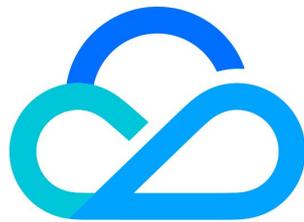


对象存储 工具指南 产品文档



腾讯云

【版权声明】

©2013-2018 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

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

COSCMD 工具

COSBrowser 工具

COS Migration 工具

FTP Server 工具

COSFS 工具

Hadoop 工具

HDFS TO COS 工具

WeCOS 小程序瘦身工具

工具指南

工具概览

最近更新时间：2018-07-19 10:24:21

对象存储提供以下开发者工具，供用户使用：

工具	功能说明
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。
COS Browser 工具	本工具支持用户通过可视化界面，方便地进行数据的上传、下载等操作。
COS Migration 工具	本工具支持用户从本地、其他云存储等多种源数据地址迁移至 COS。
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。
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。
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。
HDFS TO COS 工具	本工具用于将 HDFS 上的数据拷贝到 COS 上。
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。

环境安装与配置

Java 安装与配置

最近更新时间：2018-05-24 17:27:22

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

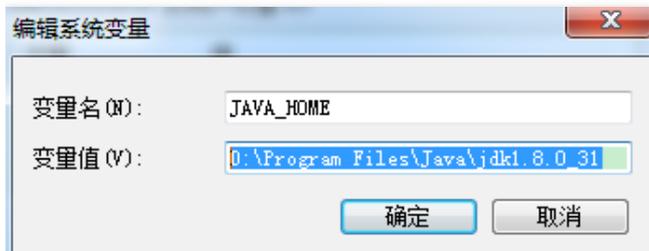


3. 配置

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



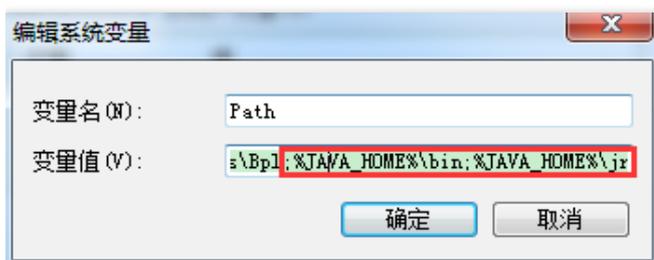
变量名(N) : **CLASSPATH**

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。

```
C:\Users\Administrator> javac
用法: javac <options> <source files>
其中, 可能的选项包括:
-g                生成所有调试信息
-g:none          不生成任何调试信息
-g: {lines, vars, source} 只生成某些调试信息
-nowarn         不生成任何警告
-verbose        输出有关编译器正在执行的操作的消息
-deprecation    输出使用已过时的 API 的源位置
-classpath <路径> 指定查找用户类文件和注释处理程序的位置
-cp <路径>      指定查找用户类文件和注释处理程序的位置
```

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, 您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

最近更新时间：2018-05-15 18:14:54

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

```
wget https://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

最近更新时间：2018-10-10 16:31:11

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

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，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 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 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  
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

```
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_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
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

最近更新时间：2018-09-10 17:22:41

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

Windows 或 Linux 系统

(请保证本地字符格式为 UTF-8，否则操作中文文件会出现异常)

软件依赖

Python 2.7/3.5/3.6，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](#) 文档。

下载与安装

- **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。如果在安装过程中出现依赖包失败的提示，请使用 `-I` 参数重试，命令如下：

```
pip install -I coscmd
```

- **pip 更新**

执行 pip 命令进行更新：

```
pip install coscmd -U
```

注意：

当 pip 版本大于等于 10.0.0 时，升级或安装依赖库时可能会出现失败，建议使用 pip 版本 9.x (pip install pip==9.0.0) 。

• 源码安装(不推荐)

下载链接：[GitHub 链接](#)

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

注意：

python 版本为 2.6 时，pip 安装依赖库时容易失败，推荐使用该方法安装。

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 //查看当面版本信息
```

help 信息如下所示：

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

...

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

positional arguments:

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

config config your information at first.

upload upload file or directory to COS.

download download file **from** COS to local.

delete delete file or files **on** COS

```

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

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

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 [-h] -a <SECRET_ID> -s <SECRET_KEY> -b <BUCKET>
(-r <REGION> | -e <ENDPOINT>) [-m <MAX_THREAD>]
[-p <PART_SIZE>] [-u <APPID>] [--do-not-use-ssl]
[--anonymous <ANONYMOUS>]
    
```

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 COS 控制台左侧栏【密钥管理】或 云 API 密钥控制台 获取	字符串
SECRET_KEY	必选参数，APPID 对应的密钥 Key 可从 COS 控制台左侧栏【密钥管理】或 云 API 密钥控制台 获取	字符串

名称	描述	有效值
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
PART_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

注意：

1. 可以直接编辑 `~/cos.conf` 文件（在 Windows 环境下，该文件是位于 `我的文档` 下的一个隐藏文件），该文件初始时不存在，是通过 `coscmd config` 命令生成，用户也可以手动创建。

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 = ap-guangzhou
max_thread = 5
part_size = 1
schema = https
```

2. 可以在配置文件中增加 `schema` 项来选择 `http / https`，默认为 `https`。
3. 可以在 `anonymous` 项中选择 `True/False`，来使用匿名模式，即签名保持为空。
4. bucket的命名规则为 `{name}-{appid}`。

指定 Bucket 的命令

- 通过 `-b <bucket>` 指定 Bucket 可以指定特定 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。

```
#命令格式
coscmd -b <bucket> method ...
#操作示例-上传文件
coscmd -b AAA-12345567 upload a.txt b.txt
#操作示例-创建 bucket
coscmd -b AAA-12344567 createbucket
```

创建 Bucket

- 建议配合 `-b <bucket>` 指定 Bucket 使用

```
#命令格式
coscmd -b <bucket> createbucket
#操作示例
coscmd createbucket
coscmd -b AAA-12344567 createbucket
```

删除 Bucket

- 建议配合 `-b <bucket>` 指定 Bucket 使用

```
#命令格式
coscmd -b <bucket> deletebucket
#操作示例
coscmd createbucket
coscmd -b AAA-12344567 deletebucket
```

上传文件或文件夹

- 上传文件命令如下：

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

- 上传文件夹命令如下：

```
#命令格式
coscmd upload -r <localpath> <cospath>
#操作示例
coscmd upload -r /home/aaa/ bbb/aaa
coscmd upload -r /home/aaa/ bbb/
#该操作会在 bbb/目录下新建一个 aaa/文件夹
coscmd upload -r /home/aaa bbb/
#上传到 bucket 根目录
coscmd upload -r /home/aaa/ /
#同步上传，跳过 md5 相同的文件
coscmd upload -rs /home/aaa/ /home/aaa
#忽略.txt 和.doc 的后缀文件
coscmd upload -rs /home/aaa/ /home/aaa --ignore *.txt*.doc
```

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)的名字补全(参考例子)。
- 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 分块上传时会每一块进行 MD5 校验。
- COSMCD 上传默认会携带 x-cos-meta-md5 的头部，值为该文件的 md5 值。
- 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 之后的 COSCMD 上传的，默认带有x-cos-meta-md5的header)。
- 使用-H 参数设置 HTTP header 时，请务必保证格式为 json，这里是个例子：`coscmd upload -H '{"Cache-Control":"max-age=31536000","Content-Language":"zh-CN"}' <localpath> <cospath>`。
- 在上传文件夹时，使用--ignore 参数可以忽略某一类文件，支持 shell 通配规则，支持多条规则，用逗号分隔。
- 目前只支持上传最大 40 T 的单文件。

下载文件或文件夹

- 下载文件命令如下：

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

- 下载文件夹命令如下：

```
#命令格式
coscmd download -r <cospath> <localpath>
#操作示例
coscmd download -r /home/aaa/ bbb/aaa
coscmd download -r /home/aaa/ bbb/
#覆盖下载当前 bucket 根目录下所有的文件
coscmd download -rf / bbb/aaa
#同步下载当前 bucket 根目录下所有的文件，跳过 md5 校验相同的文件
coscmd download -rs / bbb/aaa
#忽略.txt 和.doc 的后缀文件
coscmd download -rs / bbb/aaa --ignore *.txt,*.doc
```

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)

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

- download 接口使用分块下载，老版本的 mget 接口已经废除，请使用 download 接口
- 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（前提是下载文件夹是通过 COSCMD 的 upload 接口上传的，文件携带有 x-cos-meta-md5 头部）
- 在下载文件夹时，使用--ignore 参数可以忽略某一类文件，支持 shell 通配规则，支持多条规则，用逗号分隔

删除文件或文件夹

- 删除文件命令如下：

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

- 删除文件夹命令如下：

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

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

复制文件或文件夹

- 复制文件命令如下：

```
#命令格式
coscmd copy <sourcepath> <cospath>
#操作示例
coscmd copy bucket-appid.cos.ap-guangzhou.myqcloud.com/a.txt aaa/123.txt
```

- 复制文件夹命令如下：

```
#命令格式
coscmd copy -r <sourcepath> <cospath>
#操作示例
coscmd copy -r bucket-appid.cos.ap-guangzhou.myqcloud.com/coscmd/ aaa
coscmd copy -r bucket-appid.cos.ap-guangzhou.myqcloud.com/coscmd/ aaa/
```

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）

注意：

sourcepath 的样式如下：`<bucketname>-<appid>.cos.<region>.myqcloud.com/<cospath>`

打印文件列表

- 打印命令如下：

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

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，并且会在末尾返回列出文件的数量和大小之和
- 使用 `-n num` 设置打印数量的最大值

注意：

`<cospath>` 为空默认打印当前 Bucket 根目录

显示文件信息

- 命令如下：

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

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：

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

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)

- 命令如下：

使用如下命令设置 Bucket 的访问控制：

```
#命令格式
coscmd putbucketacl [--grant-read GRANT_READ] [--grant-write GRANT_WRITE] [--grant-full-control GRANT_FULL_CONTROL]
#操作示例
coscmd putbucketacl --grant-read 12345678,12345678/11111 --grant-write anyone --grant-full-control 12345678/22222
```

使用如下命令设置 Object 的访问控制：

```
#命令格式
coscmd putobjectacl [--grant-read GRANT_READ] [--grant-write GRANT_WRITE] [--grant-full-control GRANT_FULL_CONTROL] <cospath>
#操作示例
coscmd putobjectacl --grant-read 12345678,12345678/11111 --grant-write anyone --grant-full-control 12345678/22222 aaa/aaa.txt
```

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_FILL_CONTORL` 代表被赋权的帐号
- 若赋权根帐号，使用 `rootid` 的形式
- 若赋权子账户，使用 `rootid/subid` 的形式
- 若需要对所有人赋权，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：

```
#命令格式
coscmd getbucketacl
#操作示例
coscmd getbucketacl
```

- 使用如下命令设置 Object 的访问控制：

```
#命令格式
coscmd putbucketacl <cospath>
#操作示例
coscmd getobjectacl aaa/aaa.txt
```

恢复归档文件

- 命令如下：

```
#命令格式
coscmd restore <cospath>
#操作示例
coscmd restore a.txt -d 3 -t Expedited
#/操作示例
coscmd restore a.txt -d 3 -t Bulk
```

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

COSBrowser 工具

最近更新时间：2018-07-12 18:08:11

功能概述

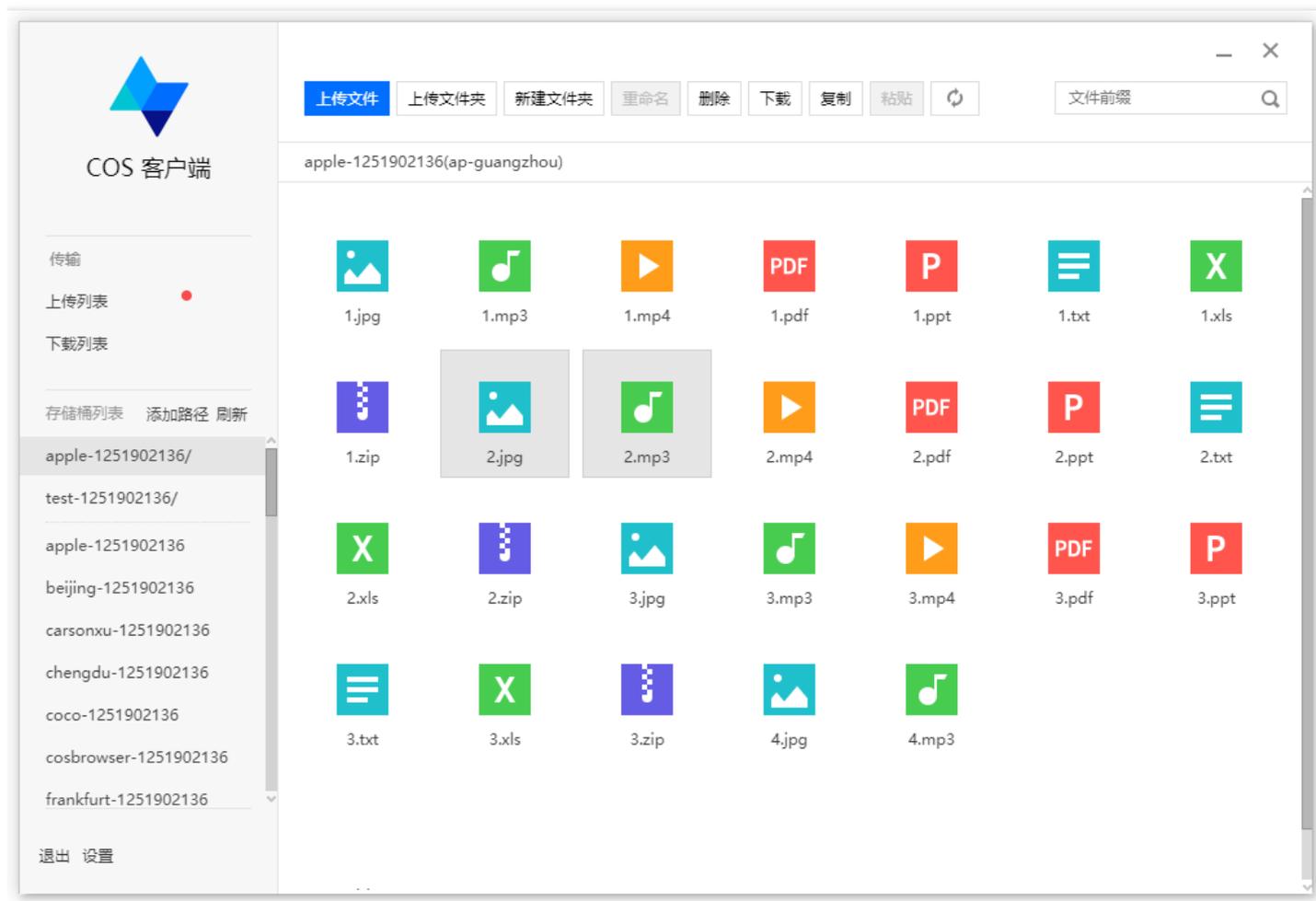
COSBrowser 工具即 COS 桌面版工具，用户可以使用该工具进行可视化、方便的数据上传、下载等操作。

下载与安装

下载安装包（下载后双击安装）：

- [COSBrowser Windows 桌面端](#)
- [COSBrowser macOS 桌面端](#)

软件界面



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。

更新日志

更新日志：[change log](#)

反馈和建议

欢迎提交反馈和建议：[cosbrowser issues](#)

COS Migration 工具

最近更新时间：2018-09-17 14:29:28

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，用户可以将源地址数据快速迁移至 COS 中，它具有以下特点：

- 丰富的数据源
 - 本地数据：将本地存储的数据迁移到 COS。
 - 其他云存储：目前支持 AWS S3，阿里云 OSS，七牛存储迁移至 COS，后续会不断扩展。
 - URL 列表：根据指定的 URL 下载列表进行下载迁移到 COS。
 - Bucket 相互复制：COS 的 Bucket 数据相互复制，支持跨账号跨地域的数据复制。
- 断点续传
- 分块上传
- 并行上传
- 迁移校验

使用环境

系统环境

Linux 或 Windows 环境

软件依赖

- JDK1.7 X64 或以上，有关 JDK 的安装与配置请参考 [Java 安装与配置](#)。

使用方法

1. 获取工具

下载链接：[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
|--src #工具的源码
|--tmp #临时文件存储目录
|--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，用户根据迁移需求填写对应的标识。例如，需要将本地数据迁移至 COS，则 [migrateType] 的配置内容是 type=migrateLocal。

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

目前支持的迁移类型如下：

migrateType	描述
migrateLocal	从本地迁移至 COS
migrateAws	从 AWS S3 本地迁移至 COS
migrateAli	从阿里 OSS 迁移至 COS
migrateQiniu	从七牛迁移至 COS
migrateUrl	下载 URL 迁移到 COS
migrateBucketCopy	从源 Bucket 复制到目标 Bucket

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，包含了要迁移到得目标 COS 的账户信息
[common]
secretId=AKIDXXXXXXXXXXXXXXXXXX
secretKey=GYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
bucketName=mybcket-1251668577
region=ap-guangzhou
storageClass=Standard
cosPath=/
https=off
tmpFolder=./tmp
smallFileThreshold=5242880
smallFileExecutorNum=64
bigFileExecutorNum=8
entireFileMd5Attached=on
daemonMode=off
daemonModeInterval=60
executeTimeWindow=00:00, 24:00
```

名称	描述	默认值
secretId	用户密钥 SecretId，可在 云 API 密钥控制台 查看	-
secretKey	用户密钥 SecretKey，可在 云 API 密钥控制台 查看	-

名称	描述	默认值
bucketName	目的 Bucket 的名称, 合法命名规则为 {name}-{appid}, 即 Bucket 名必须包含 APPID, 例如 movie-1251000000	-
region	目的 Bucket 的 Region 信息。COS 的地域简称请参照 可用地域	-
storageClass	存储类型: Standard - 标准存储, Standard_IA - 低频存储	Standard
cosPath	要迁移到的 COS 路径。/ 表示迁移到 Bucket 的根路径下, /aaa/bbb/ 表示要迁移到 Bucket 的 /aaa/bbb/ 下, 若 /aaa/bbb/ 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, 如 /a/b/c; Windows 下分隔符为两个反斜杠, 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
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	on
daemonMode	是否启用 daemon 模式: on 表示开启, off 表示关闭。daemon 表示程序会循环不停的去执行同步, 每一轮同步的间隔由 daemonModelInterVal 参数设置	off
daemonModelInterVal	表示每一轮同步结束后, 多久进行下一轮同步, 单位为秒	60

名称	描述	默认值
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，休眠态暂停迁移并会保留迁移进度，直到下一个时间窗口自动继续执行	00:00,24:00

3.3 配置数据源信息

根据 [migrateType] 的迁移类型配置相应的分节。例如 [migrateType] 的配置内容是 type=migrateLocal，则用户只需配置 [migrateLocal] 分节即可。

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	本地路径，要求格式为绝对路径： Linux 下分隔符为单斜杠，如 /a/b/c； Windows 下分隔符为两个反斜杠，如E:\\a\\b\\c。
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，比如设置为300，表示只上传更新了5分钟以上的文件

3.3.2 配置阿里 OSS 数据源 migrateAli

若从阿里云 OSS 迁移至 COS，则进行该部分配置，具体配置项及说明如下：

```
# 从阿里 OSS 迁移到 COS 配置分节
[migrateAli]
bucket=mybucket-test
accessKeyId=xxxxxxxxxx
accessKeySecret=yyyyyyyyyyyy
endPoint= OSS -cn-shenzhen.aliyuncs.com
```

```

prefix=
proxyHost=
proxyPort=
    
```

配置项	描述
bucket	阿里云 OSS Bucket 名称
accessKeyId	用户的密钥 accessKeyId
accessKeySecret	用户的密钥 accessKeySecret
endPoint	阿里云 endpoint 地址
prefix	要迁移的路径的前缀, 如果是迁移 Bucket 下所有的数据, 则 prefix 为空
proxyHost	如果要使用代理进行访问, 则填写代理 IP 地址
proxyPort	代理的端口

3.3.3 配置 AWS 数据源 migrateAws

若从 AWS 迁移至 COS, 则进行该部分配置, 具体配置项及说明如下:

```

# 从 AWS 迁移到 COS 配置分节
[migrateAws]
bucket=aws-emr-test
accessKeyId=xxxxxxxxx
accessKeySecret=yyyyyyyyyyyyyyyyy
endPoint=s3.us-east-1.amazonaws.com
prefix=
proxyHost=
proxyPort=
    
```

配置项	描述
bucket	AWS 对象存储 Bucket 名称
accessKeyId	用户的密钥 accessKeyId
accessKeySecret	用户的密钥 accessKeySecret
endPoint	AWS 的 endpoint 地址, 必须使用域名, 不能使用region
prefix	要迁移的路径的前缀, 如果是迁移 Bucket下所有的数据, 则 prefix 为空
proxyHost	如果要使用代理进行访问, 则填写代理 IP 地址
proxyPort	代理的端口

3.3.4 配置七牛数据源 migrateQiniu

若从七牛迁移至 COS，则进行该部分配置，具体配置项及说明如下：

```
# 从七牛迁移到COS配置分节
[migrateQiniu]
bucket=mybuckettest
accessKeyId=xxxxxxxxxx
accessKeySecret=yyyyyyyyyyyyyyyyyy
endPoint=www.bkt.clouddn.com
prefix=
proxyHost=
proxyPort=
```

配置项	描述
bucket	七牛对象存储 Bucket 名称
accessKeyId	用户的密钥 accessKeyId
accessKeySecret	用户的密钥 accessKeySecret
endPoint	七牛下载地址，对应 downloadDomain
prefix	要迁移的路径的前缀, 如果是迁移 Bucket下所有的数据，则 prefix 为空
proxyHost	如果要使用代理进行访问，则填写代理 IP 地址
proxyPort	代理的端口

3.3.5 配置 URL 列表数据源 migrateUrl

若从指定 URL 列表迁移至 COS，则进行该部分配置，具体配置项及说明如下：

```
# 从 URL 列表下载迁移到 COS 配置分节
[migrateUrl]
urllistPath=D:¥¥folder¥¥urllist.txt
```

配置项	描述
urllistPath	URL 列表文件的地址，内容为文本文件，一行一条。参数格式要求为绝对路径： Linux 下分隔符为单斜杠，如 /a/b/c； Windows 下分隔符为两个反斜杠，如 E:\a\b\c。 如果填写的是目录，则会将该目录下的所有文件视为 urllist 文件去扫描迁移

3.3.6 配置 Bucket 相互复制 migrateBucketCopy

若从指定 URL 列表迁移至 COS，则进行该部分配置，具体配置项及说明如下：

```
# 从源 Bucket 迁移到目标 Bucket 配置分节
[migrateBucketCopy]
srcRegion=ap-shanghai
srcBucketName=mysrcbucket-1251668555
srcSecretId=xxxxxxxxxxx
srcSecretKey=yyyyyyyyyyyyyyyyyy
srcCosPath=/
```

配置项	描述
srcRegion	源 Bucket 的 Region 信息，请参照 可用地域
srcBucketName	源 Bucket 的名称，合法命名规则为 {name}-{appid}，即 Bucket 名必须包含 APPID，例如 movie-1251000000
srcSecretId	源 Bucket 隶属的用户的密钥 SecretId，可在 云 API 密钥 查看。如果是同一用户的数据，则 srcSecretId 和 common 中的 SecretId 相同，否则是跨账号 Bucket 拷贝。
srcSecretKey	源 Bucket 隶属的用户的密钥 secret_key，可在 云 API 密钥 查看。如果是同一用户的数据，则 srcSecretId 和 common 中的 secretId 相同，否则是跨账号 Bucket 拷贝。
srcCosPath	要迁移的 COS 路径，表示该路径下的文件要迁移至目标 Bucket

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/
```

特别说明

- 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，且属性和 db 中存在的一致，则跳过迁移，否则进行迁移。这里的属性根据迁移类型的不同而不同，对于本地迁移，会判断 `mtime`。对于其他云存储迁移与 Bucket 复制，会判断源文件的 `etag` 和长度是否与 db 一致。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

- 读取配置文件，根据迁移 type，读取响应的配置分节，并执行参数的检查。
- 根据指定的迁移类型，扫描对比 db 下对所迁移文件的标识，判断是否允许上传。
- 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` 表示因为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。执行过程示意图如下图所示：

```
chengw@vm_24_72_centos [18:23:10]:~/code/java/cos_migrate_tool
$ sh start_migrate.sh
[skip] task info: [taskType: migrateAws] [bucket: chengwus3sdkgz-1251668577] [cosPath: /aws0403_17/aaa/bbbslash.txt]
[DownloadOk] [key: slash.txt] [byteDownload/ byteTotal/ percentage: 6/ 6/ 100.00%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 10240/ 209715200/ 0.00%]
[UploadInProgress] [key: /aws0403_17/slash.txt] [byteSent/ byteTotal/ percentage: 6/ 6/ 100.00%]
[ok] task info: [taskType: migrateAws] [bucket: chengwus3sdkgz-1251668577] [cosPath: /aws0403_17/slash.txt]
[DownloadInProgress] [key: aws_200M.txt.copy] [byteDownload/ byteTotal/ percentage: 10240/ 209715200/ 0.00%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 236126/ 209715200/ 0.11%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 392798/ 209715200/ 0.19%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 514654/ 209715200/ 0.25%]
[DownloadInProgress] [key: aws_200M.txt.copy] [byteDownload/ byteTotal/ percentage: 96863/ 209715200/ 0.05%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 688734/ 209715200/ 0.33%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 827998/ 209715200/ 0.39%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 949854/ 209715200/ 0.45%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1089118/ 209715200/ 0.52%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1228382/ 209715200/ 0.59%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1315422/ 209715200/ 0.63%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1385054/ 209715200/ 0.66%]
[DownloadInProgress] [key: aws_200M.txt.copy] [byteDownload/ byteTotal/ percentage: 166495/ 209715200/ 0.08%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1506910/ 209715200/ 0.72%]
```

- 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，对未成功的会跳过。运行完成结果示意图如下图所示：

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

常见问题

1. 迁移工具中途异常退出怎么办？

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, 会对未上传完的文件进行续传。

2. 对于迁移成功的文件，用户通过控制台或其他方式删除了 COS 上的文件，迁移工具会将这些文件进行重新上传吗？

不会。原因是，所有迁移成功的文件会被记录在 db 中，迁移工具运行之前会先扫描 db 目录，对于已被记录的文件不会再次上传，具体原因请参照 [迁移机制及流程](#)。

3. 迁移失败，日志显示 403 Access Deny，该怎么办？

请确认密钥信息，Bucket 信息，Region 信息是否正确，并且是否具有操作权限。如果是子账号，请让父账号授予相应的权限；如果是本地迁移和其他云存储迁移，需要对 Bucket 具有数据写入和读取权限；如果是 Bucket copy，还需要对源 Bucket 具有数据读取权限。

4. 从其他云存储迁移 COS 失败，显示 Read timed out，该怎么办？

一般来说，这种失败情况是由网络带宽不足所造成，导致从其他云存储下载数据超时。比如，将 AWS 海外的数据迁移到 COS，在下载数据到本地时由于带宽能力不足，导致时延较高，可能会出现 read time out。因此，解决方法为增大机器的网络带宽能力，建议在迁移之前用 wget 测试下载速度。

5. 迁移失败，日志显示 503 Slow Down，该怎么办？

这是触发频控所导致，COS 目前对一个账号具有每秒 800 QPS 的操作限制。建议调小配置中小文件的并发度，并重新运行工具，则会将失败的重新运行。

6. 迁移失败，日志显示 404 NoSuchBucket，该怎么办？

请确认您的密钥信息，Bucket 信息，Region 信息是否正确。

7. 运行异常，显示如下的信息该怎么办？

```
Exception in thread "main" java.lang.ExceptionInInitializerError
    at org.rocksdb.RocksDB.loadLibrary(RocksDB.java:64)
    at org.rocksdb.RocksDB.<clinit>(RocksDB.java:35)
    at org.rocksdb.Options.<clinit>(Options.java:25)
    at com.qcloud.cos_migrate_tool.record.RecordDb.init(RecordDb.java:43)
    at com.qcloud.cos_migrate_tool.task.TaskExecutor.initRecord(TaskExecutor.java:94)
    at com.qcloud.cos_migrate_tool.task.TaskExecutor.run(TaskExecutor.java:146)
    at com.qcloud.cos_migrate_tool.app.App.main(App.java:60)
Caused by: java.lang.UnsupportedOperationException: Cannot determine JNI library name for ARCH='x86' OS='windows 10' name='rocksdb'
    at org.rocksdb.util.Environment.getJniLibraryName(Environment.java:78)
    at org.rocksdb.NativeLibraryLoader.<clinit>(NativeLibraryLoader.java:19)
    ... 7 more
```

此问题是因为工具使用了rocksdb，需要使用64位的JDK，请检查JDK版本是X64的JDK。

8. 其他问题

请重新运行迁移工具，若仍然失败，请将配置信息（密钥信息请隐藏）与 log 目录打包后 [提交工单](#)。

FTP Server 工具

最近更新时间：2018-06-12 10:01:06

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-125888888888。

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_port = 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

配置文件中的 masquerade_address 这个选项有何作用？何时需要配置 masquerade_address

当 FTP Server 运行在一个多网卡的 Server 上，并且 FTP Server 采用了 PASSIVE 模式时（一般地，FTP 客户端位于一个 NAT 网关之后时，都需要启用 PASSIVE 模式），此时需要使用 masquerade_address 选项来唯一绑定一个 passive 模式下用于 reply 的 IP。例如，FTP Server 有多个 IP 地址，如内网 IP 为 10.XXX.XXX.XXX，外网 IP 为 123.XXX.XXX.XXX。客户端通过外网 IP 连接到 FTP Server，同时客户端使用的是 PASSIVE 模式传输，此时，若 FTP Server 未指定 masquerade_address 具体绑定到外网 IP 地址，则 Server 在 PASSIVE 模式下，reply 时，有可能会走内网地址。就会出现客户端能连接上 Ftp server，但是不能从 Server 端获取任何数据回复的问题。

如果需要配置 masquerade_address，建议指定为客户端连接 Server 所使用的那个 IP 地址。

正确配置了 masquerade_address 选项以后，ftp server 可以正常登录，但是执行 FTP 命令：list 或者 get 等数据取回命令时，提示“服务器返回不可路由的地址”或“ftp: connect: No route to host”等错误

这个 case 多半是因为 ftp server 机器 iptables 或防火墙策略配置 reject 或者 drop 掉所有 ICMP 协议包，而 FTP 客户端在拿到 FTP Server 被动模式下返回的数据连接 IP 后，会首先发送一个 ICMP 包探测 IP 的连通性，所以客户端会提示“服务器返回不可路由的地址”等错误。

建议解决方案是：将 iptables 策略按需配置为只 reject 或 drop 希望限制的 ICMP 包类型，如只想禁掉外部 ping 类型的 ICMP 包，可以将策略修改为：`iptables -A INPUT -p icmp --icmp-type 8 -s 0/0 -j [REJECT/DROP]` 或者单独放开要访问 ftp server 的客户端的 IP。

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

最近更新时间：2018-09-07 10:02:43

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 fuse
```

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，您需要来手动安装 fuse 版本，具体命令如下：

```
# 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

1. 获取工具

Github 下载地址：[COSFS 工具](#)。

2. 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 /usr/cosfs
```

进入到该目录，编译安装：

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

3. 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。

bucketname 形如 bucketprefix-123456789, 更多关于 bucketname 的命名规范，请参见 [存储桶命名规范](#)。

示例：

```
echo bucketprefix-123456789:AKID8ILGzYjHMG8zhGtnIX7Vi4KOGxRqg1aa:LWVJqlagbFm8IG4sNIrkeSn5DLI3d  
CYi > /etc/passwd-cosfs  
chmod 640 /etc/passwd-cosfs
```

4. 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-bucketname your-mount-point -ourl=cos-domain-name -odbglevel=info
```

其中：

- 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 为存储桶对应的访问域名，形式为 `http://cos.<Region>.myqcloud.com`（适用于XML API），其中 `<Region>` 为地域简称，如：`ap-guangzhou`、`eu-frankfurt` 等。更多地域信息，请查阅 [可用地域](#)。
- -odbglevel 参数表示信息级别，可选 info、dbg，建议参照示例设置为“info”。

注意：

v1.0.5 版本之前的 cosfs 挂载命令：

```
cosfs bucketname_suffix:bucketname_prefix my-mount-point -ourl=my-cos-endpoint
```

v1.0.5 版本之前的配置文件格式是：

```
bucketname_prefix:<SecretId>:<SecretKey>
```

其中 `bucketname_suffix` 指的是 bucket 名称中的数字后缀，`bucketname_prefix` 指的是除数字后缀外的其他部分。例如 `bucketprefix-1253972369` 的 `bucketname_suffix` 为 `1253972369`，`bucketname_prefix` 为 `bucketprefix`。

示例：

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

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 bucketprefix-1253972369 /mnt -ourl=http://cos.ap-guangzhou.myqcloud.com -odbglevel=info -onoxat  
tr -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。

卸载存储桶示例：

```
fusermount -u /mnt
```

或者

```
umount -l /mnt
```

常用挂载选项

1. -omultipart_size=[size]

`multipart_size` 用来指定分块上传时，每个分块的大小，默认是 10 MB。由于分块上传对块的数目有最大限制（10000 块），所以对于大文件，例如超出 10 MB * 10000 (100 GB) 大小的文件，需要根据具体情况调整该参数。该参数单位是 MB。

2. -oallow_other

如果要允许其他用户访问挂载文件夹，可以在运行 COSFS 的时候指定 `allow_other` 参数。

3. -odel_cache

默认情况下，cosfs 为了优化性能，在 `umount` 后，不会清除本地的缓存数据。如果需要在 COSFS 退出时，自动清除缓存，可以在挂载时加入该选项。

4. -onoxattr

禁用 `get/setxattr` 功能，当前版本的 cosfs 不支持该功能，如果本地文件所在磁盘在挂载的时候使用了 `use_xattr` 选项，可能会导致 `mv` 文件到 bucket 失败。

局限性

COSFS 提供的功能、性能和本地文件系统相比，存在一些局限性。例如：

- 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。
- 不可以同时在一个挂载点上挂载、和卸载文件。您可以先 `cd` 切换到其他目录，再对挂载点进行挂载、卸载操作。

常见问题

1. 如何挂载目录？

您在挂载命令的时候，可以指定目录，命令如下：

```
cosfs my-bucket-name:/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。

如使用 v1.0.5 之前版本，则挂载命令为：

```
cosfs my-bucket-name-suffix:my-bucket-name-prefix:/my-dir /tmp/cosfs -ourl=http://cn-south.myqcloud.com -odbglevel=info -ouse_cache=/path/to/local_cache
```

2. 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。

3. 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

```
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，您需要来手动安装 fuse 版本，具体命令如下：

```
# 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
```

4. 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 不是被强制退掉，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。

5. 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

最近更新时间：2018-07-26 14:59:54

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。

使用环境

系统环境

Linux 或 Windows 系统。

软件依赖

Hadoop-2.7.2 及以上版本。

下载与安装

获取 hadoop-cos 插件

下载地址：[hadoop-cos 插件](#)

安装 hadoop-cos 插件

1. 将dep目录下的cos_hadoop_api-5.2.5.jar 和 hadoop-cos-2.7.2.jar 拷贝到 `$HADOOP_HOME/share/hadoop/tools/lib` 下。
2. 修改 hadoop_env.sh
在 `$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
```

使用方法

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.defaultFS</name>
<value>cosn://<bucket-appid> </value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>${HADOOP_PATH}/tmp</value>
</property>
<property>
<name>dfs.name.dir</name>
<value>${HADOOP_PATH}/name</value>
</property>
<property>
<name>fs.cosn.userinfo.secretId</name>
<value>xxxxxxxxxxxxxxxxxxxxxxxxxxxx</value>
<description>Tencent Cloud Secret Id </description>
</property>
<property>
<name>fs.cosn.userinfo.secretKey</name>
<value>xxxxxxxxxxxxxxxxxxxxxxxxxxxx</value>
<description>Tencent Cloud Secret Key</description>
</property>
<property>
<name>fs.cosn.userinfo.region</name>
<value>ap-xxx</value>
<description>The region where the bucket is located</description>
</property>
<property>
<name>fs.cosn.impl</name>
<value>org.apache.hadoop.fs.CosFileSystem</value>
<description>The implementation class of the CosN Filesystem</description>
</property>
<property>
<name>fs.AbstractFileSystem.cosn.impl</name>
<value>org.apache.hadoop.fs.CosN</value>
<description>The implementation class of the CosN AbstractFileSystem.</description>
</property>

<property>
<name>fs.cosn.buffer.dir</name>
<value>${hadoop.tmp.dir}/cos</value>
```

```

<description>The local path where the cosn filesystem should store files before write to cos.</description>
</property>
<property>
<name>fs.cosn.upload.buffer</name>
<value>disk</value>
<description>
There are two options to choose from:
"disk": Use some temporary files as buffer pool, and use the memory mapping technology to obtain a memor
y access speed for buffer IO approximately.
"memory": Use some direct byte buffers as buffer pool, and obtain a memory access speed for buffer IO.
Default is disk mode.
</description>
</property>
<property>
<name>fs.cosn.upload.buffer.size</name>
<value>134217728</value>
<description>The total size of the buffer pool</description>
</property>
<property>
<name>fs.cosn.block.size</name>
<value>8388608</value>
<description>Block size to use cosn filesystem, which is the part size for MultipartUpload. Considering the CO
S 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>10</value>
<description>The number of seconds to sleep between each COS retry.</description>
</property>
</configuration>
    
```

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。可通过 云 API 密钥 控制台 查看	无	是
fs.cosn.impl	cosn 对 FileSystem 的实现类，固定为 org.apache.hadoop.fs.CosFileSystem	无	是

属性键	说明	默认值	必填项
fs.AbstractFileSystem.cosn.impl	cosn 对 AbstractFileSystem 的实现类，固定为 org.apache.hadoop.fs.CosN	无	是
fs.cosn.userinfo.region	请填写您的地域信息，枚举值为 可用地域 中的地域简称，如 ap-beijing、ap-guangzhou 等	无	是
fs.cosn.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。如果 buffer 指定了 disk 类型，则该目录需要预留至少 fs.cosn.upload.buffer.size 大小的空间	/tmp/hadoop_cos	否
fs.cosn.upload.buffer	流式上传时，使用的缓冲区类型。当前支持两种缓冲区类型：disk 和 memory，其中 disk 将会在 fs.cosn.buffer.dir 选项指定的文件系统目录中生成若干个文件临时文件，并使用内存映射技术将其包装为上传缓冲池。内存较大机器建议可以使用 memory 类型的缓冲区，同时缓冲区的大小至少保证大于等于一个 block 的大小。	disk	否
fs.cosn.upload.buffer.size	向 COS 流式上传文件时，本地使用的缓冲区的大小。要求至少大于等于一个 block 的大小	134217728 (128MB)	否
fs.cosn.block.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，block size 为 8MB 时，最大能够支持 78GB 的单文件上传。block 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_thread_pool	文件流式上传到 COS 时，并发上传的线程数目	CPU 核心数 × 3	否
fs.cosn.maxRetries	访问 COS 出现错误时，最多重试的次数	3	否
fs.cosn.retry.interval.seconds	每次重试的时间间隔	3	否

使用示例

命令格式为：`hadoop fs -ls -R cosn://bucket-appid/<路径>` 或 `hadoop fs -ls -R /<路径>` (需要配置 `fs.defaultFS` 选项为 `cosn://bucket-appid`)，下例中以名称为 `hdfs-test-1252681929` 的 bucket 为例，可在其后面加上具体路径。

```
hadoop fs -ls -R cosn://hdfs-test-1252681929/
-rw-rw-rw- 1 root root 1087 2018-06-11 07:49 cosn://hdfs-test-1252681929/LICENSE
drwxrwxrwx - root root 0 1970-01-01 00:00 cosn://hdfs-test-1252681929/hdfs
drwxrwxrwx - root root 0 1970-01-01 00:00 cosn://hdfs-test-1252681929/hdfs/2018
-rw-rw-rw- 1 root root 1087 2018-06-12 03:26 cosn://hdfs-test-1252681929/hdfs/2018/LICENSE
-rw-rw-rw- 1 root root 2386 2018-06-12 03:26 cosn://hdfs-test-1252681929/hdfs/2018/ReadMe
drwxrwxrwx - root root 0 1970-01-01 00:00 cosn://hdfs-test-1252681929/hdfs/test
-rw-rw-rw- 1 root root 1087 2018-06-11 07:32 cosn://hdfs-test-1252681929/hdfs/test/LICENSE
-rw-rw-rw- 1 root root 2386 2018-06-11 07:29 cosn://hdfs-test-1252681929/hdfs/test/ReadMe
```

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

最近更新时间：2018-07-19 10:31:00

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，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，如：mybucket-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

(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 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，如：mybucket-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 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

最近更新时间：2018-10-16 17:02:02

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。实际上，在开发小程序的过程中，图片资源通常会占用较大空间，容易超出官方的 1MB 限制要求。

通过使用对象存储的 WeCOS 工具，小程序项目中的图片资源会自动上传到 COS 上，并且 WeCOS 可以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。

下面将介绍使用 WeCOS 的相关准备工作和安装操作等配置。

准备工作

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

开始安装

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，文件配置示例如下：

```
{
  "appDir": "./app",
  "cos": {
    "secret_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "secret_key": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "bucket": "wxapp-1251902136",
    "region": "ap-guangzhou", //创建 bucket 时选择的地域简称
    "folder": "/", //资源存放在 bucket 的哪个目录下
  }
}
```

```
}
}
```

配置项说明

配置项	类型	说明
appDir	String	默认 <code>./app</code> ，小程序项目目录
cos	Object	必填。COS 的存储桶、地域等配置信息，请查看 存储桶概述 文档，密钥等信息请在 云API密钥控制台 中获取查看

开始使用

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 <code>./wecos_backup</code> ，备份目录
uploadFileSuffix	Array	默认 <code>[".jpg", ".png", ".gif"]</code> ，图片上传后缀名配置
uploadFileBlackList	Array	默认 <code>[]</code> ，图片资源黑名单
replaceHost	String	默认 <code>"</code> ，把指定域名替换成 targetHost
targetHost	String	默认 <code>"</code> ，使用自定义域名

配置项	类型	说明
compress	Boolean	默认 <code>false</code> ，是否开启压缩图片
watch	Boolean	默认 <code>true</code> ，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。您可以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，若您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 [腾讯云数据万象](#) 的图片压缩功能。您在 [数据万象控制台](#) 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

如上述使用方式无法满足您的需求，WeCOS 还提供了定制化模块的配置操作，通过直接引用的方式，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