IP 假设为119.xxx.xxx.xxx。此时，若 FTP Server 未显式配置 masquerade_address 为客户端访问 server 时的外网IP（119.xxx.xxx.xxx），则 FTP Server 在 Passive 模式下，给客户端回包可能会使用内网地址（10.xxx.xxx.xxx）。这时就会出现客户端能够连上 FTP Server，但却不能正常给客户端返回数据包的情况。因此，通常情况下，建议用户将 masquerade_address 配置为客户端连接 Server 时所用的那个 IP 地址。

- 配置文件中的 `listen_port` 配置项为 COS FTP Server 的监听端口，默认为2121。`passive_port` 配置项为 COS FTP Server 的数据通道监听端口范围，默认在 [60000, 65535] 区间上选择。当客户端连接 COS FTP Server 时，要确保防火墙放行 `listen_port` 和 `passive_port` 中配置的端口。

快速实践

使用 Linux ftp 命令 访问 COS FTP Server

1. 安装 Linux ftp 客户端

```
yum install -y ftp
```

2. 在 Linux 命令行，使用命令 `ftp [ip地址] [端口号]`，连接 COS FTP Server。例如以下命令。

```
ftp 192.xxx.xx.103 2121
```

- ftp 命令中，IP 的设置对应配置示例文件 `conf/vsftpd.conf.example` 中的 `masquerade_address` 配置项。在本例中 IP 设置为 192.xxx.xx.103。
 - ftp 命令中，端口的设置对应配置示例文件 `conf/vsftpd.conf.example` 中的 `listen_port` 配置项。在本例中设置为2121。
3. 运行上述命令后，出现 **Name**和 **Password** 待输入项，输入 COS FTP Server 配置项 `ftp_login_user_name` 和 `ftp_login_user_password` 中配置的内容，即可连接成功。
- **Name**: 对应配置示例文件 `conf/vsftpd.conf.example` 中的 `ftp_login_user_name` 配置项（需要进行配置）。
 - **Password**: 对应配置示例文件 `conf/vsftpd.conf.example` 中的 `ftp_login_user_password` 配置项（需要进行配置）。

使用 FileZilla 访问 COS FTP Server

1. 下载 [FileZilla 客户端](#) 并安装。

2. 在 FileZilla 客户端配置 COS FTP Server 的访问信息后，单击**快速连接**。

- **主机 (H)**: 对应配置示例文件 `conf/vsftpd.conf.example` 中 `masquerade_address` 配置项。在本例中ip设置为192.xxx.xx.103。

⚠ 注意:

如果 COS FTP Server 处于某个网关或 NAT 后，可以通过该配置项将网关的 IP 地址或域名指定给 COS FTP Server。

- **用户名 (U)**: 对应配置示例文件 `conf/vsftpd.conf.example` 中的 `ftp_login_user_name` 配置项（需要进行配置）。
- **密码 (W)**: 对应配置示例文件 `conf/vsftpd.conf.example` 中的 `ftp_login_user_password` 配置项（需要进行配置）。
- **端口 (P)**: 对应配置示例文件 `conf/vsftpd.conf.example` 中的 `listen_port` 配置项。在本例中设置为2121。

 ftp://user0@192.103:2121 - FileZilla

文件(E) 编辑(E) 查看(V) 传输(T) 服务器(S) 书签(B) 帮助(H)



主机(H): 192.103

用户名(U):

密码(W):

端口(P): 2121

快速连接(Q)

状态: 正在连接 192.103:2121...

状态: 连接建立，等待欢迎消息...

状态: 不安全的服务器，不支持 FTP over TLS。

状态: 已登录

状态: 读取目录列表...

状态: 列出"/"的目录成功

常见问题

如您在使用 FTP Server 工具过程中，有报错或对上传限制有疑问，请参见 [FTP Server 工具](#) 常见问题。

Hadoop 工具

最近更新时间：2024-11-04 20:24:12

功能说明

Hadoop-COS 基于腾讯云对象存储（Cloud Object Storage，COS）实现了标准的 Hadoop 文件系统，可以为 Hadoop、Spark 以及 Tez 等大数据计算框架集成 COS 提供支持，使其能够跟访问 HDFS 文件系统时相同，读写存储在 COS 上的数据。

Hadoop-COS 使用 cosn 作为 URI 的 scheme，因此也称为 Hadoop-COS 为 CosN 文件系统。

使用环境

系统环境

支持 Linux、Windows 和 macOS 系统。

软件依赖

Hadoop-2.6.0及以上版本。

❗ 说明：

- 目前 Hadoop-COS 已经正式被 Apache Hadoop-3.3.0 [官方集成](#)。
- 在 Apache Hadoop-3.3.0 之前版本或 CDH 集成 Hadoop-cos jar 包后，需要重启 NodeManager 才能加载到 jar 包。
- 需要编译具体 Hadoop 版本的 jar 包时，可更改 pom 文件中 hadoop.version 进行编译。

下载与安装

获取 Hadoop-COS 分发包及其依赖

下载地址：[Hadoop-COS release](#)。

安装 Hadoop-COS 插件

- 将 `hadoop-cos-{hadoop.version}-{version}.jar` 和 `cos_api-bundle-{version}.jar` 拷贝到 `$HADOOP_HOME/share/hadoop/tools/lib` 下。

❗ 说明：

根据 Hadoop 的具体版本选择对应的 jar 包，若 release 中没有提供匹配版本的 jar 包，可自行通过修改 pom 文件中 Hadoop 版本号，重新编译生成。

- 修改 `hadoop-env.sh` 文件。进入 `$HADOOP_HOME/etc/hadoop` 目录，编辑 `hadoop-env.sh` 文件，增加以下内容，将 cosn 相关 jar 包加入 Hadoop 环境变量：

```
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
```

配置方法

Hadoop 配置

修改 `$HADOOP_HOME/etc/hadoop/core-site.xml`，增加 COS 相关用户和实现类信息，例如：

```

<configuration>
  <property>
    <name>fs.cosn.credentials.provider</name>
    <value>org.apache.hadoop.fs.auth.SimpleCredentialProvider</value>
    <description>

      This option allows the user to specify how to get the credentials.
      Comma-separated class names of credential provider classes which implement
      com.qcloud.cos.auth.COSCredentialsProvider:

      1.org.apache.hadoop.fs.auth.SessionCredentialProvider: Obtain the secret id and secret
key from the URI: cosn://secretId:secretKey@examplebucket-1250000000/;
      2.org.apache.hadoop.fs.auth.SimpleCredentialProvider: Obtain the secret id and secret
key
      from fs.cosn.userinfo.secretId and fs.cosn.userinfo.secretKey in core-site.xml;
      3.org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider: Obtain the secret
id and secret key
      from system environment variables named COS_SECRET_ID and COS_SECRET_KEY.

      If unspecified, the default order of credential providers is:
      1. org.apache.hadoop.fs.auth.SessionCredentialProvider
      2. org.apache.hadoop.fs.auth.SimpleCredentialProvider
      3. org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider
      4. org.apache.hadoop.fs.auth.SessionTokenCredentialProvider
      5. org.apache.hadoop.fs.auth.CVMInstanceCredentialsProvider
      6. org.apache.hadoop.fs.auth.CPMInstanceCredentialsProvider
      7. org.apache.hadoop.fs.auth.EMRInstanceCredentialsProvider
    </description>
  </property>

  <property>
    <name>fs.cosn.userinfo.secretId</name>
    <value>xxxxxxxxxxxxxxxxxxxxxxxx</value>
    <description>Tencent Cloud Secret Id</description>
  </property>

  <property>
    <name>fs.cosn.userinfo.secretKey</name>
    <value>xxxxxxxxxxxxxxxxxxxxxxxx</value>
    <description>Tencent Cloud Secret Key</description>
  </property>

  <property>
    <name>fs.cosn.bucket.region</name>
    <value>ap-xxx</value>
    <description>The region where the bucket is located.</description>
  </property>

  <property>
    <name>fs.cosn.bucket.endpoint_suffix</name>
    <value>cos.ap-xxx.myqcloud.com</value>
    <description>
      COS endpoint to connect to.
      For public cloud users, it is recommended not to set this option, and only the correct
area field is required.
    </description>
  </property>

```

```
</property>

<property>
  <name>fs.cosn.impl</name>
  <value>org.apache.hadoop.fs.CosFileSystem</value>
  <description>The implementation class of the CosN Filesystem.</description>
</property>

<property>
  <name>fs.AbstractFileSystem.cosn.impl</name>
  <value>org.apache.hadoop.fs.CosN</value>
  <description>The implementation class of the CosN AbstractFileSystem.</description>
</property>

<property>
  <name>fs.cosn.tmp.dir</name>
  <value>/tmp/hadoop_cos</value>
  <description>Temporary files will be placed here.</description>
</property>

<property>
  <name>fs.cosn.upload.buffer</name>
  <value>mapped_disk</value>
  <description>The type of upload buffer. Available values: non_direct_memory,
direct_memory, mapped_disk</description>
</property>

<property>
  <name>fs.cosn.upload.buffer.size</name>
  <value>134217728</value>
  <description>The total size of the upload buffer pool. -1 means unlimited.</description>
</property>

<property>
  <name>fs.cosn.upload.part.size</name>
  <value>8388608</value>
  <description>Block size to use cosn filesystem, which is the part size for
MultipartUpload.
  Considering the COS supports up to 10000 blocks, user should estimate the maximum size of
  a single file.
  For example, 8MB part size can allow writing a 78GB single file.</description>
</property>

<property>
  <name>fs.cosn.maxRetries</name>
  <value>3</value>
  <description>
    The maximum number of retries for reading or writing files to
    COS, before we signal failure to the application.
  </description>
</property>

<property>
  <name>fs.cosn.retry.interval.seconds</name>
  <value>3</value>
  <description>The number of seconds to sleep between each COS retry.</description>
```

```
</property>

<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>
</property>

<property>
  <name>fs.cosn.client-side-encryption.enabled</name>
  <value></value>
  <description>Enable or disable the client encryption function</description>
</property>

<property>
  <name>fs.cosn.client-side-encryption.public.key.path</name>
  <value>/xxx/xxx.key</value>
  <description>The direct path to the public key</description>
</property>

<property>
  <name>fs.cosn.client-side-encryption.private.key.path</name>
  <value>/xxx/xxx.key</value>
  <description>The direct path to the private key</description>
</property>

</configuration>
```

其中 fs.defaultFS 不建议在生产环境进行配置，若您需要用于部分测试场景（例如 hive-testbench 等），但要确保只配置了一个 fs.defaultFS。可添加如下配置信息：

```
<property>
  <name>fs.defaultFS</name>
  <value>cosn://examplebucket-1250000000</value>
  <description>
    This option is not advice to config, this only used for some special test cases.
  </description>
</property>
```

配置项说明

属性键	说明	默认值	必填项
fs.cosn.userinfo.secretId/secretKey	填写您账户的 API 密钥信息。可登录 访问管理控制台 查看云 API 密钥。	无	是
fs.cosn.credentials.provider	配置 SecretId 和 SecretKey 的获取方式。当前支持如下获取方式： 1. org.apache.hadoop.fs.auth.SessionCredentialProvider：从请求 URI 中获取	如果不指定该配置项，默认会按照以下顺序读取： 1. org.apache.hadoop.fs.auth.SessionCredentialPr	否

	secret id 和 secret key。其格式为： <pre>cosn://{secretId}: {secretKey}@examplebucket- 1250000000/。</pre> - org.apache.hadoop.fs.auth.SimpleCredentialProvider: 从 core-site.xml 配置文件中读取 fs.cosn.userinfo.secretId 和 fs.cosn.userinfo.secretKey 来获取 SecretId 和 SecretKey。 - org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider: 从系统环境变量 COS_SECRET_ID 和 COS_SECRET_KEY 中获取。 - org.apache.hadoop.fs.auth.SessionTokenCredentialProvider: 使用 临时密钥形式 访问。 - org.apache.hadoop.fs.auth.CVMInstanceCredentialsProvider: 利用腾讯云云服务器 (CVM) 绑定的角色, 获取访问 COS 的临时密钥。 - org.apache.hadoop.fs.auth.CPMInstanceCredentialsProvider: 利用腾讯云黑石物理机 (CPM) 绑定的角色, 获取访问 COS 的临时密钥。 - org.apache.hadoop.fs.auth.EMRInstanceCredentialsProvider: 利用腾讯云 EMR 服务绑定的角色, 获取访问 COS 的临时密钥, 适用于 EMR 集群内节点访问对象存储场景。 - org.apache.hadoop.fs.auth.RangerCredentialsProvider 使用 ranger 进行获取密钥。	provider - org.apache.hadoop.fs.auth.SimpleCredentialProvider - org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider - org.apache.hadoop.fs.auth.SessionTokenCredentialProvider - org.apache.hadoop.fs.auth.CVMInstanceCredentialsProvider - org.apache.hadoop.fs.auth.CPMInstanceCredentialsProvider - org.apache.hadoop.fs.auth.EMRInstanceCredentialsProvider	
fs.cosn.useHttps	配置是否使用 HTTPS 作为与 COS 后端的传输协议。	true	否
fs.cosn.impl	cosn 对 FileSystem 的实现类, 固定为 org.apache.hadoop.fs.CosFileSystem。	无	是
fs.AbstractFileSystem.cosn.impl	cosn 对 AbstractFileSystem 的实现类, 固定为 org.apache.hadoop.fs.CosN。	无	是
fs.cosn.bucket.region	请填写待访问存储桶的地域信息, 枚举值请参见 地域和访问域名 中的地域简称, 例如: ap-beijing、ap-guangzhou 等。兼容原有配置: fs.cosn.userinfo.region。	无	是
fs.cosn.bucket.endpoint_suffix	指定要连接的 COS endpoint, 该项为非必填项目。对于公有云 COS 用户而言, 只需要正确填写上述的 region 配置即可。兼容原有配置: fs.cosn.userinfo.endpoint_suffix。配置该项时请删除 fs.cosn.bucket.region 配置项 endpoint 才能生效。	无	否
fs.cosn.tmp.dir	请设置一个实际存在的本地目录, 运行过程中产生的临时文件会暂时放于此处。	/tmp/hadoop_cos	否

fs.cosn.upload.part.size	CosN 文件系统每个 block 的大小，也是分块上传的每个 part size 的大小。由于 COS 的分块上传最多只能支持10000块，因此需要预估最大可能使用到的单文件大小。 例如，part size 为8MB时，最大能够支持78GB的单文件上传。part size 最大可以支持到2GB，即单文件最大可支持19TB。	8388608（8MB）	否
fs.cosn.upload.buffer	CosN 文件系统上传时依赖的缓冲区类型。当前支持三种类型的缓冲区：非直接内存缓冲区（non_direct_memory），直接内存缓冲区（direct_memory），磁盘映射缓冲区（mapped_disk）。非直接内存缓冲区使用的是 JVM 堆内存，直接内存缓冲区使用的是堆外内存，而磁盘映射缓冲区则是基于内存文件映射得到的缓冲区。	mapped_disk	否
fs.cosn.upload.buffer.size	CosN 文件系统上传时依赖的缓冲区大小，如果指定为-1，则表示不限制缓冲区。若不限制缓冲区大小，则缓冲区的类型必须为 mapped_disk。如果指定大小大于0，则要求该值至少大于等于一个 block 的大小。兼容原有配置 fs.cosn.buffer.size。	-1	否
fs.cosn.block.size	CosN 文件系统 block size。	134217728（128MB）	否
fs.cosn.upload_thread_pool	文件流式上传到 COS 时，并发上传的线程数目。	10	否
fs.cosn.copy_thread_pool	目录拷贝操作时，可用于并发拷贝和删除文件的线程数目。	3	否
fs.cosn.read.ahead.block.size	预读块的大小。	1048576（1MB）	否
fs.cosn.read.ahead.queue.size	预读队列的长度。	8	否
fs.cosn.maxRetries	访问 COS 出现错误时，最多重试的次数。	200	否
fs.cosn.retry.interval.seconds	每次重试的时间间隔。	3	否
fs.cosn.server-side-encryption.algorithm	配置 COS 服务端加密算法，支持 SSE-C 和 SSE-COS，默认为空，不加密。	无	否
fs.cosn.server-side-encryption.key	当开启 COS 的 SSE-C 服务端加密算法时，必须配置 SSE-C 的密钥，密钥格式为 base64 编码的 AES-256 密钥，默认为空，不加密。	无	否
fs.cosn.client-side-encryption.enabled	是否开启客户端加密，默认不开启。开启后必须配置客户端加密的公钥和私钥。此时无法使用 append、truncate 接口。	false	否
fs.cosn.client-side-encryption.public.key.path	客户端加密公钥文件的绝对路径	无	否
fs.cosn.client-side-encryption.private.key.path	客户端加密私钥文件的绝对路径	无	否
fs.cosn.crc64.checksum.enabled	是否开启 CRC64 校验。默认不开启，此时无法使用 hadoop fs -checksum 命令获取文件的	false	否

	CRC64 校验值。		
fs.cosn.crc32c.checksum.enabled	是否开启 CRC32C 校验。默认不开启，此时无法使用 <code>hadoop fs -checksum</code> 命令获取文件的 CRC32C 校验值，只能开启一种校验方式： <code>crc32c</code> 或 <code>crc64</code> 。	false	否
fs.cosn.traffic.limit	上传带宽的控制选项，819200 – 838860800 bits/s，默认值为-1，默认表示不限制。	无	否

Bucket 独立配置项

背景：跨地域访问不同的 bucket，每个 bucket 的配置项不同，为此支持了一次配置，访问多个 bucket 的需求。

核心：单独设置 bucket 维度的配置优先使用独立配置项，如果没有设置独立配置项则使用原配置，如果原配置没有设置则使用默认配置。

下面示例是给 examplebucket-1250000000 单独设置 fs.cosn.upload.buffer 的配置项：

```
fs.cosn.bucket.examplebucket-1250000000.upload.buffer ***
```

说明：

这里 fs.cosn.upload.buffer 的 subkey (trim fs.cosn.) 是 upload.buffer，则 bucket 的独立配置项是

```
fs.cosn.bucket.<bucket-appid>.<subkey>。
```

使用方式：

```
hadoop fs -ls cosn://examplebucket-1250000000/
```

简化 Bucket 独立配置项（适用于普通存储桶）

通过 `cosn://<bucket>/` 方式进行访问。

下面示例是给 examplebucket-1250000000 单独设置 fs.cosn.upload.buffer 的配置项：

```
fs.cosn.userinfo.appid 1250000000
fs.cosn.bucket.examplebucket.upload.buffer ***
```

说明：

这里 fs.cosn.upload.buffer 的 subkey (trim fs.cosn.) 是 upload.buffer，但由于已经设置了 appid，则 bucket 的独立配置项可以简化为 `fs.cosn.bucket.<bucket>.<subkey>`。

使用方式：

```
hadoop fs -ls cosn://examplebucket/
```

简化 Bucket 独立配置项（适用于元数据加速存储桶）

通过 `cosn://<bucket>/` 方式进行访问。

下面示例是给 examplebucket-1250000000 单独设置 fs.cosn.upload.buffer 的配置项：

```
fs.cosn.userinfo.appid 1250000000
fs.cosn.trsf.fs ofs.use.short.bucketname true
fs.cosn.bucket.examplebucket.upload.buffer ***
```

说明：

这里 `fs.cosn.upload.buffer` 的 `subkey` (`trim fs.cosn.`) 是 `upload.buffer`，但由于已经设置了 `appid`，则 `bucket` 的独立配置项可以简化为 `fs.cosn.bucket.<bucket>.<subkey>`。

由于旧版本 OFS Java SDK 对挂载点格式进行了访问限制，只支持 `<bucket-appid>` 方式访问，所以需要更新依赖插件至最新版本才可使用。依赖插件版本信息：

```
ofs java sdk >= 1.1.8
hadoop cos >= 8.3.0
```

使用方式：

```
hadoop fs -ls cosn://examplebucket/
```

服务端加密

Hadoop-COS 支持服务端加密，目前提供两种加密方式：COS 托管密钥方式（SSE-COS）和用户自定义密钥方式（SSE-C），Hadoop-COS 的加密功能默认为关闭状态，用户可以选择开启，通过以下方式进行配置。

SSE-COS 加密

SSE-COS 加密即 COS 托管密钥的服务端加密，由腾讯云 COS 托管主密钥和管理数据。当使用 Hadoop-COS 时，用户可以在 `$HADOOP_HOME/etc/hadoop/core-site.xml` 文件中，增加以下配置来进行实现 SSE-COS 加密。

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

SSE-C 加密

SSE-C 加密即用户自定义密钥的服务端加密。加密密钥由用户自己提供，用户在上传对象时，COS 将使用用户提供的加密密钥对用户的数据进行 AES-256 加密。当使用 Hadoop-COS 时，用户可以在 `$HADOOP_HOME/etc/hadoop/core-site.xml` 文件中，增加以下配置来进行实现 SSE-C 加密。

```
<property>
  <name>fs.cosn.server-side-encryption.algorithm</name>
  <value>SSE-C</value>
  <description>The server side encryption algorithm.</description>
</property>
<property>
  <name>fs.cosn.server-side-encryption.key</name>
  <value>MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3OD1BQkNERUY=</value> #用户需要自行配置 SSE-C 的密钥，密
  钥格式为 base64 编码的 AES-256 密钥。
  <description>The SSE-C server side encryption key.</description>
</property>
```

注意：

- Hadoop-COS 的 SSE-C 服务端加密依赖于 COS 的 SSE-C 服务端加密。因此，Hadoop-COS 不存储用户提供的加密密钥。同时需要值得注意的是，COS 的 SSE-C 服务端加密方式不存储用户提供的加密密钥，而是存储加密密钥添加了随机数据的 HMAC 值，该值用于验证用户访问对象的请求。COS 无法使用随机数据的 HMAC 值来推导出加密密钥的值或解密加密对象的内容。因此，如果用户丢失了加密密钥，则无法再次获取到该对象。

- Hadoop-COS 配置了 SSE-C 服务端加密算法时，必须在 `fs.cosn.server-side-encryption.key` 配置项中配置 SSE-C 的密钥，密钥格式为 base64 编码的 AES-256 密钥。
- Hadoop-COS 在 v5.10.0 针对非写请求会报异常 “Set SSE_COS request no such method”。

客户端加密

COSN 客户端加密采用 RSA 加密方式，密钥分为公钥和私钥，其中公钥用于文件加密过程，私钥用于文件解密过程。在上传文件时，COSN 会生成一个随机密钥，并用该密钥对文件进行对称加密。公钥会对该密钥进行加密，并将加密后的信息保存在文件元数据中。在下载文件时，COSN 会使用私钥从文件元数据中得到加密随机密钥进行解密，再使用解密后的随机密钥对文件进行对此解密。公钥和私钥只参与客户端本地计算，不会在网络上进行传输或保存在服务端，以保证主密钥的数据安全。

- 使用客户端加密功能时，您需要对主密钥的完整性和正确性负责。在对加密数据进行复制或者迁移时，您需要对加密元信息的完整性和正确性负责。因您维护不当导致主密钥用错或丢失，加密元信息出错或丢失，从而导致加密数据无法解密所引起的一切损失和后果均由您自行承担。
- 开启客户端加密后，不再支持 `append`、`truncate` 接口。
- 使用关闭了客户端加密功能的客户端对加密文件进行 `hadoop fs -cp` 命令，会丢失加密信息。
- 开启客户端加密后，默认关闭 CRC 文件校验，默认关闭异步文件分块上传。

当使用 Hadoop-COS 时，用户可以在 `$HADOOP_HOME/etc/hadoop/core-site.xml` 文件中，增加以下配置来进行实现 SSE-COS 加密。

```
<property>
  <name>fs.cosn.client-side-encryption.enabled</name>
  <value>true</value>
  <description>Enable or disable the client encryption function</description>
</property>

<property>
  <name>fs.cosn.client-side-encryption.public.key.path</name>
  <value>/xxx/xxx.key</value>
  <description>The direct path to the public key</description>
</property>

<property>
  <name>fs.cosn.client-side-encryption.private.key.path</name>
  <value>/xxx/xxx.key</value>
  <description>The direct path to the private key</description>
</property>
```

可使用以下代码生成密钥：

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;

// 使用非对称密钥 RSA 加密每次生成的随机对称密钥
public class BuildKey {
    private static final SecureRandom srand = new SecureRandom();
    private static void buildAndSaveAsymKeyPair(String pubKeyPath, String priKeyPath) throws
        IOException, NoSuchAlgorithmException {
```

```

    KeyPairGenerator keyGenerator = KeyPairGenerator.getInstance("RSA");
    keyGenerator.initialize(1024, srand);
    KeyPair keyPair = keyGenerator.generateKeyPair();
    PrivateKey privateKey = keyPair.getPrivate();
    PublicKey publicKey = keyPair.getPublic();

    X509EncodedKeySpec x509EncodedKeySpec = new X509EncodedKeySpec(publicKey.getEncoded());
    FileOutputStream fos = new FileOutputStream(pubKeyPath);
    fos.write(x509EncodedKeySpec.getEncoded());
    fos.close();

    PKCS8EncodedKeySpec pkcs8EncodedKeySpec = new
    PKCS8EncodedKeySpec(privateKey.getEncoded());
    fos = new FileOutputStream(priKeyPath);
    fos.write(pkcs8EncodedKeySpec.getEncoded());
    fos.close();
}

public static void main(String[] args) throws Exception {

    String pubKeyPath = "pub.key";
    String priKeyPath = "pri.key";
    buildAndSaveAsymKeyPair(pubKeyPath, priKeyPath);
}
}

```

设置 yarn-site.xml 的 classpath

yarn-site.xml 添加

```

<property>
  <name>yarn.application.classpath</name>
  <value>$HADOOP_CLASSPATH</value>
</property>

```

使用方法

使用示例

命令格式为 `hadoop fs -ls -R cosn://<BucketName-APPID>/<路径>`，或 `hadoop fs -ls -R /<路径>`（需要配置 `fs.defaultFS` 选项为 `cosn://BucketName-APPID`），下例中以名称为 `examplebucket-1250000000` 的 bucket 为例，可在其后面加上具体路径。

```

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

```

```
-rw-rw-rw-  1 root root      2386 2018-06-11 07:29 cosn://examplebucket-1250000000/hdfs/test/ReadMe
```

运行 MapReduce 自带的 wordcount，执行以下命令。

注意：

以下命令中 `hadoop-mapreduce-examples-2.7.2.jar` 是以 2.7.2 版本为例，若版本不同，请修改成对应的版本号。

```
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount cosn://example/mr/input cosn://example/mr/output3
```

执行成功会返回统计信息，示例如下：

```
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
COSN: Number of write operations=0
FILE: Number of bytes read=547350
FILE: Number of bytes written=1155616
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=0
HDFS: Number of bytes written=0
HDFS: Number of read operations=0
HDFS: Number of large read operations=0
HDFS: Number of write operations=0
Map-Reduce Framework
Map input records=5
Map output records=7
Map output bytes=59
Map output materialized bytes=70
Input split bytes=99
Combine input records=7
Combine output records=6
Reduce input groups=6
Reduce shuffle bytes=70
Reduce input records=6
Reduce output records=6
Spilled Records=12
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=0
Total committed heap usage (bytes)=653262848
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
```

```
File Input Format Counters
Bytes Read=36
File Output Format Counters
Bytes Written=40
```

通过 Java 代码访问 COSN

```
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 的配置项可参见
https://cloud.tencent.com/document/product/436/6884#hadoop-.E9.85.8D.E7.BD.AE
        // 以下配置是必填项
        conf.set("fs.cosn.impl", "org.apache.hadoop.fs.CosFileSystem");
        conf.set("fs.AbstractFileSystem.cosn.impl", "org.apache.hadoop.fs.CosN");
        conf.set("fs.cosn.tmp.dir", "/tmp/hadoop_cos");
        conf.set("fs.cosn.bucket.region", "ap-guangzhou");
        conf.set("fs.cosn.userinfo.secretId", "AKXXXXXXXXXXXXXXXXXX");
        conf.set("fs.cosn.userinfo.secretKey", "XXXXXXXXXXXXXXXXXX");
        conf.set("fs.cosn.userinfo.appid", "XXXXXXXXXXXX");
        // 其他配置参考官网文档
https://cloud.tencent.com/document/product/436/6884#hadoop-.E9.85.8D.E7.BD.AE
        // 是否开启 CRC64 校验。默认不开启，此时无法使用 hadoop fs -checksum 命令获取文件的 CRC64 校验值
        conf.set("fs.cosn.crc64.checksum.enabled", "true");
        String cosnUrl = "cosn://f4mxxxxxxxx-125xxxxxxxx";
        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 {
        // 创建一个文件（如果存在则将其覆盖）
        // if the parent dir does not exist, fs will create it!
        FSDataOutputStream out = fs.create(filePath, true);
        try {
            // 写入一个文件
            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 IOException {
    FileStatus fileStatus = fs.getFileStatus(path);
    if (fileStatus.isDirectory()) {
        System.out.printf("path %s is dir\n", path);
        return;
    }
    long fileLen = fileStatus.getLen();
    long accessTime = fileStatus.getAccessTime();
    long modifyTime = fileStatus.getModificationTime();
    String owner = fileStatus.getOwner();
    String group = fileStatus.getGroup();

    System.out.printf("path %s is file, fileLen: %d, accessTime: %d, modifyTime: %d, owner: %s, group: %s\n",
        path, fileLen, accessTime, modifyTime, owner, group);
}

private static void getFileChecksum(FileSystem fs, Path path) throws IOException {
    FileChecksum checksum = fs.getFileChecksum(path);
    System.out.printf("path %s, checksumType: %s, checksumCrcVal: %d\n",
        path, checksum.getAlgorithmName(), ByteBuffer.wrap(checksum.getBytes()).getInt());
}

private static void copyFileFromLocal(FileSystem fs, Path cosnPath, Path localPath) throws
IOException {
    fs.copyFromLocalFile(localPath, cosnPath);
}

private static void copyFileToLocal(FileSystem fs, Path cosnPath, Path localPath) throws
IOException {
    fs.copyToLocalFile(cosnPath, localPath);
}

private static void renamePath(FileSystem fs, Path oldPath, Path newPath) throws IOException {
    fs.rename(oldPath, newPath);
}

private static void listDirPath(FileSystem fs, Path dirPath) throws IOException {
    FileStatus[] dirMemberArray = fs.listStatus(dirPath);
}
```



```
        for (FileStatus dirMember : dirMemberArray) {
            System.out.printf("dirMember path %s, fileLen: %d\n", dirMember.getPath(),
dirMember.getLen());
        }
    }

    // 递归删除标志用于删除目录
    // 如果递归为 false 并且 dir 不为空, 则操作将失败
    private static void deleteFileOrDir(FileSystem fs, Path path, boolean recursive) throws
IOException {
        fs.delete(path, recursive);
    }

    private static void closeFileSystem(FileSystem fs) throws IOException {
        fs.close();
    }

    public static void main(String[] args) throws IOException {
        // 初始化文件
        FileSystem fs = initFS();

        // 创建文件
        Path cosnFilePath = new Path("/folder/exampleobject.txt");
        createFile(fs, cosnFilePath);

        // 读取文件
        readFile(fs, cosnFilePath);

        // 查询文件或目录
        queryFileOrDirStatus(fs, cosnFilePath);

        // 获取文件校验和
        getFileChecksum(fs, cosnFilePath);

        // 从本地复制文件
        Path localFilePath = new Path("file:///home/hadoop/ofs_demo/data/exampleobject.txt");
        copyFileFromLocal(fs, cosnFilePath, localFilePath);

        // 获取文件到本地
        Path localDownFilePath = new Path("file:///home/hadoop/ofs_demo/data/exampleobject.txt");
        copyFileToLocal(fs, cosnFilePath, localDownFilePath);

        listDirPath(fs, cosnFilePath);
        // 重命名
        mkdir(fs, new Path("/doc"));
        Path newPath = new Path("/doc/example.txt");
        renamePath(fs, cosnFilePath, newPath);

        // 删除文件
        deleteFileOrDir(fs, newPath, false);

        // 创建目录
        Path dirPath = new Path("/folder");
        mkdir(fs, dirPath);
    }
}
```

```
// 在目录中创建文件
Path subFilePath = new Path("/folder/exampleobject.txt");
createFile(fs, subFilePath);

// 列出目录
listDirPath(fs, dirPath);

// 删除目录
deleteFileOrDir(fs, dirPath, true);
deleteFileOrDir(fs, new Path("/doc"), true);

// 关闭文件系统
closeFileSystem(fs);
}
}
```

常见问题

如果您在使用 Hadoop 工具过程中，有相关的疑问，请参见 [Hadoop 工具类常见问题](#)。

COSDistCp 工具

最近更新时间：2025-05-16 17:53:22

功能说明

COSDistCp 是一款基于 MapReduce 的分布式文件拷贝工具，主要用于 HDFS 和 COS 之间的数据拷贝，它主要具有以下功能点：

- 根据长度、CRC 校验和，进行文件的增量迁移、数据校验。
- 对源目录中的文件进行正则表达式过滤。
- 对源目录中的文件进行解压缩，并转换为预期的压缩格式。
- 基于正则表达式，对文本文件进行聚合。
- 保留源文件和源目录的用户、组、扩展属性和时间。
- 配置告警和 Prometheus 监控。
- 统计文件大小分布。
- 对读取带宽进行限速。

原理说明

COSDistCp 基于 MapReduce 框架实现，为多进程+多线程的架构，可以对文件进行拷贝、数据校验、压缩、文件属性保留以及拷贝重试等工作。COSDistCp 默认会覆盖目标端已经存在的同名文件，当文件迁移或校验失败的时候，对应的文件会拷贝失败，并会在临时目录下记录迁移失败的文件信息。当您的源目录有文件新增或文件内容发生变化时，您可通过 skipMode 或 diffMode 模式，通过对比文件的长度或 CRC 校验值，进行数据校验和文件的增量迁移。

使用环境

系统环境

支持 Linux 系统。

软件依赖

Hadoop-2.6.0及以上版本、Hadoop-COS 插件5.9.3及以上版本。

下载与安装

获取 COSDistCp jar 包

- Hadoop 2.x用户可下载 [cos-distcp-1.13.16-2.8.5.jar](#)，根据 jar 包的 [MD5校验值](#) 确认下载的 jar 包是否完整。
- Hadoop 3.x用户可下载 [cos-distcp-1.13.16-3.1.0.jar](#)，根据 jar 包的 [MD5校验值](#) 确认下载的 jar 包是否完整。

安装说明

在 Hadoop 环境下，安装 [Hadoop-COS](#) 后，即可直接运行 COSDistCp 工具。

⚠ 注意：

- Hadoop-COS jar 和 cos_api-bundle jar 两个 jar 包必须放置在运行命令 `hadoop classpath` 指定的其中一个路径下，如/usr/local/services/hadoop/share/hadoop/tools/lib/。路径下目录名称（例如 .../services/... 或 .../service/... ）以客户端默认配置为准。
- 若 COS 存储桶开启了严格签名模式，请下载 Hadoop-COS v8.3.22、cos_api-bundle.jar v5.6.244.4或更新的版本。

```
hadoop jar cos-distcp-${version}.jar \  
-libjars /usr/local/services/hadoop/share/hadoop/tools/lib/cos_api-  
bundle-${version}.jar,/usr/local/services/hadoop/share/hadoop/tools/lib/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.CosFileSystem \  
-Dfs.AbstractFileSystem.cosn.impl=org.apache.hadoop.fs.CosN \  
--src /data/warehouse \  
--dest cosn://examplebucket-1250000000/warehouse
```

参数说明

您可在 Hadoop 用户下，使用命令 `hadoop jar cos-distcp-${version}.jar --help` 查看 COSDistCp 支持的参数选项，其中 `${version}` 为版本号。

以下为当前版本 COSDistCp 的参数说明，`fs.xxx` 开头等参数为 Hadoop 工具参数，参数说明请参见 [Hadoop 工具 > 配置项说明](#)。

属性键	说明	默认值	是否必填
<code>--help</code>	输出 COSDistCp 支持的参数选项 示例： <code>--help</code>	无	否
<code>--src = LOCATION</code>	指定拷贝的源目录，可以是 HDFS 或者 COS 路径 示例： <code>--src = hdfs://user/logs/</code>	无	是
<code>--dest = LOCATION</code>	指定拷贝的目标目录，可以是 HDFS 或者 COS 路径 示例： <code>--dest = cosn://examplebucket-1250000000/user/logs</code>	无	是
<code>--srcPattern = PATTERN</code>	指定正则表达式对源目录中的文件进行过滤 示例： <code>--srcPattern='.*\.log\$'</code> 注意： 您需要将参数使用单引号包围，以避免符号 <code>*</code> 被 shell 解释。	无	否
<code>--taskNumber = VALUE</code>	指定拷贝进程数，示例： <code>--taskNumber = 10</code>	10	否
<code>--workerNumber = VALUE</code>	指定拷贝线程数，COSDistCp 在每个拷贝进程中创建该参数大小的拷贝线程池 示例： <code>--workerNumber = 4</code>	4	否
<code>--filesPerMapper = VALUE</code>	指定每个 Mapper 输入文件的行数 示例： <code>--filesPerMapper = 10000</code>	500000	否
<code>--groupBy = PATTERN</code>	指定正则表达式对文本文件进行聚合 示例： <code>--groupBy = '.*group-input/(\d+)-(\d+).*'</code>	无	否
<code>--targetSize = VALUE</code>	指定目标文件的大小，单位：MB，与 <code>--groupBy</code> 一起使用 示例： <code>--targetSize = 10</code>	无	否

--outputCodec = VALUE	指定输出文件的压缩方式，可选 gzip、lzo、snappy、none 和 keep，其中： - keep 保持原有文件的压缩方式 - none 则根据文件后缀对文件进行解压， - 示例：--outputCodec = gzip 注意： 如果存在文件 /dir/test.gzip 和 /dir/test.gz，指定输出格式为 lzo，最终只会保留一个文件 /dir/test.lzo。	keep	否
--multipartUploadChunkSize = VALUE	指定 Hadoop-COS 插件传输文件到 COS 时分块的大小，COS 支持的最大分块数为 10000，您可根据文件大小，调整分块大小，单位：MB，默认为 8MB 示例：--multipartUploadChunkSize = 20	8MB	否
--cosServerSideEncryption	指定文件上传到 COS 时，使用 SSE-COS 作为加解密算法 示例：--cosServerSideEncryption	false	否
--outputManifest = VALUE	指定拷贝完成的时候，在目标目录下生成本次拷贝的目标文件信息列表（GZIP 压缩） 示例：--outputManifest = manifest.gz	无	否
--requirePreviousManifest	要求指定 --previousManifest = VALUE 参数，以进行增量拷贝 示例：--requirePreviousManifest	false	否
--previousManifest = LOCATION	前一次拷贝生成的目标文件信息 示例：--previousManifest = cosn://examplebucket-12500000000/big-data/manifest.gz	无	否
--copyFromManifest	和 --previousManifest = LOCATION 一起使用，可将 --previousManifest 中的文件，拷贝到目标文件系统 示例：--copyFromManifest	false	否
--storageClass = VALUE	指定对象存储类型，可选值为 STANDARD、STANDARD_IA、ARCHIVE、DEEP_ARCHIVE、INTELLIGENT_TIERING，关于更多支持的存储类型和介绍，请参见 存储类型概述	无	否
--srcPrefixesFile = LOCATION	指定本地文件，该文件中每行包含一个需要拷贝的源目录 示例：--srcPrefixesFile = file:///data/migrate-folders.txt	无	否
--skipMode = MODE	拷贝文件前，校验源文件和目标文件是否相同，相同则跳过，可选 none（不校验）、length（长度）、checksum（CRC 值）、length-mtime（长度+mtime 值）和 length-checksum（长度 + CRC 值） 示例：--skipMode = length	length-checksum	否
--checkMode = MODE	当文件拷贝完成的时候，校验源文件和目标文件是否相同，可选 none（不校验）、length（长度）、checksum（CRC 值）、length-mtime（长度+mtime 值）和 length-checksum（长度 + CRC 值） 示例：--checkMode = length-checksum	length-checksum	否
--diffMode = MODE	指定获取源和目的目录的差异文件列表，可选 length（长度）、checksum（CRC 值）、length-mtime（长度+mtime 值）和 length-checksum（长度 + CRC 值） 示例：--diffMode = length-checksum	无	否
--diffOutput = LOCATION	指定 diffMode 的 HDFS 输出目录，该输出目录必须为空 示例：--diffOutput = /diff-output	无	否

<code>--cosChecksumType = TYPE</code>	指定 Hadoop-COS 插件使用的 CRC 算法，可选值为 CRC32C 和 CRC64 示例： <code>--cosChecksumType = CRC32C</code>	CRC32C	否
<code>--preserveStatus = VALUE</code>	指定是否将源文件的 user、group、permission、xattr 和 timestamps 元信息拷贝到目标文件，可选值为 ugpxt（即为 user、group、permission、xattr 和 timestamps 的英文首字母） 示例： <code>--preserveStatus = ugpt</code>	无	否
<code>--ignoreSrcMiss</code>	忽略存在于文件清单中，但拷贝时不存在的文件	false	否
<code>--promGatewayAddress = VALUE</code>	指定 MapReduce 任务运行的 Counter 数据推送到的 Prometheus PushGateway 的地址和端口	无	否
<code>--promGatewayDeleteOnFinish = VALUE</code>	指定任务完成时，删除 Prometheus PushGateway 中 JobName 的指标集合 示例： <code>--promGatewayDeleteOnFinish = true</code>	true	否
<code>--promGatewayJobName = VALUE</code>	指定上报给 Prometheus PushGateway 的 JobName 示例： <code>--promGatewayJobName = cos-distcp-hive-backup</code>	无	否
<code>--promCollectInterval = VALUE</code>	指定收集 MapReduce 任务 Counter 信息的间隔，单位：ms 示例： <code>--promCollectInterval = 5000</code>	5000	否
<code>--promPort = VALUE</code>	指定将 Prometheus 指标暴露给外部的 Server 端口 示例： <code>--promPort = 9028</code>	无	否
<code>--enableDynamicStrategy</code>	指定开启任务动态分配策略，使迁移速度快的任务迁移更多的文件。 示例： <code>--enableDynamicStrategy</code> ⚠ 注意： 该模式存在一定局限性，例如任务计数器在进程异常的情况下计数不准确，请迁移完成后用 <code>--diffMode</code> 进行数据校验。	false	否
<code>--splitRatio = VALUE</code>	指定 Dynamic Strategy 的切分比例，splitRatio 值越大，则任务粒度越小 示例： <code>--splitRatio = 8</code>	8	否
<code>--localTemp = VALUE</code>	指定 Dynamic Strategy 生成的任务信息文件所在的本地文件夹 示例： <code>--localTemp = /tmp</code>	/tmp	否
<code>--taskFilesCopyThreadNum = VALUE</code>	指定 Dynamic Strategy 任务信息文件拷贝到 HDFS 上的并发度 示例： <code>--taskFilesCopyThreadNum = 32</code>	32	否
<code>--statsRange = VALUE</code>	指定统计的区间范围 示例： <code>---statsRange = 0,1mb,10mb,100mb,1gb,10gb,inf</code>	0,1mb,10mb,100mb,1gb,10gb,inf	否
<code>--printStatsOnly</code>	只统计待迁移文件大小的分布信息，不迁移数据 示例： <code>--printStatsOnly</code>	无	否
<code>--bandWidth</code>	限制读取每个迁移文件的带宽，单位为：MB/s，默认-1，不限制读取带宽。 示例： <code>--bandWidth = 10</code> ⚠ 注意：	无	否

	迁移的总带宽限制计算公式： $\text{taskNumber} * \text{workerNumber} * \text{bandWidth}$ ，可以设置 $\text{workerNumber}=1$ ，通过 taskNumber 来控制迁移并发数，通过 bandWidth 来控制单个并发的带宽。		
<code>--jobName</code>	指定迁移任务的名称。 示例： <code>--jobName = cosdistcp-to-warehouse</code>	无	否
<code>--compareWithCompatibleSuffix</code>	使用 <code>--skipMode</code> 和 <code>--diffMode</code> 参数时，是否将源文件的后缀 <code>gzip</code> 转换为 <code>gz</code> ， <code>lzop</code> 文件后缀转换为 <code>lzo</code> ，进行判断。 示例： <code>--compareWithCompatibleSuffix</code>	无	否
<code>--delete</code>	保证源目录和目标目录文件的一致性，将源目录中没有而目标目录中有的文件，移动到独立的 <code>trash</code> 目录下，同时生成文件清单。  注意： 不能同时使用 <code>--diffMode</code> 参数。	无	否
<code>--deleteOutput</code>	指定 <code>delete</code> 的 HDFS 输出目录，该目录必须为空 示例： <code>--deleteOutput = /dele-output</code>	无	否
<code>--listThreadNum</code>	并发 <code>list</code> 源目录的线程数，在对象存储产品之间的迁移上有比较好的性能优化。 示例： <code>--listThreadNum = 50</code>	无	否
<code>--createScheduleFile</code>	是否创建定时打点文件，用于统计迁移状态	false	否

使用示例

查看 help 选项

以参数 `--help` 执行命令，查看 COSDistCp 支持的参数，示例如下：

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

以上命令中，`${version}` 为 COSDistCp 版本号，例如1.0版本的 COSDistCp jar 包名为 `cos-distcp-1.0.jar`。

统计待迁移的文件大小分布信息

以参数 `--printStatsOnly` 和 `--statsRange=VALUE` 执行命令，输出待迁移文件的大小分布信息：

! 说明：

执行命令前需要 `/tmp` 作为临时文件存储目录，执行 `hadoop fs -mkdir hdfs:///tmp` 创建 `tmp` 目录。

```
hadoop jar cos-distcp-${version}.jar --src /wookie/data --dest cosn://examplebucket-1250000000/wookie/data --printStatsOnly --statsRange=0,1mb,10mb,100mb,1gb,10gb,inf
```

Copy File Distribution Statistics:

Total File Count: 4

Total File Size: 1190133760

SizeRange	TotalCount	TotalSize	
0MB ~ 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_MAX | 0(0.00%) | 0(0.00%) |
```

指定待迁移文件的源目录和目标目录

以参数 `--src` 和 `--dest` 执行命令，示例如下：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-1250000000/data/warehouse
```

COSDistCp 默认会对拷贝失败的文件重试5次，如果仍然失败，则会将失败文件信息写入 `/tmp/${randomUUID}/output/failed/` 目录下，其中，`${randomUUID}` 为随机字符串。记录失败文件信息后，COSDistCp 会继续迁移剩余文件，迁移任务并不会因为部分文件迁移失败而失败。在迁移任务完成的时候，COSDistCp 会输出计数器信息，并判断是否存在文件迁移失败，如果存在，则在提交任务的客户端抛出异常。

说明：

- `/tmp/${randomUUID}/output/failed/` 为 hadoop 集群 defaultfs 的 tmp 目录，不是本地目录。
- 请确保您的任务提交机器，配置了 MapReduce 任务的提交端 INFO 日志输出，否则将无法输出计数器信息。以腾讯云 EMR 为例，配置方法如下：通过控制台调整配置项 `yarn.log-aggregation-enable` 设置为 true，下发配置后，重启 yarn 服务即可生效。

以下类型的源文件信息包含在输出文件中：

- 存在源文件的清单中，但拷贝时源文件不存在，记录为 SRC_MISS。
- 其他原因导致的拷贝失败，统一记录为 COPY_FAILED。

您可以重新运行拷贝命令实现增量迁移，通过如下命令，获取 MapReduce 任务的日志，确定文件拷贝失败的原因，其中 `application_1610615435237_0021` 为应用 ID：

```
yarn logs -applicationId application_1610615435237_0021 > application_1610615435237_0021.log
```

查看 Counters

在拷贝任务结束时，会输出文件拷贝的统计信息（注意该统计数据是单个 mr 任务维度的，若想要全局的统计信息，请使用 `--createScheduleFile`），相关计数器如下：

```
CosDistCp Counters
  BYTES_EXPECTED=10198247
  BYTES_SKIPPED=10196880
  FILES_COPIED=1
  FILES_EXPECTED=7
  FILES_FAILED=1
  FILES_SKIPPED=5
```

文件拷贝的统计信息说明如下：

统计项	说明
BYTES_EXPECTED	根据源目录统计的需拷贝的文件总大小，单位：字节
FILES_EXPECTED	根据源目录统计的需拷贝文件数，包含目录文件
BYTES_SKIPPED	长度或校验和值相等，不拷贝的文件总大小，单位：字节
FILES_SKIPPED	长度或校验和值相等，不拷贝的源文件数
FILES_COPIED	拷贝成功的源文件数

FILES_FAILED	拷贝失败的源文件数
FOLDERS_COPIED	拷贝成功的目录数
FOLDERS_SKIPPED	跳过的目录数

指定拷贝进程数以及每个拷贝进程内的拷贝线程数

以参数 `--taskNumber` 和 `--workersNumber` 执行命令，COSDistCp 采用多进程+多线程的拷贝架构，您可以：

- 通过 `--taskNumber` 指定拷贝进程数目
- 通过 `--workerNumber` 指定每个拷贝进程内的拷贝线程数

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/ --dest cosn://examplebucket-1250000000/data/warehouse --taskNumber=10 --workerNumber=5
```

跳过具有相同校验值文件，进行增量迁移

以参数 `--skipMode` 执行命令。跳过源和目标具有相同长度和校验和文件的拷贝，默认值 `length-checksum`：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-1250000000/data/warehouse --skipMode=length-checksum
```

`--skipMode` 选项用于在拷贝文件前，校验源文件和目标文件是否相同，若相同则跳过，可选 `none`（不校验）、`length`（长度）、`checksum`（CRC 值）和 `length-checksum`（长度 + CRC 值）。

如果源和目标文件系统的校验和算法不同，则会读取源端文件计算新的校验和。如果您的源是 HDFS，您可以通过如下方式，确定 HDFS 源是否支持 COMPOSITE-CRC32C 校验算法：

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

迁移完成后的数据校验及增量迁移

以参数 `--diffMode` 和 `--diffOutput` 执行命令：

- `--diffMode` 可选值为 `length` 和 `length-checksum`。
 - `--diffMode=length` 表示根据文件大小是否相同，获取差异文件列表。
 - `--diffMode=length-checksum`，根据文件大小和 CRC 检验和是否相同，获取差异文件列表。
- `--diffOutput` 指定 diff 操作的输出目录。

如果目标文件系统为 COS，且源文件系统的 CRC 算法与之不同，则 COSDistCp 会拉取源文件计算目的文件系统的 CRC，以进行相同 CRC 算法值的对比。以下示例中，在迁移完成后，使用 `--diffMode` 参数，根据文件大小和 CRC 值，校验源和目标文件是否相同：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-1250000000/data/warehouse/ --diffMode=length-checksum --diffOutput=/tmp/diff-output
```

以上命令执行成功后，会输出以源文件系统文件列表为基准的计数器信息（请确保您的任务提交机器，配置了 MapReduce 任务的提交端 INFO 日志输出），您可以根据计数器信息，分析源和目的是否相同，计数器信息说明如下：

- 源和目的文件相同，记录为 SUCCESS。
- 目标文件不存在，记录为 DEST_MISS。
- 存在源目录的清单中，但是校验时源文件不存在，记录为 SRC_MISS。
- 源文件和目标文件大小不同，记录为：LENGTH_DIFF。
- 源文件和目标文件 CRC 算法值不同，记录为：CHECKSUM_DIFF。
- 由于读取权限不够等因素导致 diff 操作失败，记录为：DIFF_FAILED。

- 源为目录，目的为文件，记录为：TYPE_DIFF。

此外，COSDistcp 会在 HDFS 的 `/tmp/diff-output/failed` 目录下（1.0.5 及之前版本为 `/tmp/diff-output`），生成差异文件列表，您可以通过如下命令，获取除 SRC_MISS 以外的差异文件列表：

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

执行如下命令，根据差异文件列表进行增量迁移：

```
hadoop jar cos-distcp-${version}.jar --taskNumber=20 --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse/ --
previousManifest=file:///usr/local/service/hadoop/diff-manifest.gz --copyFromManifest
```

增量迁移完成后，再次运行带 `--diffMode` 参数的命令，校验文件是否完全一致。

校验源文件和目标文件是否具有相同 CRC

以参数 `--checkMode` 执行命令，文件拷贝完成时，校验源文件和目标文件长度及校验和是否一致，默认值 `length-checksum`。

从非 COS 文件系统同步到 COS 时，如果源的 CRC 算法和 Hadoop-COS 的 CRC 算法不一致，则拷贝时计算 CRC，并在拷贝完成后，获取目标 COS 文件的 CRC，和计算得到的源文件 CRC 对比校验：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-
1250000000/data/warehouse --checkMode=length-checksum
```

⚠ 注意：

在未指定 `--groupBy`，且 `--outputCodec` 为默认值时生效。

限制单文件读取带宽

以参数 `--bandWidth` 执行命令，数值单位为MB。限制每个迁移文件的读取带宽为10MB/s，示例如下：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-
1250000000/data/warehouse --bandWidth=10
```

多目录同步

新建一个本地文件（例如 `srcPrefixes.txt`），在该文件中添加需要迁移的多个目录的绝对路径，这些目录之间需要不存在父子关系，添加之后，可通过 `cat` 命令查看，示例如下：

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

使用 `--srcPrefixesFile` 参数指定该文件，执行迁移命令：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --srcPrefixesFile
file:///usr/local/service/hadoop/srcPrefixes.txt --dest cosn://examplebucket-
1250000000/data/warehouse/ --taskNumber=20
```

对输入文件进行正则表达式过滤

以参数 `--srcPattern` 执行命令，只同步 `/data/warehouse/` 目录下，以 `.log` 结尾的日志文件，示例如下：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/ --dest cosn://examplebucket-1250000000/data/warehouse --srcPattern='.*\.log$'
```

不迁移以 .temp 或 .tmp 结尾的文件：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/ --dest cosn://examplebucket-1250000000/data/warehouse/ --srcPattern='.*(?
```

指定 Hadoop-COS 的文件检验和类型

以参数 `--cosChecksumType` 执行命令，默认 CRC32C，可选 CRC32C 和 CRC64。

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-1250000000/data/warehouse --cosChecksumType=CRC32C
```

指定 COS 文件的存储类型

以参数 `--storageClass` 执行命令，示例如下：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-1250000000/data/warehouse/ --outputManifest=manifest-2020-01-10.gz --storageClass=STANDARD_IA
```

指定目标文件的压缩类型

以参数 `--outputCodec` 执行命令，您可通过该参数，将 HDFS 中的数据实时压缩备份到 COS，节省存储成本。参数可选值为：keep、none、gzip、lzop、snappy，none 选项保存的目标文件为未压缩状态，keep 保持原来文件的压缩状态。示例如下：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/logs --dest cosn://examplebucket-1250000000/data/warehouse/logs-gzip --outputCodec=gzip
```

⚠ 注意：

其中除 keep 选项外，皆会先对文件先解压，随后转换为目标压缩类型。因此，除 keep 选项外，可能会由于压缩参数等不一致，导致目标文件和源文件不一致，但解压后的文件一致；在未指定 `--groupBy`，且 `--outputCodec` 为默认值时，可通过 `--skipMode` 进行增量迁移，`--checkMode` 进行数据校验。

删除源文件

以参数 `--deleteOnSuccess` 执行命令，将 `/data/warehouse` 目录下文件从 HDFS 同步到 COS 后，立即删除源目录中的对应文件：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-1250000000/data/warehouse --deleteOnSuccess
```

⚠ 注意：

指定该选项后每迁移完一个文件，立即删除对应的源文件，并非整个迁移完成后，再删除源文件，请谨慎使用，1.7 及其以后版本不再提供该参数。

生成目标清单文件和指定上一次清单输出文件

以参数 `--outputManifest` 和 `--previousManifest` 执行命令。

- `--outputManifest` 该选项首先会在本地生成一个 gzip 压缩的 manifest.gz，并在迁移成功时，移动到 `--dest` 所指定的目录下。
- `--previousManifest` 指定上一次 `--outputManifest` 输出文件，COSDistCp 会跳过相同长度大小的文件：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-1250000000/data/warehouse/ --outputManifest=manifest.gz --previousManifest=cosn://examplebucket-1250000000/data/warehouse/manifest-2020-01-10.gz
```

⚠ 注意：

上述命令的增量迁移，只能同步文件大小变化的文件，无法同步文件内容发生变化的文件。如果文件内容可能发生变化，请参考 `--diffMode` 使用示例，根据文件的 CRC 确定文件发生变化的文件列表。

指定迁移任务分配策略为动态分配

如果您的文件分布情况是文件大小分化剧烈，例如极少量超大文件，大量小文件或者迁移机器负载不一。您可以通过 `--enableDynamicStrategy` 开启任务动态分配策略，使执行速度快的任务，迁移更多的文件，从而减少任务的执行时间：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-1250000000/data/warehouse --enableDynamicStrategy
```

迁移完成后对迁移的数据进行校验：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-1250000000/data/warehouse/ --diffMode=length-checksum --diffOutput=/tmp/diff-output
```

⚠ 注意：

该模式存在一定局限性，例如任务计数器在进程异常的情况下，可能计数不准确，请迁移完成后用 `--diffMode` 对数据进行校验。

拷贝文件的元信息

以参数 `--preserveStatus` 执行命令，将源文件或源目录的 user、group、permission 和 timestamps（modification time 和 access time）拷贝到目标文件或目标目录，该参数在将文件从 HDFS 拷贝到 CHDFS 时生效。

示例如下：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-1250000000/data/warehouse/ --preserveStatus=ugpt
```

配置 Prometheus 监控

您可以在 Yarn 的管理页面，查看 COSDistcp 迁移任务的计数器，包括当前已经迁移完成的文件数、字节数等信息。为了方便地查看迁移任务计数器的曲线变化，您只需要简单的配置，即可将 COSDistcp 迁移任务的计数器展示在 Prometheus + Grafana 监控体系中，配置 prometheus.yml（腾讯云监控可以控制台添加参考 [Prometheus 监控服务文档](#)，自建集群即下载 Prometheus 官网配置即可），添加抓取任务：

❗ 说明：

可以使用腾讯云 Prometheus 服务部署监控，参考 [Prometheus 监控服务文档](#) 在控制台添加 DistCp 相关配置。或使用自建集群，下载 [Prometheus 官网配置](#) 即可。



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

以参数 `--promPort=VALUE` 执行命令，将当前 MapReduce 任务的计数器暴露到外部：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-1250000000/data/warehouse --promPort=9028
```

下载示例 [Grafana Dashboard](#) 并导入，Grafana 展示如下：



拷贝文件失败时告警

以参数 `--completionCallbackClass` 指定回调类路径执行命令，COSDistCp 会在拷贝任务完成的时候，将收集的任务信息作为参数执行回调函数。用户自定义的回调函数，需要实现如下接口，前往 [下载回调示例代码](#)：

```
package com.qcloud.cos.distcp;
import java.util.Map;
public interface TaskCompletionCallback {
/**
 * @description: When the task is completed, the callback function is executed
 * @param jobType Copy or Diff
 * @param jobStartTime the job start time
 * @param errorMsg the exception error msg
 * @param applicationId the MapReduce application id
 * @param cosDistCpCounters the job
 */
}
```

```
void doTaskCompletionCallback(String jobType, long jobStartTime, String errorMsg, String
applicationId, Map<String, Long> cosDistCpCounters);

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

COSDistCp 内部集成了腾讯云可观测平台的告警，在任务出现异常及存在文件拷贝失败的时候，执行告警：

```
export alarmSecretId=SECRET-ID
export alarmSecretKey=SECRET-KEY
export alarmRegion=ap-guangzhou
export alarmModule=module
export alarmPolicyId=policy-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-1250000000/data/warehouse/ \
--checkMode=checksum \
--completionCallbackClass=com.qcloud.cos.distcp.DefaultTaskCompletionCallback
```

以上命令中 alarmPolicyId 为腾讯云可观测平台告警策略，可进入腾讯云可观测平台控制台进行创建和配置（告警管理 > 告警配置 > 新建策略）。

常见问题

使用 COSDistcp 迁移 HDFS 数据包含哪些阶段，如何调整迁移性能和保障数据的正确性？

COSDistcp 每迁移完成一个文件，都会根据 checkMode 对迁移的文件进行校验：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-
1250000000/data/warehouse --taskNumber=20
```

此外，您也可以迁移完成后，执行如下的命令，查看源和目的的差异文件列表：

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebucket-
1250000000/data/warehouse/ --diffMode=length-checksum --diffOutput=tmp/diff-output
```

环境中未配置 Hadoop-COS，如何运行 COSDistCp？

对于环境中未配置 Hadoop-COS 插件的用户，根据 Hadoop 版本，下载对应版本的 COSDistCp jar 包后，指定 Hadoop-COS 相关参数执行拷贝任务：

```
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-1250000000/warehouse
```

拷贝结果显示部分文件拷贝失败，如何处理？

COSDistCp 会对文件拷贝过程中出现的 IOException 重试五次，五次拷贝仍然失败，会将失败的文件信息写入

/tmp/\${randomUUID}/output/failed/ 目录下，其中，\${randomUUID} 为随机字符串。常见的拷贝失败原因包括：

1. 源文件存在拷贝清单中，但是拷贝时源文件不存在，记录为 SRC_MISS。
2. 任务发起的用户，不具备读取源文件或写入目标文件的权限，以及其他原因，记录为 COPY_FAILED。

如果日志信息记录源文件不存在，且源文件确实可以忽略，您可以通过如下命令，获取除 SRC_MISS 以外的差异文件列表：

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

如果存在除 SRC_MISS 以外的失败文件，您可以根据汇总在 /tmp/\${randomUUID}/output/logs/ 目录下的异常日志信息和拉取应用日志诊断原因，例如拉取 yarn 应用的日志，可使用如下命令：

```
yarn logs -applicationId application_1610615435237_0021 > application_1610615435237_0021.log
```

其中 application_1610615435237_0021 为应用 ID。

COSDistCp 是否会在网络等异常情况下，拷贝生成不完整文件？

在网络异常、源文件缺失和权限不足等情况下，COSDistCp 无法在目标端生成和源端同样大小的文件。

- 对于 COSDistCp 1.5 以下版本，COSDistCp 会尝试删除生成在目标端文件。如果删除失败，则需要您重新执行拷贝任务覆盖这些文件，或者手动删除这些不完整的文件。
- 对于 COSDistCp 1.5 及以上版本，且运行环境的 Hadoop COS 插件版本在 5.9.3 及以上版本时，如果拷贝到 COS 拷贝失败，COSDistCp 会调用 abort 接口终止正在上传的请求。因此，即使遇到异常情况，也不会生成不完整的文件。
- 对于 COSDistCp 1.5 及以上版本，如果运行环境的 Hadoop COS 插件版本低于 5.9.3，建议升级到 5.9.3 及其以上版本。
- 对于非 COS 的目标端，COSDistCp 会尝试删除目标端文件。

COS 存储桶中存在一些看不见的未完成上传文件，占用存储空间，如何处理？

由于机器异常、进程被 Kill 等因素，可能导致 COS 存储桶中存在一些碎片文件占用存储空间，您可参考官网 [生命周期文档](#) 配置碎片删除规则，进行清理。

迁移过程中，出现内存溢出和任务超时，如何进行参数调优？

在迁移过程中，COSDistCp 和访问 COS 和 CHDFS 的工具，基于自身逻辑，都会占用一些内存。为避免内存溢出和任务超时，您可以进行一些 MapReduce 任务的参数调整，例如：

```
hadoop jar cos-distcp-${version}.jar -Dmapreduce.task.timeout=18000 -
Dmapreduce.reduce.memory.mb=8192 --src /data/warehouse --dest cosn://examplebucket-
1250000000/data/warehouse
```

其中，将任务的超时时间 mapreduce.task.timeout 调整为 18000 秒，避免拷贝超大型文件时，出现任务超时；将 Reduce 进程的内存空间 mapreduce.reduce.memory.mb 大小调整为 8GB，避免内存溢出。

通过专线迁移，如何控制迁移任务的迁移带宽？

COSDistCp 迁移的总带宽限制计算公式为： $\text{taskNumber} * \text{workerNumber} * \text{bandWidth}$ ，您可以将 workerNumber 设置为 1，通过参数 taskNumber 控制迁移并发数，以及参数 bandWidth 控制单个并发的带宽。

迁移报错处理：Caused by: java.lang.ClassNotFoundException: org.apache.hadoop.fs.cosn.CRC64Checksum

原因：

COSDistcp 迁移文件时，在执行 Map 任务前，需要在 setup() 方法中进行相关变量或者资源的集中初始化工作。在此需要对 cosChecksumType 进行初始化，通过 loadclass 加载指定类，如果该类没有被加载过或父加载器没有加载成功，需要当前类进行加载，加载一个类需要调用 findClass 方法发现类的路径；

以上错误正是因为没有找到指定的 CRC64Checksum 类引发的。

解决方法：

- 首先检查 COSDistcp 的版本，在 cos-distcp-1.12.0 之前的版本是包含了 hadoop-cos 等的 jar 包，因此不会出现上述问题；在 cos-distcp-1.12.0 之后的版本中，是不包含其他关联的 jar 包的，因此需要下载相关 jar 包。
- 在根目录下，使用 find 查看报错类是否存在(延伸到相关类)，命令如下：

```
find / -name "*.jar" -exec grep -Hls "org/apache/hadoop/fs/cosn/CRC64Checksum" {} \;
```

- 若不存在，需在官网文档中下载 hadoop-cos 和 cos java sdk 插件包（详情请参见 [COSDistCp 工具](#)），然后将插件包放入指定的 Hadoop ClassPath 路径下，最后使用 -libjars 指定两个包的全路径；具体操作如下：
 - 运行 hadoop classpath，查看当前环境变量 CLASSPATH 的设置；
 - 将 hadoop-cos 和 cos java sdk 放入 /usr/local/services/hadoop/share/hadoop/tools/lib/ 路径下；

```
-rw-r--r-- 1 hadoop hadoop 4086455 Nov 27 19:44 hadoop-common-2.8.5.jar
-rw-r--r-- 1 hadoop hadoop 2535728 Nov 11 11:26 hadoop-common-2.8.5-tests.jar
-rw-r--r-- 1 root root 156938 Nov 21 10:04 hadoop-cos-2.8.5-8.1.8.jar
```

```
-rw-r--r-- 1 hadoop hadoop 273370 Nov 11 11:26 commons-net-3.1.jar
-rw-r--r-- 1 hadoop hadoop 130663 Nov 11 11:26 commons-pool2-2.6.0.jar
-rw-r--r-- 1 root root 11282407 Oct 30 12:27 cos_api-bundle-5.6.100.jar
-rw-r--r-- 1 hadoop hadoop 69500 Nov 11 11:26 curator-client-2.7.1.jar
-rw-r--r-- 1 hadoop hadoop 186273 Nov 11 11:26 curator-framework-2.7.1.jar
-rw-r--r-- 1 hadoop hadoop 270342 Nov 11 11:26 curator-recipes-2.7.1.jar
```

- 运行命令：

```
hadoop jar cos-distcp-1.13.0-3.1.0.jar -libjars
/usr/local/services/hadoop/share/hadoop/tools/lib/hadoop-cos-2.8.5-8.1.8.jar,
/usr/local/services/hadoop/share/hadoop/tools/lib/cos_api-bundle-5.6.100.jar .....
```

- 若存在，可能是环境中还有其他版本的相关 jar 包，需要将低版本的 jar 包移动或者删除，然后再进行上述操作。

HDFS TO COS 工具

最近更新时间：2022-12-16 15:30:25

功能说明

HDFS TO COS 工具用于将 HDFS 上的数据拷贝到腾讯云对象存储（Cloud Object Storage，COS）上。

使用环境

系统环境

Linux 或 Windows 系统。

软件依赖

JDK 1.7或1.8。

安装与配置

具体环境安装与配置请参见 [Java 安装与配置](#)。

配置方法

1. 安装 Hadoop-2.7.2 及以上版本，具体安装步骤请参见 [Hadoop 安装与测试](#)。
2. 在 [GitHub](#) 下载 HDFS TO COS 工具并解压缩。
3. 将要同步的 HDFS 集群的 core-site.xml 拷贝到 conf 文件夹中，其中 core-site.xml 中包含 NameNode 的配置信息。
4. 编辑配置文件 cos_info.conf，存储桶（Bucket）、地域（Region）以及 API 密钥信息，其中存储桶名称，由用户自定义字符串和系统生成 APPID 数字串以及中划线连接而成，例如 examplebucket-1250000000。
5. 在命令行参数中指定配置文件位置，默认位置 conf/cos_info.conf。

⚠ 注意

当命令行参数中的参数与配置文件重合时，以命令行为准。

使用方法

❗ 说明

下面以 Linux 为例介绍如何使用。

查看帮助

```
./hdfs_to_cos_cmd -h
```

文件拷贝

- 从 HDFS 拷贝到 COS，若 COS 上已存在同名文件，则会覆盖原文件。

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

- 从 HDFS 拷贝到 COS，若 COS 上已存在同名且长度一致的文件时，则忽略上传（适用于拷贝一次后，重新拷贝）。

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

这里只对长度进行判断，因为如果将 Hadoop 上的文件摘要进行计算，开销较大。

- 从 HDFS 拷贝到 COS，若 HDFS 中存在 Har 目录（Hadoop Archive 归档文件），通过指定 `--decompress_har` 参数可以自动解压 har 文件：

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

若未指定 `--decompress_har` 参数，默认按照普通的 HDFS 目录进行拷贝，即 .har 目录下的 index 和 masterindex 等文件原样拷贝。

目录信息

```
conf : 配置文件，用于存放 core-site.xml 和 cos_info.conf
log  : 日志目录
src  : Java 源程序
dep  : 编译生成的可运行的 JAR 包
```

问题与帮助

关于配置信息

请确保填写的配置信息正确，包括存储桶（Bucket）、地域（Region）以及 API 密钥信息，其中，存储桶的名字，由用户自定义字符串和系统生成 APPID 数字串由中划线连接而成，例如 examplebucket-1250000000。并保证机器的时间和北京时间一致（相差1分钟左右是正常的），如果相差较大，请重新设置机器时间。

关于 DataNode

请保证对于 DataNode，拷贝程序所在的机器也可以连接。NameNode 有外网 IP 可以连接，但获取的 block 所在的 DataNode 机器是内网 IP，是无法直接连接上的。因此建议同步程序放在 Hadoop 的某个节点上执行，保证对 NameNode 和 DataNode 皆可访问。

关于权限

请使用 Hadoop 命令下载文件，检查是否正常，再使用同步工具同步 Hadoop 上的数据支持。

关于文件覆盖

对于 COS 上已存在的文件，默认进行重传覆盖。除非用户明确的指定 `-skip_if_len_match`，当文件长度一致时，则跳过上传。

关于 cos path

cos path 默认认为是目录，最终从 HDFS 上拷贝的文件都会存放在该目录下。

关于从腾讯云 EMR HDFS 拷贝数据

从腾讯云 EMR HDFS 拷贝数据到 COS，建议使用高性能 Distcp 工具，请参见 [Hadoop 文件系统与 COS 之间的数据迁移](#)。

GooseFS-Lite 工具

最近更新时间：2025-02-19 10:33:33

功能说明

GooseFS-Lite 工具支持将 [对象存储（Cloud Object Storage, COS）](#) 存储桶挂载到本地，像使用本地文件系统一样直接操作腾讯云对象存储中的对象，相比于 COSFS 工具，GooseFS-Lite 可提供更高的大文件读写速度，不受本地磁盘的性能限制。GooseFS-Lite 支持 POSIX 文件系统的主要功能，例如文件顺序、随机读，顺序写、目录操作等功能。

使用限制

GooseFS-Lite 仅适合挂载后对文件进行简单的管理，不支持本地文件系统的一些功能用法。请注意以下使用限制：

- 不支持对文件进行随机写和 truncate 操作。
- 多个客户端挂载同一个 COS 存储桶时，依赖用户自行协调多个客户端的行为。例如避免多个客户端写同一个文件等。
- 文件/文件夹的 rename 操作非原子操作。
- 不支持读取和 rename 当前挂载点正在写入的文件。
- 元数据操作如 list directory，性能较差，因为需要远程访问 COS 服务器。
- 不支持 soft/hard link。
- 追加写性能较差，涉及服务端数据拷贝和下载被追加文件。
- 不建议在小内存场景使用，例如在容器内存或者 CVM 内存小于 2G 的情况。
- 容器环境目前只支持 [腾讯云容器服务（Tencent Kubernetes Engine, TKE）](#) 内使用，非 TKE 容器暂不支持。
- 不建议在大量随机读且对性能有较高要求的场景中使用。

⚠ 注意：

外网挂载和非低频存储的追加写操作，会产生下载流量费用。

使用环境

- [KonaJDK 11](#)
- Linux X86_64

使用方法

步骤1：安装依赖

CentOS/TencentOS Server

```
yum install -y fuse-devel
```

Ubuntu

```
apt install -y libfuse-dev
```

其他 Linux 发行版

编译安装 libfuse2.9.7

```
wget "https://github.com/libfuse/libfuse/releases/download/fuse-2.9.7/fuse-2.9.7.tar.gz"
tar xvf fuse-2.9.7.tar.gz
cd fuse-2.9.7
./configure
```

```
make -j8
make install
```

步骤2：安装 GooseFS-Lite

将 GooseFS-Lite 安装到当前目录，并将 `goosefs-lite` 软链到 `/usr/bin/goosefs-lite`，方便后续直接使用 `goosefs-lite` 命令。

```
curl -fssL https://downloads.tencentgoosefs.cn/goosefs-lite/install.sh | sh -x
cd goosefs-lite-*
sudo bash bin/install.sh
```

步骤3：安装 KonaJDK11

在 `goosefs-lite-<具体版本号>` 目录下（以1.0.6版本为例，在 `goosefs-lite-1.0.6` 目录下）使用如下命令安装 KonaJDK 到 `/usr/local/konajdk11`下：

```
sudo bash bin/install-jdk.sh https://github.com/Tencent/TencentKona-
11/releases/download/kona11.0.22/TencentKona-11.0.22.b1-jdk_linux-x86_64.tar.gz
```

如下所示，有两种方式可以选择：

一：使用 `konajdk` 的下载链接，二：将自行下载 `konajdk` 安装包到指定目录。使用如下命令安装，即可让 `goosefs-lite` 自动使用该 Java 运行环境。

```
Usage:
Command: install-jdk.sh http[s]://host/path
Example: install-jdk.sh https://github.com/Tencent/TencentKona-
11/releases/download/kona11.0.22/TencentKona-11.0.22.b1-jdk_linux-x86_64.tar.gz
or
Command: install-jdk.sh /path/to/jdk.tar.gz
Example: install-jdk.sh /Downloads/TencentKona-11.0.22.b1-jdk_linux-x86_64.tar.gz
```

如果希望更灵活地安装 Java 环境，可以参考 [手动安装JDK](#)。并在 `conf/goosefs-env.sh` 中修改环境变量使其生效。

步骤4：修改配置文件

在 `goosefs-lite-<具体版本号>` 目录下（以1.0.6版本为例，在 `goosefs-lite-1.0.6` 目录下）修改配置文件有以下两种方式：

- 使用 `sed` 修改以下三个参数，其中 `SECRET_ID`、`SECRET_KEY` 和 `REGION` 按照实际填写：
 - 将 `fs.cosn.userinfo.secretKey` 配置为腾讯云密钥 Key。
 - 将 `fs.cosn.userinfo.secretId` 配置为腾讯云密钥 ID。
 - 将 `fs.cosn.bucket.region` 配置为存储桶地域。

```
sed -i '/<name>fs.cosn.userinfo.secretId</name>/{N;s/<value>[^<]*</value>/<value>$SECRET_ID</value>/' conf/core-site.xml
sed -i '/<name>fs.cosn.userinfo.secretKey</name>/{N;s/<value>[^<]*</value>/<value>$SECRET_KEY</value>/' conf/core-site.xml
sed -i '/<name>fs.cosn.bucket.region</name>/{N;s/<value>[^<]*</value>/<value>$REGION</value>/' conf/core-site.xml
```

- 也可使用 `vim` 编辑 `conf/core-site.xml` 文件，修改参数。
 - 将 `fs.cosn.userinfo.secretKey` 配置为腾讯云密钥 Key。
 - 将 `fs.cosn.userinfo.secretId` 配置为腾讯云密钥 ID。
 - 将 `fs.cosn.bucket.region` 配置为存储桶地域。

配置文件描述

在 `goosefs-lite-<具体版本号>/conf` 目录下（以1.0.6版本为例，在 `goosefs-lite-1.0.6/conf` 目录下）目录下，可以看到以下文件：

- **acl-site.properties**：指定目录的权限、用户名、组名、类似 Linux Posix 语义的0755、uid、gid 等。
- **core-site.xml**：Hadoop-COS 的配置文件（goosefs-lite 的数据流基于 Hadoop-COS，因此参数和配置文件几乎通用）
- **goosefs-env.sh**：各类环境变量，如 JVM 参数等。
- **goosefs-lite.properties**：goosefs-lite 相关配置。
- **log4j.properties**：日志配置。如需要开启 debug 日志，可取消注释该文件最后一行，重新挂载即可。
- **acl-site.properties**：指定目录的权限、用户名、组名、类似 Linux Posix 语义的0755、uid、gid 等。

⚠ 注意：

- 建议用户尽量避免在配置中使用永久密钥，采取配置子账号密钥或者临时密钥的方式有助于提升业务安全性。为子账号授权时建议按需授权子账号可执行的操作和资源，避免发生预期外的数据泄露。
- 如果您一定要使用永久密钥，建议对永久密钥的权限范围进行限制，可通过限制永久密钥的可执行操作、资源范围和条件（访问 IP 等），提升使用安全性。

步骤5：将存储桶挂载到本地目录

在 `goosefs-lite-<具体版本号>` 目录下（以1.0.6版本为例，在 `goosefs-lite-1.0.6` 目录下）执行如下命令，将密钥文件中配置的存储桶挂载到指定目录：

```
./bin/goosefs-lite mount <MountPoint> cosn://<BucketName>/
```

其中：

- **<MountPoint>** 为本地挂载目录（例如 `/mnt/goosefs-lite-mnt-dir`），此目录必须为空，否则无法挂载。
- **<BucketName>** 为存储桶名称（例如 `examplebucket-1250000000`）。

示例：

```
mkdir -p /mnt/goosefs-lite-mnt
./bin/goosefs-lite mount /mnt/goosefs-lite-mnt/ cosn://examplebucket-1250000000/
```

查看本地挂载点对应的 COS 存储桶，输出信息依次为进程 ID、本地挂载点和 COS 路径：

```
$ ./bin/goosefs-lite stat
pid      mount_point      cos_path
13815    /mnt/goosefs-lite-mnt/  cosn://examplebucket-1250000000/
```

如果您需要在命令行中，同时指定多个挂载参数，可以使用逗号分隔多个参数，如下命令设置挂载点只读，且允许除其他用户访问挂载点：

```
./bin/goosefs-lite mount -o "ro,allow_other" mnt/ cosn://examplebucket-1250000000/
```

其中：

- **-o allow_other**：如果要允许其他用户访问挂载文件夹，可以在运行 GooseFS-Lite 的时候指定该参数。
- **-o ro**：将挂载点设置为只读，不允许写入和删除操作。

❗ 说明：

单个参数可通过 `-o` 指定，例如 `-o ro`；多个参数可通过逗号分割，例如 `-o "ro,allow_other"`。

步骤6：卸载挂载点

在 `goosefs-lite-<具体版本号>` 目录下（以1.0.6版本为例，在 `goosefs-lite-1.0.6` 目录下），执行如下命令，即可卸载挂载点：

```
$. /bin/goosefs-lite umount /mnt/goosefs-lite-mnt
Unmount fuse at /mnt/goosefs-lite-mnt/ (PID: 17206).
```

```
# 如果卸载异常，可以使用以下命令强制卸载
$ sudo umount -l /mnt/goosefs-lite-mnt
```

步骤7：参数调优

GooseFS-Lite 包含两个配置文件，分别为 `conf/core-site.xml` 及 `conf/goosefs-lite.properties`。您可以通过修改 `conf/core-site.xml` 调优上传、下载带宽。常用参数如下，更多参数可参见 [Hadoop-COS](#) 文档。

属性键	说明	默认值	必填项
<code>fs.cosn.useHttps</code>	配置是否使用 HTTPS 作为与 COS 后端的传输协议。	true	否
<code>fs.cosn.upload.part.size</code>	分块上传的每个 part size 的大小。由于 COS 的分块上传最多只能支持10000块，因此需要预估最大可能使用到的单文件大小。例如，part size 为8MB时，最大能够支持78GB的单文件上传。part size 最大可以支持到2GB，即单文件最大可支持19TB。	8388608 (8MB)	否
<code>fs.cosn.upload_thread_pool</code>	文件流式上传到 COS 时，并发上传的线程数目。	32	否
<code>fs.cosn.read.ahead.block.size</code>	预读块的大小。	1048576 (1MB)	否
<code>fs.cosn.read.ahead.queue.size</code>	预读队列的长度。	6	否
<code>fs.cosn.trsf.fs ofs.tmp.cache.dir</code>	元数据加速存储桶的临时文件目录。	无	元数据加速桶必填
<code>fs.cosn.trsf.fs ofs.user.appid</code>	元数据加速存储桶的 Appid。	无	元数据加速桶必填
<code>fs.cosn.trsf.fs ofs.bucket.region</code>	元数据加速存储桶所在的地域，如 ap-shanghai, ap-beijing。	无	元数据加速桶必填

您可以通过修改 `conf/goosefs-lite.properties` 对 GooseFS-Lite 的行为进行调整。常用参数如下：

属性	说明	默认值	必填项
<code>goosefs.fuse.list.entries.cache.enabled</code>	是否开启客户端 List 缓存	true	否
<code>goosefs.fuse.list.entries.cache.max.size</code>	客户端 List 最大缓存的条目数，单位：条	100000	否
<code>goosefs.fuse.list.entries.cache.max.expiration.time</code>	客户端 List 缓存的有效时间，单位：ms	15000	否
<code>goosefs.fuse.async.release.max.wait.time</code>	open 和 rename 操作的文件正在被写入时，等待写入操作完成的时间，单位：ms	5000	否
<code>goosefs.fuse.umount.timeout</code>	卸载文件系统时，等待未完成操作的时间，单位：ms	120000	否

当您的读取和写入并发度较大，您可以通过如下方式，调整 GooseFS-Lite 最大 JVM 运行内存，避免 FullGC 和 OutOfMemoryError。JVM 默认值为 `-Xmx512m -XX:MaxDirectMemorySize=512m -XX:+UseG1GC -XX:G1HeapRegionSize=32m`，调整方式如下：

```
export JAVA_OPTS="-Xms2G -Xmx2G"
./bin/goosefs-lite mount /mnt/goosefs-lite-mnt/ cosn://examplebucket-1250000000/
ps -ef|grep goosefs-lite|grep -v grep
```

常见问题

缺少 libfuse 库文件，如何处理？

```
java.lang.UnsatisfiedLinkError: /tmp/libjnfuse2359305659114946750.so: libfuse.so.2: cannot open shared object file: No such file or directory
at java.base/java.lang.ClassLoader$NativeLibrary.load0(Native Method)
at java.base/java.lang.ClassLoader$NativeLibrary.load(ClassLoader.java:2445)
at java.base/java.lang.ClassLoader$NativeLibrary.loadLibrary(ClassLoader.java:2501)
at java.base/java.lang.ClassLoader.loadLibrary0(ClassLoader.java:2700)
at java.base/java.lang.ClassLoader.loadLibrary(ClassLoader.java:2630)
at java.base/java.lang.Runtime.load0(Runtime.java:768)
at java.base/java.lang.System.load(System.java:1837)
at com.qcloud.cos.goosefs.jnifuse.utils.NativeLibraryLoader.loadLibraryFromJar(NativeLibraryLoader.java:105)
at com.qcloud.cos.goosefs.jnifuse.utils.NativeLibraryLoader.loadLibrary(NativeLibraryLoader.java:83)
at com.qcloud.cos.goosefs.jnifuse.LibFuse.loadLibrary(LibFuse.java:48)
at com.qcloud.cos.goosefs.jnifuse.LibFuse.<clinit>(LibFuse.java:32)
at com.qcloud.cos.goosefs.jnifuse.AbstractFuseFileSystem.<clinit>(AbstractFuseFileSystem.java:41)
at com.qcloud.cos.goosefs.fuse.GooseFSLiteFuse.main(GooseFSLiteFuse.java:64)
at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.base/java.lang.reflect.Method.invoke(Method.java:566)
at org.springframework.boot.loader.MainMethodRunner.run(MainMethodRunner.java:53)
at java.base/java.lang.Thread.run(Thread.java:829)
```

需要安装 libfuse，请参照以下方式进行操作：

方式一

1. 执行如下命令，安装 fuse-devel。

- 如果是 CentOS 或者 TencentOS 系统，执行如下命令：

```
yum install fuse-devel
```

- 如果是 Ubuntu 系统，执行如下命令：

```
apt install libfuse-dev
```

2. 安装完成后，执行如下命令，查看安装是否成功。

```
find / -name libfuse.so*
```

方式二

更新旧版本 libfuse.so.2.9.2，安装步骤如下：

❗ 说明：

CentOS 7默认安装的是 libfuse.so.2.9.2。

1. 下载 [libfuse 源码](#)，并编译生成 libfuse.so.2.9.7。

```
tar -zxvf fuse-2.9.7.tar.gz
cd fuse-2.9.7/ && ./configure && make && make install
echo -e '\n/usr/local/lib' >> /etc/ld.so.conf
ldconfig
```

2. 编译及生成 libfuse.so.2.9.7后，按照以下步骤进行替换：

2.1 执行以下命令，查找旧版本 libfuse.so.2.9.2库链接。

```
find / -name libfuse.so*
```

2.2 执行以下命令，将 libfuse.so.2.9.7 拷贝至旧版本库 libfuse.so.2.9.2 所在位置。

```
cp /usr/local/lib/libfuse.so.2.9.7 /usr/lib64/
```

2.3 执行以下命令，删除旧版本 libfuse.so 库的所有链接。

```
rm -f /usr/lib64/libfuse.so
rm -f /usr/lib64/libfuse.so.2
```

2.4 执行以下命令，建立与被删除旧版本链接类似的 libfuse.so.2.9.7 库链接。

```
ln -s /usr/lib64/libfuse.so.2.9.7 /usr/lib64/libfuse.so
ln -s /usr/lib64/libfuse.so.2.9.7 /usr/lib64/libfuse.so.2
```

如何配置开机挂载？

1. 编辑文件 /usr/lib/systemd/system/goosefs-lite.service，增加如下内容。您可以将 examplebucket-1250000000 换为您的存储桶：注意，以下 JAVA_OPTS 中 -Xms 和 -Xmx 配置的内存值不能超过所在节点的物理内存上限的 50%。如节点物理内存 16GB，那么建议最多配置 -Xms8G -Xmx8G。

下面以 goosefs-lite-1.0.6 版本为例：

```
[Unit]
Description=The Tencent Cloud GooseFS Lite for COS
Requires=network-online.target
After=network-online.target

[Service]
Type=forking
User=root
Environment="JAVA_OPTS=-Xms2G -Xmx4G -XX:MaxDirectMemorySize=1G -XX:+UseG1GC -XX:G1HeapRegionSize=32m"
ExecStart=/usr/local/goosefs-lite-1.0.6/bin/goosefs-lite mount /mnt/goosefs-mnt cosn://examplebucket-1250000000/
ExecStop=/usr/local/goosefs-lite-1.0.6/bin/goosefs-lite umount /mnt/goosefs-mnt
Restart=always
RestartSec=5

[Install]
WantedBy=multi-user.target
```

2. 执行如下命令，执行挂载命令和查看后台 Daemon 进程状态。

```
# 让 goosefs-lite 的 systemd 配置生效
systemctl daemon-reload
# 启动后台 Fuse 进程
systemctl start goosefs-lite
# 查看后台 Daemon 进程状态
systemctl status goosefs-lite
```



```
# 查看挂载点列表
/usr/local/goosefs-lite-1.0.6/bin/goosefs-lite stat
# 如果是修改systemd配置，则在修改后reload并restart。
```

设置为开机启动时尝试挂载：

```
systemctl enable goosefs-lite
```

3. 卸载挂载点，重启机器，并查看 Fuse 进程状态。

```
# 执行卸载，注意：请勿在数据写入的时卸载，否则会导致数据不完整
systemctl stop goosefs-lite
# 重启操作系统，请谨慎操作，不要影响业务
reboot -h now
# 查看后台 Daemon 进程状态
systemctl status goosefs-lite
# 查看挂载点列表
/usr/local/goosefs-lite-1.0.6/bin/goosefs-lite stat
```

GooseFS-Lite 每天在某个时间段里 CPU 使用率较高，且向 COS 发出大量 Head、List 请求，产生大量请求次数费用，该怎么处理？

这通常是由于您机器上存在定时扫盘任务导致的，Linux 系统上常见的扫盘程序是 updatedb，您可以将 GooseFS-Lite 挂载点目录，添加到 updatedb 的配置文件 /etc/updatedb.conf 文件的 PRUNEPATHS 配置项中，避免该程序的扫盘行为。此外，您可以使用 Linux 工具 auditd，查找访问 GooseFS-Lite 挂载点的程序。

操作步骤如下：

1. 安装 auditd。

- 如果是 Ubuntu 系统，执行如下命令：

```
apt-get install auditd -y
```

- 如果是 CentOS 系统，执行如下命令：

```
yum install audit audit-libs
```

2. 启动 auditd 服务。

```
systemctl start auditd
systemctl enable auditd
```

3. 监控挂载目录。

❗ 说明：

`-w` 指定 GooseFS-Lite 挂载目录，`-k` 为输出在 audit 日志中的 key。

```
auditctl -w /usr/local/service/mnt/ -k goosefs_lite_mnt
```

4. 根据日志确定访问程序。

audit 的日志目录：/var/log/audit，查询命令如下：

```
ausearch -i | grep 'goosefs_lite_mnt'
```

5. 停止 auditd 服务。

如果您需要停止 auditd 服务，可以使用如下命令：

```
/sbin/service auditd stop
```

⚠ 注意：

如果访问挂载点的程序一直在运行，新启动的 auditd，并不会监控到该程序的访问行为；程序中关于挂载目录的多次调用，只会记录第一次。

GooseFS-Lite 安装时报 “cannot allocate memory” 错误，如何处理？

```
OpenJDK 64-Bit Server VM warning: INFO: os::commit_memory(0x0000000080000000, 2147483648, 0) failed; error='Cannot allocate memory' (errno=12)
#
# There is insufficient memory for the Java Runtime Environment to continue.
# Native memory allocation (mmap) failed to map 2147483648 bytes for committing reserved memory.
# An error report file with more information is saved as:
# /usr/goosefs-lite/goosefs-lite-1.0.2/hs_err_pid2860293.log
~
```

该报错主要是因为 GooseFS-Lite 运行过程时分配内存异常报错，一般为申请内存大于实际可用内存。

您可以进入 `./bin/goosefs-lite` 文件中修改 `JAVA_OPT` 参数配置为合理内存数值，至少需要保证申请的内存数量小于实例可用内存数量。

怎么查看最新包版本？

执行如下命令，返回值即最新版本号。

```
curl -fsSL https://downloads.tencentgoosefs.cn/goosefs-lite/LATEST_VERSION
```

如何查看日志？

问题的排查需要依赖日志，以下是相关日志所在位置。

- 对于 **goosefs-lite 1.0.3版本及以上**，默认日志目录在 `/data/goosefs/logs/fuse` 下。
例如：您挂载 **goosefs-lite** 的用户为 `root`，挂载点路径为 `/data1/data2`，那么日志路径为：
`/data/goosefs/logs/fuse/root/data1/data2`。
- 对于 **goosefs-lite 1.0.0-1.0.2版本**，默认日志目录在 `/data/goosefs/logs` 下。

另一种查看挂载点日志路径的方式，首先重新挂载，然后 `ps aux | grep ${MOUNT_POINT}`，从输出中可以看到 `Error_File` 或者 `goosefs.logs.dir`，这个路径的父目录则是该挂载点的全部日志所在位置。

系统日志：如果是 **centos** 或者 **tlinux** 系统，则是 `/var/log/message*`，如果是 **ubuntu**，则是 `/var/log/syslog`。

如果是需要开启 **debug** 日志，进入 `conf/log4j.properties`，注释最下面一行内容，重新挂载即可。

⚠ 注意：

开启 **debug** 日志会影响性能，正常情况无需开启。

挂载失败报错“Name or service not known”，如何处理？

```
2023-05-16 19:23:20,668 [Thread-8] ERROR jnifuse.AbstractFuseFileSystem (AbstractFuseFileSystem.java:statfsCallback) -
Failed to statfs /:
java.lang.RuntimeException: java.net.UnknownHostException: VM-36-104-centos: VM-36-104-centos: Name or service not known
    at com.qcloud.cos.goosefs.util.network.NetworkAddressUtils.getLocalIpAddress(NetworkAddressUtils.java:522)
    at com.qcloud.cos.goosefs.util.network.NetworkAddressUtils.getLocalHostName(NetworkAddressUtils.java:444)
    at com.qcloud.cos.goosefs.util.network.NetworkAddressUtils.getLocalHostMetricName(NetworkAddressUtils.java:463)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.constructSourceName(MetricsSystem.java:202)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.lambda$static$0(MetricsSystem.java:89)
    at com.qcloud.cos.goosefs.util.CommonUtils$2.firstTime(CommonUtils.java:799)
    at com.qcloud.cos.goosefs.util.CommonUtils$2.get(CommonUtils.java:794)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.getMetricNameWithUniqueId(MetricsSystem.java:390)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.lambda$getClientMetricName$5(MetricsSystem.java:341)
    at java.util.concurrent.ConcurrentHashMap.computeIfAbsent(ConcurrentHashMap.java:1660)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.getClientMetricName(MetricsSystem.java:340)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.getMetricName(MetricsSystem.java:271)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.timer(MetricsSystem.java:529)
    at com.qcloud.cos.goosefs.fuse.GooseFSFuseUtils.call(GooseFSFuseUtils.java:287)
    at com.qcloud.cos.goosefs.fuse.GooseFSLiteJniFuseFileSystem.statfs(GooseFSLiteJniFuseFileSystem.java:1091)
    at com.qcloud.cos.goosefs.jnifuse.AbstractFuseFileSystem.statfsCallback(AbstractFuseFileSystem.java:289)
```

通常是因为无法解析域名，可以通过 ping 对应的域名尝试。如上图所示报错，可执行如下命令：

```
ping VM-36-104-centos
```

如果同样返回失败。可以通过修改 `/etc/hosts` 的方式，配置对应的 ip。一般可以配 `127.0.0.1`。

操作步骤如下：

1. 在 `/etc/hosts` 文件中添加一行。这里的 VM-36-104-centos 替换成您的主机名。

```
127.0.0.1 VM-36-104-centos
```

2. 之后重新 ping 测试，确认可以正常解析后，重新挂载即可。

默认环境 JDK 不是 KonaJDK11，怎么使用 GooseFS-lite？

1. 下载 [KonaJDK11](#) 的包，并解压。
2. 复制 konajdk 中 java 二进制程序的绝对路径。例如，解压的 jdk 在 `/root/konajdk11` 下。

```
# 确定java版本
/root/konajdk11/bin/java -v
# 修改当前shell的JAVA环境变量，不影响其他shell和进程
export JAVA=/root/konajdk11/bin/java
# 挂载
goosefs-lite mount /mnt cosn://bucket-appid
```

3. 使用 `ps aux | grep goosefs-lite`，可以看到进程开头是 `/root/konajdk11/bin/java`，说明已经使用了我们指定的 java 版本，操作完成。

挂载点本来是正常的，使用过程中突然无法正常使用了，如何处理？

假设当前的问题挂载点是 `/tmp/mount_point`。

1. 首先 `ps aux | grep /tmp/mount_point` 查看当前是否有进程正在使用这个挂载点，包括 `goosefs-lite`。如果有，使用 `kill` 命令杀掉对应的进程。

2. 使用 `ls` 尝试访问挂载点，如果返回为空，说明挂载点已经卸载成功。接着重新挂载即可。

```
ls /mount_point
```

3. 如果抛出 **transport is not connected** 之类的错误，则需要执行 `umount -l /mount_point` 强制卸载。（此命令执行需要 root 权限）

通常这类情况是进程被 `kill -9` 或系统 `oom-killer` 导致。可以在系统日志或者 `goosefs-lite` 日志中找到。

4. 观察挂载点日志目录（`/data/goosefs/logs/fuse/$USER/$MOUNT_POINT`）下是否有 `hs_error` 开头的日志（文件中描述程序退出前的堆栈及原因）。

抛异常：Unsupported or unrecognized SSL message，如何处理？

当前环境不支持关闭 `https` 模式，需要修改配置。解决方式为在 `core-site.xml` 配置文件中，增加如下内容：

```
<property>
  <name>fs.cosn.useHttps</name>
  <value>false</value>
</property>
```

GooseFS-Lite 怎么使用内网域名访问 COS？

在 `core-site.xml` 配置文件中，将 `fs.cosn.bucket.region` 属性删除，并添加 `fs.cosn.bucket.endpoint_suffix`。

想了解更多参数，可以阅读以下文档：

- [GitHub - tencentyun/hadoop-cos](#)
- [对象存储 Hadoop 工具](#)

挂载的时候或者使用过程中出现403 Forbidden，如何处理？

通常日志中 `ERROR` 类型日志会描述缺少什么权限，按需补充即可。

注意：对于 `head bucket` 权限，需要设置到桶级别而不能仅仅路径级别，例如，`cosn://bucket-appid/path` 这种是不生效的，必须是 `cosn://buket-appid`。此权限不会暴露桶内有哪些对象。

报错 “fuse: failed to open /dev/fuse Operation not permitted”，如何处理？

1. 检查是否拥有 root 权限。
2. 如果是容器中，检查容器启动时是否带有 `--privileged`，如果没有，加上。
3. 检查 `fuse` 内核模块是否安装：执行命令 `lsmod | grep fuse`，查看是否有返回结果，如果没有说明 `fuse` 内核模块缺失。

文件写入失败，报错 “part num: 10001, the parameter partNumber is not valid.”，如何处理？

这种情况说明分块上传的块数超出限制了，COS 最大只支持1万个 `part` 上传。所以 `goosefs-lite` 默认最大支持 `8MB*10000`（约78GB）的文件。如果需要支持更大的文件，需要调整参数 `fs.cosn.upload.part.size`。例如，`fs.cosn.upload.part.size` 修改为 `16777216`（即16MB），则可以支持 `16MB*10000` 的大文件。

在线辅助工具

COS 签名工具

最近更新时间：2024-10-14 16:45:21

功能概述

COS 签名工具是腾讯云对象存储（Cloud Object Storage，COS）为用户提供的 Web 工具，可用于生成请求签名。您可以在工具页面上填入指定的参数，生成请求签名，以及校验请求签名的正确性。

- 目前 COS 存在 XML 和 JSON 两个不同版本的 API，两类 API 的签名入参有所差异，COS 推荐您使用 XML 版本的 API。JSON API 是腾讯云 COS 服务在推出 XML API 前为用户提供接入使用 COS 的 API 接口，接口与标准 XML 的 API 底层架构相同，数据互通，可以交叉使用，但是接口不兼容。
- 有关 XML 版本的签名介绍文档，请参见 XML 版本 [请求签名](#)。
- 有关 JSON 版本的签名介绍文档，请参见 JSON 版本 [签名算法](#)。

注意事项

- 本签名工具不识别和提示您填写的错误参数。如果您使用本工具计算请求签名并与 SDK 或其他工具的计算结果进行比较，本工具不会主动纠正您填写的错误参数。
- 本签名工具将识别您填写的不合法参数，并进行相应的提示。但仍可能生成一个无法通过签名验证的请求签名。
- 本签名工具将识别及提示您尚未填写的必填参数项，且如果您未完成必填参数项的填写，您将无法生成请求签名。

工具地址

单击进入 [COS 签名工具](#)。

使用方法

XML 版本签名工具

输入基础配置信息

- 单击 [COS 签名工具](#)，进入 COS 签名工具页面。
- 在基础信息栏中，选择签名类型，填写签名有效时间。
 - 签名类型：选择 XML API 签名。
 - 签名有效时间：您可以单击[获取（60分钟有效期）](#)，即可获取一个有效时长为60分钟的签名。您也可以自行输入一个有效起止时间用于复现在该起止时间下的签名结果。相关签名有效时间的介绍，请参见 [请求签名](#)。

❗ 说明

基础信息中的参数均为必填项。

输入 API 密钥信息

在 API 密钥栏中，填写您的 API 密钥信息。

❗ 说明

- API 密钥中的信息均为必填项。
- API 密钥的参数信息，建议使用子账号密钥，授权遵循最小权限指引，降低使用风险。子账号密钥获取可参见 [子账号访问密钥管理](#)。
- 填写时请确保该信息的准确性。填写错误将导致您的签名被视为无效签名。

输入 HTTP 参数信息

在 HTTP 参数栏中，填写相关参数。主要参数如下：

- **HttpMethod**: 必选项。HTTP 请求方法, 包括 GET、POST、PUT、DELETE、HEAD。
- **HttpURI**: 必填项。HTTP 请求 URI 部分, 即您需要发起请求的对象名称。
- **HttpParameters**: 可选项。HTTP 请求参数。当您需验证 url 参数时可填写该参数。其中, key 小写, value 需要进行 URLEncode, 多个 key 以字典排序。
例如, “prefix=abc” 代表指定访问对象前缀为 abc 的对象。
- **HttpHeaders**: 可选项。HTTP 请求头部。当您需验证 url 参数时可填写该参数。其中, key 小写, value 需要进行 URLEncode, 多个 key 以字典排序。
例如, `Host:bucket1-1254000000.cos.ap-beijing.myqcloud.com` 代表该签名可访问账户 1254000000 的存储桶 bucket1 下的指定文件, 对应的 HttpHeaders 为 `host=bucket1-1254000000.cos.ap-beijing.myqcloud.com`。

相关 HTTP 请求参数的详细描述, 请参见 [请求签名](#)。

生成签名及查阅过程参数

单击生成签名, 即可在右侧的结果反馈中, 查看请求签名结果。如下图所示:

COS 签名工具将分别展示生成的最终签名及计算签名过程中的过程参数。相关过程参数的详细介绍, 请参见 [请求签名](#)。

JSON 版本签名工具

输入基础配置信息

1. 单击 [COS 签名工具](#), 进入 **COS 签名工具** 页面。
2. 在基础信息栏中, 填写签名类型、存储桶名称以及当前时间。
 - 签名类型: 选择 JSON API 签名。
 - 存储桶名称: 填写需要访问的存储桶名称, 格式如 bucketname-appid。
 - 当前时间: 目前系统的时间, 是一个符合 Unix Epoch 时间戳规范的数值, 单位为秒。您也可以填入一个指定的时间用于复现在指定时间戳下的签名结果。

! 说明

基础信息中的参数均为必填项。

输入 API 密钥信息

在 API 密钥栏中, 填写您的 API 密钥信息。

! 说明

- 建议使用子账号密钥, 授权遵循最小权限指引, 降低使用风险。子账号密钥获取可参见 [子账号访问密钥管理](#)。
- API 密钥信息为必选项, 填写时请确保该信息的准确性。填写错误将导致您的签名被视为无效签名。

输入 HTTP 参数信息

在 HTTP 参数栏中, 填写相关参数。主要参数如下:

- **ExpiredTime**: 必选项。签名的失效时间, 单位为秒。您可以在当前时间的参数上加上一个有效时长, 得到签名的失效时间。单次签名时, 失效时间必须设置为0。
- **RandomId**: 必选项。无符号10进制整数的随机串。
- **FilePath**: 可选项。标识存储资源的相对路径。格式如: `/[dirname]/[filename]`。
 - 当操作对象为文件夹时, filename 缺省。filename 中要包含文件后缀名。
 - 当 FilePath 为空时, 生成的请求签名将适用于访问存储桶内所有对象。

相关 HTTP 请求参数的详细描述, 请参见 [签名算法](#)。

生成签名及查阅过程参数

单击生成签名, 即可在右侧的结果反馈中, 查看请求签名结果。

COS 签名工具将分别展示生成的最终签名及计算签名过程中的过程参数。相关过程参数的详细介绍, 请参见 [签名算法](#)。

COS 请求工具

最近更新时间：2024-11-15 21:48:52

功能概述

COS 请求工具是腾讯云对象存储（Cloud Object Storage，COS）为您提供的 Web 端调试工具，集成在云 API 3.0 Explorer 平台上，可用于 API 接口的调试工作。

⚠ 注意：

COS 请求工具发送的请求将被真实发送到 COS 的业务服务器，所有操作均等同于真实操作，当您选择 DELETE 类操作时请谨慎操作。

目前 COS 请求工具所支持的 API 版本为 XML API，不支持 JSON API 版本。

- JSON API 是腾讯云 COS 服务在推出 XML API 前为用户提供接入使用 COS 的 API 接口，接口与标准 XML 的 API 底层架构相同，数据互通，可以交叉使用，但是接口不兼容。
- XML API 具有更加丰富的功能和特性，腾讯云对象存储强烈建议您升级使用 XML API 版本。

注意事项

当您单击**发送请求**，确认将已填写必填参数的请求发送至 COS 服务端时，COS 将对您的存储桶和对象进行相应的操作，该操作无法撤回，无法回滚，请您谨慎操作。

工具地址

单击进入 [COS 请求工具](#)。

使用方法

在界面上方搜索**对象存储**产品，选择所需的 API 接口，填写该接口下相应的参数，单击**发送请求**后获取相应的请求响应结果。

COS 请求工具的整体页面，从左至右依次是接口栏、参数栏和结果栏。您可以在不同的栏目执行相应的操作，最终在结果栏发送请求并获取响应结果和相关的过程参数信息。

有关 COS 请求工具的详细操作请查看如下步骤。

1. 选择对象存储产品

在界面上方搜索**对象存储**产品，将可以看到与 COS 相关的 API 接口。

💡 说明：

COS 请求工具集成在云 API 3.0 平台，该平台上搭载着其他腾讯云产品的 API 调试工具，您同样可以根据您的需要在该平台上选择其他产品并调试相应的接口。

2. 选择需要调试的 API 接口

您可以根据您的需求选择相应的 API 接口进行调试。API 接口栏中展示了 COS 相关的多种 API 接口：例如 Service 接口、Bucket 类接口、Object 类接口等。

- 在 Service 类接口中，例如 GET Service。此 API 可列出您账号下的所有存储桶信息，您需要输入您的 API 密钥信息。如果您需要获取您账户在指定区域内的存储桶信息，可在参数栏中选择对应的地域。如需了解更多该 API 的详细信息，请参见 [GET Service](#) 文档。
- 在 Bucket 类接口中，包含对存储桶进行操作的 API 接口，例如 PUT Bucket lifecycle。如需了解更多的 Bucket 类接口，请参见 [Bucket 接口](#)。
- 在 Object 类接口中，包含各个可对对象进行操作的 API 接口，例如 PUT Object。如需了解更多的 Object 类接口，请参见 [Object 接口](#)。

更多接口介绍，请参见 [操作列表](#)。

3. 输入 API 所需的参数信息

参数栏将展示您所选的 API 接口的必填参数。有关 COS 的各个 API 接口的参数信息，请参见 [API 文档](#) 了解。

API 密钥信息是在调用 API 接口这一环节中必填的参数。当您使用 API 接口对您的存储桶或者对象等资源进行操作时，您需要输入 API 密钥信息，用于授权此次 API 请求操作。建议使用子账号密钥，授权遵循最小权限指引，降低使用风险。子账号密钥获取可参见 [子账号访问密钥管理](#)。

说明：

对于每一个 API 接口，COS 请求工具在每一个必填的参数项后均加上了红色星号标识以提醒您该参数项为必填项。您也可以勾选**只看必填参数**，以减少参数栏上非必填的参数显示。

4. 发送请求及查看响应结果

当您选中 API，填写好相应的参数后，在**在线调用**选项卡中单击**发送请求**，此时系统会将您的请求发送到服务器，根据您的请求操作您的存储桶或者对象资源。

注意：

COS 请求工具发送的请求将真实发送到 COS 的业务服务器，所有操作均等同于真实操作，当您选择 **DELETE** 类操作时请谨慎操作。

发送请求后，返回的结果和请求参数将会显示在结果栏下方。在结果栏下方，您可以在**请求参数**中查看您的 HTTP 请求体信息，在**响应结果**中查看本次请求的响应体信息，在**签名过程**中查看本次请求中涉及的签名及其生成流程，同时系统会在 **Curl** 中为您提供 Curl 调用的语句。

示例：

以 GET Object 为例，当发送请求以获取一份名为 0001.txt 的文件时，**请求参数**中将展示您本次请求的请求参数信息，本次请求的请求示例如下：

```
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=*****&q-sign-time=1543398166;1543405366&q-key-time=1543398166;1543405366&q-header-list=host&q-url-param-list=&q-signature=*****
```

首行展示的是您的 HTTP Verb 及访问的链接，次行展示的是访问的域名，最后一行展示的是本次请求的签名信息。对于 PUT 类的请求，其请求头部信息较为复杂，但同样存在一些公共请求头部。有关公共请求头部的信息，请参见 [公共请求头部](#)。

签名过程中可以查看本次请求中涉及的签名及其生成流程。有关签名算法的详细介绍，请参见 [请求签名](#) 详细了解，如果您需要对请求签名进行生成和调试，建议您使用 [COS 签名工具](#)。

COS 返回的响应结果如下：

```
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-hash-crc64ecma:12870159134757904019
x-cos-request-id: NWJmZTYzMtlfOWUxYzBiMDlfOTA4NF8yMWI2****
x-cos-version-id: MTg0NDY3NDI1MzAzODkyMjU****
hello!
```

首行 200 OK 代表该请求返回的状态码信息，如请求失败，则会返回相应的错误码，有关错误码详细信息，请参见 [错误码](#) 文档。此后为响应头部信息，不同 API 的响应体均有所差异，但存在一些公共响应头部信息。有关公共响应头部的详细信息，请参见 [公共响应头部](#)。

COS 策略生成工具

最近更新时间：2024-10-23 15:03:12

功能概述

COS 策略生成工具是腾讯云对象存储（Cloud Object Storage，COS）为用户提供的 Web 端工具，支持图形化配置，一键生成 JSON 格式的策略。

针对不同策略类型，在填入指定参数后，可以一键生成结构化的JSON策略语法，便于后端处理。目前支持三种策略类型的JSON，分别是用户策略、存储桶策略、STS策略。

关于各策略类型的概述和相关信息，请参见 [用户策略（CAM策略）](#)、[存储桶策略（Bucket Policy）](#)、[使用临时密钥访问COS](#)。

注意事项

- 本策略生成工具不识别和提示您填写的错误参数。若您生成的JSON格式策略需要实际生效，请务必按照您的业务需求进行准确填写。
- 本策略生成工具将识别您填写的不合法参数，并进行相应的提示。但仍可能生成一个无法实际生效的策略。
- 本策略生成工具将识别及提示您尚未填写的必填参数项，且如果您未完成必填参数项的填写，您将无法生成JSON格式的策略。

工具地址

1. 您可以单击 [COS策略生成工具](#) 直接进入该页面。
2. 通过 [对象存储控制台](#) 访问该页面，详情可参考 [操作步骤](#)。

适用场景

COS 策略生成工具适用于以下情况：

- 熟悉用户策略、存储桶策略、STS 策略，但不熟悉JSON语法的用户。
- 需要通过 [API](#) 或 [SDK](#) 添加用户策略、存储桶策略、STS 策略的用户。
- 需要批量添加用户策略、存储桶策略、STS 策略的用户。

用户策略（CAM策略）

用户可以在 [CAM](#) 中，对于主账号名下的不同类型用户，授予不同的权限。这些权限通过访问策略语言描述，并以用户为出发点进行授权，因此被称为[用户策略](#)。更多内容可参考 [用户策略](#) 文档。

❗ 说明：

- 用户策略需要撰写完成后，再对子用户、用户组或角色执行关联操作。
- 用户策略不支持将操作和资源权限授予匿名用户。

当您关心用户能做什么，推荐用户策略时，可通过查找CAM用户，并检查其所属用户组的权限来了解用户能做什么。推荐场景有：

- 要配置对象存储（Cloud Object Storage，COS）服务级权限时，例如创建桶（PutBucket）、列举桶（GetService）。
- 需要使用主账号下所有COS桶和对象。
- 要对主账号下的大量CAM用户授予相同权限。

存储桶策略（Bucket Policy）

存储桶策略作用于配置的存储桶和存储桶内对象，您可以通过存储桶策略为CAM子账号、其他主账号、甚至匿名用户授权存储桶及存储桶内对象的操作权限。更多内容可参考 [存储桶策略](#) 文档。

当您关心谁能访问这个COS桶时，推荐使用存储桶策略。可通过查找存储桶，并检查存储桶策略来了解谁能访问。推荐场景有：

- 针对某个存储桶进行授权。
- 相比ACL，存储桶策略更加灵活。
- 相比用户策略，存储桶策略支持跨账号授权及匿名用户授权。

STS策略

STS 策略与临时密钥相关。临时密钥是由 [安全凭证服务（Security Token Service,STS）](#) 提供的临时访问凭证，在申请临时密钥过程中，可以通过设置 Policy 参数（即STS策略），限制操作和资源，为临时密钥增加临时策略约束使用者的权限。更多内容可参考 [使用临时密钥访问COS](#)、[临时密钥生成及使用指引](#) 和 [用于前端直传 COS 的临时密钥安全指引](#) 文档。

 注意：

申请临时密钥之前，您必须拥有访问管理（Cloud Access Management，CAM）用户（腾讯云主账号或子账号）。

临时密钥的推荐场景有：

- 授权第三方临时访问COS。例如，用户开发了客户端 App，将数据存储在 COS 存储桶上，此时将永久密钥直接存放在APP客户端不安全，但又需要授予客户端上传、下载权限。
- 前端数据直传等临时授权。可参考[Web 端直传实践](#)、[小程序直传实践](#)、[移动应用直传实践](#)、[Flutter 直传实践](#)、[uni-app 直传实践](#)使用临时密钥。

操作步骤

1. 登录 [对象存储控制台](#)。
2. 在左侧导航栏中，单击常用工具 > 策略生成工具。
3. 在 [策略生成工具](#) 页面，根据所选的策略类型，按以下说明选择填写各项参数。

 注意：

- 当您在授权时，请务必根据业务需要，按照 [最小权限原则](#) 进行授权，限定用户执行受限的操作（如仅授权读操作），访问指定前缀的资源。
- 若您直接授予其他用户所有资源（resource:*），或者所有操作（action:*）权限，则可能会因权限范围过大导致预期外的越权操作，引起数据安全风险。

CAM策略

策略生成工具

策略生成工具

请求工具

诊断工具

当您授权的时候，建议严格遵循最小权限原则，限定用户执行受限的操作（如仅授权读操作），访问指定前缀的资源，避免授予过大的权限，导致预期外的越权操作，引起数据安全风险。

策略类型：☒ CAM 策略 ☐ 存储桶策略 ☐ STS 策略

效力：☒ 允许 ☐ 禁止

资源路径

地域	存储桶名称	具体路径	操作
		当前列表为空	添加操作

操作

操作名称	操作
	当前列表为空

条件

条件名	条件运算符	条件值	操作
		当前列表为空	添加条件

[生成策略](#)

结果反馈

[一键清除](#)

JSON格式的策略

```
{}

```

1. 在当前页面，选择策略类型为 **CAM策略**。
2. 参数字段说明如下，关于配置项的更多说明请参见 [访问管理授权指南](#)、[访问管理策略语法结构](#) 和 [访问策略语言概述](#)。

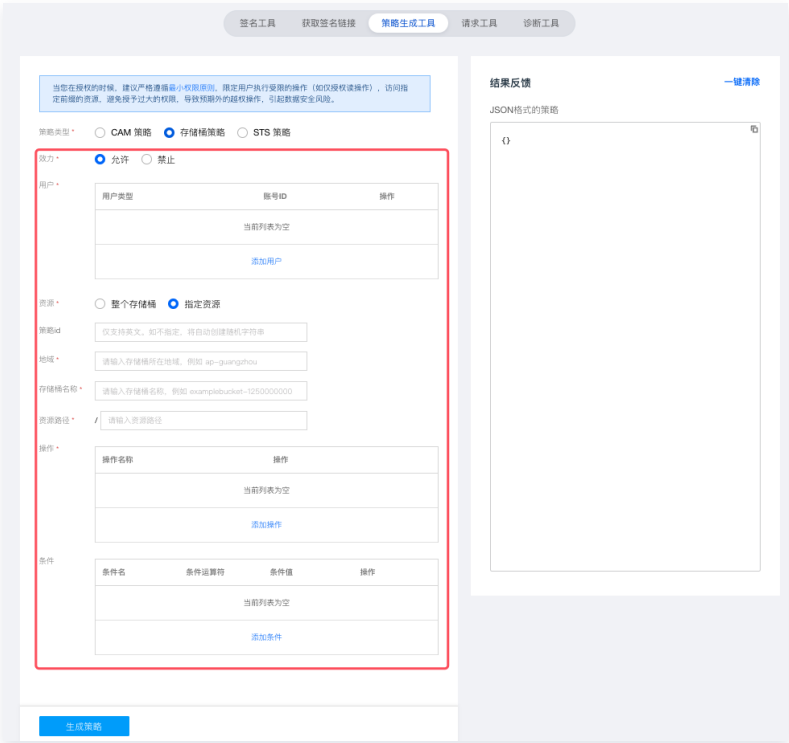
参数名称	是否必填	字段名	说明
效力	是	Effect	支持选择“允许”或“拒绝”，对应策略语法中的“Allow”和“Deny”。

版权所有：腾讯云计算（北京）有限责任公司

第207 共217页

资源路径	是	Resource	资源路由由 地域、存储桶名称、具体路径 组成，支持添加整个存储桶或者指定资源 ● 整个存储桶：若您希望配置将资源范围指定为整个存储桶，仅需填写地域、存储桶名称。 ● 指定资源：若您希望将资源范围限定到指定文件夹，需要指定地域、存储桶名称、具体路径。 ❗ 说明： 地域、存储桶名称和具体路径等信息，可参考当前表格的对应参数说明进行填写。
地域	是	/	存储桶的所在地域，例如 ap-beijing。关于地域的更多信息请参见 地域和访问域名 。
存储桶名称	是	/	存储桶名称由自定义名称和 APPID 两部分构成，例如 examplebucket-1250000000。关于存储桶名称和 APPID 的更多信息请参见 存储桶命名规范 、 APPID 、 对象存储API术语信息 。
具体路径	是	/	具体的资源路径。 ● 若您不指定具体路径，默认为 “*”。 ● 若您希望指定具体路径，需要填写。例如： ○ 需要指定存储桶中的/folder/文件夹下所有对象，填写 folder/* ○ 需要指定存储桶中的/folder/exampleobject对象，填写 folder/exampleobject ○ 需要指定存储桶中的所有以.txt为后缀对象，填写 *.txt
操作	是	Action	添加、删除您需要授权的操作。关于操作的更多信息请参见 操作列表 。
条件	否	Condition	授予权限时指定条件，例如限制用户来访 IP。更多条件说明请参见 条件键说明及使用示例 。

存储桶策略



1. 在当前页面，选择策略类型为 **存储桶策略**。
2. 参数字段说明如下，关于配置项的更多说明请参见 [添加存储桶策略](#) 和 [访问策略语言概述](#)。

参数名称	是否必填	字段名	说明
效力	是	Effect	支持选择“允许”或“拒绝”，对应策略语法中的“Allow”和“Deny”。
用户	是	Principal	支持添加、删除被授权用户，包括所有用户(*)、主账号、子账号和云服务，配置项说明如下： 所有用户：若您希望为匿名用户开放操作权限，可以选择此项。配置后，请求对象时无需携带签名，所有用户可直接通过链接访问对象，存在数据泄露风险，请谨慎设置。 主账号：若您希望为指定主账号开放操作权限，可以选择此项。选择后需要进一步指定具体主账号 UIN。 子账号：若您希望为指定子账号开放操作权限，可以选择此项。选择后需要进一步指定具体子账号 UIN。 云服务：若您希望为指定云服务开放操作权限，可以选择此项。选择后需要进一步指定具体云服务。
资源	是	Resource	资源用六段式描述，支持添加整个存储桶或者指定资源，配置项说明如下： 整个存储桶：若您希望配置存储桶配置相关的权限，或者将资源范围指定为整个存储桶，可以选择此项。 指定资源：若您希望将资源范围限定到指定文件夹，可以选择此项。选择后，需要在“资源路径”进一步指定具体的目录。 说明： 选择后需要填写地域、主账号UIN、存储桶名称和资源路径等信息，可参考当前表格的对应参数说明进行填写。
策略 ID	否	Sid	当前策略规则的 ID，仅支持英文。如不显式指定该字段，将自动创建随机字符串。

地域	是	/	存储桶的所在地域，例如 ap-beijing。关于地域的更多信息请参见 地域和访问域名 。
主账号 UIN	是	/	若您选择的用户为“子账号”或者“云服务”，需要填写主账号UIN。关于UIN的概念请参见 UIN 。
存储桶名称	是	/	存储桶名称由自定义名称和 APPID 两部分构成，例如 examplebucket-1250000000。关于存储桶名称和 APPID 的更多信息请参见 存储桶命名规范 、 APPID 、 对象存储API术语信息 。
资源路径	是	/	具体的资源路径。 - 若您选择的资源为“整个存储桶”，资源路径默认为整个桶。 - 若您选择的资源为“指定资源”，需要填写具体的资源路径。例如： - 需要指定存储桶中的/folder/文件夹下所有对象，填写 folder/* - 需要指定存储桶中的/folder/exampleobject对象，填写 folder/exampleobject - 需要指定存储桶中的所有以.txt为后缀对象，填写 *.txt
操作	是	Action	添加、删除您需要授权的操作。关于操作的更多信息请参见 操作列表 。 ⚠ 注意： CDN 服务账号仅支持 GetObject、HeadObject、OptionsObject操作。
条件	否	Condition	授予权限时指定条件，例如限制用户来访 IP。更多条件说明请参见 条件键说明及使用示例 。

STS策略

签名工具 获取签名链接 策略生成工具 请求工具 诊断工具

当前在授权的时候，建议严格遵循最小权限原则，限定用户执行具体的操作（如仅授权读操作），访问指定范围的资源，避免授予过大的权限，导致额外的越权操作，引起数据安全风险。

策略类型 ☐ CAM 策略 ☐ 存储桶策略 ☒ STS 策略

效力 ☒ 允许 ☐ 禁止

资源路径

地域	存储桶名称	资源路径	操作
当前列表为空			
添加操作			

操作

操作名称	操作
当前列表为空	
添加操作	

条件

条件名	条件运算符	条件值	操作
当前列表为空			
添加条件			

[生成策略](#)

结果反馈 [一键清除](#)

JSON格式的策略

```
{}
```

1. 在当前页面，选择策略类型为 **STS策略**。
2. 参数字段说明如下，关于配置项的更多说明请参见 [获取联合身份临时访问凭证](#) 和 [访问策略语言概述](#)。

参数名称	是否必填	字段名	说明
------	------	-----	----

效力	是	Effect	支持选择“允许”或“拒绝”，对应策略语法中的“Allow”和“Deny”。
资源路径	是	Resource	资源路由由 地域、存储桶名称、具体路径 组成，支持添加整个存储桶或者指定资源 整个存储桶：若您希望配置将资源范围指定为整个存储桶，仅需填写地域、存储桶名称。 指定资源：若您希望将资源范围限定到指定文件夹，需要指定地域、存储桶名称、具体路径。 说明： 地域、存储桶名称和具体路径等信息，可参考当前表格的对应参数说明进行填写。
地域	是	/	存储桶的所在地域，例如 ap-beijing。关于地域的更多信息请参见 地域和访问域名 。
存储桶名称	是	/	存储桶名称由自定义名称和 APPID 两部分构成，例如 examplebucket-1250000000。关于存储桶名称和 APPID 的更多信息请参见 存储桶命名规范 、 APPID 、 对象存储API术语信息 。
具体路径	是	/	具体的资源路径。 - 若您不指定具体路径，默认为“*”。 - 若您希望指定具体路径，需要填写。例如： - 需要指定存储桶中的/folder/文件夹下所有对象，填写 folder/* - 需要指定存储桶中的/folder/exampleobject对象，填写 folder/exampleobject - 需要指定存储桶中的所有以.txt为后缀对象，填写 *.txt
操作	是	Action	添加、删除您需要授权的操作。关于操作的更多信息请参见 操作列表 。
条件	否	Condition	授予权限时指定条件，例如限制用户来访 IP。更多条件说明请参见 条件键说明及使用示例 。

4. 单击生成策略。

右侧的结果反馈中将展示JSON格式的策略。该工具支持多次生成策略，并且提供一键复制和一键清除的能力。

签名工具 获取签名链接 策略生成工具 请求工具 诊断工具

您当前在授权的时候，建议严格遵循最小权限原则，限定用户执行受限的操作（如仅授权读操作），访问指定数据的资源，避免授予过大的权限，导致额外的越权操作，引起数据安全风险。

策略类型

☐ CAM 策略

☐ 存储桶策略

☒ STS 策略

效力

☒ 允许

☐ 禁止

资源路径

地域

ap-shanghai(上海)

存储桶名称

example-bucket

具体路径

/ test1

操作

删除

添加操作

操作

操作名称

写操作

操作

删除

添加操作

条件

条件名

IP

条件运算符

等于

条件值

192.168.1.1

操作

删除

添加条件

生成策略

结果反馈 一键清除

JSON格式的策略

```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "action": [
        "name/cos:PutObject",
        "name/cos:PostObject",
        "name/cos:DeleteObject",
        "name/cos:InitiateMultipartUpload",
        "name/cos:UploadPart",
        "name/cos:CompleteMultipartUpload",
        "name/cos:AbortMultipartUpload"
      ],
      "resource": [
        "qcs::cos:ap-shanghai:uid/bucket:prefix//bucket/example/test1"
      ],
      "condition": {
        "ip_equal": {
          "qcs:ip": [
            "192.168.1.1"
          ]
        }
      }
    ]
  ]
}
```

自助诊断工具

最近更新时间：2023-02-10 15:22:50

功能概述

对象存储（Cloud Object Storage，COS）自助诊断工具是腾讯云对象存储为用户提供的 Web 工具，可用于错误请求的自助诊断排查。您可以在工具页面上填入请求的 RequestId（获取方法可参见[获取 RequestId 操作指引](#)），单击诊断后，工具会对请求进行智能诊断，并展示该请求的基本信息以及帮助指引和诊断提示，帮助您快速定位使用 COS API 的错误问题。

工具地址

点此进入 [COS 自助诊断工具](#)。

使用方法

- 单击 [COS 自助诊断工具](#)，进入“COS 自助诊断工具”页面。
- 在顶部的 RequestId 输入框中，输入待诊断的 RequestId，并单击**开始诊断**。

RequestId

NjBiMWYyMTRfMzhhNzcyNjRfNGU2Zi8xYjFhC

开始诊断

- 稍侯片刻，便能看到相应的智能诊断结果。

诊断结果

诊断提示

匿名访问无当前资源的 [cos:GetObject] 权限，请确认 存储桶权限 或 文件权限 是否私有读。

帮助指引

错误码 [AccessDenied]，签名或者权限不正确，拒绝访问。请参考 [请求签名](#)、[ACL 概述](#) 和 [访问 COS 时返回403错误码](#)。

诊断结果是否对您有帮助？如诊断有误，可以 [联系我们](#)。

有帮助

没帮助

请求信息			
状态码	403 Forbidden	请求时间	2021-05-29 15:49:40
错误码	AccessDenied	请求方法	GET

其中，诊断结果分为两部分：诊断结果和请求信息。

- 诊断结果为请求诊断后的建议与帮助，您可以根据建议与帮助快速定位到 COS API 的错误问题。
- 请求信息是该 RequestId 对应的基本信息。

- 对诊断结果进行反馈

您可以单击诊断结果下方的**有帮助**或**没帮助**，将您对此次诊断结果的意见反馈给我们，以便我们继续优化工具。

常见问题

除了利用 RequestId 进行诊断排查外，该工具还提供丰富的常见问题样例以及对应的解决方案，方便您更好地使用 COS API 以及排查相关问题。此外，若您有疑问，您也可以 [联系我们](#)。

版权所有：腾讯云计算（北京）有限责任公司

第212 共217页

注意事项

标准的 COS 请求 RequestId 具有以下几个特征：

1. 以大写字母 N 开头。
2. 长度不得小于30个字符。
3. 遵循 base64字符标准。

文档预览接入体验工具

最近更新时间：2025-02-28 20:22:12

功能概述

文档预览接入体验工具是腾讯云数据万象为您提供的 Web 工具，您可以在工具页面配置文档预览参数、调用高级 API 方法，体验文档预览效果。同时该工具可以根据相应配置项生成代码，助力您快速接入 COS 文档预览。

❗ 说明：

- 文档预览接入体验工具用于文档转 HTML 预览接入调试。目前仅支持公有读文件的调试。
- 目前支持的输入文件类型包含如下格式：
 - 演示文件：pptx、ppt、pot、potx、pps、ppsx、dps、dpt、pptm、potm、ppsm。
 - 文字文件：doc、dot、wps、wpt、docx、dotx、docm、dotm。
 - 表格文件：xls、xlt、et、ett、xlsx、xltx、csv、xlsb、xlsm、xltn、ets。
 - PDF 文件：pdf。

工具地址

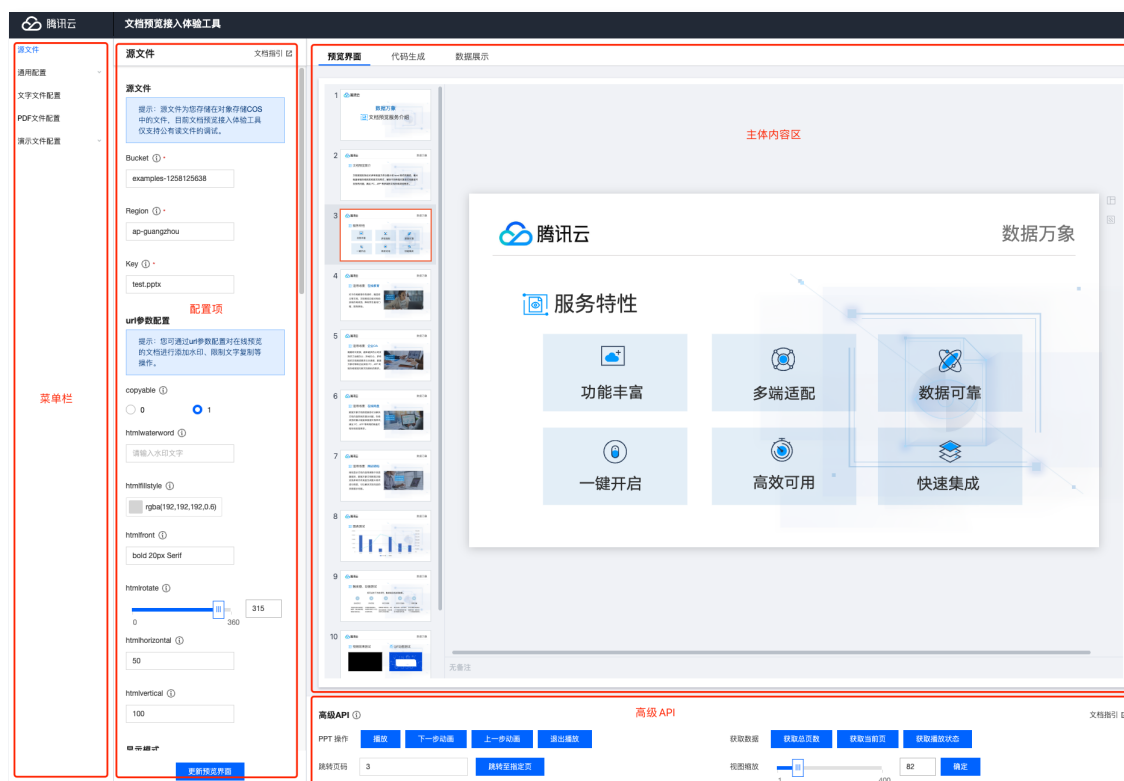
单击进入 [文档预览接入体验工具](#)。

界面介绍

文档预览接入体验工具的整体页面可以分为菜单栏、配置项、高级 API 区、主体内容。

- 菜单栏：配置项菜单，分为源文件、通用配置、文字文件配置、PDF 文件配置、演示文件配置。
- 配置项：您可在配置项中填写文档预览的相关参数。单击**更新预览界面**后，预览界面、代码生成、数据展示页会相应更新。
- 高级 API 区：展示文字文件/ PDF 文件/演示文件/表格文件对应可调用的高级 API 。您可单击页面相应按钮调用对应高级 API 方法。
- 主体内容：包含预览界面页、代码生成页、数据展示页。
 - 预览界面页中可以查看当前配置或调用高级 API 后的文档预览效果。
 - 代码生成页中可以查看根据当前配置项和调用过的高级 API 生成的代码，支持一键复制。
 - 数据展示页中可以支持查看事件监听和部分高级 API 的调用结果。

您可以在不同的栏目执行相应的操作，最终效果会体现在主体内容区。如下图所示：



操作步骤

有关文档预览接入体验工具的详细操作请参见如下步骤：

步骤1：输入源文件

源文件为您存储在对象存储 COS 中的文件，您需在配置项中输入 **Bucket**、**Region**、**Key**。

步骤2：填写配置项并更新预览界面

您可以根据您的需求填写相应配置项。通用配置中的配置项对文字文件、PDF 文件、演示文件、表格文件都生效。文字文件配置只对文字类型文件生效，PDF 文件配置只对 PDF 类型文件生效、演示文件配置只对演示类型文件生效。在填写完配置项后，您需单击**更新预览界面**让配置生效，预览效果和相应代码可在主体内容中查看。

步骤3：调用高级 API

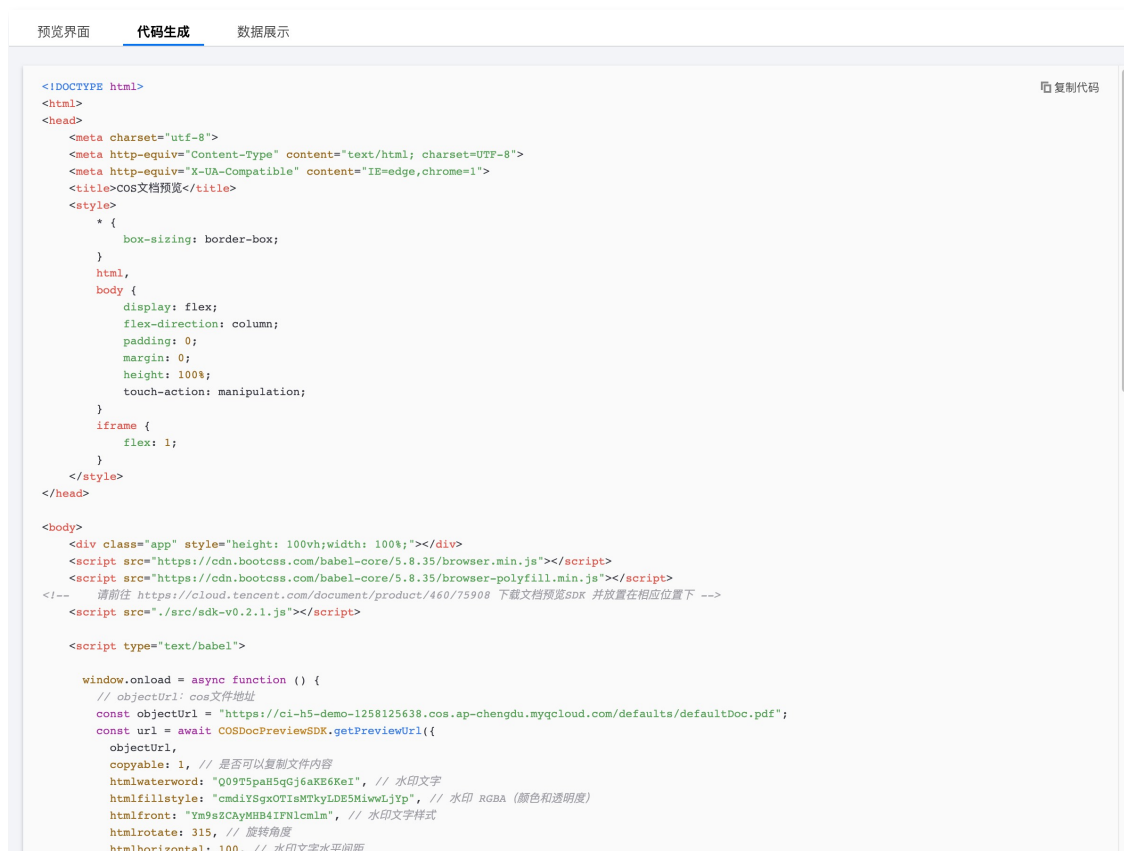
您可单击高级 API 区的按钮调用相应高级 API ，可以在预览界面页查看效果、代码生成页查看调用的高级 API 代码、数据展示页查看相应输出内容。

步骤4：查看结果

在预览界面页中，您可以查看当前配置或调用高级 API 后的文档预览效果。如下图所示：



在代码生成页中，您可以查看根据当前配置项和调用过的高级 API 生成的代码，支持一键复制。如下图所示：



在数据展示页中，您可以查看事件监听和部分高级 API 的调用结果。如下图所示：

预览界面	代码生成	数据展示
操作	结果	获取时间
【 文件-fileOpen 】	打开成功	2022-10-24 20:27:11
【 PDF文件-总页数 】	10	2022-10-24 20:27:14
【 PDF文件-当前页数 】	1	2022-10-24 20:27:15