

腾讯云对象存储

开发者工具

产品文档



腾讯云

【版权声明】

©2013-2017 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【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。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。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模式的承诺或保证。

文档目录

文档声明.....	2
开发者工具.....	4
工具概览	4
环境安装与配置	5
Java 安装与配置	5
Python 安装与配置.....	10
Hadoop 安装与测试	13
COSCMD 工具.....	25
COS Browser 工具	41
迁移工具	43
本地同步工具.....	49
FTP Server 工具.....	54
COSFS 工具.....	60
Hadoop 工具.....	68
HDFS TO COS 工具	76
WeCOS 小程序瘦身工具	80

开发者工具

工具概览

对象存储提供以下开发者工具，供用户使用：

工具	功能说明	版本限制
COSCMD 工具	本工具支持用户使用简单的命令行指令实现对对象的批量上传、下载、删除等操作。	适用于 COS V4、V5 版本
迁移工具	本工具支持从 AWS S3, 阿里云 OSS 和七牛等服务中迁移文件到 COS；同时也支持文件列表迁移，即从一系列给定的 URL 中迁移文件到 COS。	适用于 COS V4 版本
本地同步工具	本工具支持将本地目录下文件、子目录同步到 COS 上。	适用于 COS V4 版本
FTP Server 工具	本工具支持用户使用 FTP 客户端从 COS 上传、下载文件。	适用于 COS V5 版本
COSFS 工具	在 Linux 系统中，使用本工具可以把存储桶挂载到本地文件系统中，通过本地文件系统操作 COS 上的对象。	适用于 COS V4、V5 版本
Hadoop 工具	本工具支持用户使用 Hadoop 处理存储在 COS 上的对象，如 MapReduce，Hive 等。	适用于 COS V4 版本
HDFS TO COS 工具	本工具用于将 HDFS 上的数据拷贝到 COS 上。	适用于 COS V5 版本
WeCOS 小程序瘦身工具	本工具将小程序项目中的图片资源自动上传至 COS 上，并将项目代码中图片资源引用地址替换为 COS 线上地址，移除项目中的图片资源，达到为小程序瘦身的目的。	适用于 COS V4 版本

环境安装与配置

Java 安装与配置

JDK 是 Java 软件开发工具包，本文以 JDK 1.7 和 1.8 版本为例，分别介绍了 Windows 和 Linux 系统下 JDK 的安装与环境配置过程。

Windows

1. 下载 JDK

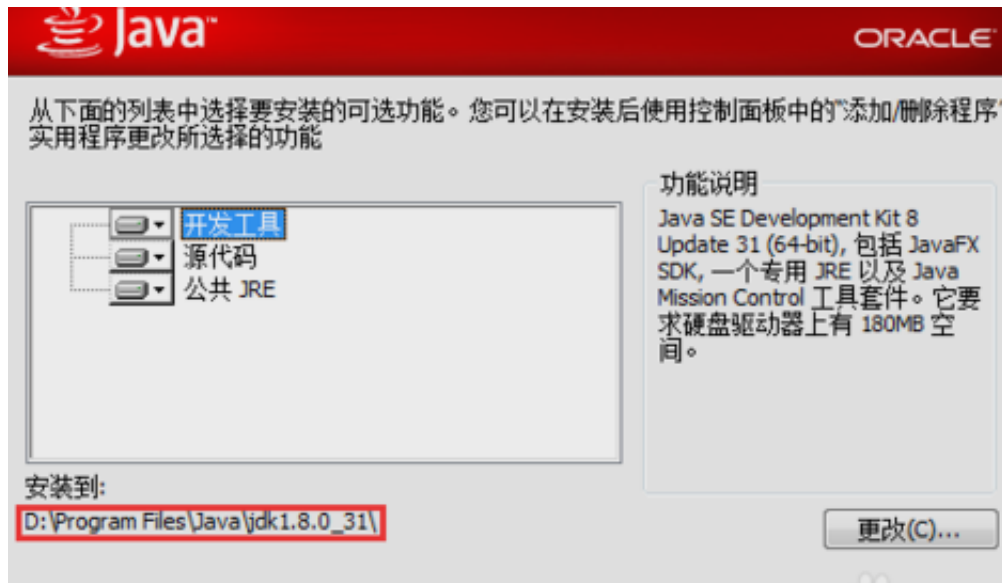
进入 [Oracle 官方网站](#) 下载合适的 JDK 版本，准备安装。

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

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



单击“更改”以将 Java 安装到其他文件夹。

安装到: D:\Program Files\Java\jre1.8.0_31 更改(C)...

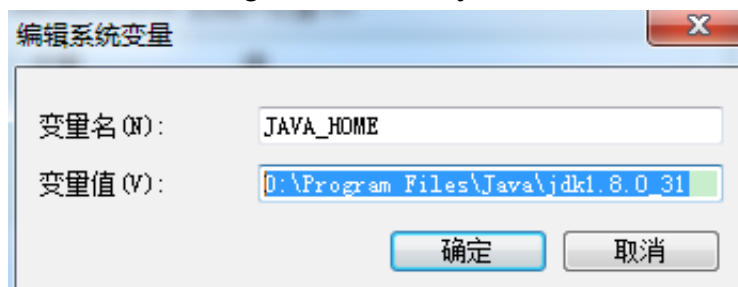
3. 配置

安装完成后，右键单击【计算机】>

单击【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 // 请根据自己的实际安装路径配置



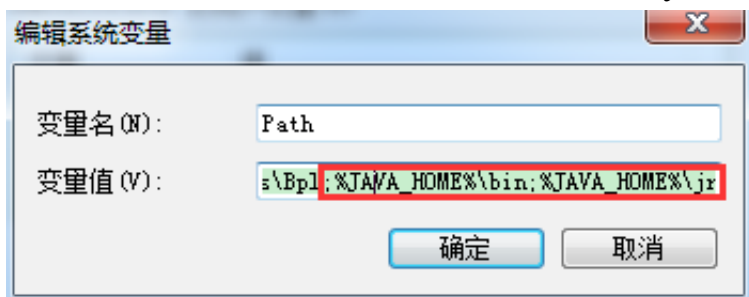
变量名(N)：CLASSPATH

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;



4. 测试

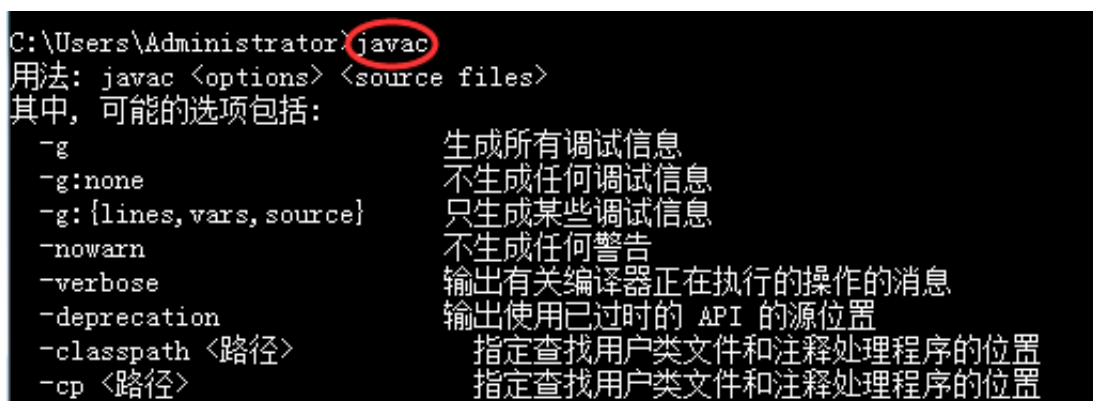
测试配置是否成功：【开始】（或快捷键：Win+R）>【运行】（输入

cmd

）>【确定】（或按 Enter 键），输入命令

javac

并回车。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：这里需要下载 Linux 版本。这里以jdk-8u151-linux-x64.tar.gz为例，你下载的文件可能不是这个版本，这没关系，只要后缀(.tar.gz)一致即可。

2. 创建目录

在

/usr/

目录下创建

java

目录，

```
mkdir /usr/java
```

```
cd /usr/java
```

把下载的文件 jdk-8u151-linux-x64.tar.gz 放在/usr/java/目录下。

3. 解压 JDK

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

4. 设置环境变量

修改 /etc/profile

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。

让修改生效：

```
source /etc/profile
```

5. 测试

```
java -version
```

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

本文档以 Python 2.7 版本为例，详细介绍 Windows 和 Linux 系统下 Python 的安装与环境配置过程。

Windows

1. 下载

进入 [Python 官网](#) 选择合适的版本下载，本示例中我们选择下载 [Python 2.7.13](#) 版本。

2. 安装

下载好 Python 安装包后双击 Python 安装包，按照默认提示，一步步进行安装。

3. 环境变量配置

安装完成后，右键单击【计算机】>单击【属性】>【高级系统设置】>【环境变量】>【系统变量(S)】找到“Path”（没有就新建），并在“变量值”末尾添加 Python 的安装路径：

;C:\Python27

（请更改为您的安装路径），单击【确定】保存。

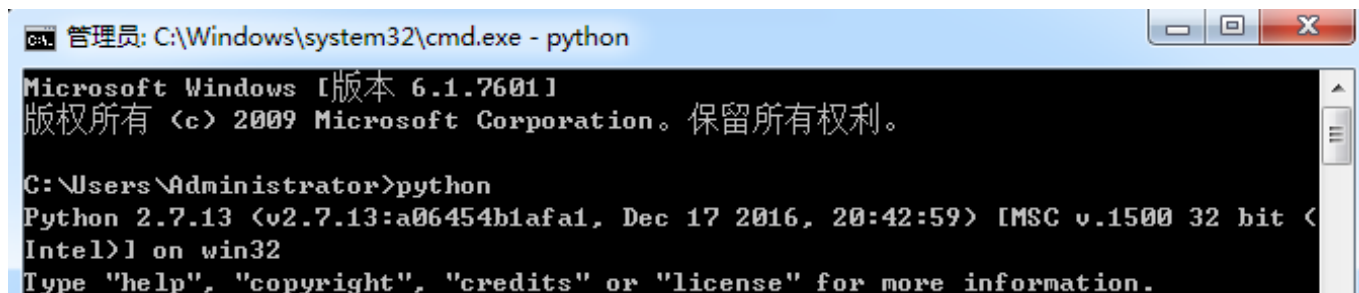


4. 测试配置是否成功

单击【开始】（或快捷键：Win+R）>【运行】（输入

cmd

) > 【确定】（或者按 Enter 键），在弹出的窗口中输入命令 Python 并回车。出现以下信息，说明 Python 2.7 已经安装配置好：



```

管理员: C:\Windows\system32\cmd.exe - python
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>python
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
    
```

Linux

1. 查看 Python 版本

Linux 的 yum 自带 Python，首先查看默认 Python 版本

`python -V`

若已经是 Python 2.7 及以上版本，则忽略以下步骤；否则（此处假设现有版本为 Python 2.6.6），输入以下命令：

`yum groupinstall "Development tools"`

2. 安装编译 Python 需要的组件

`yum install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel`

3. 下载并解压 Python 2.7

```
wgethttps://www.python.org/ftp/python/2.7.12/Python-2.7.12.tar.xz  
tar xf Python-2.7.12.tar.xz
```

4. 编译与安装 Python

```
cd Python-2.7.12 //进入目录  
./configure --prefix=/usr/local  
make && make install //安装  
make clean  
make distclean
```

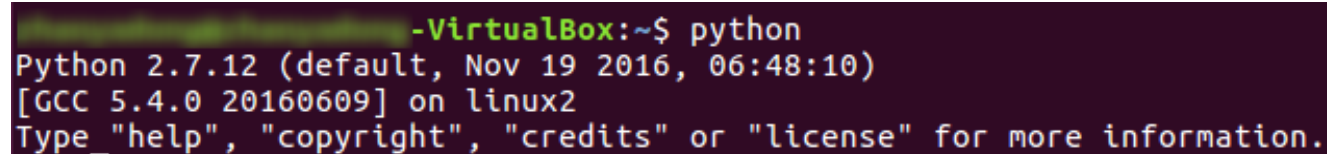
5. 将系统 Python 命令指向 Python 2.7

```
mv /usr/bin/python /usr/bin/python2.6.6  
ln -s /usr/local/bin/python2.7 /usr/bin/python
```

6. 测试配置是否成功

```
python
```

出现以下信息，说明 Python 2.7 已经安装配置好：



```
-VirtualBox:~$ python  
Python 2.7.12 (default, Nov 19 2016, 06:48:10)  
[GCC 5.4.0 20160609] on linux2  
Type "help", "copyright", "credits" or "license" for more information.
```

注意：

如果出现权限的问题，建议在命令前添加 `sudo` 尝试解决。

Hadoop 安装与测试

Hadoop 工具依赖 Hadoop-2.7.2 及以上版本，实现了以腾讯云 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

安装

1. 准备若干台机器。
2. 安装配置系统：[CentOS-7-x86_64-DVD-1611.iso](#)。
3. 安装 Java 环境：[jdk-8u144-linux-x64.tar.gz](#)，具体操作请参见 [Java 安装与配置](#)。
4. 安装 Hadoop-2.7.4 包：[hadoop-2.7.4.tar.gz](#)。

网络配置

使用

```
ifconfig -a
```

查看各台机器的 IP，相互 ping 一下，看是否可以 ping 通，同时记录每台机器的 IP。

配置 CentOS

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-8u144-linux-x64.tar.gz](#) 安装包到

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
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-2.7.4.tar.gz](http://hadoop.apache.org/tarballs/hadoop-2.7.4.tar.gz) 安装包到

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

```
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. 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>
```

7. 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>
<property>
<name>yarn.resourcemanager.address</name>
<value>master:8032</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>master:8030</value>
</property>
```

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

8. 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
```

9. 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/
```

```
[root@VM_06_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
```

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-000000
a          5
dasdada    1
ds         2
qwe        1
ret        1
s          1
v          2
vdfd       1
wqere      1
```

注意：

单机模式与伪分布式模式的操作方法的详细过程可以参考官网：[Hadoop入门](#)。

COSCMD 工具

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。

使用限制

适用于 COS V4、V5 版本；

使用环境

系统环境

Windows 或 Linux 系统

软件依赖

Python 2.7

并装有最新版本的 pip

安装及配置

环境安装与配置详细操作请参考 [Python 安装与配置](#)。

下载与安装

- 手动安装

下载链接：[GitHub 链接](#)

在该项目根目录下使用如下命令安装

```
python setup.py install
```

- pip 安装

执行

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。

- pip 更新

下执行

pip

命令进行更新：

pip install coscmd -U

注意：

无论是在 Linux 还是 Windows 环境下，都可以通过以上的方法安装或更新

使用方法

查看 help

用户可通过

-h

或

--help

命令来查看工具的 help 信息。

coscmd -h //查看当前版本信息

help 信息如下图所示：

```
usage: coscmd [-h] [-d] [-b BUCKET] [-v]
```

```
{config,upload,download,delete,list,info,mget,restore,signurl,createbucket,deletebucket,putobjectacl,
getobjectacl,putbucketacl,getbucketacl}
```

...

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

positional arguments:

```
{config,upload,download,delete,list,info,mget,restore,signurl,createbucket,deletebucket,putobjectacl,
getobjectacl,putbucketacl,getbucketacl}
```

config config your information at first.

upload upload file or directory to COS.

download download file from COS to local.

delete delete file or files on COS
list list files on COS
info get the information of file on COS
mget download big file from COS to local(Recommand)
restore restore
signurl get download url
createbucket create bucket
deletebucket delete bucket
putobjectacl set object acl
getobjectacl get object acl
putbucketacl set bucket acl
getbucketacl get bucket acl

optional arguments:

-h, --help show this help message and exit
-d, --debug debug mode
-b BUCKET, --bucket BUCKET
set bucket
-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，例如：

coscmd upload -h //查看 upload 命令使用方法

配置参数

COSCMD 工具在使用前需要进行参数配置。用户可以通过如下命令来配置：

coscmd config -a <access_id> -s <secret_key> -b <bucket> -r <region> [-m <max_thread>] [-p <parts_size>]

上述示例中使用

<>

的字段为必选参数，使用

[]

的字段为可选参数。其中：

名称	描述	有效值
secret_id	必选参数，APPID 对应的密钥 ID，可从控制台获取，参考 基本概念 。	字符串
secret_key	必选参数，APPID 对应的密钥 Key，可从控制台获取，参考 基本概念 。	字符串
bucket	必选参数，指定的存储桶名称，bucket 的命名规则为{name}-{appid}，参考 创建存储桶 。	字符串
region	必选参数，存储桶所在地域。参考 可用地域 。	字符串
max_thread	可选参数，多线程上传时的最大线程数（默认为 5），有效值：1~10	数字
parts_size	可选参数，分块上传的单块大小（单位为 MB，默认为 1MB），有效值：1~10	数字

也可以直接编辑

~/cos.conf

文件（对于windows环境下，该文件是位于

我的文档

下的一个隐藏文件）。

配置完成之后的

.cos.conf

文件内容示例如下所示：

```
[common]
secret_id = AChT4ThiXAbpBDEFGhT4ThiXAbpHIJK
secret_key = WE54wreefvds3462refgwewerewr
bucket = ABC-1234567890
region = cn-south
max_thread = 5
part_size = 1
```

指定 Bucket 的命令

通过

-b <bucket> 指定bucket

可指定特定 Bucket。Bucket 的命名规则为

{name}-{appid}

，此处填写的存储桶名称必须为此格式。

coscmd -b <bucket> method ... //命令格式

coscmd -b AAA-12345567 upload a.txt b.txt //操作示例-上传文件

coscmd -b AAA-12344567 createbucket //操作示例-创建 Bucket

创建 Bucket

建议配合

-b <bucket> 指定bucket

使用

coscmd -b <bucket> createbucket //命令格式

coscmd createbucket //操作示例

coscmd -b AAA-12344567 createbucket //操作示例

删除 Bucket

建议配合

-b <bucket> 指定bucket

使用

coscmd -b <bucket> deletebucket //命令格式

coscmd createbucket //操作示例

coscmd -b AAA-12344567 deletebucket //操作示例

上传文件或文件夹

- 上传文件命令如下：

```
coscmd upload <localpath> <cospath> //命令格式  
coscmd upload /home/aaa/123.txt bbb/123.txt //操作示例  
coscmd upload /home/aaa/123.txt bbb/ //操作示例
```

- 上传文件夹命令如下：

```
coscmd upload -r <localpath> <cospath> //命令格式  
coscmd upload -r /home/aaa/ bbb/aaa //操作示例  
coscmd upload -r /home/aaa/ bbb/ //操作示例  
coscmd upload -r /home/aaa/ / //上传到bucket根目录
```

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）。

注意：

- 上传文件时需要将 COS 上的路径包括文件（夹）的名称补全。
- COSCMD 支持大文件断点上传功能。当分片上传大文件失败时，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。

下载文件或文件夹

- 下载文件命令如下：

```
coscmd download <cospath> <localpath> //命令格式
coscmd download bbb/123.txt /home/aaa/111.txt //操作示例
coscmd download bbb/123.txt /home/aaa/ //操作示例
```

- 下载文件夹命令如下：

```
coscmd download-r <cospath> <localpath> //命令格式
coscmd download -r /home/aaa/ bbb/aaa //操作示例
coscmd download -r /home/aaa/ bbb/ //操作示例
coscmd download -r / bbb/aaa //下载当前bucket根目录下所有的文件
```

请将

<>

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

注意：

- 若本地存在同名文件，则会下载失败。使用

-f

参数覆盖本地文件

- 将以上命令中的

download

替换为

mget

，则可以使用分块下载，在带宽足够的条件下速度会提升 2~3 倍。

删除文件或文件夹

- 删除文件命令如下：

```
coscmd delete <cospath> //命令格式  
coscmd delete bbb/123.txt //操作示例
```

- 删除文件夹命令如下：

```
coscmd delete -r <cospath> //命令格式  
coscmd delete -r bbb/ //操作示例  
coscmd delete -r / //操作示例
```

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。

注意：

批量删除需要输入确定，使用

-f

参数跳过确认

打印文件列表

- 打印命令如下：

```
coscmd list <cospath> //命令格式  
coscmd list -a //操作示例  
coscmd list bbb/123.txt -r -n 10 //操作示例
```

请将

<>

中的参数替换为您需要打印文件列表的 COS 上文件的路径（ cospath ）。

- 使用

-a

打印全部文件

- 使用

-r

递归打印

- 使用

-n num

设置打印数量的最大值

注意：

<cospath>

为空时默认打印当前 Bucket 根目录。

显示文件信息

命令如下：

```
coscmd info <cospath> //命令格式  
coscmd info bbb/123.txt //操作示例
```

请将

<>

中的参数替换为您需要显示的 COS 上文件的路径（cospath）。

获取带签名的下载 URL

命令如下：

```
coscmd sigurl<cospath> //命令格式  
coscmd signurl bbb/123.txt //操作示例  
coscmd signurl bbb/123.txt -t 100//操作示例
```

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

设置打印签名的有效时间（单位为秒）。

设置访问控制（ACL）

- 使用如下命令设置 Bucket 的访问控制：

```
coscmd putbucketacl [--grant-read GRANT_READ] [--grant-write GRANT_WRITE] [--grant-full-control GRANT_FULL_CONTROL] //命令格式
```

```
coscmd putbucketacl --grant-read 12345678,12345678/11111 --grant-write anyone --grant-full-control 12345678/22222 //操作示例
```

- 使用如下命令设置 Object 的访问控制：

```
coscmd putbucketacl [--grant-read GRANT_READ] [--grant-write GRANT_WRITE] [--grant-full-control GRANT_FULL_CONTROL] <cospath> //命令格式
```

```
coscmd putbucketacl --grant-read 12345678,12345678/11111 --grant-write anyone --grant-full-control 12345678/22222 aaa/aaa.txt //操作示例
```

ACL 设置指南

- --grant-read

代表读的权限。

- --grant-write

代表写的权限。

- `--grant-full-control`

代表读写的权限。

- `GRANT_READ / GRANT_WRITE / GRANT_FULL_CONTROL`

代表被赋权的账号。

- 若赋权根帐号，请使用 `rootid` 的形式；
- 若赋权子账户，请使用 `rootid/subid` 的形式；
- 若需要对所有人赋权，请使用 `anyone` 的形式。
- 同时赋权的多个帐号用逗号

,

隔开。

请将参数替换为您所需要删除的 COS 上文件的路径（`cospath`）。

获取访问控制（ACL）

- 使用如下命令设置 Bucket 的访问控制：

`coscmd getbucketacl //命令格式`

`coscmd getbucketacl //操作示例`

- 使用如下命令设置 Object 的访问控制：

`coscmd putbucketacl <cospath> //命令格式`

```
coscmd getobjectacl aaa/aaa.txt //操作示例
```

恢复归档文件

命令如下：

```
coscmd restore <cospath> //命令格式
```

```
coscmd restore a.txt -d 3 -t Expedited//操作示例
```

```
coscmd restore a.txt -d 3 -t Bulk///操作示例
```

请将

<>

中的参数替换为您需要打印文件列表的 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

具体复原过程类型，枚举值：Expedited，Standard，Bulk；默认值：Standard

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：

//显示upload的详细操作信息

coscmd -d upload <localpath> <cospath> //命令格式

coscmd -d upload /home/aaa/123.txt bbb/123.txt //操作示例

COS Browser 工具

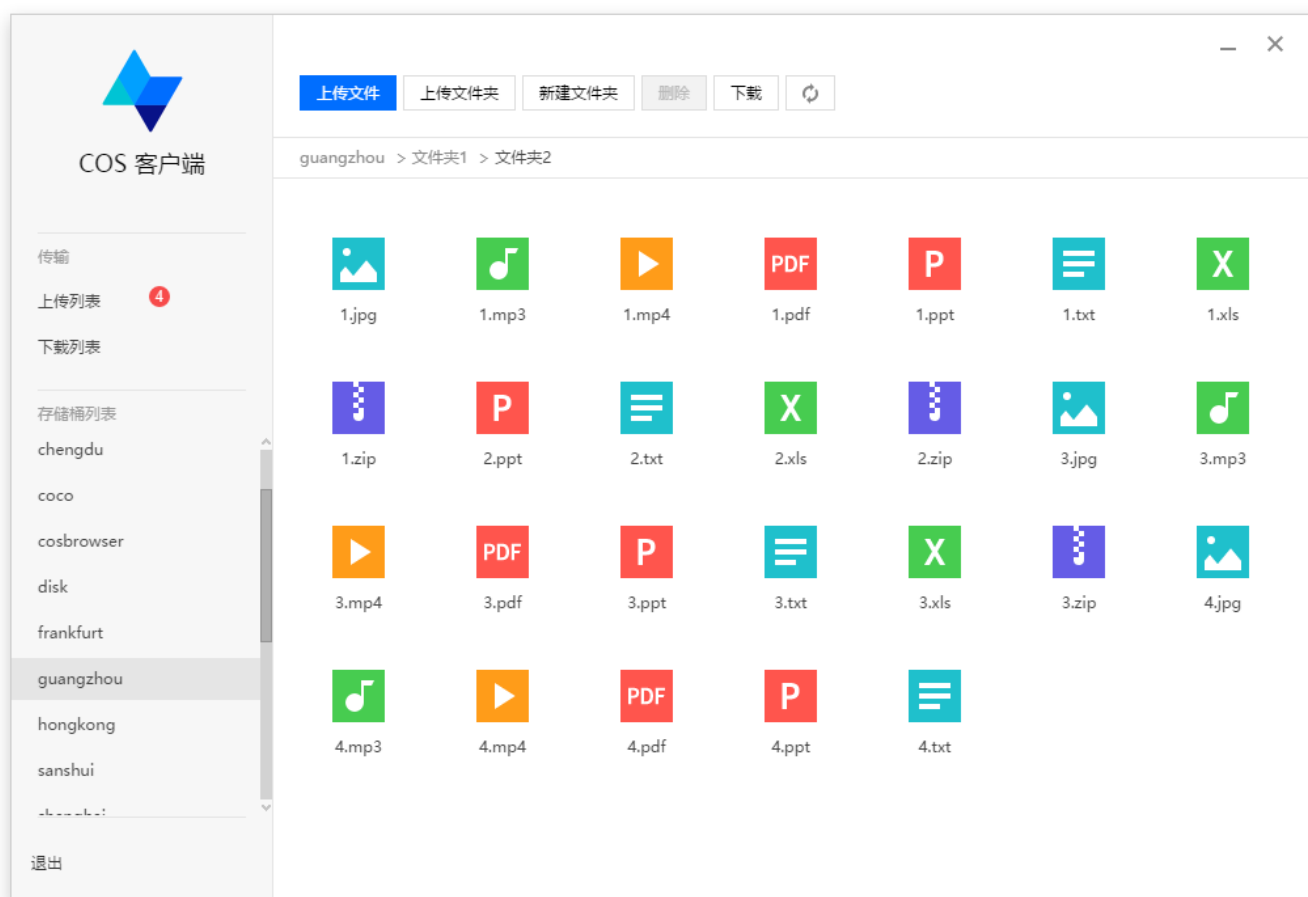
COS Browser (COS 桌面版)

COS Browser , 用户可以可视化、方便地进行数据的上传、下载等操作。

使用环境

Windows 32/64 位系统均可

软件界面



下载与安装

下载安装版 (依赖 .Net Framework 4.5 , 下载后双击安装) :

- [Window 64位 安装版](#)
- [Window 32位 安装版](#)

下载解压版（下载后解压文件，双击 cosbrowser.exe 即可运行）：

- [Window 64位 解压版](#)
- [Window 32位 解压版](#)

使用方法

用户可使用云 API 密钥 SecretId 和 SecretKey 进行登录，该密钥可在 [访问管理控制台](#) 获取，登录后会保留登录信息。

目前支持批量上传、下载、删除功能。拖拽上传下载暂不支持。

迁移工具

功能说明

本迁移工具支持从 AWS S3，阿里云 OSS 和七牛云等服务迁移文件到对象存储 COS；同时也支持文件列表迁移，即从一系列给定的 URL 中迁移文件到对象存储 COS。

使用限制

只适用于 COS V4 版本

使用环境

系统环境

Linux 或 Mac 系统

软件依赖

Python 2.7.x，同时应该安装 pip，gcc 与 python-dev。

安装及配置

1. Python 2.7.x 安装请参考 [Python 安装与配置](#)。
2. 安装 pip，gcc 与 python-dev：

```
sudo yum install python-pip python-devel gcc
```

使用方法

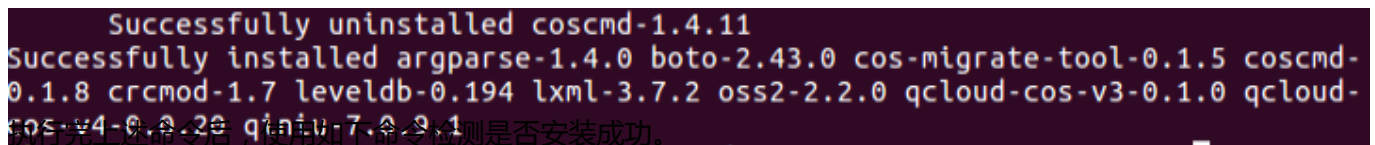
获取工具

[Github 地址](#)

安装方法

推荐使用 pip 安装，安装 pip 的方法，可以参考 [pip 官网](#) 或者使用 apt/yum 等包管理工具安装 python-pip 包。

```
pip install -U cos_migrate_tool
```

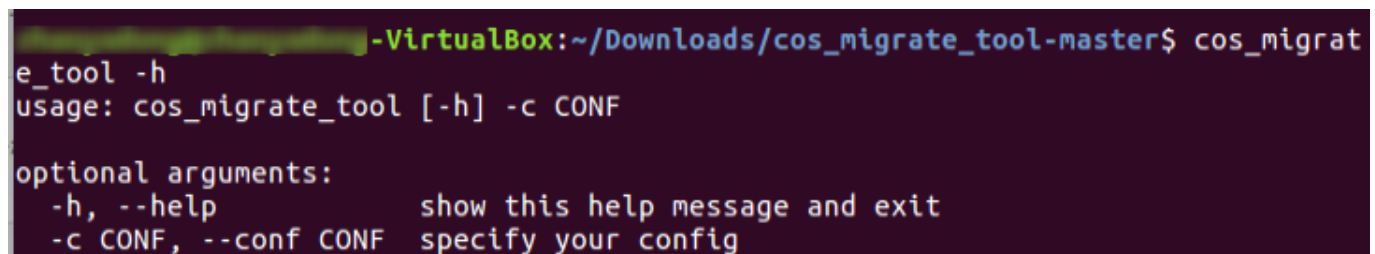


```
Successfully uninstalled coscmd-1.4.11
Successfully installed argparse-1.4.0 boto-2.43.0 cos-migrate-tool-0.1.5 coscmd-0.1.8 crcmod-1.7 leveldb-0.194 lxml-3.7.2 oss2-2.2.0 qcloud-cos-v3-0.1.0 qcloud-cos-v4-0.0.20 qiniu-7.0.0
```

是否安装成功。

```
cos_migrate_tool -h
```

返回结果如下图所示，则表示已安装成功：



```
-VirtualBox:~/Downloads/cos_migrate_tool-master$ cos_migrate_tool -h
usage: cos_migrate_tool [-h] -c CONF

optional arguments:
  -h, --help            show this help message and exit
  -c CONF, --conf CONF  specify your config
```

配置文件

配置文件模板（路径：

```
./cos_migrate_tool-master/conf
```

），配置内容举例如下：

假设用户迁移数据从 OSS 到 COS，common 部分配置基本信息，workspace 是工作目录。source 部分配置数据源的信息，配置 OSS 的属性。destination 部分配置 COS 属性。

配置文件请删除注释文字，即

注释

部分，空白模板参考 [GitHub 页面](#)。

迁移 OSS 中文件

[common]

workspace=/tmp/tmp6

threads=20 # 线程数,不配置默认10

[source]

type=oss

accesskeyid= # oss accesskey id

accesskeysecret= # oss accesskey secret

bucket= # 要迁移的bucket名

endpoint= # oss 的endpoint，例如 oss-cn-beijing.aliyuncs.com

[destination]

type=cosv4

region=sh # cos 的 region，如sh, gz

accesskeyid= # cos 的 accesskeyid

appid= # cos 的 appid

accesskeysecret= # cos 的 accesskeysecret

bucket=sdctest # cos 的 bucket

prefix_dir=/dir21 # cos 的目录，迁移的文件都会位于该目录下，不配置该项则迁移到根目录

迁移七牛云中的文件

[common]

workspace=/tmp/tmp11

[source]

type=qiniu

accesskeyid= # qiniu 的 accesskeyid

accesskeysecret= # qiniu 的 accesskeysecret

bucket= # 要迁移的qiniu的bucket

domain_url= # qiniu 的下载域名

[destination]

type=cosv4

region=sh # cos 的 region , 如sh, gz

accesskeyid= # cos 的 accesskeyid

appid= # cos 的 appid

accesskeysecret= # cos 的 accesskeysecret

bucket=sdctest # cos 的 bucket

prefix_dir=/dir21 # cos 的目录, 迁移的文件都会位于该目录下, 不配置该项则迁移到根目录

迁移 S3 中的文件

[common]

workspace=/tmp/tmp21

[source]

type=s3

accesskeyid= # s3 的 accesskey id

accesskeysecret= # s3 的 accesskey secret

bucket= # s3 的要迁移的bucket名

region= # 中国区s3必填

[destination]

type=cosv4

region=shanghai

accesskeyid=

```
appid=  
accesskeysecret=  
bucket=
```

迁移列表文件

```
[common]  
workspace=
```

```
[source]  
type=url  
url_list_file=/tmp/urllist.txt # 要迁移的文件url列表文件，文件每一行为一个完整的url  
timeout=3 # http请求的超时时间
```

```
[destination]  
type=cosv4  
region=  
accesskeyid=  
appid=  
accesskeysecret=  
bucket=
```

运行工具

在安装成功后，系统会有一个可执行命令

```
cos_migrate_tool
```

，之后的迁移过程都是使用该命令。执行命令的方式如下：

```
cos_migrate_tool -c /path/to/your/conf
```

配置文件自行编写，模版参考 [配置文件](#)

。在配置文件中，需要配置一个工作目录，之后迁移过程中产生的临时文件都在放置在该目录，请保证目录空间足够大，如果并行执行多个迁移任务，推荐使用不同的目录。

在迁移过程中，你可以查看你设定的工作目录下面的 fail_file.txt 来查看迁移失败的文件列表。

卸载方法

执行如下命令：

```
pip uninstall cos_migrate_tool
```

常见问题

1. pip 命令不存在。

使用

```
apt install python-pip
```

或者

```
yum install python-pip
```

命令安装PIP。

2. 使用 pip 安装迁移工具不成功。

尝试执行

```
sudo pip install cos_migrate_tool
```

。

本地同步工具

功能说明

将本地目录下的子文件和子目录同步到 COS 上。

实现机制

COS 本地同步工具会获取用户本地的文件列表，执行上传并将上传的结果记录在本地。每次运行工具都会重新拉取本地文件列表，与db目录下已上传成功的本地文件历史记录进行比对, 然后采取如下的策略:

1. 如果本地新增了文件, 则会新增上传到COS
2. 如果本地修改了文件, 则会覆盖上传COS上已存在的文件
3. 如果文件未发生过变化, 则会跳过，不进行任何处理。

使用限制

只适用于 COS V4 V5 版本

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

具体安装与配置说明请参考 [Java 安装与配置](#)。

使用方法

获取工具包

下载链接：[本地同步工具](#)

解压缩工具包并进入工具包路径：

- Windows：

解压并保存到到某个目录, 比如

C:\Users\Administrator\Downloads\cos_sync

- Linux:

```
unzip cos_sync_tools_v5-master.zip && cd cos_sync_tools_v5-master
```

配置说明

配置文件位于

conf/config.ini

，修改配置示例如下，请根据自己实际情况进行修改，

```
# 用户的密钥 secret_id (可在 https://console.cloud.tencent.com/capi 查看)
secret_id=XXXXXXXXXXXXXXXXXXXXXXX
# 用户的密钥 secret_key (可在 https://console.cloud.tencent.com/capi 查看)
secret_key=YYYYYYYYYYYYYYYYYYYYYYY
# Bucket的命名规则为{name}-{appid}，即bucket名必须包含appid, 例如movie-1251000000
bucket=mybucket-1251000000
```

```
# bucket的地域region信息. 请填写XML API 对应的region, COS的xml api region的简称请参照
https://cloud.tencent.com/document/product/436/6224
region=ap-beijing-1
# 存储类型, 标准(standard), 低频(standard_ia), 近线(nearline)
storage_class=standard
# 本地路径
local_path=/mydata/
# cos路径
cos_path=/mycospath
# 是否使用HTTPS传输(传输速度较慢, 适用于对传输安全要求高的场景), 1开启, 0关闭
enable_https=0
```

配置信息说明：

名称	描述	有效值
secret_id	APPID 对应的密钥 ID，可从控制台获取，参考 基本概念	字符串
secret_key	APPID 对应的密钥 Key，可从控制台获取，参考 基本概念	字符串
bucket	指定要同步的存储桶名称，需要提前在控制台建立，参考 创建存储桶	字符串
enable_https	启用 https 进行传输。1：使用 https；0：使用 http	数字
region	存储桶所属地域。枚举值为 可用地域 中适用于 XML API 的地域简称，如 ap-beijing 等。	字符串
local_path	需要同步的本地绝对路径。Windows 路径需用双斜线 “\\” 分割。 Linux 范例：/home/user/dir；Windows	字符串

名称	描述	有效值
	范例：C:\\document\\dir	
cos_path	同步到 COS 的目的路径，需以 / 为结尾以表示目录，根目录为 /	字符串

注意：

Windows 路径使用 “\\” 进行分割。

因为如果使用 “\” ，配置文件中某些特殊字符会被当做被转义。

使用软件

执行同步工具：

Windows：请双击【start_cos_sync.bat】。

Linux:

```
sh start_cos_sync.sh
```

执行完成后，会输出上传成功, 失败, 跳过忽略的统计，以及全部执行的时间。

问题与帮助

常见问题

同步完成后，在 COS 上不小心删除了文件，再运行工具会上传吗？

不会。工具对于已同步的文件列表是记录在本地的，不会从 COS 拉取文件列表。

已同步记录的数据库保存在哪？删除了再运行工具会发生什么？

同步结果记录会保存在 db

目录下的数据文件里。如果删除了再运行工具，工具会试图将本地文件全量再上传到 COS，如果 COS 上已存在文件则会进行覆盖。

是否支持中文文件名和目录？

支持。目前支持所有 UTF-8 编码的路径或文件。

如果有偶发的失败问题，可以重跑同步程序，不会重复上传已经成功上传的文件。

其他错误

请 [提交工单](#)。并告知同步工具的 config.json 相关配置(不用提供密钥) 以及打包的 log 目录。

FTP Server 工具

COS FTP Server 工具支持通过 FTP 协议直接操作 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。

使用环境

系统环境

操作系统：Linux，推荐使用腾讯云 CentOS 7 系列 CVM，暂不支持 Windows 系统。

Python 解释器版本：Python 2.7，可参考 [Python 安装与配置](#) 进行安装与配置。

依赖库：

- requests
- argparse

下载与安装

GitHub 链接：[COS FTP Server 工具](#)。

下载完成后，直接运行

```
cos ftp server
```

目录下的

```
setup.py
```

即可，需要联网安装依赖库。

```
python setup.py install # 这里可能需要sudo或者root权限
```

特别说明

本工具使用 COS XML 接口开发

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。

说明

- 目前只支持操作一个 Bucket，后期可能会支持同时操作多个 Bucket。
- FTP Server 工具暂不支持断点续传功能。
- 不支持上传空文件（0B），支持的最大文件为 200 GB。

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，如 FileZilla。）

配置文件

conf/vsftpd.conf

为 FTP Server 工具的配置文件，相关配置项的说明如下：

[COS_ACCOUNT]

cos_secretid = XXXXXX

cos_secretkey = XXXXXX

SecretId 和 SecretKey 可以在以下地址获取：<https://console.cloud.tencent.com/cam/capi>

cos_bucket = BucketName-appid

要操作的bucket，bucket的格式为：bucektname-appid组成。示例：cos_bucket = mybucket-12588888888888。

cos_region = ap-xxx

Bucket 所在的地域，目前支持的地域请参照【可用地域-适用于 XML API 部分】：<https://cloud.tencent.com/document/product/436/6224>

cos_user_home_dir = /home/cos_ftp/data

FTP Server 的工作目录。

[FTP_ACCOUNT]

login_users = user1:pass1:RW;user2:pass2:RW

FTP 账户配置。配置格式为<用户名:密码:读写权限>，多个账户用分号分割。

[NETWORK]

`masquerade_address = XXX.XXX.XXX.XXX`

当 FTP Server 处于某个网关或 NAT 后时，可以通过该配置项将网关的 IP 地址或域名指定给 FTP Server。一般情况下，无需配置。

`listen_port = 2121`

Ftp Server 的监听端口，默认为 2121，请注意防火墙需要放行该端口。

`passive_ports = 60000,65535`

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。

[FILE_OPTION]

`single_file_max_size = 21474836480`

默认单文件大小最大支持到 200 GB，不建议设置太大。

[OPTIONAL]

以下设置，如无特殊需要，建议保留default设置。如需设置，请填写一个合理的整数。

`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

配置中OPTIONAL选项是用于调整上传性能的可选项，一般情况下保持默认值即可。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。 `max_connection_num` 为最大连接数的限制选项，设置为0表示不限制最大连接数，可以根据机器情况进行调整。

运行

正确填写配置文件后，直接通过 Python 运行根目录下的

`ftp_server.py`

即可启动 FTP Server。也可以配合screen 的命令将 FTP Server 放到后台运行。

python ftp_server.py

运行命令后，见到如下图示，即代表 FTP Server 服务启动成功，您可以开始使用 FTP 客户端对配置的 IP 和端口进行访问了。

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

停止

Ctrl + C

即可取消 FTP Server 运行（直接运行，或 screen 方式放在后台运行）。

FAQ

上传大文件的时候，中途取消，为什么 COS 上会留有已上传的文件？

由于适用于 COS V5 版本的 FTP Server

提供了完全的流式上传特性，用户文件上传的取消或断开，都会触发大文件的上传完成操作。因此，COS 会认为用户数据流已经上传完成，并将已经上传的数据组成一个完整的文件。

如果用户希望重新上传，可以直接以原文件名上传覆盖；也可手动删除不完整的文件，重新上传。

为什么 FTP Server 配置中要设置最大上传文件的限制？

COS 的分片上传数目最大只能为 10000 块，且每个分片的大小限制为 1 MB ~ 5 G。

这里设置最大上传文件的限制是为了合理计算一个上传分片的大小。

FTP Server 默认支持 200 GB 以内的单文件上传，但是不建议用户设置过大，因为单文件大小设置越大，上传时的分片缓冲区也会相应的增大，这可能会耗用户的内存资源。因此，建议用户根据自己的实际情况，合理设置单文件的大小限制。

如果上传的文件超过最大限制，会怎么样？

当实际上传的单文件大小超过了配置文件中的限制，系统会返回一个 `IOError` 的异常，并且在日志中标注错误信息。

其他问题，请 [提交工单](#)，并在工单上附上完整的

`cos_v5.log`

日志，便于我们进一步排查和解决问题。

COSFS 工具

功能说明

COSFS 工具支持将 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。COSFS 的主要功能包括：

- 支持 POSIX 文件系统的大部分功能，如：文件读写、目录操作、链接操作、权限管理、uid/gid 管理等功能；
- 大文件传输功能；
- MD5 数据校验功能。

使用限制

本工具可以支持对COS V4、V5 版本存储的访问，但是域名均需使用 COS V5 域名。

使用环境

系统环境

主流 Linux 系统

软件环境

本工具编译需要 C++ 编译环境。依赖于 automake、git、libcurl-devel、libxml2-devel、fuse-devel、make、openssl-devel 等软件，安装方法参考 [环境安装](#)。

环境安装

Ubuntu 系统下安装环境依赖包方法：

```
sudo apt-get install automake autotools-dev g++ git libcurl4-gnutls-dev libfuse-dev libssl-dev
```

```
libxml2-dev make pkg-config
```

CentOS 系统下安装环境依赖包方法：

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

注意在centos6.5及较低版本，可能会提示fuse版本太低，在安装过程的configure操作时返回

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

此时，你需要来手动安装fuse版本，具体步骤

```
# yum remove -y fuse-devel
# wget https://github.com/libfuse/libfuse/releases/download/fuse_2_9_4/fuse-2.8.4.tar.gz
# tar -zxvf fuse-2.8.4.tar.gz
# cd fuse-2.8.4
# ./configure
# make
# make install
# export PKG_CONFIG_PATH=/usr/lib/pkgconfig:/usr/lib64/pkgconfig:/usr/local/lib/pkgconfig
# modprobe fuse
# echo "/usr/local/lib" >> /etc/ld.so.conf
# ldconfig
# pkg-config --modversion fuse
2.8.4 //看到版本表示安装成功
```

使用方法

获取工具

Github 获取地址：[COSFS 工具](#)

安装工具

您可以直接将下载的源码上传至指定目录，也可以使用 GitHub 下载到指定目录，下面以使用 GitHub 将源码目录下载到

`/usr/cosfs`

为例：

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

进入到该目录，编译安装：

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

配置文件

在

`/etc/passwd-cosfs`

文件中，配置您的存储桶的名称，以及该存储桶对应的 SecretId 和 SecretKey，相关概念参见[对象存储基本概念](#)。使用冒号隔开，注意冒号为半角符号。并为

/etc/passwd-cosfs

设置可读权限。命令格式如下：

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

其中：

bucketname/ SecretId/ SecretKey 需要替换为用户的真实信息。

示例：

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

运行工具

将配置好的存储桶挂载到指定目录，命令行如下：

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

其中：

- your-APPID/ your-bucketname 需要替换为用户真实的信息；
- your-mount-point 替换为本地需要挂载的目录（如 /mnt）；
- cos-domain-name 为存储桶所属地域对应域名，形式为

http://cos.<Region>.myqcloud.com

，其中 Region 为 [可用地域](#) 中适用于 XML API 的地域简称，如：

`http://cos.ap-guangzhou.myqcloud.com`

、

`http://cos.eu-frankfurt.myqcloud.com`

等。

- `-odbglevel` 参数表示信息级别，照写即可。

示例：

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

另外，如果对性能有要求，可以使用本地磁盘缓存文件，命令中加入 `-ouse_cache` 参数，示例如下：

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

`/local_cache_dir`

为本地缓存目录，如果不需要本地缓存或本地磁盘容量有限，可不指定该选项。

卸载存储桶：

```
fusermount -u /mnt
```

或者

```
umount -l /mnt
```

注意事项

- COSFS 提供的功能和性能和本地文件系统相比，具有一些局限性。具体包括：随机或者追加写文件会导致整个文件的重写。
- 多个客户端挂载同一个 COS 存储桶时，依赖用户自行协调各个客户端的行为。例如避免多个客户端写同一个文件等。
- 不支持 hard link 。不适合高并发读/写的场景。
- 挂载、卸载文件时，不要同时在挂载点上。可以先 cd 到其他目录，再对挂载点进行挂载、卸载操作。

常见问题

- 如何挂载目录

在挂载命令的时候，可以指定目录，如

```
cosfs appid:my-bucket:/my-dir /tmp/cosfs -ourl=http://cn-south.myqcloud.com  
-odbglevel=info -ouse_cache=/path/to/local_cache  
注意，my-dir必须以/开头
```

- 为什么之前可用写文件，突然不能写了？

由于cos鉴权产品策略调整，所以老版本的cosfs工具会导致策略校验不过，因此需要拉取最新的cosfs工具重新mount

- 在centos6.5及较低版本，提示fuse版本太低，该如何解决？

如在configure操作时，提示

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

此时，你需要来手动安装fuse版本，具体步骤

```
# yum remove -y fuse-devel
# wget https://github.com/libfuse/libfuse/releases/download/fuse_2_9_4/fuse-2.8.4.tar.gz
# tar -zxvf fuse-2.8.4.tar.gz
# cd fuse-2.8.4
# ./configure
# make
# make install
# export
PKG_CONFIG_PATH=/usr/lib/pkgconfig:/usr/lib64/pkgconfig:/usr/local/lib/pkgconfig
# modprobe fuse
# echo "/usr/local/lib" >> /etc/ld.so.conf
# ldconfig
# pkg-config --modversion fuse
2.8.4 //看到版本表示安装成功
```

- 为什么cosfs在正常使用过程中，突然退出了，重新挂载显示"unable to access MOUNTPOINT /path/to/mountpoint: Transport endpoint is not connected"

如果cosfs不是被强制Kill掉，那么检查机器上的fuse版本是否低于2.9.4，libfuse在低于2.9.4版本的情况下可能会导致cosfs异常退出。

建议更新fuse版本，或下载cosfs V1.0.2及以上版本。下载地址:

<https://github.com/tencentyun/cosfs/releases>

- 为什么通过cosfs上传的文件Content-Type全是"application/octet-stream"?

cosfs是根据/etc/mime.types和上传的文件后缀进行比对，自动设置Content-Type，建议查看机器上是否存在该文件。

对于ubuntu可以通过sudo apt-get install mime-support来添加

对于centos可以通过`sudo yum install mailcap`来添加

或者手动添加，每种格式一行，例如：`image/png png`

Hadoop 工具

功能说明

Hadoop cosn 插件实现了以腾讯云 COS 作为底层存储文件系统运行上层计算任务的功能，使用 Hadoop 大数据处理引擎，如 MapReduce，Hive、Spark、Tez 等，可以处理存储在腾讯云对象存储 COS 上的数据。

使用限制

只适用于 COS V4 版本

使用环境

系统环境

Linux 或 Windows 系统

软件依赖

Hadoop-2.7.2 及以上版本

安装及配置

具体 Hadoop 安装与配置 请参考 [Hadoop 安装与测试](#)。

使用方法

安装 Maven

- Linux :

```
sudo apt-get install maven
```

- Windows :

下载链接：[Maven](#)

安装与配置请参考：[Windows 环境下 Maven 安装与环境变量配置](#)

获取 cos-java-sdk

下载地址：[cos-java-sdk](#)

进入存放路径，运行以下命令进行编译，获取 target 目录下的 cos_api-4.2.jar：

```
mvn clean package -Dmaven.test.skip=true
```

获取 hadoop-cos 插件

下载地址：[hadoop-cos 插件](#)

因为 cosn 依赖 SDK，请将上一步编译的 cos_api-4.2.jar 拷贝到

```
src/main/resources
```

下，然后运行以下命令进行编译，获取 target 目录下的 hadoop-cos-2.7.2.jar：

```
mvn clean package -Dmaven.test.skip=true
```

插件安装方法

修改 hadoop_env.sh

在

`$HADOOP_HOME/etc/hadoop`

目录下，进入 `hadoop_env.sh`，增加如下内容，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
```

将获取的 `cos_api-4.2.jar` 和 `hadoop-cos-2.7.2.jar` 拷贝到

`$HADOOP_HOME/share/hadoop/tools/lib`

下，同时将本地 Maven 仓库下的

`/org/json/json/20140107`

目录下的 `json-20140107.jar` 和

`/org/apache/httpcomponents/httpmime/4.2.5`

目录下的 `httpmime-4.2.5.jar` 拷贝到该目录。

一般本地 Maven 仓库的位置默认在

`${user.home}/.m2/repository`

目录下，由 Maven 的配置文件 `${MAVEN_HOME}/conf/settings.xml` 里的 `localRepository` 变量控制。

修改配置文件使用插件

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>hadoop.tmp.dir</name>
    <value>/data/rabbitliu/work/hadoop/hadoop_test</value>
  </property>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
  <property>
    <name>dfs.name.dir</name>
    <value>/data/rabbitliu/work/hadoop/hadoop_test/name</value>
  </property>
  <property>
    <name>fs.cos.userinfo.appid</name>
    <value>1252448703</value>
  </property>
  <property>
    <name>fs.cos.userinfo.secretId</name>
    <value>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</value>
  </property>
  <property>
    <name>fs.cos.userinfo.secretKey</name>
    <value>xxxxxxxxxxxxxxxxxxxxxxxx</value>
  </property>
  <property>
    <name>fs.cosn.impl</name>
    <value>org.apache.hadoop.fs.cosnative.NativeCosFileSystem</value>
  </property>
  <property>
    <name>fs.cos.buffer.dir</name>
    <value>/data/rabbitliu/work/hadoop/hadoop_test/cos_buf</value>
```

```
</property>
<property>
<name>fs.cos.userinfo.region</name>
<value>tj</value>
</property>
</configuration>
```

注意：

配置文件中含有 COS 的几个属性：

- fs.cos.userinfo.appid 属性：填写您使用的 腾讯云账户的 APPID。可通过 [控制台-账号信息](#) 查看。
- fs.cos.userinfo.secretId/secretKey 属性：填写您账户的API 密钥信息。可通过 [云 API 密钥控制台](#) 查看。
- fs.cosn.impl 为 cosn 的实现类，固定为 org.apache.hadoop.fs.cosnative.NativeCosFileSystem。
- fs.cos.buffer.dir 请设置一个实际存在的目录，运行过程中产生的临时文件会暂时放于此处。
- fs.cos.userinfo.region 请填写您的地域信息，枚举值为 [历史版本地域列表](#) 中的地域简称，如 sh, gz, sgp 等。

使用软件（以 Linux 为例）

使用 hadoop fs 常用命令

命令格式为：

```
hadoop fs- -ls cosn://Bucket 路径
```

，下例中以名称为 example 的 Bucket 为例，可在其后面加上具体路径。

```
hadoop fs -ls cosn://example/
```

Found 7 items

```
-rw-rw-rw- 1 example example 3669 2016-10-25 21:23 cosn://example/b.txt
drwxrwxrwx - example example 0 1970-01-01 08:00 cosn://example/dir1
drwxrwxrwx - example example 0 1970-01-01 08:00 cosn://example/mr
-rw-rw-rw- 1 example example 16952 2016-10-25 21:37 cosn://example/qcloud_sign.proto
-rw-rw-rw- 1 example example 2048 2016-10-25 21:48 cosn://example/test2K.txt
-rw-rw-rw- 1 example example 52428800 2016-10-27 16:40 cosn://example/test50MB.txt
drwxrwxrwx - example example 0 1970-01-01 08:00 cosn://example/xx1
```

运行 MapReduce 自带的 wordcount

注意：

以下命令中 `hadoop-mapreduce-examples-2.7.2.jar` 是以 2.7.2 版本为例，如版本不同，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

HDFS TO COS 工具

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（COS）上。

使用限制

适用于 COS V5 版本

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考 [Java 安装与配置](#)。

配置及使用方法

配置方法

1. 安装 Hadoop-2.7.2 及以上版本，具体安装步骤请参考 [Hadoop 安装与测试](#)。
2. 通过 [GitHub 地址](#) 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

，包括 APPID、存储桶（Bucket）、地域（Region）以及 API 密钥信息。

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

（以 Linux 为例）

查看帮助

./hdfs_to_cos_cmd -h

执行结果如下图所示：

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

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
```

这里只做长度的判断，因为如果将 Hadoop 上的文件摘要算出，开销较大。

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 APPID、存储桶（Bucket）、地域（Region）以及 API 密钥信息。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 DateNode

请保证对于 DateNode，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 DateNode 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 DateNode 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 默认为是目录，最终从 HDFS 上拷贝的文件都会存放在该目录下。

WeCOS 小程序瘦身工具

功能说明

为了提升小程序体验流畅度，微信限制编译后的代码包大小需小于 1MB，大于 1MB 的代码包将上传失败。在开发小程序的过程中，图片资源通常会占用较大空间，很容易超出官方的 1MB 限制。使用 WeCOS，您在开发过程中不再需要关心图片资源占用空间的问题，专注于逻辑开发。

通过 WeCOS，小程序项目中的图片资源会自动上传到 COS 上，且 WeCOS 自动替换代码中图片资源地址的引用为线上地址，移除项目目录中的图片资源，从而减小代码包大小，解决包大小超过限制的问题。

准备工作

1. 进入 [腾讯云官网](#)，注册腾讯云账户，指引参考 [注册腾讯云](#)。
2. 登录 [对象存储控制台](#)，开通对象存储服务，创建存储桶，指引参考 [创建存储桶](#)
3. 通过 [GitHub 地址](#) 下载 WeCOS 工具。
4. 在 [Node.js 官网](#) 下载环境并安装。

安装

使用以下命令安装 WeCOS 工具：

```
npm install -g wecos
```

基本配置

在您的小程序目录同级下创建

```
wecos.config.json
```

文件，文件配置示例及配置项说明如下：

```
{
  "appDir": "./app",
  "cos": {
    "appid": "1234567890",
    "bucketname": "wxapp",
    "folder": "/", //资源存放在bucket的哪个目录下
    "region": "wx", //创建bucket时选择的地域简称
    "secret_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    "secret_key": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}
```

配置项	类型	说明
appDir	[String]	默认 ./app ，小程序项目目录
cos	[Object]	必填，COS 上对应存储桶的配置信息，部分信息可在 控制台 查看

使用

在配置文件同级目录下执行以下使用命令：

wecos

注意：

执行使用命令前，需要在配置文件同级目录下创建

wecos.config.json

文件。

高级配置

配置项	类型	说明
backupDir	[String]	默认 ./wecos_backup ，备份目录
uploadFileSuffix	[Array]	默认 (.jpg .png .gif)，图片上传后缀名配置
uploadFileBlackList	[Array]	默认 []，图片资源黑名单
replaceHost	[String]	默认 "，把指定域名替换成 targetHost
targetHost	[String]	默认 "，使用自定义域名
compress	[Boolean]	默认 false，是否开启压缩图片
watch	[Boolean]	默认 true，是否开启实时监听项目目录

设置备份目录

由于 WeCOS 在运行时会自动将项目下的图片上传至 COS 然后删除，这样可能存在丢失源文件的风险，我们提供了备份源文件的功能，每上传一张图片，会在项目同级的某个目录下备份该文件。您可通过以下配置来修改备份目录名，若不需要使用该功能，可以设置为空值：

```
"backupDir": "./wecos_backup"
```

设置图片后缀

当需要限制上传图片的格式（例如只允许

jpg

格式) 时, 可通过 WeCOS 提供的图片后缀配置项来定义。WeCOS 默认支持 jpg、png、gif 三种格式, 若你还需要添加其他格式 (例如 webp), 可在该配置项中添加, 示例如下:

```
"uploadFileSuffix": [".jpg", ".png", ".gif", ".webp"]
```

设置图片黑名单

开发过程中, 不希望上传某些特定的图片时, 可通过 WeCOS 的黑名单配置来实现, 配置后上传程序会自动忽略掉这些图片。黑名单配置支持目录或具体到文件名的写法。

```
"uploadFileBlackList": ["/images/logo.png", "/logo/"]
```

自定义域名

当您希望 COS 文件链接使用您的自有域名时, 可配置 targetHost 代替默认域名 (配置时可省略

http://

) :

```
"targetHost": "http://example.com"
```

想为代码中的图片链接替换域名时, 可通过配置 replaceHost targetHost 来实现。

```
"replaceHost": "http://wx-12345678.myqcloud.com",
```

```
"targetHost": "https://example.com"
```

开启图片压缩

图片上传到 COS

之后虽然大大减轻了程序包的大小，但如果图片自身体积过大导致访问延迟，也会影响到用户体验。

WeCOS 在图片上云的基础功能上还额外提供了基于 [腾讯云万象优图](#) 的图片压缩功能。您在 [万象优图控制台](#) 创建与 COS 同名的存储桶后，进入存储桶，在样式页面开启图片压缩功能后，资源将被压缩后上传。

```
"compress": true
```

设置实时监听

WeCOS 默认实时监听项目目录变化，自动处理图片资源。在开发过程中，若觉得实时监听不方便，或只需要一次性处理就停止，可以按照如下命令行修改该配置，程序将只执行一次后退出。

```
"watch": false
```

高级用法

当上述使用命令行来执行的方式无法满足您的需求，您还可以使用其他方式（例如定制化成自己的模块），我们提供了直接引用的使用方法满足您的个性化需求。配置示例如下：

```
var wecos = require('wecos');

// option 可选 (String|Object)
// 传入 String，指定配置文件路径
// 传入 Object，指定配置项
wecos([option]);

// 指定配置文件路径
wecos('./wecos-config.js');

// 指定配置项
wecos({
  appDir: './xxx',
```

```
cos: {  
  ...  
}  
});
```

相关资源

- [WeCOS-UGC-DEMO](#) —— 小程序用户资源上传 COS DEMO
- [COS-AUTH](#) —— COS 鉴权服务器 DEMO