

腾讯云对象存储

开发者工具

产品文档



腾讯云

【版权声明】

©2015-2016 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
迁移工具	4
本地同步工具.....	11
批量删除工具.....	16
FTP Server工具.....	20
COS-Fuse工具.....	25
Hadoop工具.....	29
WeCOS 小程序瘦身工具.....	35

迁移工具

功能说明

本迁移工具支持从AWS S3, 阿里云 OSS和七牛等服务中迁移文件到COS对象存储；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对象存储。只适用于COS 4.0

使用环境

系统环境

Linux , Mac OS

软件依赖

工具运行平台为*nix，需要有Python2.6及以上环境，同时机器应该安装pip, gcc与python-dev。您可以使用系统自带的包管理器安装相关依赖。

在centos上，使用如下命令安装：

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

在ubuntu/debian上，使用如下命令安装：

```
sudo apt-get install python-pip python-dev gcc
```

使用方法

获取工具

Github项目地址：https://github.com/tencentyun/cos_migrate_tool

安装方法

推荐使用pip安装，安装pip的方法，可以参考[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
```

执行完上述命令后，可以尝试使用如下命令检测是否安装成功。

```
cos_migrate_tool -h
```

配置文件

配置文件模板, common 部分配置基本配置，workspace是上述的工作目录。source 部分配置数据源的信息，如果你想迁移oss到cos，该部分就是配置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 [链接](#)

```
[common]
workspace=/tmp/tmp6      # 目录
threads=20                # 并发数100
```

```
[source]
type=oss
accesskeyid=
accesskeysecret=
bucket=
endpoint=
```

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

迁移OSS中文件

```
[common]
workspace=/tmp/tmp6

[source]
type=oss
accesskeyid=      # oss accesskey id
accesskeysecret=  # oss accesskey secret
bucket=           # 桶名bucket
```

```
endpoint=                # oss endpoint oss-cn-beijing.aliyuncs.com

[destination]
type=cosv4
region=shanghai          # cos region shanghai, guangzhou
accesskeyid=              # cos accesskeyid
appid=                   # cos appid
accesskeysecret=          # cos accesskeysecret
bucket=sdktest            # cos bucket
prefix_dir=/dir21         # cos 目录名
```

迁移七牛中的文件

```
[common]
workspace=/tmp/tmp11

[source]
type=qiniu
accesskeyid=              # qiniu accesskeyid
accesskeysecret=          # qiniu accesskeysecret
bucket=                   # qiniu bucket
domain_url=               # qiniu 域名

[destination]
type=cosv4
region=shanghai          # cos region shanghai, guangzhou
accesskeyid=              # cos accesskeyid
appid=                   # cos appid
accesskeysecret=          # cos accesskeysecret
bucket=sdktest            # cos bucket
```

```
prefix_dir=/dir21          # cos 00000000000000000000000000000000
```

迁移S3中的文件

```
[common]
workspace=/tmp/tmp21

[source]
type=s3
accesskeyid=          # s3 0 accesskey id
accesskeysecret=      # s3 0 accesskey secret
bucket=               # s3 00000bucket0

[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      # 源url和目标url
timeout=3                          # http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，模版参考上文内容。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。只适用于COS 4.0

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，并与已成功的本地数据库进行比对和同步（上传或删除）。

使用环境

系统环境

Linux 或 Windows 系统

软件依赖

JDK 1.7 或 1.8

使用方法

获取工具包

下载链接：[工具包 cos_sync.zip](#)

解压缩工具包并进入工具包路径：

```
unzip cos_sync.zip && cd cos_sync
```

配置说明

配置文件位于工具包目录/conf/config.json：

```
{
  "appid"           : "xxxxxx",
  "secret_id"       : "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "secret_key"      : "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "bucket"          : "xxxxxx",
  "timeout"         : "60",
  "thread_num"      : "20",
  "delete_sync"     : "1",
  "daemon_mode"     : "0",
  "daemon_interval" : "60",
  "enable_https"    : "0",
  "region"          : "gz",

  "local_path"      : "/home/test/data",
  "cos_path"        : "/mysyncfolder"
}
```

名称	描述	有效值
app_id	需要进行操作的 APPID，可从控制台获取	APPID 数字
secret_id	APPID 对应的密钥 ID，可从控制台获取	字符串
secret_key	APPID 对应的密钥 Key，可从控制台获取	字符串
bucket	指定要同步的 bucket 名称，需要提前在控制台建立，参考 创建 Bucket 。	字符串
timeout	连接 COS 的超时时间，在网络状况不佳的情况下可以增大该值。 单位: 秒	数字

名称	描述	有效值
thread_num	并发的线程数，增大该值可以提高并发度，以及更快的上传速度。减小该值，会降低上传速度。	数字
delete_sync	删除本地文件时同步删除 COS 文件。 1：本地删除时同步删除远程文件；0：忽略本地已删除文件	数字
daemon_mode	后台进程模式运行。 1：循环运行同步工具； 0：运行一次后退出。	数字
daemon_interval	后台进程模式下，检查本地文件变化的时间间隔。 单位：秒	数字
enable_https	启用 https 进行传输。 1：使用 https | 0：使用 http	数字
region	bucket所在的区域。如华北天津(tj), 华东上海(sh), 华南广州(gz)	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 范例：C:\\document\\dir	字符串
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

使用软件

执行同步工具（Windows：请双击 start_cos_sync.bat）

```
sh start_cos_sync.sh
```

执行完成后，会输出创建和删除成功与失败的统计，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。

常见错误

如果发生同步失败的情况，请先查看log目录下的error日志。常见的错误返回码如下所示。

code: -3, connection timeout

连接到 COS 服务超时，请检查解析和端口是否正常。

检查 DNS 方法：如 region 为 sh，则本地执行 `dig sh.file.myqcloud.com` 查看获取到的 IP 是否为腾讯云的 IP，可以通过外部 ping 工具验证（例如[站长工具 - Ping](#)）。如果为腾讯云机房，应当解析为 10.\.*.* 的地址。其他 region 地区同理 *

检查端口方法：在 IP 解析正确的情况下，执行

```
telnet sh.file.myqcloud.com 80
```

查看是否会返回

```
Escape character is '^['.
```

，如果无返回请检查本地防火墙配置以及网络是否通畅。

```
code:-133, ERROR_CMD_BUCKET_NOTEXIST
```

请确认配置文件中region是否设置正确。区域与配置的对应关系请参照配置说明。

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

版本历史

- 2016.11.17 4.x JAVA同步工具版本上线

批量删除工具

功能说明

该工具可批量删除 COS 中指定路径下的所有文件和目录。只适用于COS 4.0

使用环境

系统环境

Linux、OS X 或 Windows 系统

软件依赖

Python 2.7.x

使用方法

获取工具包

□Github项目地址：https://github.com/tencentyun/cos4_rm_recursive

在下载路径下运行以下命令进行解压：

```
sudo tar -xzvf cos_rm_recursive.tar.gz
```

配置说明

修改配置文件 conf/config.json

```
{
```

```
"appid":10000202,
"secret_id":"*****",
"secret_key":"*****",
"region":"guangzhou",
"log_level":4,
"log_file_name":"del_file.log",
"dir_thread_num":2,
"file_thread_num":5,
"log_out_to_screen":1,
}
```

其中：

- appid、secret_id、secret_key：修改为自身项目对应信息，可从控制台获取
- region：园区: shanghai, guangzhou, tianjin, guangzhoup
- log_level：记录log级别，0：debug、1：info、2：warning、3：error
- log_file_name：记录log的文件名
- dir_thread_num：递归搜索文件夹线程数
- file_thread_num：删除文件线程数
- log_out_to_screen：log是否打印到屏幕,0:不打印到屏幕,1:打印到屏幕

修改配置文件 conf/bucketlist.txt

此配置文件标识需要删除的bucket名称及文件夹，文件夹必须以/结尾，有类似如下结构：

```
testbucket, /folder1/
testbucket, /folder2/
```

说明

1. 可以直接填写根目录/，即bucketName/，表示删除此bucket下的所有文件及文件夹
2. 删除指定目录时，目录本身也会被删除
3. 注意文件夹之间不嵌套,比如folder1,folder2,folder3不能有嵌套关系

使用软件

执行删除命令：

```
python rm_recursive.py
```

结果返回

执行完成后，屏幕上以及日志文件中会显示如下关键信息：

```
$/rm_recursive.py
delete task started .....
delete task ended

rm task finished
timecost:5.16713190079 (s)
delfilenum=0,deldirnum=0,delfilefailnum=0,deldirfailnum=0
```

常见问题

- 由于COS系统文件夹删除机制，当文件夹下存在文件时，则文件夹本身删除会失败，一旦出现文件删除失败，则该文件的所有父目录均会删除失败。
- COS

系统本身存在操作频率控制，因此当线程数设置过大时，容易出现触发频控，删除工具会屏幕上打印

ERROR_PROXY_FREQ_CONTROL

，需要适当减少删除文件线程数量。

FTP Server工具

COS FTP V4 用于通过FTP协议往COS上传和下载文件.只适用于COS 4.0

功能说明

COS FTP v4是依赖COS 4.X的FTP服务端工具。通过COS FTP工具可以通过FTP协议从COS上传和下载文件。

上传机制：上传先落地FTP的本地磁盘，

上传COS后删除，返回给客户端成功。（后期版本流式上传COS，不落本地磁盘）

下载机制：下载直接流式返回给客户端

支持FTP命令

- put
- get
- mput
- mget
- delete
- mkdir
- ls
- cd
- bye
- quit
- size

不支持FTP命令

- append

适用COS版本

4.x

系统要求

Linux (推荐腾讯云Centos系列CVM)

依赖库

针对Centos等使用yum安装的系统，build.sh会自动下载以下依赖。其他系统请自行安装。

cmake
boost
openssl-devel
asio-devel
libidn-devel

使用方法

获取程序包

下载地址：[COS FTP V4 github](#)

源码结构

目录	说明
bin	编译后生成的可执行程序
conf	配置文件目录
data	上传时存储临时数据的目录，上传成功或失败后删除
dep	FTP Server所依赖的COS 4.X CPP SDK
include	头文件所在的目录
lib	依赖的库目录
log	日志目录

目录	说明
opbin	有关清理日志, 自动拉起的脚本
src	FTP源码目录
build.sh	编译脚本
start.sh	启动FTP server的脚本
stop.sh	停止程序的脚本

配置

配置文件conf/vsftpd.conf中的是vsftpd的相关配置,可以参考以下配置说明

1. COS

#cos, set your app info in cos

```
cos_appid=1000000
cos_secretid=xxxxxxxxxxxxxxxxxxxxxxxxxxxx
cos_secretkey=xxxxxxxxxxxxxxxxxxxxxx
# bucket bucket bucket (gz), (sh), (tj)
cos_bucket=test
cos_region=gz
# domain cos cos ( )
# domain cdn cdn ( )
cos_download_domain=cos
# , build.sh
cos_user_home_dir=/home/test
/cosftp_data/
```

2. FTP (- : : .)

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

3. IP, IP IP (IP FTP FTP CVM IP)

```
pasv_address=115.115.115.115
```

```
4. iptables -F, iptables -I (iptables 1025 ~ 65535  iptables)
listen_port=2121
```

编译

1. 因为FTP需要使用本地磁盘，因此请将FTP源码程序放在一个存储空间较大的盘。(腾讯云初始的机器购买的数据盘需要手动格式化并挂载, 请参考 <https://www.qcloud.com/doc/product/213/2974>)
2. 以root身份运行build.sh
(因为build.sh里会调用yum进行安装依赖库，推荐使用腾讯云主流的Centos系列系统，如果是其他系统，如ubuntu，请修改opbin/env_init.sh)

运行

```
1. cd cos_ftp (cd build.sh), su cos_ftp
2. sh start.sh (FTP monitor, CT)
3. FTP server (2121) pasv
FTP (yum install ftp) FTP 127.0.0.1 2121
4. FTP
```

停止

```
sh stop.sh
```

常见问题

1. 如何查看 vsftpd 的监听端口 (netstat -tulnp | grep vsftpd)
2. 如何查看 FTP 服务的日志文件
3. 如何配置 FTP 服务使用 COS 作为存储后端 (在 FTP 的 data 目录下创建 COS 的配置文件)
4. 如何配置 FileZilla 使用 append 模式写入文件 (在 FileZilla 的 settings.xml 文件中配置 append 模式)
5. 如何查看 log 文件

COS-Fuse工具

功能说明

COSFS 是基于 libfuse 实现的一个用户态的文件系统，通过 COSFS 可以将远程的腾讯云 COS 挂载在用户机器上，像访问本地磁盘一样访问腾讯云 COS，用户或者程序操作本地文件系统相当于直接操作腾讯云 COS。目前仅支持文件读取。只适用于COS 4.0

使用环境

系统环境

Linux

依赖动态库

dl z ssl crypto stdc++ pthread

静态库(程序包已携带)

cos sdk fuse curl jsoncpp

使用方法

获取程序包

当前版本：COSFS-v4.2.0 [点击下载](#)

编译

```
cmake .  
make
```

配置文件 cosfs_config.json

主要是 cosfs 依赖的 COS SDK 使用，参数说明：

```
"AppID":1251668577,
"SecretID":"AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO",
"SecretKey":"FZjRSu0mJ9YJijVXXY57MAdCl4uylaA7",
"Region":"sh",                      // 指定 COS 桶所在区域 URL 地址
"SignExpiredTime":360,               // 签名过期时间
"CurlConnectTimeoutInms":10000,     // CURL 连接超时时间
"CurlGlobalConnectTimeoutInms":360000, // CURL 全局连接超时时间
"UploadSliceSize":1048576,          // 上传切片大小 512k, 1M, 2M, 3M (默认为 512k)
"IsUploadTakeSha":0,                // 是否上传时计算 sha 值
"DownloadDomainType":2,
// 下载域名类型 1: cdn, 2: cos, 3: innercos, 4: self domain
"SelfDomain":"","                  // 是否使用 self domain
"UploadThreadPoolSize":5            // 上传线程池大小
"AsynThreadPoolSize":2              // 异步线程池大小
"LogoutType":0                      // 退出类型 0: 正常, 1: 异常, 2: 异常 syslog
```

一般只需要修改如下COS信息参数

AppID、SecretID、SecretKey、Region、DownloadDomainType

运行

(1) 挂载 bucket 到本地

```
./cosfs bucket /mnt/mointpoint/ -o cfgfile=cos_config.json
```

```
// /mnt/mointpoint/
// bucket COS bucket
```

(2) 挂载 bucket 下的某个目录到本地

```
./cosfs bucket:/folder /mnt/mointpoint/ -o cfgfile=cosfs_config.json
```

```
// /mnt/mointpoint/
// bucket/folder COS bucket
```

卸载

先要安装 fuse(yum install fuse) ,才能执行

```
fusermount -u /mnt/mointpoint/
```

如果提示设备忙,则需退出 /mnt/mointpoint/ ,如果还是无法卸载,则执行

```
umount -l /mnt/mointpoint/
```

备注：

目前支持读相关操作,暂时不支持写操作：

linux命令: cd, ls, ll, cat, more, cp

系统接口: open(), read(), stat(), close(), lseek()

Hadoop工具

功能说明

可以使用 Hadoop 大数据处理引擎，如 MapReduce，Hive 处理存储在腾讯云对象存储 COS 上的数据。只适用于COS 4.0

实现机制：Hadoop cosn 插件实现了以腾讯云 COS 作为底层存储的文件系统运行上层的计算任务，如 MapReduce、Hive、Spark、Tez 等。cosn 支持以任意形式上传到 COS 上的数据被 Hadoop 处理。

使用环境

系统环境

Linux、Windows 系统

软件依赖

Hadoop-2.7.2 及以上

使用方法

获取 cos-java-sdk

下载地址：<https://github.com/tencentyun/cos-java-sdk-hadoop-v4>

在下载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：<https://github.com/tencentyun/hadoop-cosn-v4>

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

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
```

将上两步获取的 cos_api-4.2.jar 和 hadoop-cos-2.7.2.jar 拷贝到 \$HADOOP_HOME/share/hadoop/tools/lib 下，同时需要拷贝 SDK 依赖的 json-20140107.jar 和 httpmime-4.2.5.jar 到该目录（这两个 JAR 包可在本地的 maven 仓库下找到）。

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>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</value>
</property>
<property>
  <name>fs.cos.userinfo.secretKey</name>
  <value>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</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>
</configuration>
```

注意其中含有 COS 的几个属性：

- fs.cos.userinfo.appid 属性 填写您使用的 COS 账户 APPID
- fs.cos.userinfo.secretId/secretKey 属性 填写您账户的密钥信息
- 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

使用软件

使用 hadoo fs 常用命令

```
[rabbitliu@VM_83_1_centos bin]$ hadoop fs -ls cosn://rabbitliu/
Found 7 items
-rw-rw-rw-    1 rabbitliu rabbitliu          3669 2016-10-25 21:23 cosn:/
/rabbitliu/b.txt
drwxrwxrwx    - rabbitliu rabbitliu
              0 1970-01-01 08:00 cosn://rabbitliu/dir1
drwxrwxrwx    - rabbitliu rabbitliu
              0 1970-01-01 08:00 cosn://rabbitliu/mr
-rw-rw-rw-    1 rabbitliu rabbitliu        16952 2016-10-25 21:37 cosn:/
/rabbitliu/qcloud_sign.proto
-rw-rw-rw-    1 rabbitliu rabbitliu         2048 2016-10-25 21:48 cosn:/
/rabbitliu/rabbit_test2K.txt
-rw-rw-rw-    1 rabbitliu rabbitliu    52428800 2016-10-27 16:40 cosn:/
/rabbitliu/rabbit_test50MB.txt
drwxrwxrwx    - rabbitliu rabbitliu
              0 1970-01-01 08:00 cosn://rabbitliu/xx1
```

运行 MapReduce 自带的 wordcount

```
bin/hadoop jar s
```

```
hadoop/hadoop/mapreduce/hadoop-  
mapreduce-examples-2.7.2.jar wordcount cosn://rabbitliu/mr  
/input cosn://rabbitliu/mr/output3
```

执行成功会返回一些统计信息如下：

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
```

WeCOS 小程序瘦身工具

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，从而减小小程序包大小，解决包大小超过限制的烦恼。

为什么需要 WeCOS

□□□□□□□□□□□□□□□□□□□□□□□□ 1MB □□□ 1MB □□□□□□□□□□□□

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。这时候，使用 WeCOS，可以让你在开发过程中不需要关心图片资源占用多少空间的问题，专注于自己的逻辑开发。

准备工作

- 进入 [腾讯云官网](#)，注册帐号
- 登录 [云对象存储服务（COS）控制台](#)，开通COS服务，创建Bucket
- 下载 [WeCOS 工具](#)
- 安装 [Node.js](#) 环境

安装

```
npm install -g wecos
```

基本配置

在你的小程序目录同级下创建

wecos.config.json

文件

wecos.config.json

配置项例子：

```
{
  "appDir": "./app",
  "cos": {
    "appid": "1234567890",
    "bucketname": "wxapp",
    "folder": "/", //□□□□□bucket□□□□□
    "region": "wx", //□□bucket□□□□□□□□
    "secret_key": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "secret_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}
```

配置项	类型	说明
appDir	[String]	默认 ./app , 小程序项目目录
cos	[Object]	必填，填写需要上传到COS对应的配置信息，部分信息可在 COS控制台 查看

使用

在配置文件同级目录下命令行执行

wecos

注意，执行前需要在该目录下创建

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

配置项	类型	说明
		, 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，因此我们也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

为了方便使用，可以通过以下配置来修改备份目录名，如果不需要使用该功能，可以设置为空值

```

"backupDir": "./wecos_backup"

```

设置图片后缀

有些时候，我们需要限制上传图片的格式，例如只允许

jpg

格式，可以通过 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，假如你还需要添加其他格式，例如 webp，可以在该配置项中添加

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

设置图片黑名单

开发过程中，某些特定的图片我们不希望被上传，可以通过 WeCOS 的黑名单配置来解决这个问题，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

如果希望 COS 文件链接使用自定义的域名，可以配置 targetHost 代替默认域名，可以省略：

http://

:

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

如果代码中的图片链接想换一个域名，可以配置 `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，访问速度也会影响到用户体验

令人激动的是，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。

首先，你需要在[万象优图控制台](#)创建 COS 的同名bucket

然后，开启该选项，资源将被压缩后上传（注：如果图片已经小到一定程度，压缩后大小可能不会变化）

```
"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，如果觉得实时监听不方便，或者只需要一次性处理就停止，可以修改该配置，程序将只会执行一次后退出

```
"watch": false
```

高级用法

如果你除了上述使用命令行来执行的方式外，还想使用其他方式，例如定制化成自己的模块，我们也提供了直接引用的使用方法满足你个性化的需求

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

```
/**
 * option [] [String|Object]
 * [] String[]
 * [] Object[]
 */
wecos([option]);
```

```
//[]
wecos('./wecos-config.js');
```

```
//[]
wecos({
  appDir: './xxx',
  cos: {
    ...
  }
})
```

```
});
```

相关

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