

对象存储
工具指南
产品文档



腾讯云

【版权声明】

©2013-2024 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默示的承诺或保证。

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

- COSBrowser 工具

 - COSBrowser 简介

 - 桌面端使用说明

 - 移动端使用说明

 - 安装与登录

 - 移动端功能列表

 - 存储桶管理与操作

 - 文件管理与操作

 - 数据监控

- COSCLI 工具（公测）

 - COSCLI 简介

 - 下载与安装配置

 - 通用选项

 - 常用命令

 - 生成与修改配置文件 - config

 - 创建存储桶 - mb

 - 删除存储桶 - rb

 - 存储桶标签 - bucket-tagging

 - 查询存储桶或文件列表 - ls

 - 获取不同类型文件的统计信息 - du

 - 上传下载或拷贝文件 - cp

 - 同步上传下载或拷贝文件 - sync

 - 删除文件 - rm

 - 获取文件哈希值 - hash

 - 列出分块上传中产生的碎片 - lsparts

 - 清理碎片 - abort

 - 取回归档文件 - restore

 - 获取预签名 URL - signurl

 - 常见问题

COSCMD 工具

COS Migration 工具

FTP Server 工具

Hadoop 工具

COSDistCp 工具

Hadoop-cos-DistChecker 工具

HDFS TO COS 工具

在线辅助工具

- COS 请求工具

自助诊断工具

工具指南

工具概览

最近更新时间：2024-01-06 10:43:44

工具	功能说明
COSBrowser 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、生成访问链接等操作。
COSCLI 工具	COSCLI 是腾讯云对象存储提供的客户端命令行工具。通过 COSCLI 工具，您可以通过简单的命令行指令对您 COS 中的对象（Object）实现批量上传、下载、删除等操作。
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 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	Hadoop 工具可以为 Hadoop、Spark 以及 Tez 等大数据计算框架集成 COS 提供支持，便于读写存储在 COS 上的数据。
COSDistCp 工具	COSDistCp 是一款基于 MapReduce 的分布式文件拷贝工具，主要用于 HDFS 和 COS 之间的数据拷贝。
Hadoop-cos-DistChecker 工具	Hadoop-cos-DistChecker 是一个在使用 hadoop distcp 命令从 HDFS 迁移数据到 COS 上后，用于校验迁移目录完整性的工具。
HDFS TO COS 工具	本工具用于将 HDFS 上的数据拷贝到 COS 上。
自助诊断工具	自助诊断工具是腾讯云对象存储为用户提供的 Web 工具，可用于错误请求的自助诊断排查。

上传能力

各工具对于上传能力的支持情况如下表所示：

工具	简单上传	分块上传	断点上传	高级上传	一致性校验	生成预签名 URL

COSBrowser 工具	支持	支持	支持	支持，默认 8MB触发分块上传	支持 md5 校验	支持文件下载
COSCLI 工具	支持	支持	支持	支持，可自行分块触发阈值	支持 crc64 校验	支持
COSCMD 工具	支持	支持	支持	支持，默认 10MB触发分块上传	支持 md5 校验	支持
COS Migration 工具	支持	支持	支持	支持，可自行分块触发阈值	支持 md5 校验	不适用
FTP Server 工具	支持	支持	支持	支持，可自行分块触发阈值	不支持	不适用
COSFS 工具	支持	支持	支持	支持，可自行分块触发阈值	支持 md5 校验	不适用
Hadoop 工具	支持	支持	不支持，HDFS 协议冲突	支持，可自行分块触发阈值	支持 crc64、crc32 校验	不适用
COSDistCp 工具	支持	支持	不支持，HDFS 协议冲突	支持，可自行分块触发阈值	支持文件大小、crc64、crc32 校验	不适用
HDFS TO COS 工具	支持	支持	不支持，HDFS 协议冲突	支持，可自行分块触发阈值	支持文件名称和大小校验	不适用

说明

高级上传对简单上传和分块上传进行封装，支持根据文件大小选择上传方式。

环境安装与配置

Java 安装与配置

最近更新时间：2024-01-06 10:43:44

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

Windows

1. 下载 JDK

进入 [Oracle 官方网站](#) 下载合适的 JDK 版本。

2. 安装

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

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

3. 配置

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

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

变量名(N)：**CLASSPATH** 变量值

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

变量名(N)：**Path**

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

4. 测试

测试配置是否成功：单击【开始】（或快捷键：Win+R）>【运行】（输入 `cmd`）>【确定】（或按 Enter 键），输入命令 `javac` 并回车。若出现用法格式和命令参数等信息，则说明环境变量配置成功。

Linux

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

1. 下载 JDK

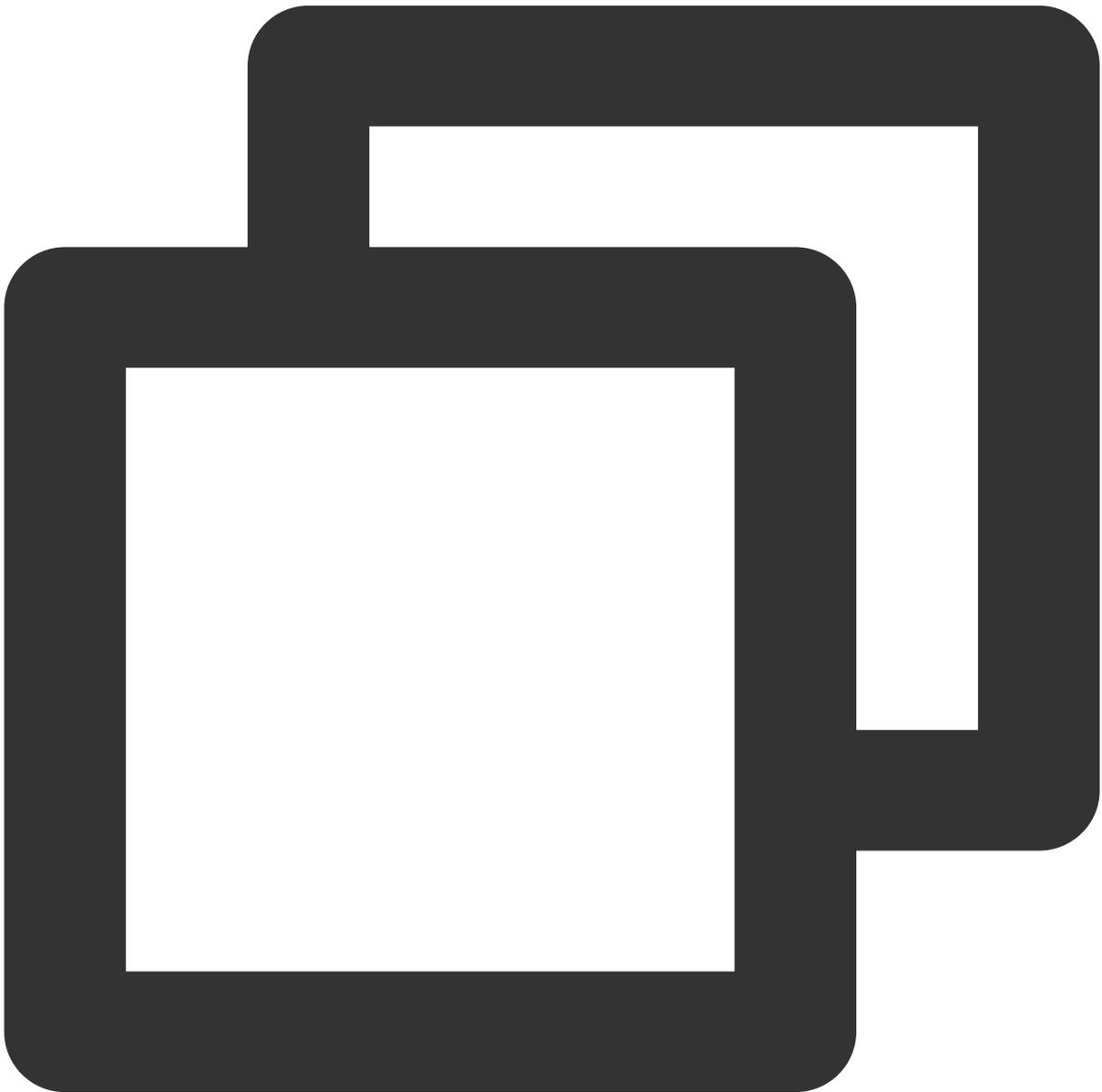
进入 [Oracle 官方网站](#) 下载合适的 JDK 版本，准备安装。

注意

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

2. 创建目录

执行如下命令，在 `/usr/` 目录下创建 `java` 目录。

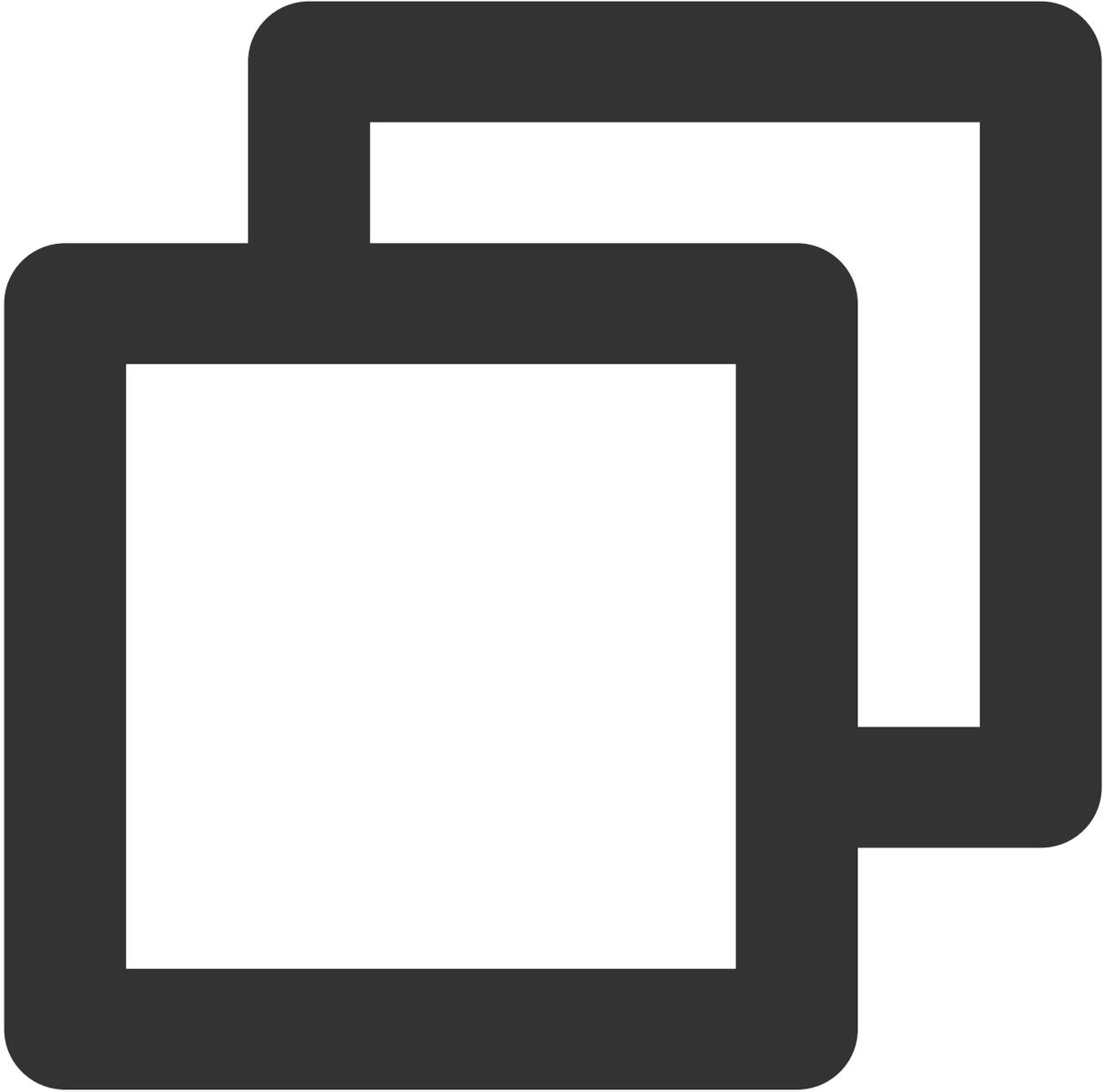


```
mkdir /usr/java  
cd /usr/java
```

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

3. 解压 JDK

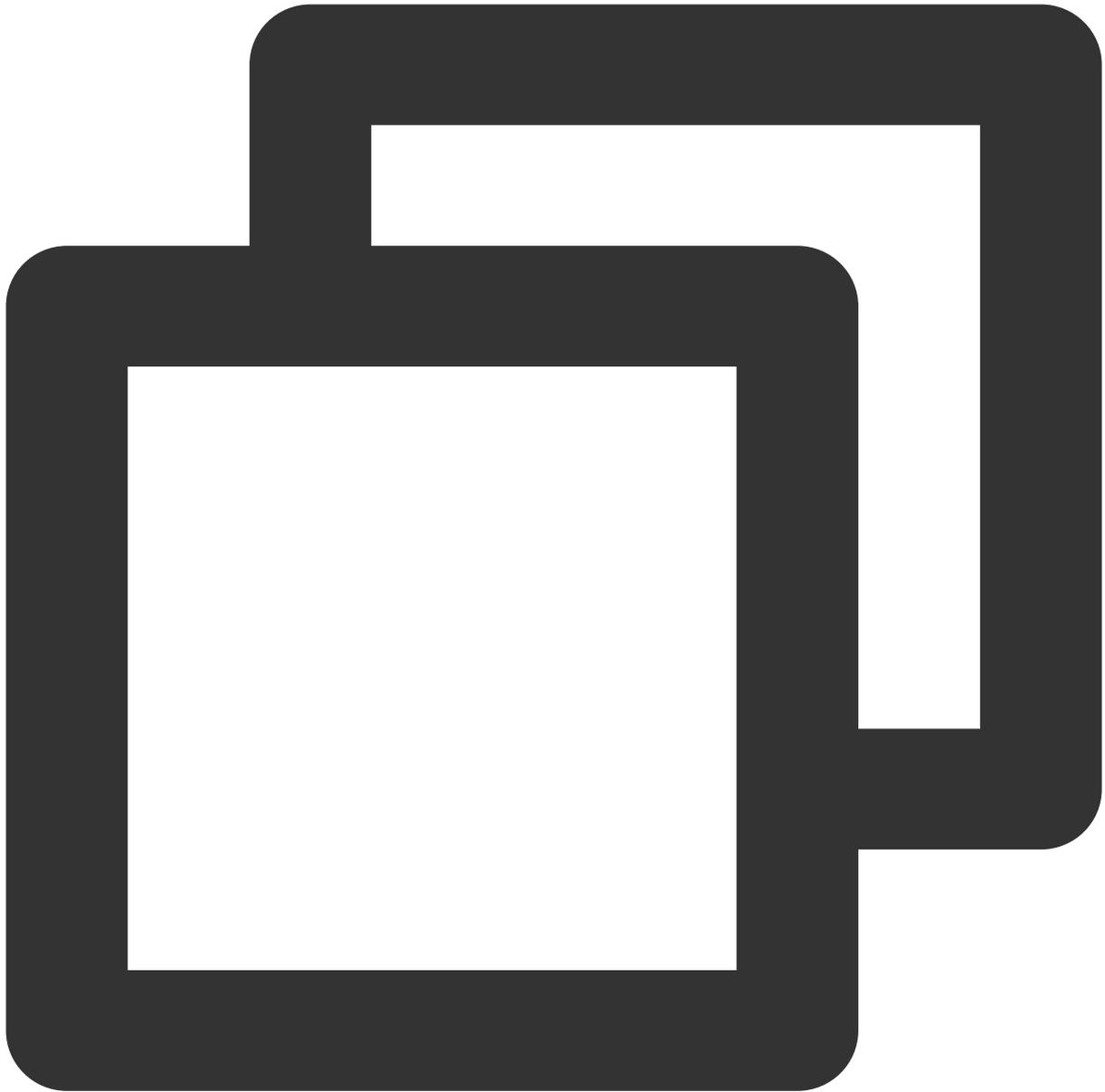
执行如下命令，解压文件。



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

4. 设置环境变量

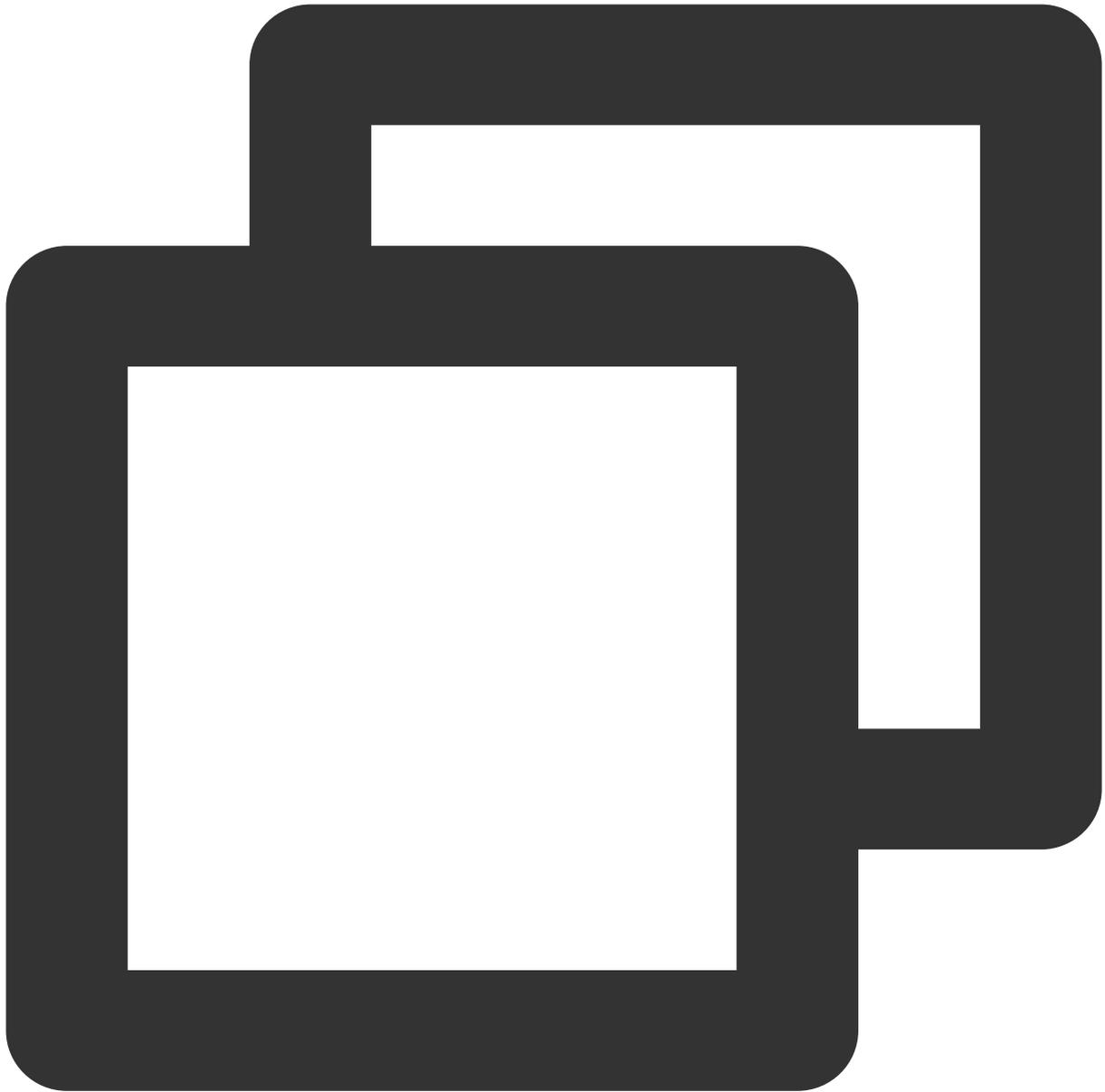
编辑 `/etc/profile` 文件，在 `profile` 文件中添加如下内容并保存：



```
# set java environment
JAVA_HOME=/usr/java/jdk1.8.0_151
JRE_HOME=/usr/java/jdk1.8.0_151/jre
CLASS_PATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/lib
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
export JAVA_HOME JRE_HOME CLASS_PATH PATH
```

注意

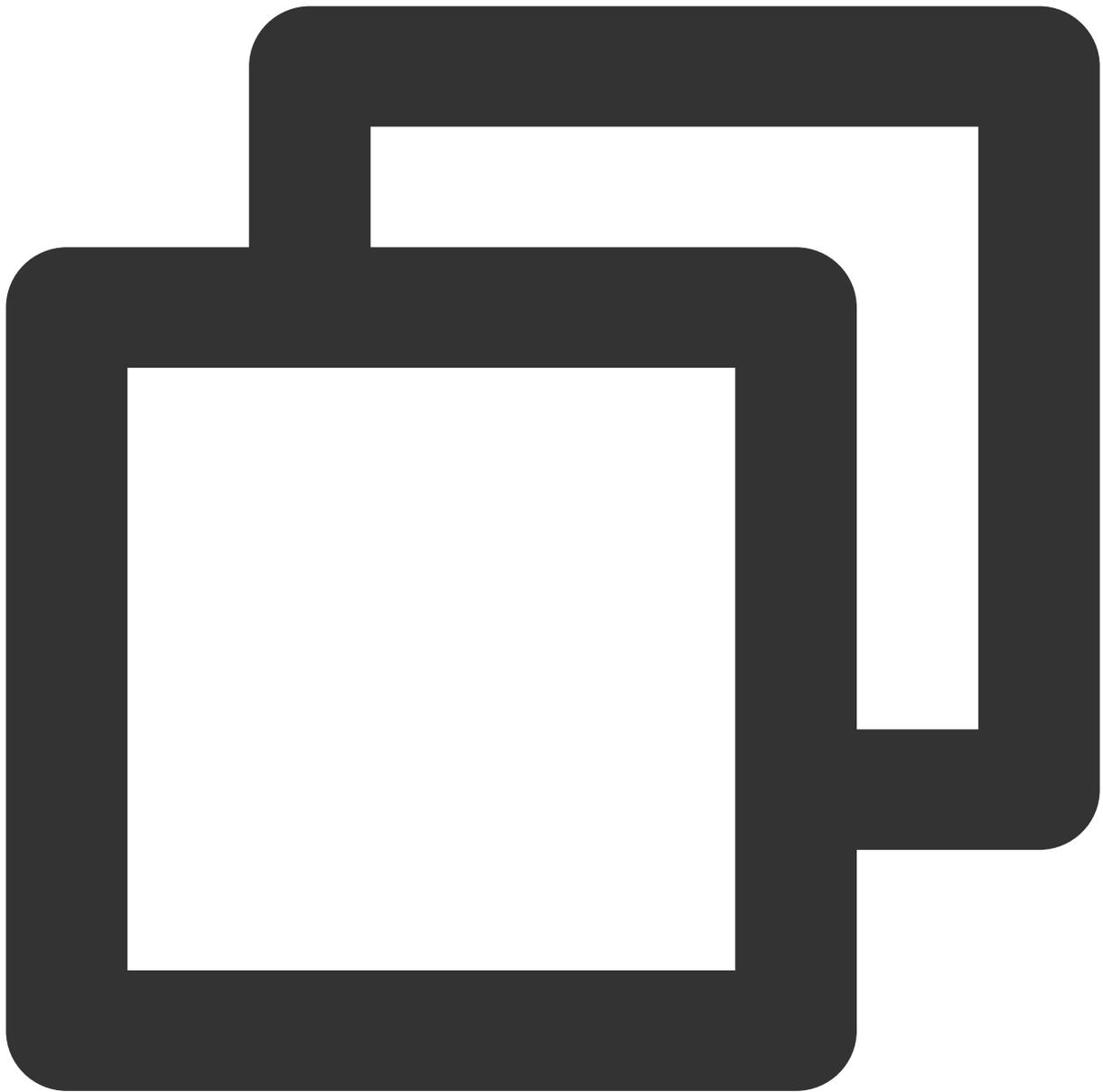
其中 `JAVA_HOME`、`JRE_HOME` 请根据自己的实际安装路径及 JDK 版本配置。
使之修改生效，执行如下：



```
source /etc/profile
```

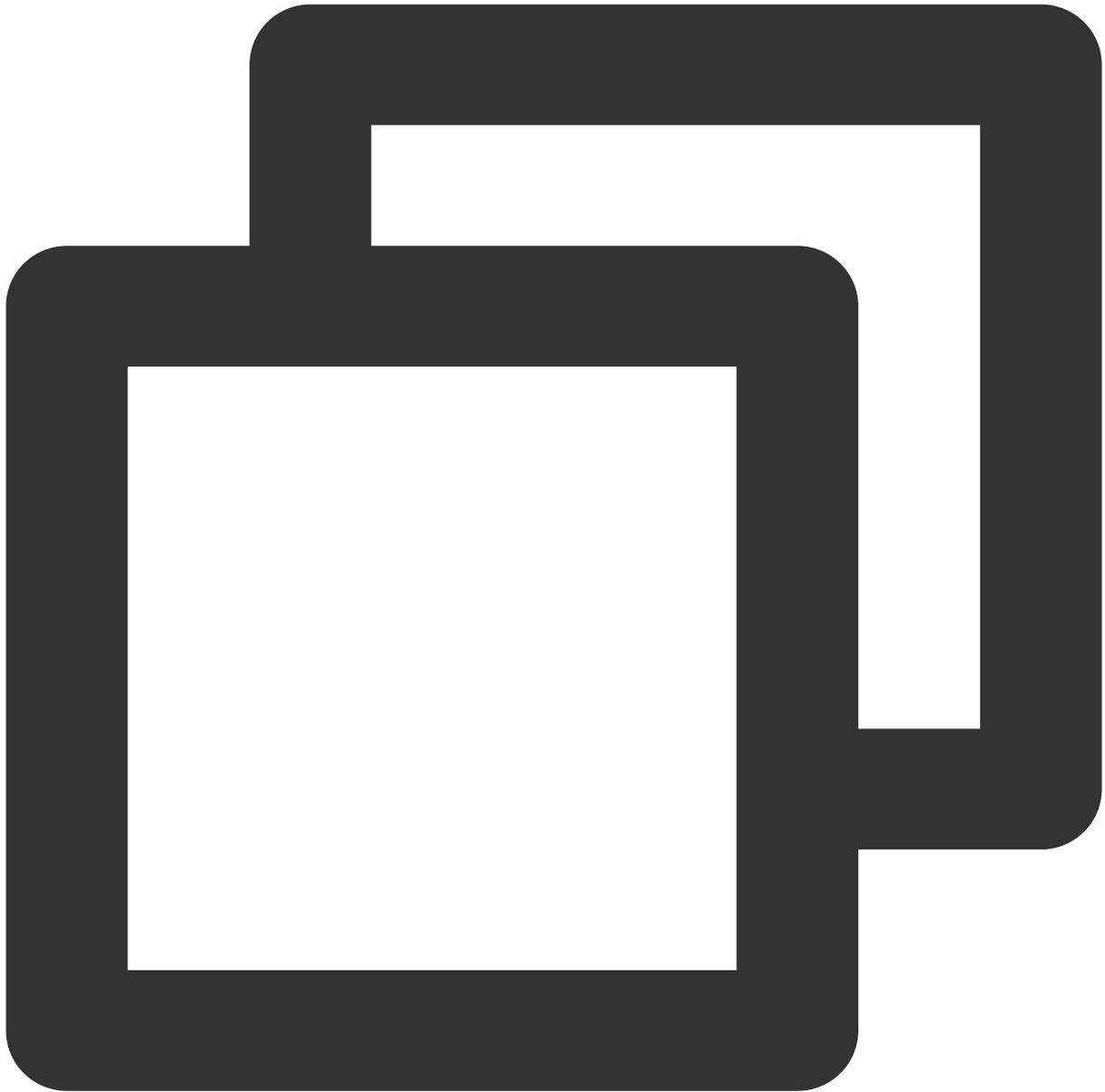
5. 测试

执行如下命令进行测试。



```
java -version
```

若显示 Java 版本信息，则说明 JDK 安装成功：



```
java version "1.8.0_151"  
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)  
Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)
```

Python 安装与配置

最近更新时间：2024-01-06 10:43:44

本文档简要介绍在不同操作系统下，Python 开发环境的安装方法。

通过安装包安装

1. 下载

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

注意

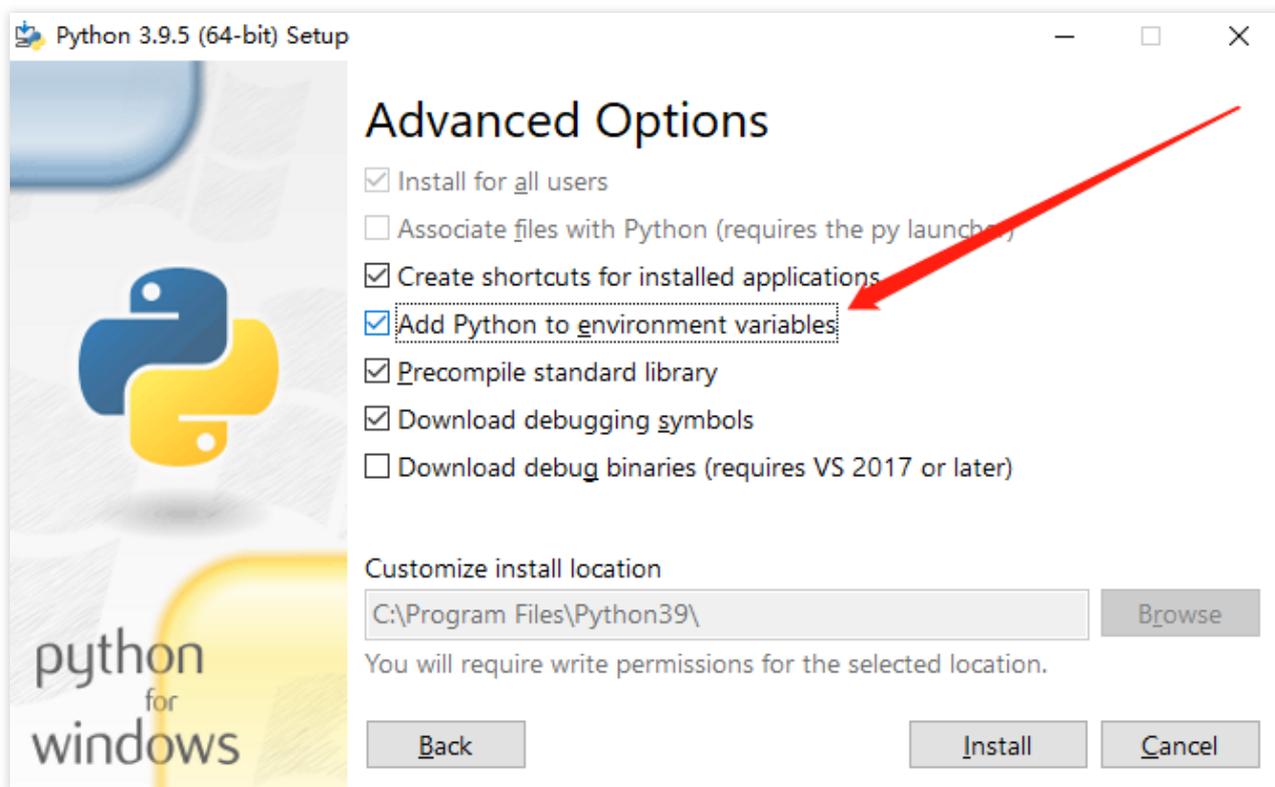
Python 官方已于2020年1月1日停止对 Python2 的维护，建议安装使用 Python3。

2. 安装

下载安装包后，根据安装包的引导即可完成 Python 开发环境的安装。

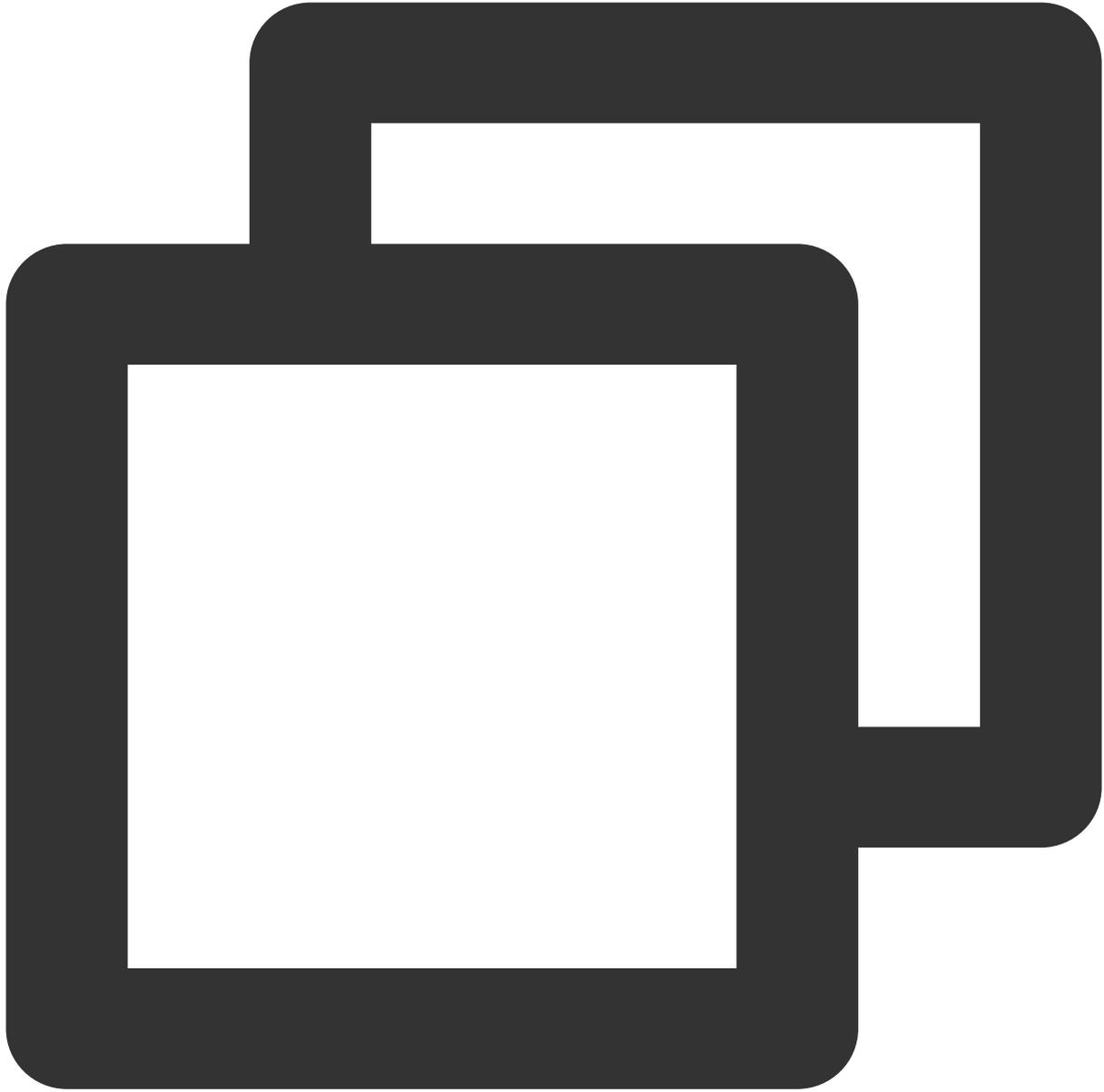
说明

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



3. 验证

在终端中执行以下命令查看 Python 版本：



```
python -V
```

若终端输出 Python 版本号则证明安装成功。

说明

Windows 系统用户在安装完毕后，可能需要重启计算机。

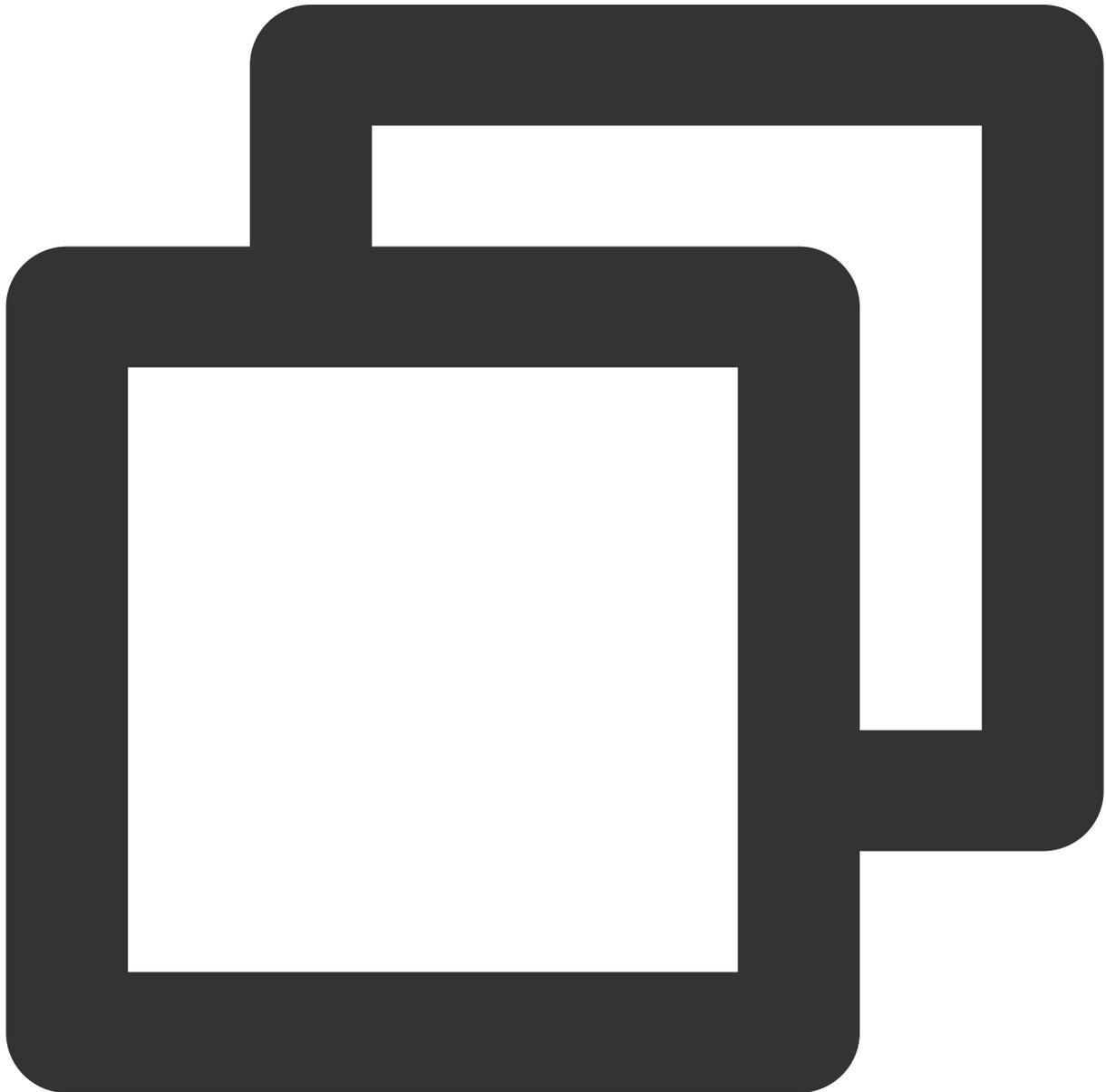
4. 环境变量配置

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

通过包管理器安装

Mac OS

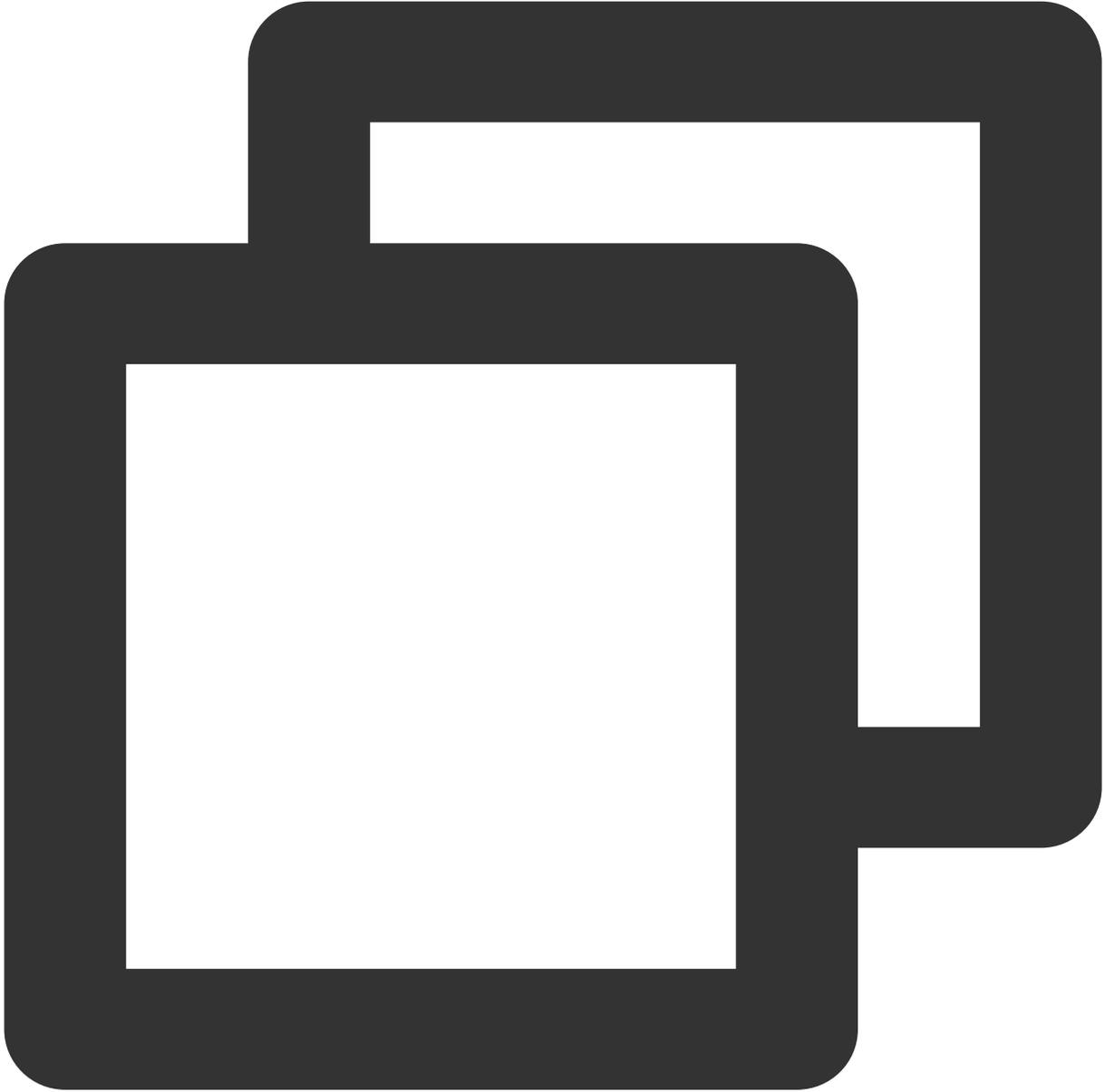
使用 Mac OS 的用户可以先安装 [HomeBrew](#)，再通过 HomeBrew 来安装 Python：



```
brew install python
```

Ubuntu

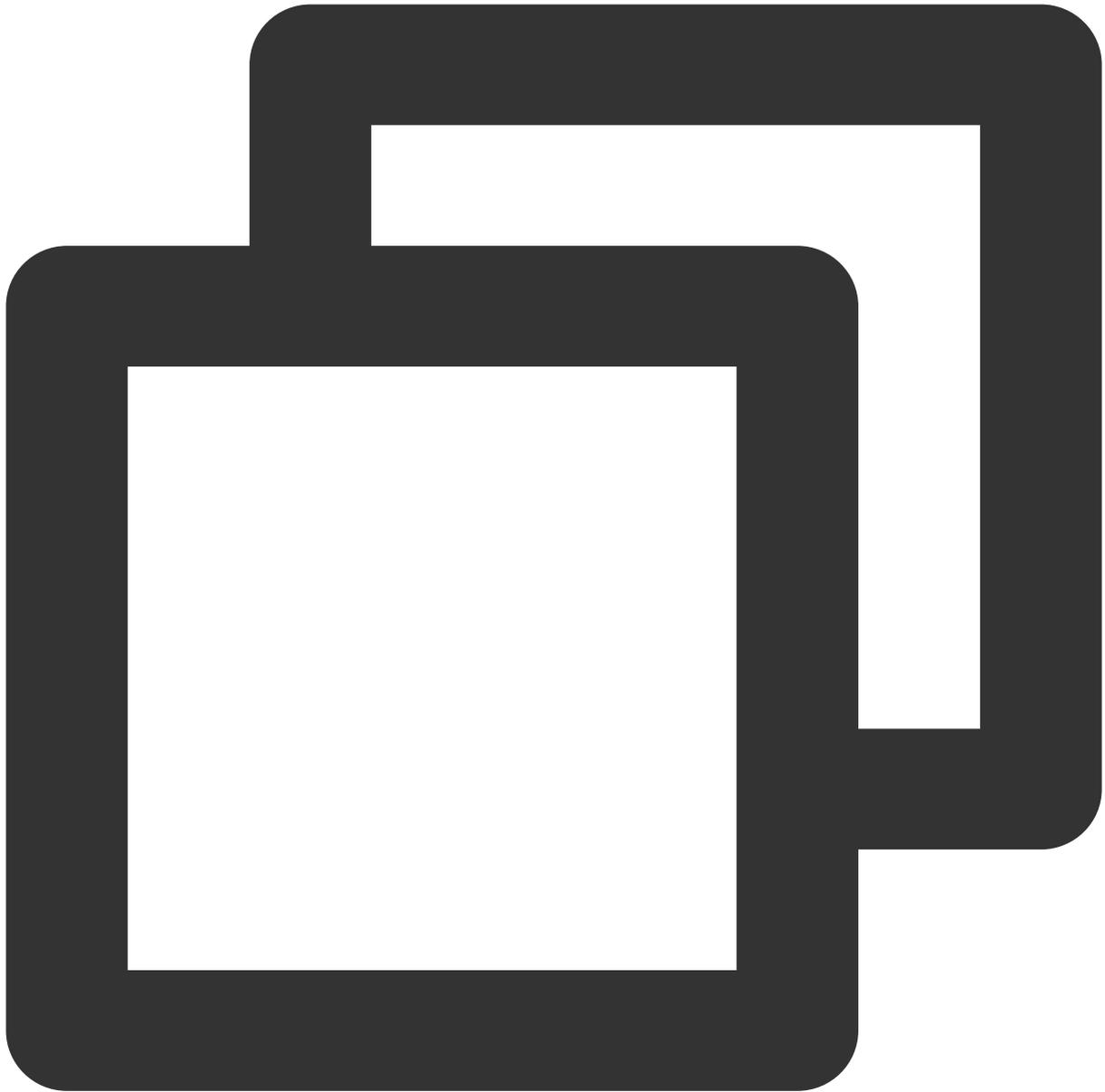
使用 Ubuntu 的用户可以使用 Ubuntu 自带的 apt (Advanced Packaging Tool) 包管理器来安装 Python :



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

CentOS

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



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

Hadoop 安装与测试

最近更新时间：2024-01-06 10:43:44

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

准备环境

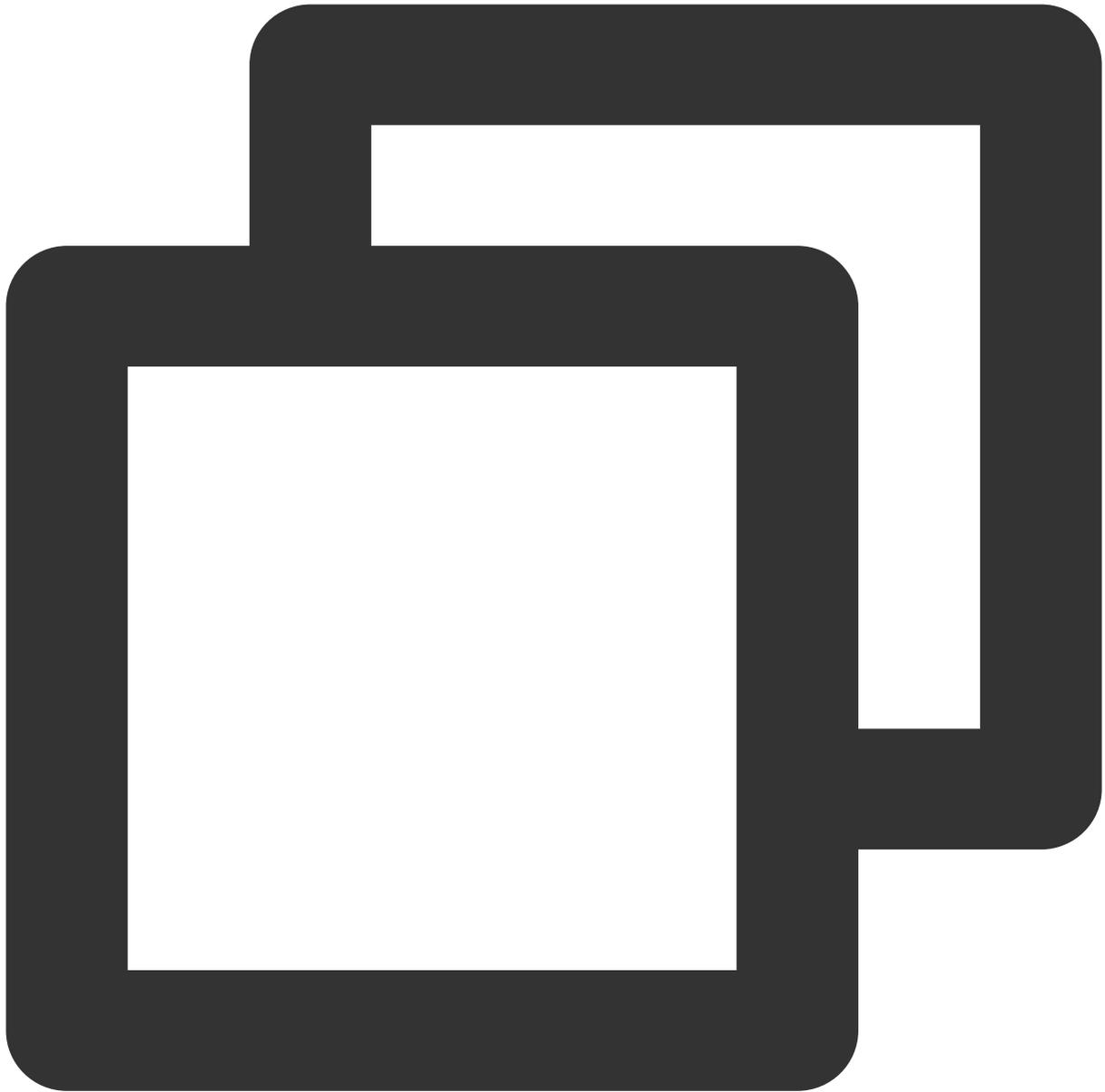
1. 准备若干台机器。
2. 安装配置系统：[CentOS-7-x86_64-DVD-1611.iso](#)。
3. 安装 Java 环境，具体操作请参见 [Java 安装与配置](#)。
4. 安装 Hadoop 可用包：[Apache Hadoop Releases Download](#)。

网络配置

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

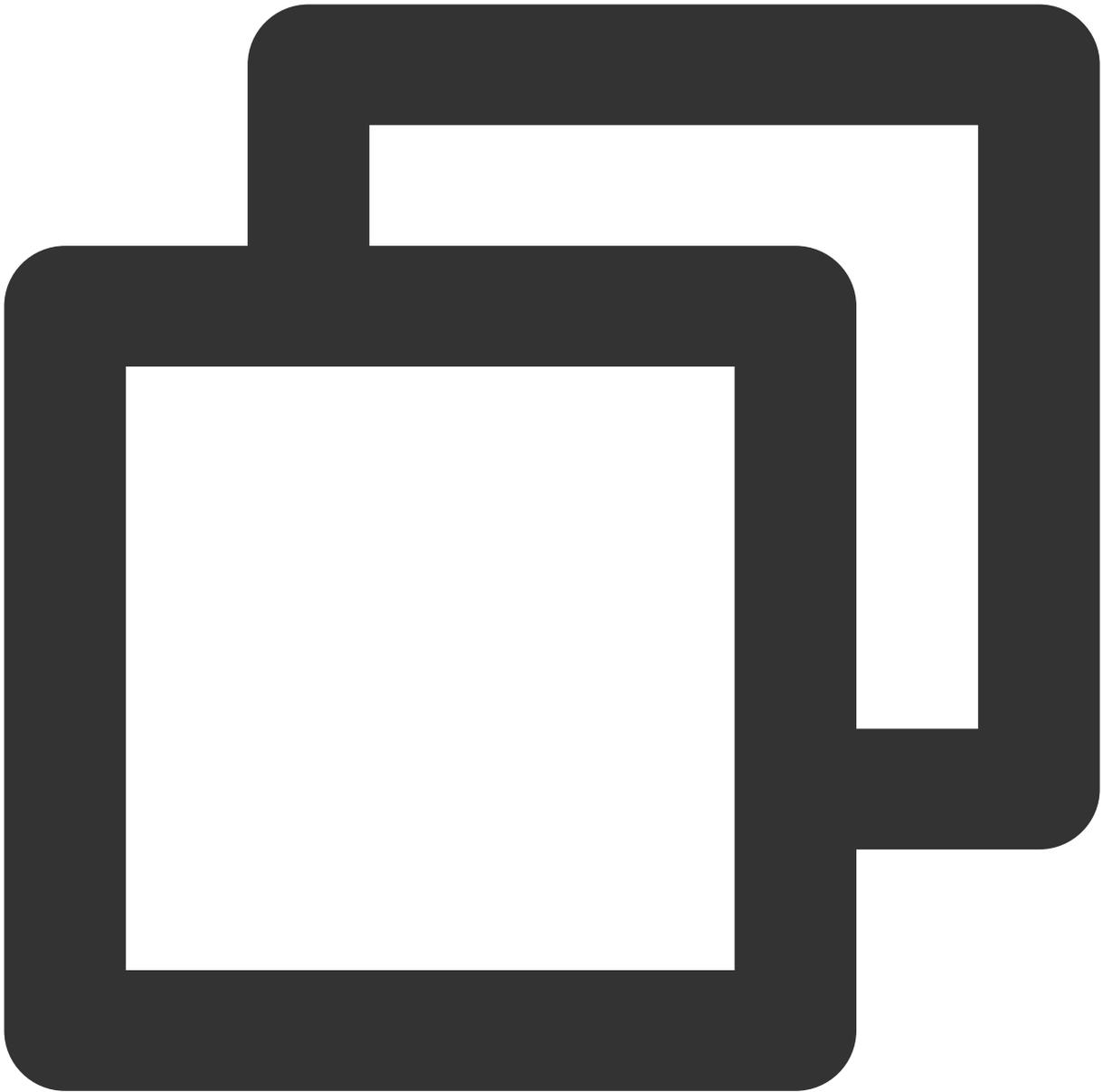
配置 CentOS

配置 hosts



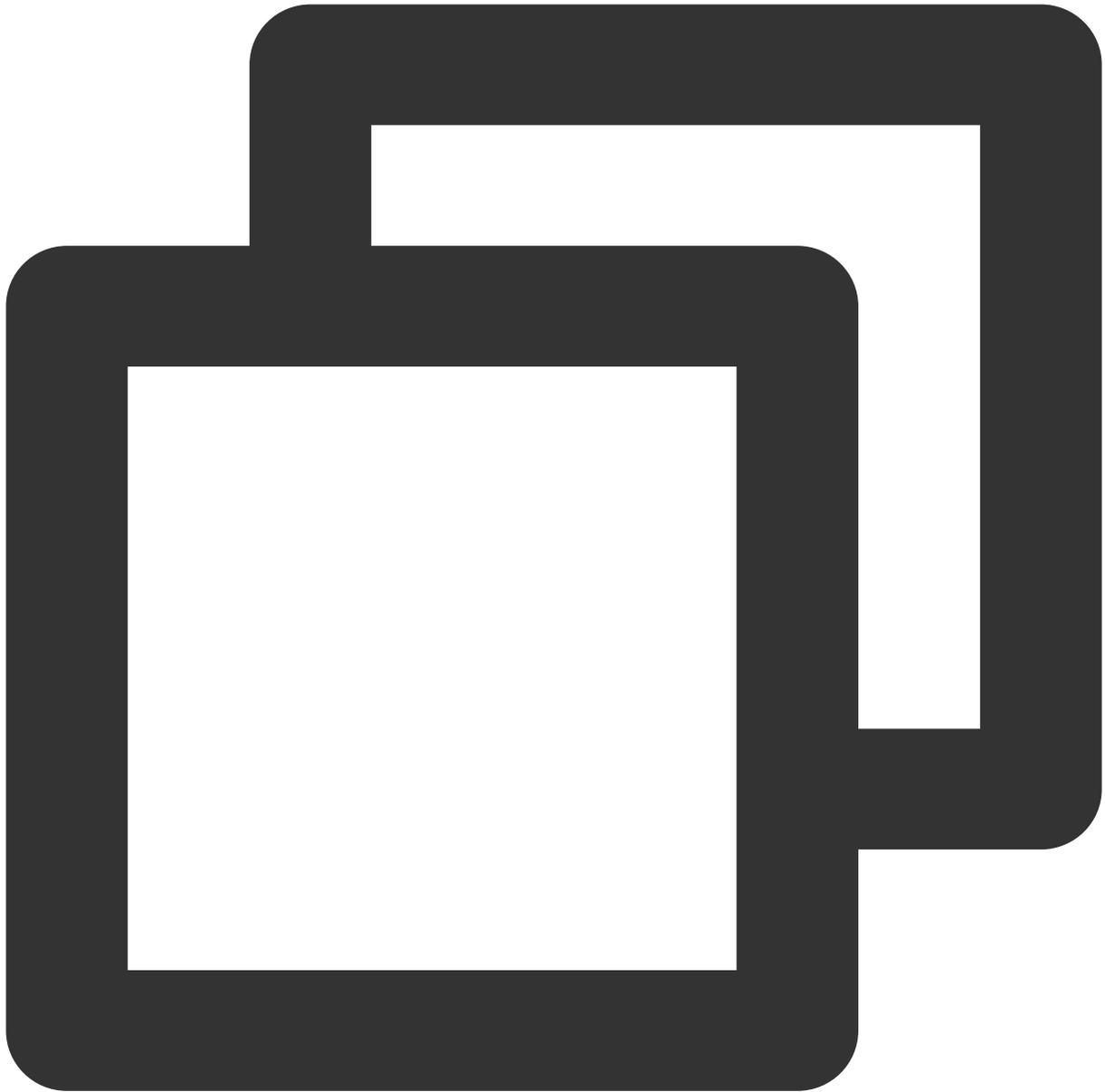
```
vi /etc/hosts
```

编辑内容：



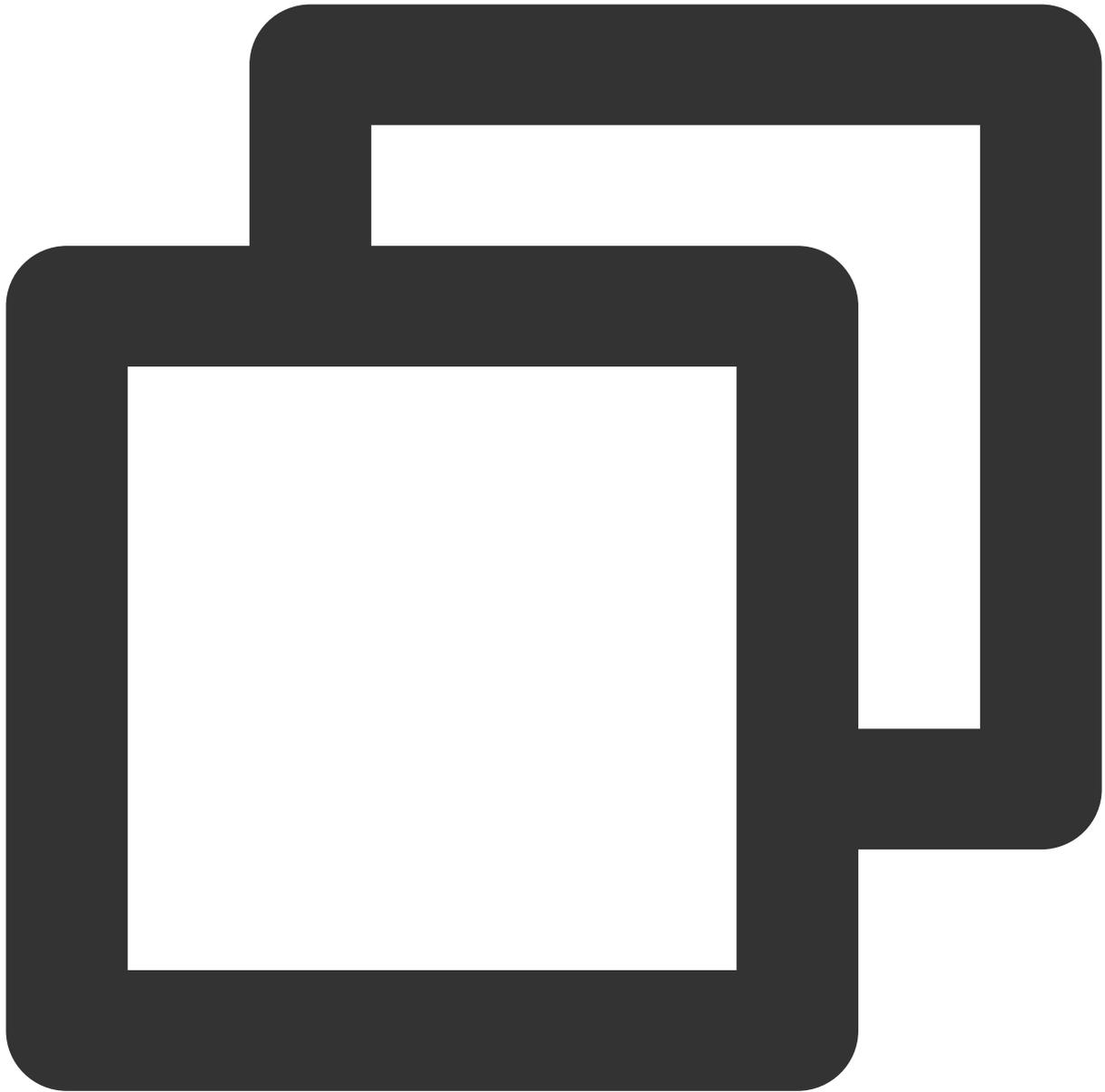
```
202.xxx.xxx.xxx master
202.xxx.xxx.xxx slave1
202.xxx.xxx.xxx slave2
202.xxx.xxx.xxx slave3
//IP 地址替换为真实 IP
```

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
```

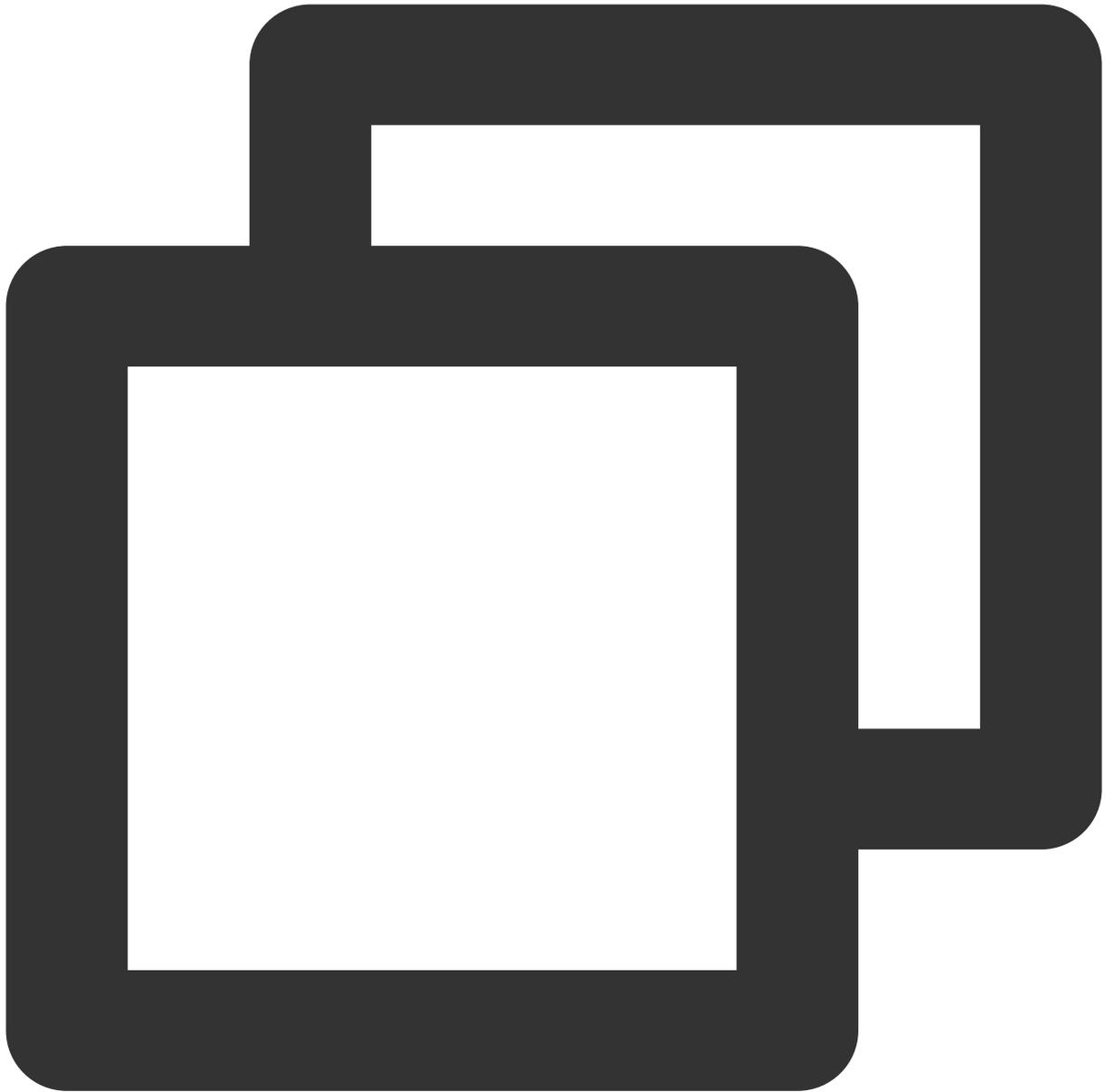
时间同步



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

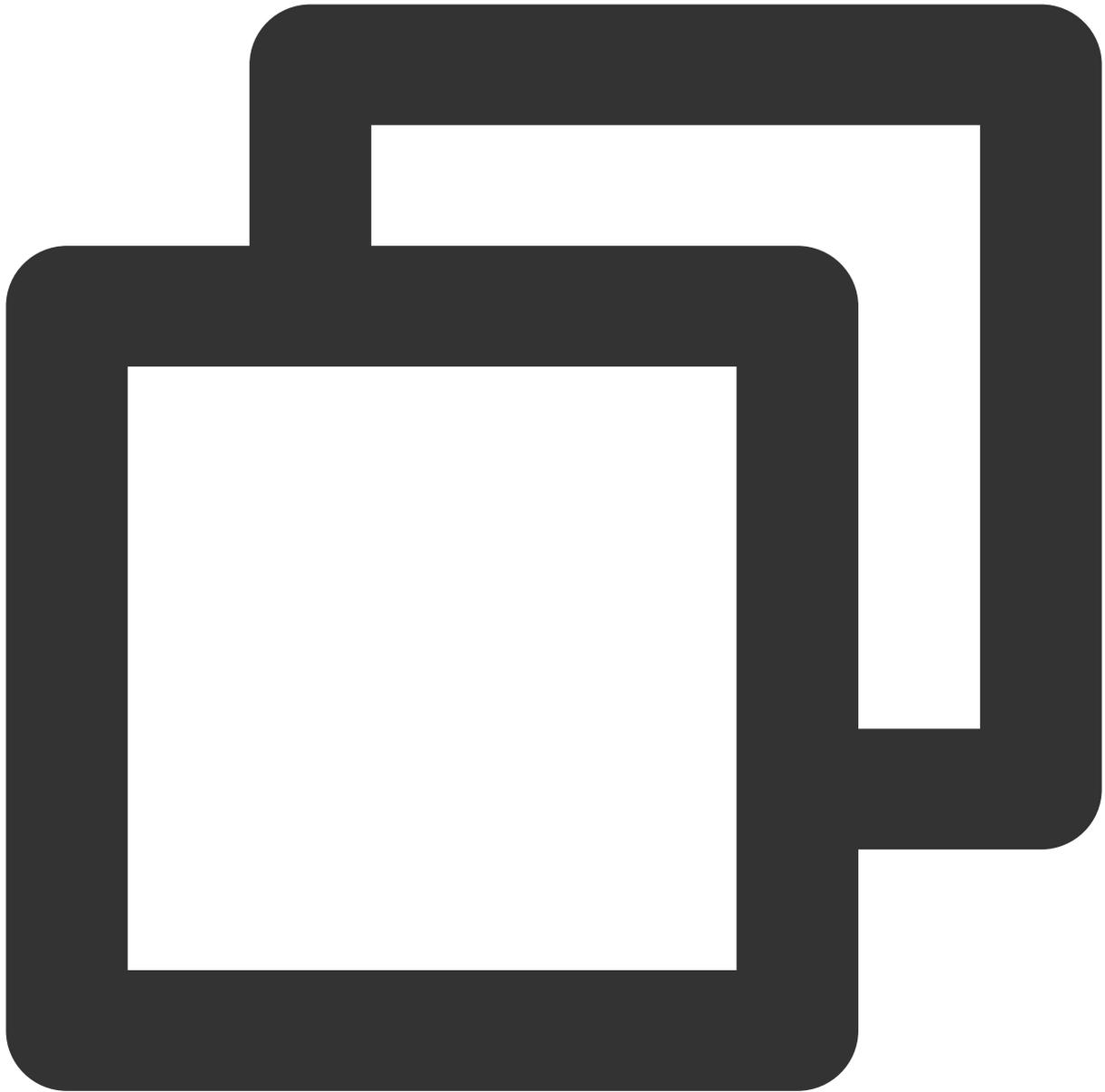
安装配置 JDK

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



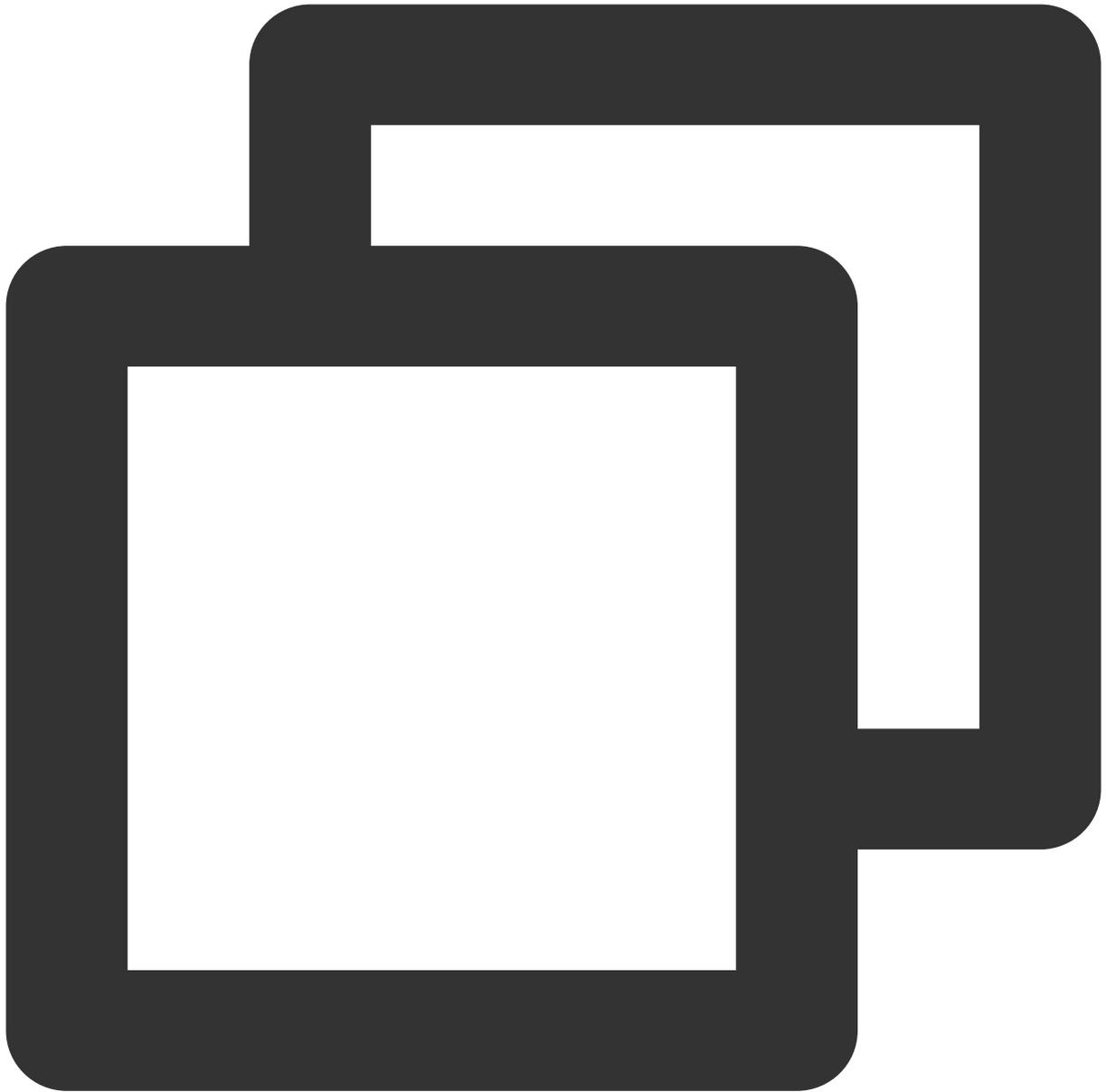
```
mkdir /usr/java  
tar -zxvf jdk-8u144-linux-x64.tar.gz -C /usr/java/  
rm -rf jdk-8u144-linux-x64.tar.gz
```

各个主机之间复制 JDK



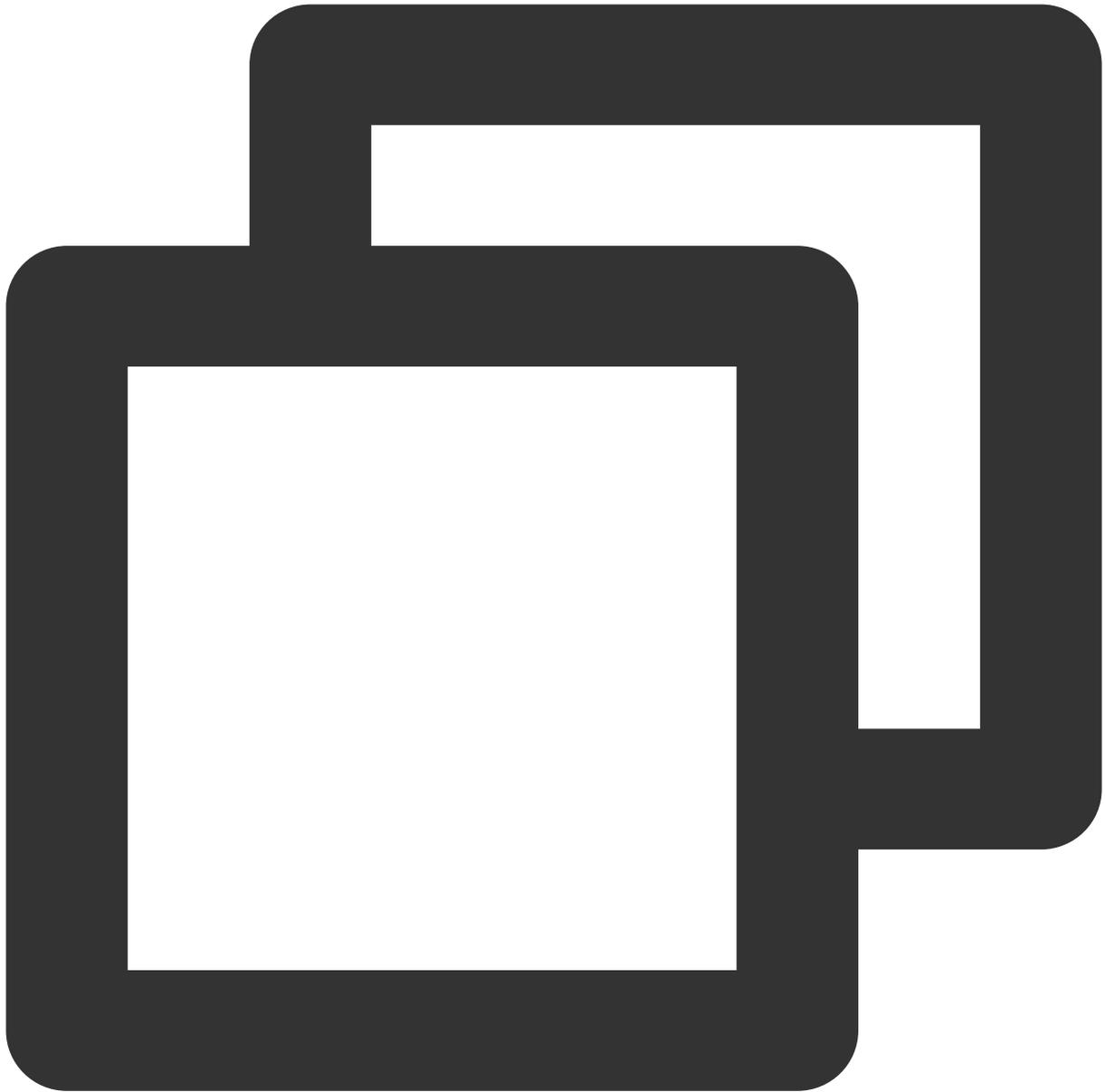
```
scp -r /usr/java slave1:/usr  
scp -r /usr/java slave2:/usr  
scp -r /usr/java slave3:/usr  
.....
```

配置各个主机 JDK 环境变量



```
vi /etc/profile
```

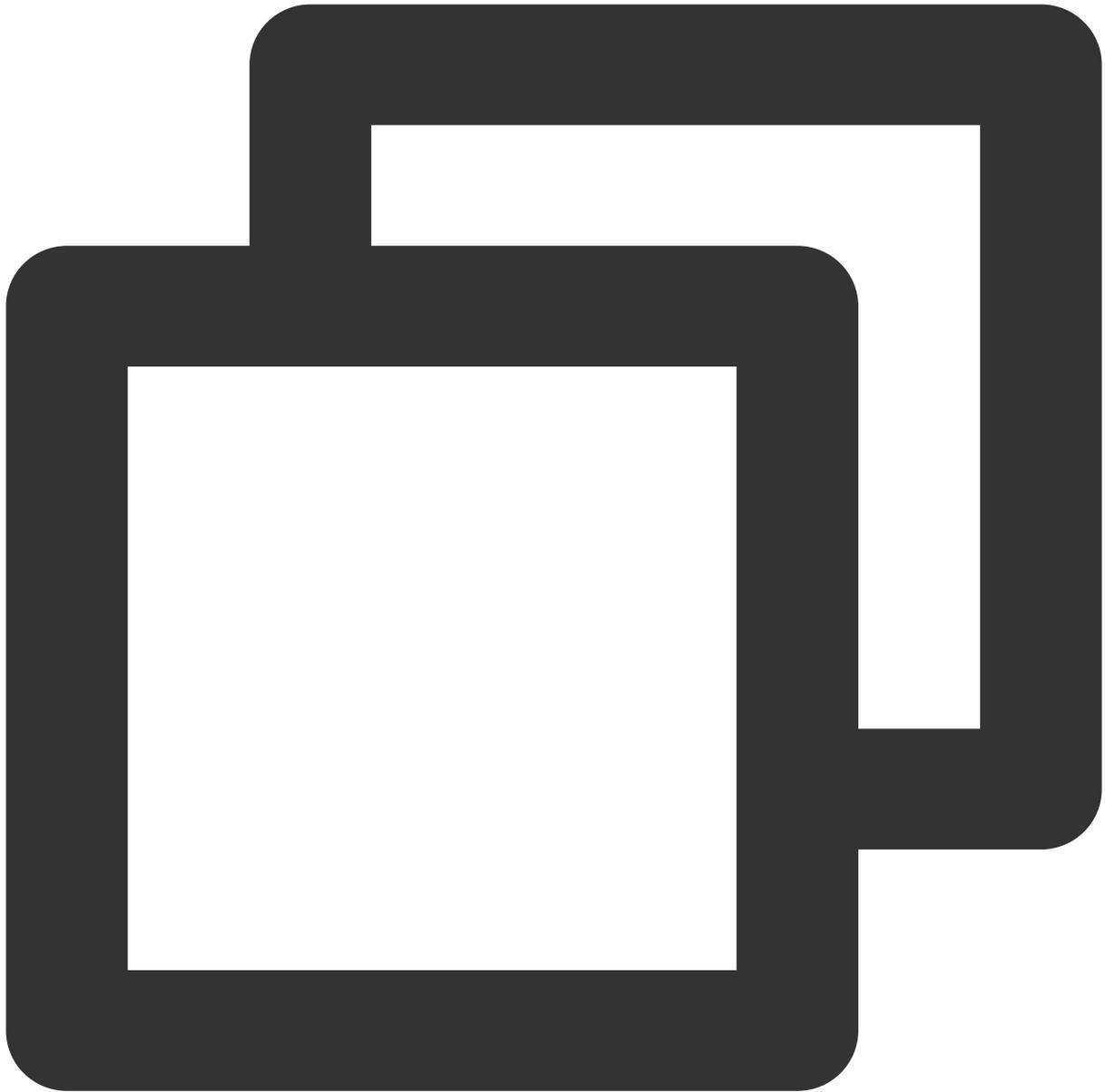
编辑内容：



```
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
```

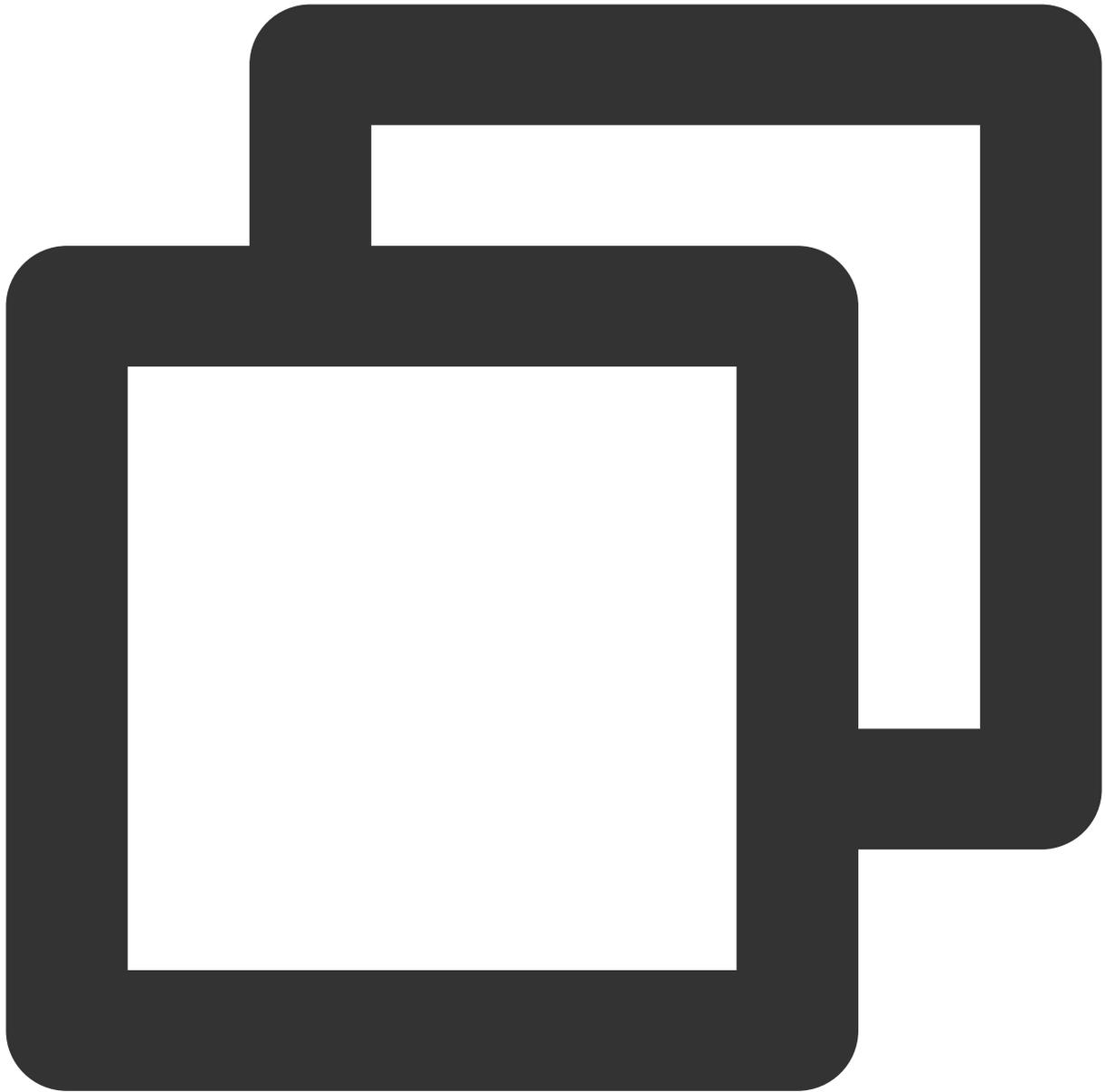
配置 SSH 无密钥访问

分别在各个主机上检查 SSH 服务状态：



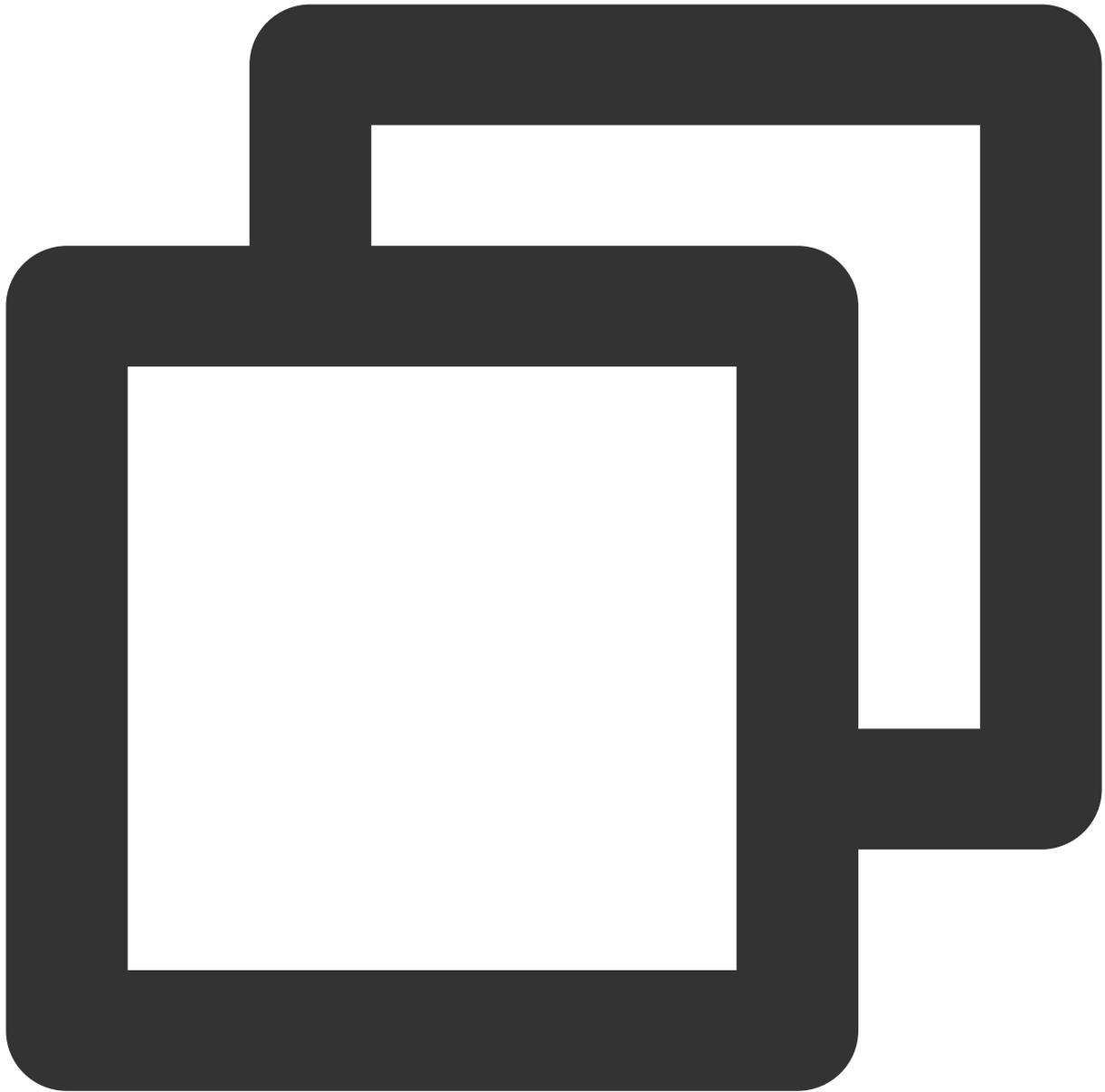
```
systemctl status sshd.service //检查 SSH 服务状态
yum install openssh-server openssh-clients //安装 SSH 服务, 如果已安装, 则不用执行该步骤
systemctl start sshd.service //启动 SSH 服务, 如果已安装, 则不用执行该步骤
```

分别在各个主机上生成密钥：



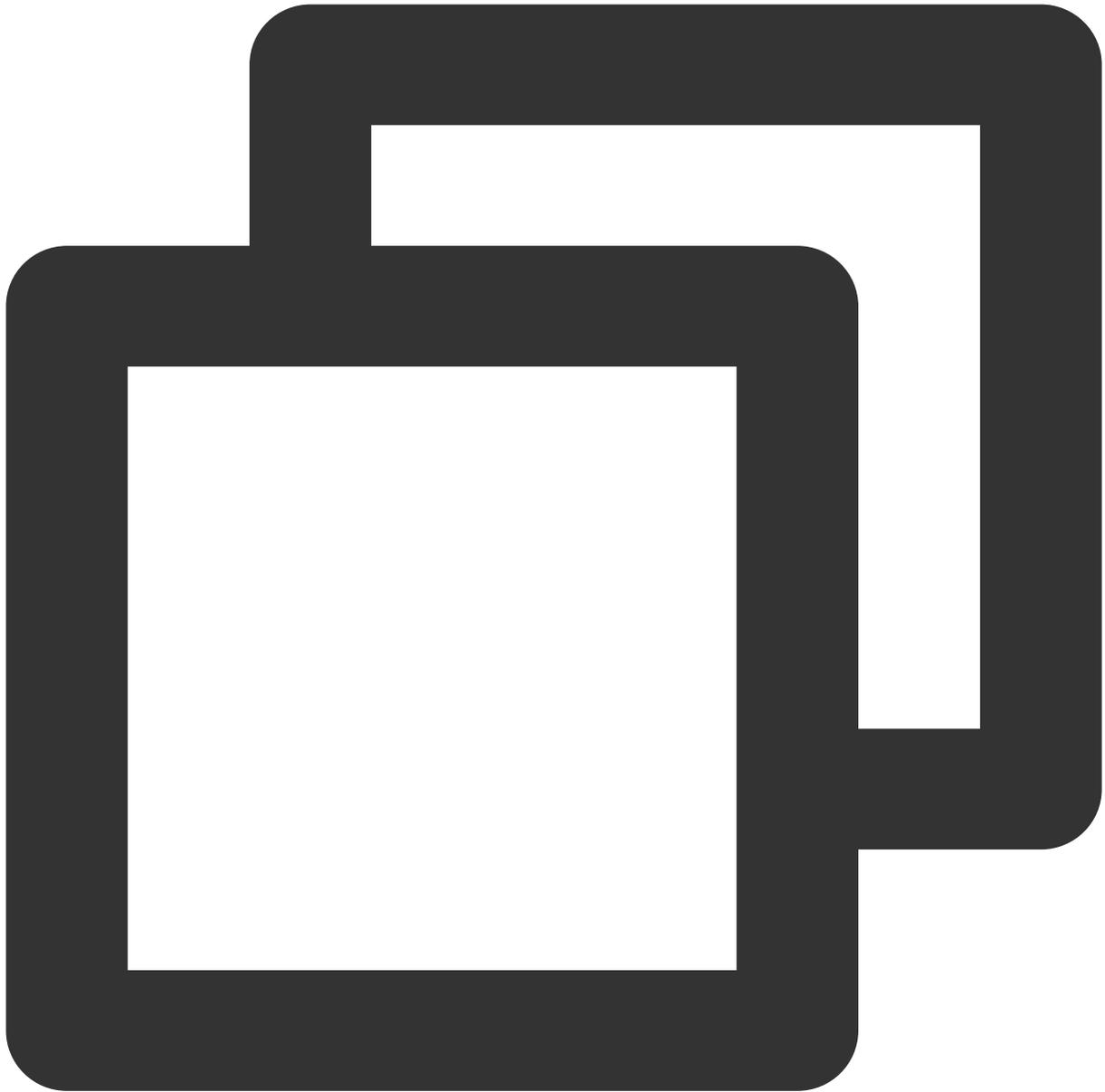
```
ssh-keygen -t rsa //生成密钥
```

在 slave1 上：



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

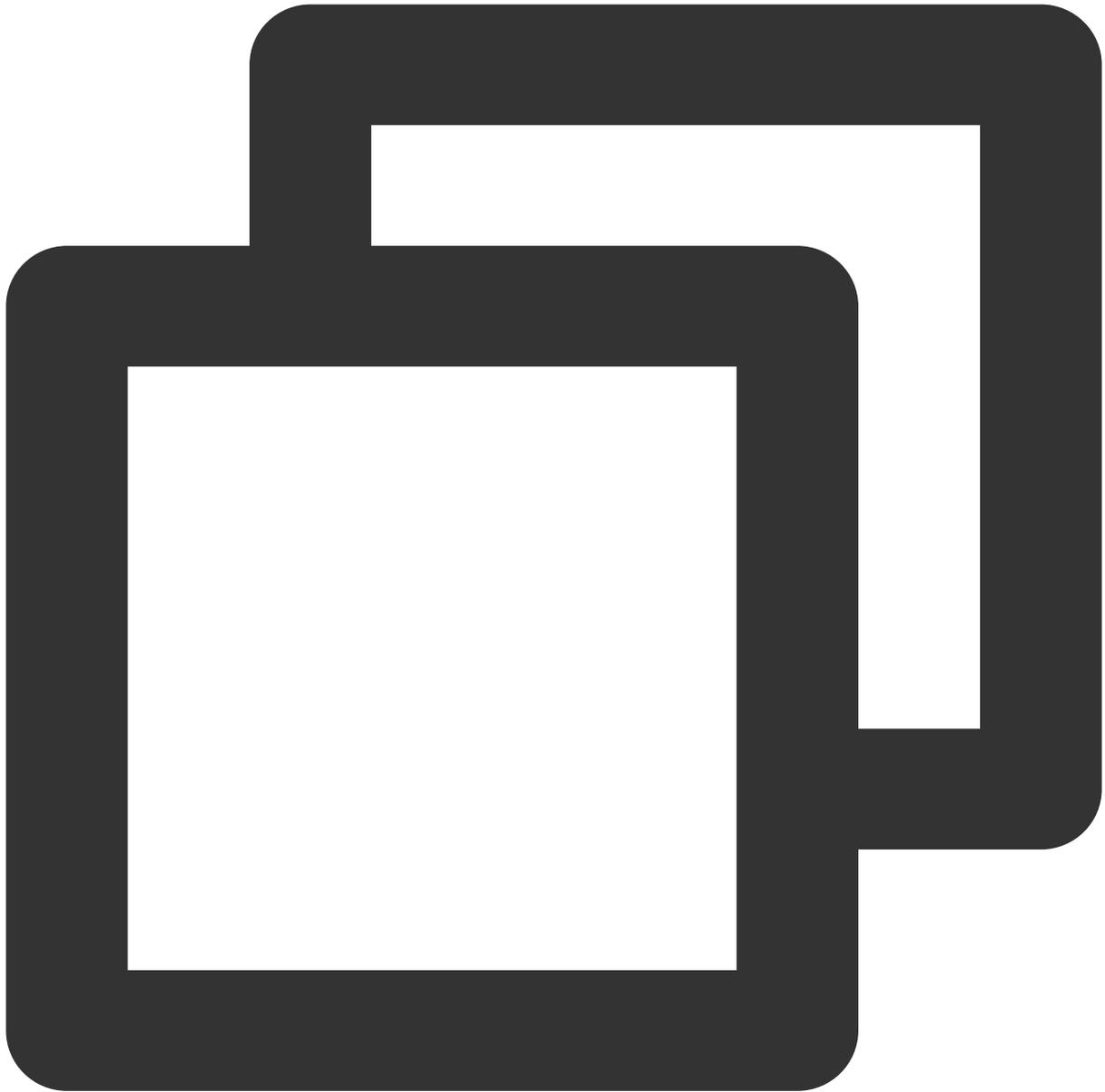
在 slave2 上：



```
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：

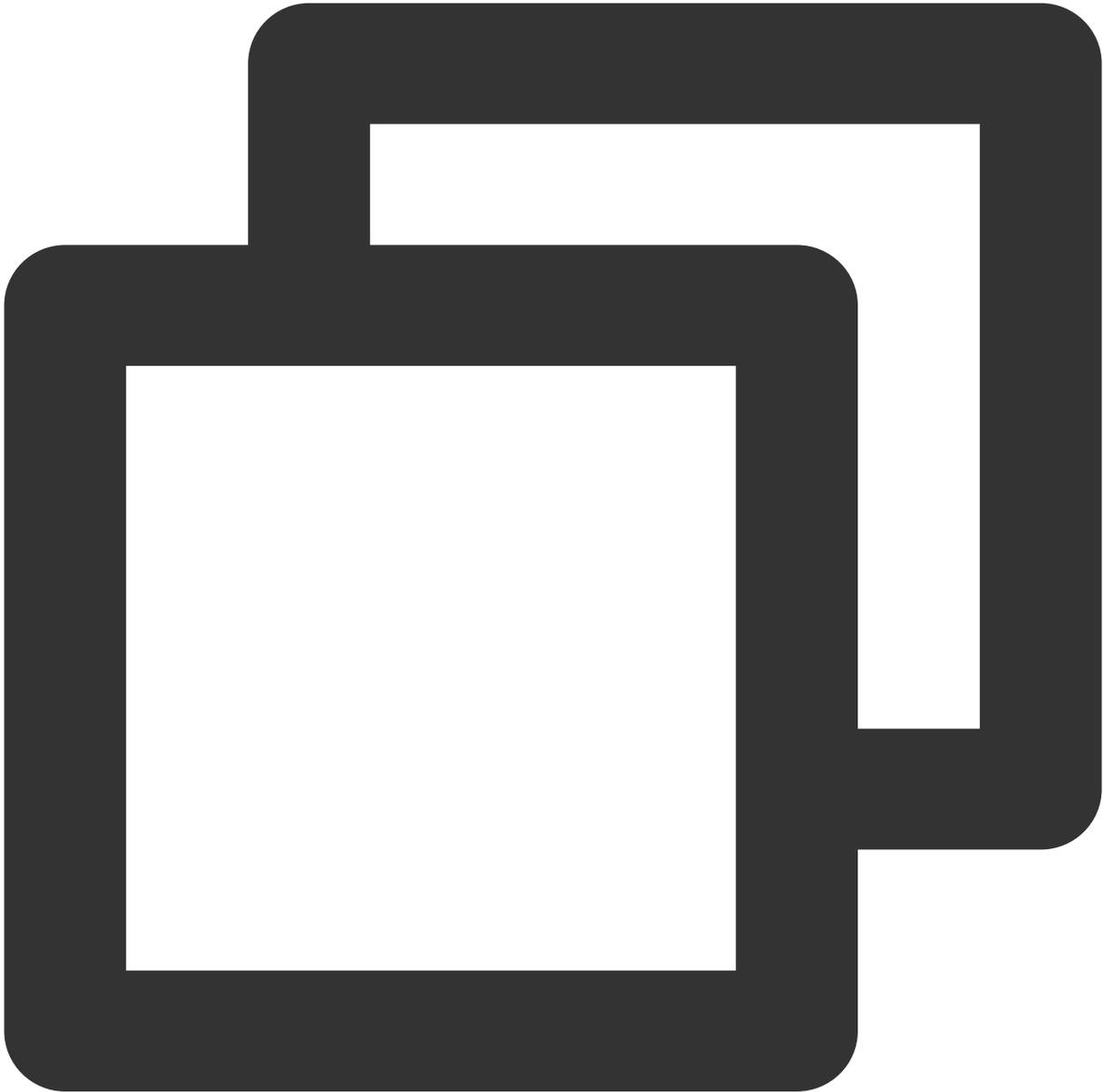


```
cd ~/.ssh
cat id_rsa.pub >> authorized_keys
cat slave1.id_rsa.pub >>authorized_keys
cat slave2.id_rsa.pub >>authorized_keys
scp authorized_keys slave1:~/.ssh
scp authorized_keys slave2:~/.ssh
scp authorized_keys slave3:~/.ssh
```

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。



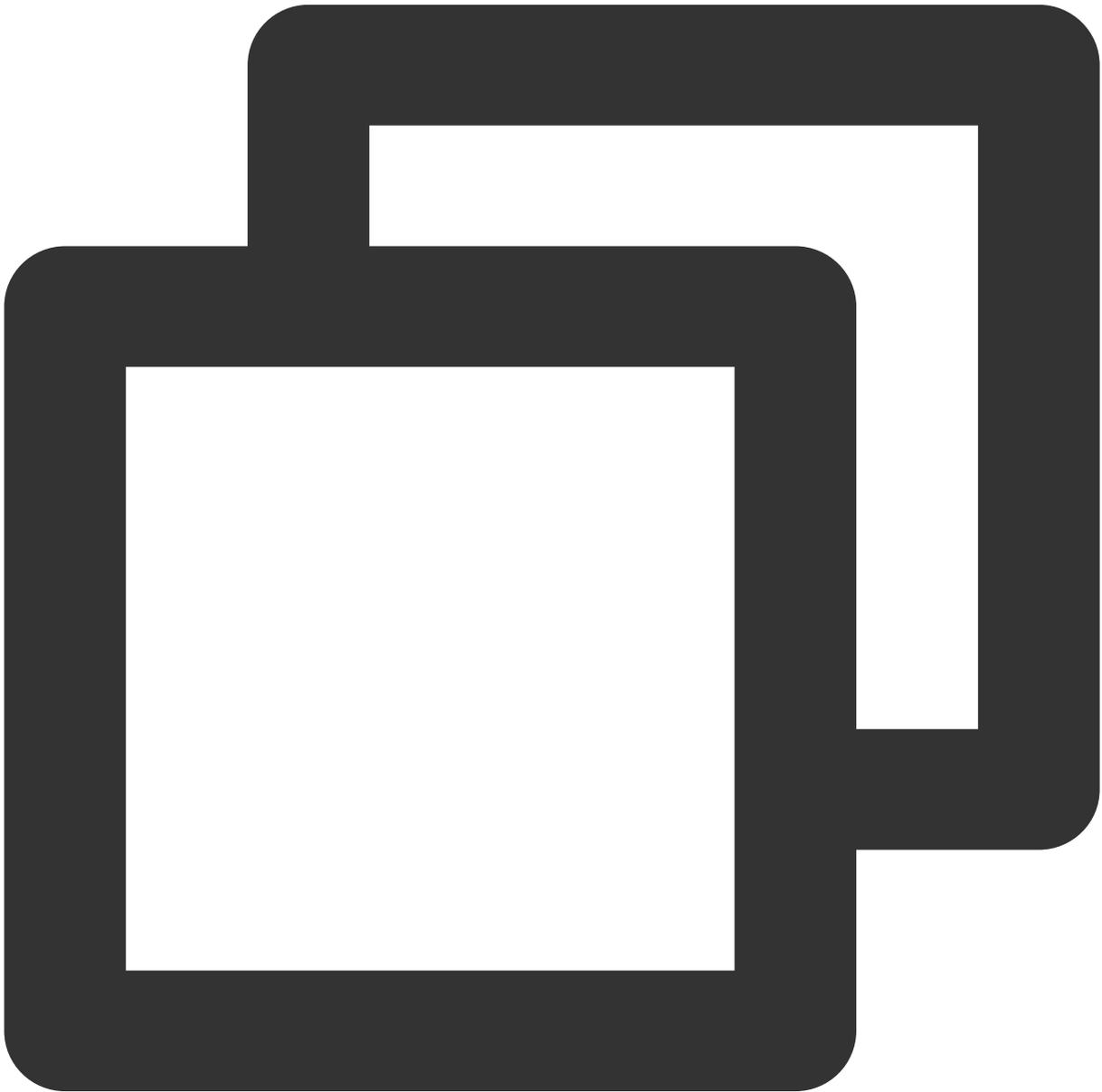
```
tar -zxvf hadoop-2.7.4.tar.gz -C /usr
rm -rf hadoop-2.7.4.tar.gz
mkdir /usr/hadoop-2.7.4/tmp
mkdir /usr/hadoop-2.7.4/logs
mkdir /usr/hadoop-2.7.4/hdf
```

```
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。

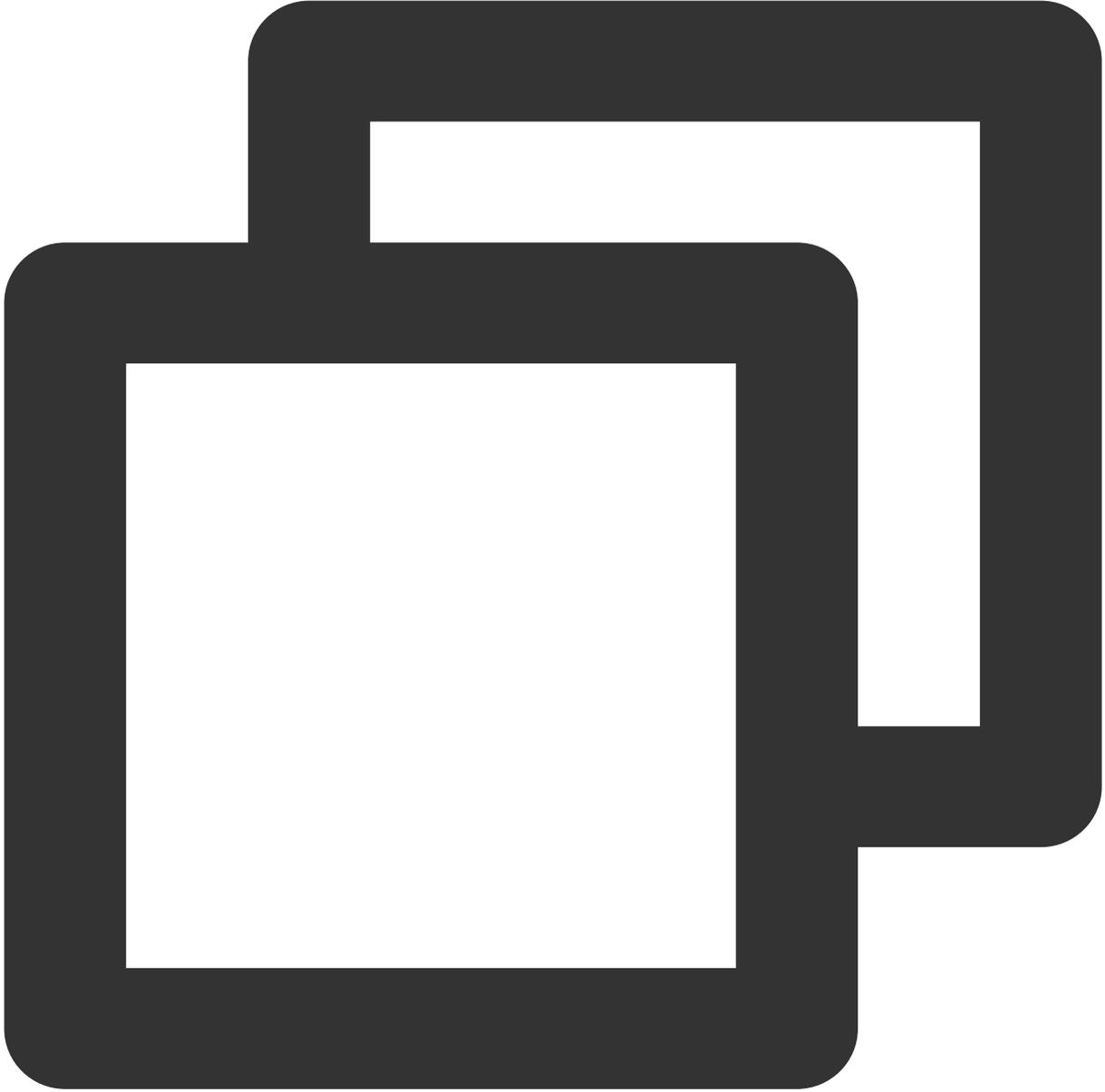
配置 Hadoop

1. 修改 `hadoop-env.sh` 文件，增加如下内容：



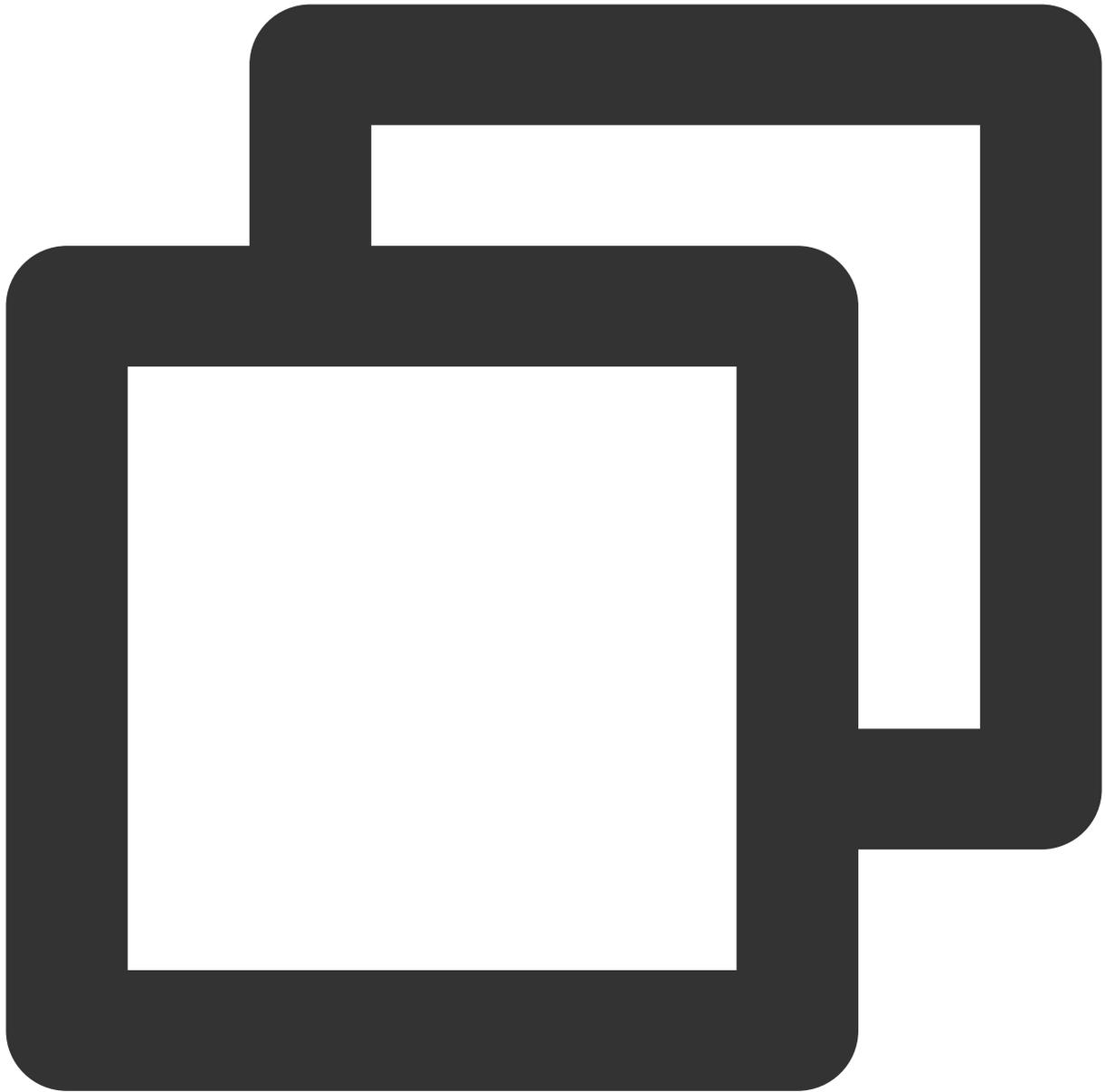
```
export JAVA_HOME=/usr/java/jdk1.8.0_144
```

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



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

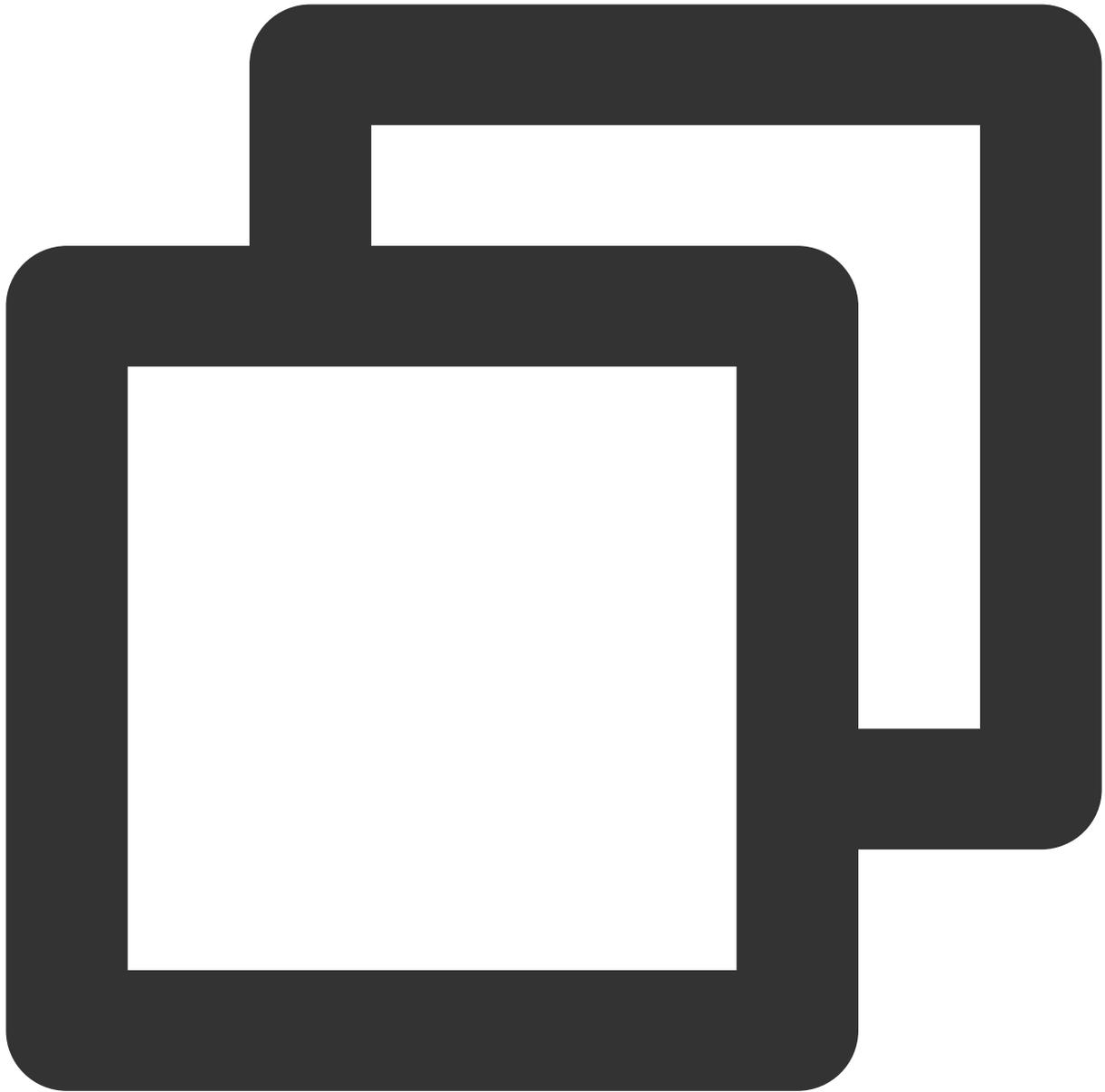
2. 修改 `yarn-env.sh`



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

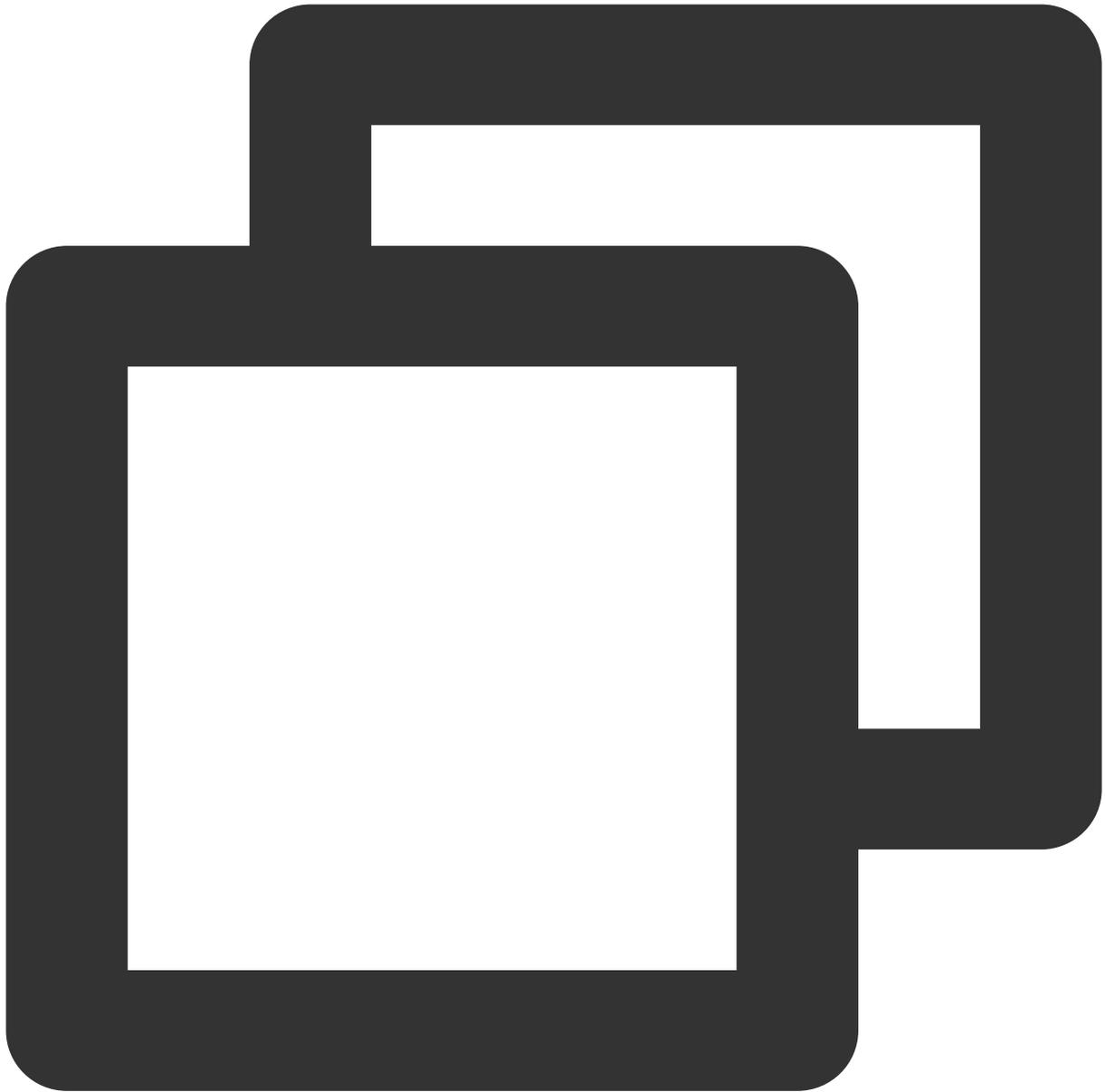
3. 修改 `slaves`

配置内容：



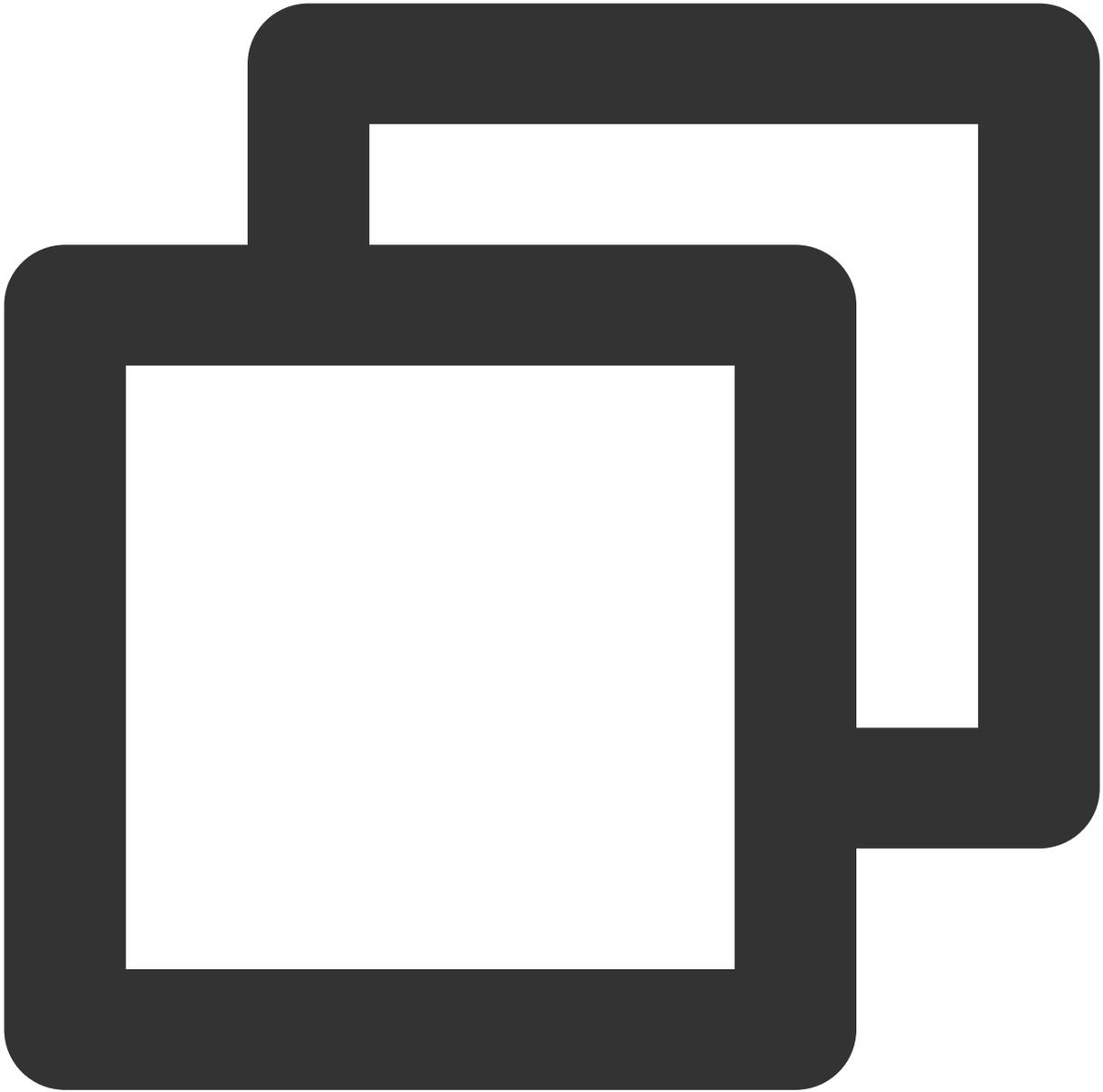
```
删除：  
localhost  
添加：  
slave1  
slave2  
slave3
```

4. 修改 `core-site.xml`



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

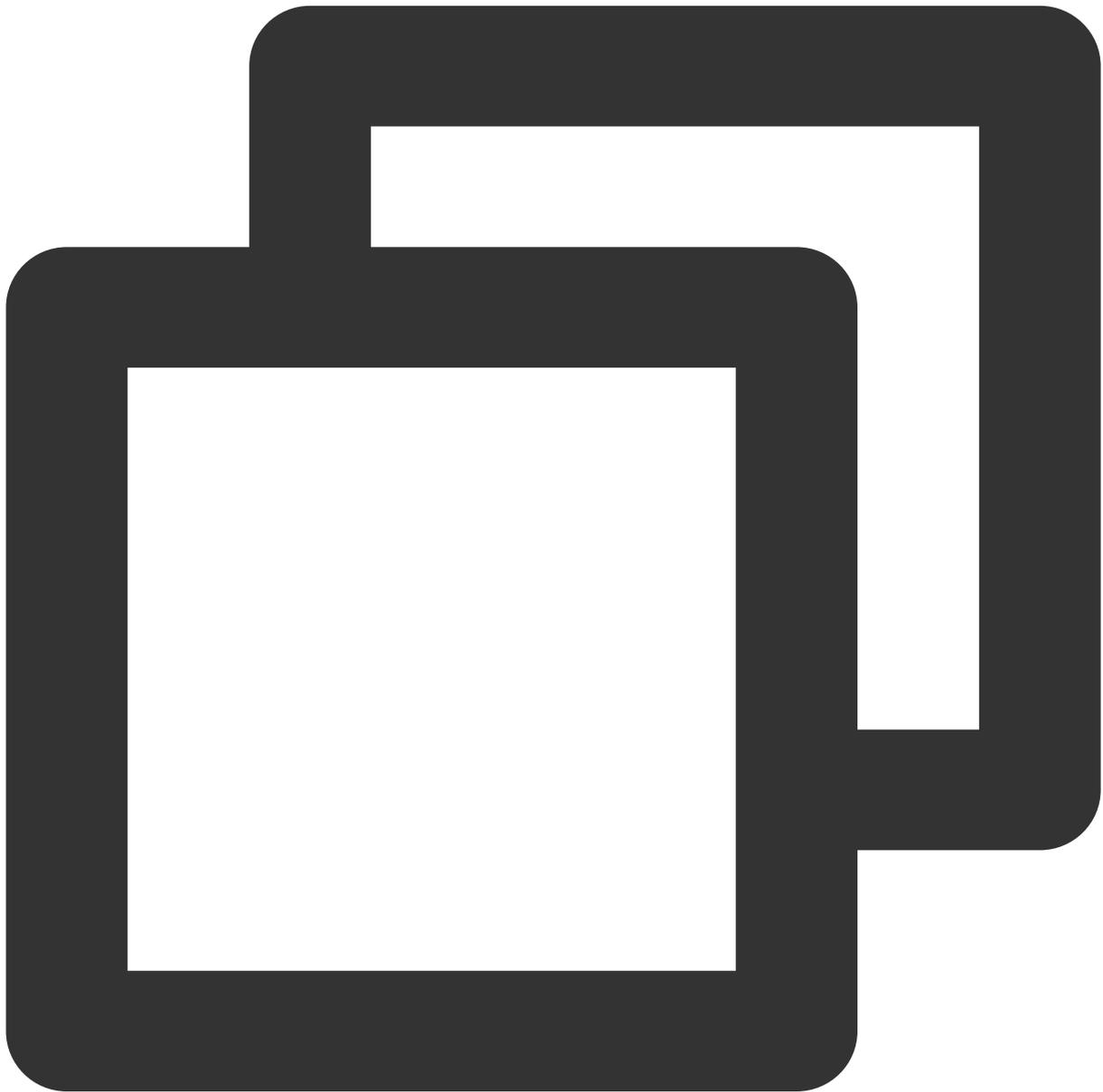
5. 修改 `hdfs-site.xml`



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

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

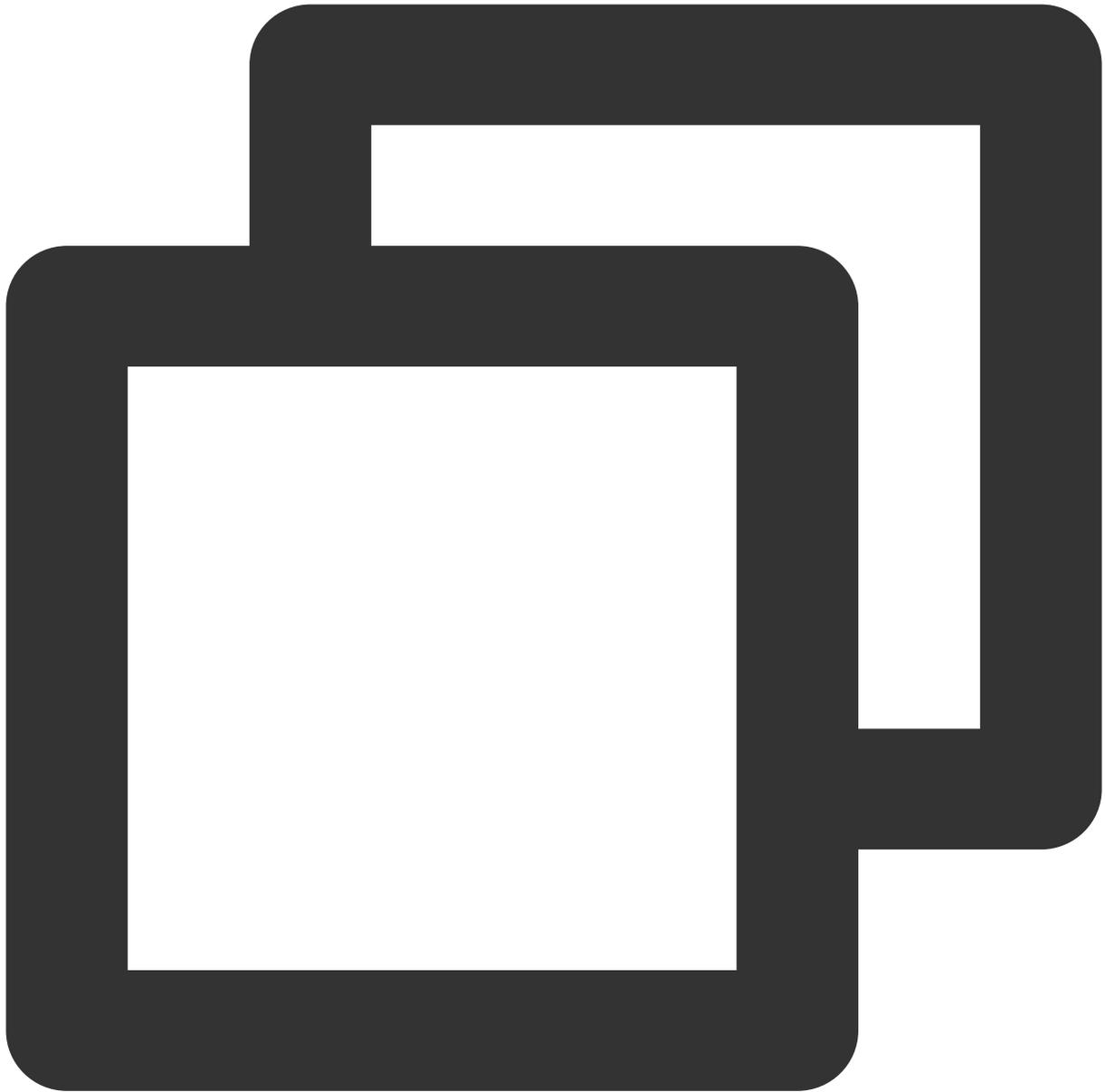
6. 修改 `mapred-site.xml`



```
<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>
```

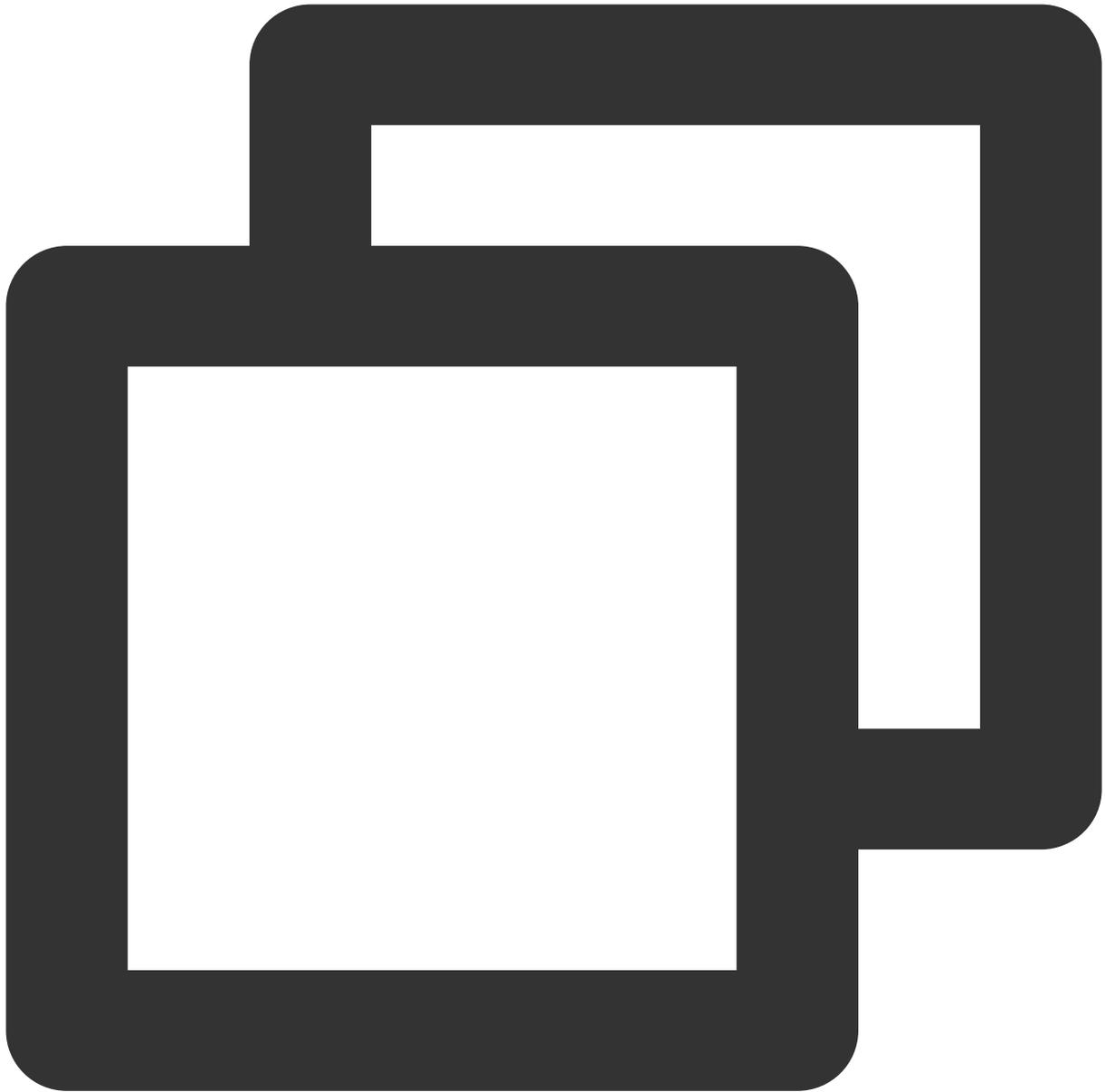
7. 修改 `yarn-site.xml`



```
<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>
```

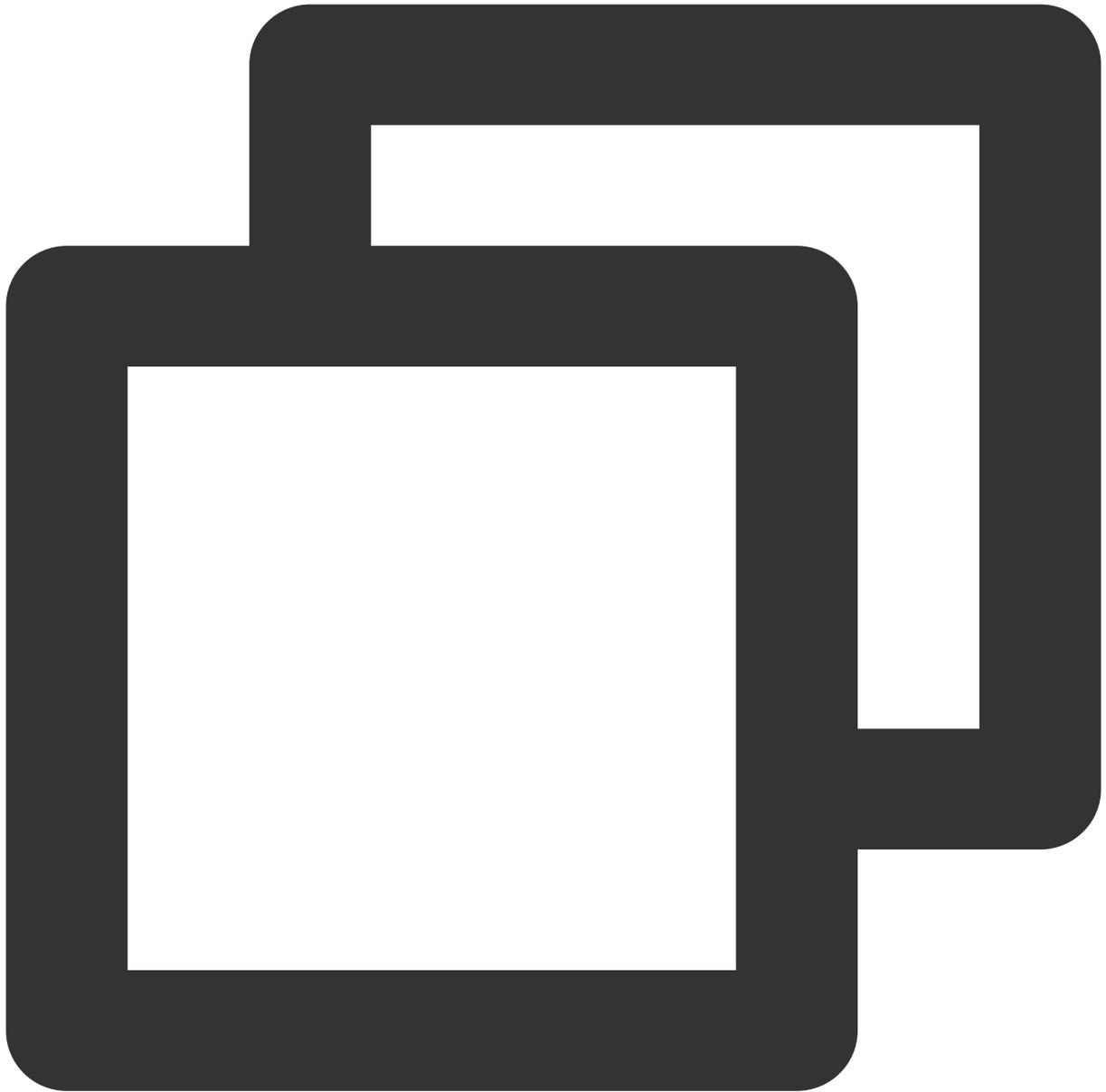
8. 各个主机之间复制 Hadoop



```
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
```

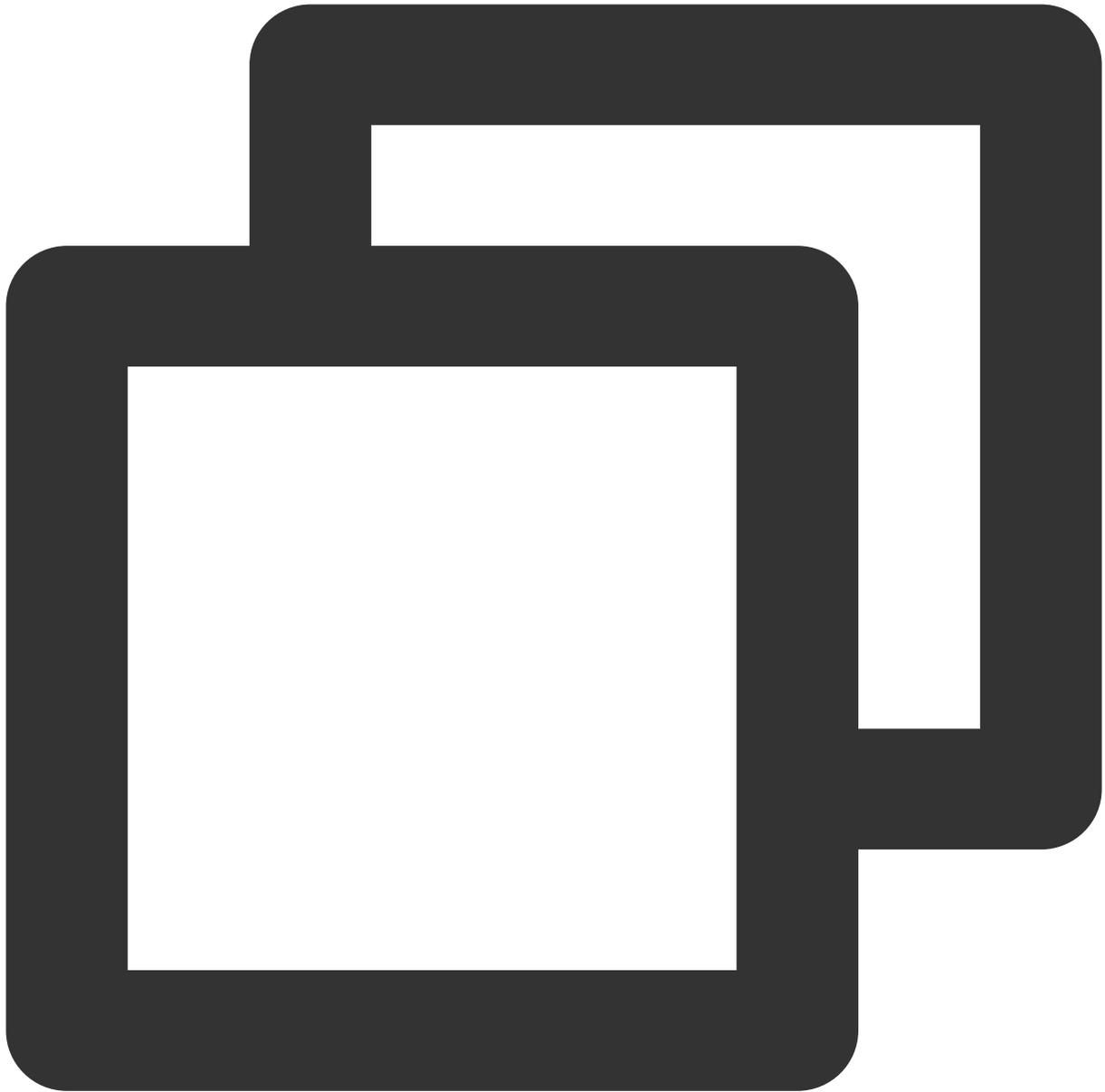
9. 各个主机配置 Hadoop 环境变量

打开配置文件：



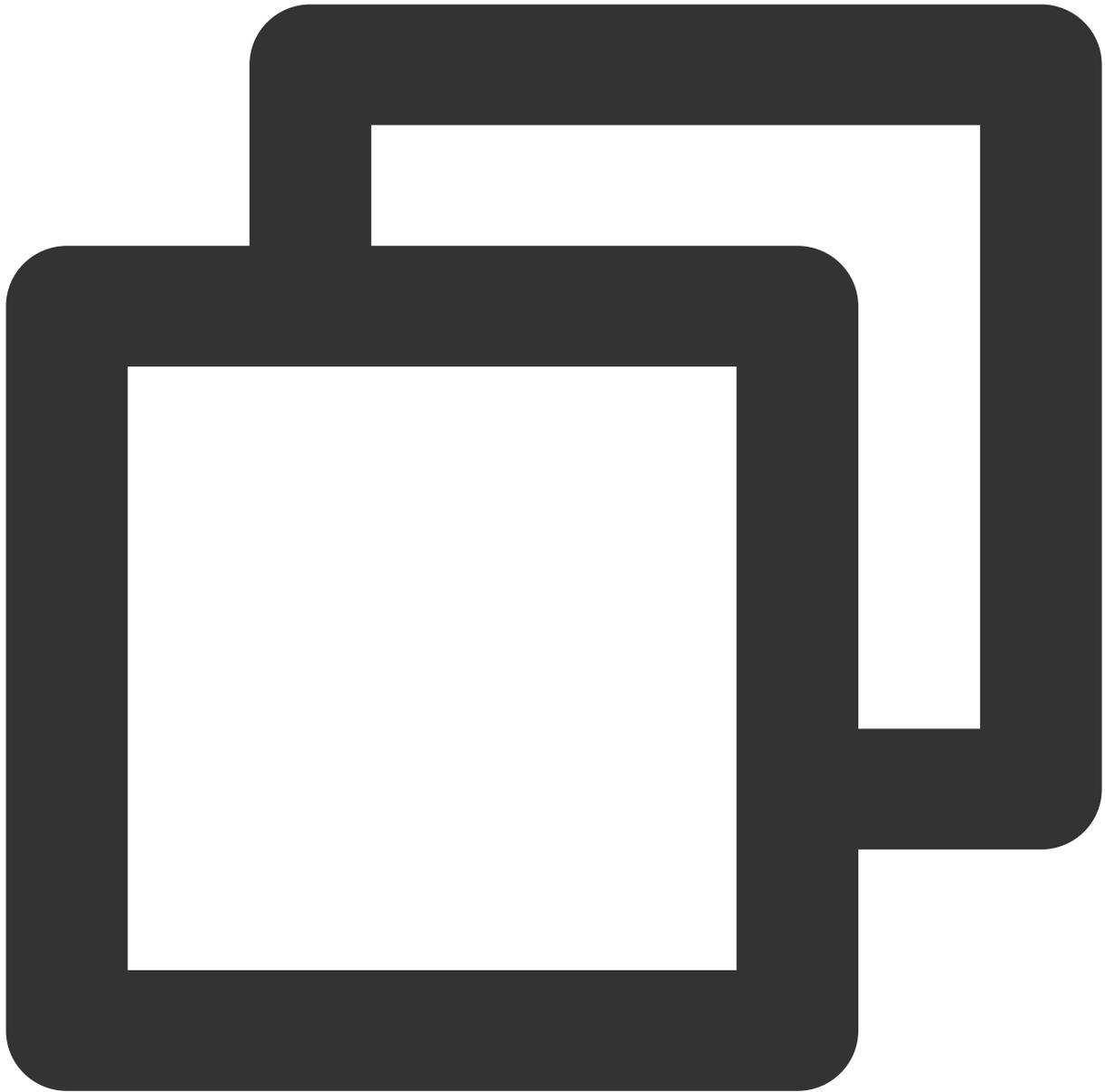
```
vi /etc/profile
```

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
```

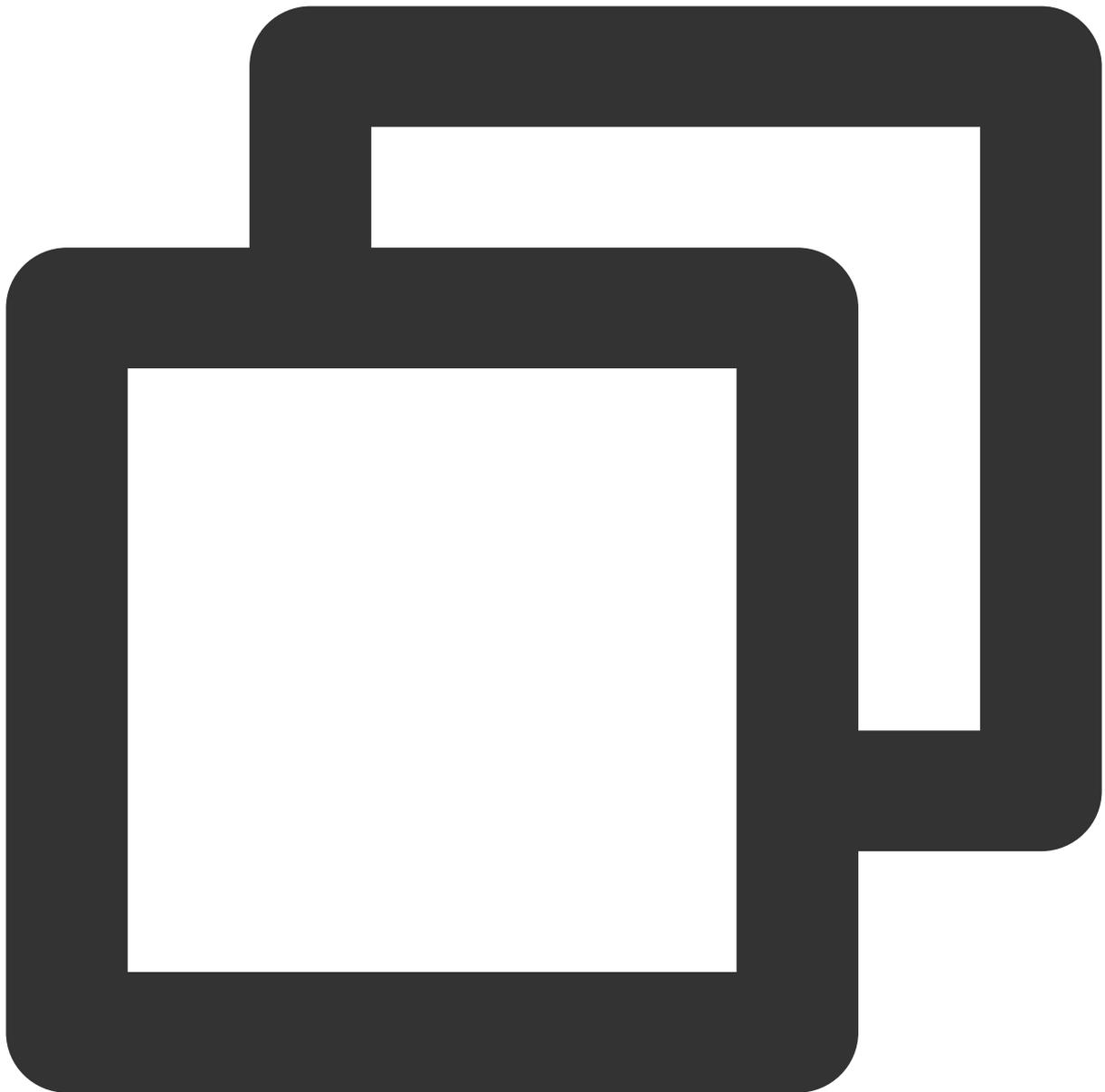
使配置文件生效：



```
source /etc/profile
```

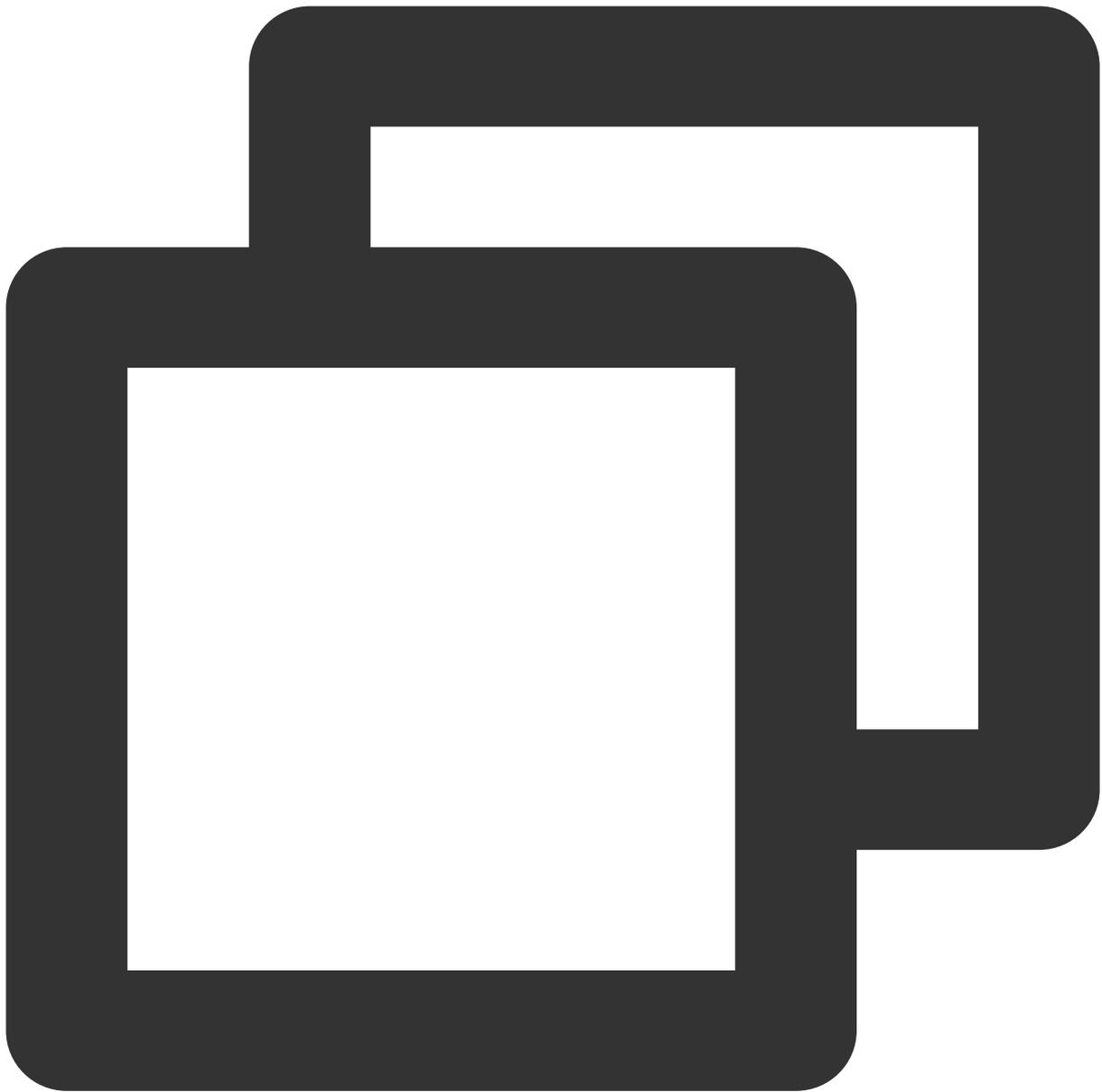
启动 Hadoop

1. 格式化 namenode



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

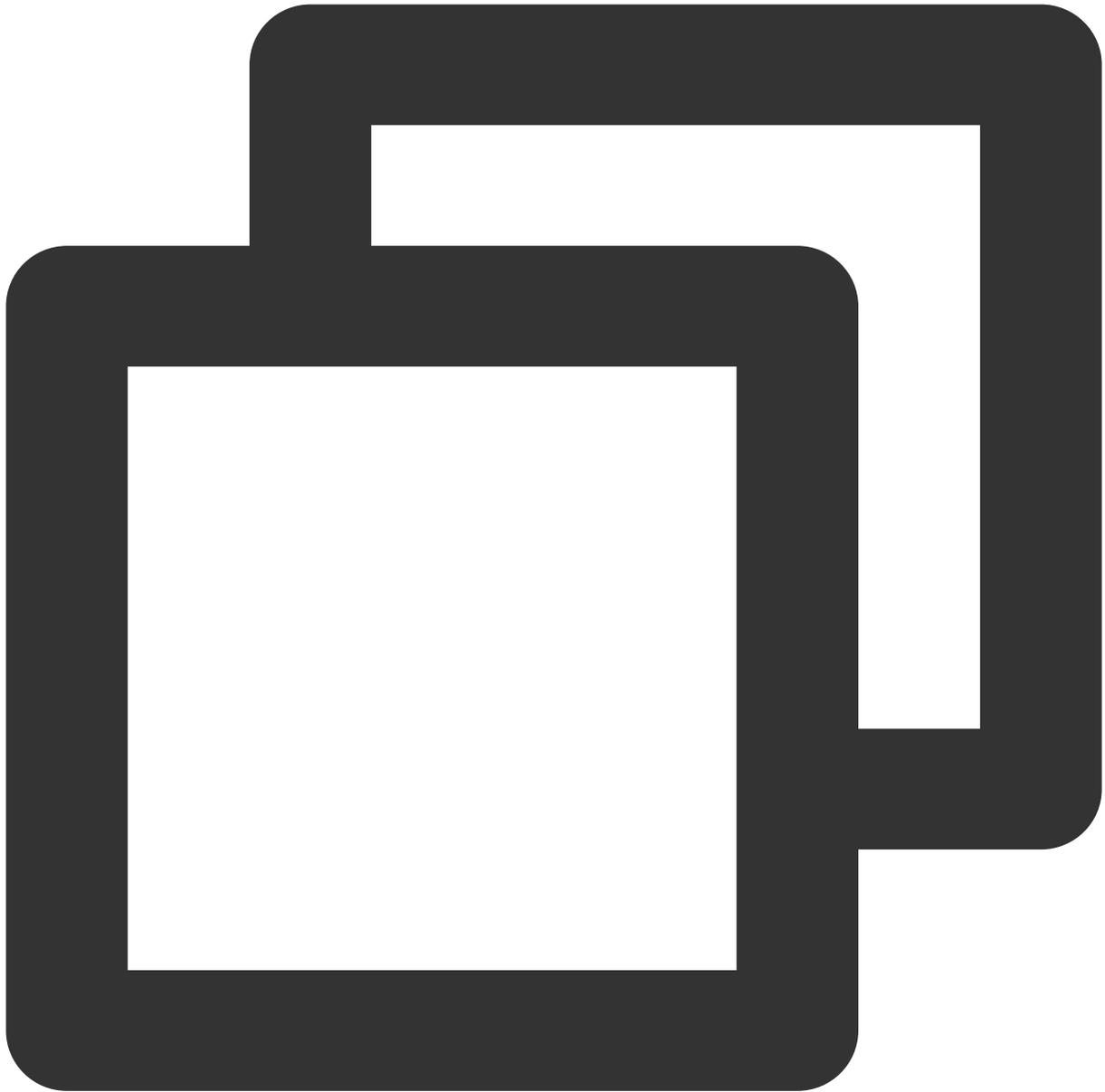
2. 启动



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

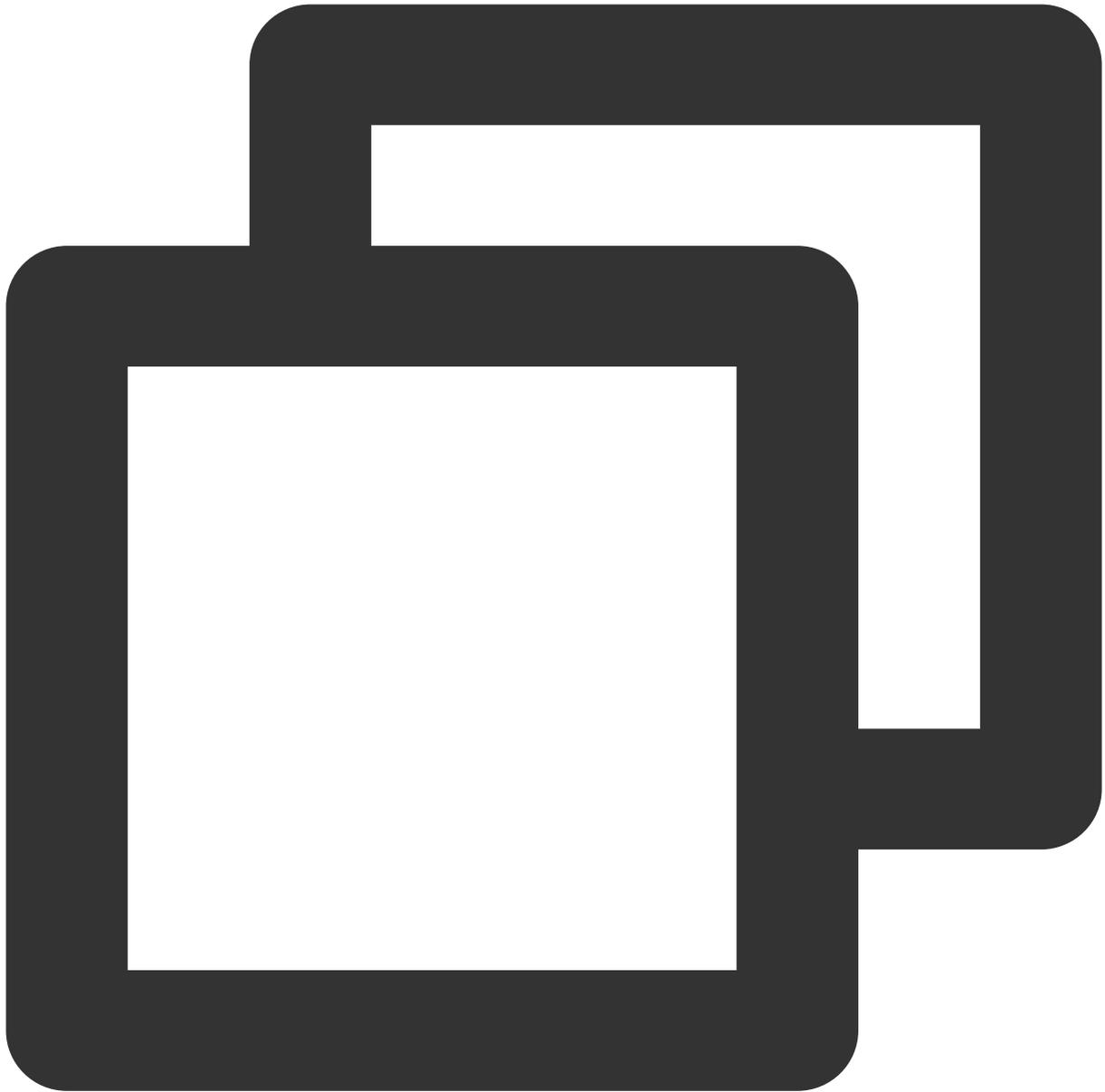
3. 检查进程

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
```

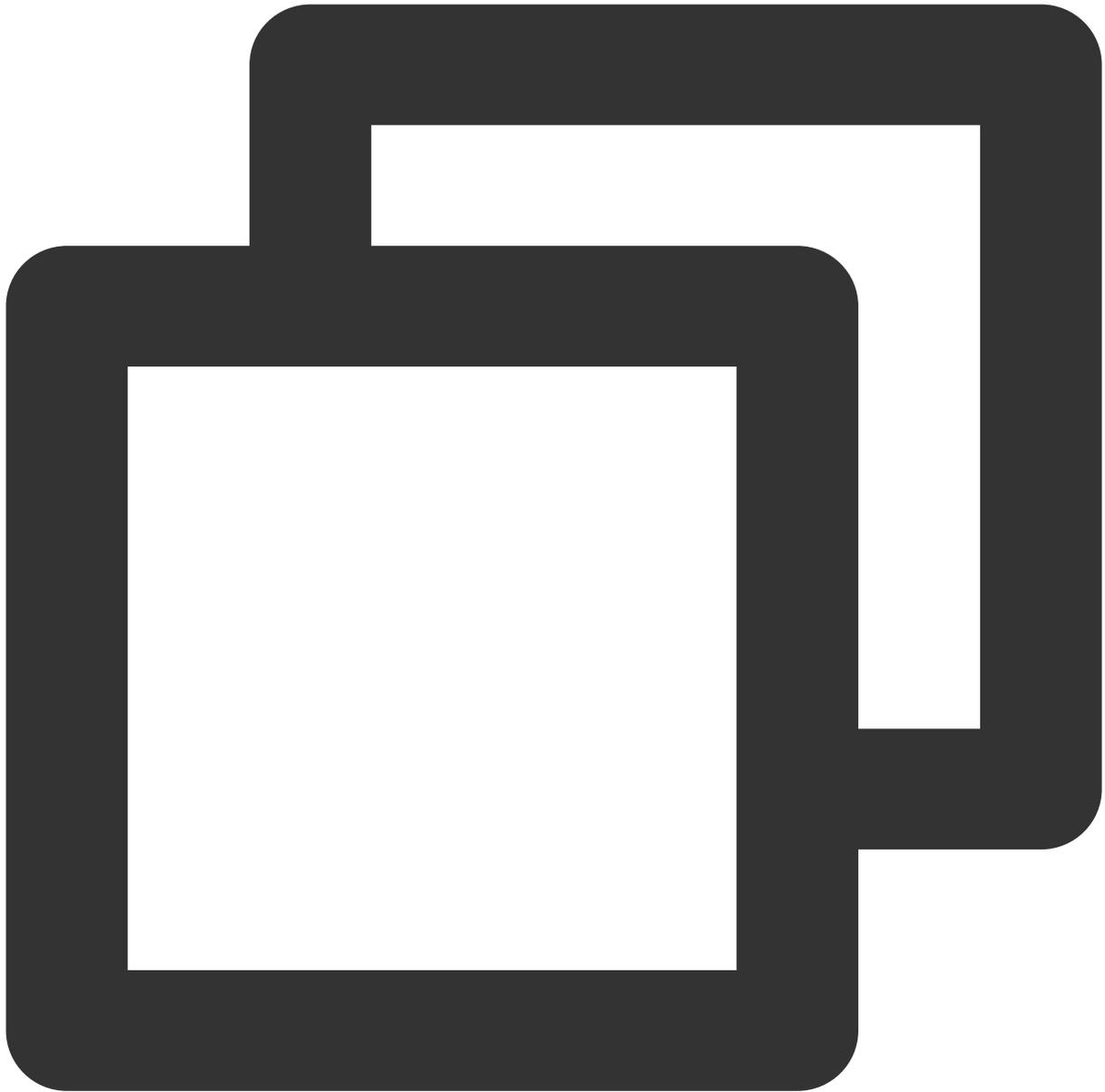
各个 slave 主机包含 DataNode、NodeManager 等，则表示启用成功，例如：



```
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：

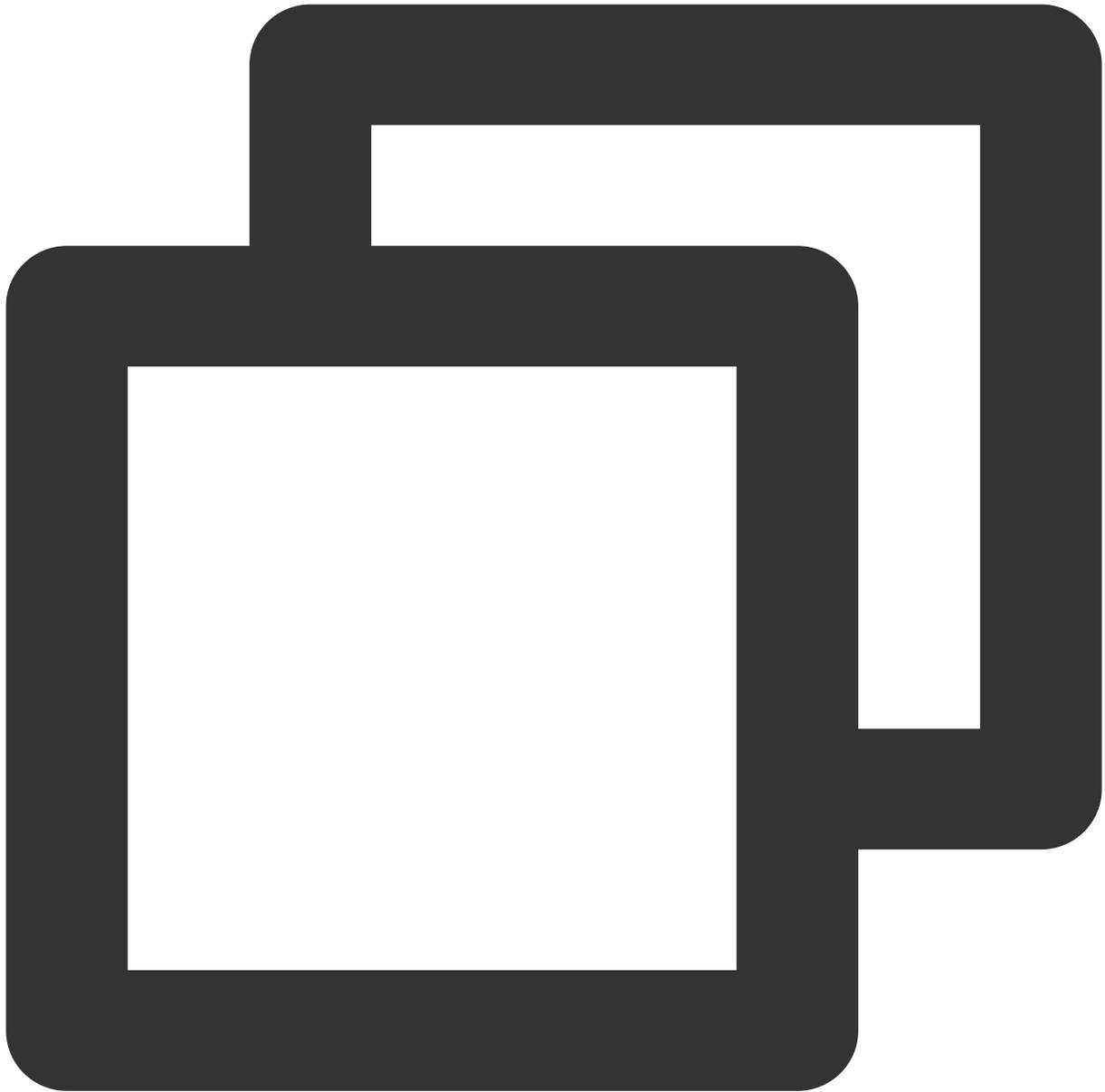


```
hadoop fs -mkdir input
hadoop fs -put input.txt /input
hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar wordcount /in
```

```
[root@VM_96_24_centos /usr/hadoop-2.7.4]# hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examp
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/applicati
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
```

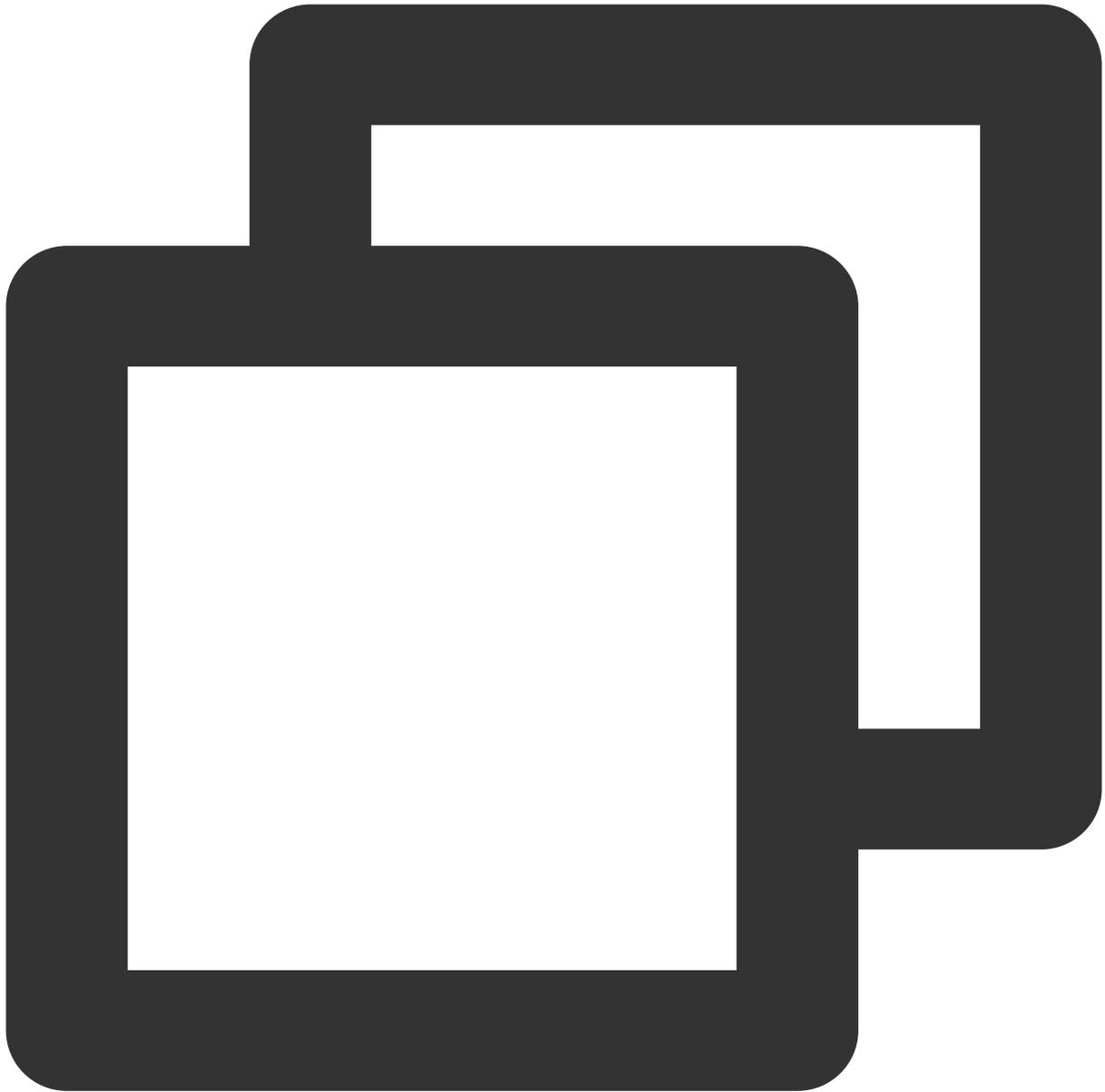
出现如上图结果就说明 Hadoop 安装已经成功了。

[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
```

说明

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

COSBrowser 工具

COSBrowser 简介

最近更新时间：2024-04-17 15:02:44

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

[桌面端使用说明](#)

[移动端使用说明](#)

下载地址

COSBrowser 分类	支持平台	系统要求	下载地址
桌面端	Windows	Windows 7 32/64位以上、Windows Server 2008 R2 64位以上	Windows
	macOS	macOS 10.13以上	macOS Intel 芯片版 macOS Apple 芯片版
	Linux	需带有图形界面并支持 ApplImage 格式 注意：CentOS 启动客户端需在终端执行 <code>./cosbrowser.ApplImage --no-sandbox</code>	Linux
移动端	Android	Android 4.4以上	Android
	iOS	iOS 11以上	iOS
网页版	Web	Chrome/FireFox/Safari/IE10+等浏览器	Web

桌面端功能列表

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

注意

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

Windows 用户可在操作系统的“Internet 选项”中查询。

macOS 用户可在“网络偏好设置”中查询。

Linux 用户可在系统设置 > 网络 > 网络代理中查询。

COSBrowser 桌面端支持以下功能：

功能	功能说明
创建/删除存储桶	支持创建、删除存储桶。
查看存储桶详情	支持查看存储桶基本信息。
查看统计数据	支持查看存储桶的当前存储量及对象总数。
权限管理	支持修改存储桶、对象的相关权限。
设置版本控制	支持开启、暂停存储桶版本控制。
添加访问路径	支持添加访问路径。
上传文件/文件夹	支持单个上传、批量上传、增量上传文件或文件夹至存储桶。 注意： 1. 批量上传文件不要超过10万个。 2. 不支持断点续传。 3. 若迁移数据，或批量上传数据，请使用 迁移服务平台 MSP 。
下载文件/文件夹	支持单个下载、批量下载、增量下载文件或文件夹至本地。 注意： 1. 批量下载文件不要超过10万个。 2. 不支持断点续传。
删除文件/文件夹	支持单个删除、批量删除存储桶中的文件或文件夹。
文件同步	支持将本地文件实时同步至存储桶中。
复制粘贴文件	支持单个复制、批量复制一个目录下的文件或文件夹至另一个目录。
文件重命名	支持重命名存储桶中的文件。
新建文件夹	支持在存储桶中新建文件夹。
查看文件详情	支持查看存储桶中的文件基本信息。
生成文件链接	支持通过请求临时签名的方式生成带有时效的文件访问链接。
文件/文件夹分享	支持分享文件和文件夹，支持设置分享的有效时间。
导出文件链接	支持批量导出文件链接。
文件预览	支持预览存储桶中的媒体文件（图片、视频、音频）。

搜索文件	支持以前缀搜索的方式对存储桶中的文件进行搜索。
搜索存储桶	支持搜索已创建的存储桶。
查看历史版本或文件碎片	支持在已开启版本控制的存储桶中，查看文件的历史版本。 支持查看存储桶内的文件碎片详情。
文件对比	支持将本地文件夹与存储桶里的文件进行对比。
视频转码	支持对已开启媒体处理功能的存储桶中的文件进行转码。
生成授权码	通过授权码可实现临时登录 COSBrowser 客户端。
图片处理	支持缩放、裁剪、旋转等基本图片处理和文字水印、图片水印，并支持生成处理后的图片链接。
设置网络代理	支持设置网络代理来访问 COS。
设置上传/下载的文件并发数	支持设置文件上传、下载传输并发数。
设置上传/下载的分块并发数	支持设置文件分块上传、下载的分块数。
设置上传/下载失败重试数	支持设置文件上传、下载失败时的重试次数。
设置单线程上传/下载限速	支持设置单线程上传限速和下载限速。
设置上传二次校验	支持对上传至存储桶中的文件进行二次校验。
查看本地日志	支持将用户对 COSBrowser 的操作记录以本地日志的形式保存。

移动端功能列表

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

更新日志

桌面端更新日志：[changelog](#)。

移动端更新日志：[changelog_mobile](#)。

反馈和建议

如您在使用 COSBrowser 时有任何疑问或建议，欢迎反馈给我们：

桌面端反馈：[issues](#)。

移动端反馈：[issues_mobile](#)。

桌面端使用说明

最近更新时间：2024-04-17 14:39:17

下载与安装

软件下载

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

软件安装

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

注意

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

登录软件

COSBrowser 桌面端支持通过三种方式登录：永久密钥登录、腾讯云账号登录、共享链接登录。支持同一账号多设备多点登录。

永久密钥登录

通过云 API 密钥（即 SecretID、SecretKey）进行登录使用，该密钥可在访问管理控制台的 [API 密钥管理](#) 页面中创建并获取，成功登录后密钥将保存在 **历史会话** 中，方便下次继续使用，软件登录界面如下所示。其他配置项说明如下：

存储桶/访问路径：若当前使用的密钥其被主账号允许访问的范围只有存储桶或存储桶下某个目录的权限，那么此项必填。填写后可快速进入对应的文件路径。格式为：`Bucket` 或 `Bucket/Object-prefix`，例如当前使用的密钥仅被允许访问存储桶 `examplebucket-1250000000` 下的 `doc` 文件夹，则填 `examplebucket-1250000000/doc`。

备注：可对当前填入的永久密钥进行说明。例如操作人员、用途等。在历史密钥界面中管理历史会话时可区分不同的 SecretID。

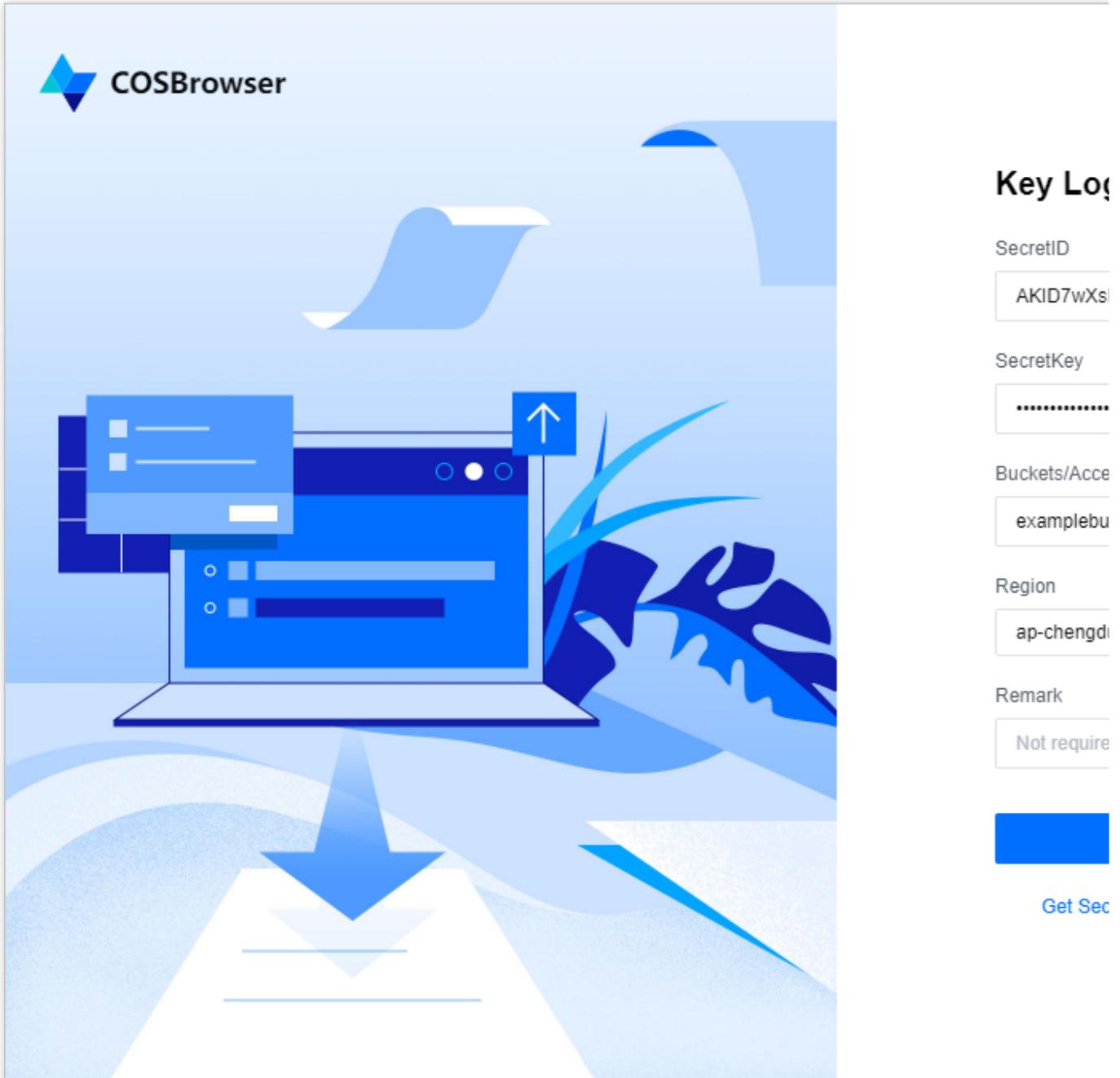
记住会话：

若不勾选，则仅本次登录有效，退出登录则会清空填入的云 API 密钥（如果当前密钥已保存在历史会话中，则会从历史会话中移除）；

若勾选，则会记住填入的云 API 密钥，并可在历史会话中进行管理。

注意

COSBrowser 不支持使用项目密钥进行登录。



腾讯云账号登录

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

共享链接登录

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

基本功能

注意

此处功能介绍以 Windows 版本 v 2.11.1 为例，其他版本差异，请参见[更新日志](#)。

1. 创建/删除存储桶

功能	说明	如何操作
创建存储桶	可以通过客户端直接创建存储桶	<ol style="list-style-type: none">1. 在存储桶列表，单击左上角创建桶2. 正确填写存储桶名称，并选择地域、访问权限，也可根据用户需求设置存储桶标签3. 单击确定，即可完成创建
删除存储桶	删除存储桶前，请确认存储桶中数据已清空	<ol style="list-style-type: none">1. 在存储桶列表，单击对应存储桶右侧的 ... > 删除2. 确认存储桶中的数据已全部清空，单击确定进行删除

2. 查看存储桶详情

您可通过单击存储桶列表右侧的**详情**图标，查看存储桶详情。存储桶详细信息包含存储桶名称、地域、访问权限，多 AZ 特性、版本控制、全球加速等状态以及存储桶域名。

注意

多 AZ 特性目前仅支持部分地域，例如北京、广州、上海、新加坡地域。有关多 AZ 特性的说明可参见[多 AZ 特性概述](#)。

3. 查看统计数据

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

4. 权限管理

COSBrowser 支持对存储桶及文件的权限管理。

存储桶权限管理：可通过单击存储桶列表右侧操作栏中的**权限管理**进行操作。

对象权限管理：可通过单击文件右侧操作栏中的**权限管理**进行操作。

说明

有关 COS 的权限说明，可查看 [ACL概述](#)。

5. 设置版本控制

COSBrowser 支持开启/关闭存储桶版本控制。

可通过单击存储桶列表右侧操作栏中的**详情**，在存储桶详情弹窗中，单击版本控制编辑图标进行操作。

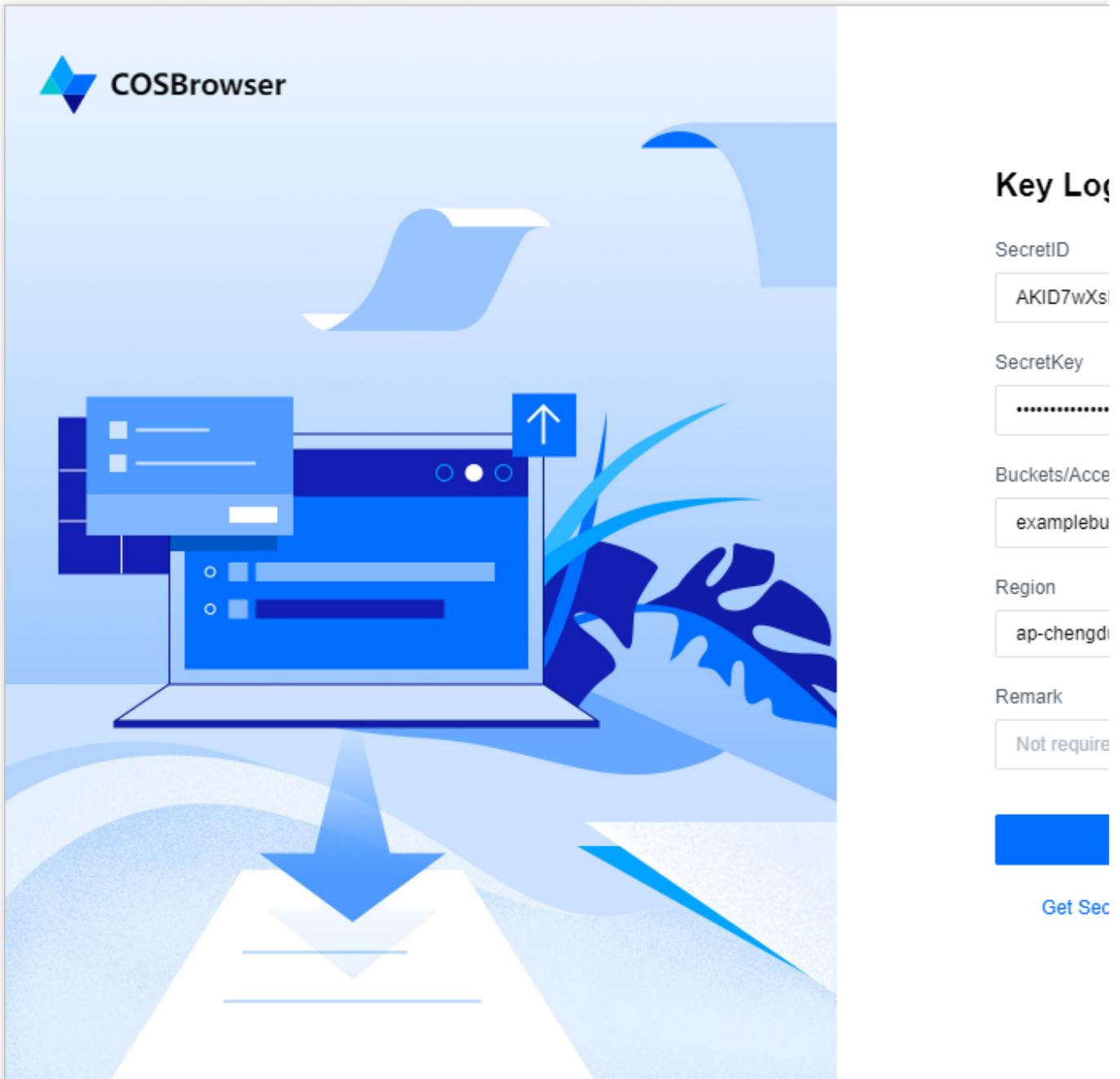
说明

有关版本控制的说明，可查看 [版本控制概述](#)。

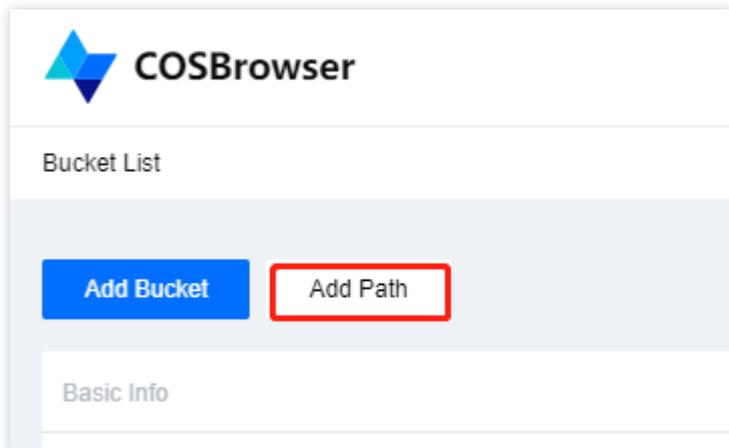
6. 添加访问路径

若您使用无访问存储桶列表权限的子账号进行登录，可以通过**添加访问路径**的方式进行访问，COSBrowser 提供了两种添加访问路径的方式：

- (1) 在登录界面直接添加访问路径，并选择好对应的存储桶地域信息，登录完成后，即可管理资源。



(2) 子账号登录后，在存储桶列表页左上角，单击**添加路径**，并输入指定的路径进入存储桶管理资源。



7. 上传文件/文件夹

上传功能	说明	如何操作
上传文件	COSBrowser 支持多种上传方式，支持单个或批量上传	上传文件的几种方式如下，在指定的存储桶或路径内： <ol style="list-style-type: none"> 1. 单击上传，选择文件后，直接上传文件。 2. 在文件列表空白处右键单击上传文件，进行上传。 3. 通过鼠标将文件拖拽至文件列表窗口进行上传。
上传文件夹及其文件	若存储桶或路径内存在同名文件或文件夹，则默认覆盖	上传文件夹的几种方式如下，在指定的存储桶或路径内： <ol style="list-style-type: none"> 1. 单击上传，选择文件夹，直接上传文件夹。 2. 在文件列表空白处右键选择上传文件夹，进行上传。 3. 通过鼠标将文件夹拖拽至文件列表窗口即可完成上传。
增量上传	增量上传指执行上传操作前，对上传文件与存储桶已有对象做比对，若存在同名对象，则跳过该文件不执行上传操作	增量上传操作步骤如下，在指定的存储桶或路径内： <ol style="list-style-type: none"> 1. 使用上传文件夹的方式，单击下一步。 2. 存储方式选择跳过，单击上传后，即可完成增量上传。

注意

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

8. 下载文件/文件夹

下载功能	说明	如何操作
------	----	------

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

注意

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

9. 删除文件/文件夹

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

10. 文件同步

用户可以通过文件同步功能，将指定本地文件夹中的文件自动实时地上传或者定时上传至存储桶中。具体操作步骤如下：

1. 单击界面右上方的**工具箱>文件同步**。

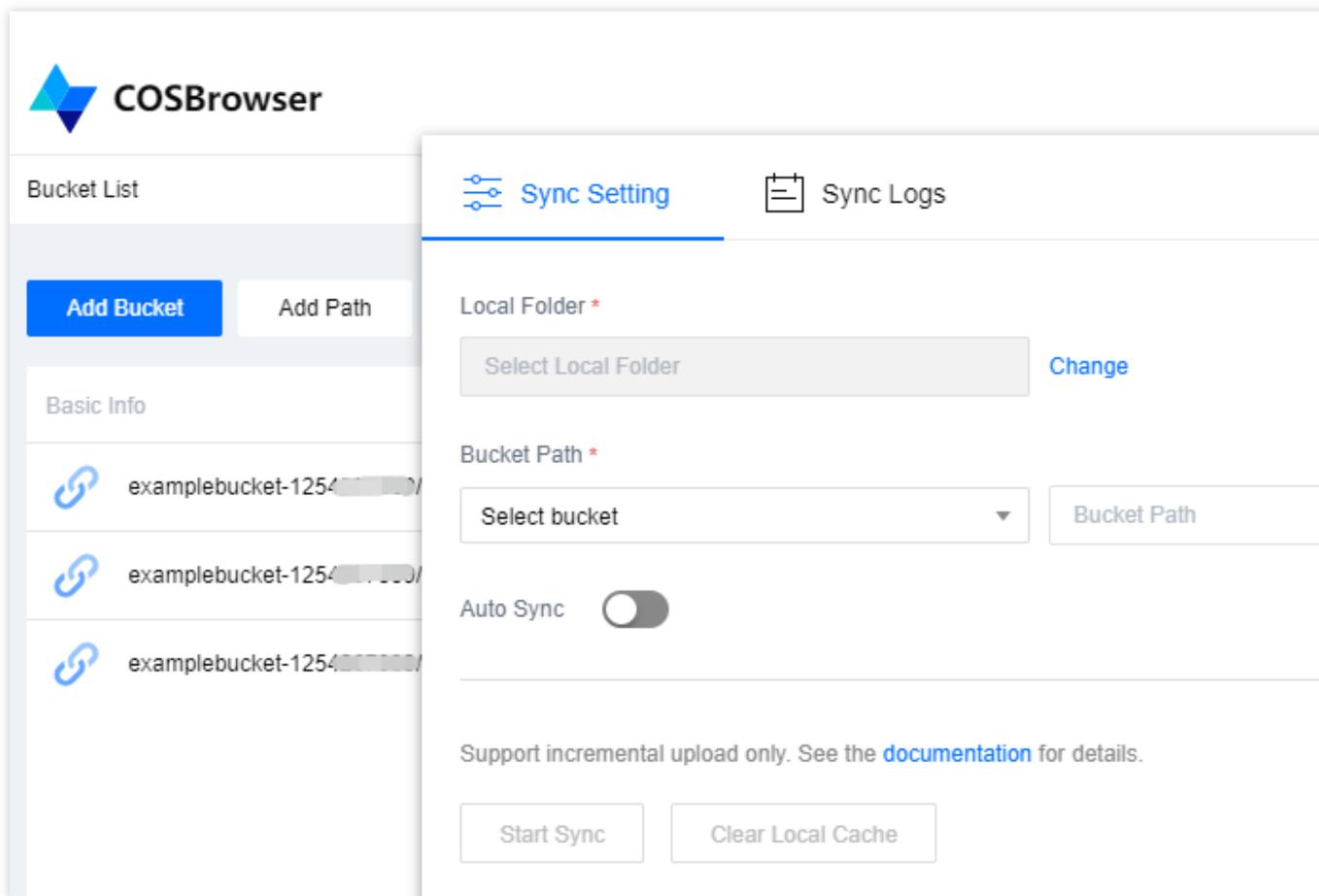
2. 在弹窗中配置以下信息：

本地文件夹：指定将要同步上传的本地文件夹。

存储桶目录：上传到 COS 的存储桶目录。

文件后缀过滤：需要过滤的文件后缀。在同步时，将忽略带有该后缀的文件，如有多个，以英文 `;` 分隔。

同步类型：可选择单次同步、自动同步或者定时同步，然后单击**开始同步**，即可开启文件同步功能。



3. 单击**同步日志**，可查看文件的同步历史记录。

注意

同步是指在上传文件时，系统会自动识别存储桶是否存在同样的文件，通过同步功能仅会上传存储桶中不存在的文件。

目前仅支持将本地文件同步上传至存储桶，不支持逆向操作。

文件同步功能支持设置手动同步、自动同步和定时同步。

11. 复制粘贴文件

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

注意

对于已复制的文件或文件夹，若其粘贴的目标路径中包含同名文件，则默认覆盖。

12. 文件重命名

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

说明

文件夹无法进行重命名操作。

13. 新建文件夹

在指定的存储桶或路径内，单击界面内的**新建文件夹**或右键单击**新建文件夹**，输入文件夹名并确认，完成文件夹的创建。

注意

文件夹名称长度限制在255个字符内，可用数字、英文和可见字符的组合。

文件夹名称不可包含 `\\ / : * ? " | < >` 等特殊字符。

不允许以 `..` 作为文件夹名称。

文件夹无法进行重命名操作，请谨慎命名。

14. 查看文件详情

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

Detail ✕

Name	exampleobject.txt
Size	500KB (512,000B)
Last Modified	2019-09-29 12:11:38
ACL	Inherit
Storage Class	Standard Storage
ETag	"c939165a4566ac3eba011f641e94c519"
Headers	content-type: text/plain; charset=utf-8 x-cos-metadata-directive: Replaced
Object location	https://examplebucket-125-.cos.ap-chengdu.myqcloud.com/exampleobject.txt
Create temporary object url	Create temporary object url

Close

15. 生成文件链接

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

生成文件链接有以下几种方式：

表格视图下，单击文件右侧的分享图标，可一键生成链接并复制。若文件为公有读权限，则链接不带签名永久有效。若文件为私有读权限，则链接带签名，2小时内有效。

选择文件后，右键单击**复制链接**，可一键生成链接并复制。若文件为公有读权限，则链接不带签名永久有效。若文件为私有读权限，则链接带签名，2小时内有效。

在文件详情中，单击**创建临时链接**，可以设置指定域名（例如开启了 CDN 加速域名，则可设置）的临时链接、链接类型、有效时间。

16. 文件/文件夹分享

通过操作栏中的**分享**图标或单击右键菜单中的**分享**选项，可以将 COS 中的某个文件夹分享出去，并且支持设置分享的有效时间。

说明

只支持单个文件夹的分享，不支持批量文件共享。

多人共享文件，很容易造成文件版本混乱。建议开启存储桶的版本控制功能，可以回溯至文件的历史版本。

Share Folders/Files
✕

Name doc ... 1 folders/files.

Permissions Read only Read & write

Valid time hours ▾

The longest valid time is 2 hours (If log in with a sub-account, the maximum validity period can be extended to 1.5 days)

Extraction code Auto generate Custom

ee9b31

Please enter the 6-digit extraction code

OK
Close

参数	说明
权限	您可以设置分享文件夹的访问权限。 只读：通过访问链接可以拉取文件夹列表及下载文件夹中的文件。 读写：通过访问链接可以拉取文件夹列表、下载文件夹中的文件、上传文件至分享文件夹、新建文件夹。
有效时间	单位为分钟、小时或天。 使用密钥登录客户端：主账号有效期的取值范围为 1 分钟到 2 小时；子账号有效期的取值范围为 1 分钟到 30 天。 使用腾讯云账号登录客户端：能设置的最长分享时间都只有 2 小时。默认值为当前账号允许的最长有效期。
提取码	6 位，默认系统自动生成，用户可自定义。支持输入数字、字母、符号

注意

在链接有效期内，收到分享链接及提取码的用户都可以通过该链接访问该文件夹。

17. 导出文件链接

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

18. 文件预览

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

复制链接：生成文件访问链接并复制。

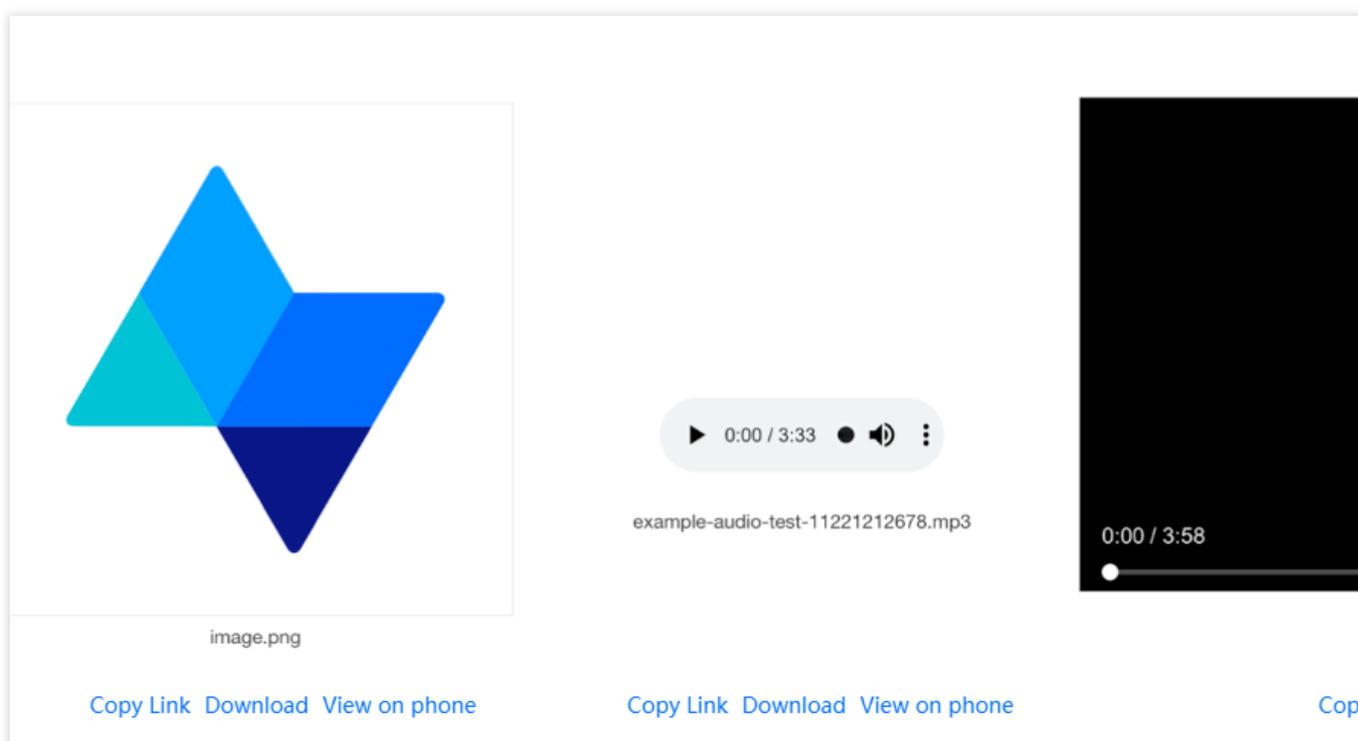
下载：将文件下载至本地，若本地存在同名文件，则默认覆盖。

手机查看：在预览界面会生成文件访问二维码，通过手机扫码可直接在手机上查看此文件。

注意

预览支持大多数图片格式，视频格式仅支持 mp4、webm，音频格式仅支持 mp3、wav。

文件预览会产生下行流量，请酌情使用。



19. 搜索文件

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

20. 搜索存储桶

可通过左侧存储桶列表上方的搜索框，输入存储桶名称快速定位存储桶。

21. 查看历史版本或文件碎片

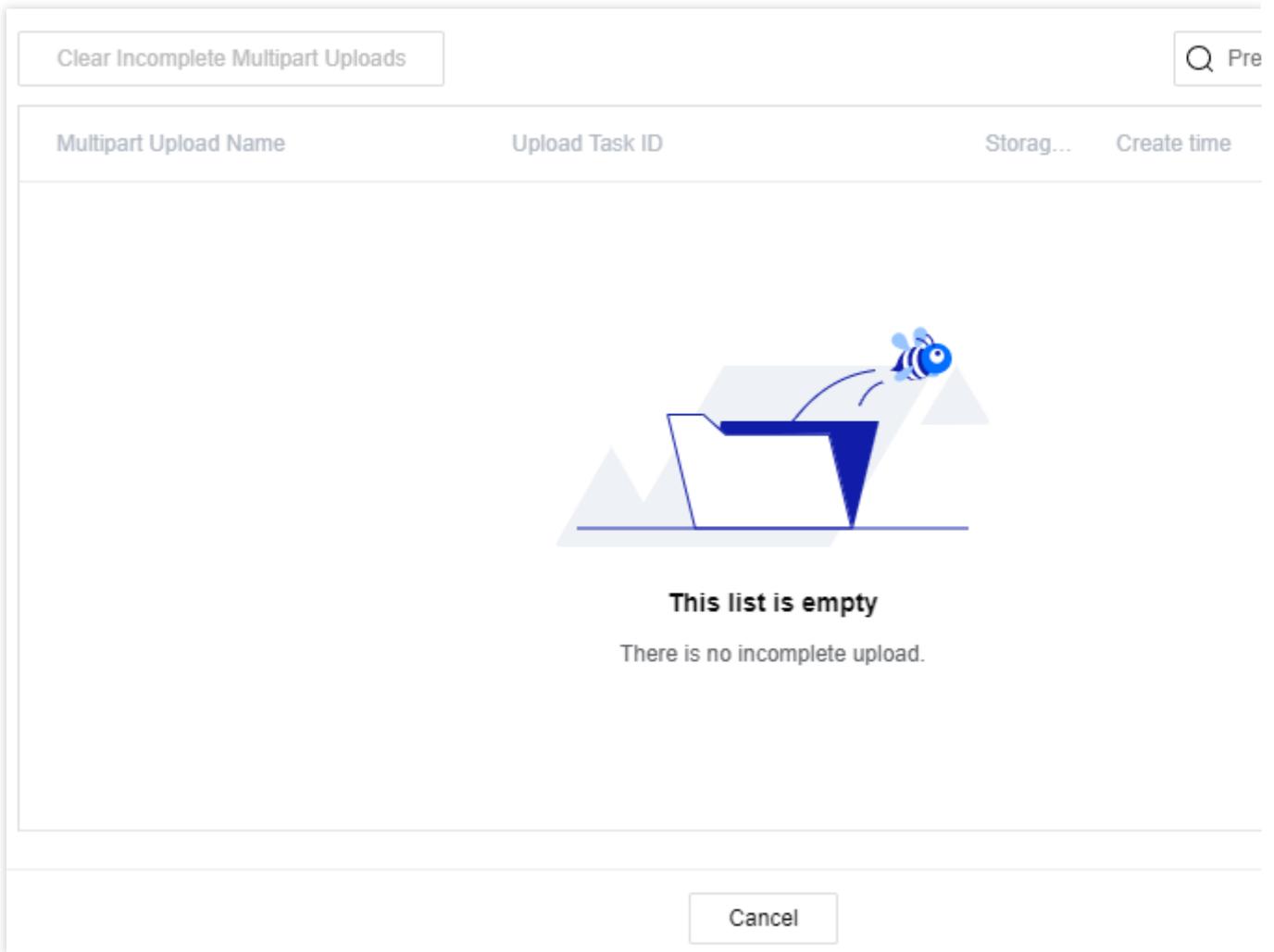
当您的存储桶开启了版本控制后，可单击文件列表上方的**查看 > 历史版本**，查看文件的历史版本。历史版本列表中，支持按前缀搜索和清空所有历史版本（仅保留最新版本）。

Multiversion list

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

Cancel

当您在上传对象过程中，暂停或取消上传时可能产生文件碎片。您可单击文件列表上方的**查看 > 文件碎片**来查看文件碎片，支持按前缀搜索和清空所有文件碎片。



22. 文件对比

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

23. 视频转码

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

24. 生成授权码

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

登录客户端。

操作说明如下：

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

25. 图片处理

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

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

软件设置

系统功能	说明	如何操作
设置网络代理	COSBrowser 默认会使用系统配置的代理来尝试网络连接，请确保您的代理配置正常或请停用无法连接互联网的代理配置	<ol style="list-style-type: none"> 1. 选择高级设置 > 代理。 2. 设置网络代理来进行网络连接。
设置上传/下载的文件并发数	COSBrowser 支持设置上传、下载的文件并发数。	<ol style="list-style-type: none"> 1. 选择高级设置 > 下载/上传。 2. 设定批量传输的并发数。
设置上传/下载的分块并发数	COSBrowser 支持分块上传、下载文件，当传输的文件超过一定大小时，会默认使用分块的方式传输	<ol style="list-style-type: none"> 1. 选择高级设置 > 下载/上传。 2. 设定分块传输的并发数。
设置上传/下载失败重试数	COSBrowser 在文件传输的时候，会默认重试失败的任务	<ol style="list-style-type: none"> 1. 选择高级设置 > 下载/上传。 2. 设定传输失败的重试次数。
设置单线程上传/下载限速	COSBrowser 支持设置单线程上传限速和单线程下载限速。总上传（或下载）限速 = 单线程上传（或下载）限速 x 文件并发数 x 分块并发数	<ol style="list-style-type: none"> 1. 选择高级设置 > 上传（或下载）。 2. 设置单线程上传（或下载）限速，单位为MB/s。
设置上传二次校验	COSBrowser 支持在上传后进行二次校验，检查线上文件大小和状态是否正确	<ol style="list-style-type: none"> 1. 选择高级设置 > 上传。 2. 勾选上传完二次校验。
查看本地日志	COSBrowser 会记录使用者的操作，以 cosbrowser.log 的日志形式保存在本地	<ol style="list-style-type: none"> 1. 选择高级设置 > 关于。

		2. 单击 本地日志 ，系统将打开本地日志所在目录。
--	--	-----------------------------------

移动端使用说明

安装与登录

最近更新时间：2024-01-06 10:43:44

下载与安装

软件下载

支持平台	系统要求	下载地址
Android	Android 4.4及以上	Android
iOS 移动端	iOS 11及以上	iOS

软件安装

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

登录方式

COSBrowser 移动端支持以下登录方式：

邮箱登录：通过邮箱创建或关联了指定邮箱的腾讯云账号，可以通过输入邮箱账号密码进行登录。

永久密钥登录：用户可使用云 API 密钥 `SecretId` 和 `SecretKey`（不支持项目密钥）进行登录，该密钥可在访问管理控制台的 [API 密钥管理](#) 页面中创建和获取，登录后将永久保持该账号的登录状态。

说明

子账号用户可以使用密钥方式登录。

移动端功能列表

最近更新时间：2024-01-06 10:43:44

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

数据监控

操作名	操作描述
用量概览	支持展示近期的数据使用情况。
存储桶监控	支持从存储桶维度查看近期的数据使用情况。
小组件	数据概览支持小组件模式，在不打开 App 的情况下也能浏览数据使用情况。

存储桶管理与操作

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

文件操作与管理

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

存储桶管理与操作

最近更新时间：2024-01-06 10:43:44

说明

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

查看存储桶列表

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

添加访问路径

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

说明

仅子账号支持该功能。

操作步骤

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

创建存储桶

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

操作步骤

1. 在存储桶列表中，点击右上角**+**。
2. 在弹出的操作列表中，点击**创建存储桶**。
3. 在弹出的创建存储桶页面，配置如下信息：

名称：请输入自定义的存储桶名称。设置后不可修改。命名说明请参见存储桶的 [命名规范](#)。

所属地域：请选择与您业务（或用户数量）相对集中的物理区域所对应的 COS 地域，设置后不可修改。地域的更多信息请参见 [地域和访问域名](#)。

访问权限：存储桶默认提供三种访问权限：私有读写、公有读私有写和公有读写，设置后仍可修改。详细信息请参见 [存储桶访问权限](#)。

4. 确认信息无误后，点击**确定**，即可创建存储桶。

在存储桶列表界面中，您可以看到刚才已创建的存储桶。

搜索存储桶

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

查看存储桶基本信息

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

基本信息包含存储桶名称、地域、创建时间及是否开启多 AZ。

存储桶权限管理

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

公共权限：私有读写、公有读私有写和公有读写。关于公共权限的说明，请参见存储桶概述中的 [权限类别](#)。

用户权限：主账号默认拥有存储桶所有权限（即完全控制）。另外 COS 支持添加子账号有数据读取、数据写入、权限读取、权限写入，甚至**完全控制**的最高权限。

1. 进入存储桶列表页，找到您需要设置或修改访问权限的存储桶，点击右侧的****...****。
2. 在弹出的操作列表中，点击**权限**，即可进入存储桶权限页面。

修改公共权限

在存储桶权限页面，点击公共权限配置项，即可修改公共权限。

设置用户权限

在存储桶权限页面，点击添加用户，即可设置存储桶的访问权限。

编辑或删除用户权限

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

开启全球加速

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

说明

开启全球加速功能，不会影响原有的存储桶默认域名，您仍然可以正常使用。

操作步骤

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

存储桶传输配置

如果您想让某个桶通过自定义域名进行传输，那么您可以在该页面进行配置。

设置上传域名

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

设置下载域名

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

文件管理与操作

最近更新时间：2024-01-06 10:43:44

说明

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

新建文件夹

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

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

删除文件夹

注意

删除文件夹会删除该文件夹以及所有子目录下的文件。

操作步骤

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

上传文件

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

上传图片 and 视频

COSBrowser 支持将图库的图片或者视频批量上传到 COS。

操作步骤

1. 进入存储桶列表页或者文件列表页，点击右上角**+**，弹出操作列表。
2. 在操作列表中，点击**上传图片**。

3. 在弹出图库的文件列表，选中要上传的文件，点击**下一步**。

4. (可选) 配置上传参数，COSBrowser 支持在上传时设置文件的属性，目前支持设置以下属性：

存储类型：根据不同的业务场景，您可以为不同的对象设置不同的存储类型，默认存储类型为标准存储。关于存储类型的介绍，请参见 [存储类型](#)。

注意

若您的存储桶开启了多 AZ 配置，则**存储类型**只能选择为拥有多 AZ 特性的存储类型，例如标准存储（多 AZ）；若同时还开启了智能分层存储配置，则还可选择智能分层存储（多 AZ）类型。

访问权限：您可以为不同的对象设置不同的访问权限，默认访问权限为继承权限（即继承存储桶权限）。关于访问权限的更多信息，请参见 [访问控制基本概念](#)。

服务端加密：您在上传对象的同时，配置对象的服务器加密属性。腾讯云 COS 将为上传的对象添加数据加密的保护策略，帮助您在数据写入前自动加密，并在您访问该数据时自动解密。目前支持 SSE-COS、SSE-KMS（仅支持北京、上海、广州地域）。详情请参见 [服务端加密概述](#)。

对象标签：对象标签由标签的键（tagKey）、标签的值（tagValue）与“=”相连组成，例如 group = IT。您可以对指定的对象进行标签的设定、查询、删除操作。

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

5. 点击**上传**，即可对图库中的图片和视频进行批量上传。

上传文件链接

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

您也可以通过下面的步骤主动上传文件链接：

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

上传第三方分享

您也可以将其他 App 的文件分享到 COSBrowser 进行上传。

注意

该功能需要第三方应用支持将文件分享到其他 App。

操作步骤

以 QQ App 为例：

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

上传文件管理器文件

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

操作步骤

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

文件备份

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

注意

图库备份功能只支持主账号。

相册模块只显示图片和视频文件，如果想要查看完整文件，请前往存储桶列表，搜索备份存储桶进行查看。

设置备份

前往**个人 > 相册备份**，点击开启自动备份照片或视频，如下图所示：

自动备份照片：开启之后会备份图库中的所有图片。

自动备份视频：开启之后会备份图库中的所有视频。

仅在 WIFI 环境下备份：开启之后只会在 Wi-Fi 环境下备份文件。

所属地域：选择备份地域后，我们会在该地域默认创建一个名称格式为“from-phone-日期-APPID”的存储桶。

注意

所属地域只有在开启备份前可设置，保存后将不可修改。

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

管理备份文件

COSBrowser 支持对文件进行批量上传和下载操作。

操作步骤

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

将文件添加到备份存储桶

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

操作步骤

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

文件下载

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

下载到 App

COSBrowser 提供了多种下载入口，方便您随时随地下载文件。

方式一：

- 1.1 进入文件列表页面，找到要下载的文件，单击右侧的**...**。
- 1.2 在弹出的操作列表，点击**下载**即可。

方式二：

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

方式二：

进入文件预览页面，点击右上角的**...，弹出操作列表，单击**下载**即可。

保存到图库

COSBrowser 支持将图片保存到本地图库，操作步骤如下：

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

批量操作

COSBrowser 支持批量下载、删除、复制、移动存储桶中的文件。

批量下载

1. 点击存储桶，进入文件列表页，单击右上角**...**。
 2. 在弹出的操作菜单，点击**批量操作**。
 3. 批量选择文件，点击底部操作栏的**下载**，即可将选中的文件批量下载至 App。
- 您还可以点击文件列表页右上角的传输按钮，进入传输列表页查看相关任务。

批量删除

1. 点击存储桶，进入文件列表页，单击右上角**...**。

2. 在弹出的操作菜单，点击**批量操作**。
3. 批量选择文件，点击底部操作栏的**删除**，即可将选中的文件批量删除。

批量复制

说明

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

1. 点击存储桶，进入文件列表页，单击右上角**...**。
2. 在弹出的操作菜单，点击**批量操作**。
3. 批量选择文件，点击底部操作栏的**更多 > 复制**，即可将选中的文件批量复制到新的存储桶或者文件夹。

批量移动

说明

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

1. 点击存储桶，进入文件列表页，单击右上角**...**。
2. 在弹出的操作菜单，点击**批量操作**。
3. 批量选择文件，点击底部操作栏的**更多 > 移动**，即可将选中的文件批量移动到新的存储桶或者文件夹。

文件预览

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

图片在线预览

COSBrowser 支持在不下载图片的情况下实现图片的在线预览。

开启图片预览

您可以通过以下三种方式实现图片预览：

1. 全局开启

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

2. 一次性全局开启

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

3. 单个图片开启预览

如果您在**个人 > 设置**中关闭了图片预览并且非首次打开 App，那么进入图片详情的时候，会提示您是否开启预览，点击**点击预览**，即可实现对单个图片的预览。

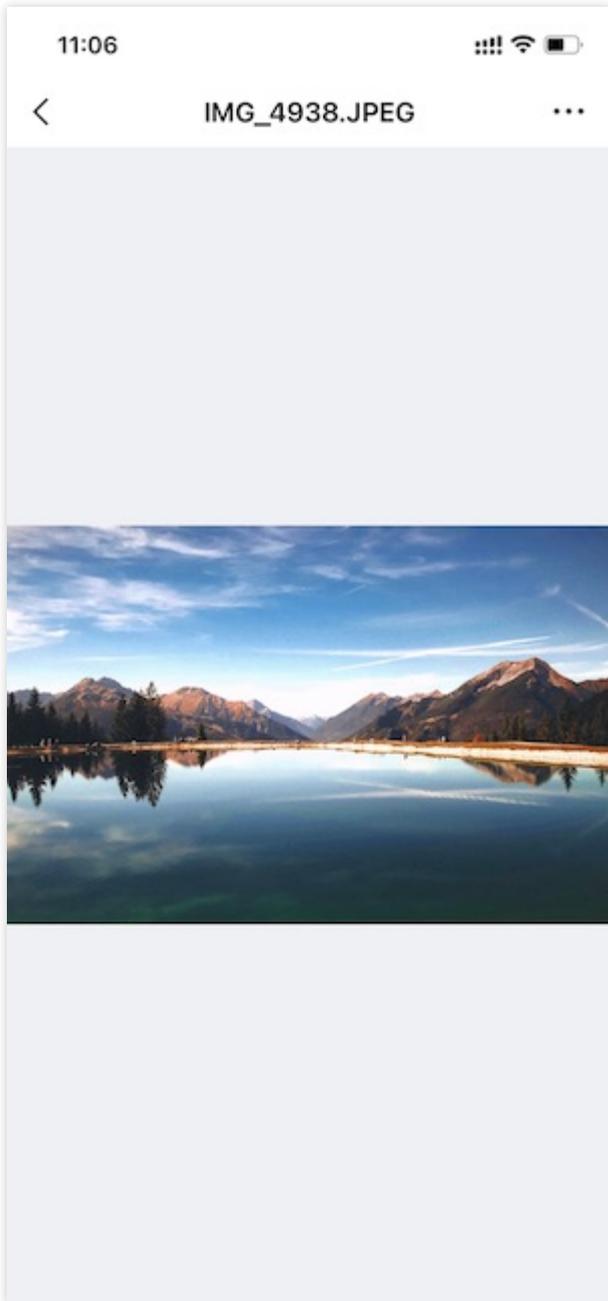
说明

该开关不会修改全局**图片预览**开关的状态。

大图模式

COSBrowser 提供了大图模式功能，可以更加清晰的查看图片细节。

1. 在文件列表页，点击想要查看大图的图片，进入文件详情页。
2. 点击图片，即可进入大图查看模式，在该模式下您可以双指缩放查看图片的细节。



音频播放

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

操作步骤

1. 进入文件列表页，点击音频文件，进入文件详情。

2. 点击播放按钮，进行播放。

在播放的过程中，可以进行快进、暂停。

视频播放

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

操作步骤

1. 进入文件列表页，点击需要播放的视频文件，进入文件详情。

2. 点击播放，进行播放。

在播放的过程中，可以进行快进、暂停。

说明

在 App 进入后台后，仍然可通过悬浮窗的方式观看视频。

文档预览

COSBrowser 支持对 PDF 等多种格式的文件进行预览。

操作步骤

1. 点击要预览的文档，进入文件详情页。如果尚未开启文档预览功能，单击**点击开启**。

2. 进入文档预览功能配置页面，单击**立即开启**，即可开启文档预览。

3. 开启成功后，返回上级页面，单击**点击预览**。

4. 进入文档预览页面，左滑可查看下一页。

文件解压

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

1. 进入文件列表页，点击压缩文件右侧的**** ... ****。

2. 弹出操作列表，单击**在线解压**。

3. 解压成功后，当前目录下会出现解压后的文件。

文件分享

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

分享文件夹

说明

只支持单个文件夹的分享，不支持批量文件共享。

主账号可设置的最长分享时间为2小时，子账号可设置的最长分享时间为1.5天。

多人共享文件，很容易造成文件版本混乱，建议开启存储桶的版本控制功能，可以回溯至过去的文件版本。

生成分享二维码/链接

具体操作步骤如下：

1. 点击文件夹右侧的**...**选项，弹出操作列表，点击分享。
2. 在弹出的分享页面中，可选择二维码分享或链接分享。
3. （可选）配置分享参数。配置项说明如下。可不设置，保持默认设置。

权限 您可以设置分享文件夹的访问权限。

只读：通过访问链接可以拉取文件夹列表及下载文件夹中的文件。

读写：通过访问链接可以拉取文件夹列表、下载文件夹中的文件、上传文件至分享文件夹、新建文件夹。

有效时间：单位为分钟、小时或天。移动端默认设置主账号有效期为2小时；子账号有效期为24小时。移动端的有效时间无法设置。

提取码：6位，默认系统自动生成，用户可自定义。支持输入数字、字母、符号。

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

查看他人分享的文件夹

您可以通过移动端、桌面端、浏览器三种方式打开他人分享过来的文件夹。

方式一：在移动端查看

1. 在**登录页**或**个人页**，点击**扫一扫**扫描分享二维码。
2. 输入提取码，点击**确定**进入分享文件夹。

方式二：在桌面端查看

1. 在登录界面点击**共享链接登录**。
2. 输入已获取的 URL 地址、提取码并登录。

方式三：在浏览器查看

1. 打开浏览器，输入分享的 URL 地址，打开链接。
2. 输入提取码，点击**提取**进入分享文件夹。

分享文件

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

注意

使用非永久密钥方式登录时，无法配置文件链接的有效时长，默认1小时内有效。

若文件为公有读权限，则链接不携带签名，永久有效。若文件为私有读权限，则链接携带临时签名，在1小时内有效。

操作步骤

1. 点击文件，进入文件详情界面。

配置项说明如下：

指定域名：可设置链接的域名（可选）。

有效时长：设置链接有效时间（可选）。

2. 确认配置无误后，点击生成链接。

3. 发送已生成的文件链接。

文件重命名

说明

文件夹无法进行重命名操作。

操作步骤

1. 单击文件右侧的**...**。

2. 在弹出的文件操作列表，点击**重命名**。

3. 在弹出框中，输出新的文件名，点击**确定**即可。

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

文件搜索

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

关键字搜索

输入搜索的关键字，会筛选出当前文件夹以及所有子文件夹下包含该关键字的文件。

类型搜索

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

操作步骤

以搜索文件夹为例：

1. 点击搜索框后，点击文件夹。

2. 该目录下所有的该类型的文件（在该例子中是文件夹）都会被罗列出来。

排序或筛选对象

COSBrowser 支持对存储桶内的文件进行排序、筛选操作。

说明

当前支持对文件列表按文件名、大小、修改时间进行排序，按存储类型进行筛选。

操作步骤

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

文件权限管理

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

注意

对象的访问权限只在用户通过默认域名访问时有效。通过 CDN 加速域名和自定义域名访问时，以存储桶访问权限为准。

访问策略规则存在数量限制，详情请参见 [规格与限制](#)。

修改公共权限

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

设置用户权限

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

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

数据监控

最近更新时间：2024-01-06 10:43:44

COSBrowser 移动端推出数据概览功能，展示了您在使用对象存储（Cloud Object Storage, COS）近30天的数据使用情况，包含了总流量、总读写请求、存储量趋势、流量（外网下行流量、内网流量、CDN 回源流量）趋势、请求数趋势、有效请求比例趋势、低频存储数据取回量、归档存储数据取回量。

说明

移动端目前不支持子账号查看数据，如需登录子账号查看数据，请使用控制台方式进行查看，操作指引请参见 [查看数据概览](#)。

该功能的数据并非实时数据，延时约为2小时。该数据仅为监控数据做参考，如需查看准确的计费计量数据，请使用控制台方式进行查看，操作指引请参见 [查看数据概览](#)。

用量概览

COSBrowser 提供存储数据概览页面，您可以在该页面中查看对象数量、存储用量、请求数以及流量等数据。

存储桶监控

COSBrowser 提供了以存储桶维度的数据概览，可以通过下面两种方式查看存储桶的监控数据。

1. 点击下方的**首页**，点击**用户概览**，即可切换查看每个存储桶的监控数据。
2. 点击下方的**资源**，找到存储桶，点击存储桶右边的**更多**按钮，弹出操作列表，点击**监控**，即可进入存储桶页面。

小组件

COSBrowser 支持以小组件的方式添加到桌面，可以在不打开 COSBrowser 的情况下，即可随时随地查看监控数据。

添加小组件

1. 下载并安装 iOS COSBrowser App 后，长按界面任意处至界面抖动，点击界面右上角加号。
2. 找到 COSBrowser 并点击。
3. 选择您喜欢的布局，点击底部添加按钮。

自定义展示数据

小组件还支持通过配置自定义想要展示的数据范围，数据监控范围可指定某一个特定的存储桶或某一种存储类型进行跟踪监控。存储类型支持展示包括：标准存储、低频存储、归档存储。

具体操作步骤如下：

1. 打开 iOS COSBrowser App，点击**我的 > 设置 > 小组件配置**，进入小组件设置页面。
2. 按照以下配置项说明进行配置：

选择桶：如果未选择存储桶，则默认展示当前账号所有存储桶的概览数据；如果选择某个存储桶，则显示已选择存储桶的概览数据。

选择显示数据类型：小组件默认显示标准存储的监控数据，用户可以手动设置其他存储类型，例如标准存储、低频存储、归档存储。

说明

如果您配置了存储桶或者存储类型后，想要快速重置配置，可点击重置按钮。重置后，即展示当前用户所有存储桶的标准存储类型数据的监控情况。

移除小组件

长按小组件，点击移除小组件，即可删除。

COSCLI 工具（公测）

COSCLI 简介

最近更新时间：2024-01-06 10:43:44

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

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

功能列表

生成与修改配置文件 - [config](#)

创建存储桶 - [mb](#)

删除存储桶 - [rb](#)

存储桶标签 - [bucket-tagging](#)

查询存储桶或文件列表 - [ls](#)

获取不同类型文件的统计信息 - [du](#)

上传下载或拷贝文件 - [cp](#)

同步上传下载或拷贝文件 - [sync](#)

删除文件 - [rm](#)

获取文件哈希值 - [hash](#)

列出分块上传中产生的碎片 - [lsparts](#)

清理碎片 - [abort](#)

取回归档文件 - [restore](#)

获取预签名 URL - [signurl](#)

下载与安装配置

最近更新时间：2024-06-27 14:41:28

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

步骤一：下载 COSCLI 工具

您可以根据业务场景选择下载 COSCLI 工具的地址。如果您的服务器在国内，推荐使用国内站点下载地址。

国内站点 下载地址	Github 下载 地址（国外 站点使用）	SHA256 校验
Windows-386	Windows-386	654d197278681ffd7ee6d4ae2343a6f42cc5a1d7b8d9f0c304e3de490d754d0e
Windows-amd64	Windows-amd64	2b7510f1493f79ccef0987574198d10a5d14e998db3bf5d3fda1fec579620cbc
Mac-amd64	Mac-amd64	b83a4ebcb0722dc2f388262ecd59b37213969cc57b9bf49f7f6feff3da1d8887
Mac-arm64	Mac-arm64	fc28d75eec8452f1461600c404bf9a359b765dd351fe0c7245f09cce062e200c
Linux-386	Linux-386	8edce114a86790e773e90ceaf785de3a9d6ef9ec51788c03414096ee7b138527
Linux-amd64	Linux-amd64	a63592af60ddf8d12ab137fbfae5aab253a2ac7c0b2bc5571dbafcd7cac9477d
Linux-arm	Linux-arm	bf530c1d6e0583139e8e94854d93cb5aa0210dc6c6907c8602eec45f04239dca
Linux-arm64	Linux-arm64	aad0466d6c340a00aeaab8879d1e2695df0387edae48d38aceec6e4777f38efa

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

```
Mac-amd64 : wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-v1.0.0-darwin-amd64
```

```
Mac-arm64 : wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-v1.0.0-darwin-arm64
```

```
Linux-386 : wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-v1.0.0-linux-386
```

```
Linux-amd64 : wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-v1.0.0-  
linux-amd64
```

```
Linux-arm : wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-v1.0.0-  
linux-arm
```

```
Linux-arm64 : wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-v1.0.0-  
linux-arm64
```

说明：

Github上当前版本号为 v1.0.0，如需获取工具的最新版本、历史版本和更新日志，请前往 [release](#) 进行查看。

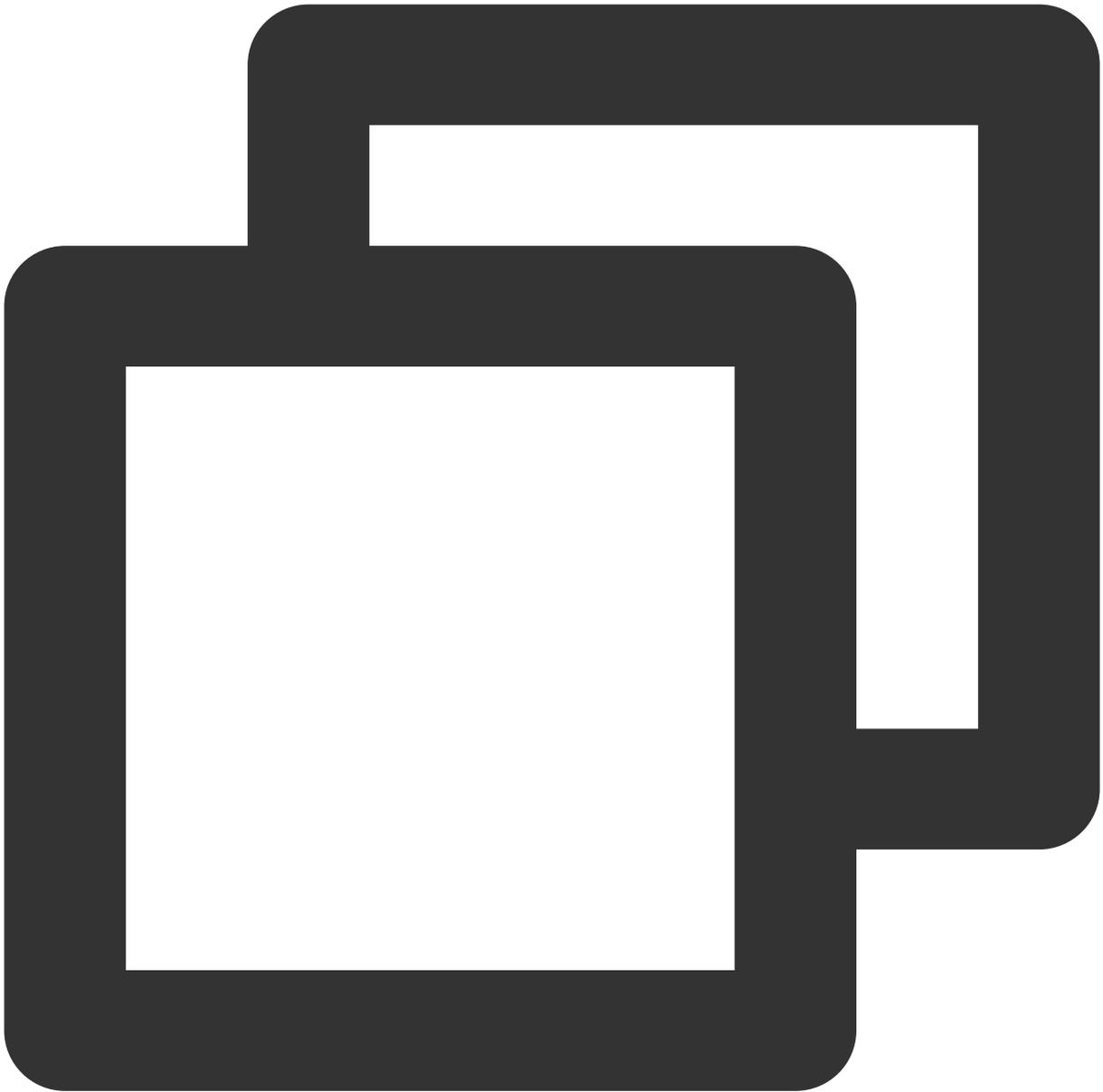
步骤二：安装 COSCLI 工具

Windows

1. 将已下载的 Windows 版本 COSCLI 工具，移动至 C:\Users\<用户名> 目录下。
2. 将 `coscli-windows.exe` 重命名为 `coscli.exe`。
3. 按 `win+r` 键打开 运行 程序。
4. 在对话框中，输入 `cmd`，按 `Enter` 打开命令行窗口。
5. 在命令行窗口中，输入 `coscli --version`，若打印出以下信息则代表安装成功：

说明：

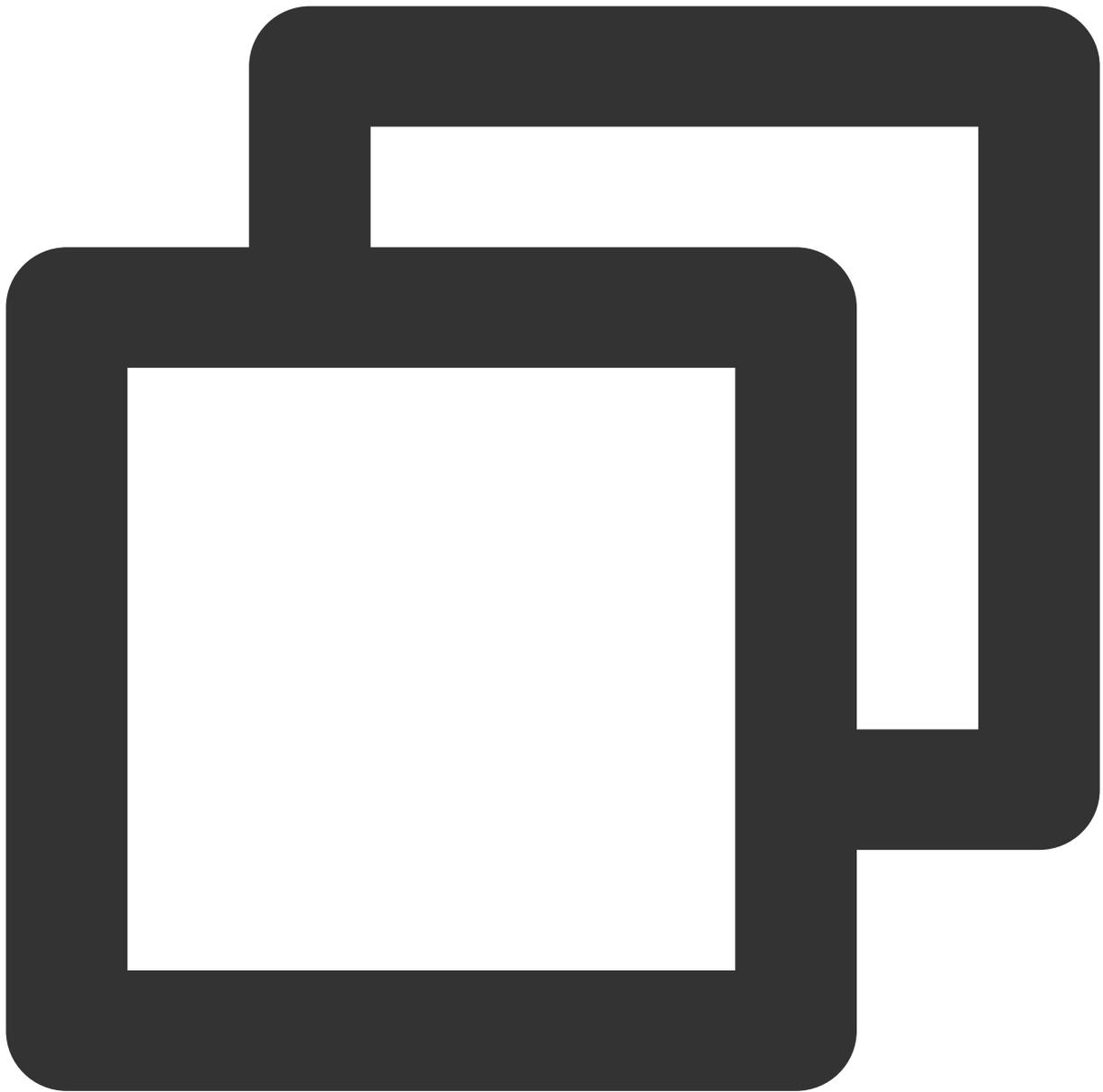
在 Windows 系统下，在不同的命令行客户端中使用 COSCLI 的方式可能略有不同，若输入 `coscli [command]` 后 COSCLI 无法正常工作，请尝试 `./coscli [command]` 的格式。



```
coscli version v1.0.0
```

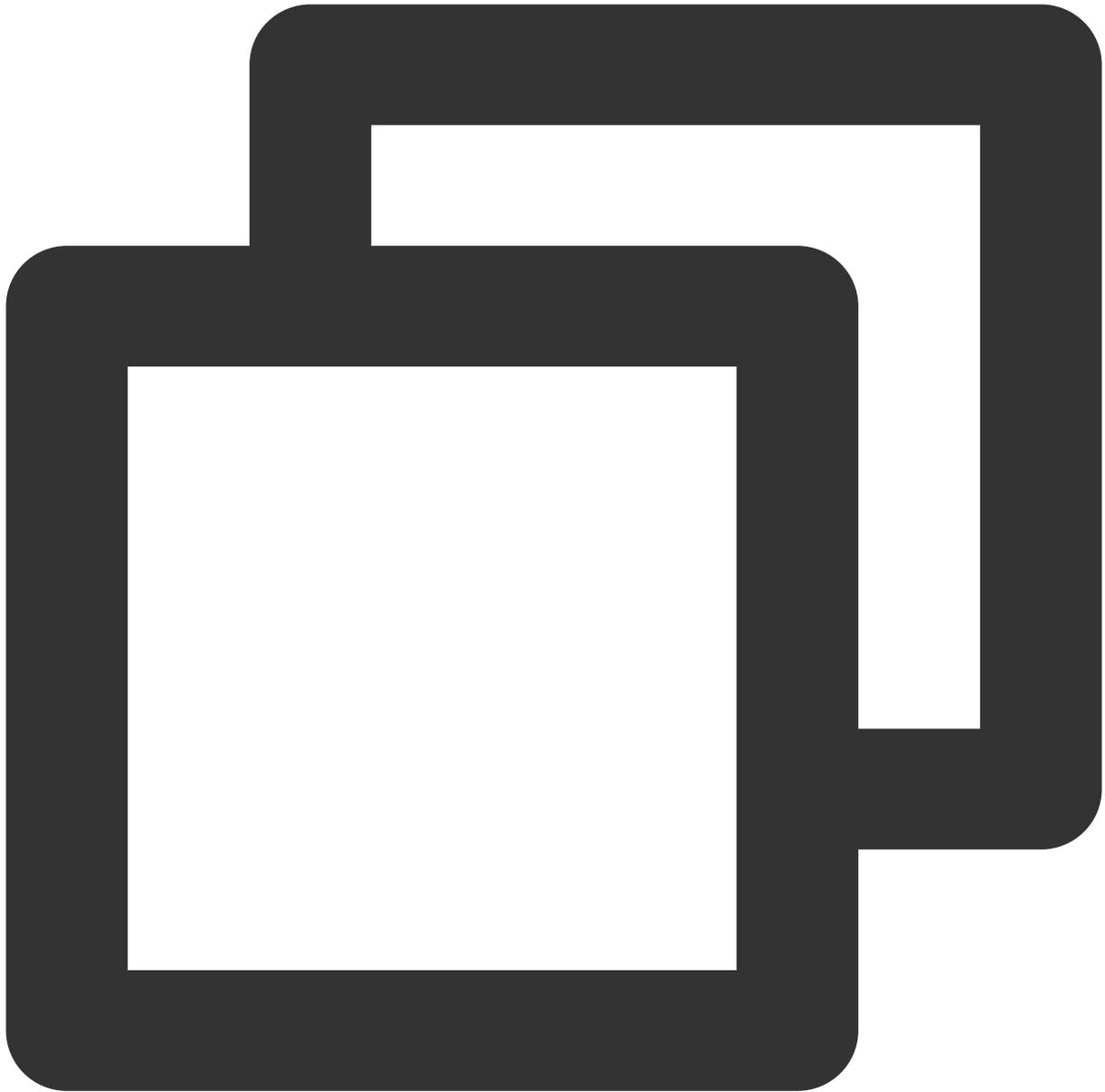
Mac

1. 运行以下命令，重命名 Mac 版本 COSCLI 文件：



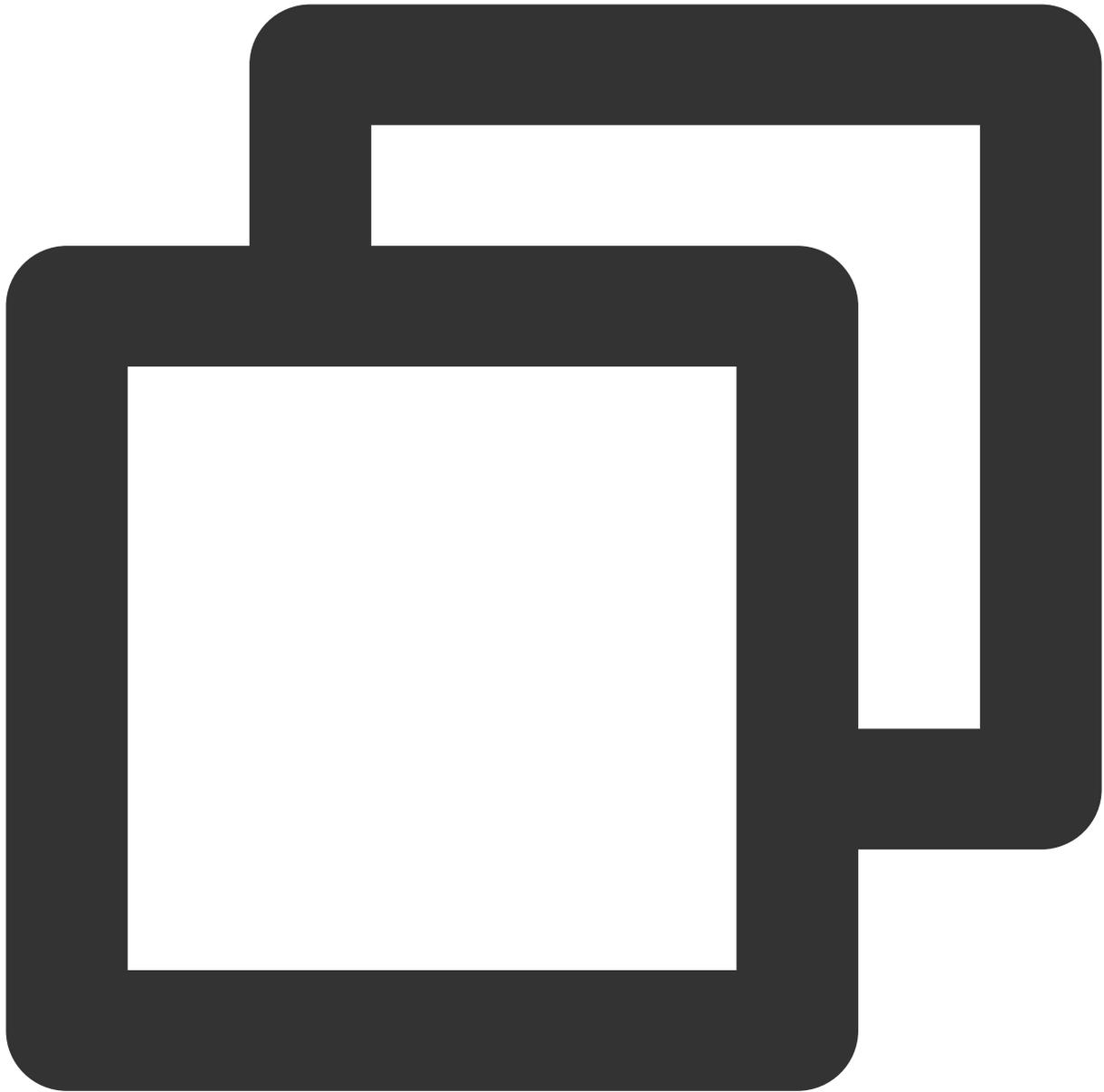
```
mv coscli-mac coscli
```

2. 运行以下命令修改文件执行权限：



```
chmod 755 coscli
```

3. 在命令行中输入 `./coscli --version` ，若打印出以下信息则代表安装成功：



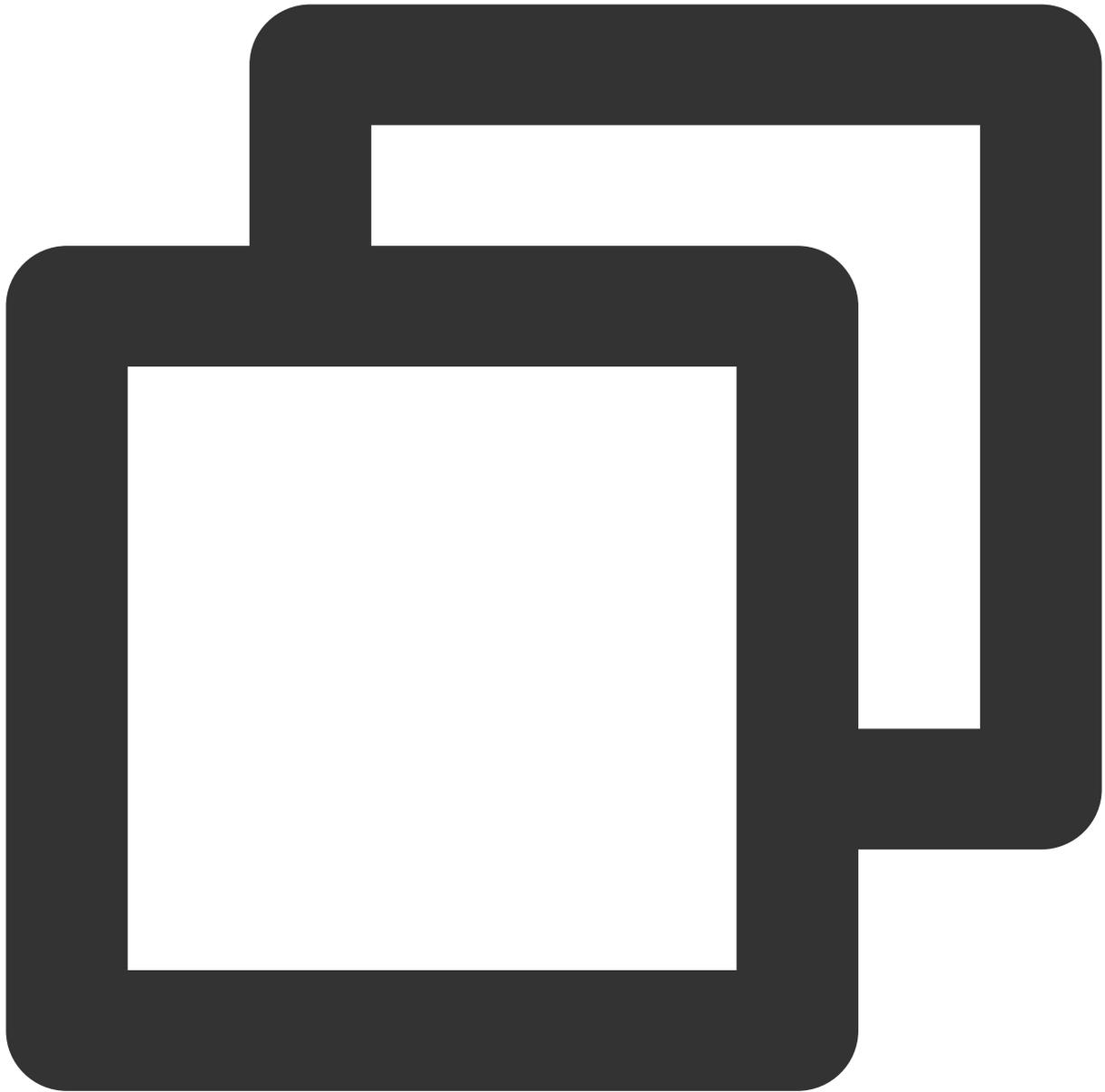
```
coscli version v1.0.0
```

说明：

在 Mac 系统下使用 COSCLI 时，若弹出 无法打开 “coscli”，因为无法验证开发者 的提示，可以前往 [设置 > 安全性与隐私 > 通用](#) 中选择 [仍要打开 coscli](#) ，之后即可正常使用 COSCLI。

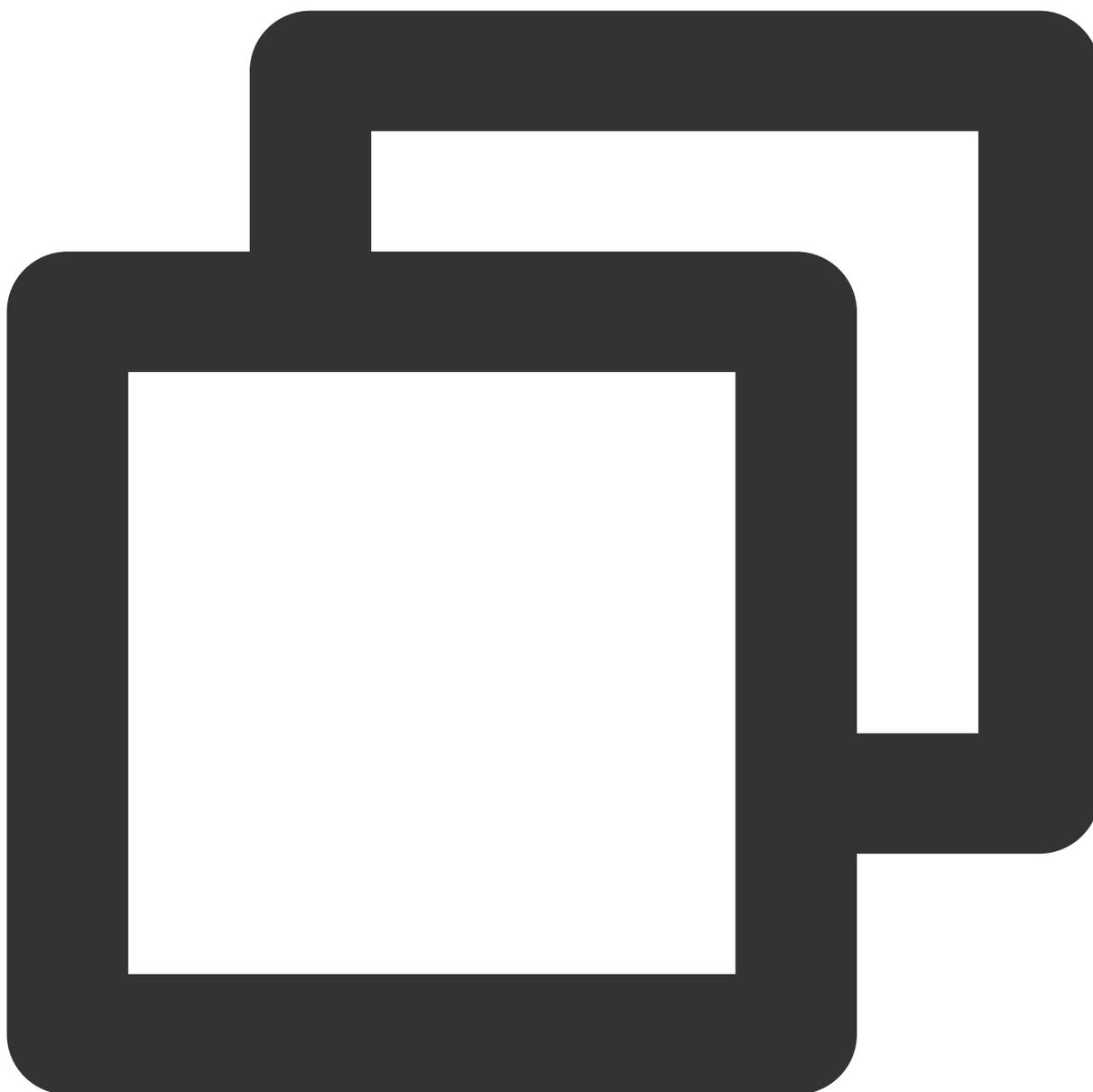
Linux

1. 运行以下命令，重命名 Linux 版本 COSCLI 文件：



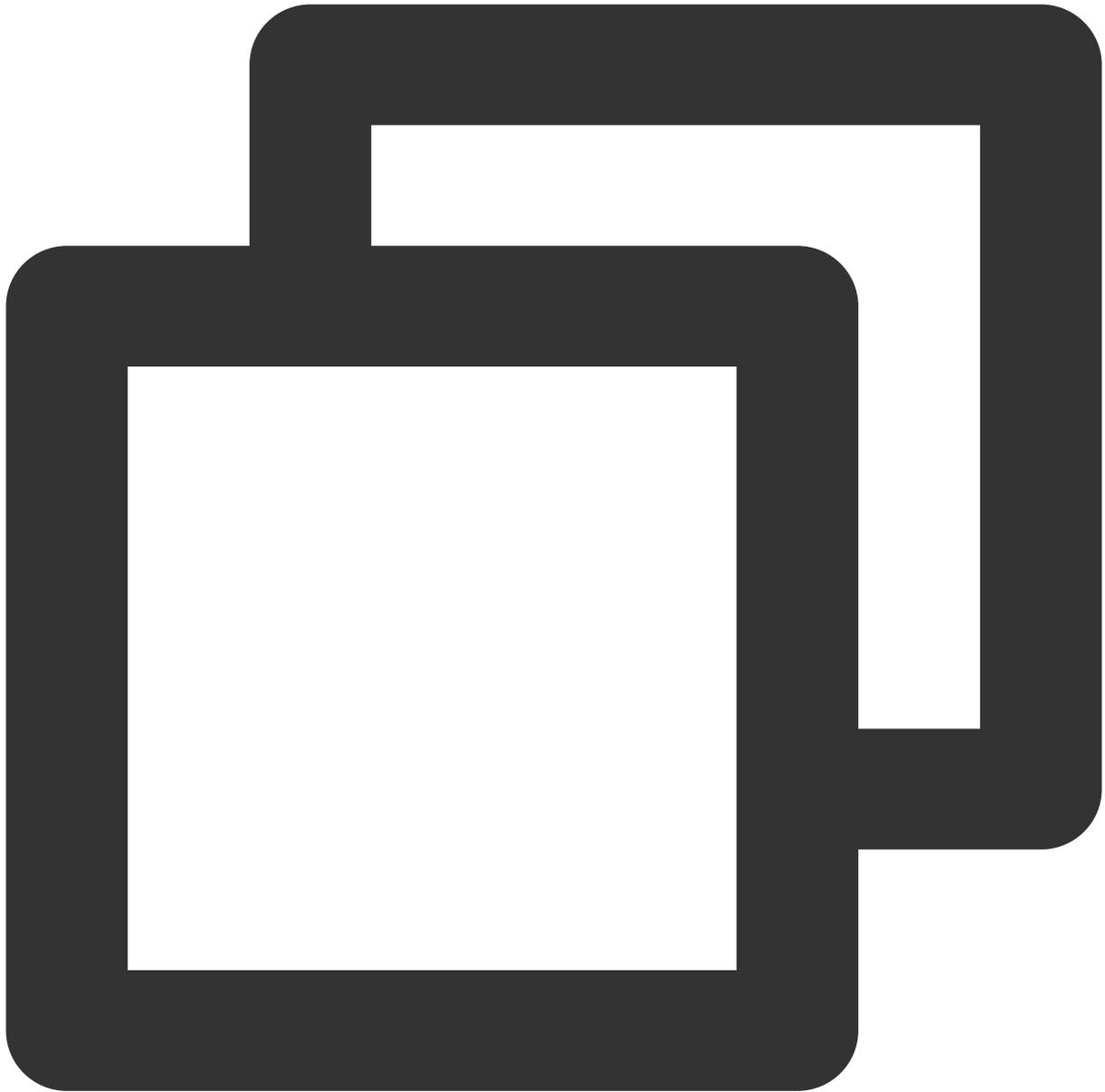
```
mv coscli-linux coscli
```

2. 运行以下命令修改文件执行权限：



```
chmod 755 coscli
```

3. 在命令行窗口中输入 `./coscli --version`，若打印出以下信息则代表安装成功：



```
coscli version v1.0.0
```

步骤三：配置COSCLI 工具

注意：

建议用户通过 [临时密钥](#) 使用工具，通过临时授权的方式进一步提高工具使用的安全性。申请临时密钥时，请遵循 [最小权限指引原则](#)，防止泄露目标存储桶或对象之外的资源。

如果您一定要使用永久密钥，建议遵循 [最小权限指引原则](#) 对永久密钥的权限范围进行限制。

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

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

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

配置文件中各配置项的说明如下：

配置项	说明
Secret ID	密钥 ID，建议使用子账号密钥，授权遵循最小权限指引，降低使用风险。子账号密钥获取可参见 子账号访问密钥管理 。
Secret Key	密钥 Key，建议使用子账号密钥，授权遵循最小权限指引，降低使用风险。子账号密钥获取可参见 子账号访问密钥管理 。
Session Token	临时密钥 token，当使用临时密钥时需要配置，若不使用可以直接按 <code>Enter</code> 跳过。关于临时密钥的更多信息，详见 使用临时密钥访问 COS 。
Mode	设置身份模式，支持枚举值 <code>SecretKey</code> 和 <code>CvmRole</code> 。可以为空，空值默认为 <code>SecretKey</code> ，表示使用密钥请求 COS。当 Mode 为 <code>CvmRole</code> 时，表示是用 管理实例角色 请求 COS。
Cvm Role Name	设置 CVM 角色实例名称，详情参考 管理实例角色 。
protocol	网络传输协议，默认为 <code>https</code> 。如需更改为 <code>http</code> ，可直接进入配置文件进行修改即可。
APPID	APPID 是您在成功申请腾讯云账户后所得到的账号，由系统自动分配，可从 账号信息 中获取。一个存储桶的全称由 <code>Bucket Name</code> 和 <code>APPID</code> 这两个元素组成，格式为 <code><BucketName-APPID></code> ，详情请参见 存储桶命名规范 。
Bucket Name	存储桶名称，和 APPID 一起构成存储桶全称，格式为 <code><BucketName-APPID></code> ，详情请参见 存储桶命名规范 。
Bucket Endpoint	存储桶所在地域的域名，默认域名格式为 <code>cos.<region>.myqcloud.com</code> ，其中 <code><region></code> 代表存储桶地域，例如 <code>ap-guangzhou</code> 、 <code>ap-beijing</code> 等，COS 支持的地域列表可参见 地域与访问域名 。 如果存储桶开启了全球加速，可配置全球加速域名。例如，全球加速域名配置为 <code>cos.accelerate.myqcloud.com</code> ；内网全球加速域名配置为 <code>cos-internal.accelerate.tencentcos.cn</code> 。

Bucket Alias	存储桶别名，配置后可以在使用时用 <code>BucketAlias</code> 代替 <code>BucketName-APPID</code> ，减少所需输入的命令长度，如果不配置此项， <code>BucketAlias</code> 的值是 <code>BucketName-APPID</code> 的值。
OFS Bucket	元数据加速桶标记，用于标识该桶是否开启 元数据加速 功能。目前元数据加速功能为公测功能，需要 联系我们 申请使用。公测期间暂不收费，您可以关注您的 站内信 或者 计费概述 ，了解最新的计费动态。
Auto Switch Host	设置是否关闭自动切换备份域名。可选值 <code>true false</code> ，可以为空。 若不设置或值为 <code>false</code> ，则会进行备份域名切换； 若设置为 <code>true</code> ，则不进行备份域名切换。

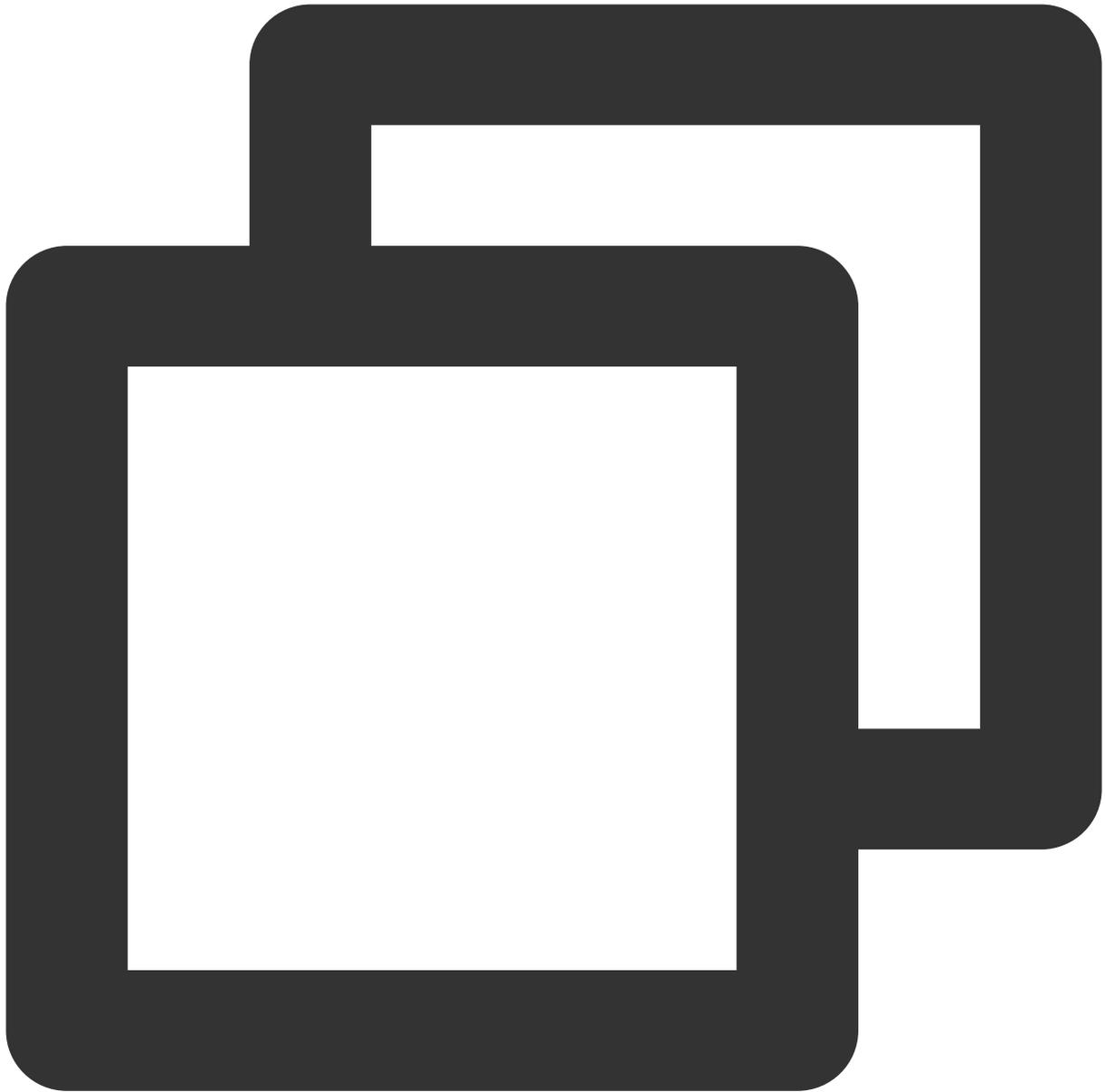
在初次配置时，COSCLI 只会要求您配置一个存储桶的信息，如果您希望配置多个存储桶，您后续可以使用 `./coscli config add` 命令添加存储桶配置。如您需要修改配置文件，或获取更多有关配置文件的操作，请参考 [config 命令](#) 或使用 `./coscli config --help` 命令来快速查看和配置文件相关的指令。

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

其他配置方法

除了使用 `./coscli config init` 交互式地生成配置文件外，您也可以直接手动编写 COSCLI 的配置文件。

COSCLI 的配置文件格式是 `yaml` 格式，配置文件示例如下：



```
cos:  
  base:  
    secretid: XXXXXXXXXXXXXXXXX  
    secretkey: XXXXXXXXXXXXXXXXX  
    sessiontoken: ""  
    protocol: https  
  buckets:  
  - name: examplebucket1-1250000000  
    alias: bucket1  
    region: ap-shanghai  
    endpoint: cos.ap-shanghai.myqcloud.com
```

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

注意：

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

通用选项

最近更新时间：2024-07-11 14:12:41

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

选项说明

以下为 COSCLI 的通用选项，这些选项可以在工具的所有命令中使用：

注意

建议用户通过 [临时密钥](#) 使用工具，通过临时授权的方式进一步提高工具使用的安全性。申请临时密钥时，请遵循 [最小权限指引原则](#)，防止泄漏目标存储桶或对象之外的资源。

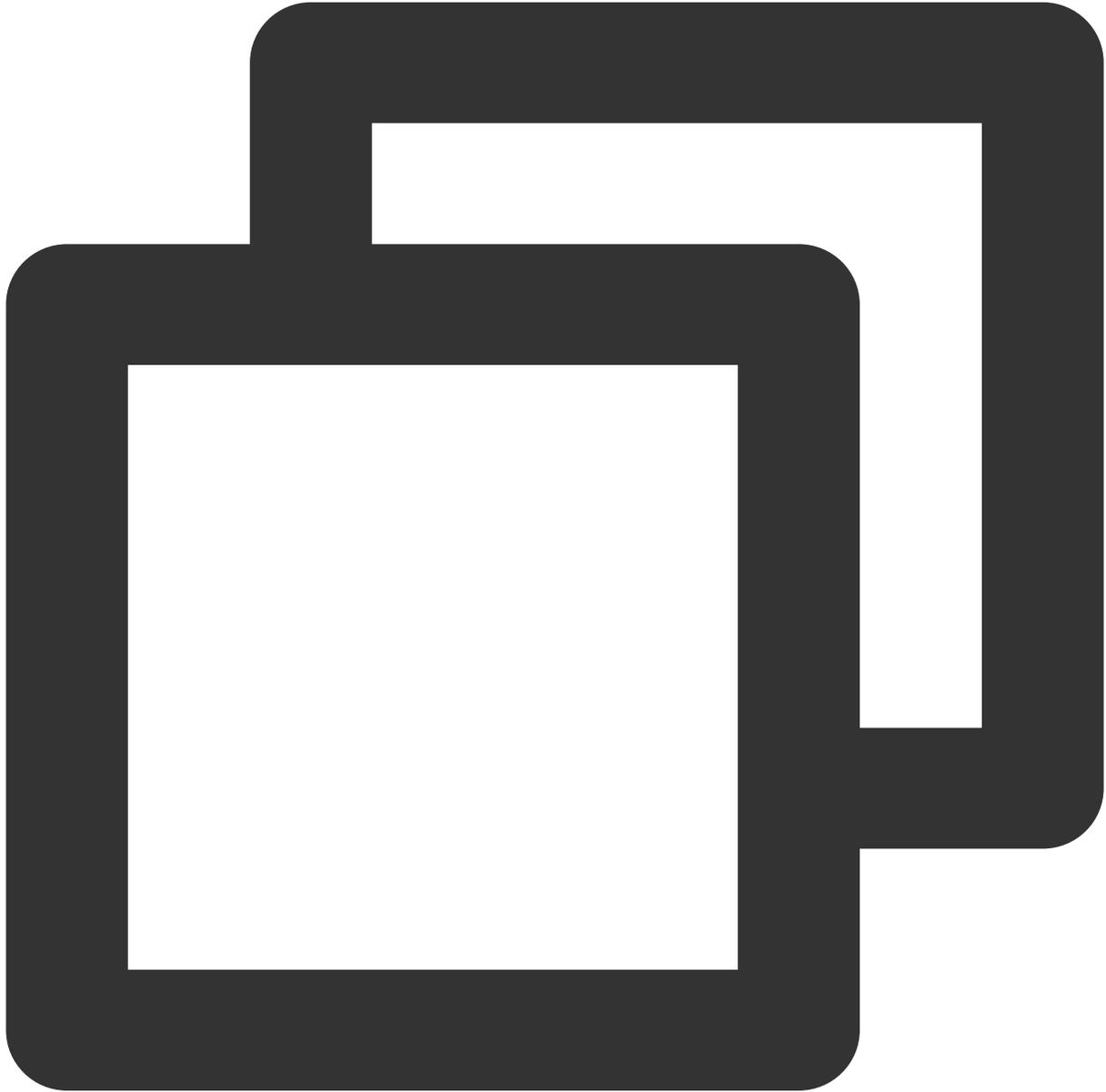
如果您一定要使用永久密钥，建议遵循 [最小权限指引原则](#) 对永久密钥的权限范围进行限制。

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

操作示例

示例1：切换存储桶并上传对象

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

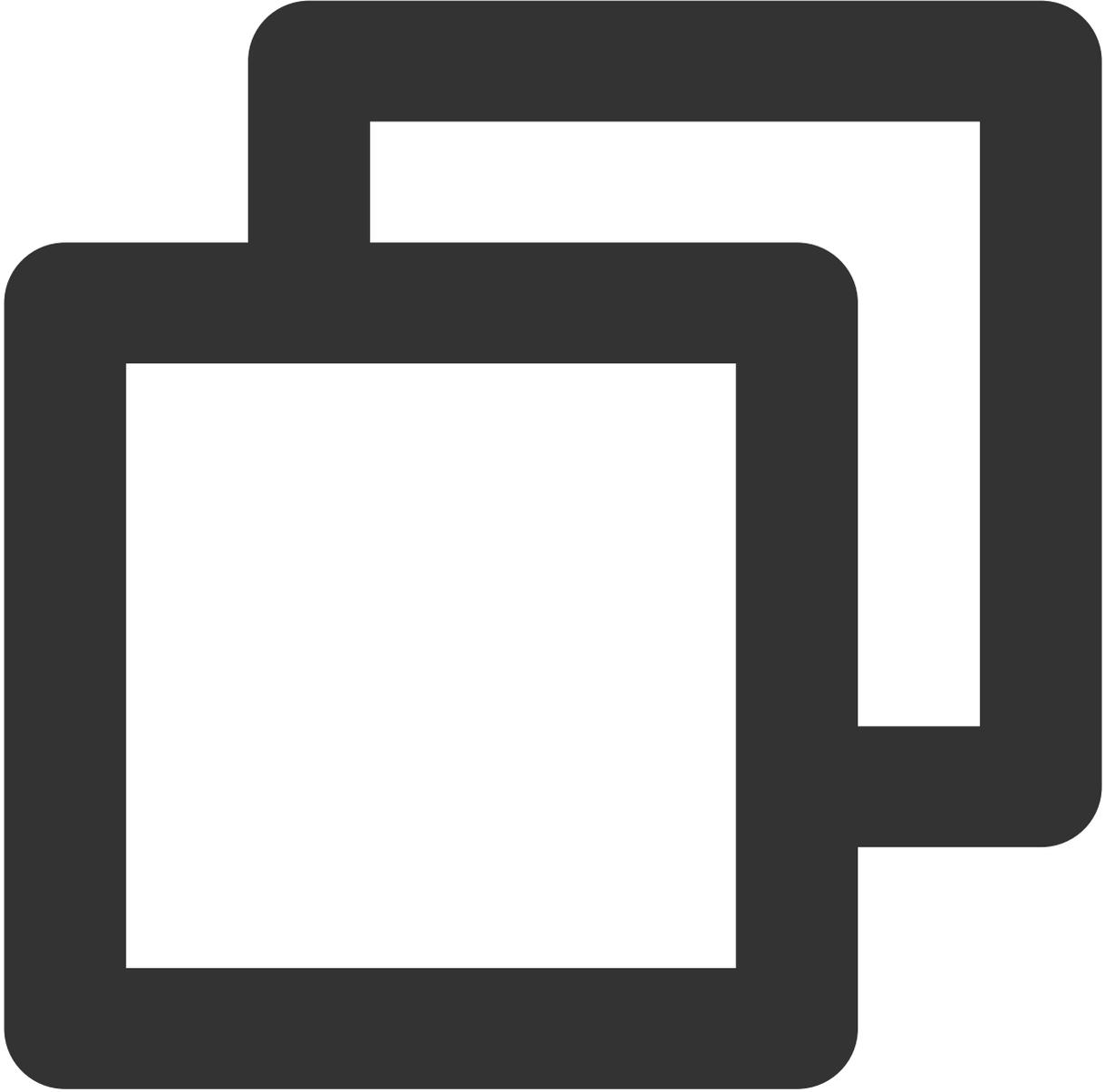


```
./coscli cp test.txt cos://examplebucket-1250000000/test.txt -e cos.ap-chengdu.myqc
```

示例2：切换用户账号查看文件列表

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

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



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

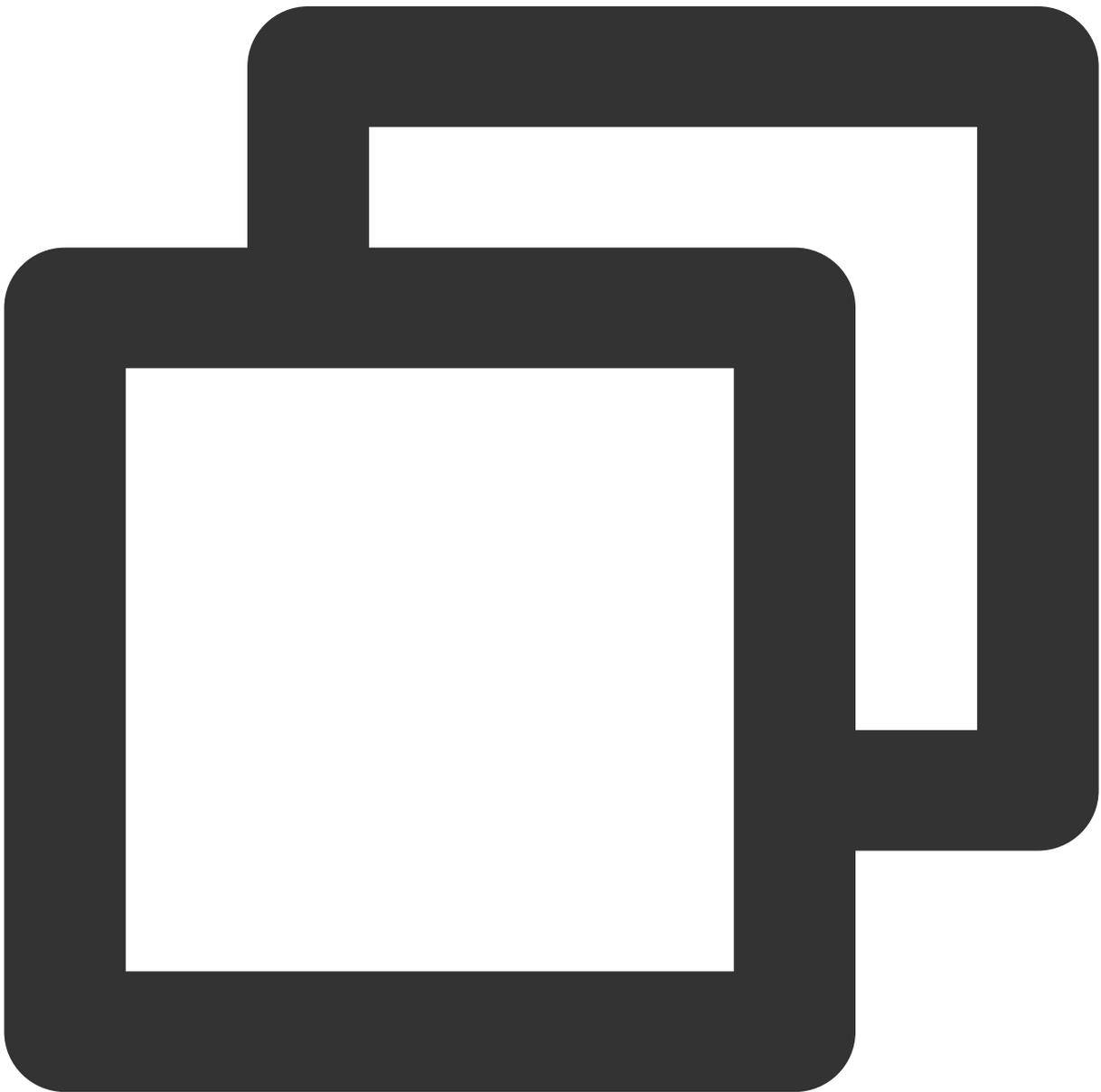
常用命令

生成与修改配置文件 - config

最近更新时间：2024-06-20 14:11:15

命令格式

以下 config 命令用于生成与修改配置文件：



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

说明：

正确填写各配置项后，您可以使用 `./coscli config show` 来查看配置信息。

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

生成的配置文件，默认网络传输协议为 `https`。如需更改为 `http`，可直接进入配置文件进行修改即可。

`config` 命令包含以下子命令：

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

`config` 及其子命令包含以下可选 `flag`：

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

`config add` 子命令包含以下可选 `flag`：

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

注意：

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

`config delete` 子命令包含以下可选 flag：

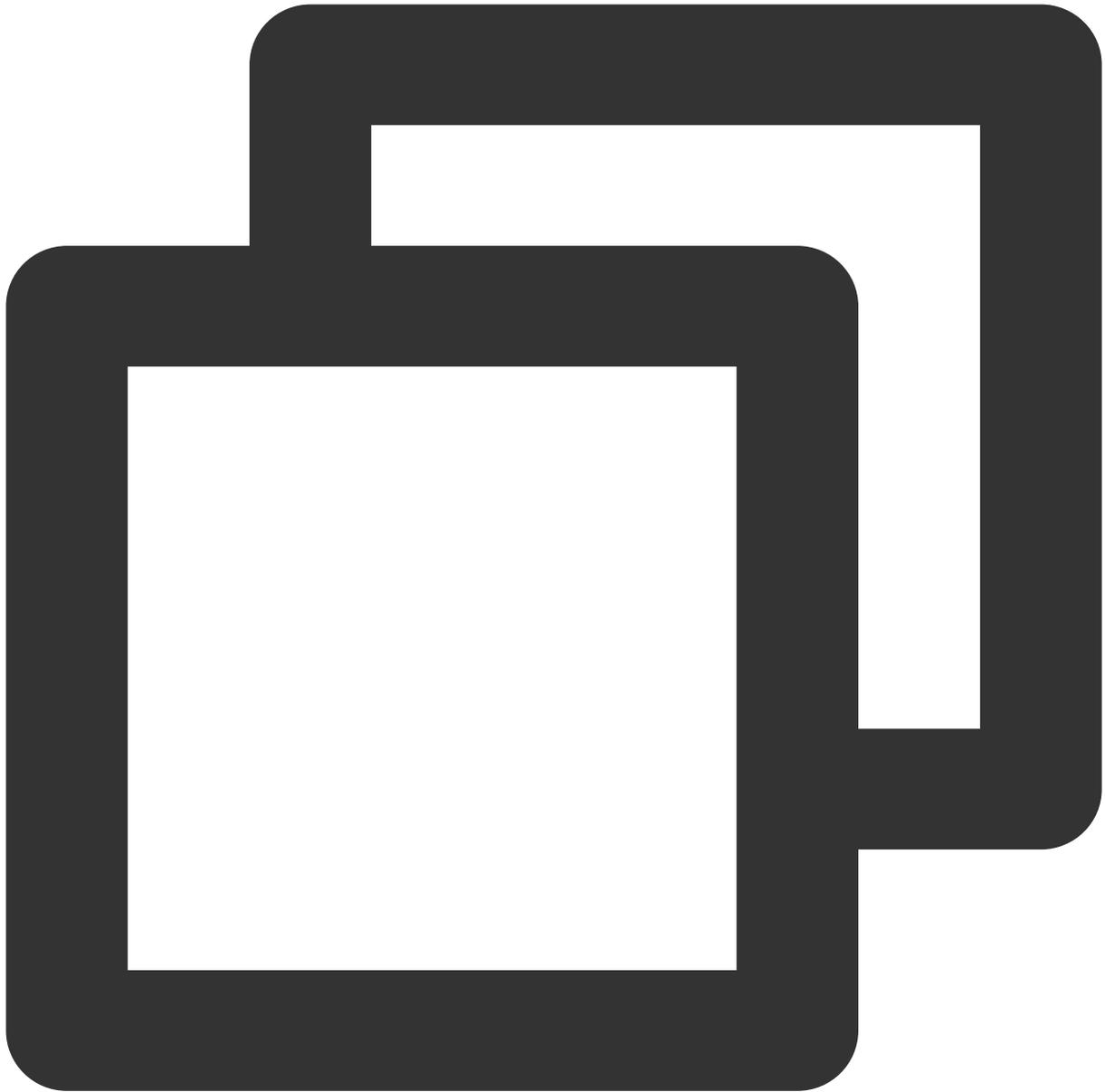
flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
-a	--alias	存储桶别名。

`config set` 子命令包含以下可选 flag：

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

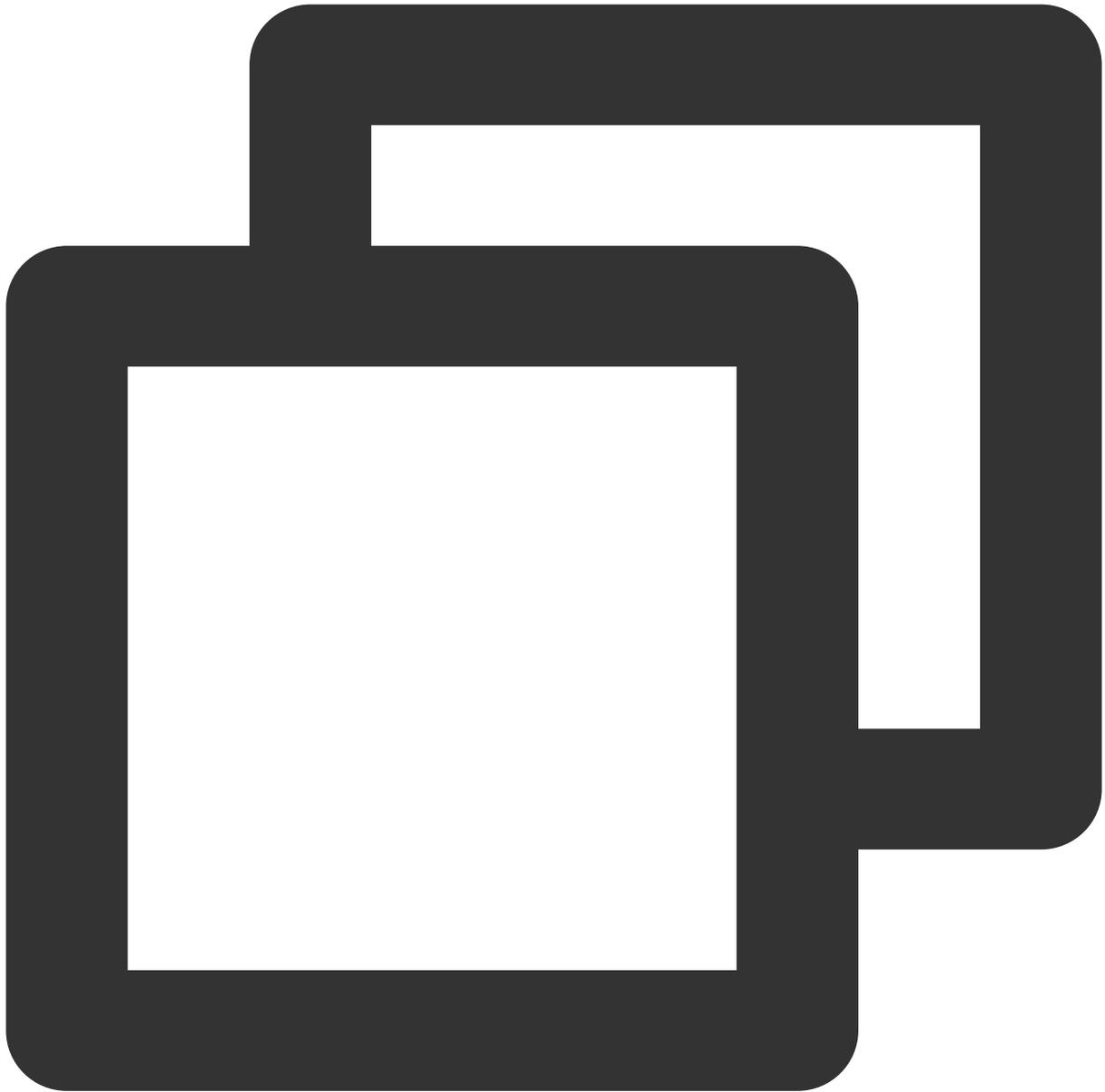
操作示例

添加一个新的存储桶配置



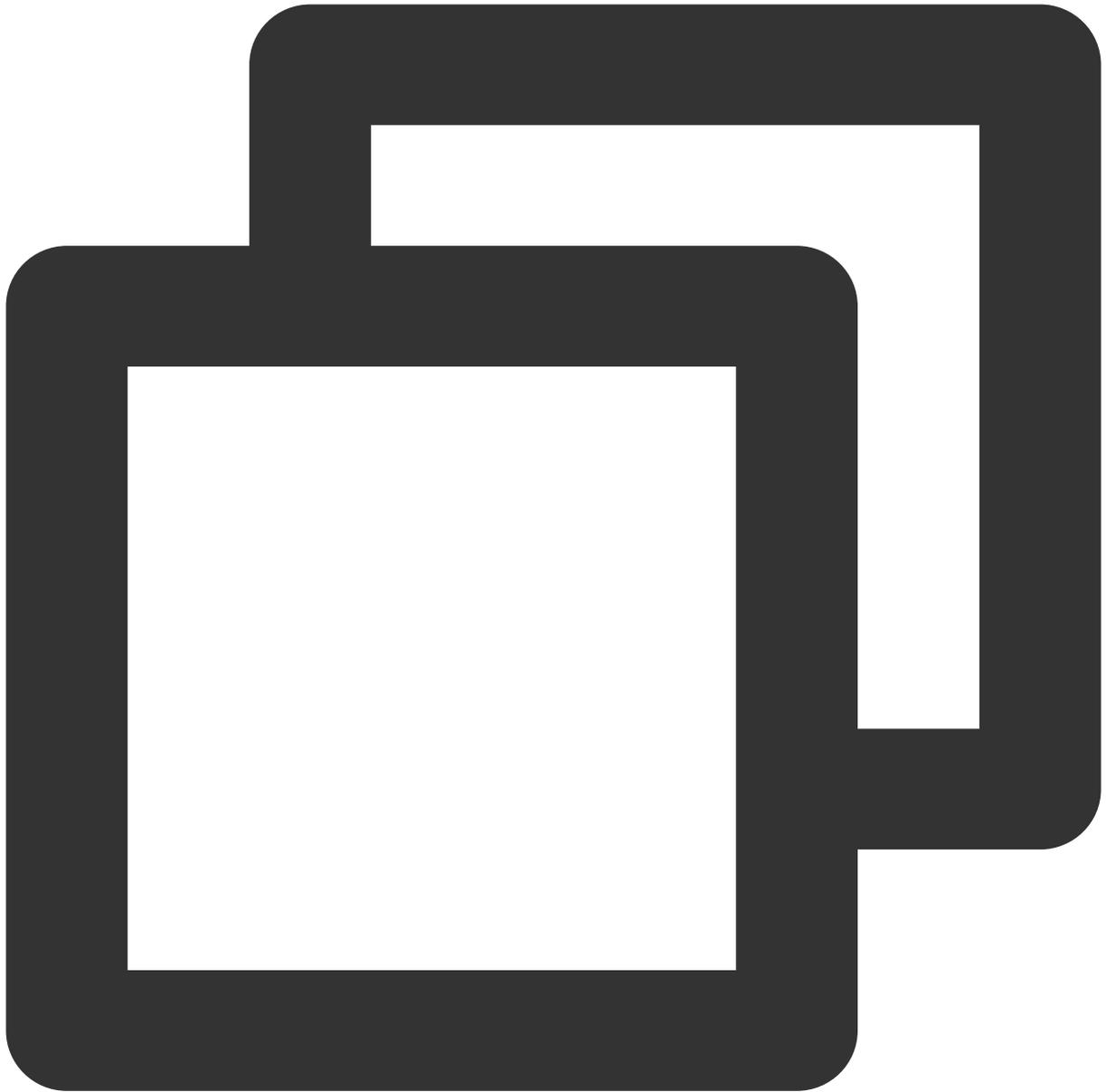
```
./coscli config add -b examplebucket3-1250000000 -r ap-chengdu -e cos.ap-chengdu.my
```

删除一个已经存在的存储桶配置



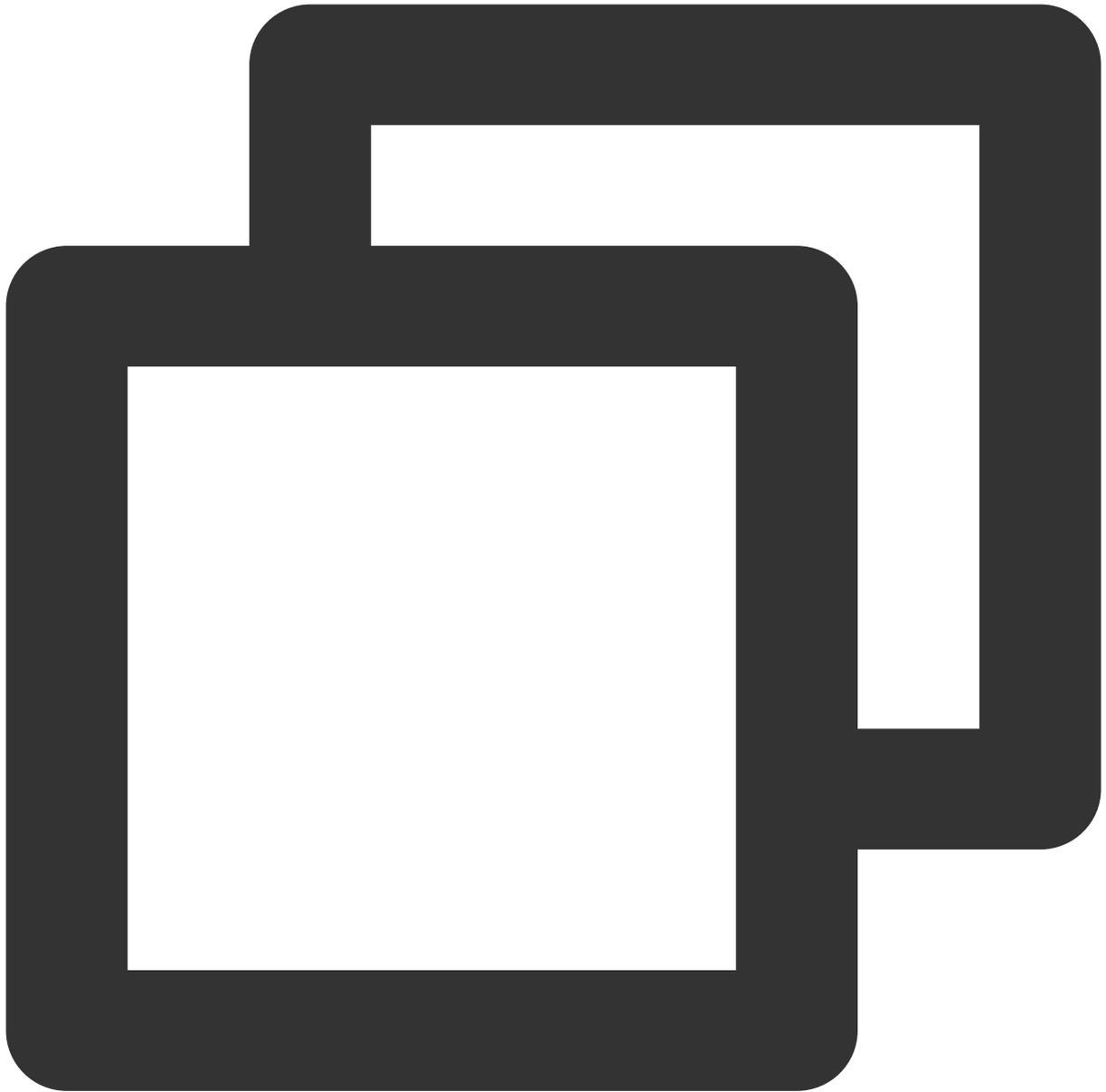
```
./coscli config delete -a bucket3
```

修改默认配置文件中的 **session-token**



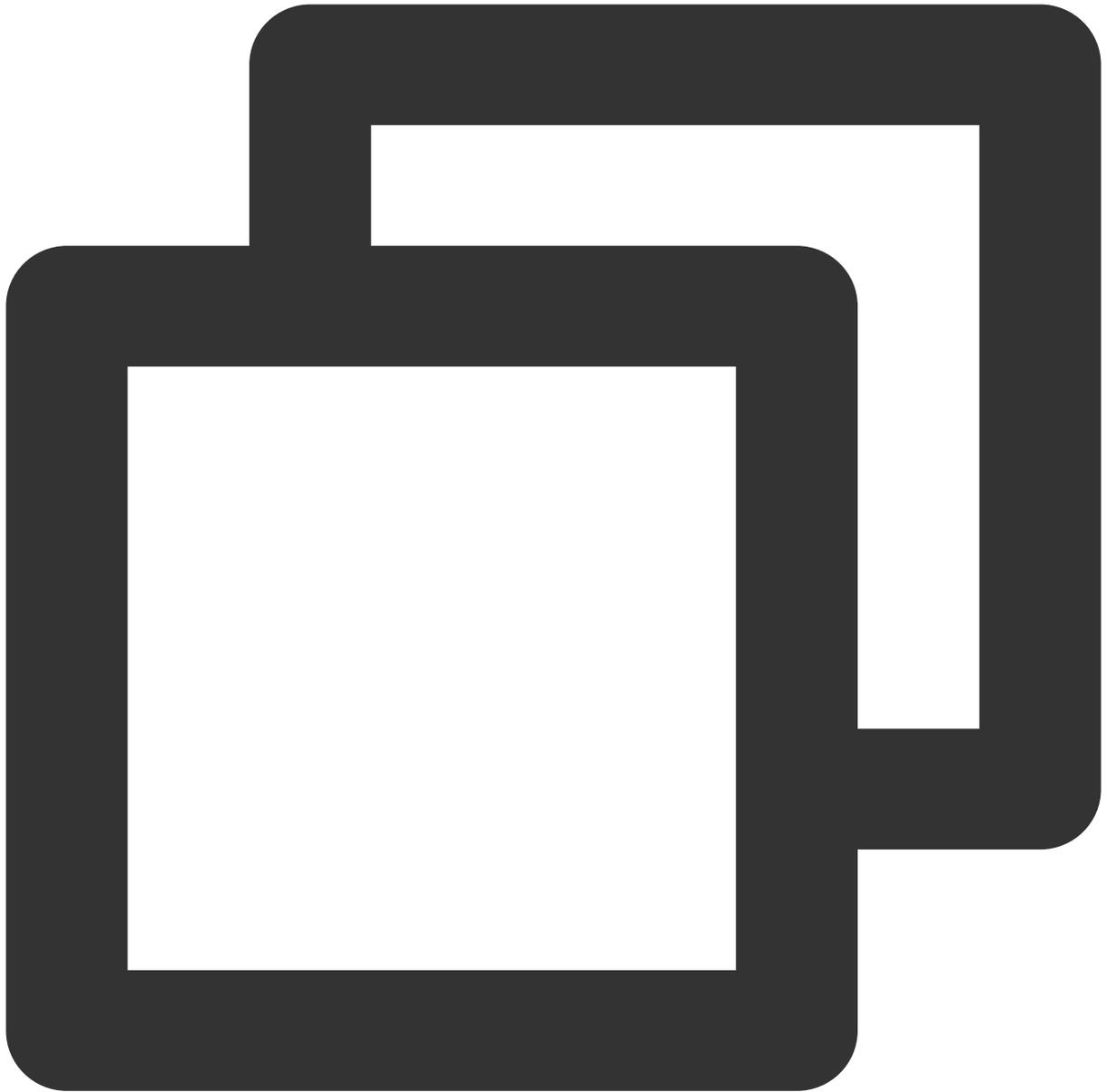
```
./coscli config set --session_token test-token123
```

打印指定配置文件中的信息



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

修改默认配置文件中的 **mode** 和 **cvmrolename**



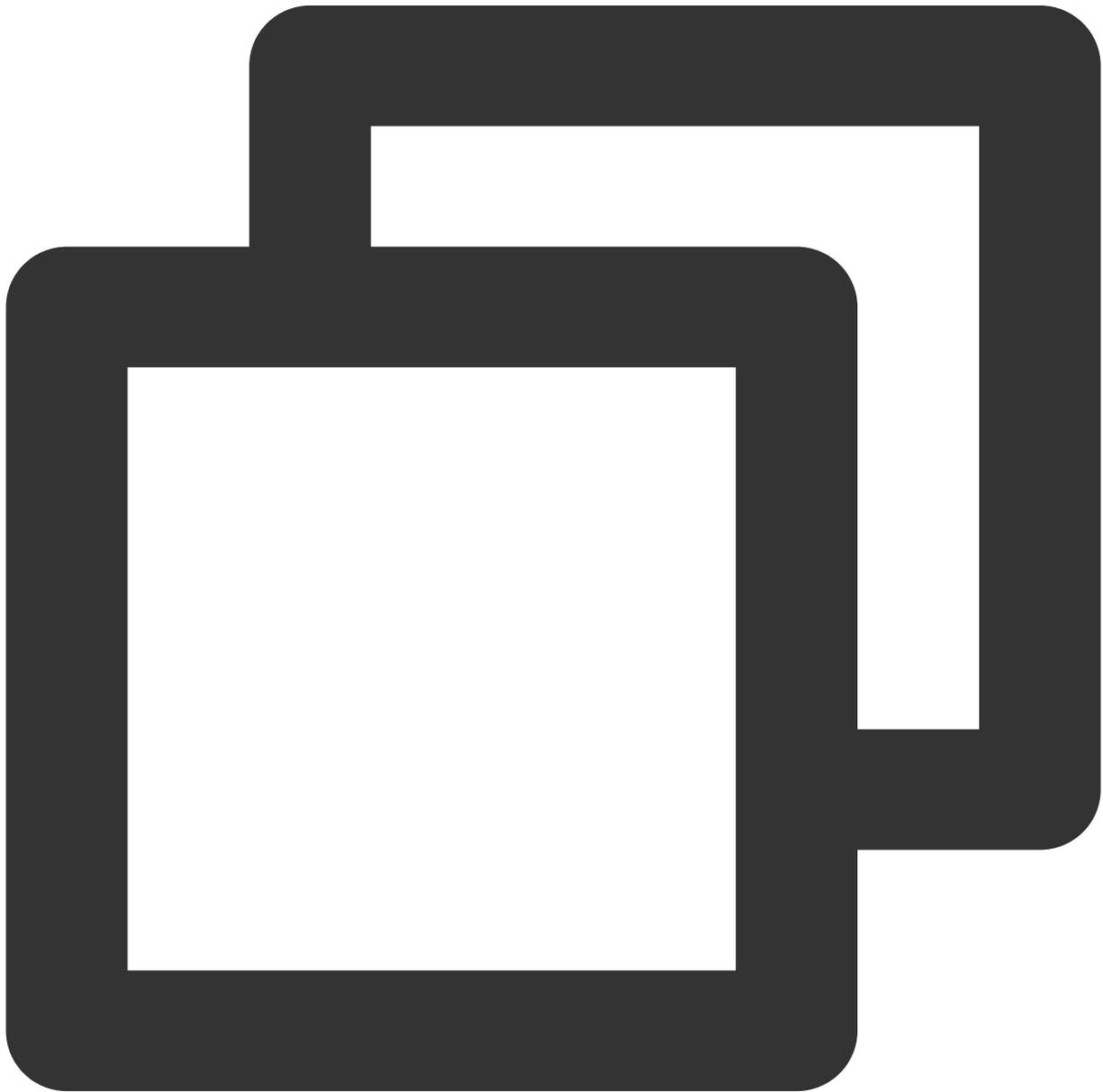
```
./coscli config set --mode CvmRole --cvm_role_name testName
```

创建存储桶 - mb

最近更新时间：2024-01-06 15:54:24

mb 命令用于创建存储桶。

命令格式



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

注意

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

`mb` 命令包含以下参数：

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

`mb` 命令包含以下可选 flag：

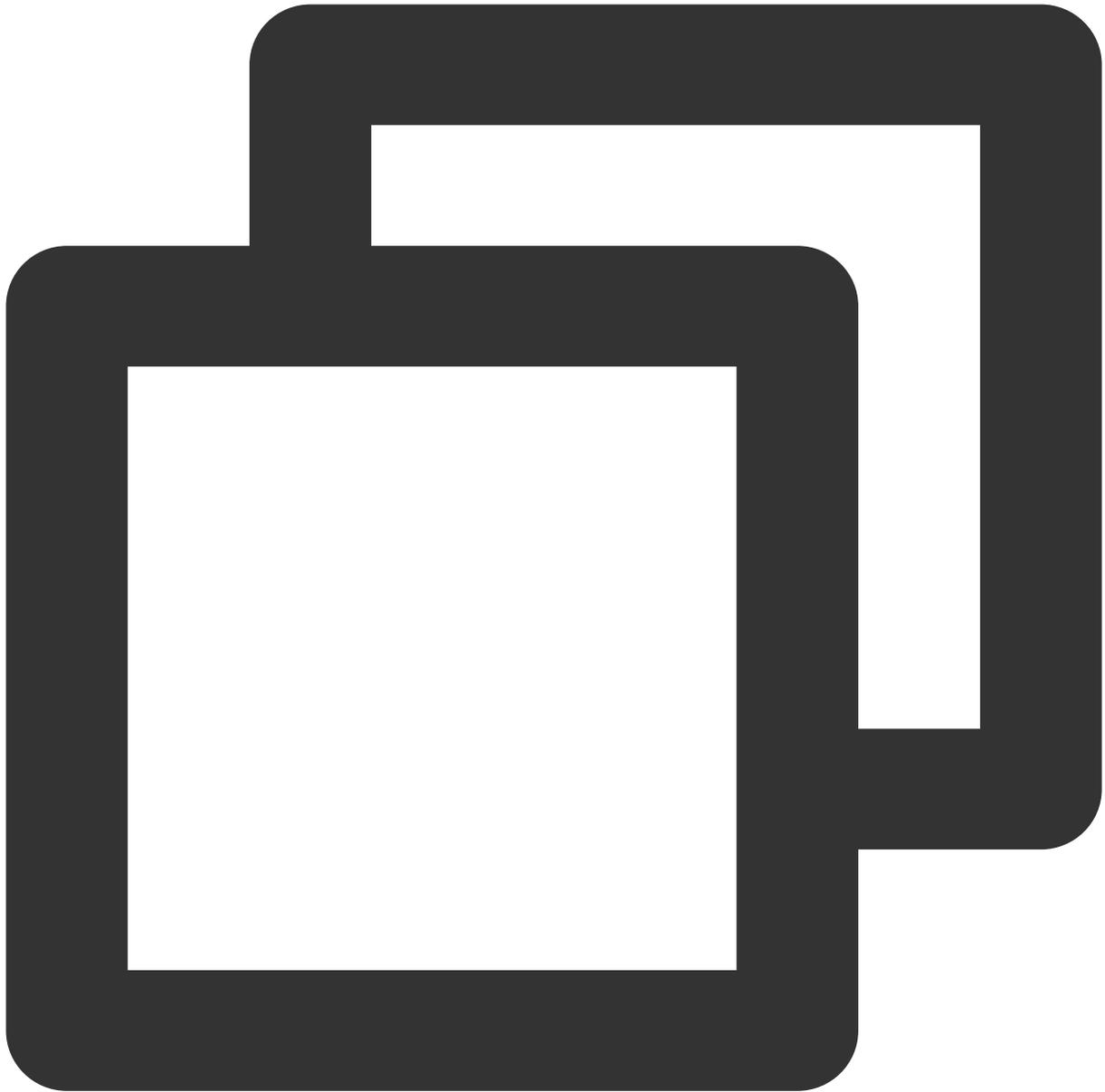
flag 简写	flag 全称	flag 用途
<code>-h</code>	<code>--help</code>	查看该命令的具体用法
<code>-r</code>	<code>--region</code>	存储桶地域

注意

使用 `mb` 命令成功创建存储桶后，推荐您在配置文件中新增该桶的信息，方便后续使用桶别名快速操作。具体命令用法，可参见下方操作示例。

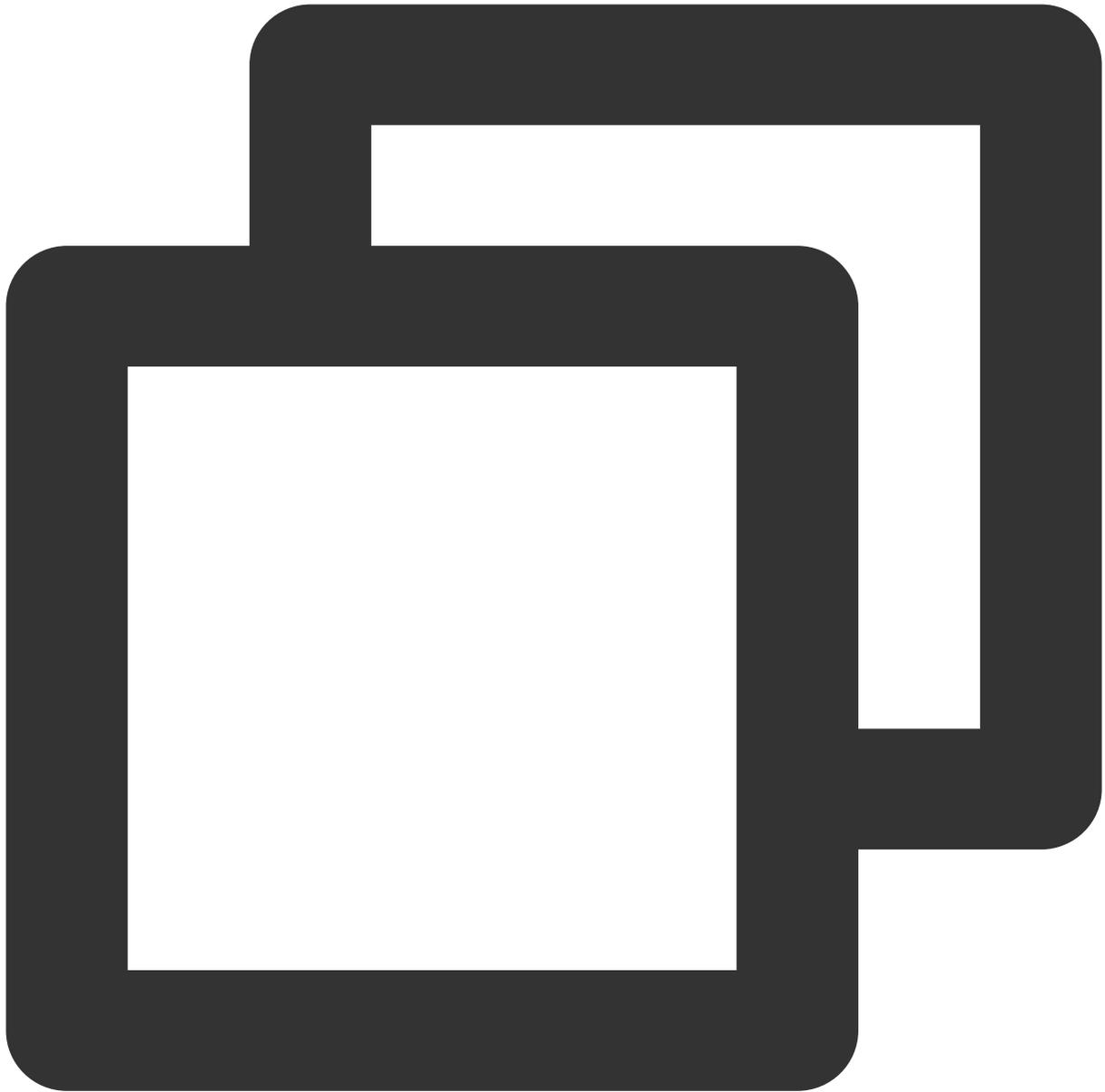
关于此命令的其他通用选项（例如切换存储桶、切换用户账号等），请参见 [通用选项](#) 文档。

操作示例



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

如果您希望为刚创建的存储桶配置别名，需要使用以下命令更新配置文件：



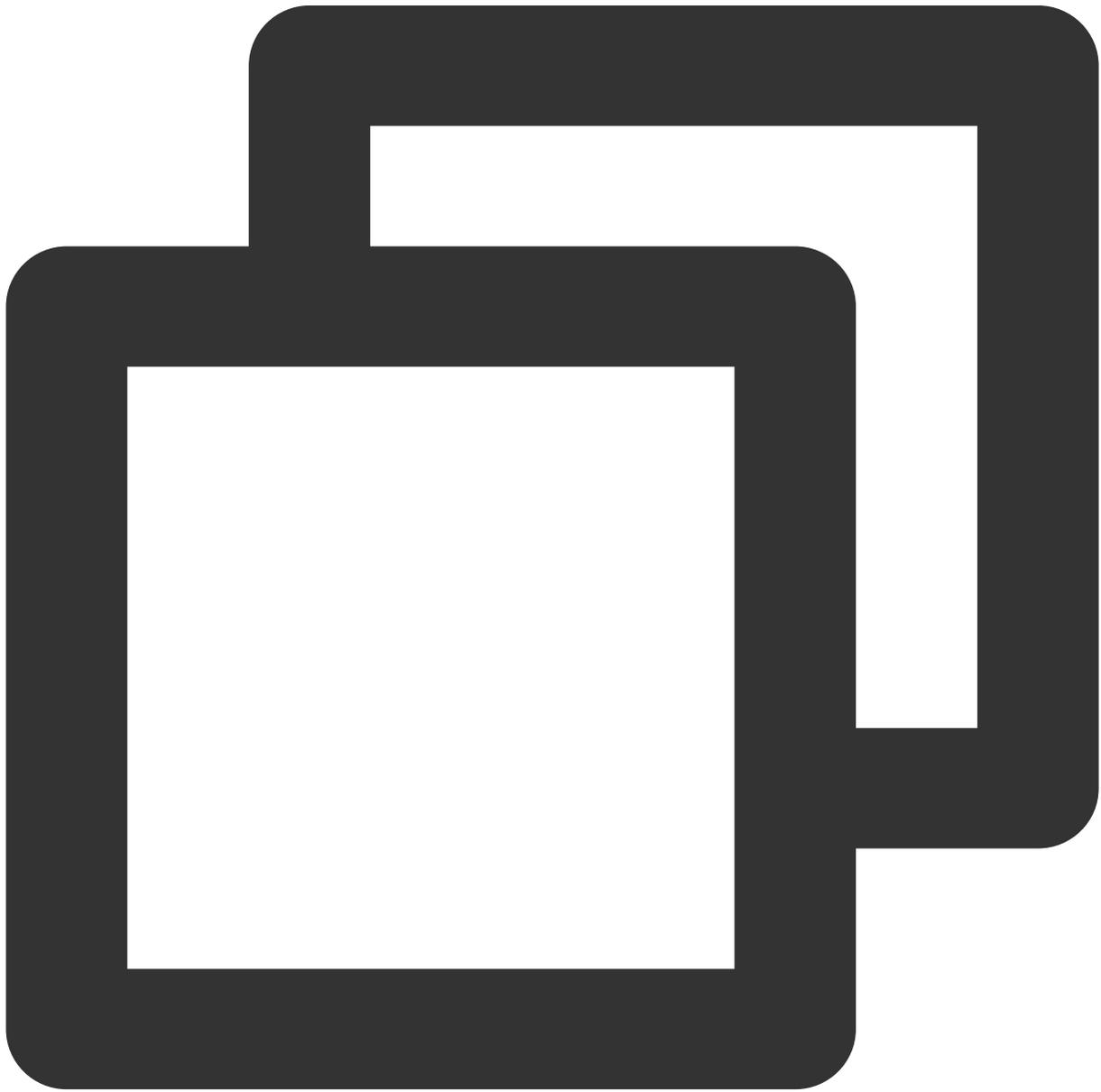
```
// 更新配置文件  
./coscli config add -b bucket3-1250000000 -e cos.ap-chengdu.myqcloud.com -a bucket3  
// 更新后可用 cos://bucket3 访问此存储桶
```

删除存储桶 - rb

最近更新时间：2024-01-06 15:54:24

rb 命令用于删除存储桶。

命令格式



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

rb 命令包含以下参数：

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

rb 命令包含以下可选 flag：

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

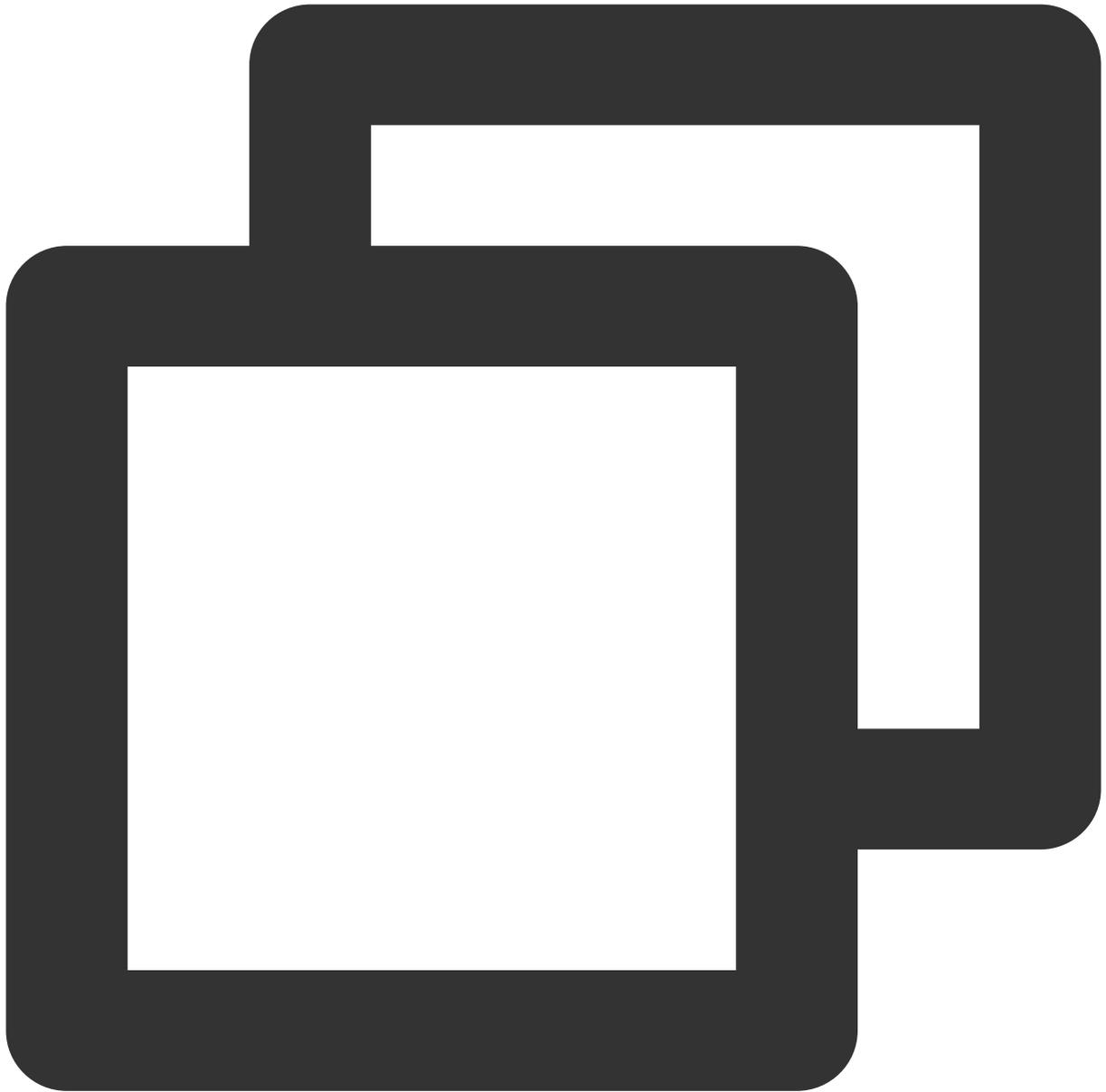
注意

rb 命令仅可删除空存储桶，如果您的桶中还有文件请使用 [删除文件](#) 命令和 [清理碎片](#) 命令分别清空存储桶中的文件和文件碎片后再删除存储桶。

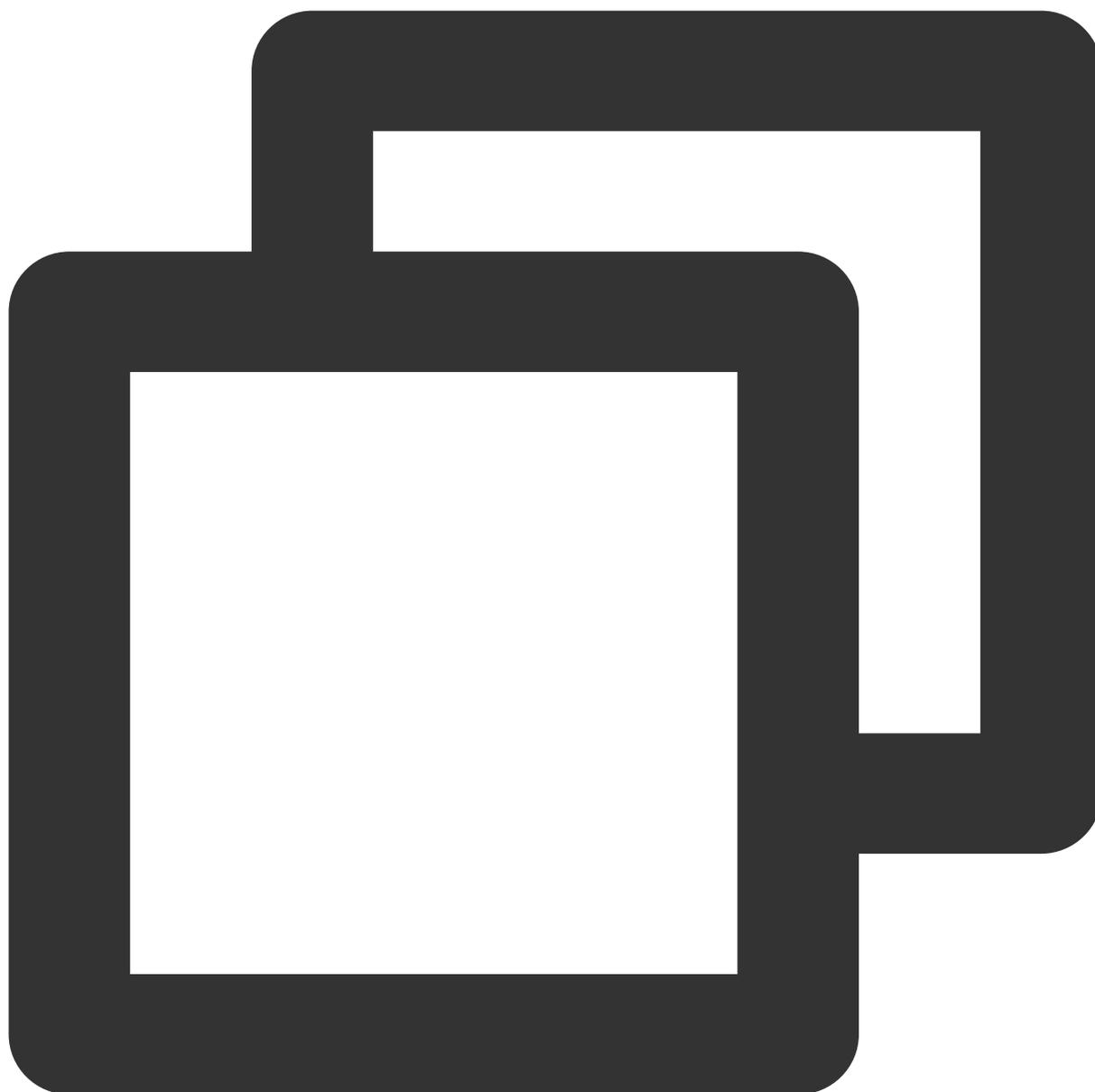
使用 rb 命令成功删除存储桶后，推荐您在配置文件中删除该桶的信息。具体命令用法，可参见下方操作示例。

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

操作示例



```
// 删除存储桶 bucket3  
./coscli rb cos://bucket3-1250000000 -e cos.ap-chengdu.myqcloud.com
```



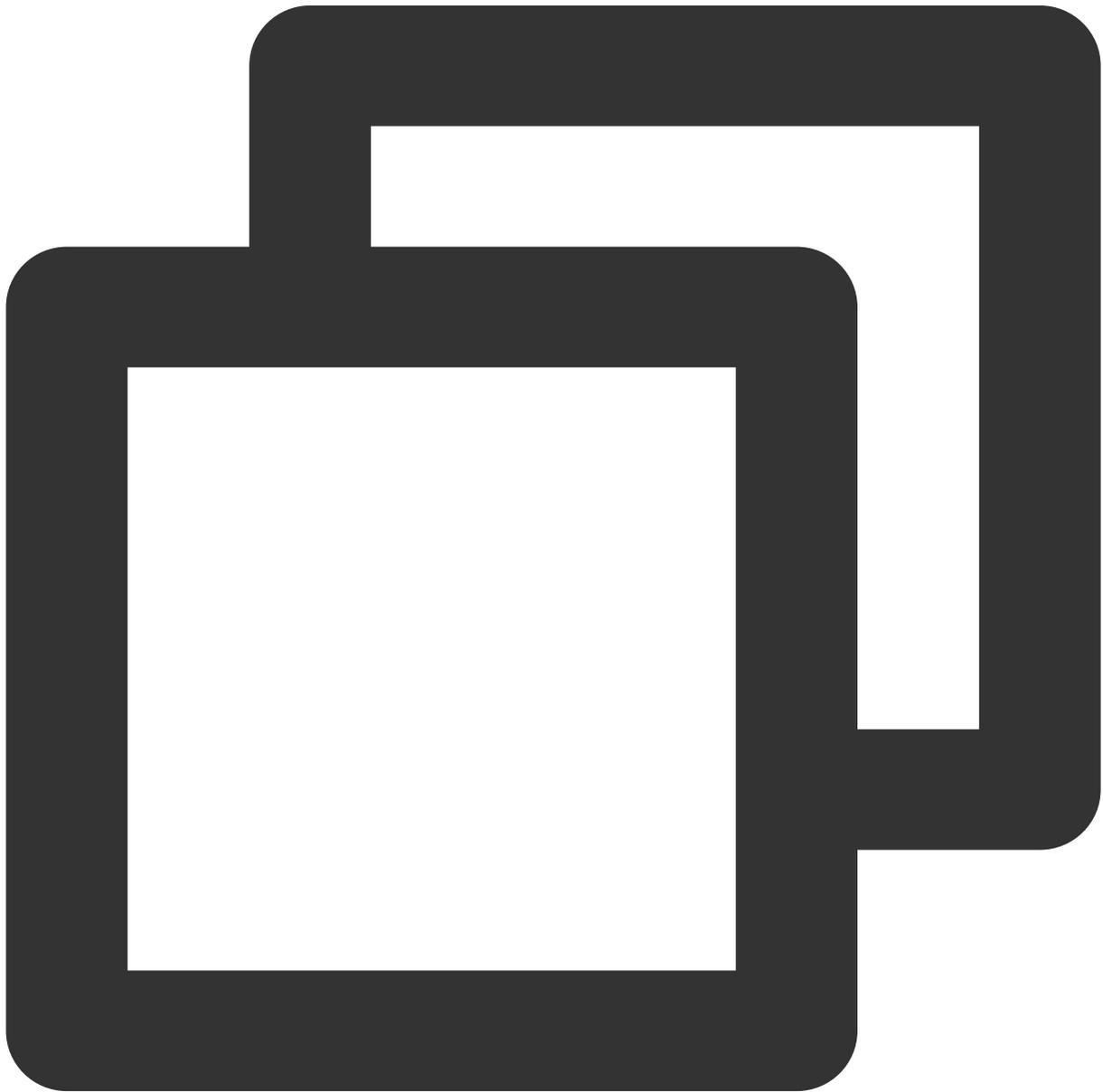
```
// 更新配置文件  
./coscli config delete -a bucket3
```

存储桶标签 - bucket-tagging

最近更新时间：2024-01-06 15:54:24

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

命令格式



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

bucket-tagging 命令包含以下参数：

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

bucket-tagging 命令包含以下可选 flag：

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

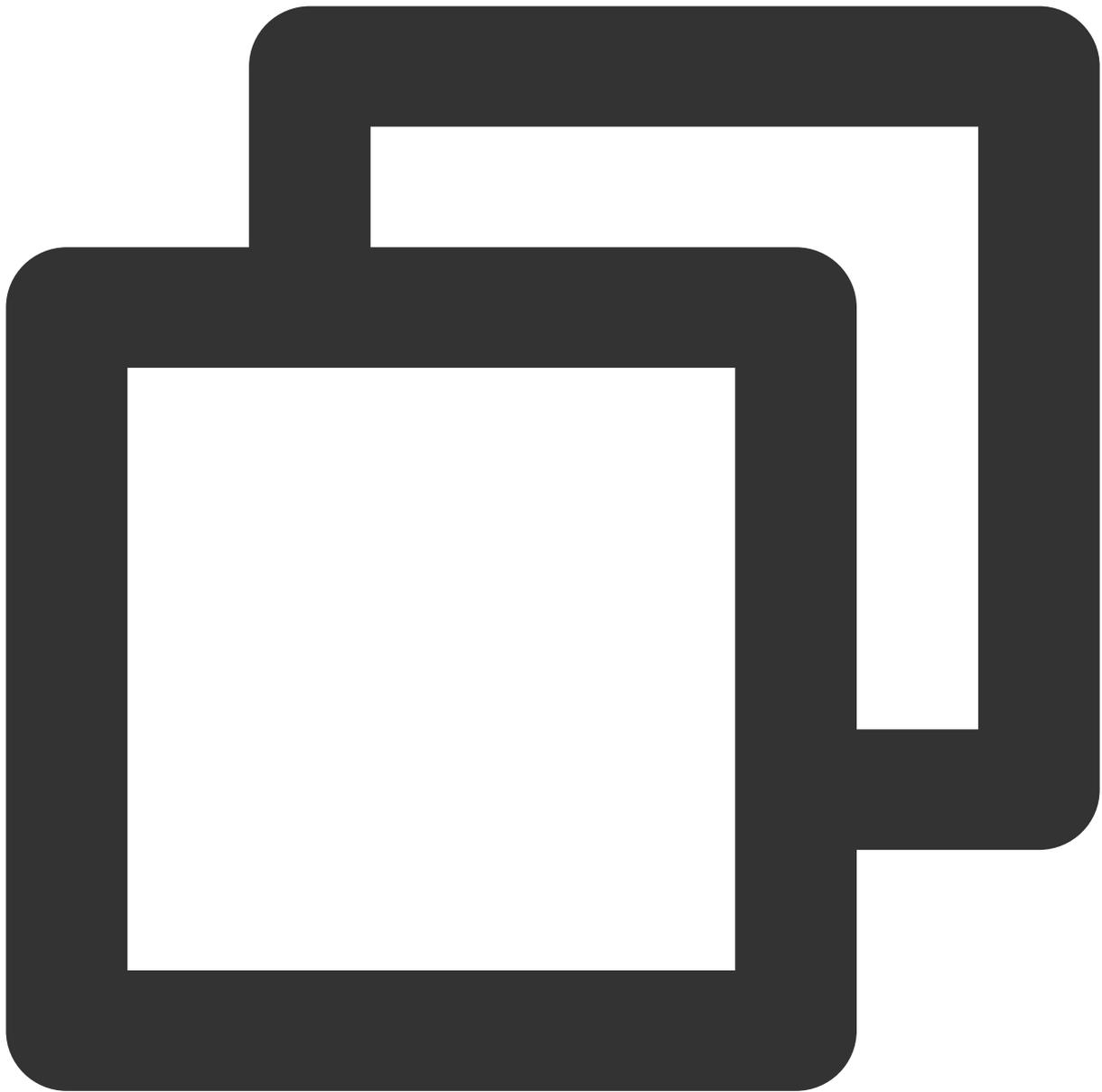
说明

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

添加或修改存储桶标签

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

命令格式

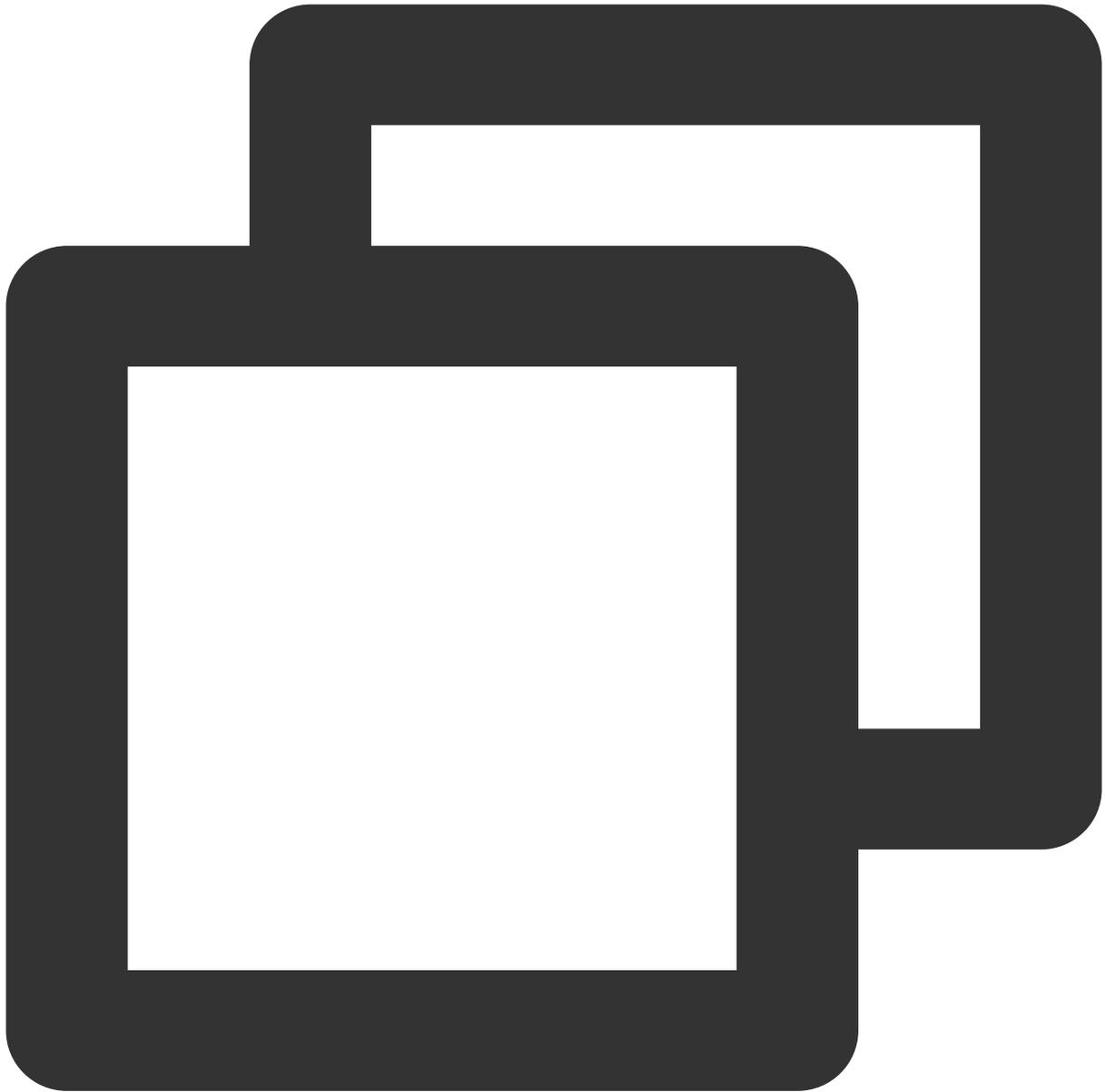


```
./coscli bucket-tagging --method put cos://bucketAlias key1#value1 key2#value2
```

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

操作示例

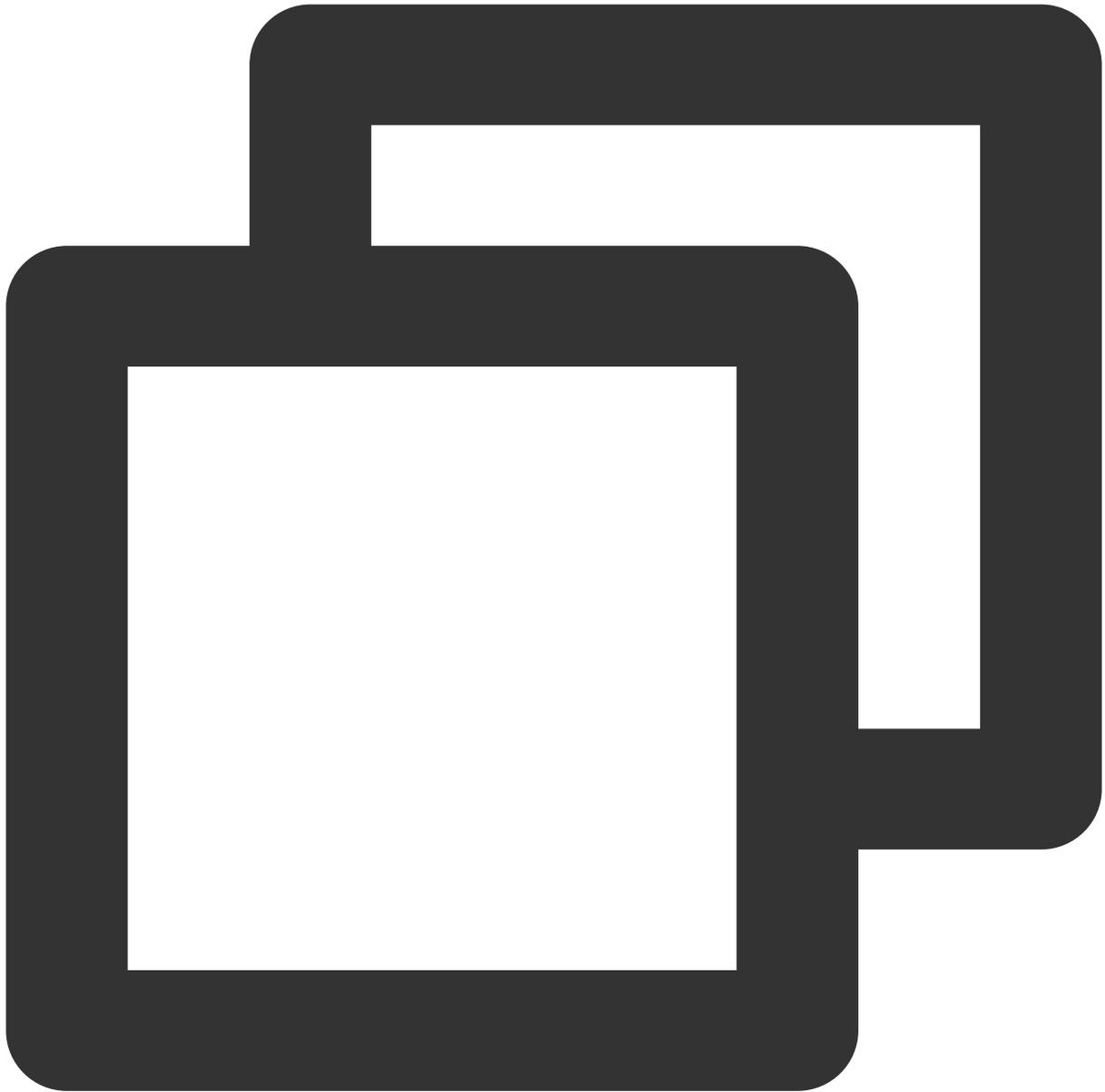
为桶别名为 `example-alias` 的存储桶配置两组标签，其中一组标签的 `key` 为1，`value` 为111，一组标签的 `key` 为2，`value` 为222。命令如下：



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

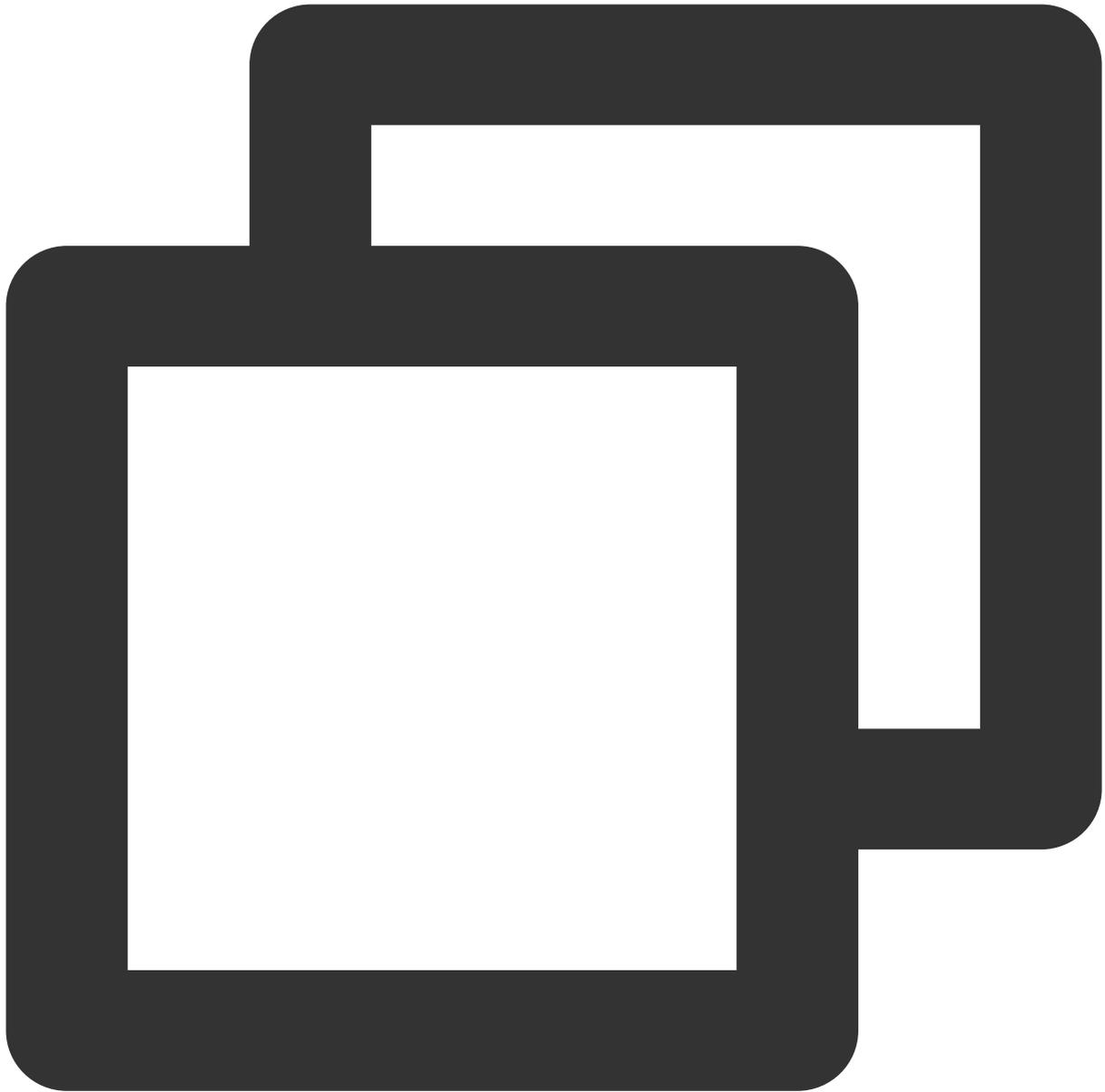
查询存储桶标签

命令格式



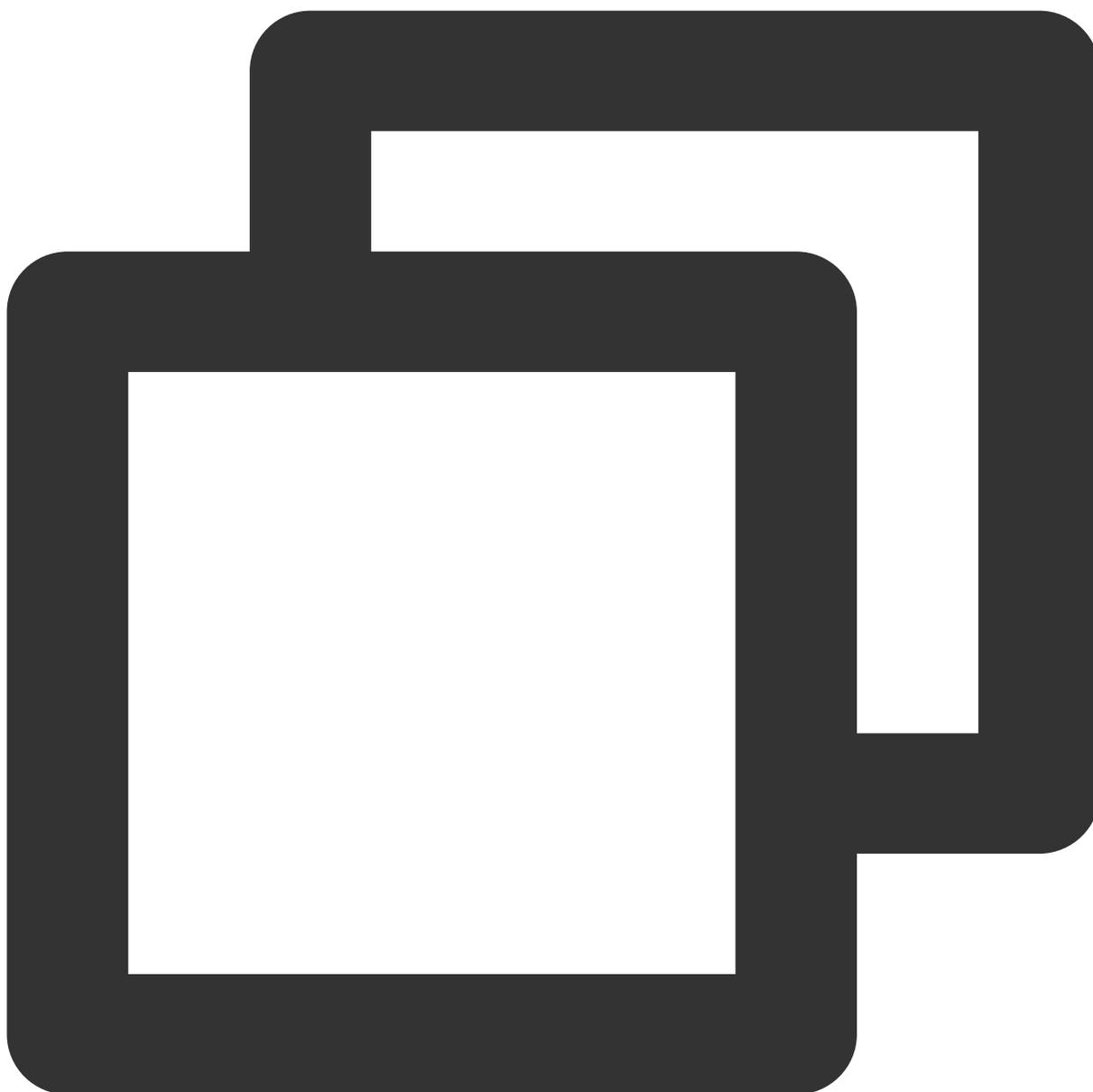
```
./coscli bucket-tagging --method get cos://bucketAlias
```

操作示例



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

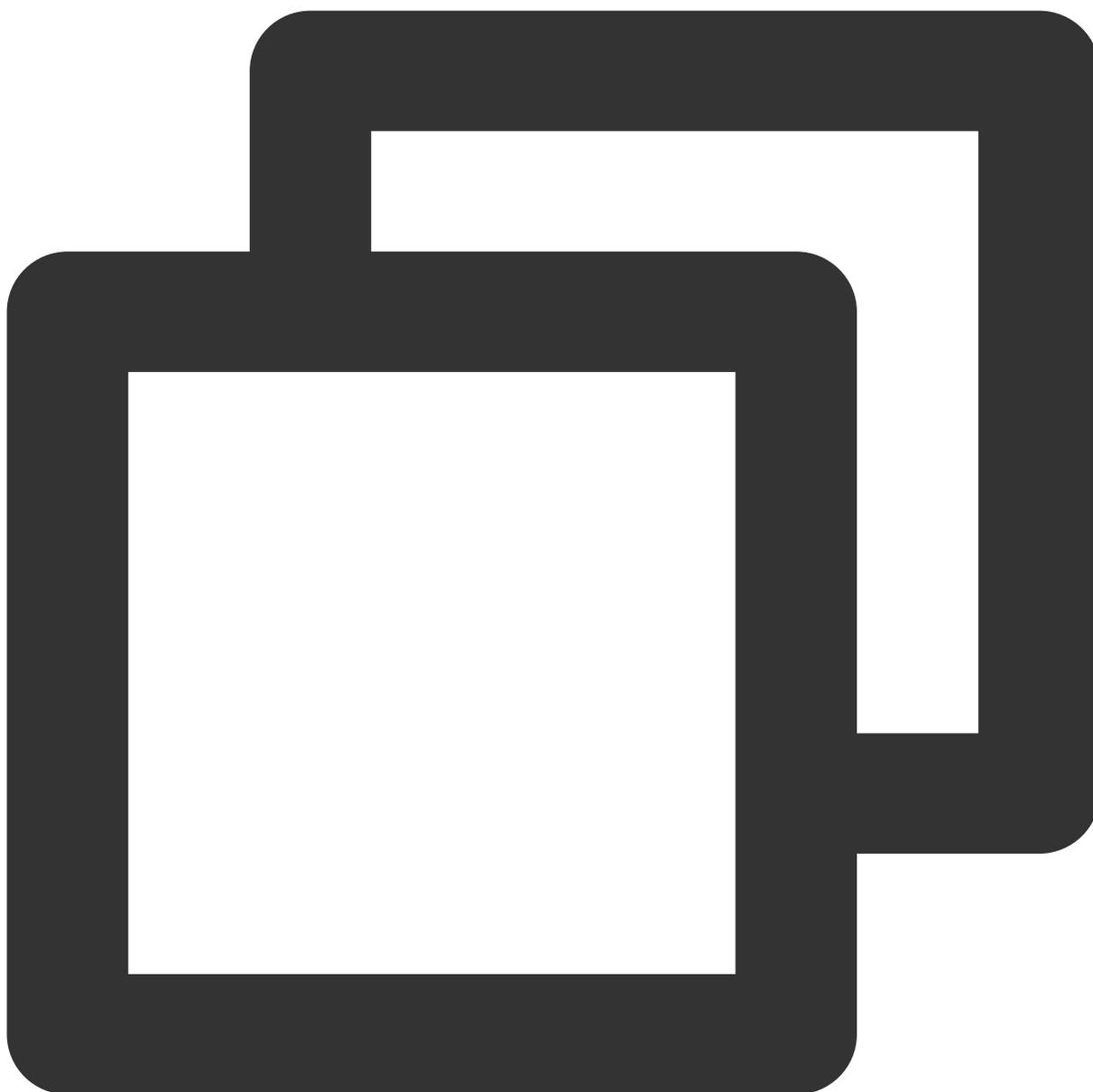
以下输出结果表明桶别名为 `exmaple-alias` 的存储桶配置了两组标签，其中一组标签的 `key` 为1，`value` 为111，一组标签的 `key` 为2，`value` 为222。



KEY	VALUE
1	111
2	222

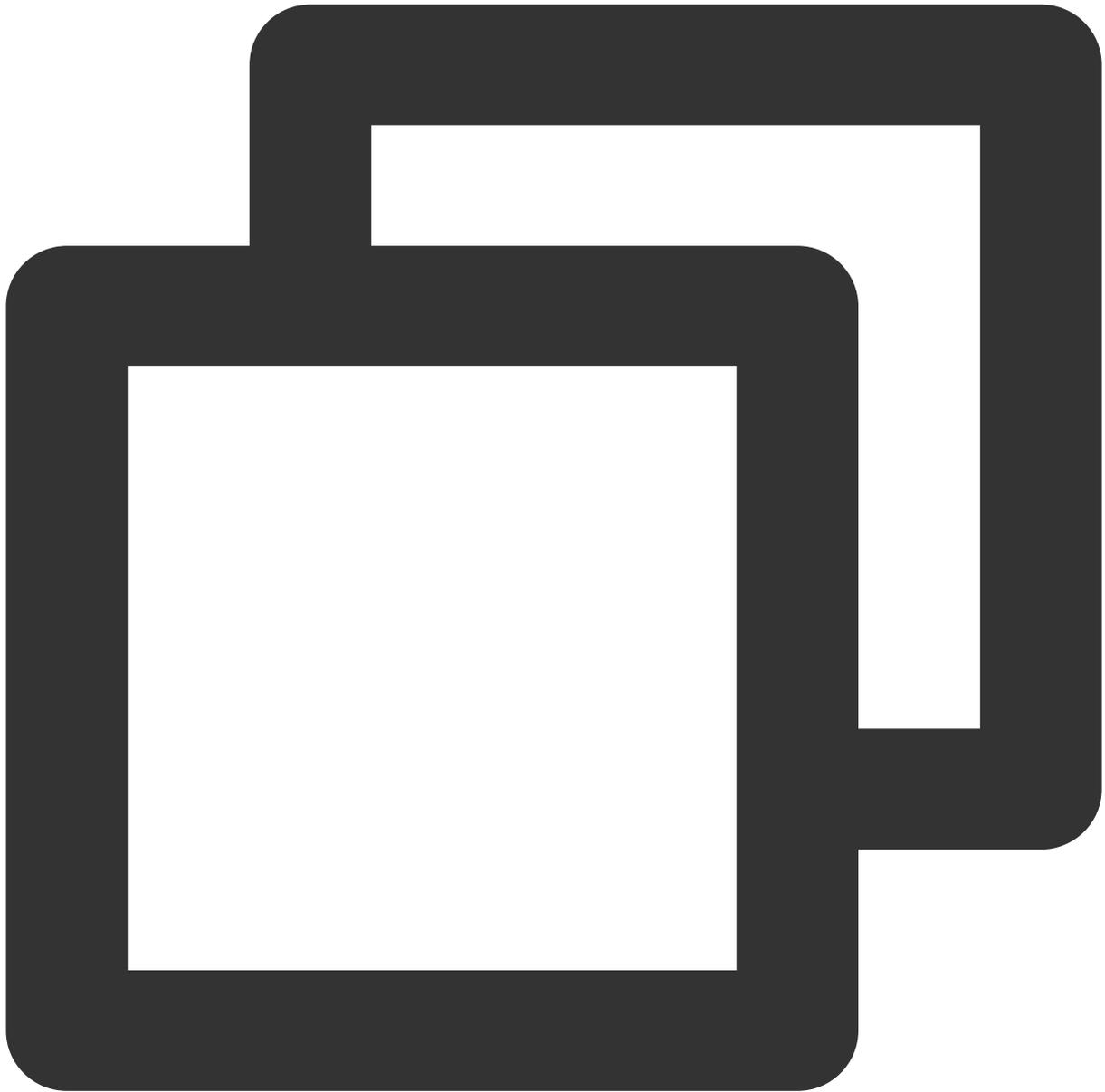
删除存储桶标签

命令格式



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

操作示例



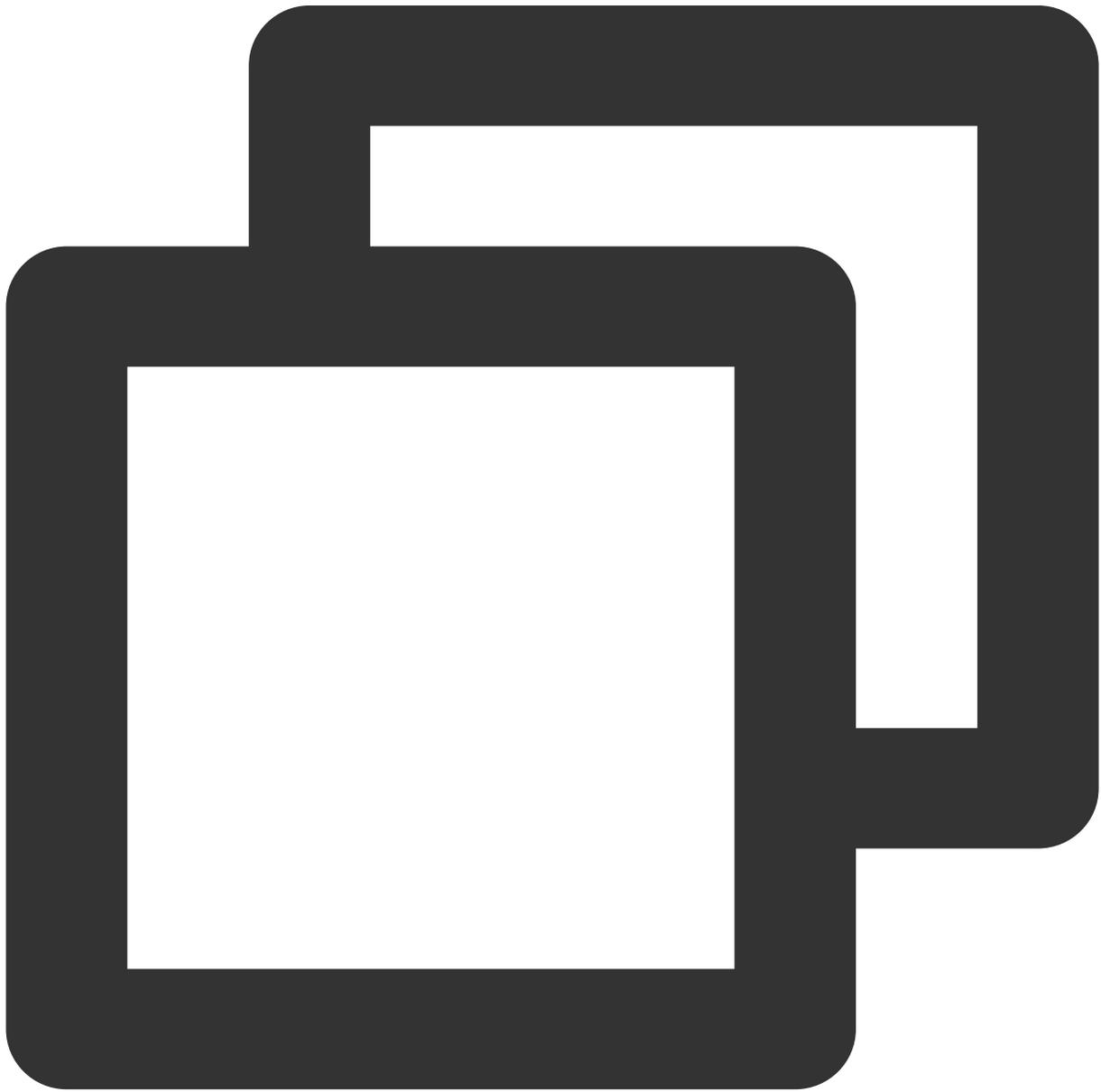
```
./coscli bucket-tagging --method delete cos://exmaple-alias
```

查询存储桶或文件列表 - ls

最近更新时间：2024-06-27 14:45:47

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

命令格式



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

ls 命令包含以下参数：

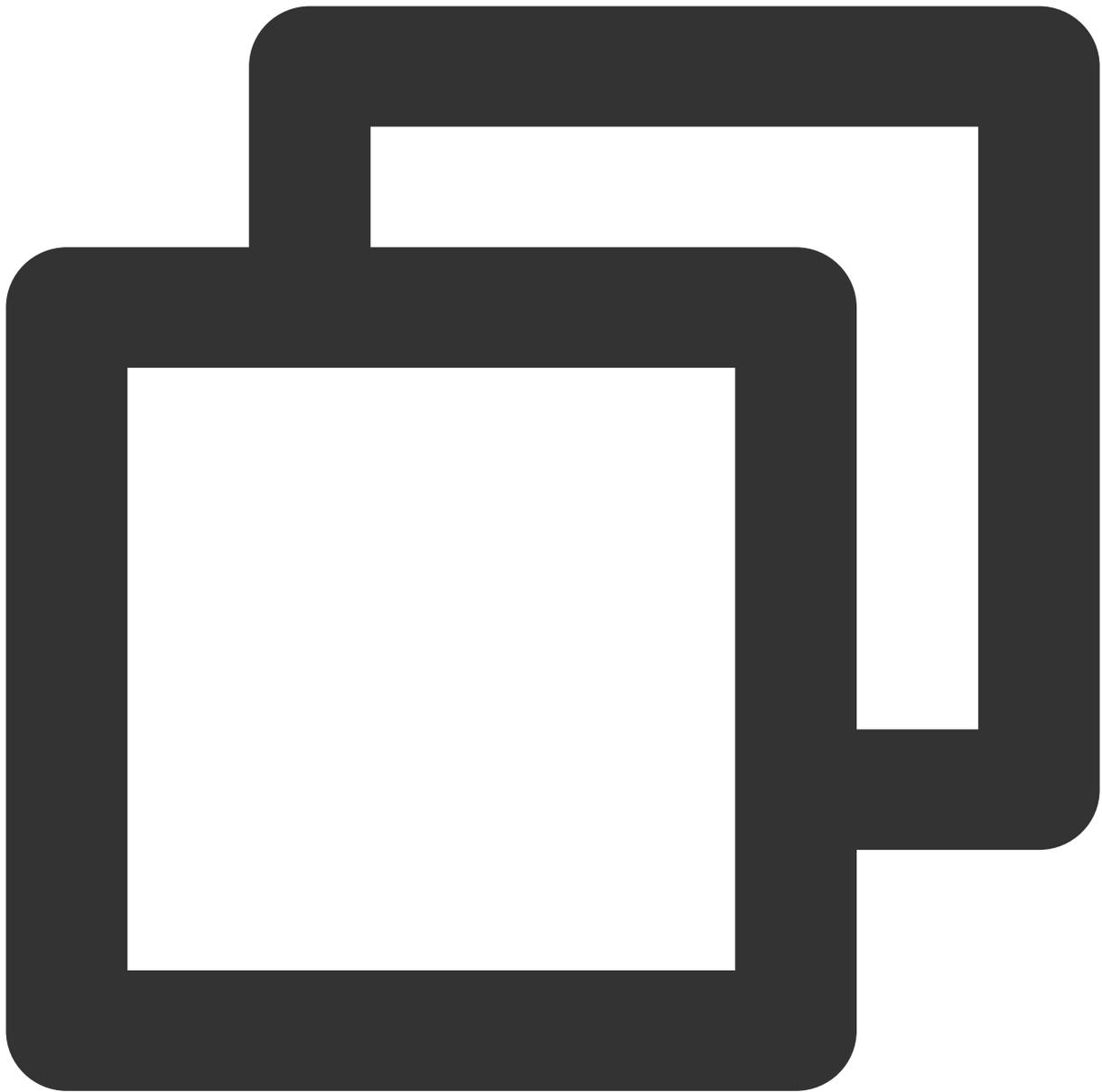
参数格式	参数用途	示例
cos://<bucket-name>	可选参数。指定需要访问的存储桶。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> flag。	使用桶别名访问： <code>cos://example-alias</code> 使用桶名称访问： <code>cos://examplebucket-1250000000</code>
/prefix/	可选参数。指定某一文件夹	/picture/

ls 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件
-r	--recursive	是否递归地遍历文件夹，并列出所有文件
无	--limit	指定列出的最大数量（传0或不传则默认为10000）

说明

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

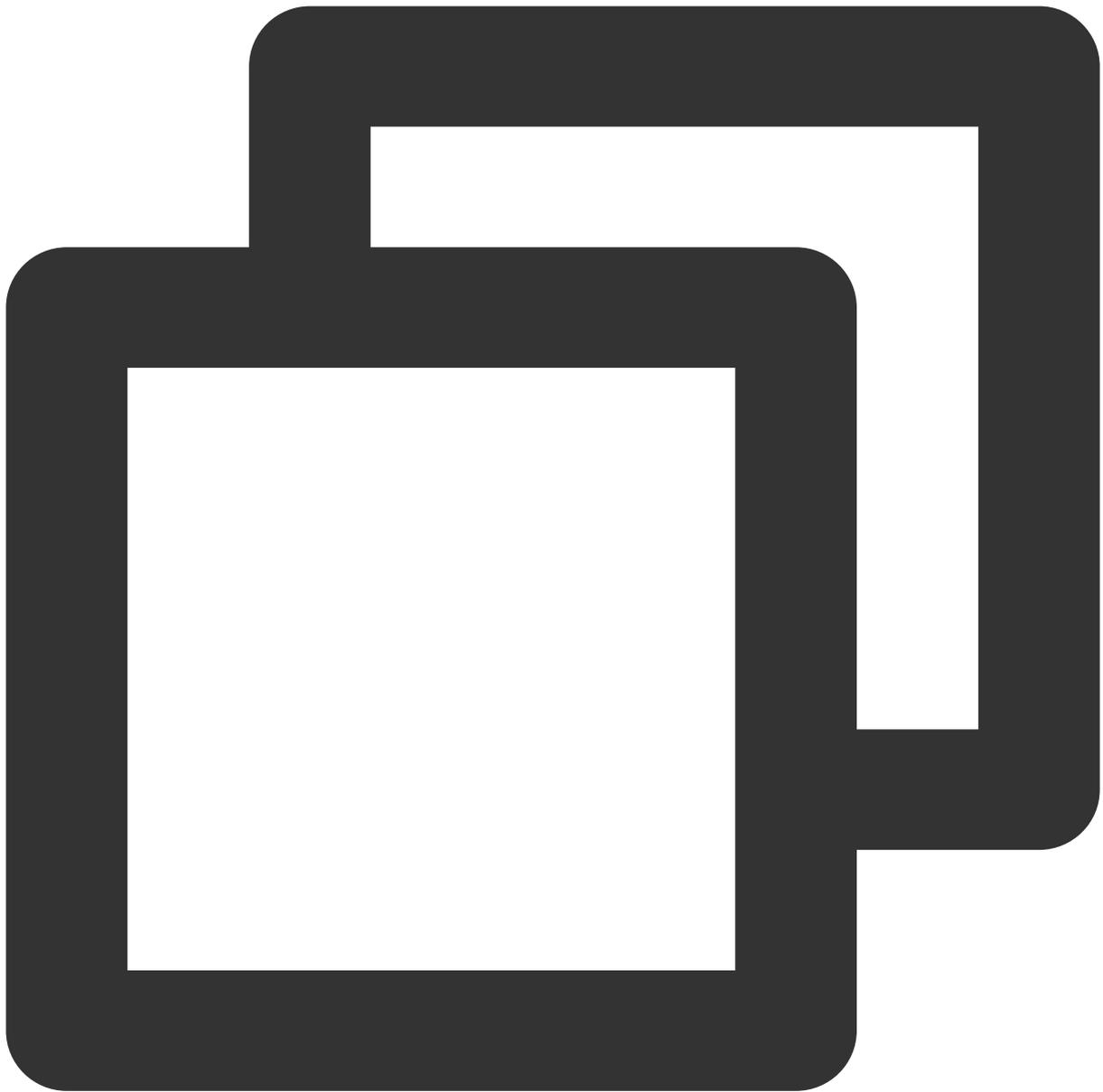


```
./coscli ls cos://bucket1 -r --include ".*.mp4"
```

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

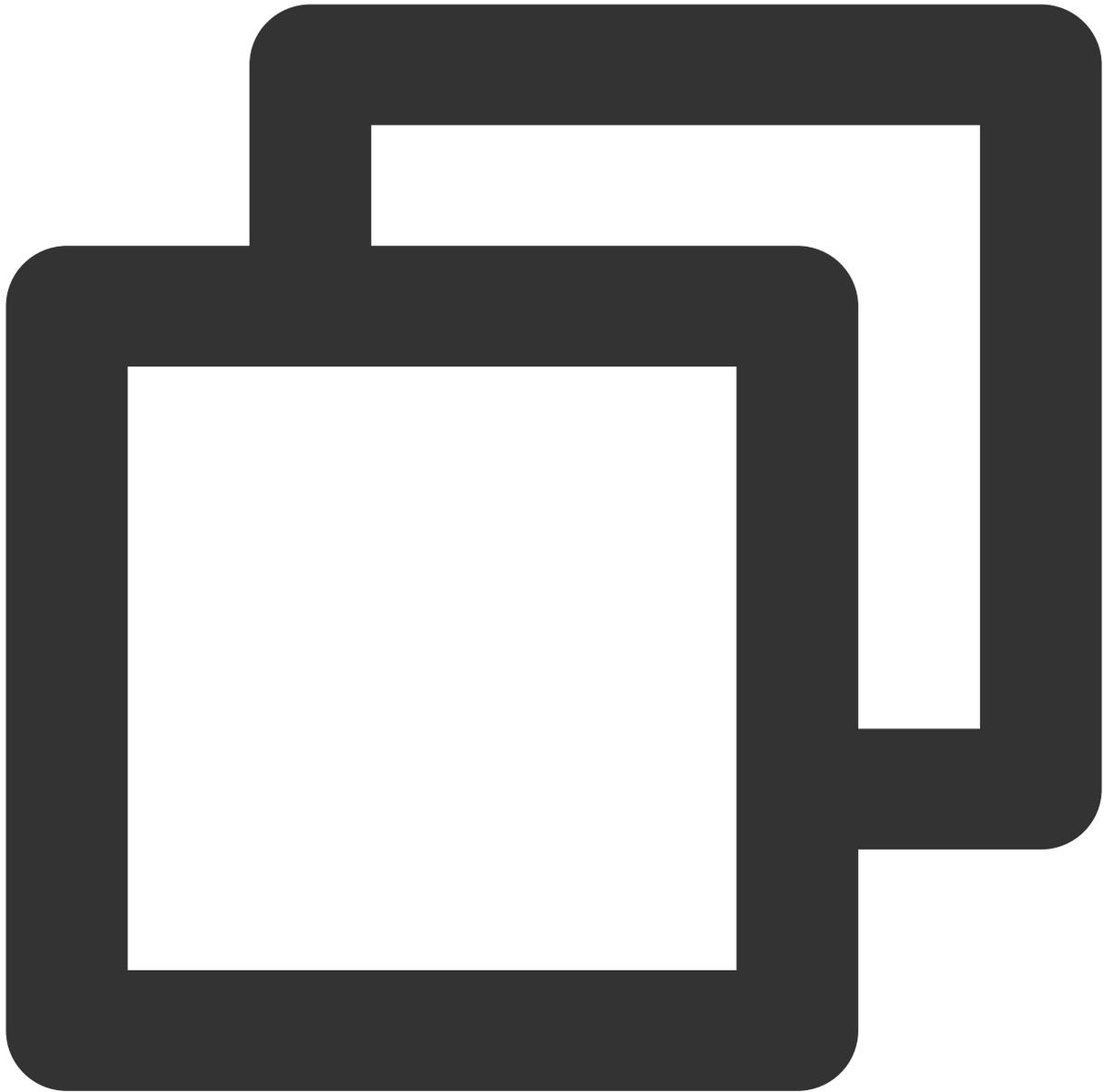
操作示例

列出当前账号下所有存储桶



```
./coscli ls
```

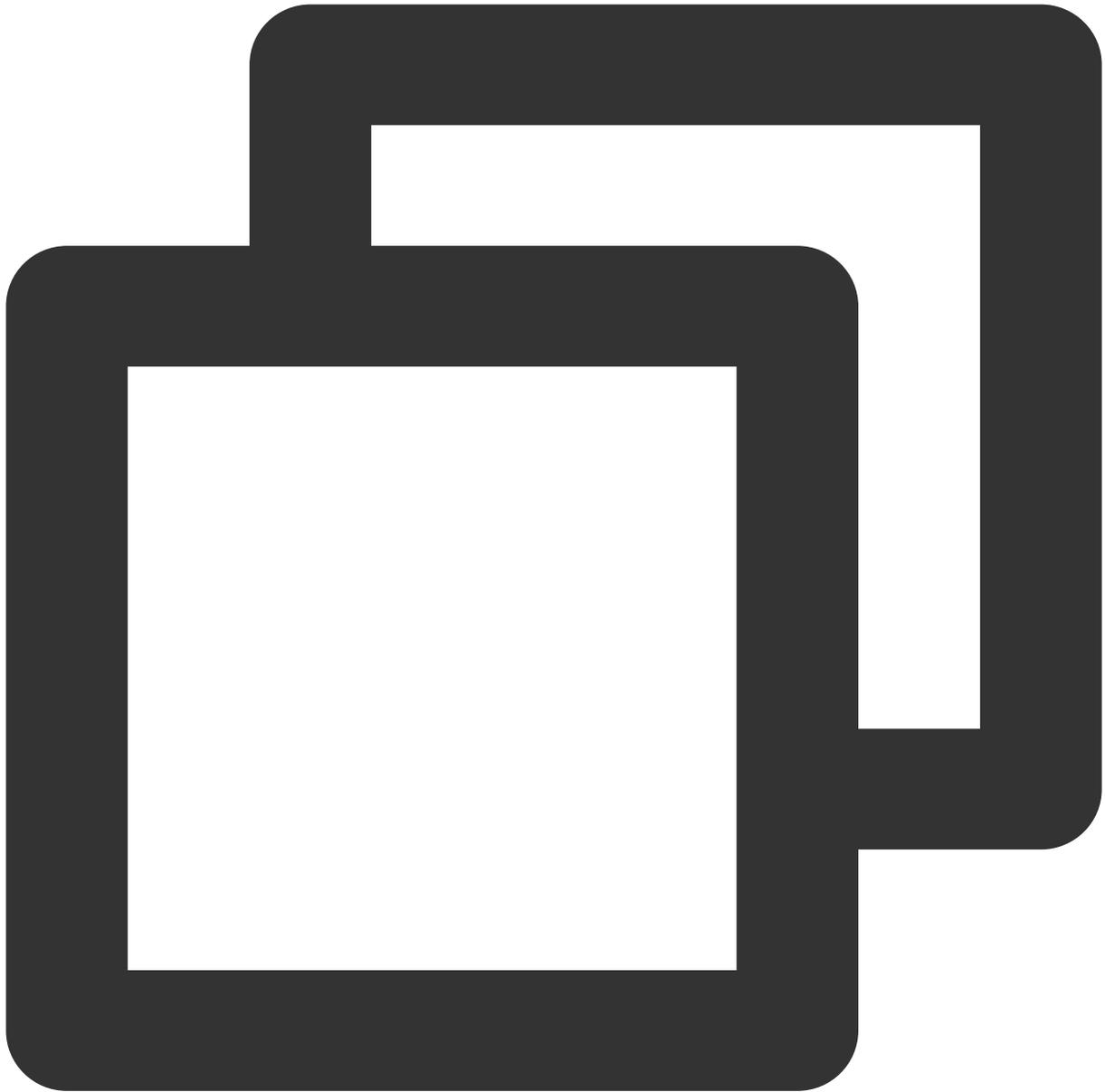
返回的信息包括：桶名称，地域，创建时间，存储桶总数。示例如下：



BUCKET NAME	REGION	CREATE DATE
examplebucket-1250000000	ap-nanjing	2022-01-01T00:00:00Z
TOTAL BUCKETS:		1

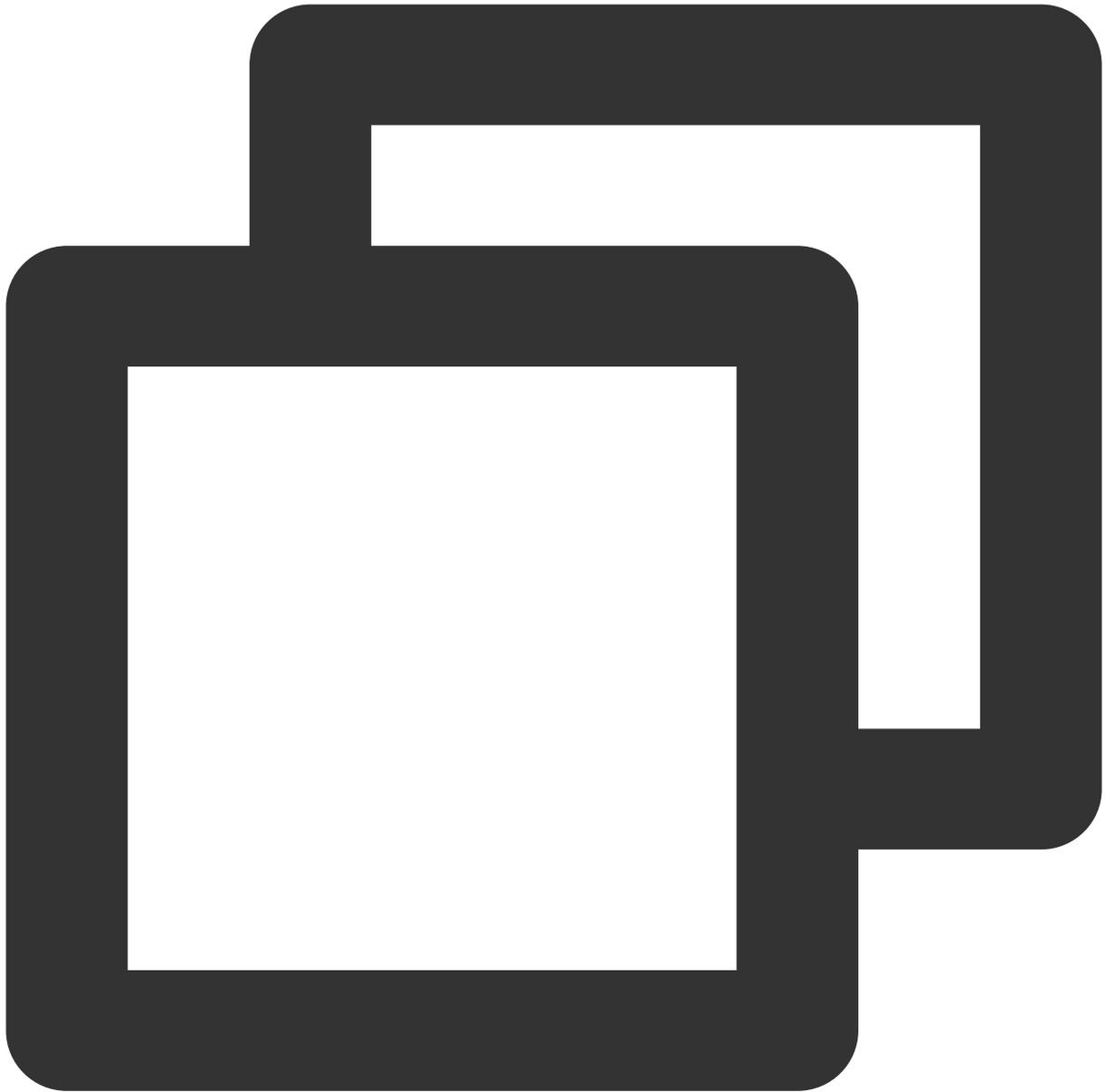
列出文件

列出 **bucket1** 存储桶中的所有文件



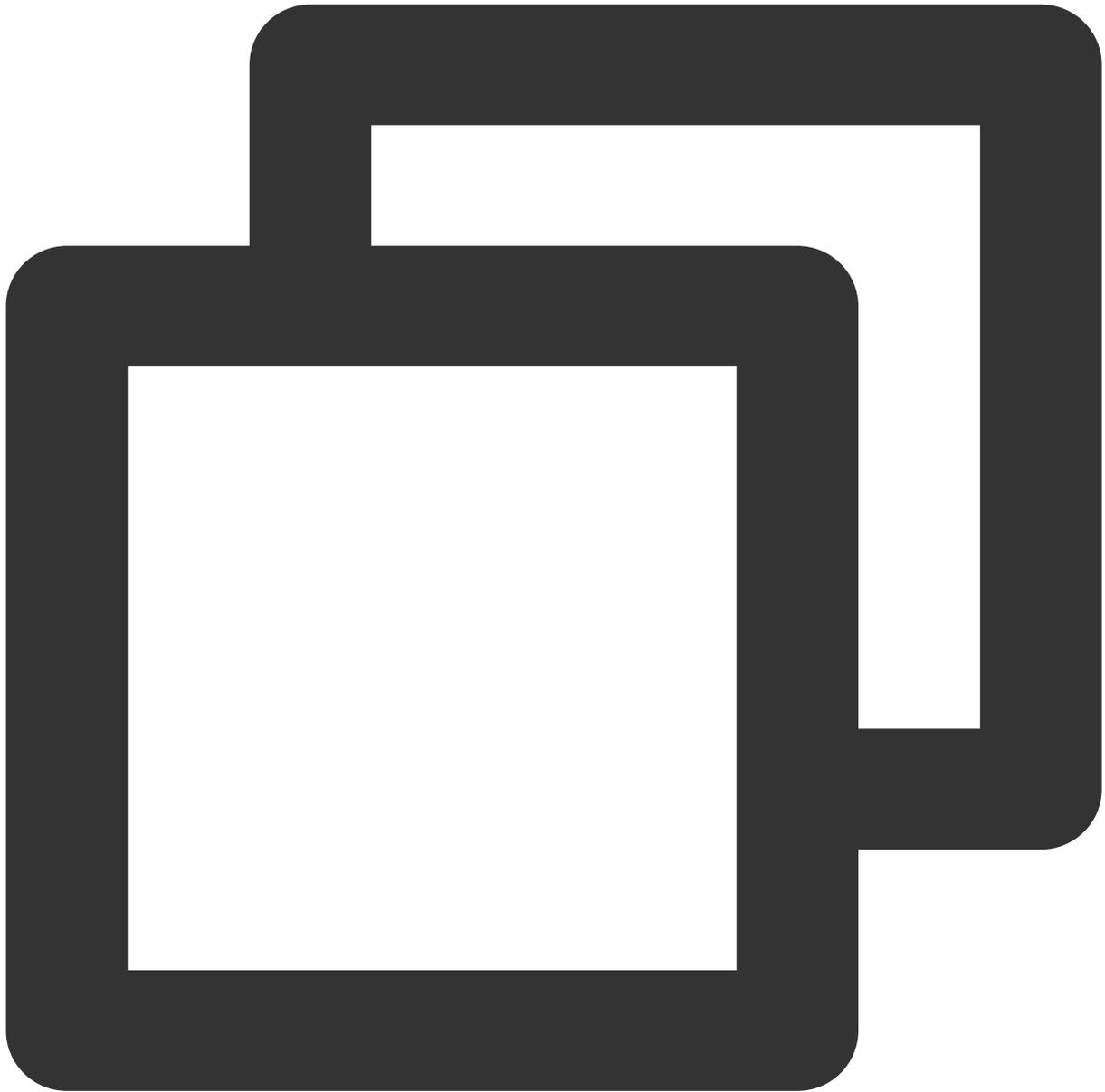
```
./coscli ls cos://bucket1
```

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



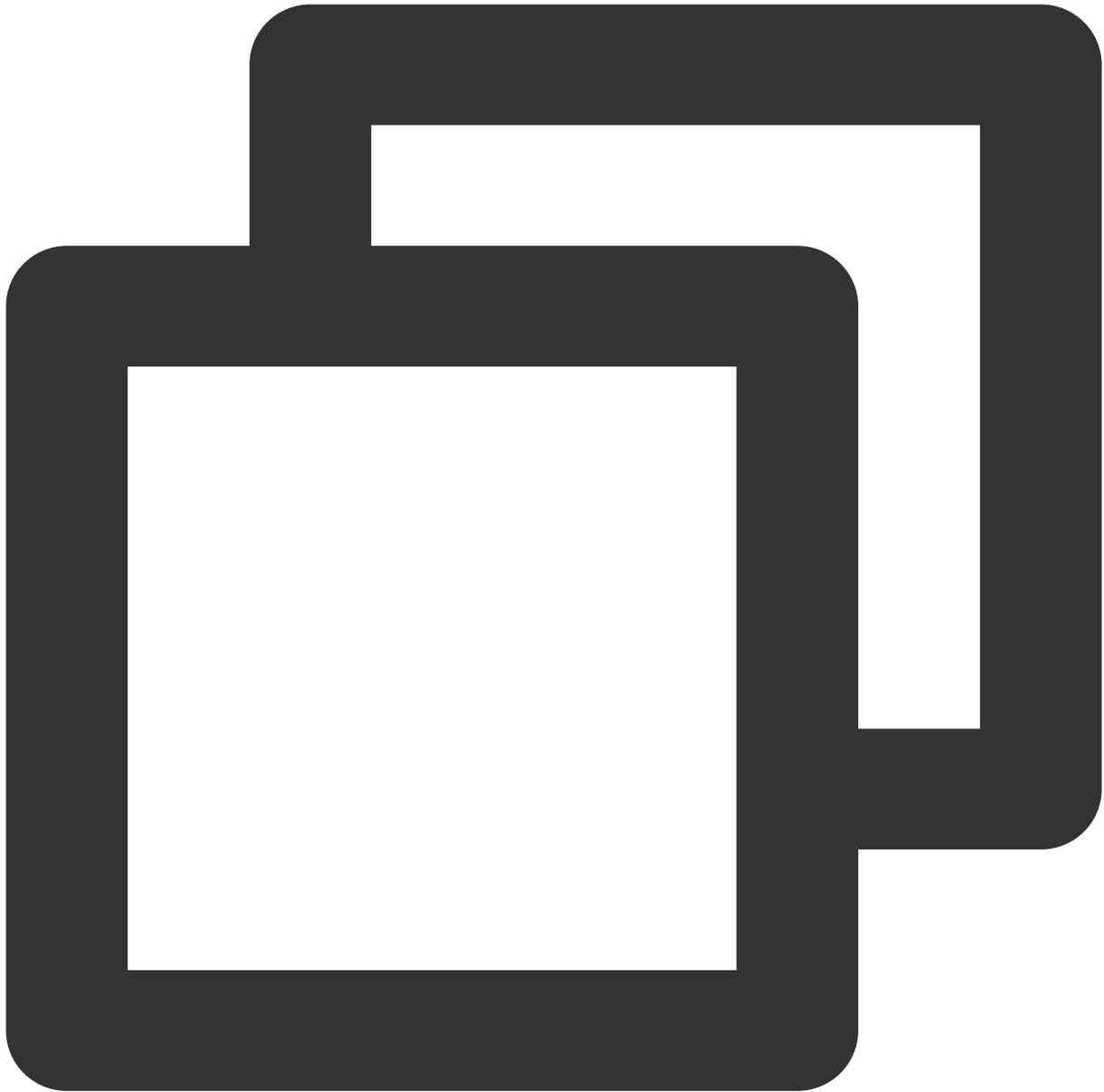
KEY	TYPE	LAST MODIFIED	SIZE
test.txt	STANDARD	2022-01-01T00:00:00.000Z	2 B
TOTAL OBJECTS:			1

列出 bucket1 存储桶中 picture 文件夹下的所有文件和文件夹



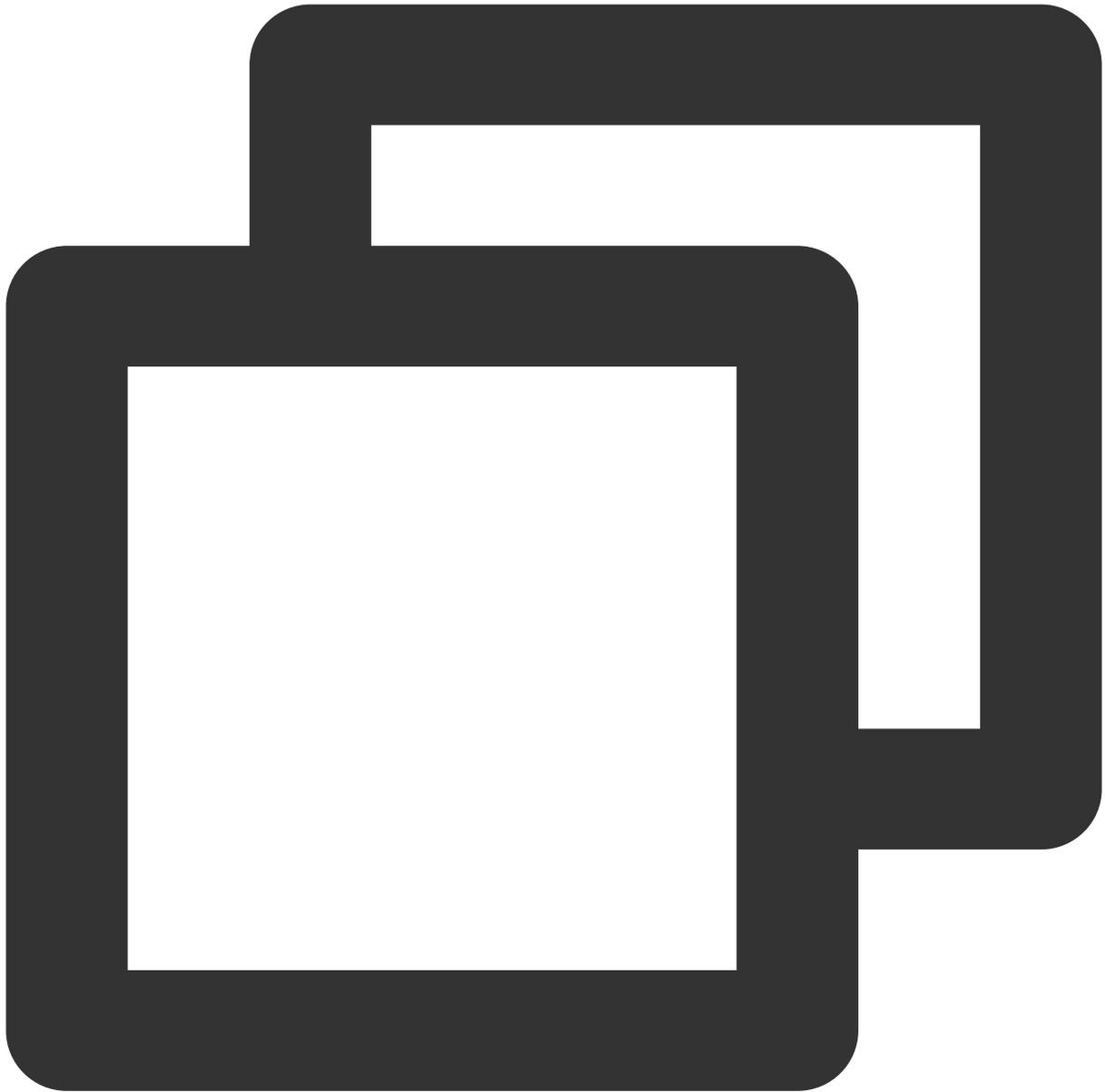
```
./coscli ls cos://bucket1/picture/
```

普通列出仅返回查询路径所在层级的数据，不对子路径进行展开。示例如下：



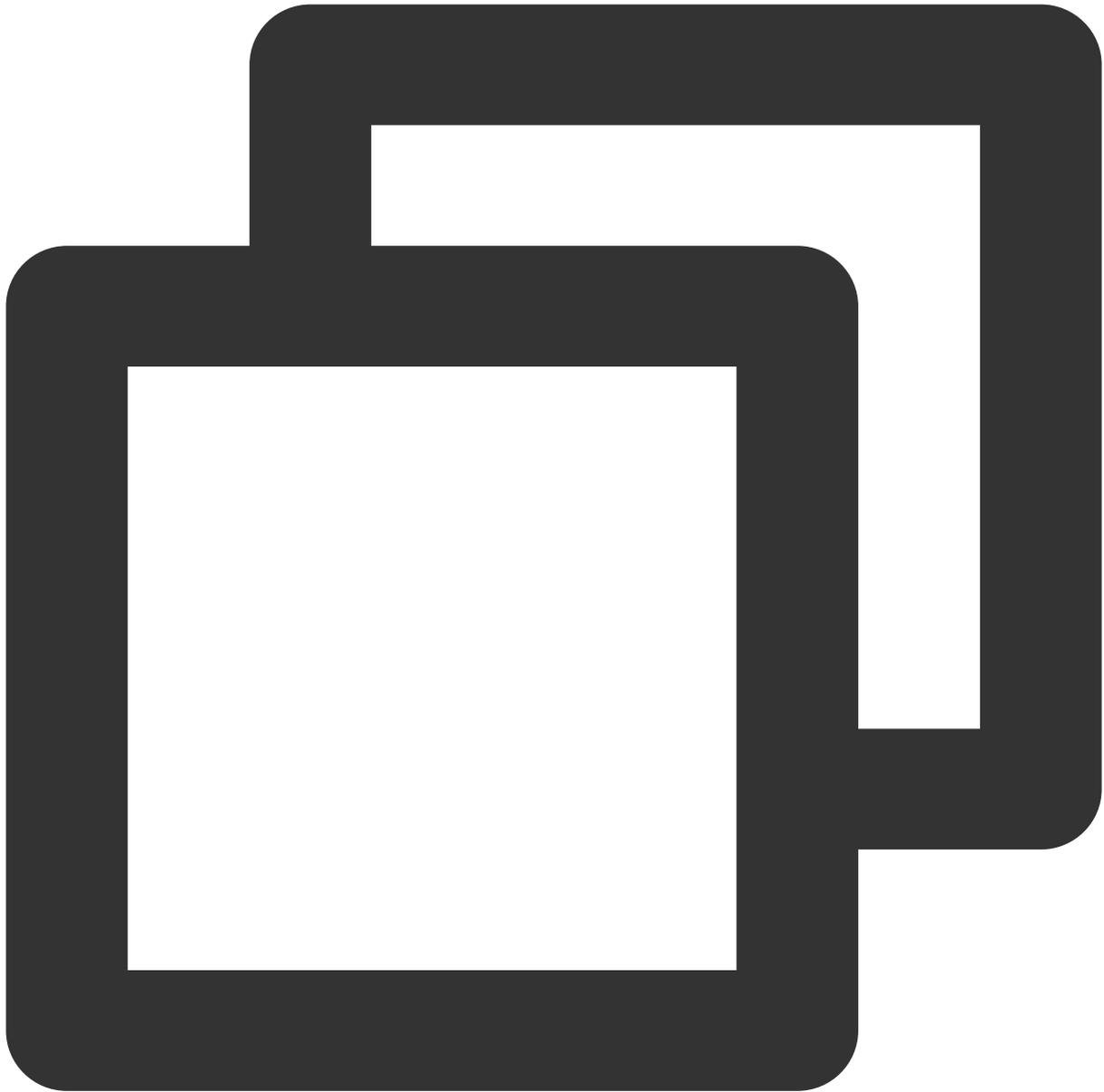
KEY	TYPE	LAST MODIFIED	SIZE
picture/subfolder	DIR		
picture/pic1.png	STANDARD	2022-01-01T00:00:00.000Z	162 B
TOTAL OBJECTS:			2

递归列出 bucket1 存储桶中 picture 文件夹下的所有文件



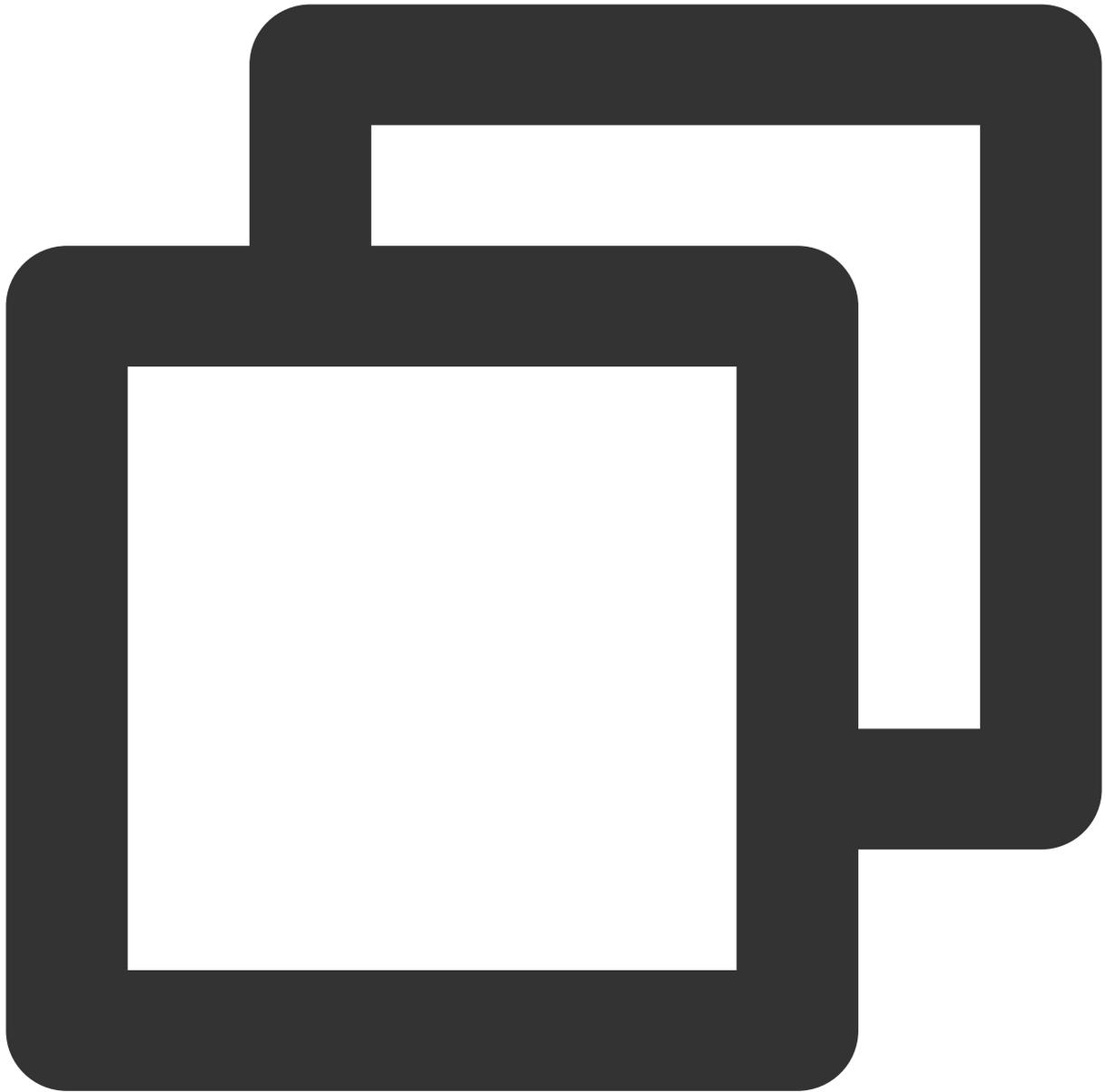
```
./coscli ls cos://bucket1/picture/ -r
```

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



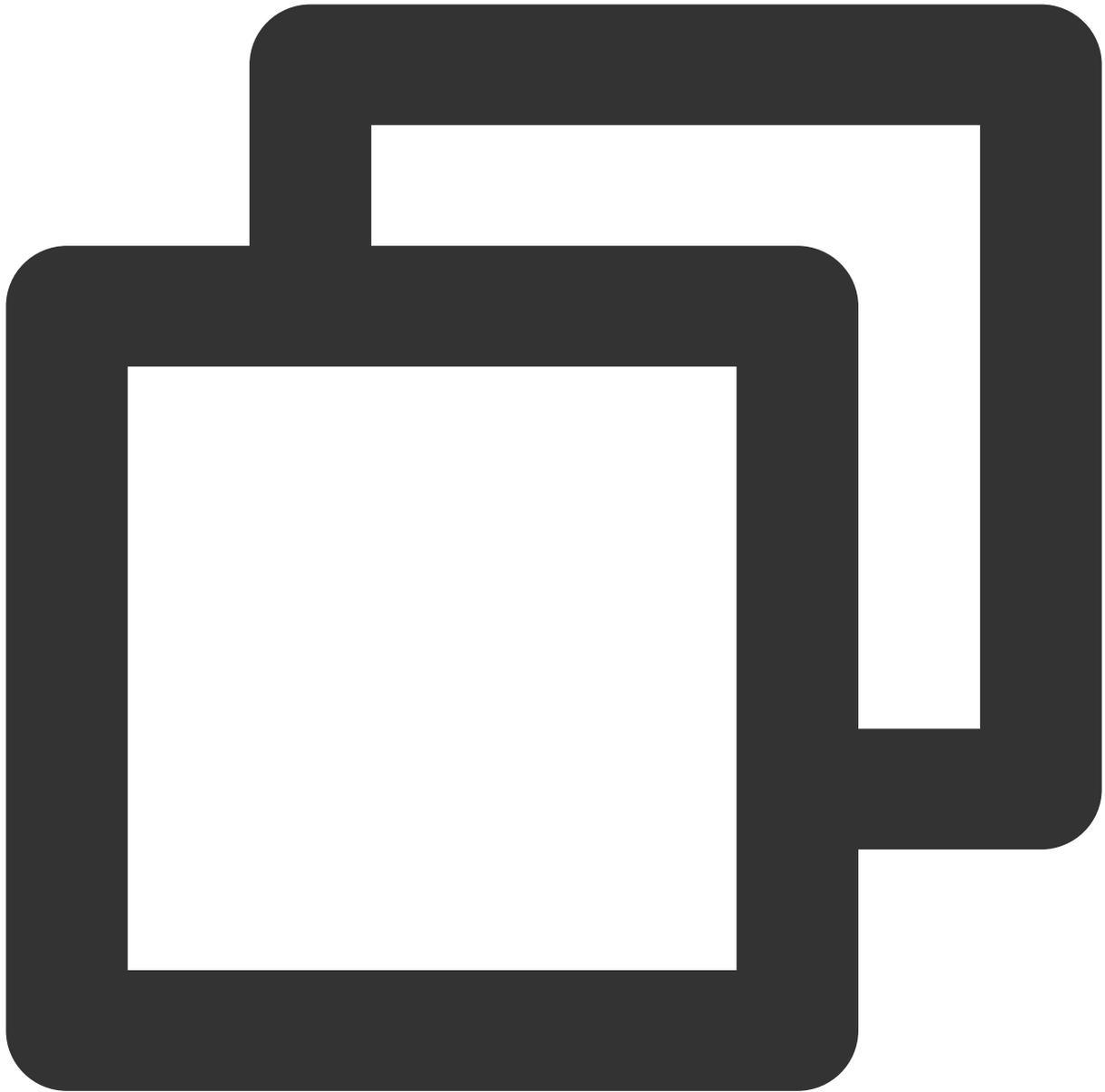
KEY	TYPE	LAST MODIFIED	SIZE
picture/subfolder/pic2.png	DIR		
picture/subfolder	DIR		
picture/pic1.png	STANDARD	2022-01-01T00:00:00.000Z	162 B
TOTAL OBJECTS:			2

递归列出 bucket1 存储桶中所有的 .mp4 类型文件



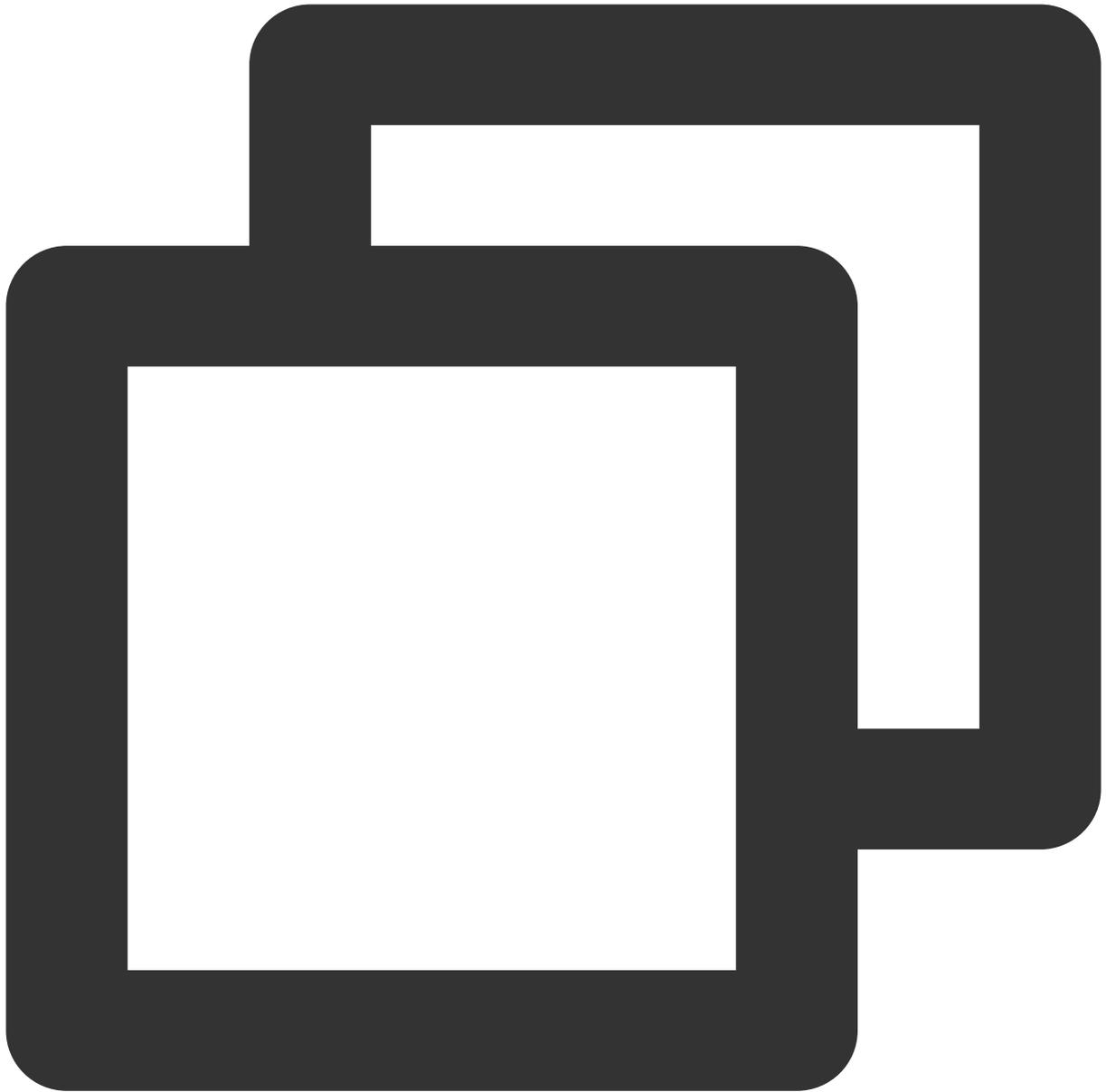
```
./coscli ls cos://bucket1 -r --include .*mp4
```

递归列出 **bucket1** 存储桶中的所有不是 **.mp4** 类型的文件



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

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



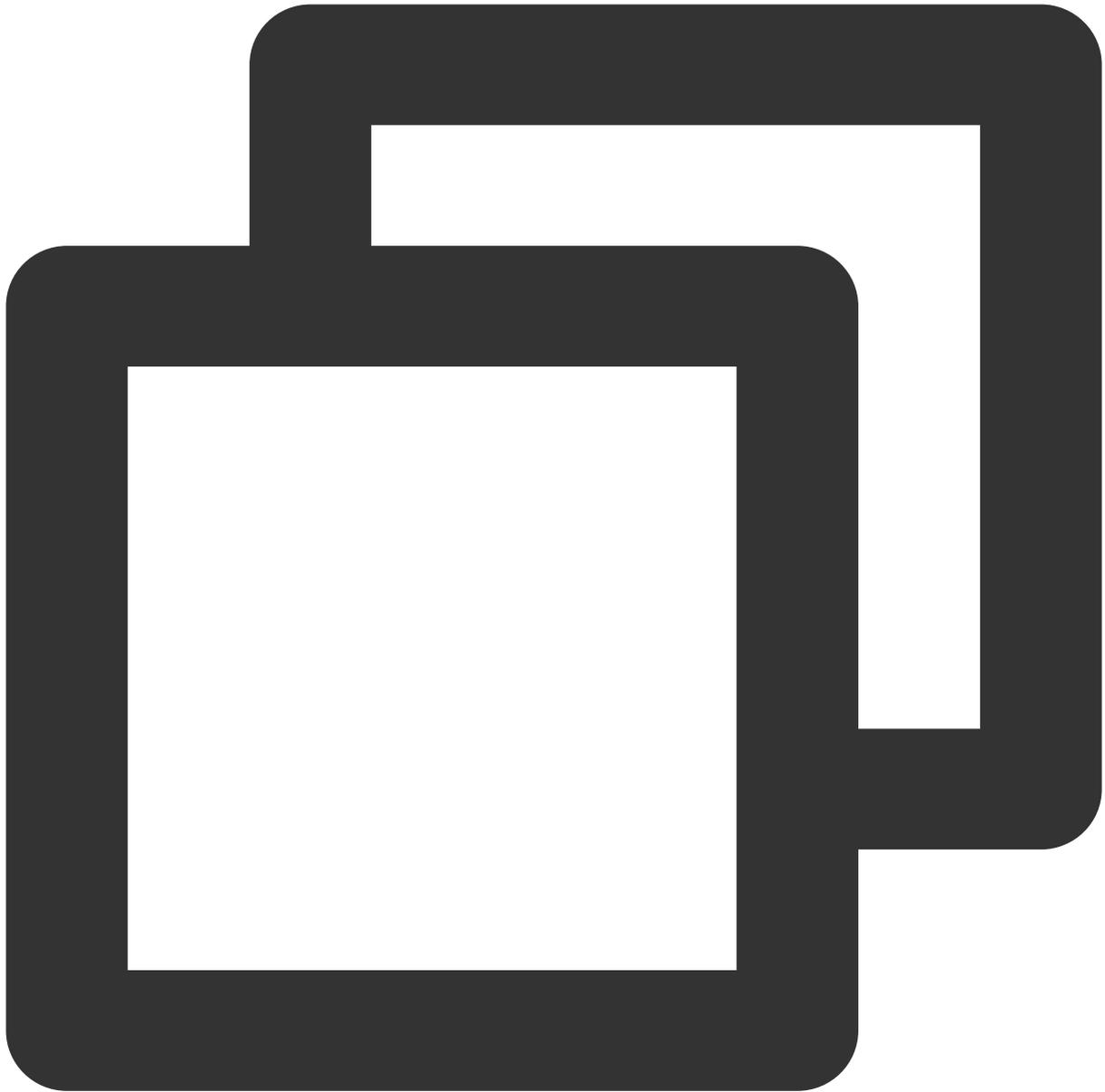
```
./coscli ls cos://bucket1/picture -r --include ^picture/test.* --exclude .*\.jpg
```

获取不同类型文件的统计信息 - du

最近更新时间：2024-01-06 15:54:24

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

命令格式



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

du 命令包含以下参数：

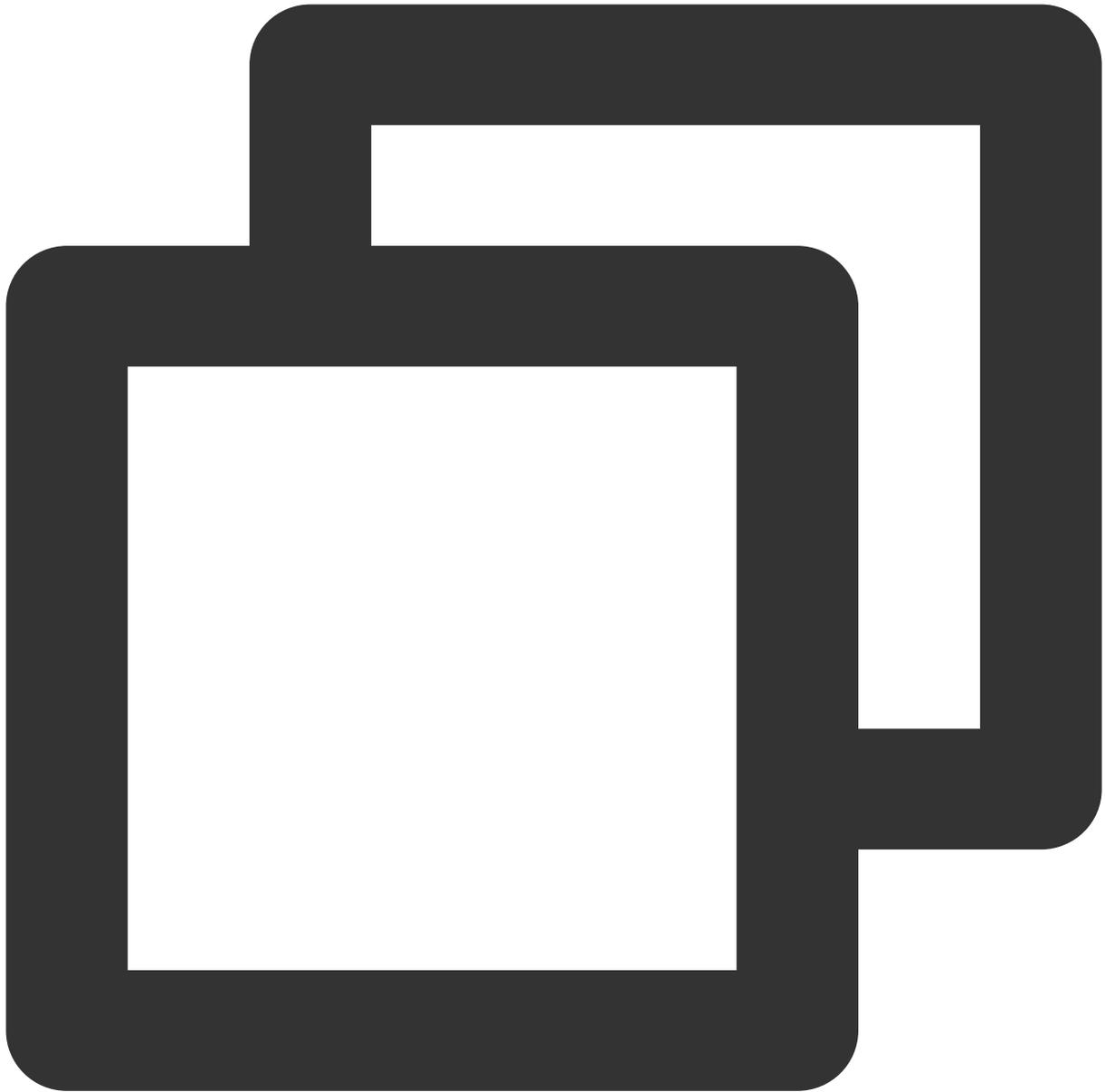
参数格式	参数用途	示例
cos://<bucket-name>	指定需要访问的存储桶。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint flag</code> 。	使用桶别名访问：cos://example-alias 使用桶名称访问： cos://examplebucket-1250000000
/prefix/	可选参数。指定某一文件夹	/picture/

du 命令包含以下可选 flag :

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

说明

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

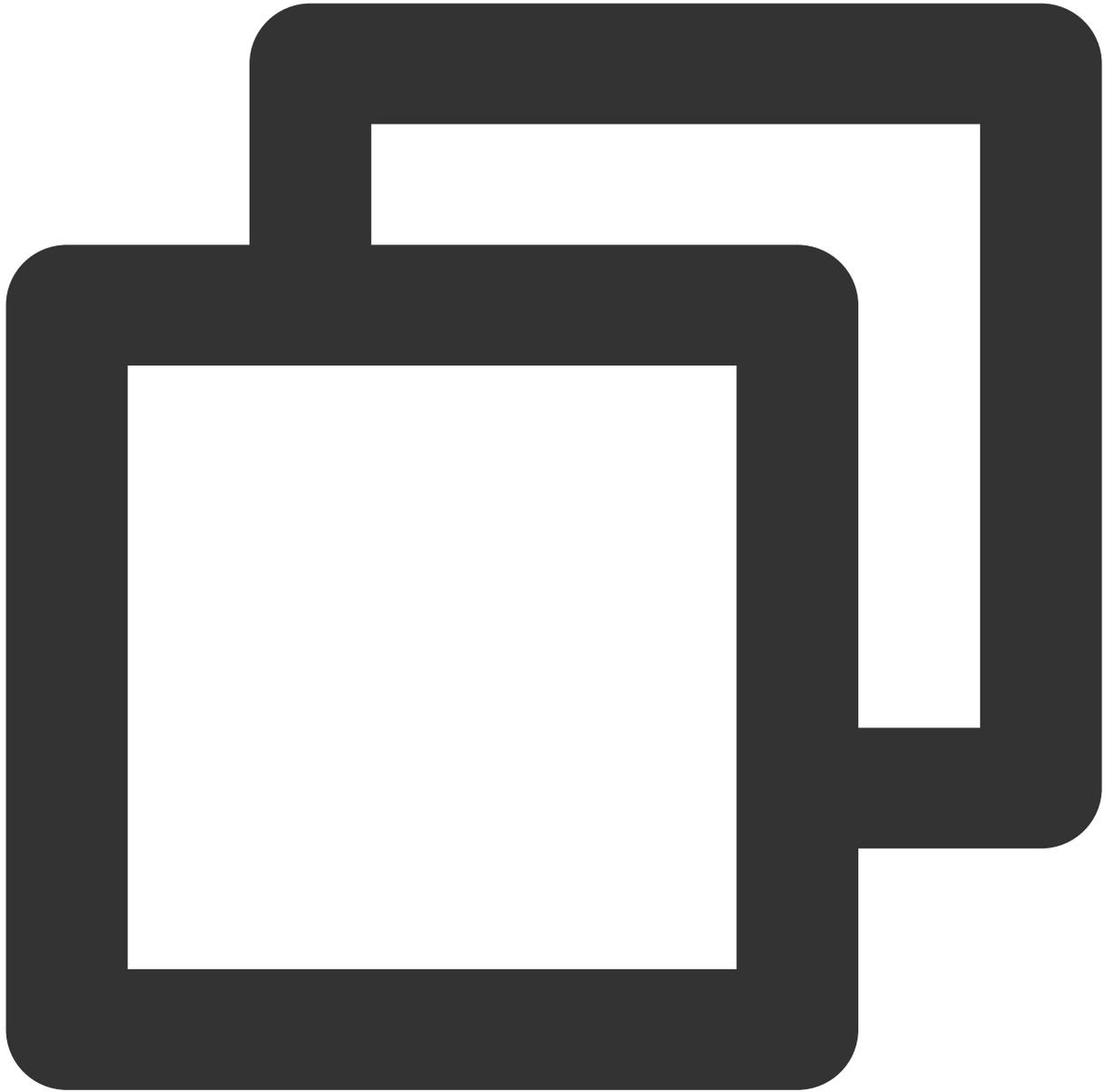


```
./coscli du cos://bucket1/picture/ --include ".*.mp4"
```

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

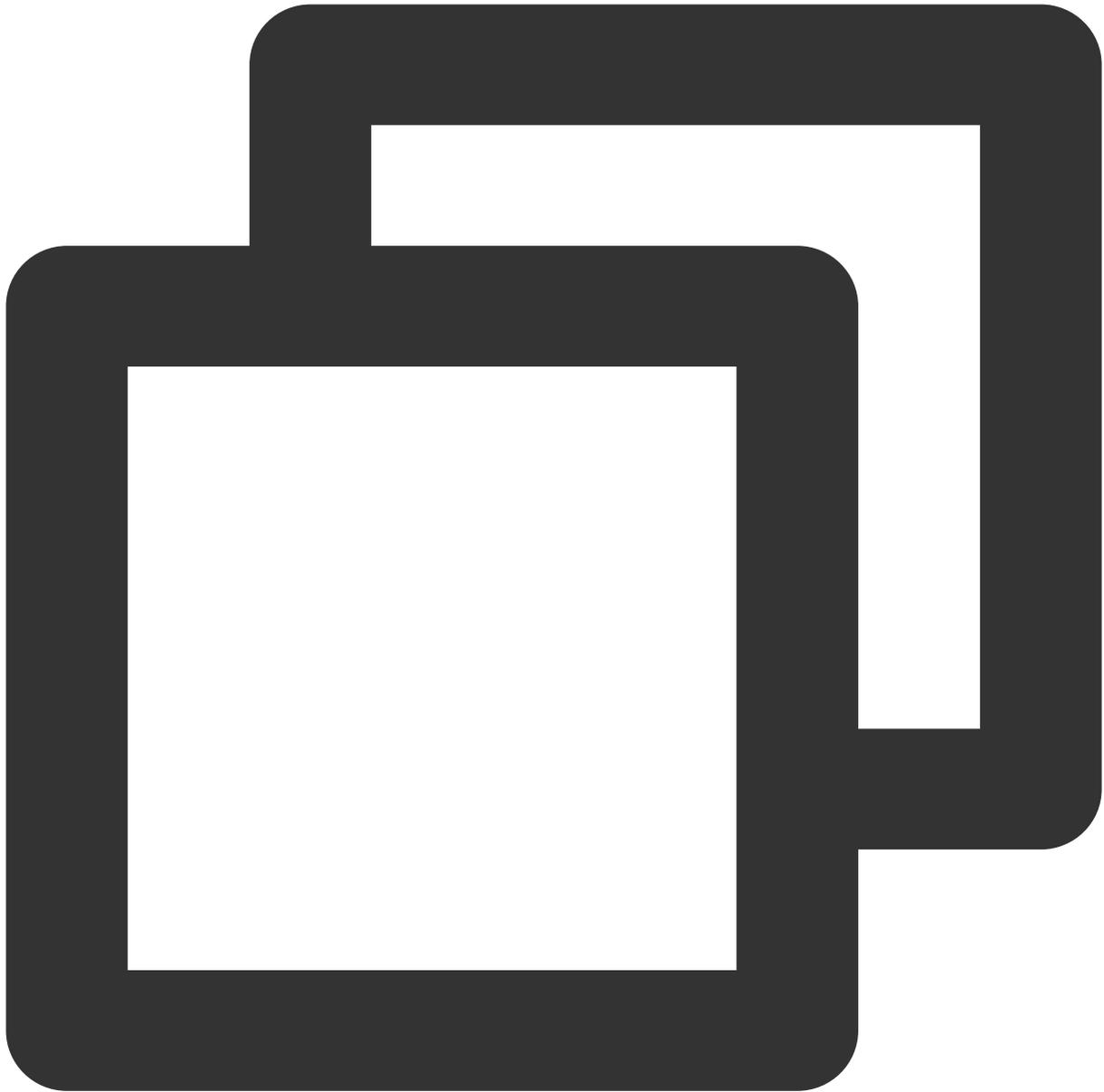
操作示例

列出 bucket1 存储桶内文件的统计信息



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

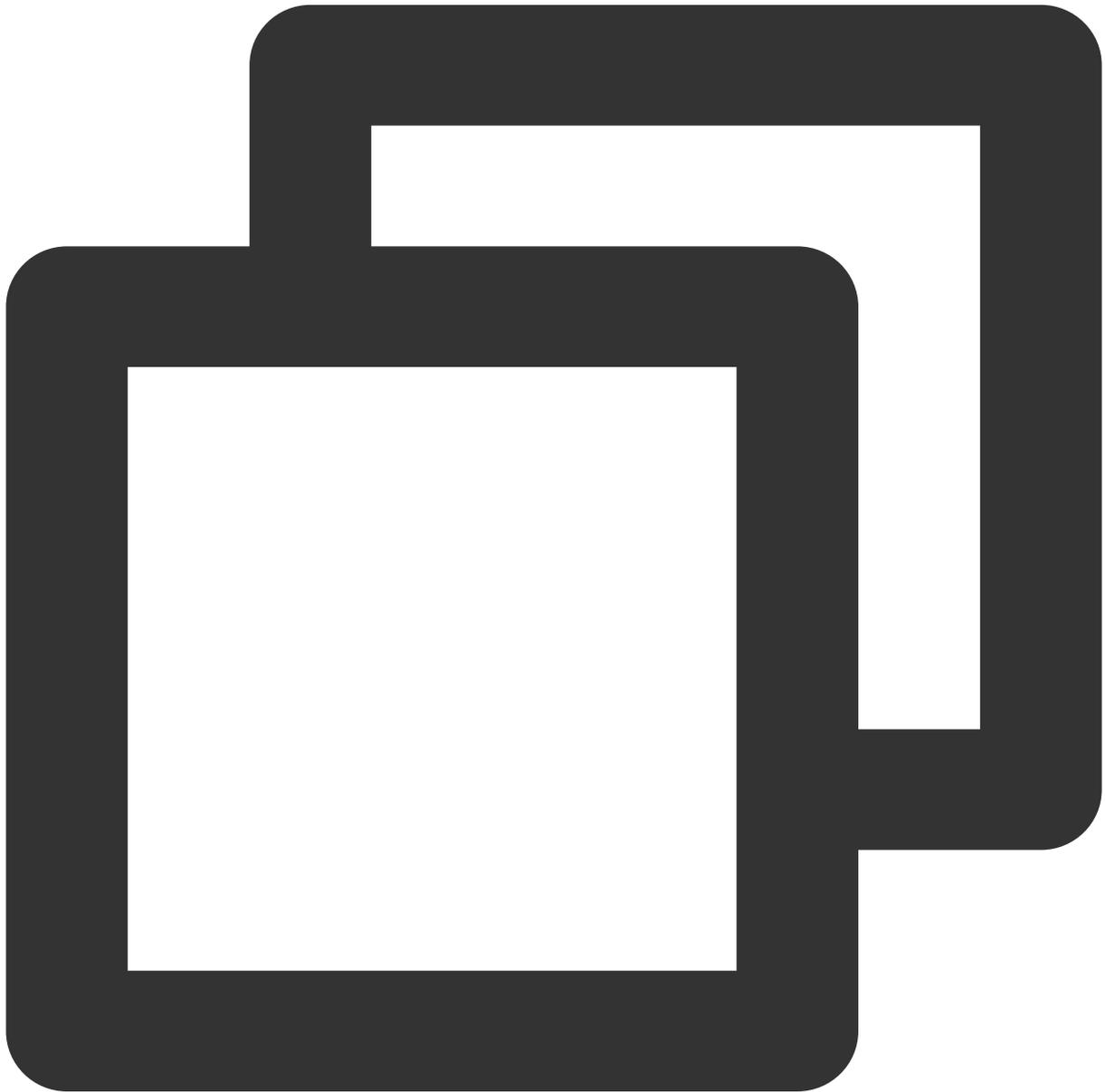
返回的结果示例如下，输出的信息包括：该桶各存储类型下的对象数量和大小、桶内对象总数、桶内对象总容量。



STORAGE CLASS	OBJECTS COUNT	TOTAL SIZE
STANDARD	2	164 B
STANDARD_IA	0	0 B
INTELLIGENT_TIERING	0	0 B
ARCHIVE	0	0 B
DEEP_ARCHIVE	0	0 B
MAZ_STANDARD	0	0 B
MAZ_STANDARD_IA	0	0 B
MAZ_INTELLIGENT_TIERING	0	0 B
MAZ_ARCHIVE	0	0 B

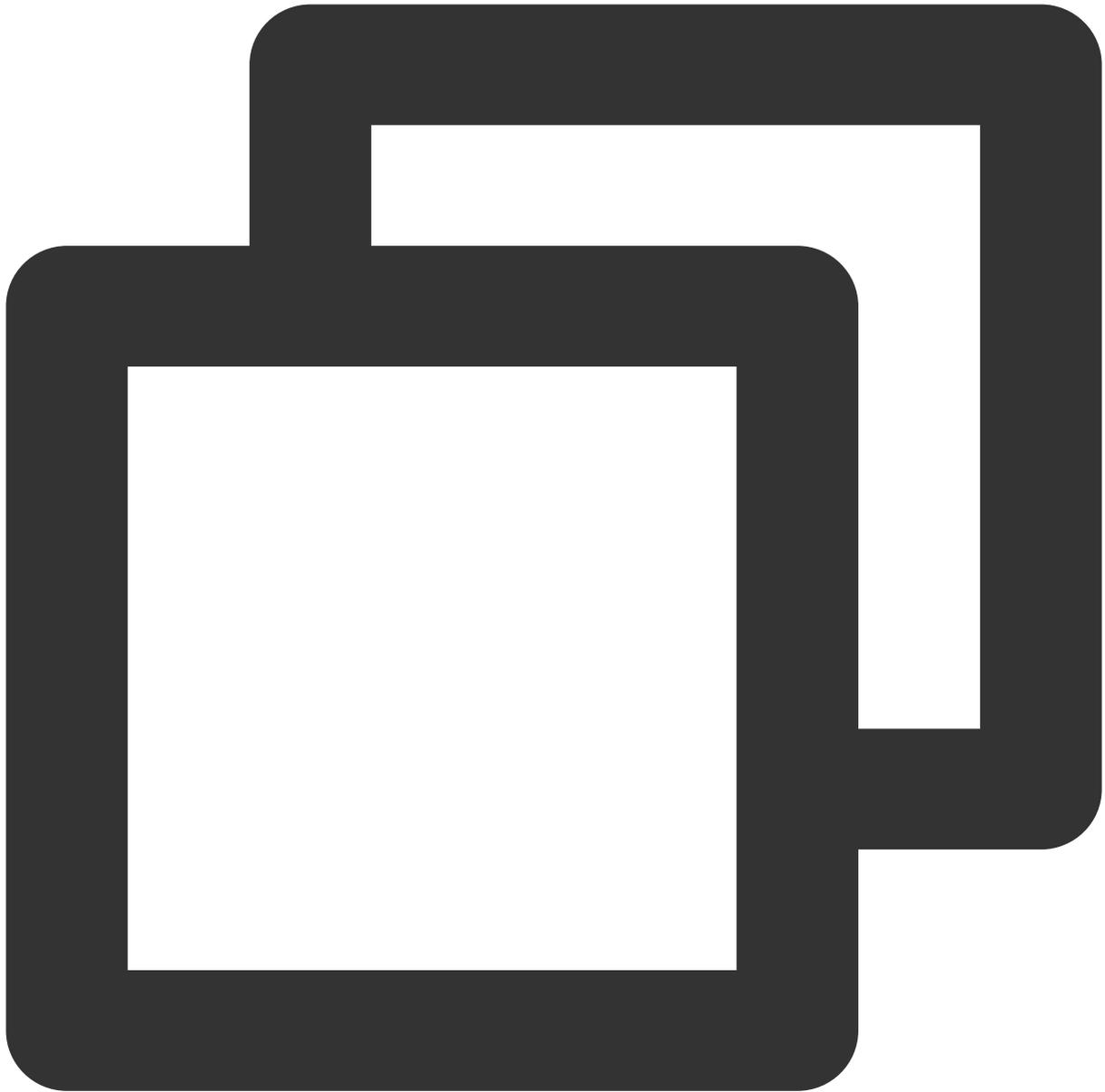
```
-----+-----+-----  
INFO[2022-12-14 17:35:41] Total Objects Count: 2  
INFO[2022-12-14 17:35:41] Total Objects Size: 164 B
```

列出 bucket1 存储桶 picture 文件夹下文件的统计信息



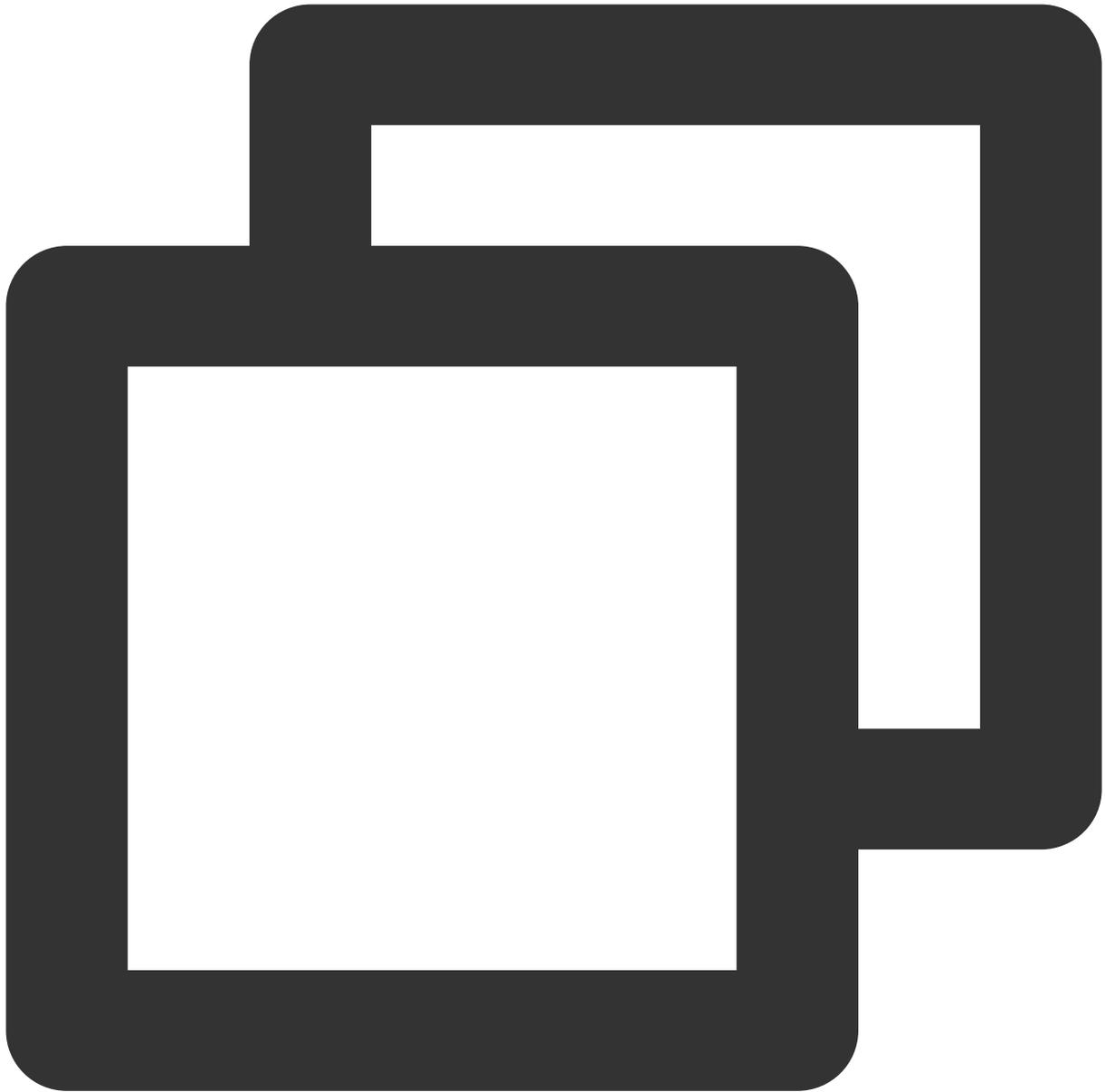
```
./coscli du cos://bucket1/picture/
```

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



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

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



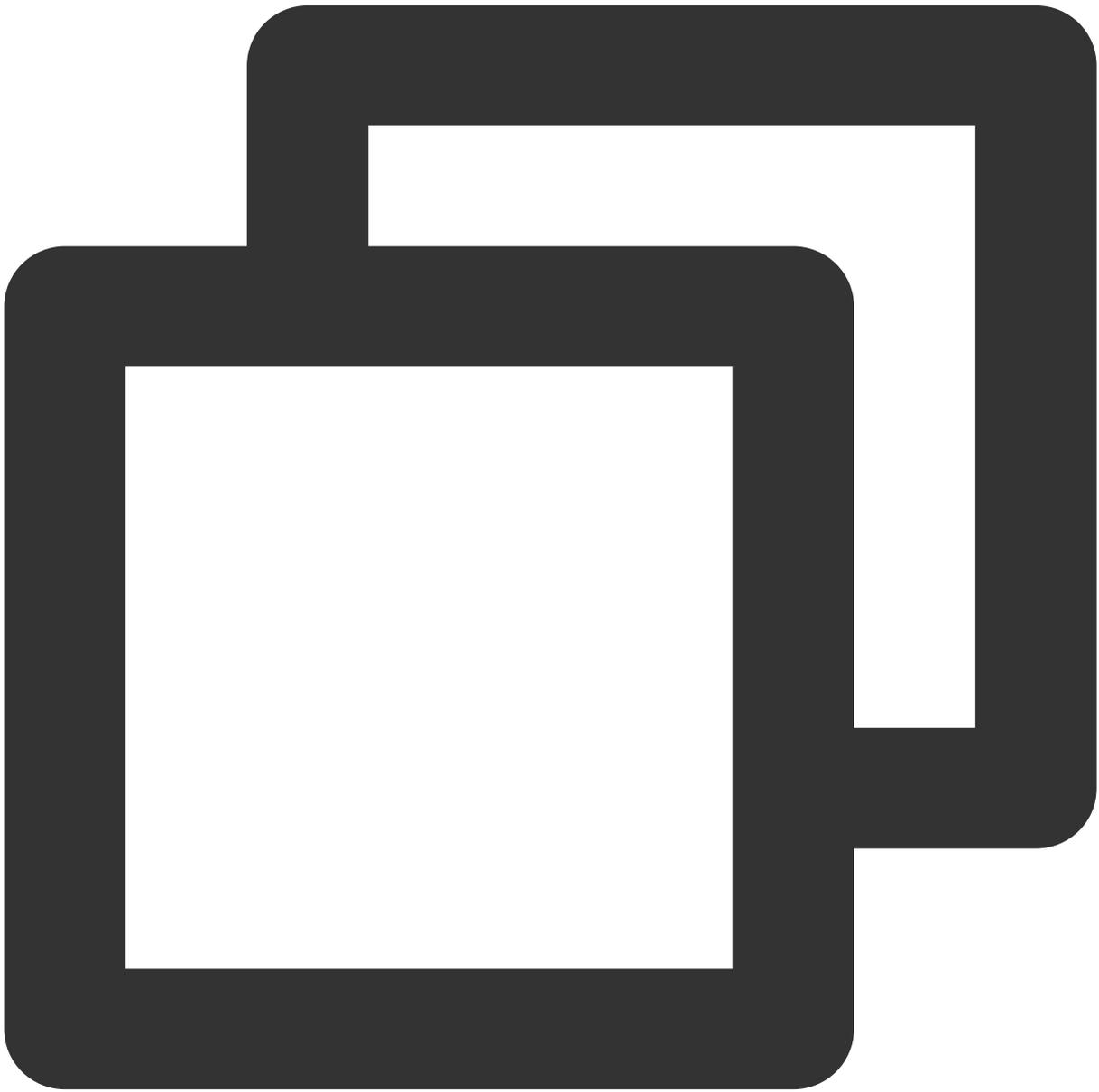
```
./coscli du cos://bucket1/picture/ --exclude *.*.md
```

上传下载或拷贝文件 - cp

最近更新时间：2024-06-27 14:46:34

cp 命令用于上传、下载或拷贝文件。

命令格式



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

cp 命令包含以下参数：

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

cp 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件
-r	--recursive	是否递归遍历文件夹下所有文件
无	--storage-class	指定上传文件的存储类型（默认 STANDARD），更多存储类型，请参见 存储类型概述
无	--part-size	文件分块大小（默认 32MB），单位 MB
无	--thread-num	并发线程数（默认并发 5）
无	--rate-limiting	单链接速率限制（0.1~100MB/s），单位 MB/s
无	--meta	上传文件的元信息。包括部分 HTTP 标准属性（HTTP Header）以及以 <code>x-cos-meta-</code> 开头的用户自定义元数据（User Meta）。文件元信息格式为 <code>header:value#header:value</code> ，示例为 <code>Expires:2022-10-12T00:00:00.000Z#Cache-Control:no-cache#Content-Encoding:gzip#x-cos-meta-x:x</code>
无	--routines	指定文件间并发上传或下载线程的文件数量，默认为 3
无	--fail-output	此选项决定是否启用上传或下载失败时文件的错误输出（默认为 <code>true</code> ，开

		后)。如果启用，文件传输失败将被记录在指定目录内（如果没有指定，默认为 <code>./coscli_output</code> ）。如果禁用，只会将错误文件的数量输出到控制台。
无	<code>--fail-output-path</code>	此选项用于指定记录上传或下载失败文件的错误输出文件夹。通过提供自定义文件夹路径，您可以控制错误输出文件夹的位置和名称。如果未设置此选项，将使用默认的错误日志文件夹 <code>./coscli_output</code> 。
无	<code>--retry-num</code>	限频重试次数（默认为 <code>0</code> ，不重试）。可选 <code>1-10</code> 次。当多台机器同时在一个COS目录上执行下载操作时，可以通过指定此参数进行重试来规避限频错误。
无	<code>--err-retry-num</code>	错误重试次数（默认为 <code>0</code> ）。指定 <code>1-10</code> 次，或者设置为 <code>0</code> 表示不进行重试。
无	<code>--err-retry-interval</code>	重试间隔（仅在指定 <code>--err-retry-num</code> 为 <code>1-10</code> 时可用）。指定 <code>1-10</code> 秒的重试间隔，如果不指定或设置为 <code>0</code> ，每次重试间隔将在 <code>1-10</code> 秒内的随机。
无	<code>--only-current-dir</code>	是否仅上传当前目录中的文件，忽略子目录及其内容（默认为 <code>false</code> ，不忽略）
无	<code>--disable-all-symlink</code>	是否上传时忽略所有软链接子文件和软链接子目录（默认为 <code>true</code> ，不上传）
无	<code>--enable-symlink-dir</code>	是否上传软链接的子目录（默认为 <code>false</code> ，不上传）
无	<code>--disable-crc64</code>	是否关闭CRC64数据验证。（默认为 <code>false</code> ，启用验证）

说明

`cp` 命令在上传和下载大文件时会自动启用并发上传/下载。

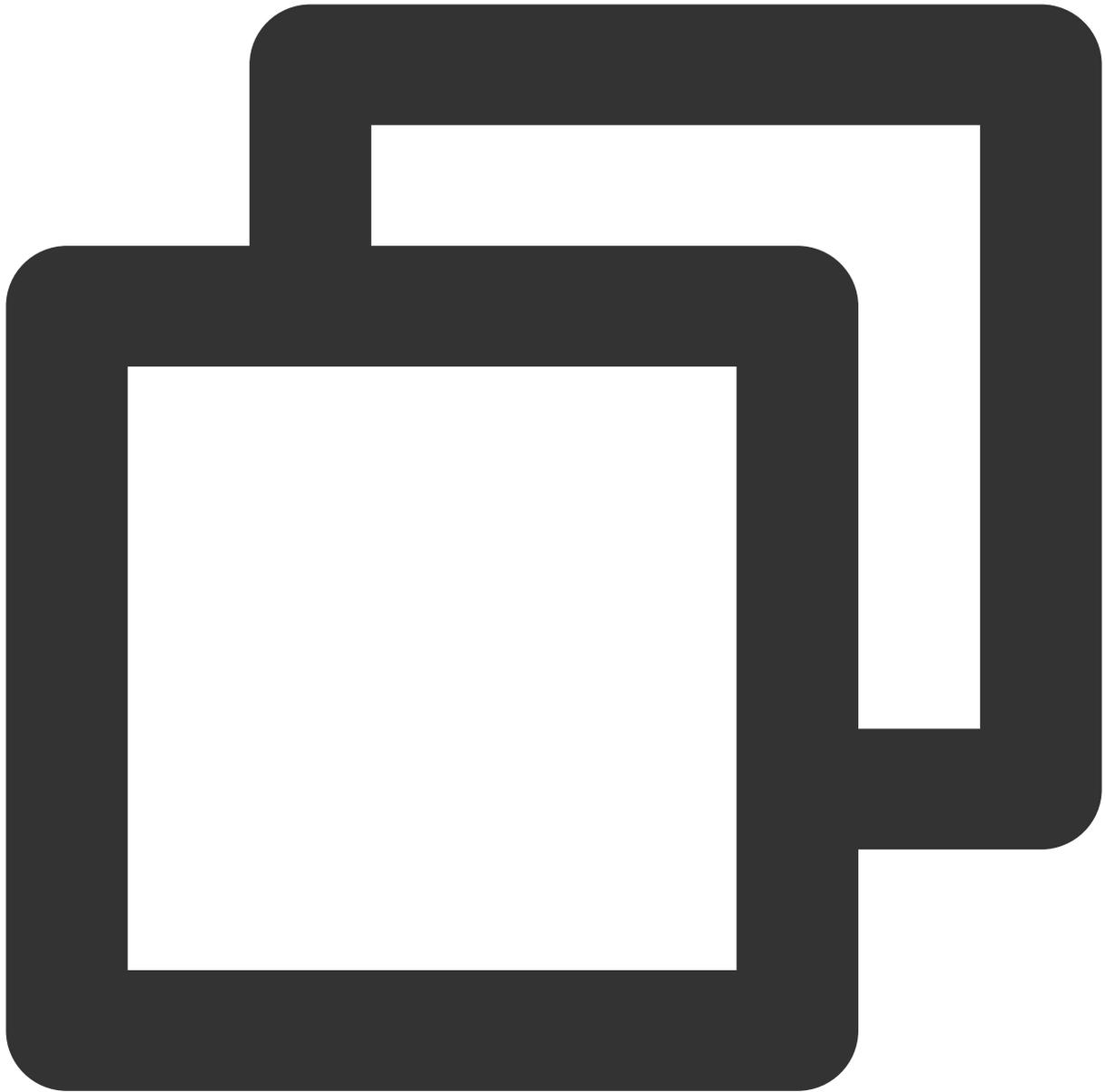
当文件大于 `--part-size` 时，COSCLI 会先将文件按 `--part-size` 进行切块，之后用 `--thread-num` 个线程并发地执行上传/下载任务。

每个线程都会维护一个链接，对于每个链接，您可以使用 `--rate-limiting` 参数对单链接进行限速，当启用并发上传/下载时，总速率为 `--thread-num * --rate-limiting`。

当以分块形式上传/下载文件时，会默认开启断点续传。

`--include` 和 `--exclude` 支持标准正则表达式的语法，您可以使用它来过滤出符合特定条件的文件。

使用 `zsh` 时，您可能需要在 `pattern` 串的两端加上双引号。



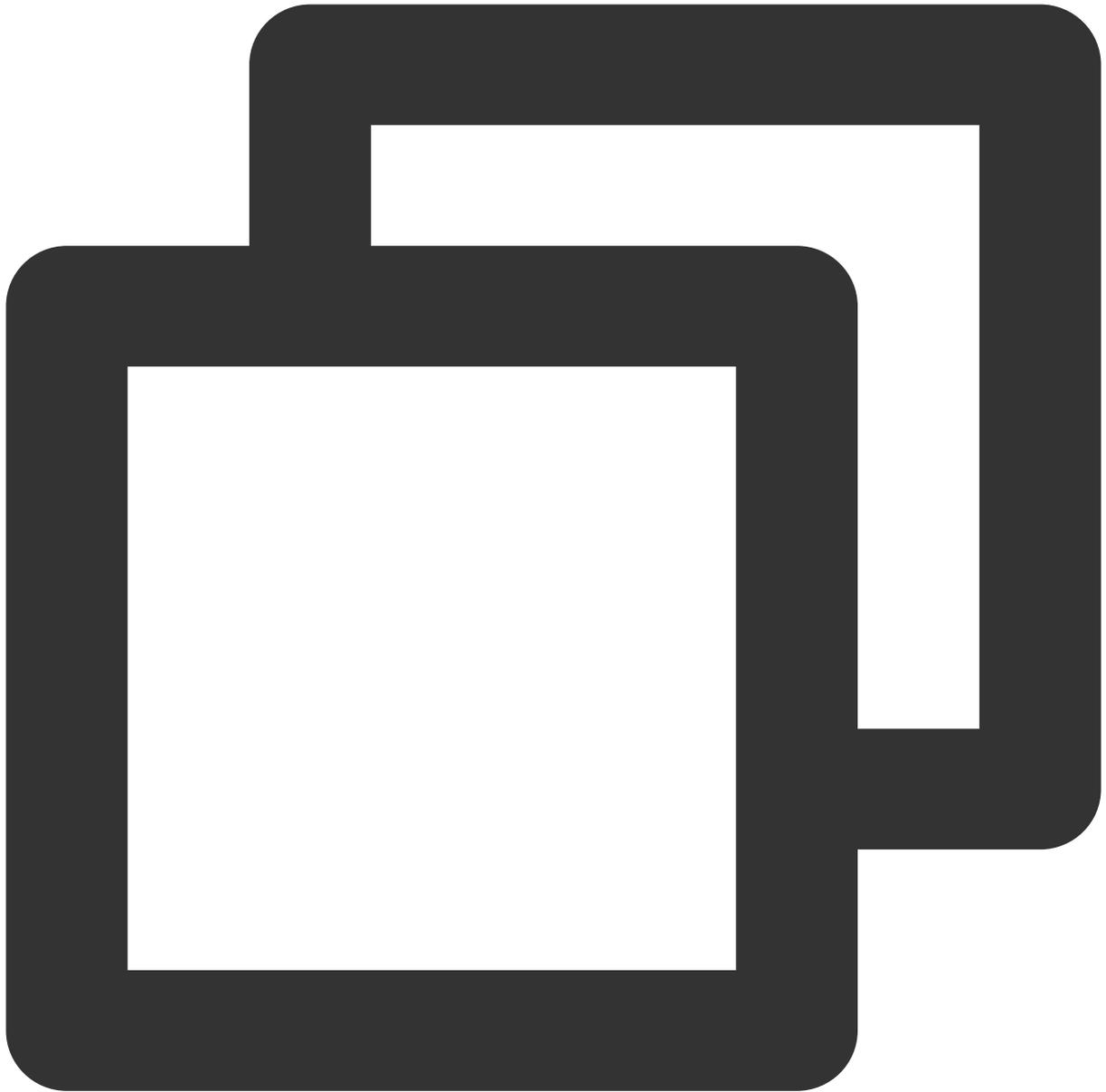
```
./coscli cp ~/test/ cos://bucket1/example/ -r --include ".*.txt" --meta=x-cos-met
```

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

操作示例

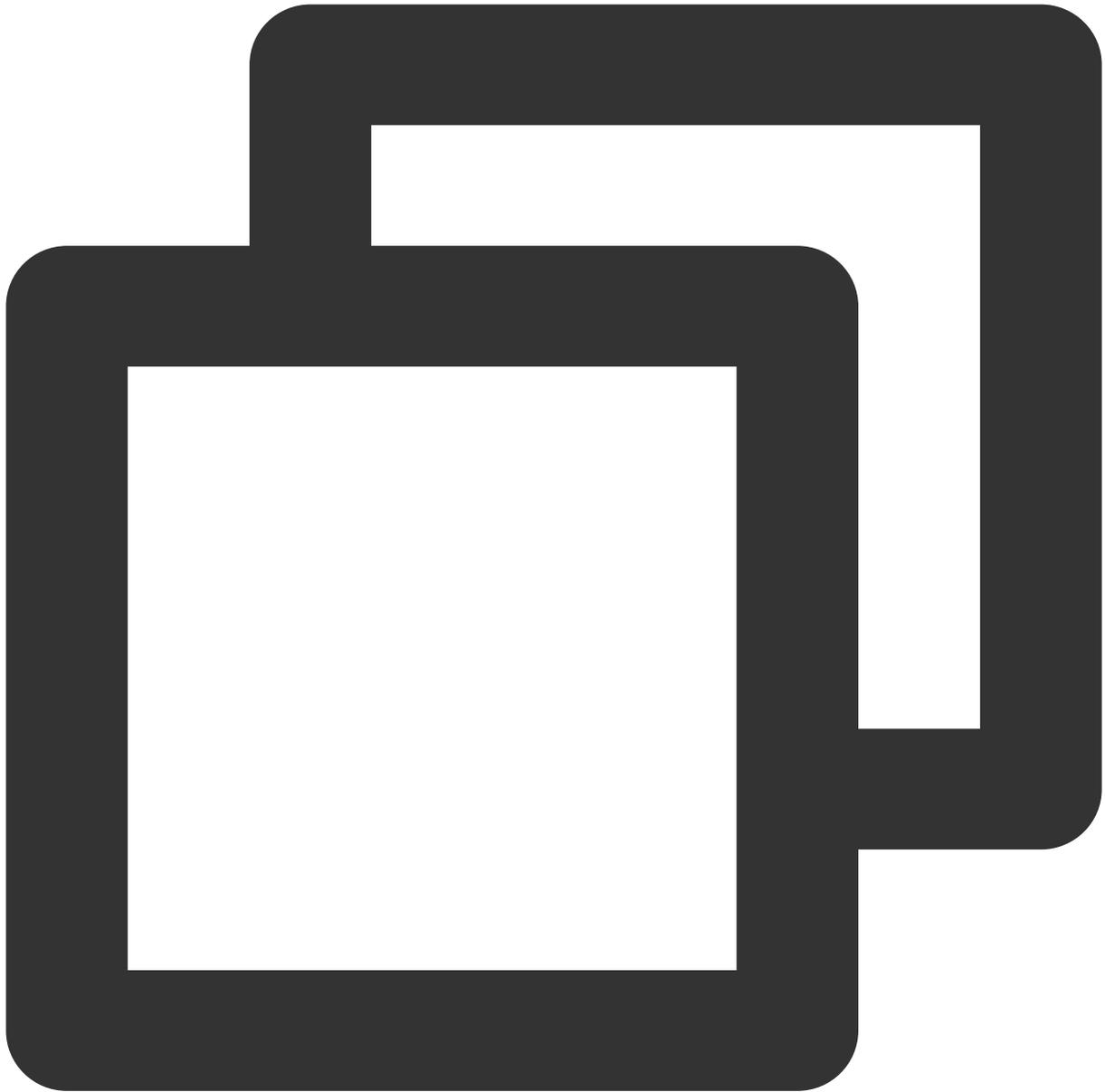
上传操作

上传单文件



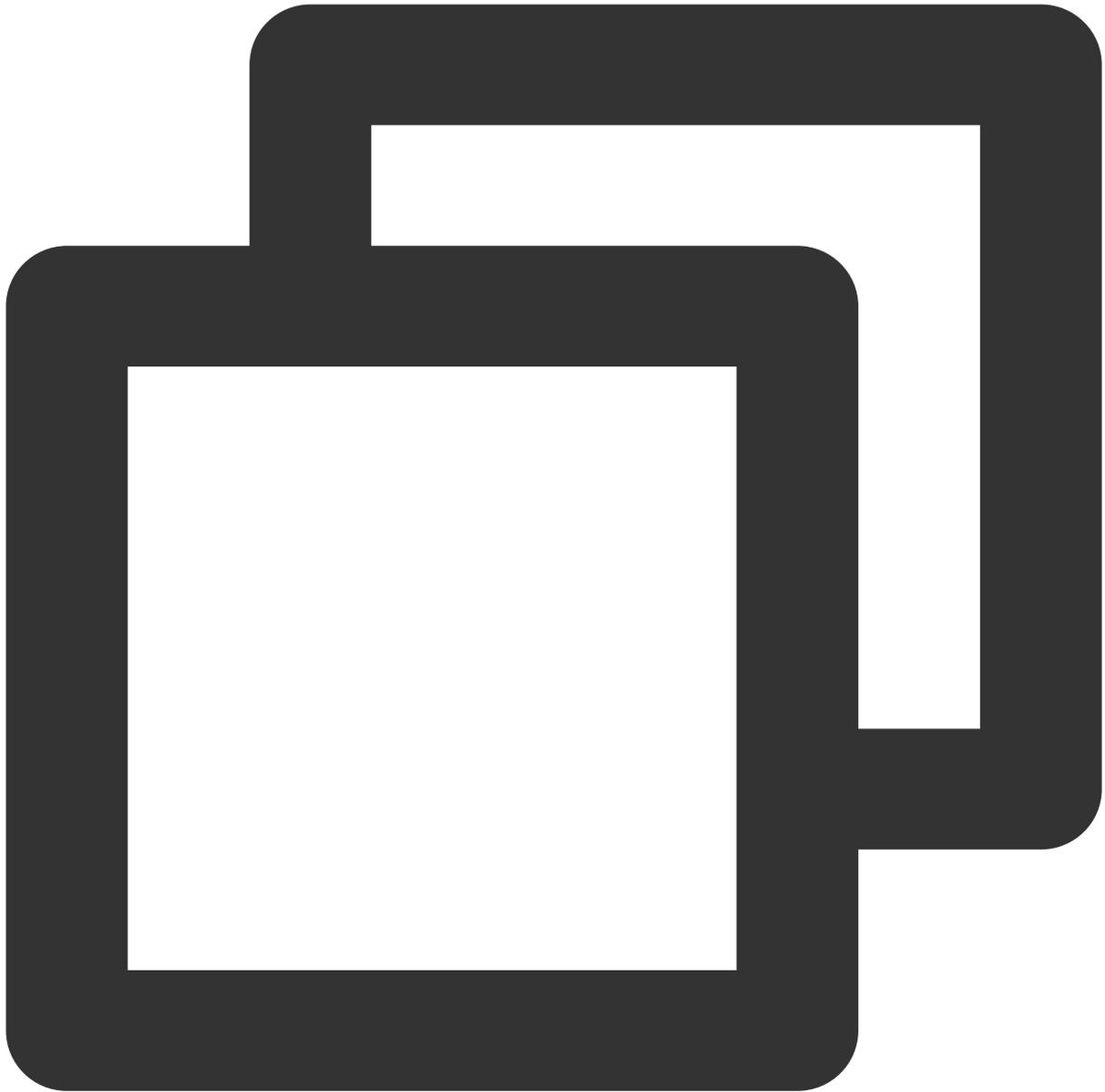
```
./coscli cp ~/example.txt cos://bucket1/example.txt
```

将本地 **test** 文件夹下的所有文件上传至 **bucket1** 桶中的 **example** 文件夹下



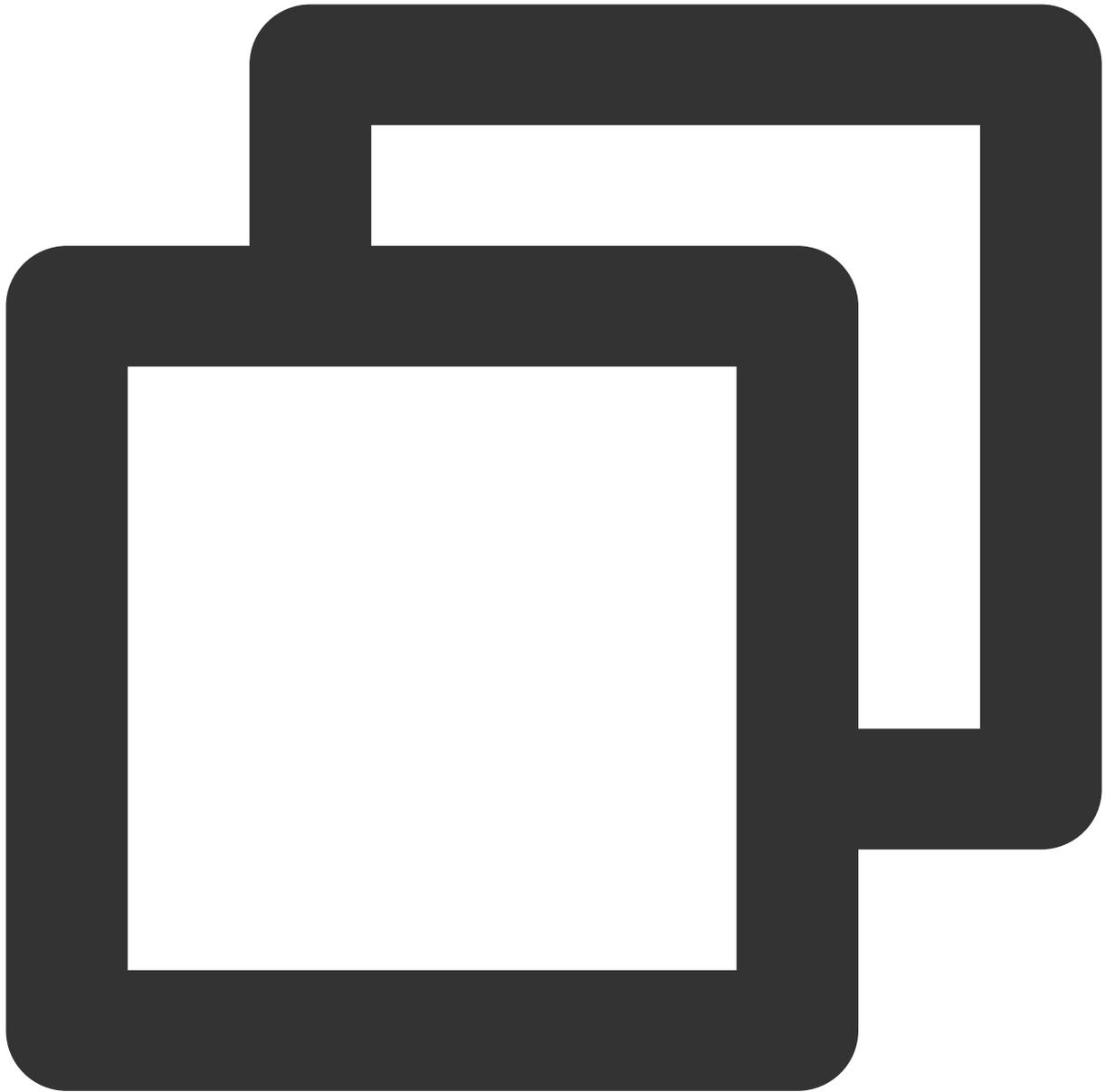
```
./coscli cp ~/test/ cos://bucket1/example/ -r
```

将本地 **test** 文件夹下的所有 **.mp4** 类型文件上传至 **bucket1** 桶中的 **example** 文件夹下



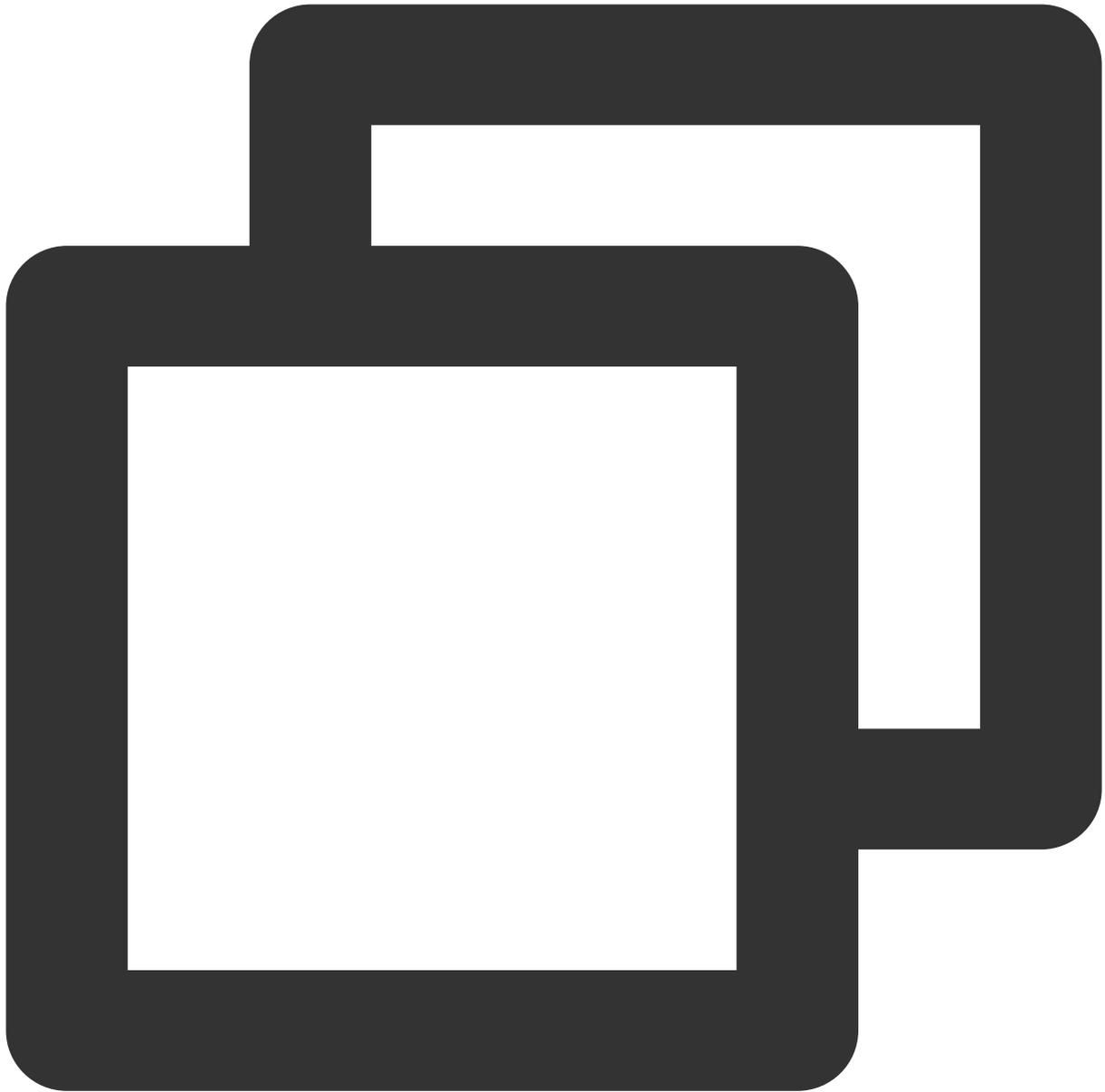
```
./coscli cp ~/test/ cos://bucket1/example/ -r --include *.mp4
```

将本地 **test** 文件夹下的所有非 **.md** 类型文件上传至 **bucket1** 桶中的 **example** 文件夹下



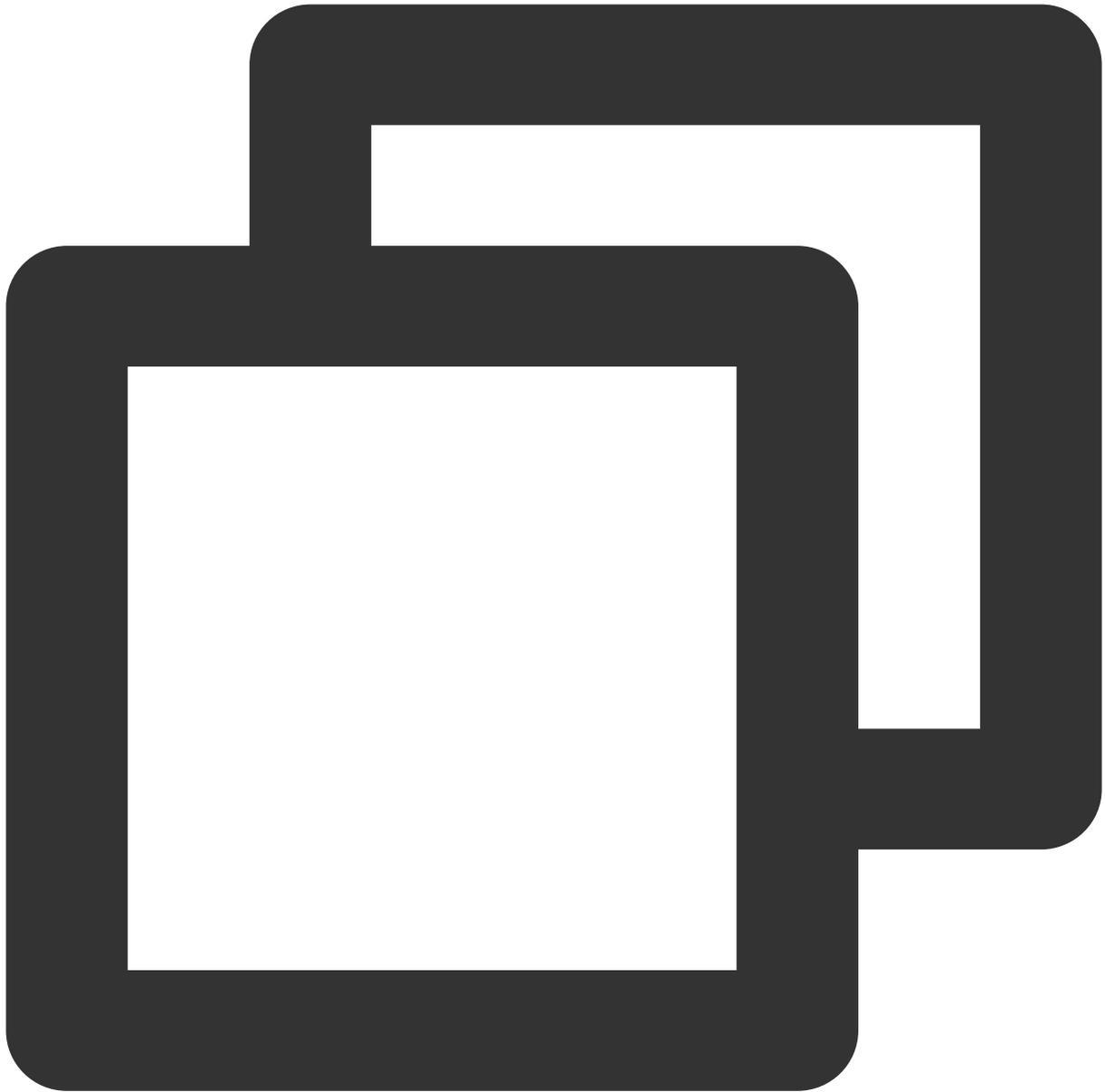
```
./coscli cp ~/test/ cos://bucket1/example/ -r --exclude *.*.md
```

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



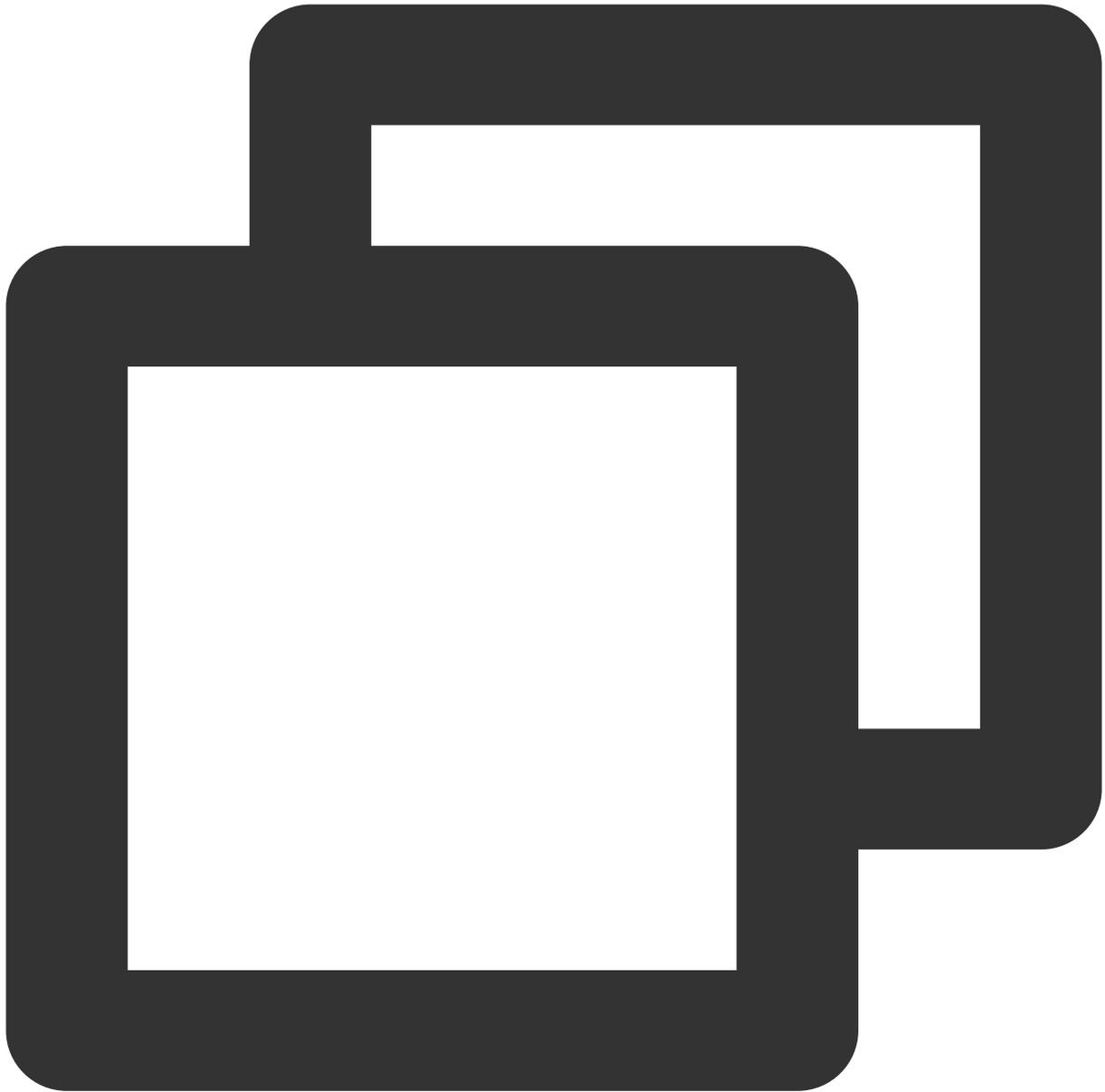
```
./coscli cp dir/ cos://bucket1/example/ -r --exclude dirD/*
```

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



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

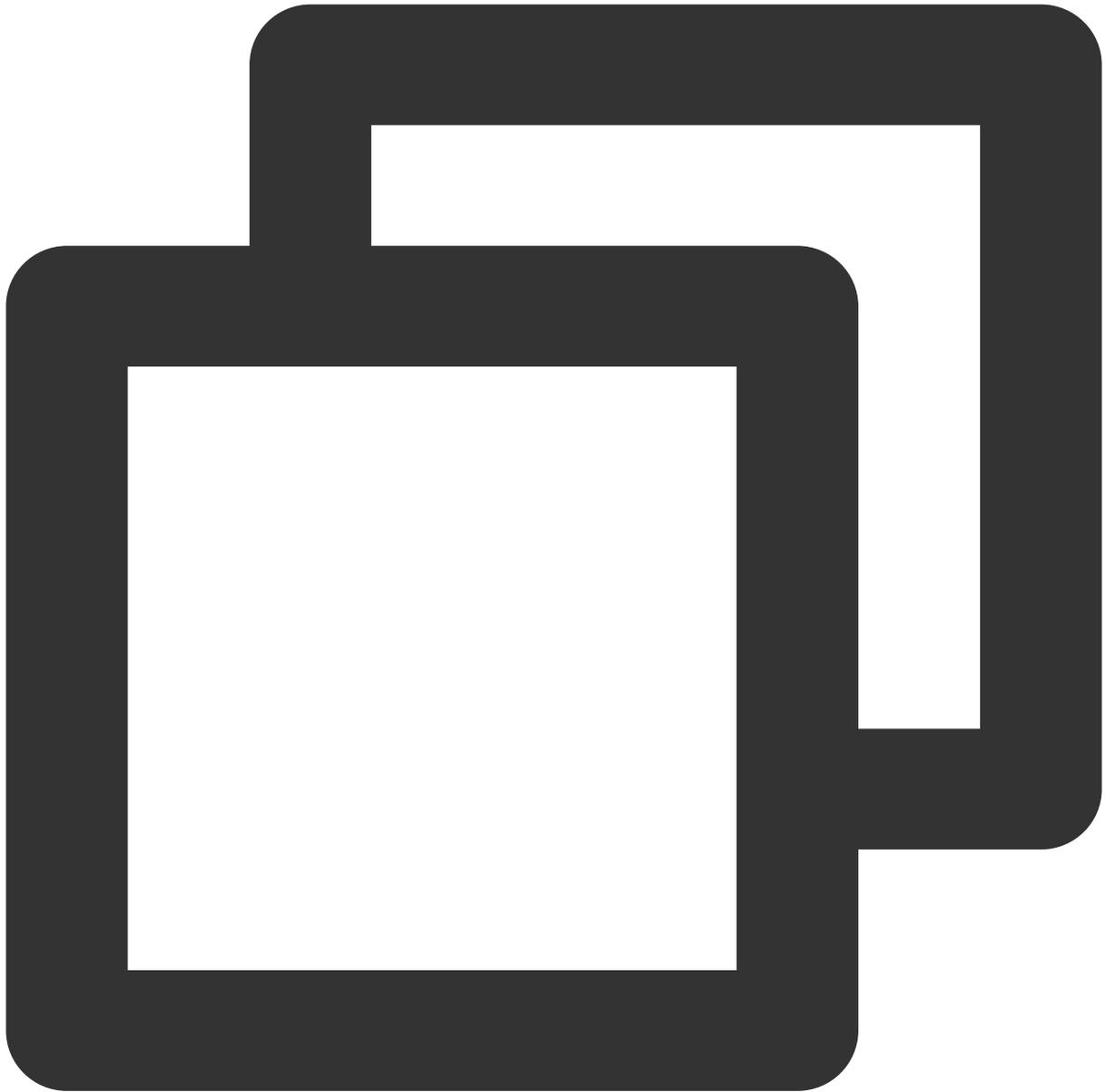
将本地的 **file.txt** 文件上传至 **bucket1** 桶并设置单链接速率限制为**1.3MB/s**



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

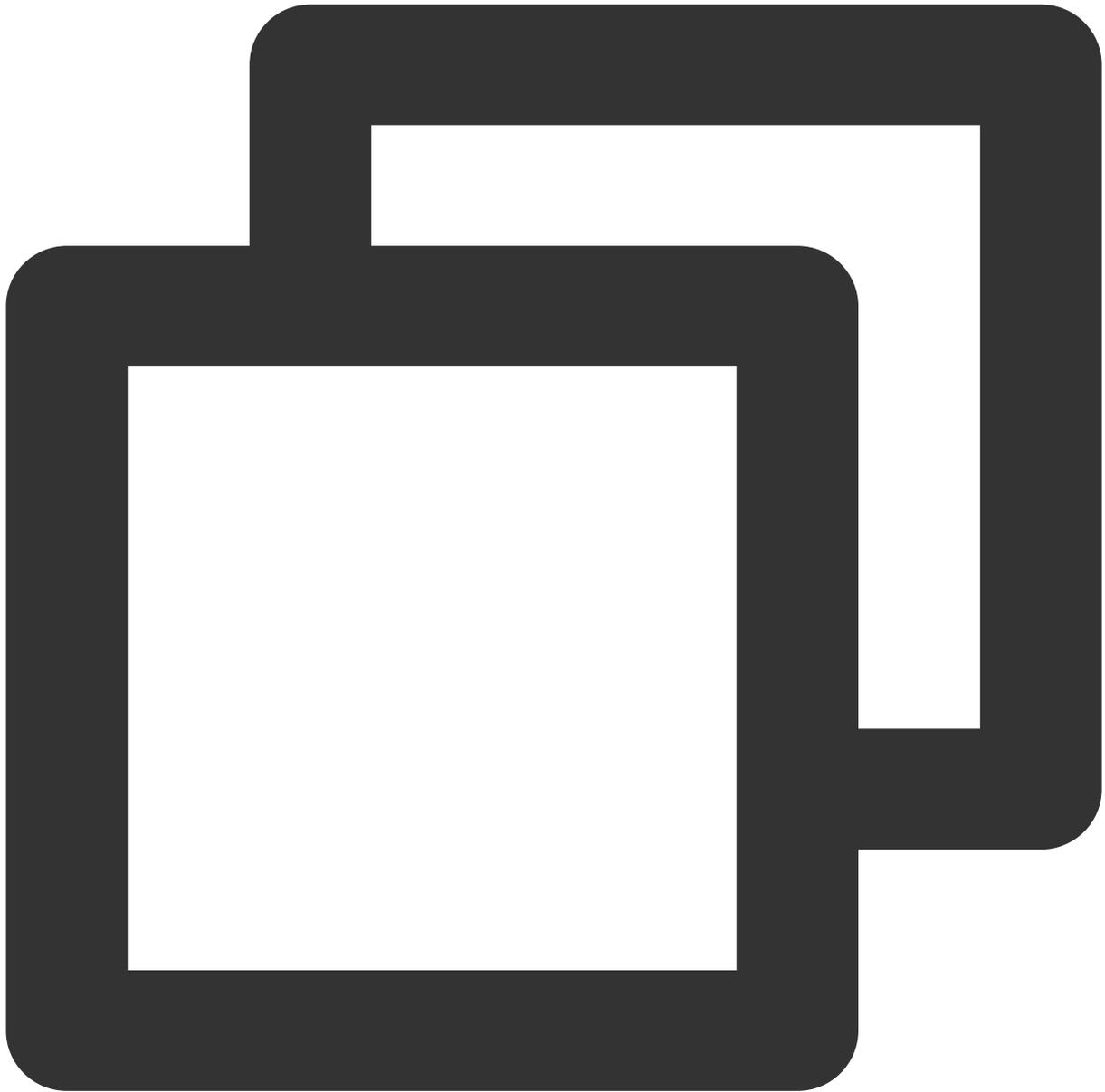
下载操作

下载单文件



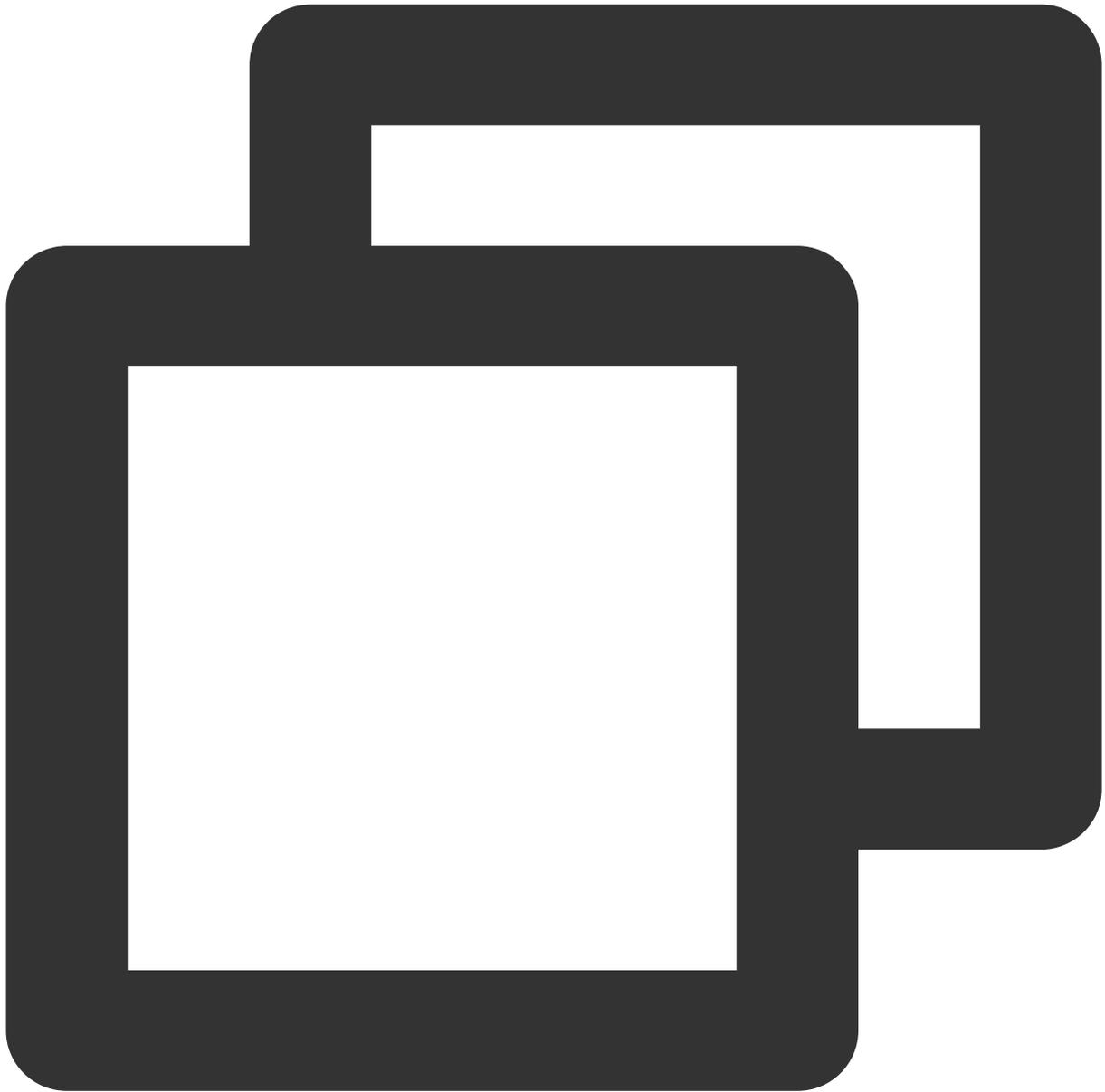
```
./coscli cp cos://bucket1/example.txt ~/example.txt
```

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



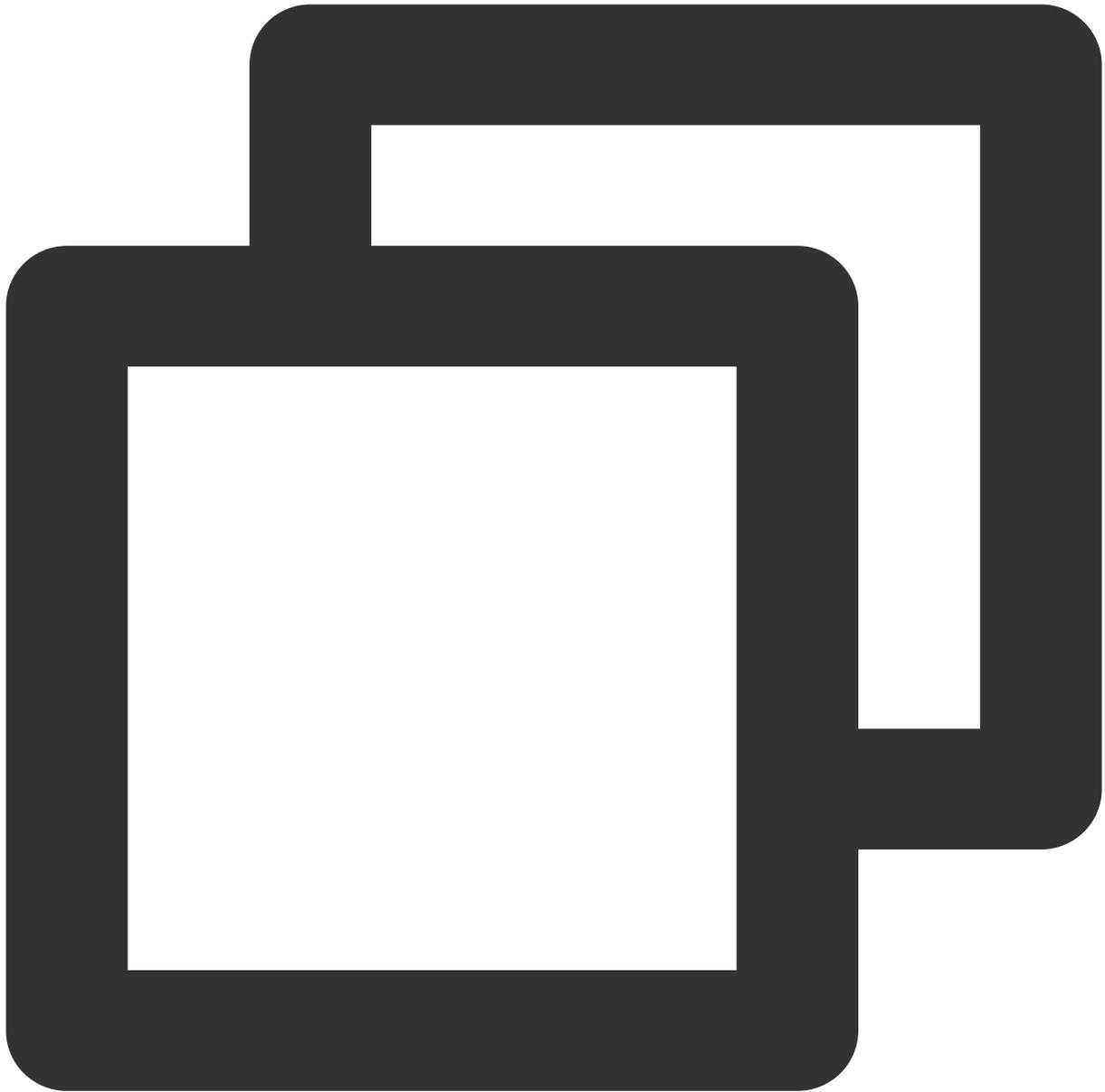
```
./coscli cp cos://bucket1/example/ ~/test/ -r
```

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



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

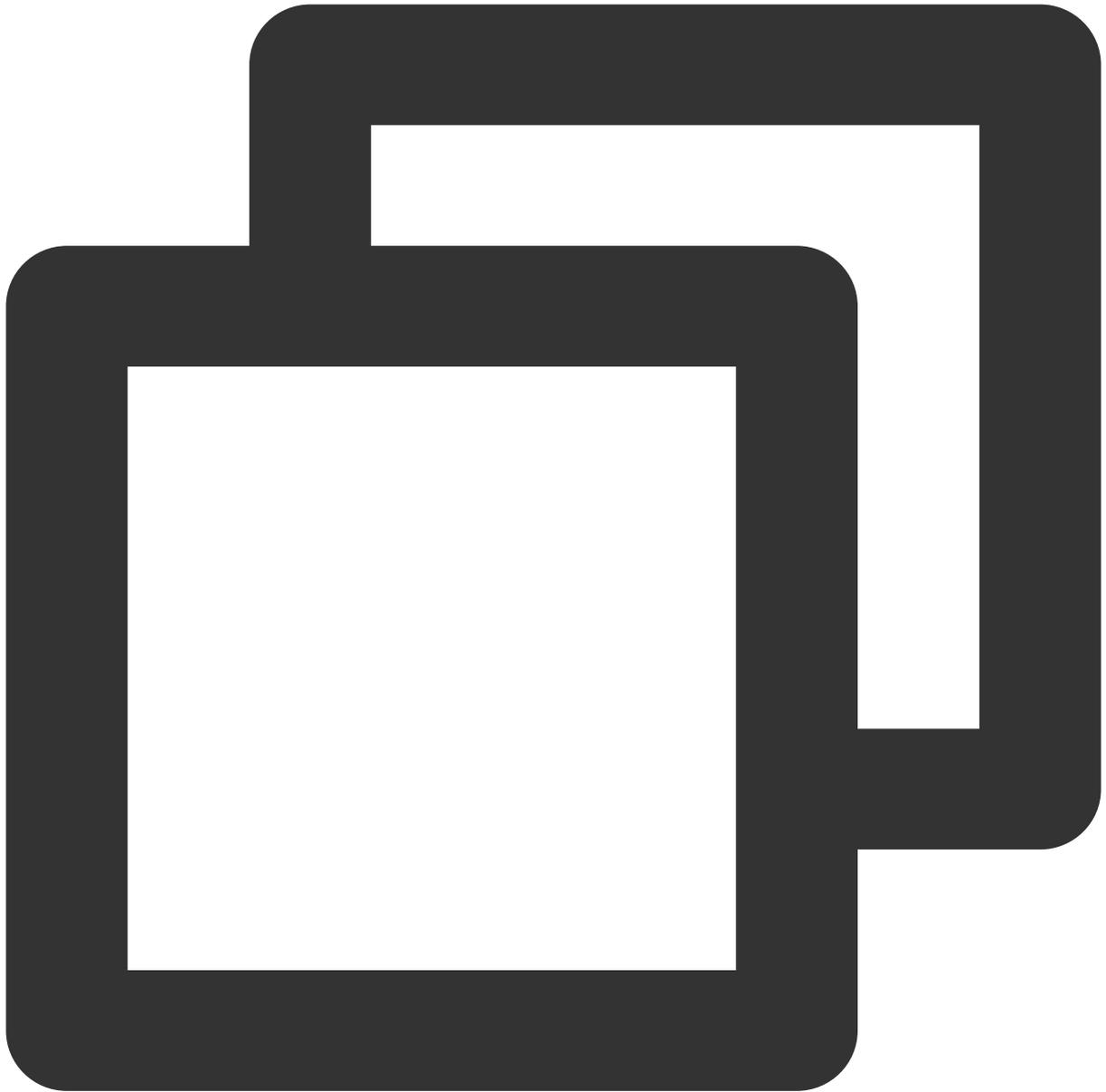
将 **bucket1** 桶中的 **example** 文件夹下所有非 **.md** 类型文件下载至本地 **test** 文件夹



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

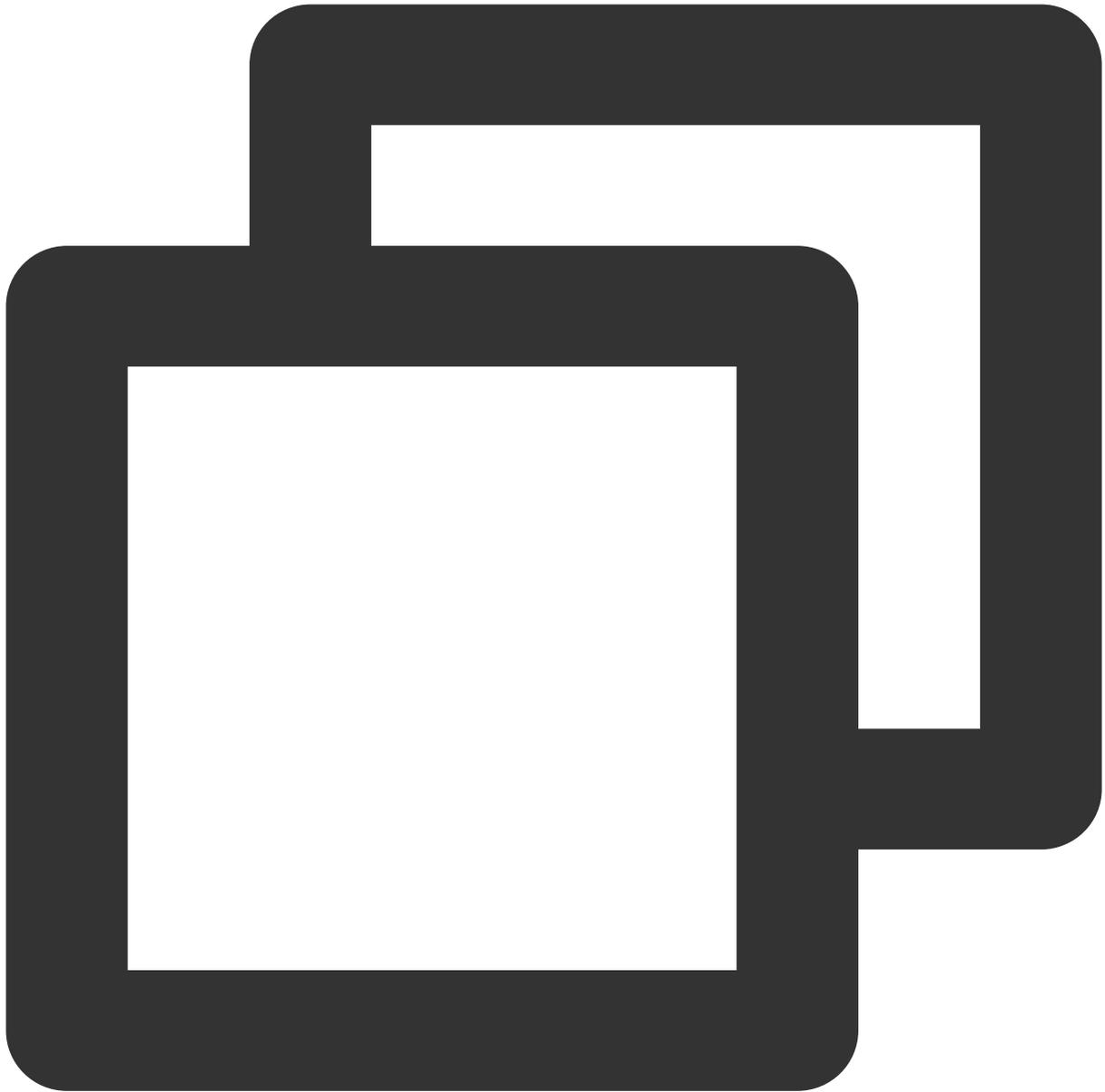
拷贝操作

桶内拷贝单文件



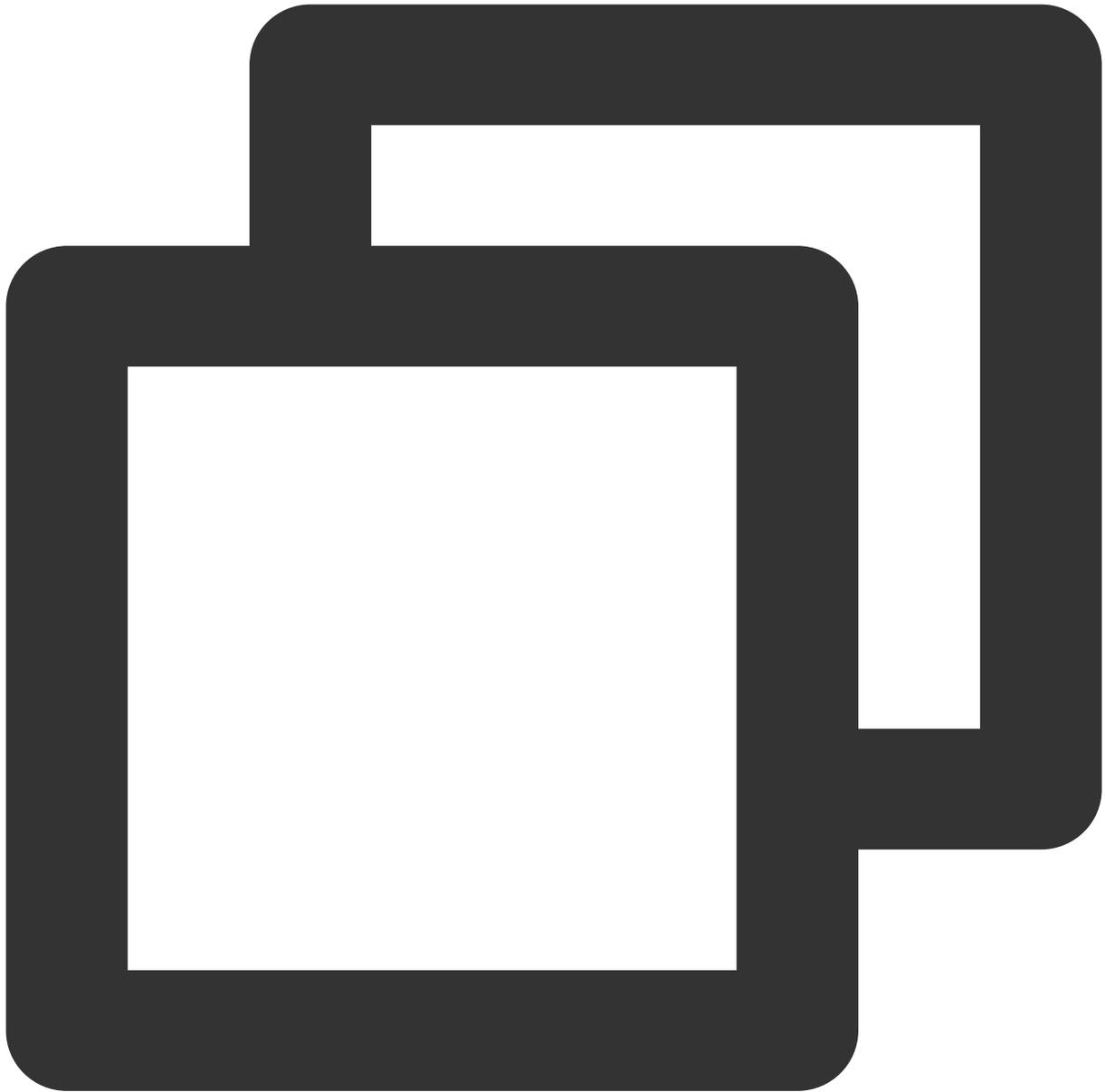
```
./coscli cp cos://bucket1/example.txt cos://bucket1/example_copy.txt
```

跨桶拷贝单文件



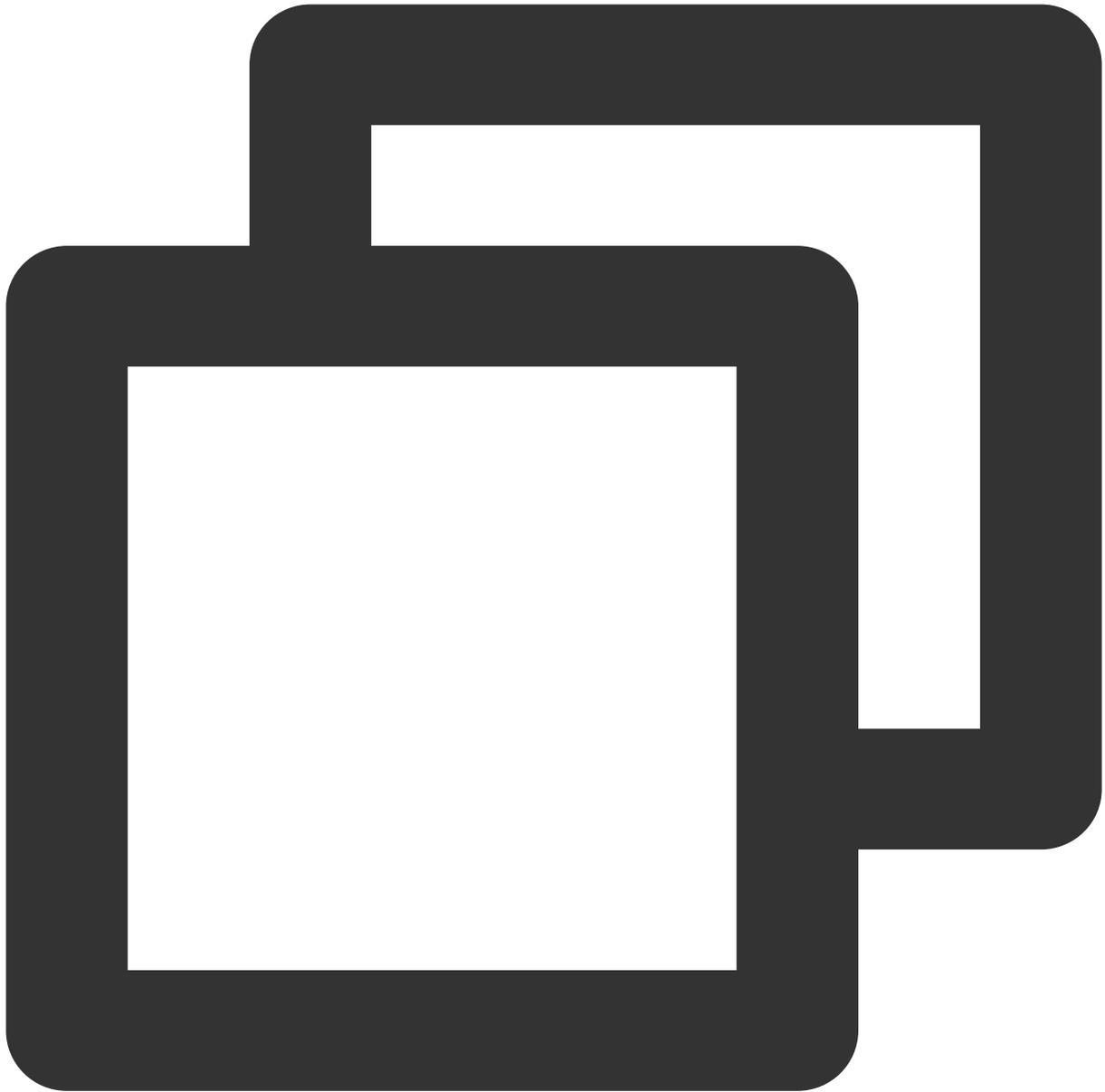
```
./coscli cp cos://bucket1/example.txt cos://bucket2/example_copy.txt
```

将 **bucket1** 桶中 **example1** 文件夹下的所有文件拷贝至 **bucket2** 桶中 **example2** 文件夹下



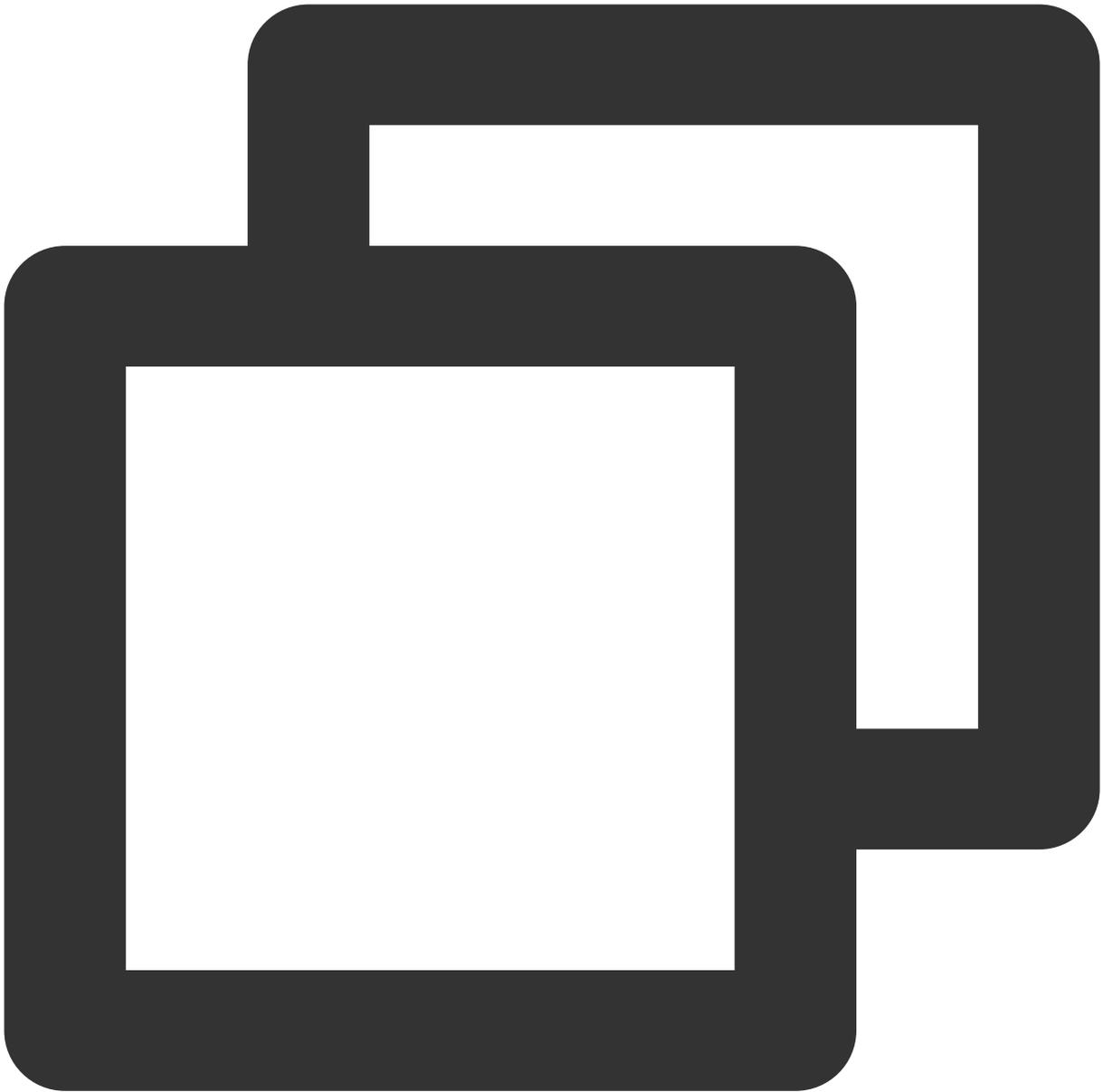
```
./coscli cp cos://bucket1/example1/ cos://bucket2/example2/ -r
```

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



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

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



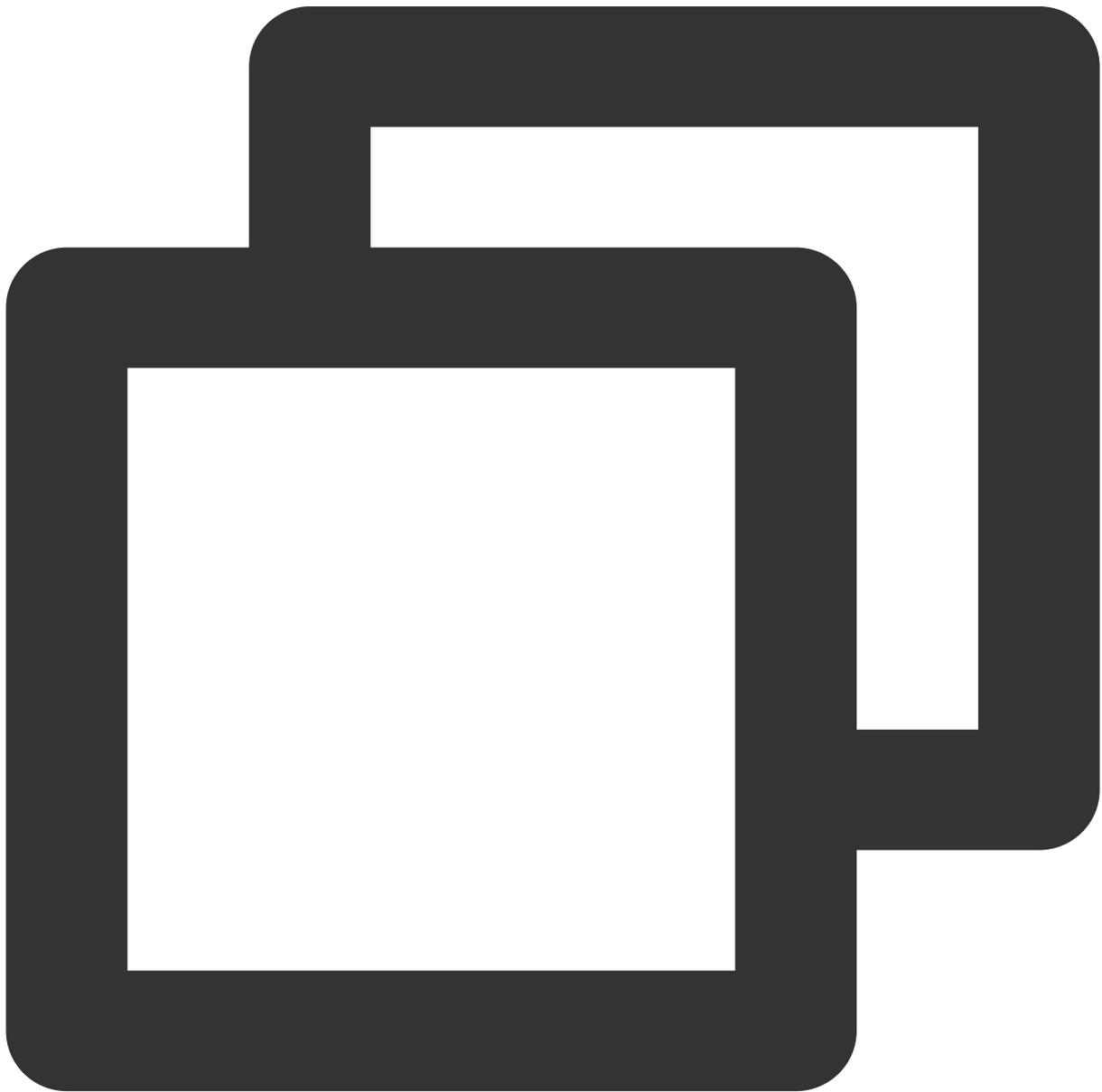
```
./coscli cp cos://bucket1/example1/ cos://bucket2/example2/ -r --exclude .*md
```

同步上传下载或拷贝文件 - sync

最近更新时间：2024-06-28 16:47:42

命令格式

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



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

cp 命令包含以下参数：

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

sync 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件
-r	--recursive	是否递归遍历文件夹下所有文件
无	--storage-class	指定上传文件的存储类型（默认 STANDARD），更多存储类型，请参见 存储类型概述
无	--part-size	文件分块大小（默认32MB，最大支持5GB）
无	--thread-num	并发线程数（默认并发5）
无	--rate-limiting	单链接速率限制（0.1 - 100MB/s）
无	--snapshot-path	指定保存上传或者下载文件时的快照信息所在的目录。在下次上传或者下载文件时， <code>coscli</code> 会读取指定目录下的快照信息进行增量上传或者下载。本选项用来加速目录文件同步。
无	--meta	上传文件的元信息。包括部分 HTTP 标准属性（HTTP Header）以及以 <code>x-cos-meta-</code> 开头的用户自定义元数据（User Meta）。文件元信息格式为 <code>header:value#header:value</code> ，示例为 <code>Expires:2022-10-</code>

		12T00:00:00.000Z#Cache-Control:no-cache#Content-Encoding:gzip#x-cos-meta-x:x
无	--routines	指定文件间并发上传或下载线程的文件数量，默认为 3
无	--fail-output	此选项决定是否启用上传或下载失败时文件的错误输出（默认为 true，开启）。如果启用，文件传输失败将被记录在指定目录内（如果没有指定，默认为 ./coscli_output）。如果禁用，只会将错误文件的数量输出到控制台。
无	--fail-output-path	此选项用于指定记录上传或下载失败文件的错误输出文件夹。通过提供自定义文件夹路径，您可以控制错误输出文件夹的位置和名称。如果未设置此选项，将使用默认的错误日志文件夹 ./coscli_output。
无	--retry-num	限频重试次数（默认为 0，不重试）。可选 1-10 次。当多台机器同时在一个COS目录上执行下载操作时，可以通过指定此参数进行重试来规避限频错误。
无	--err-retry-num	错误重试次数（默认为 0）。指定 1-10 次，或者设置为 0 表示不进行重试。
无	--err-retry-interval	重试间隔（仅在指定 --err-retry-num 为 1-10 时可用）。指定 1-10 秒的重试间隔，如果不指定或设置为 0，每次重试间隔将在 1-10 秒内的随机。
无	--only-current-dir	是否仅上传当前目录中的文件，忽略子目录及其内容（默认为 false，不忽略）
无	--disable-all-symlink	是否上传时忽略所有软链接子文件和软链接子目录（默认为 true，不上传）
无	--enable-symlink-dir	是否上传软链接的子目录（默认为 false，不上传）
无	--disable-crc64	是否关闭CRC64数据验证。（默认为 false，启用验证）
无	--delete	删除指定目标路径中的任何其他文件，仅保留此次同步的文件（默认为 false，不删除）。建议在使用--delete选项之前启用版本控制，以防止意外删除数据。
无	--backup-dir	同步已删除文件的备份，用于保存目标端已删除但源端不存在的文件(仅下载时生效，且在 --delete=true 时必传)。上传及桶拷贝请使用版本控制来恢复意外删除数据。
无	--force	强制操作，不提示确认。（默认为 false）

说明

`sync` 命令在上传和下载大文件时会自动启用并发上传/下载。

当文件大于 `--part-size` 时，COSCLI 会先将文件按 `--part-size` 进行切块，之后用 `--thread-num` 个线程并发地执行上传/下载任务。

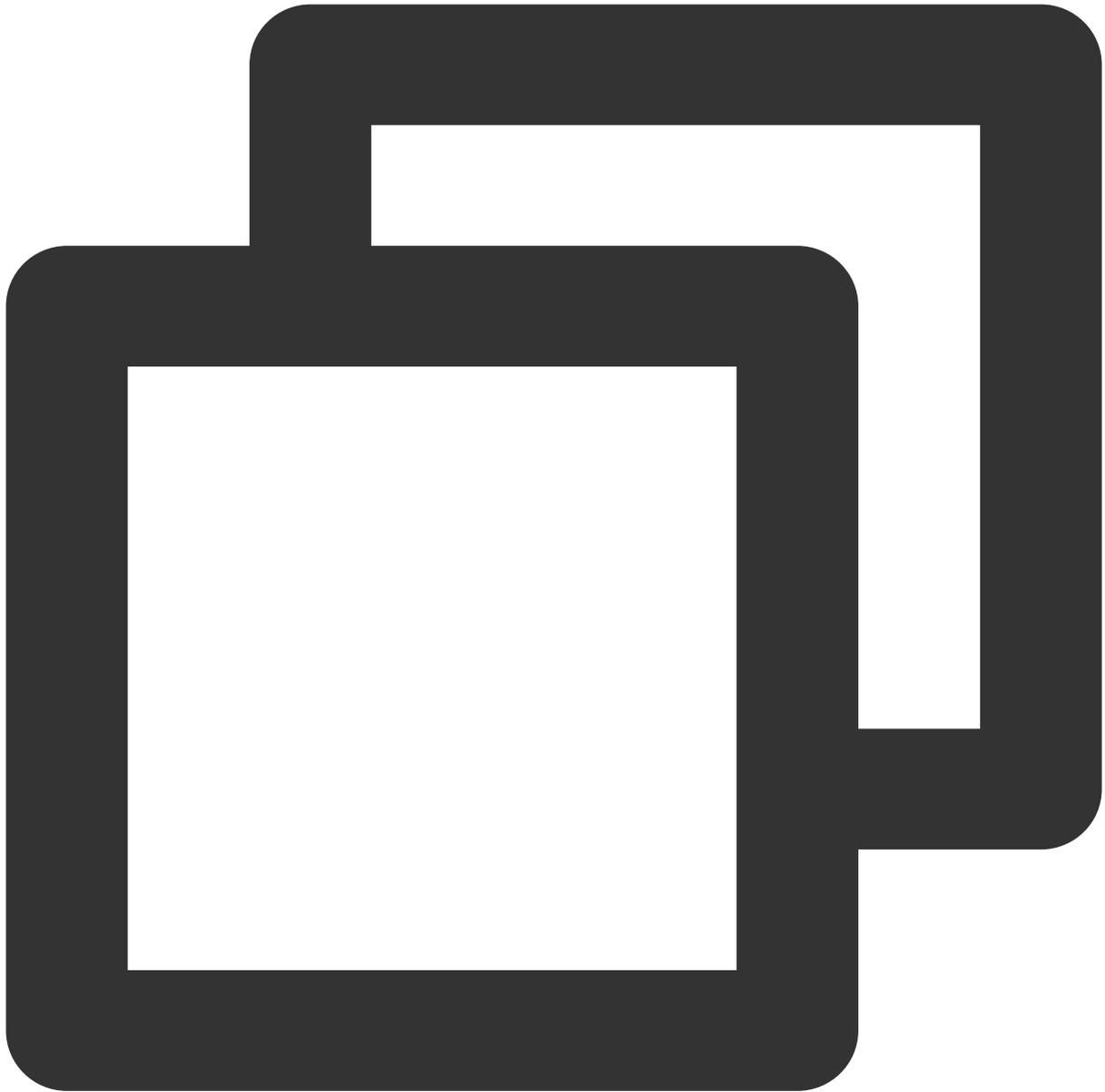
每个线程都会维护一个链接，对于每个链接，您可以使用 `--rate-limiting` 参数对单链接进行限速，当启用并发上传/下载时，总速率为 `--thread-num * --rate-limiting`。

当以分块形式上传/下载文件时，会默认开启断点续传。

`--include` 和 `--exclude` 支持标准正则表达式的语法，您可以使用它来过滤出符合特定条件的文件。

使用 `zsh` 时，您可能需要在 `pattern` 串的两端加上双引号。

`snapshot-path` 不要设置到为待迁移目录或其子目录。coscli 不会主动删除 `snapshot-path` 文件夹中的快照信息。为避免快照信息过多，请定期删除 `snapshot-path` 文件夹内无用的快照。

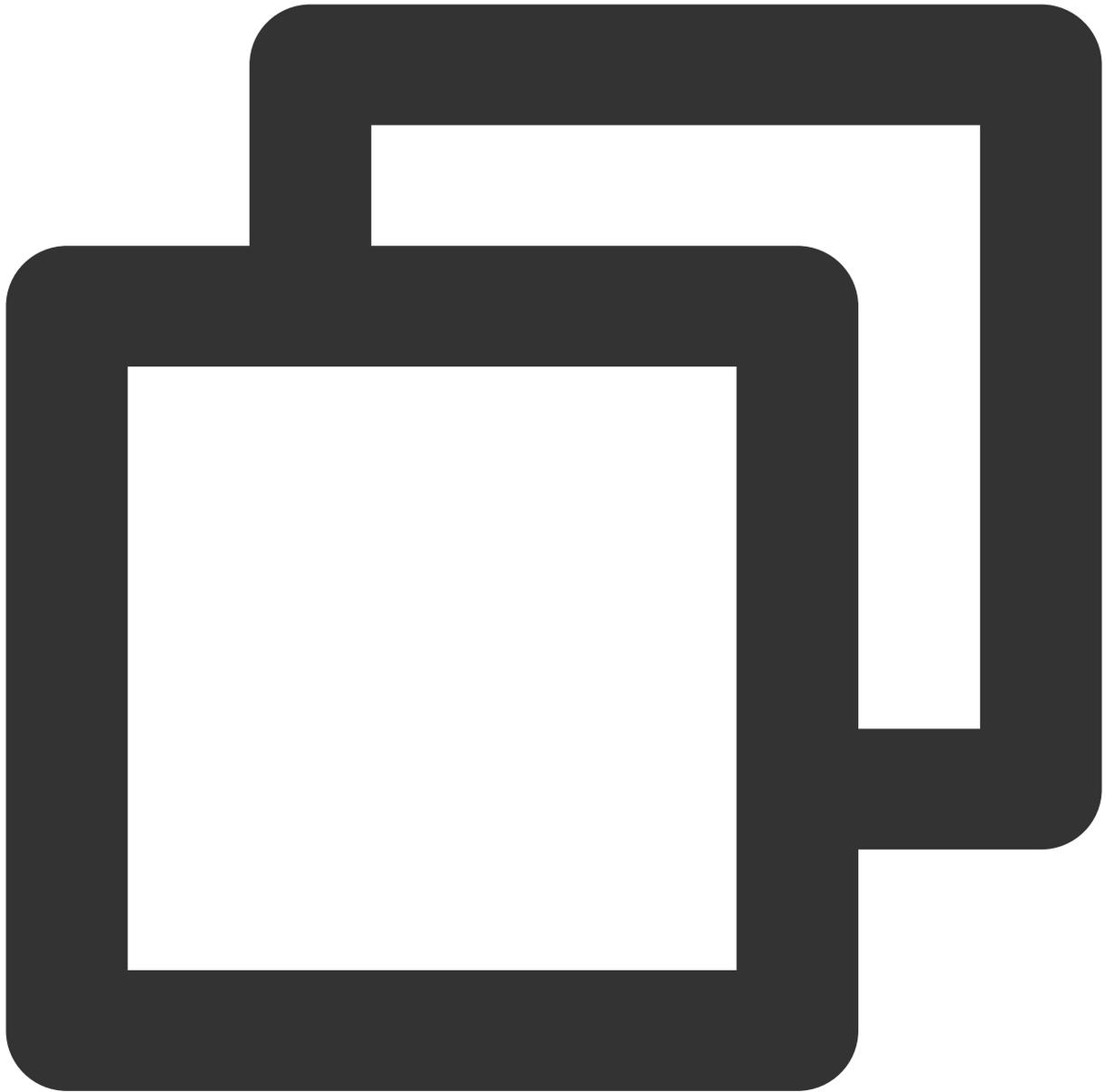


```
./coscli sync ~/test/ cos://bucket1/example/ -r --include ".*.txt" --snapshot-p
```

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

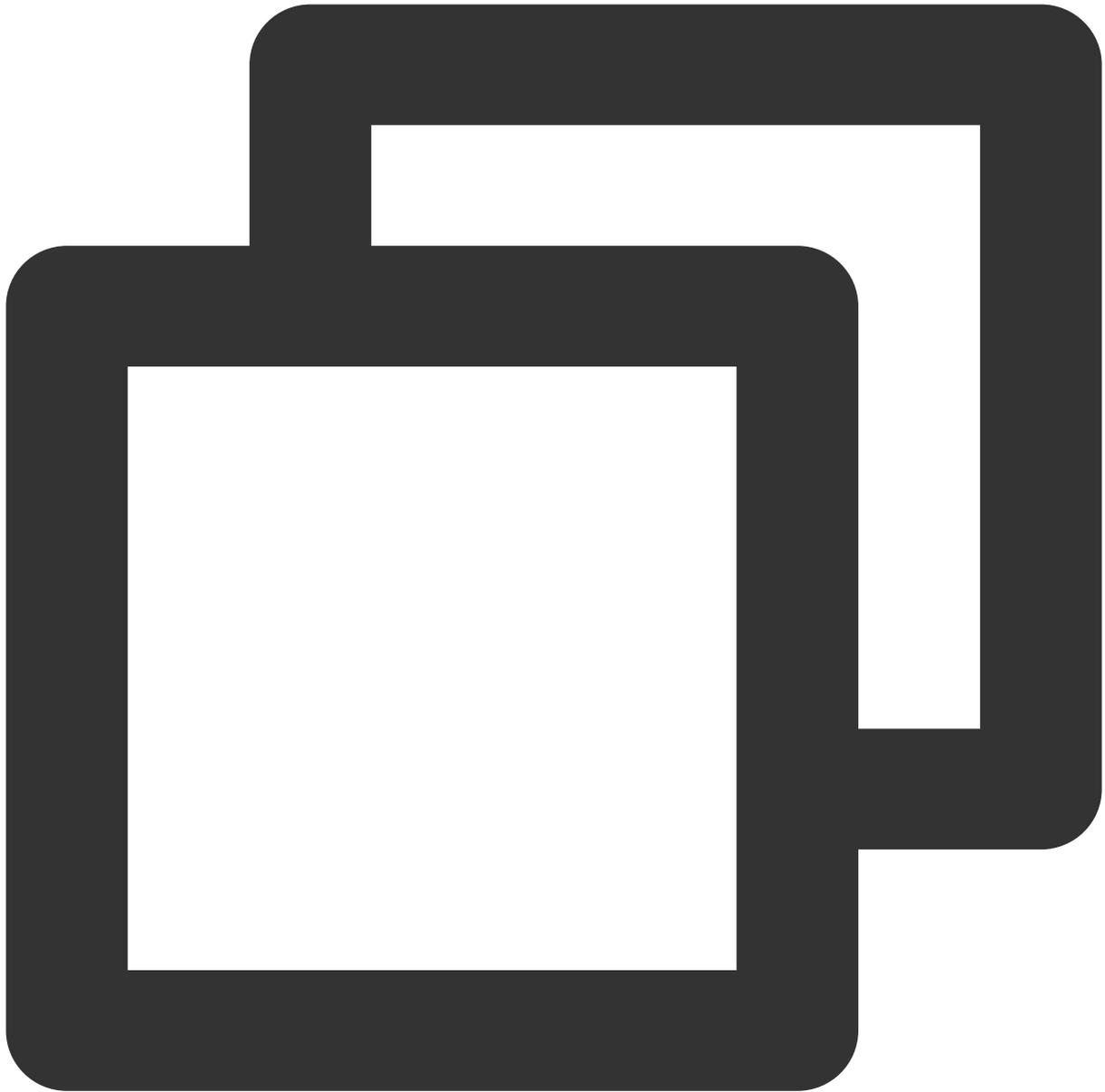
操作示例

同步上传文件



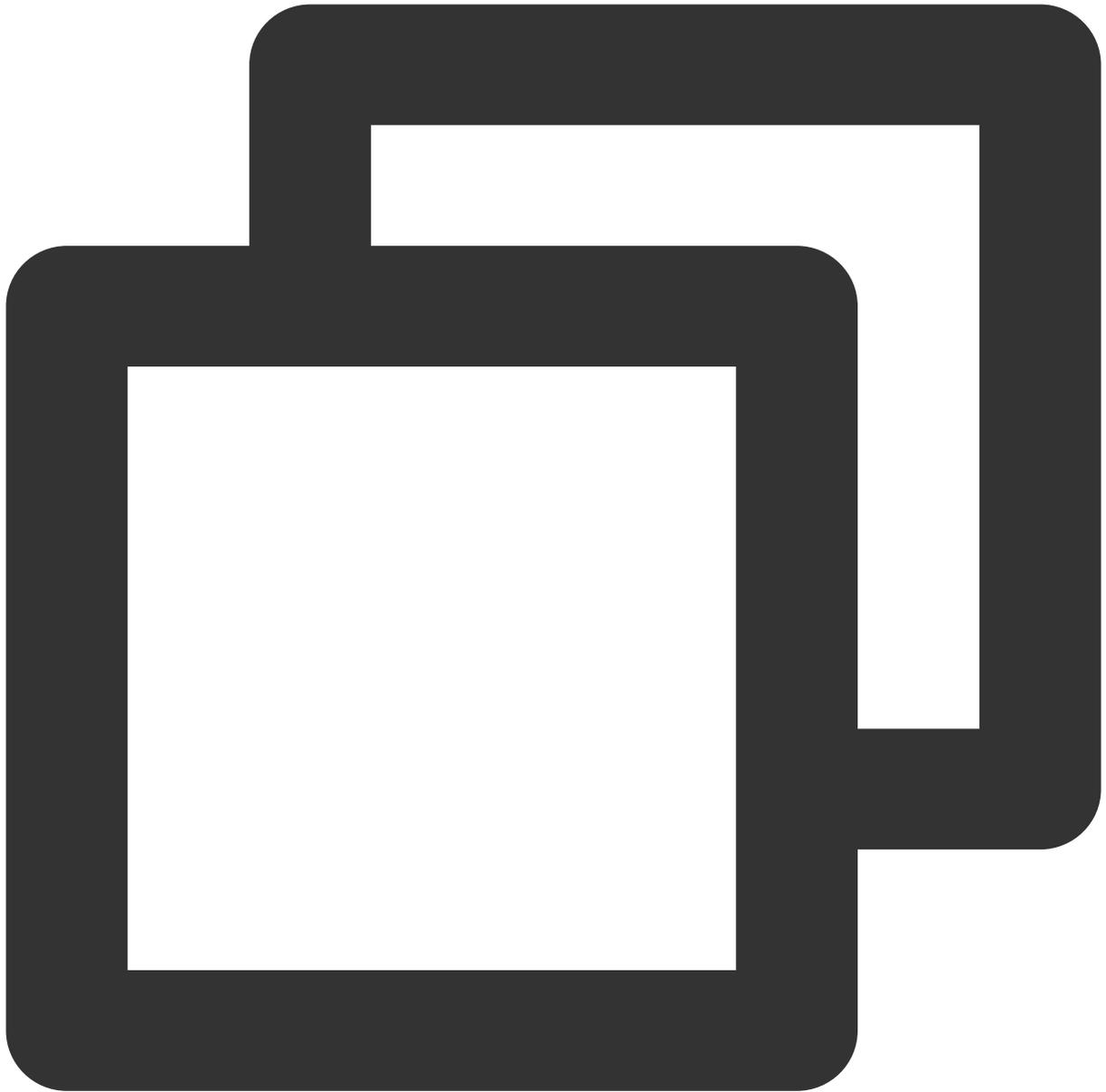
```
./coscli sync ~/example.txt cos://bucket1/example.txt
```

同步下载文件



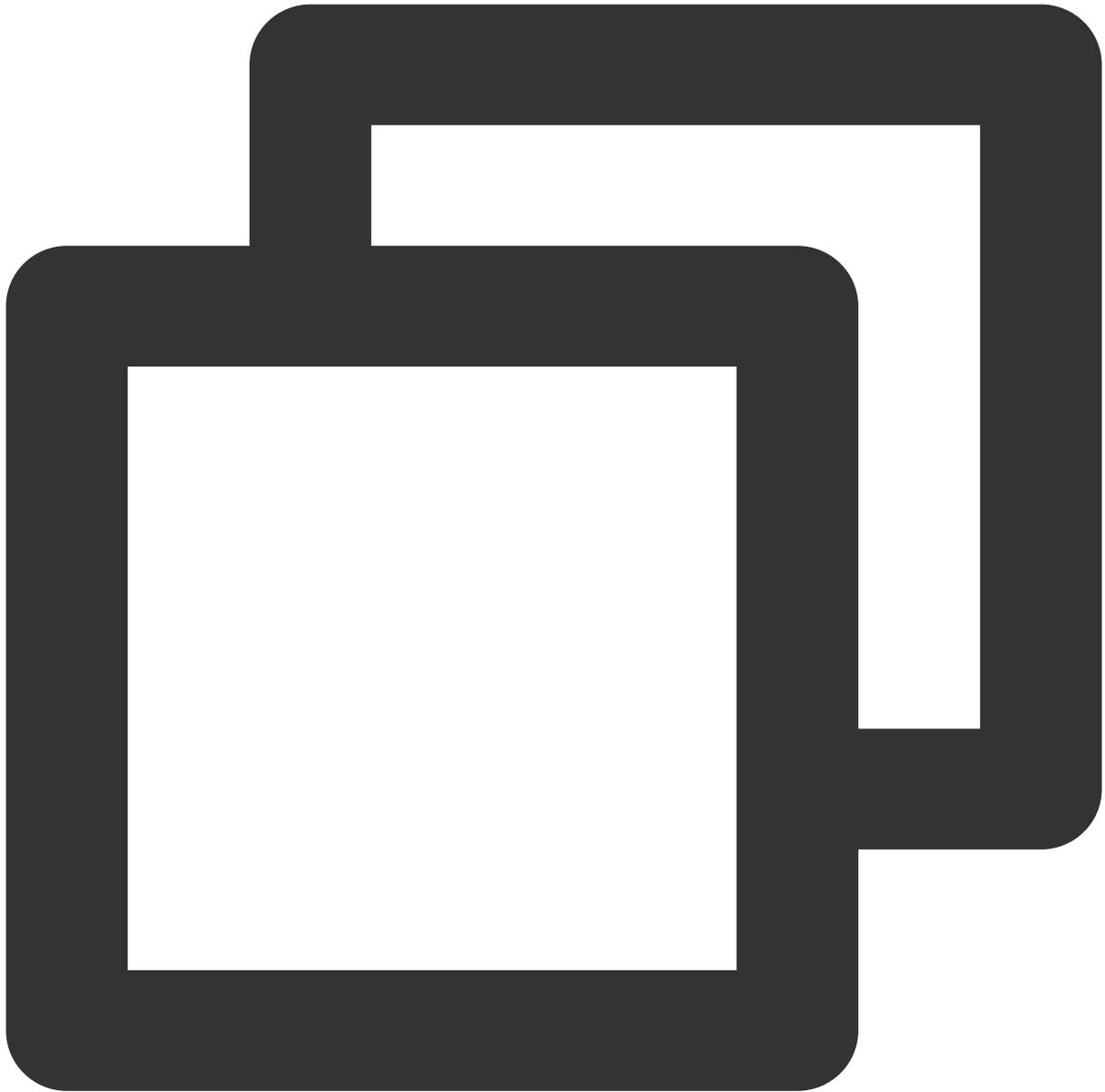
```
./coscli sync cos://bucket1/example.txt ~/example.txt
```

桶内同步拷贝文件



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

跨桶同步拷贝文件



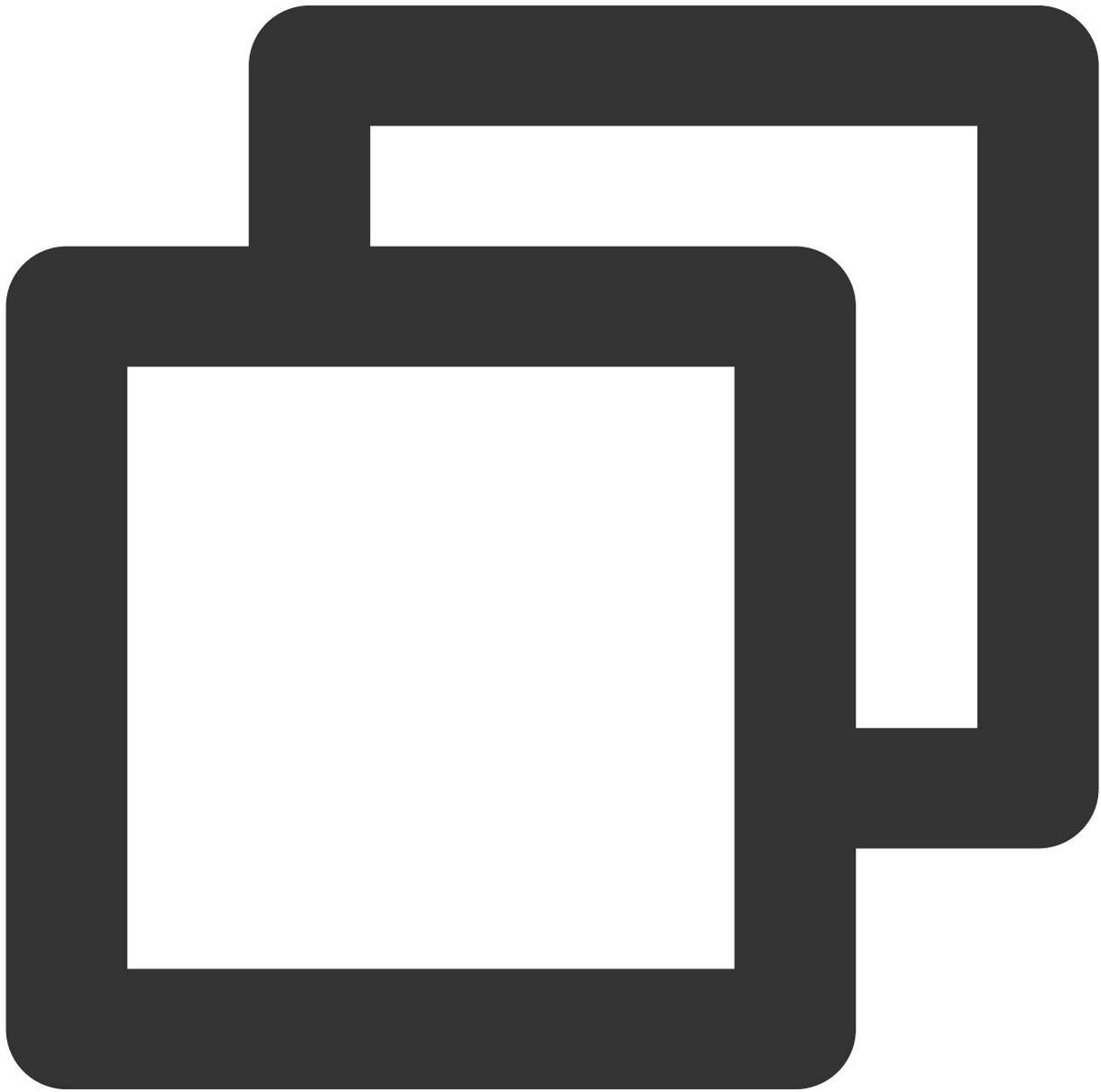
```
./coscli sync cos://bucket1/example.txt cos://bucket2/example_copy.txt
```

删除文件 - rm

最近更新时间：2024-06-28 16:49:26

rm 命令用于删除文件。

命令格式



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

rm 命令包含以下参数：

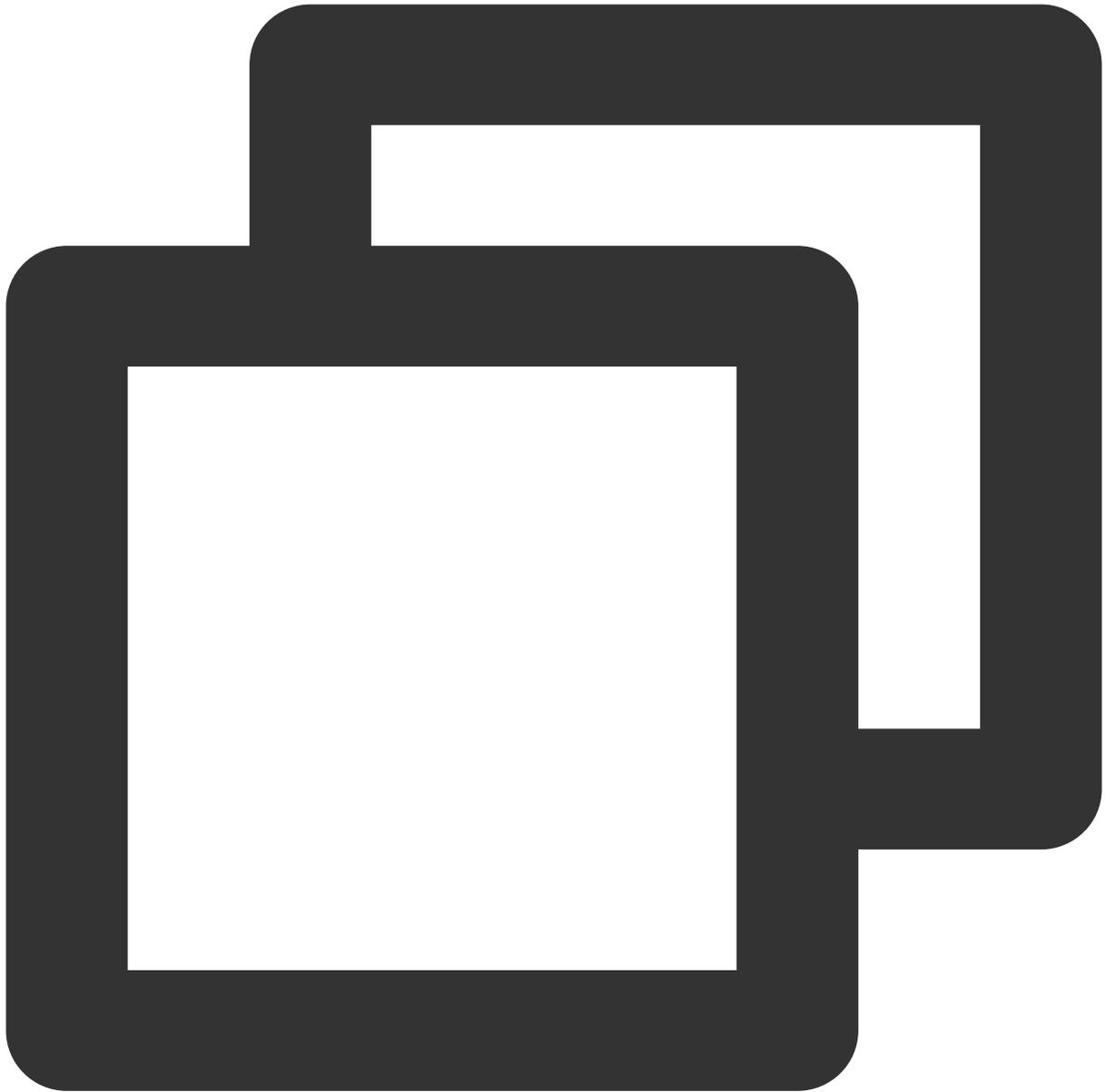
参数格式	参数用途	示例
cos://<bucket-name>	指定需要访问的存储桶。支持使用 配置参数 中的桶别名，或桶名称进行访问。如使用桶名称访问，需要额外携带 <code>endpoint</code> <code>flag</code> 。	使用桶别名访问： cos://example-alias 使用桶名称访问： cos://examplebucket-1250000000
/prefix/	可选参数。指定某一文件夹	/picture/

rm 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件
-r	--recursive	是否递归地遍历文件夹下所有文件
-f	--force	强制删除（删除文件前不弹出确认信息）
无	--fail-output	此选项决定是否启用上传或下载失败时文件的错误输出（默认为 <code>true</code> ，开启）。如果启用，文件传输失败将被记录在指定目录内（如果没有指定，默认为 <code>./coscli_output</code> ）。如果禁用，只会将错误文件的数量输出到控制台。
无	--fail-output-path	此选项用于指定记录上传或下载失败文件的错误输出文件夹。通过提供自定义文件夹路径，您可以控制错误输出文件夹的位置和名称。如果未设置此选项，将使用默认的错误日志文件夹 <code>./coscli_output</code> 。

说明

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

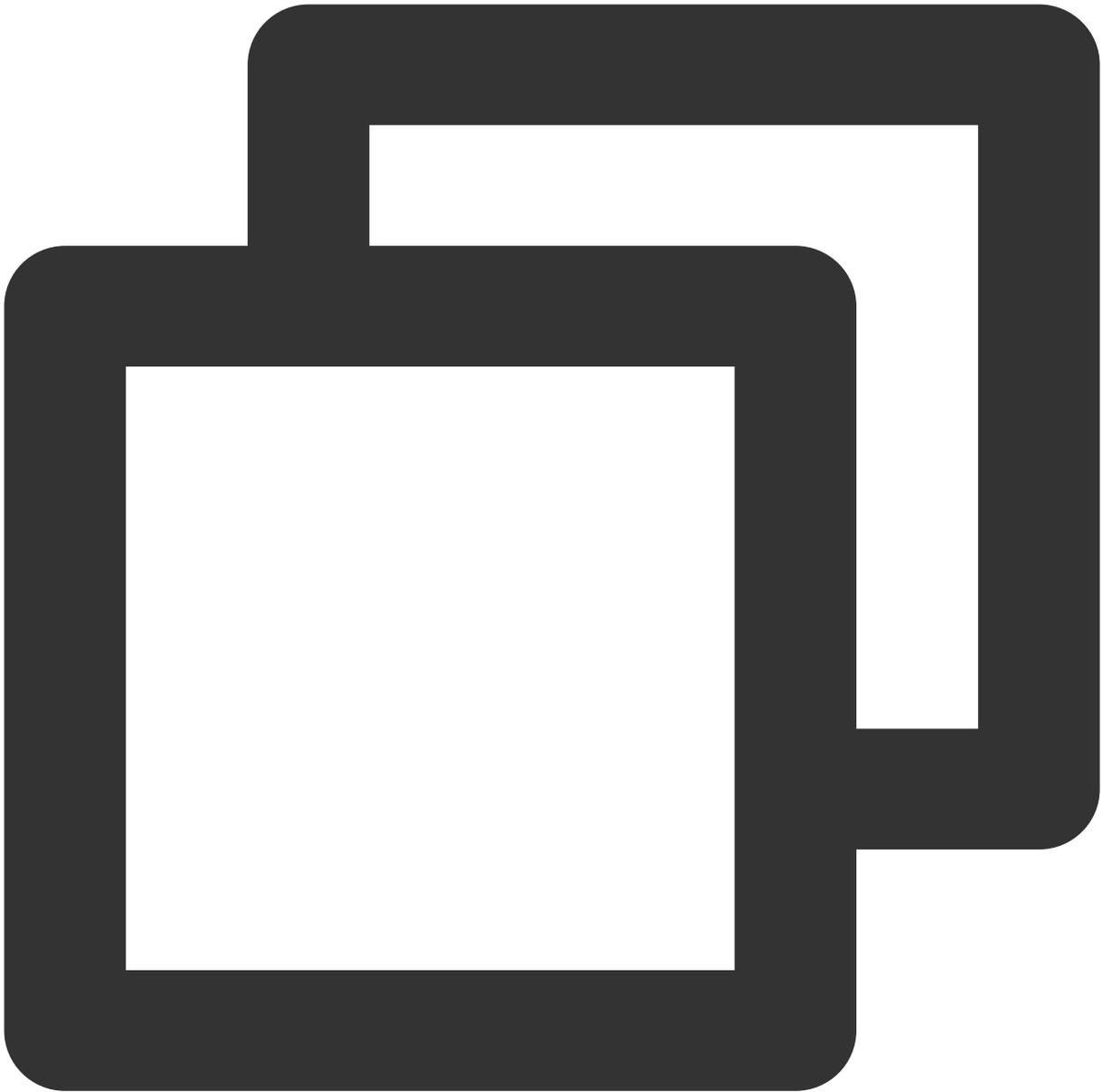


```
./coscli rm cos://bucket1/example/ -r --include ".*.mp4"
```

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

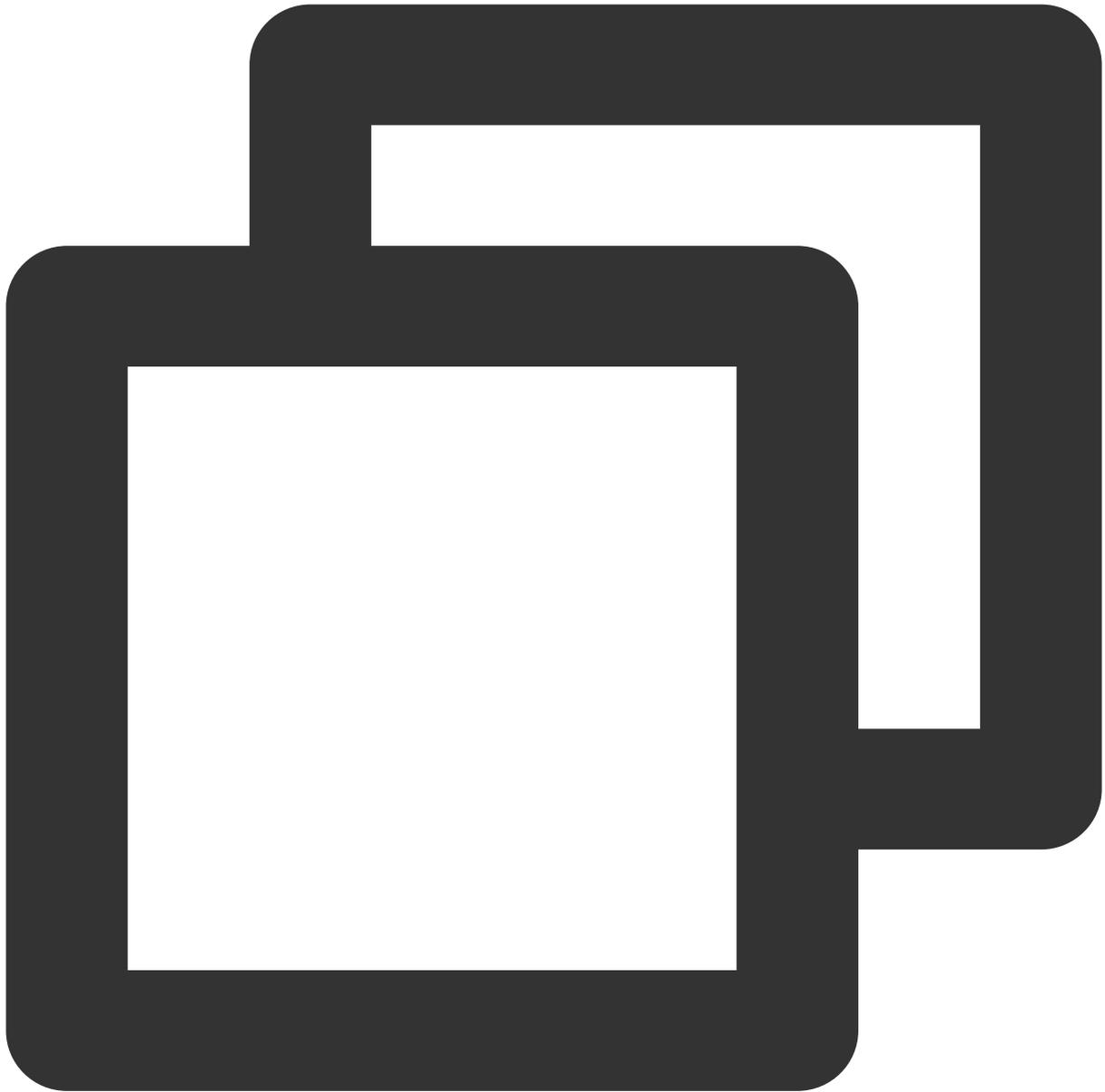
操作示例

删除文件



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

删除 **pictrue** 文件夹下的所有文件



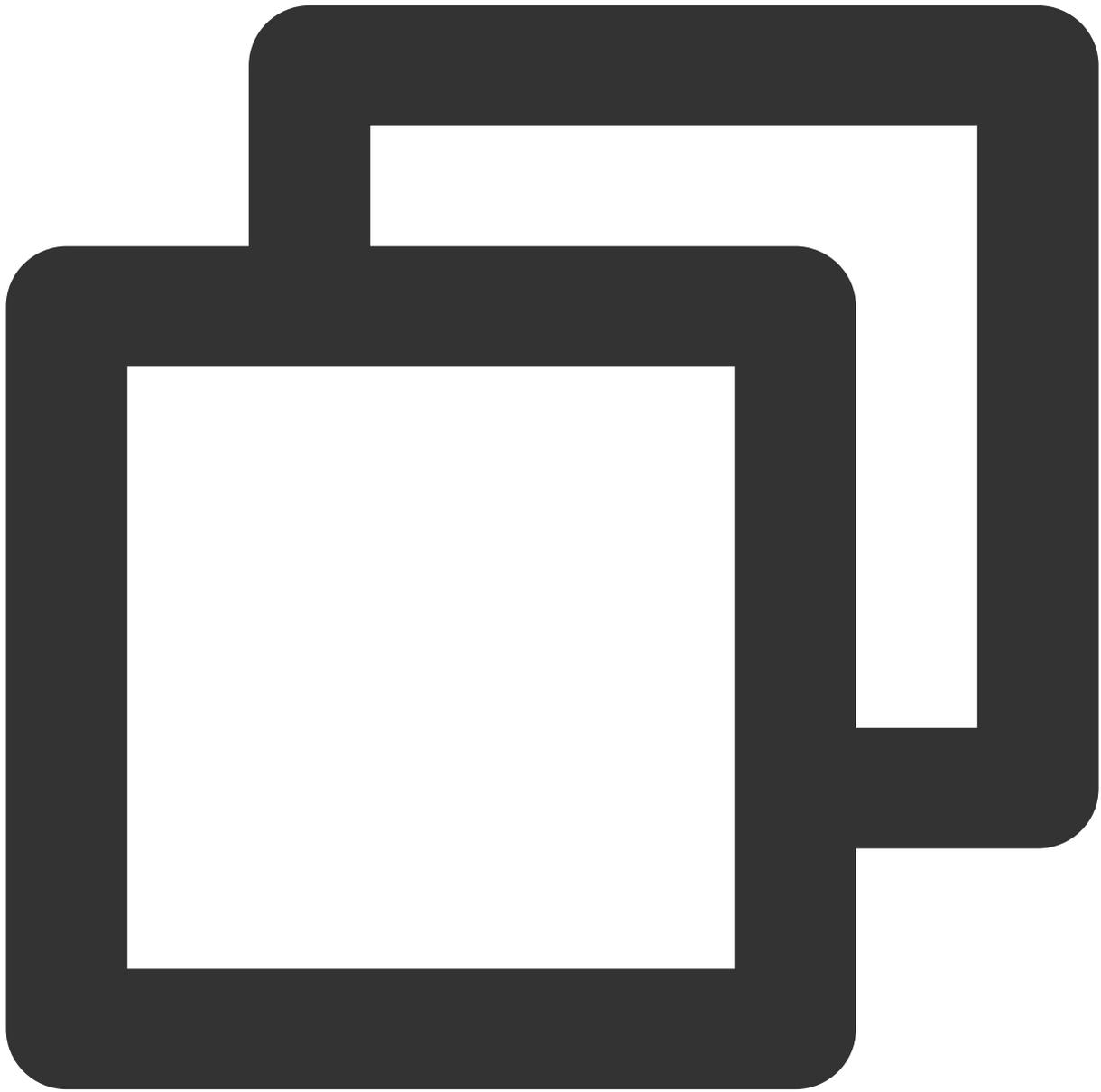
```
./coscli rm cos://bucket1/pictrue/ -r
```

获取文件哈希值 - hash

最近更新时间：2024-01-06 15:54:24

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

命令格式



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

hash 命令包含以下参数：

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

hash 命令包含以下可选 flag：

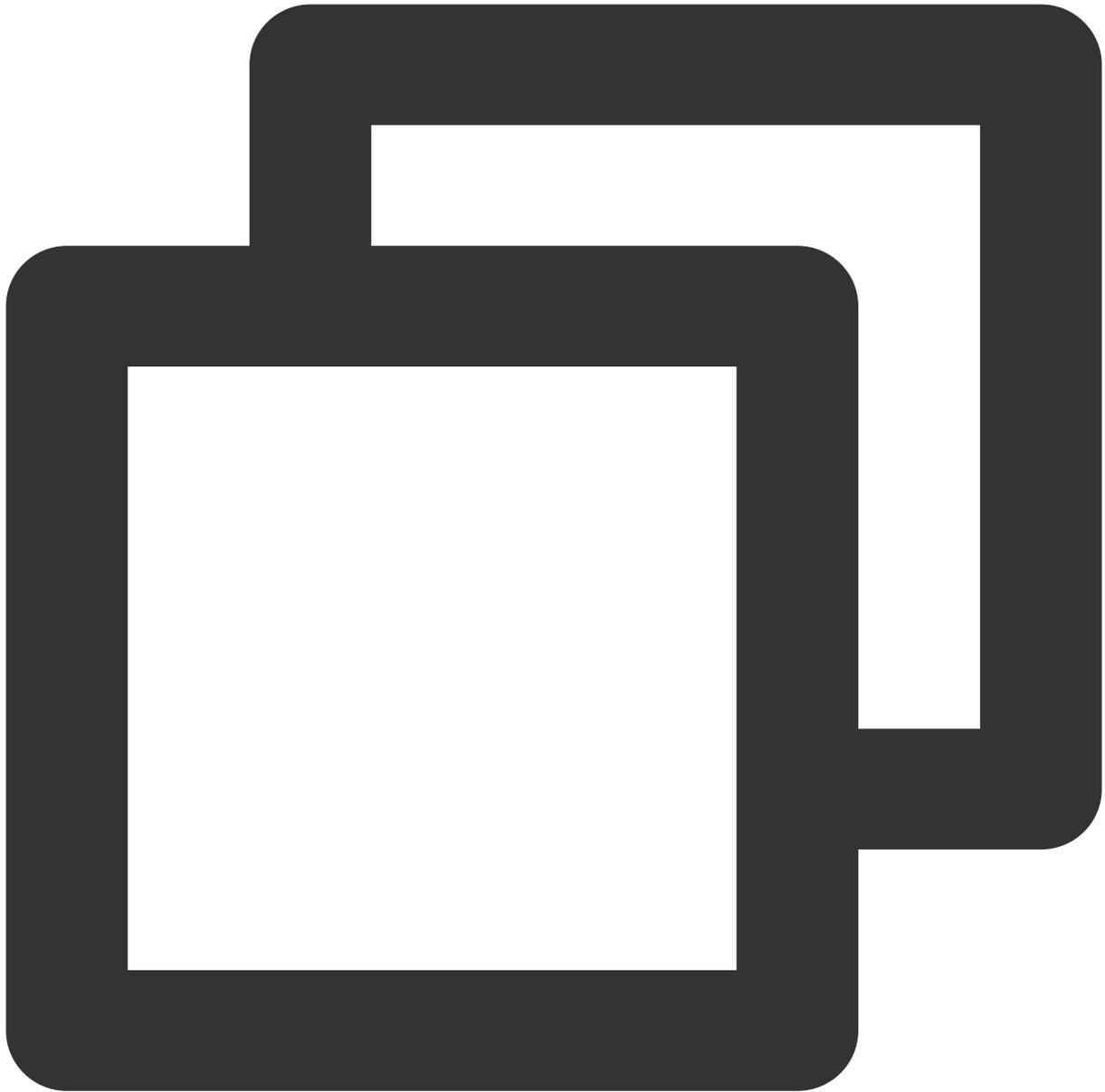
flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--type	哈希类型（md5 或 crc64，默认为 crc64）

说明

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

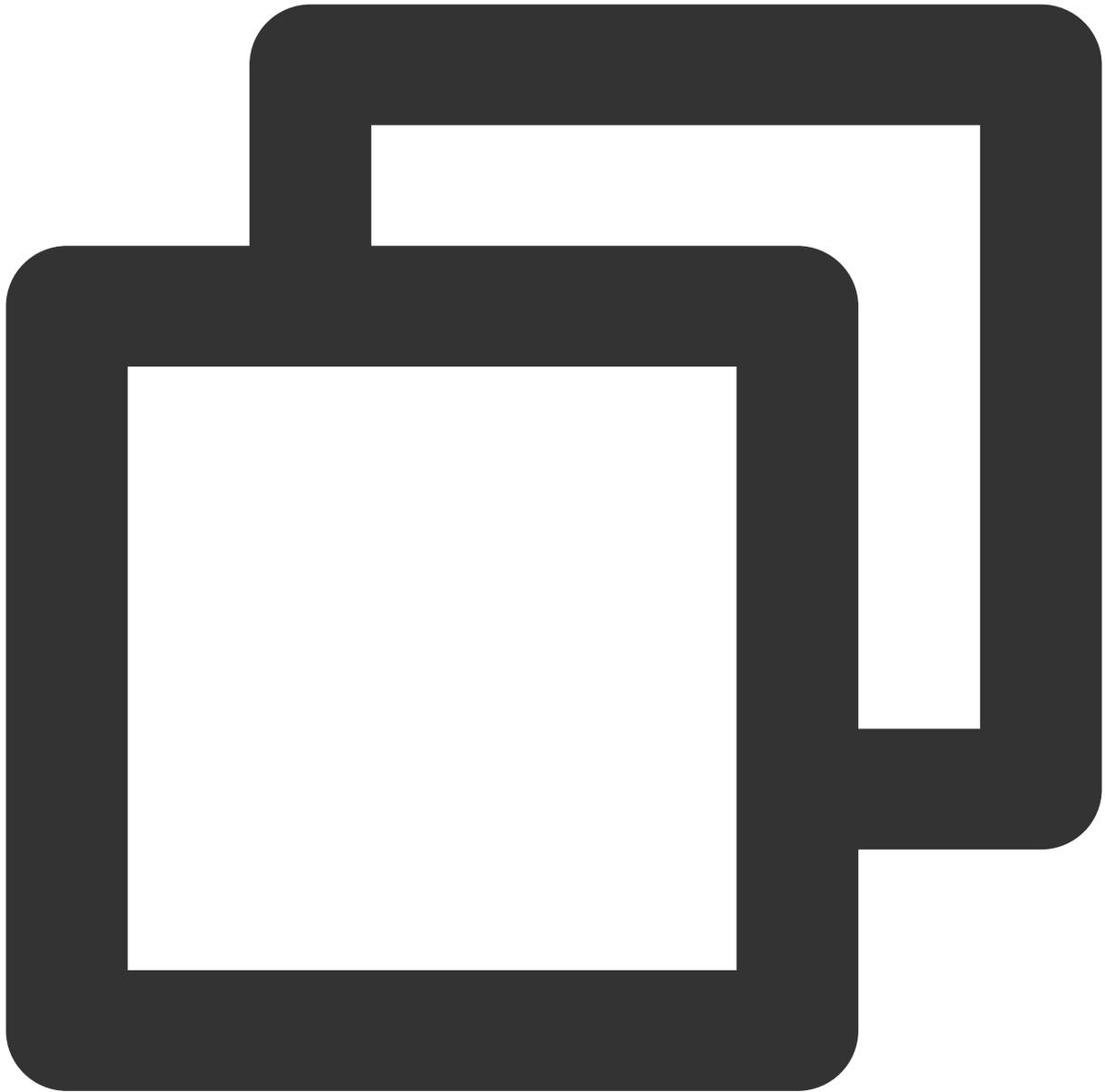
操作示例

计算本地文件的 crc64



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

获取 COS 文件的 md5



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

列出分块上传中产生的碎片 - lsparts

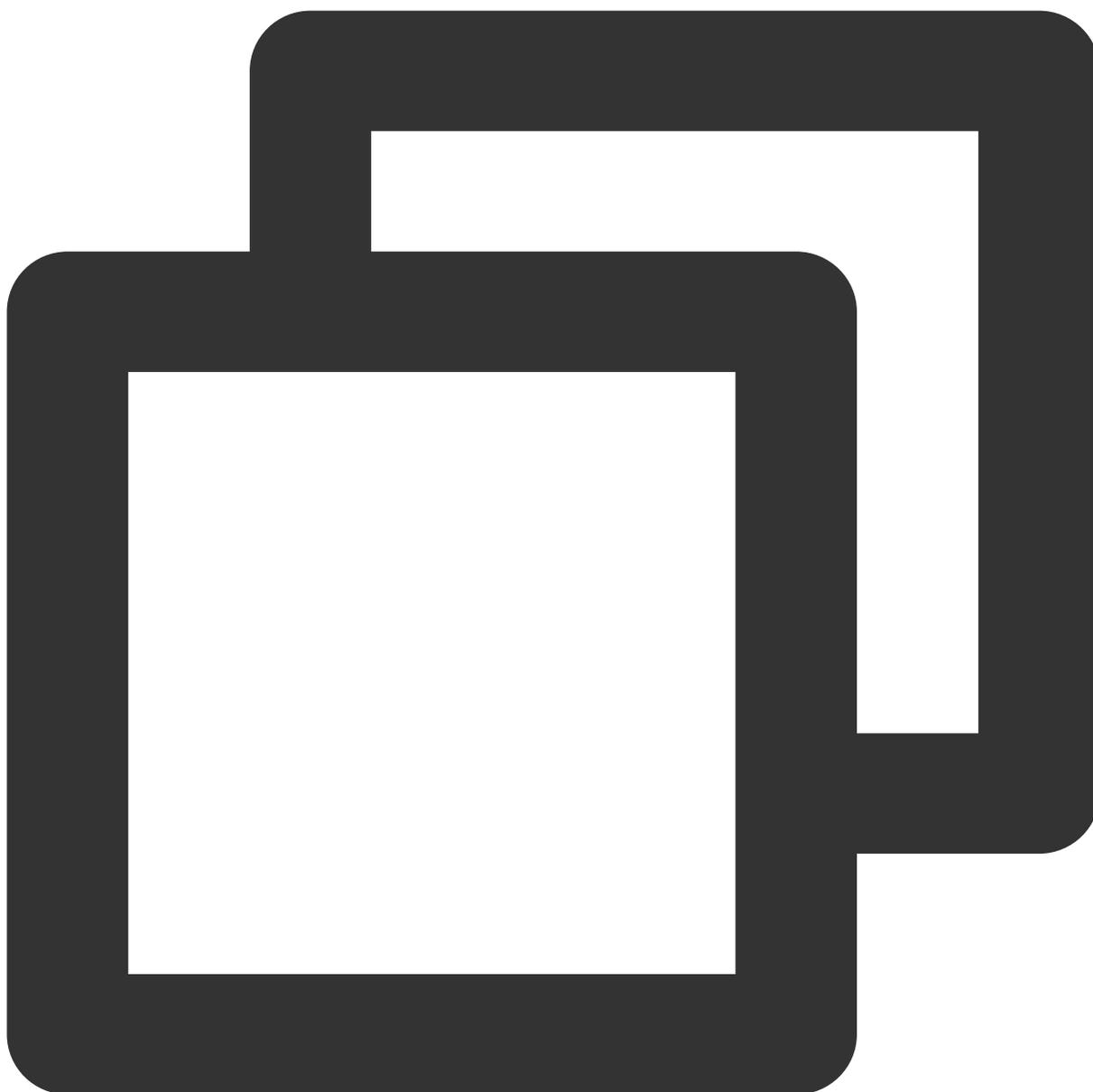
最近更新时间：2024-01-06 15:54:24

lsparts 命令列出分块上传中产生的文件碎片。

注意

关于分块上传的更多信息，请参见 [分块上传](#)。

命令格式



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

lsparts 命令包含以下参数：

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

/prefix/	可选参数。指定某一文件夹	/picture/
----------	--------------	-----------

lsparts 命令包含以下可选 flag：

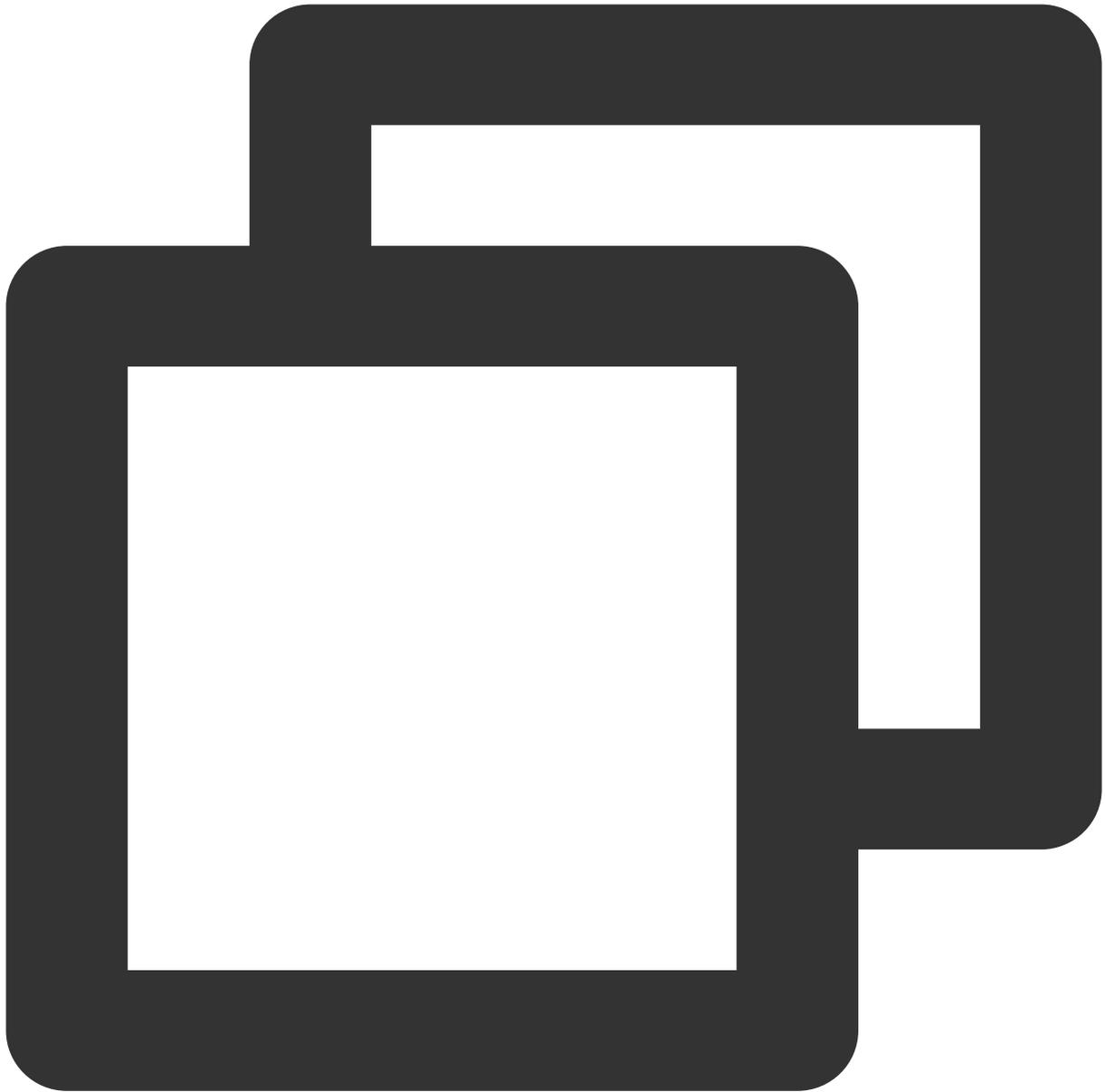
flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件
无	--limit	指定列出的最大数量（0 - 1000）

说明

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

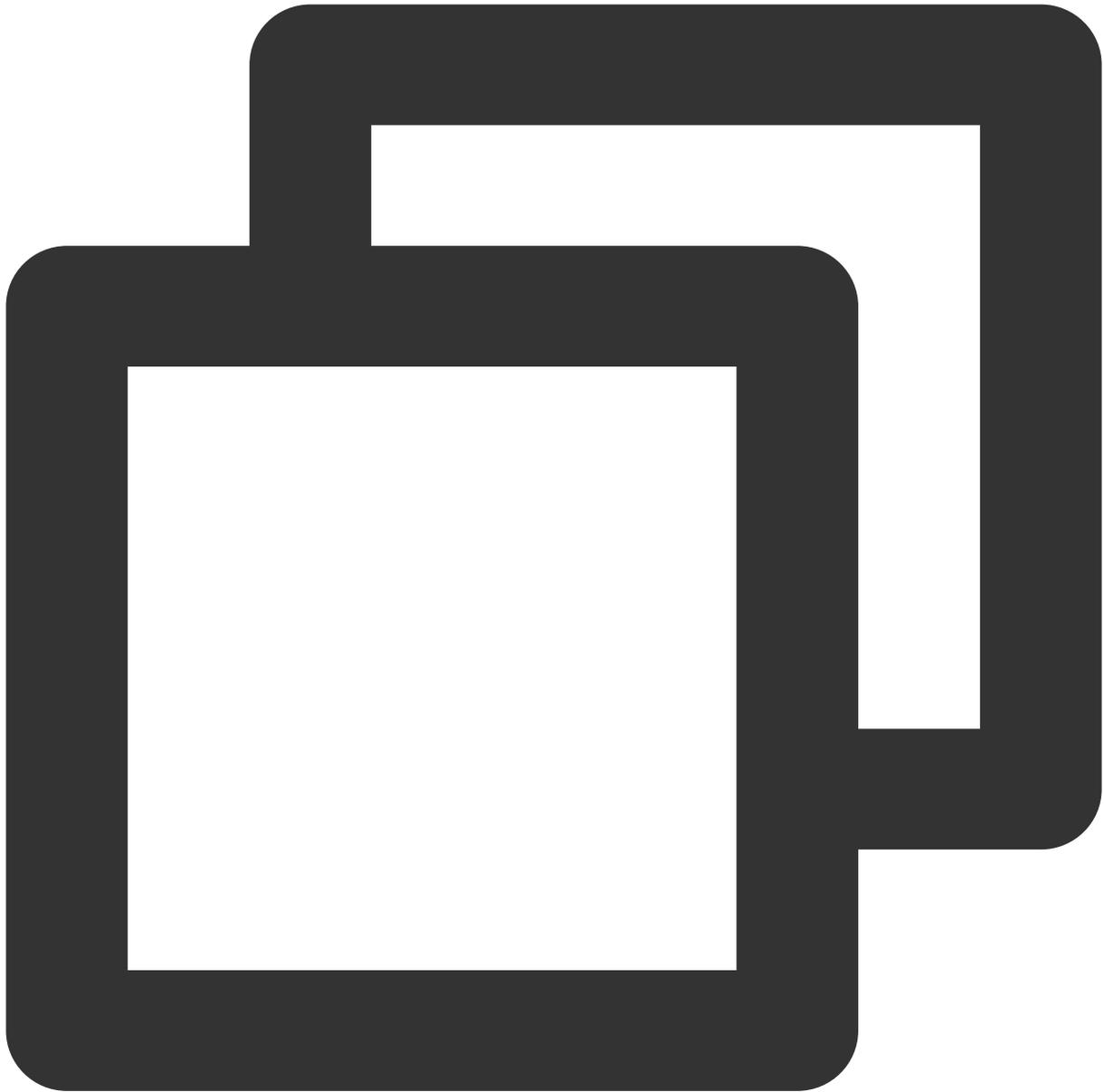
操作示例

列出 **bucket1** 桶内的所有文件碎片



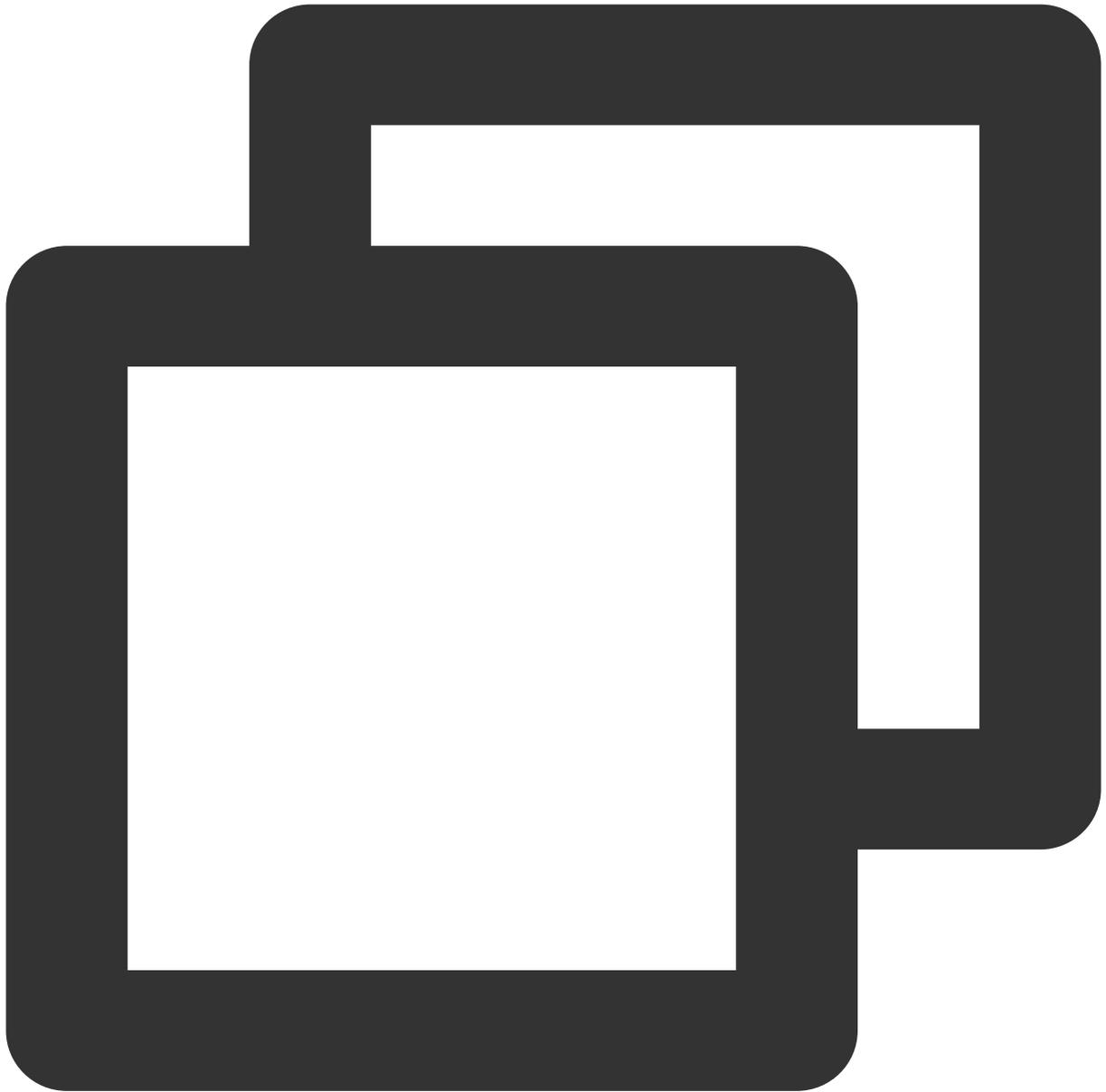
```
./coscli lsparts cos://bucket1
```

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



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

列出 bucket1 桶内 pictrue 文件夹下的所有碎片



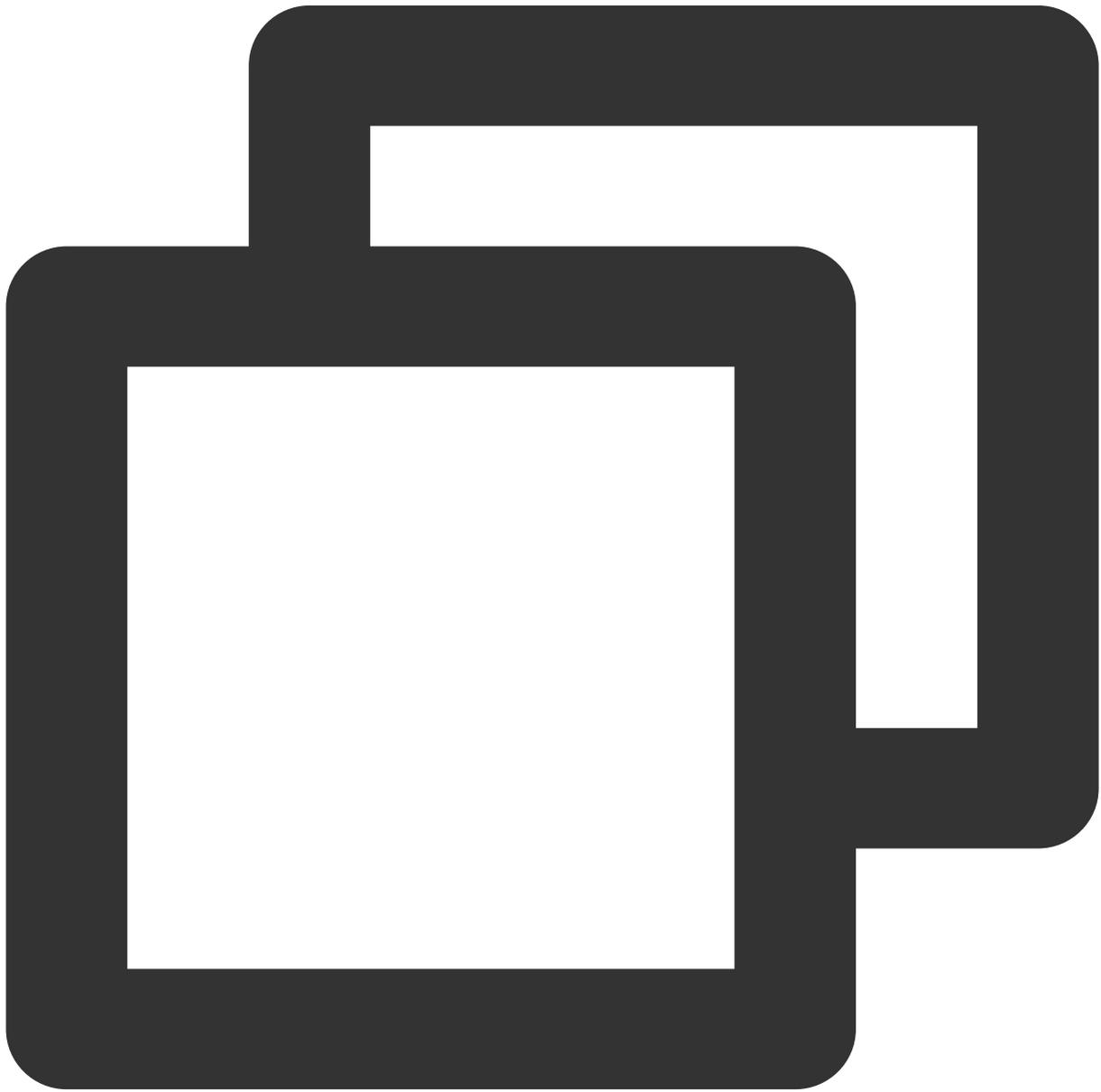
```
./coscli lsparts cos://bucket1/pictrue/
```

清理碎片 - abort

最近更新时间：2024-01-06 15:54:24

abort 命令用于清理分块上传过程中产生的文件碎片。

命令格式



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

abort 命令包含以下参数：

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

abort 命令包含以下可选 flag：

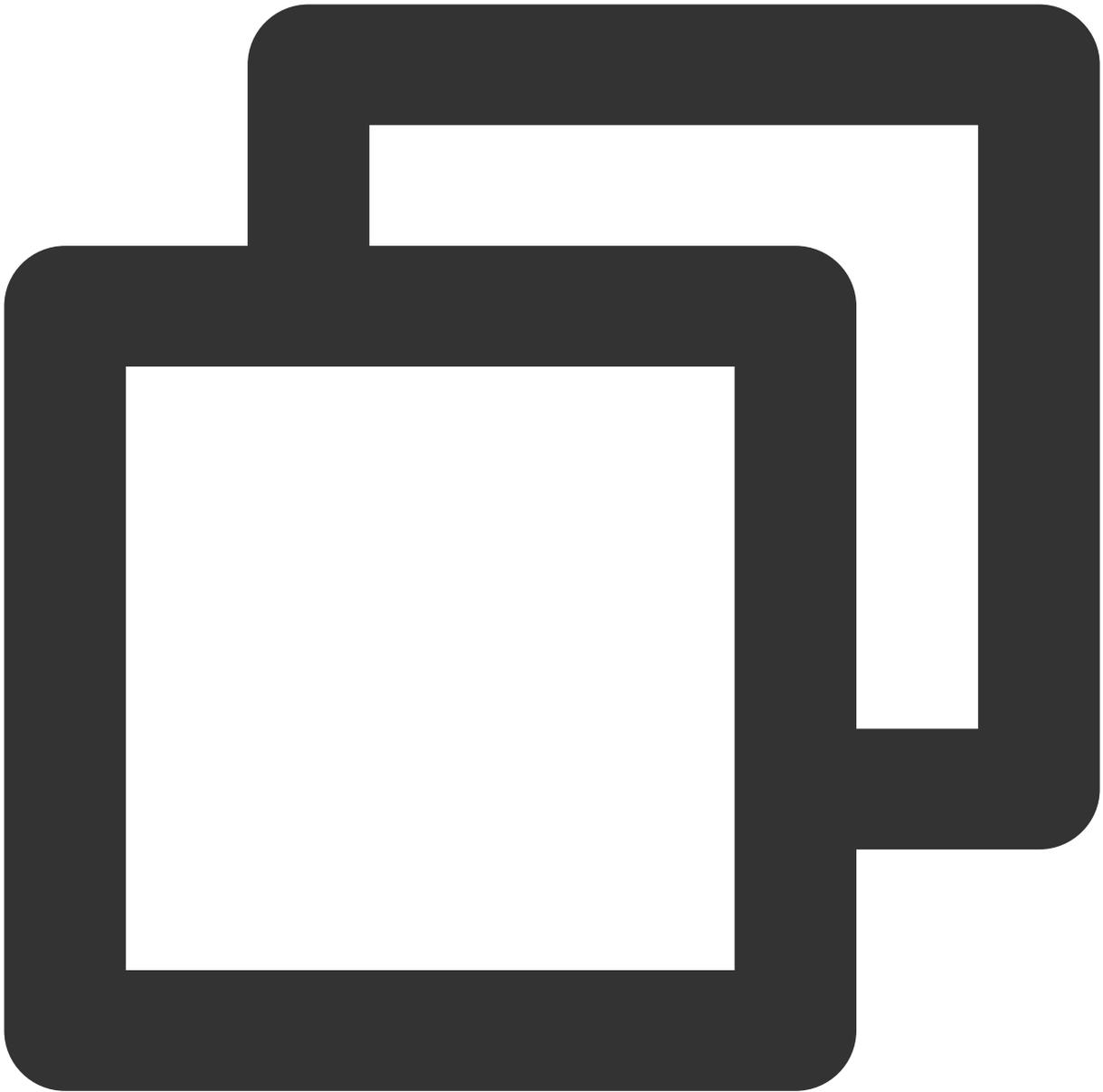
flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件

说明

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

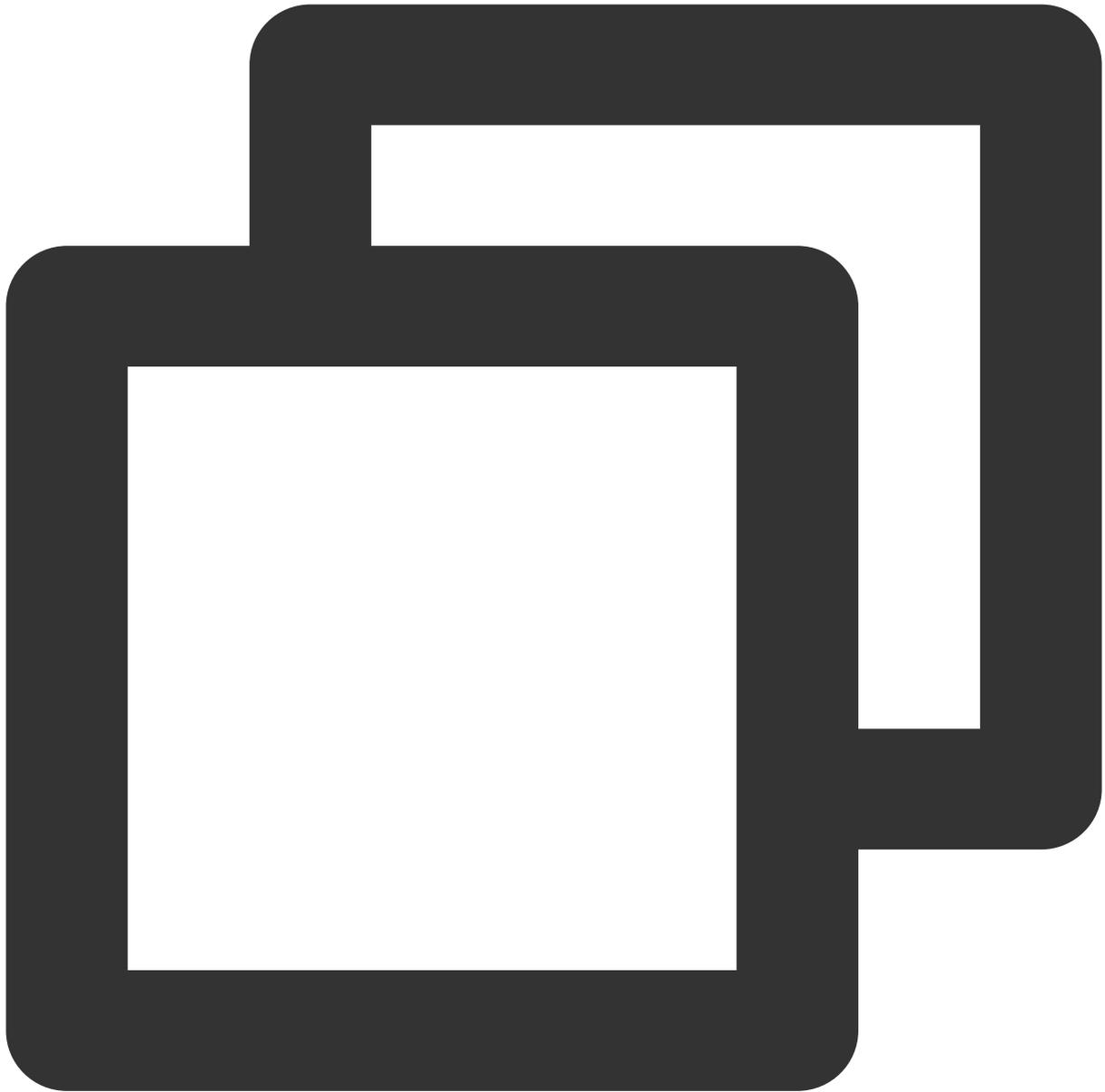
操作示例

清除 bucket1 桶内的所有文件碎片



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

清除 bucket1 桶内 picttrue 文件夹下的所有碎片



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

取回归档文件 - restore

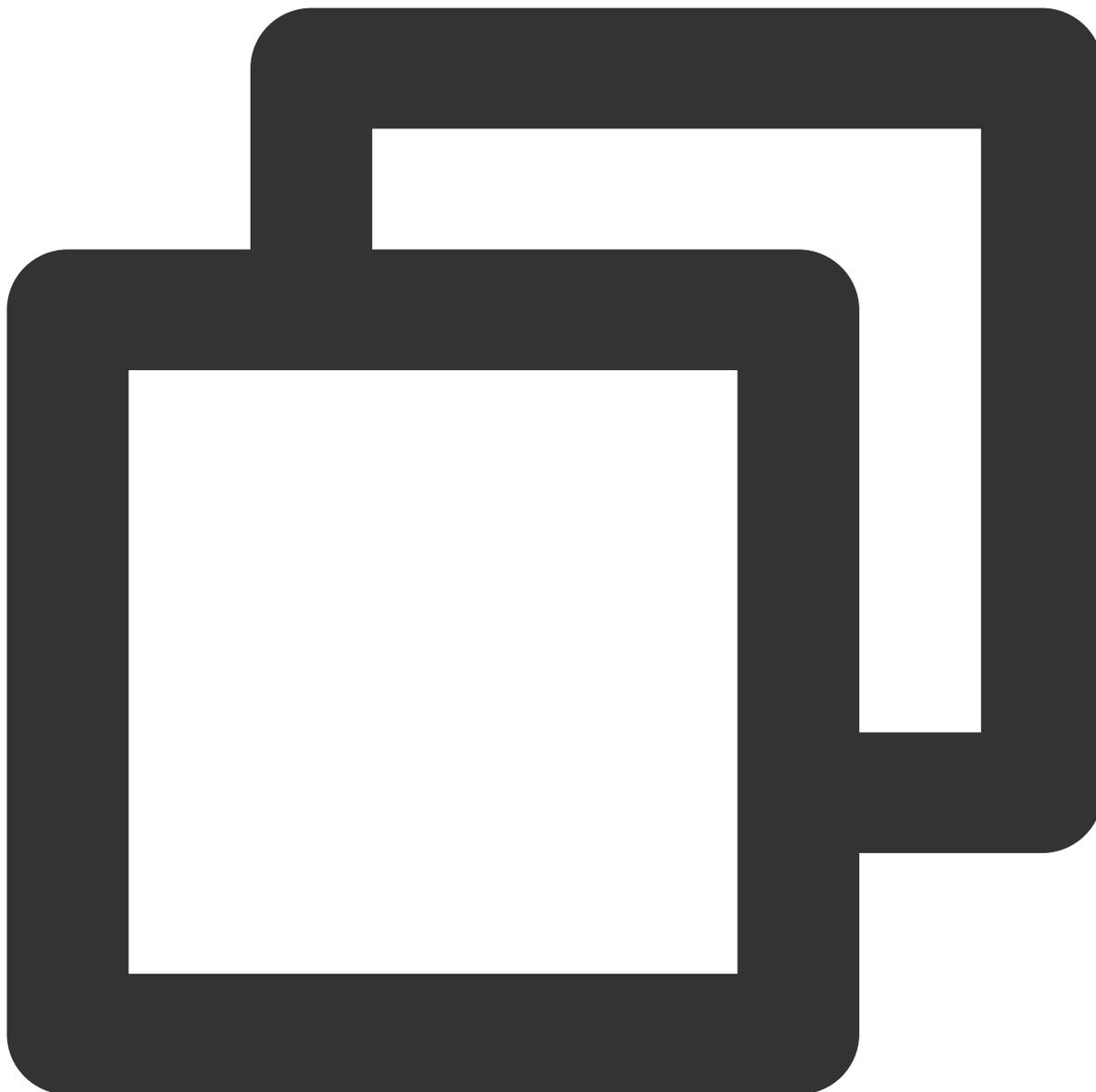
最近更新时间：2024-01-06 15:54:24

restore 命令用于取回归档文件。

注意

关于取回归档文件的更多信息，请参见 [POST Object restore - API 文档](#)。

命令格式



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

restore 命令包含以下参数：

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

restore 命令包含以下可选 flag：

flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
无	--include	包含特定模式的文件
无	--exclude	排除特定模式的文件
-d	--days	指定解冻对象所生成的临时文件的过期时间（默认为3天）
-m	--mode	指定恢复模式（默认 Standard）
-r	--recursive	递归遍历文件夹

说明

`--include` 和 `--exclude` 支持标准正则表达式的语法，您可以使用它来过滤出符合特定条件的文件。

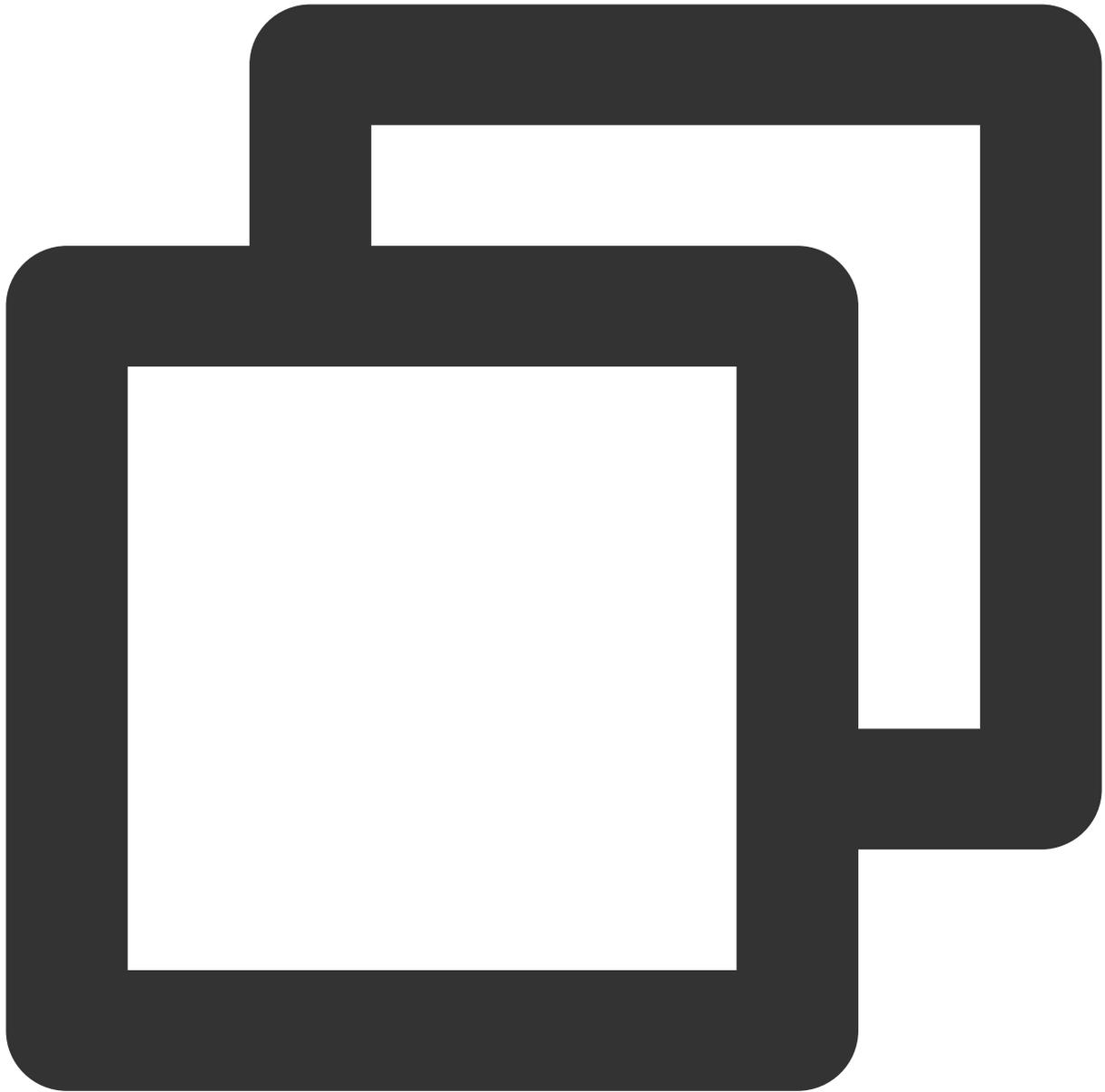
使用 `zsh` 时，您可能需要在 `pattern` 串的两端加上双引号。 ``

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

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

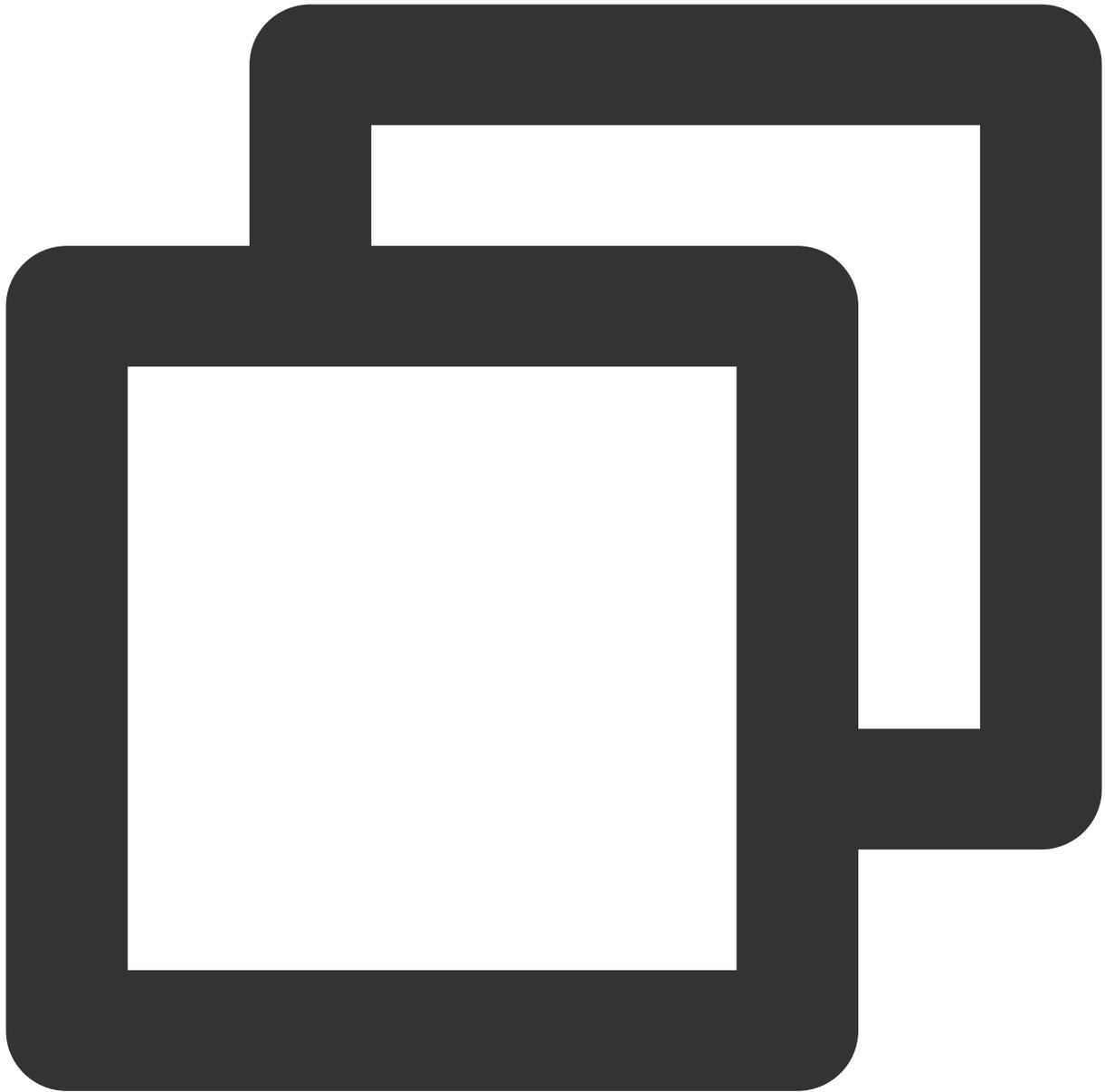
操作示例

以标准取回模式取回 `bucket1` 桶内的归档文件



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

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



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

说明

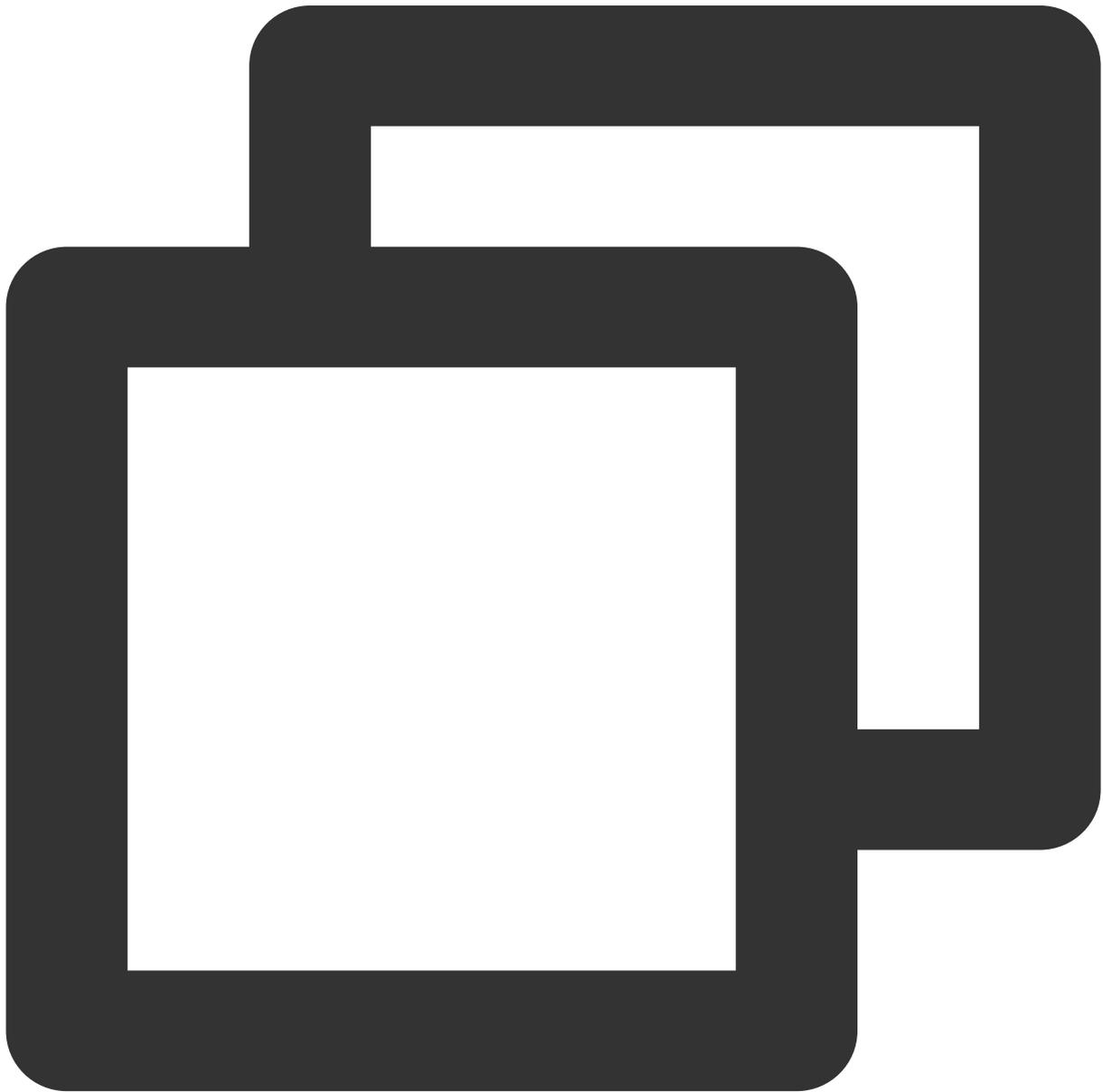
在执行此命令前，您需要保证文件夹下所有文件都是相同类型的（例如 ARCHIVE 类型）。如果存在不同类型的文件，请配合使用 `--include` 或 `--exclude` 将相同类型的文件过滤出来。

获取预签名 URL - signurl

最近更新时间：2024-01-06 15:54:24

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

命令格式



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

signurl 命令包含以下参数：

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

signurl 命令包含以下可选 flag：

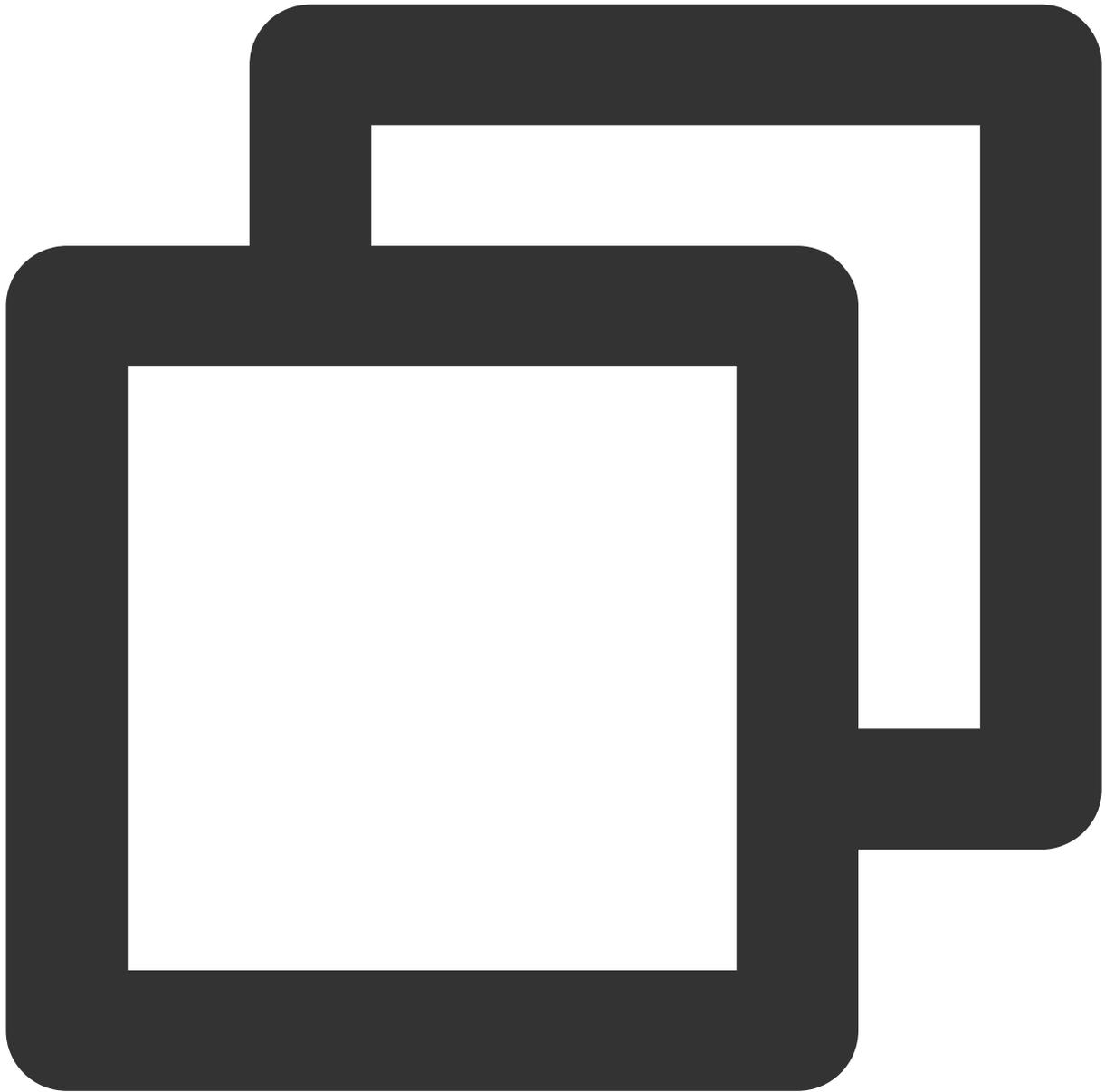
flag 简写	flag 全称	flag 用途
-h	--help	查看该命令的具体用法
-t	--time	设置 URL 过期时间（默认 1000s）

说明

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

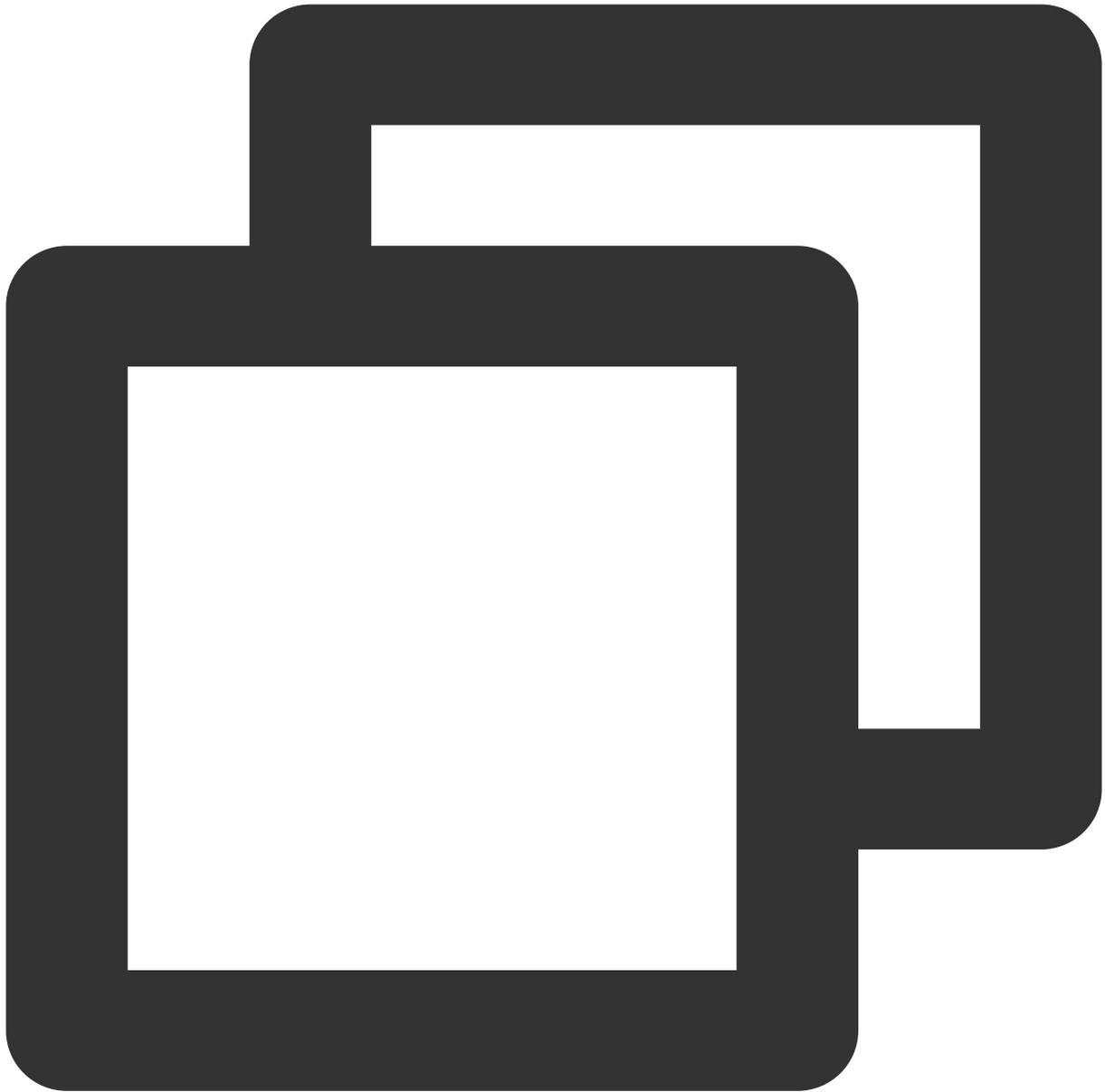
操作示例

获取 bucket1 桶内 picttrue.jpg 的预签名 URL



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

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



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

常见问题

最近更新时间：2024-01-06 10:43:44

COSCLI 工具与 COSCMD 工具有什么区别？

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

COSCMD 工具

最近更新时间：2024-06-28 16:58:02

功能说明

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

使用环境

系统环境

支持 Windows、Linux 和 macOS 系统。

说明：

请保证本地字符格式为 UTF-8，否则操作中文版的文件会出现异常。

请确保本机时间已经与国际标准时间校准，如误差过大，将导致无法正常使用。

软件依赖

Python 2.7 和 Python 3。

最新版本的 pip。

说明：

建议用户直接安装集成了 pip 的 Python 版本。

安装及配置

环境安装与配置详细操作请参见 [Python 安装与配置](#)。

pip 环境安装与配置详细操作请参见 [官网 pip 安装说明](#)。

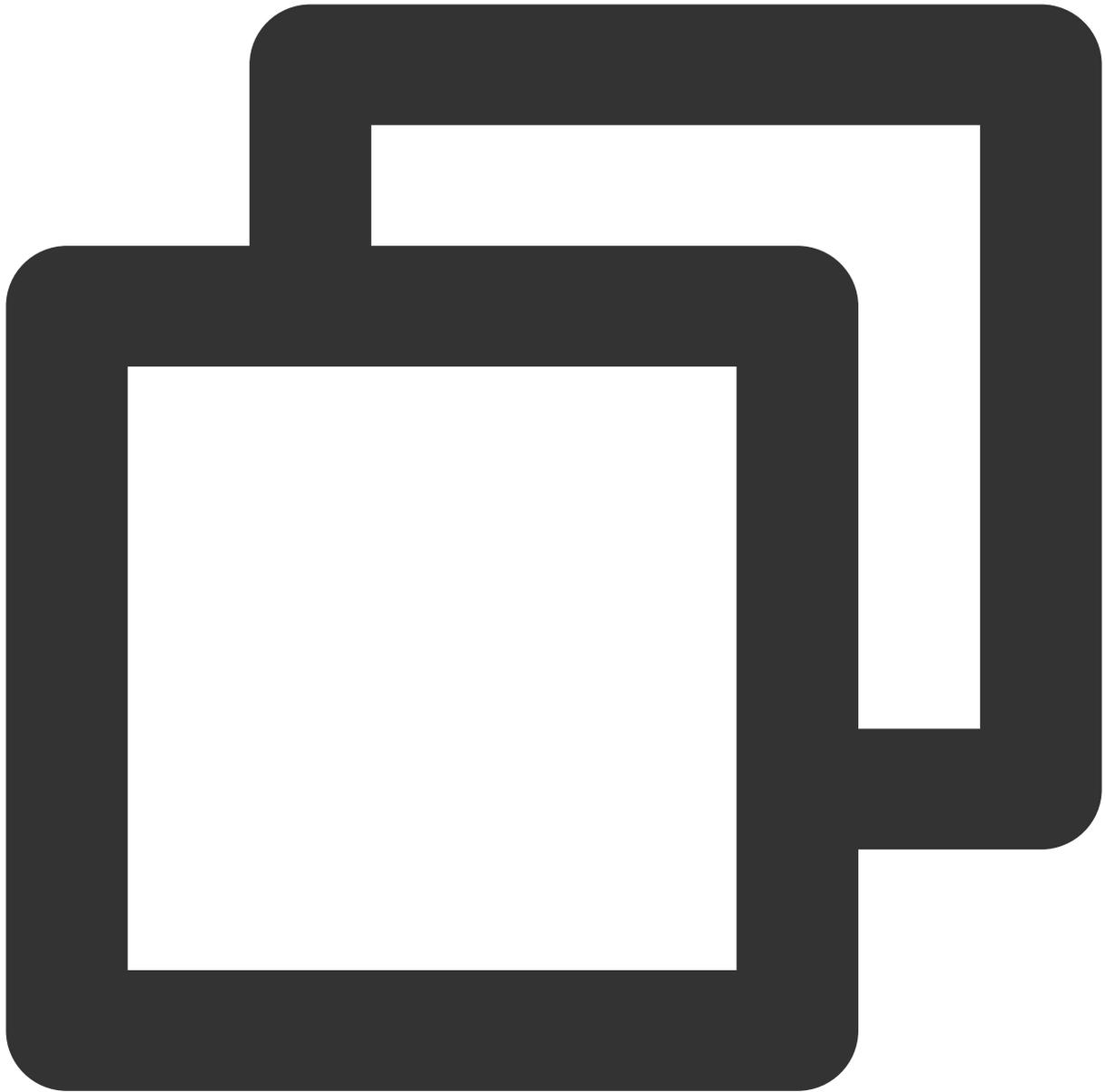
下载与安装

下面为用户提供以下三种方式安装 COSCMD。

方式一：

1. 通过 pip 安装

执行 `pip` 命令进行安装：



```
pip install coscmd
```

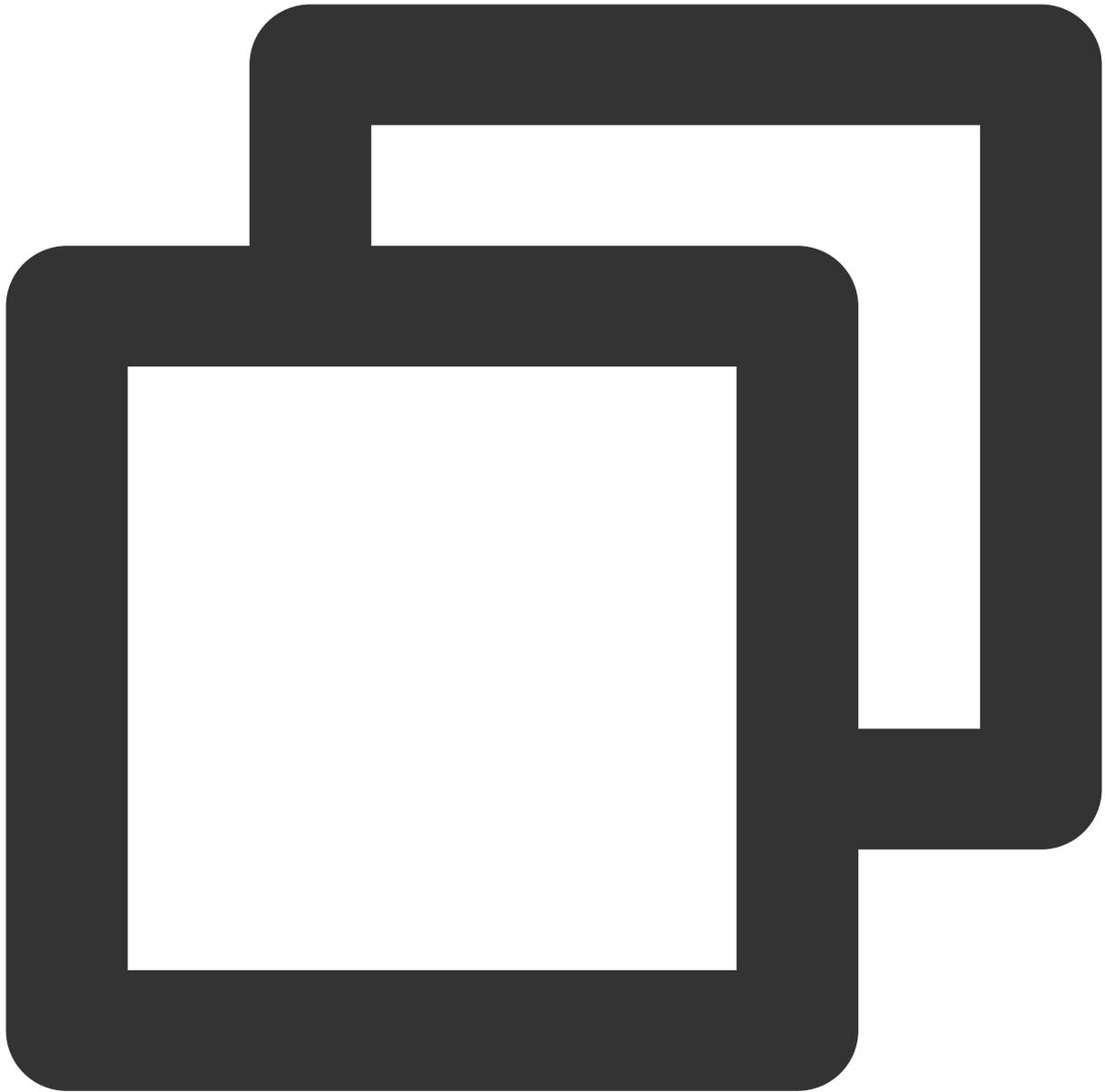
安装成功之后，用户可以通过 `-v` 或者 `--version` 命令查看当前的版本信息。

注意：

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

2. pip 更新

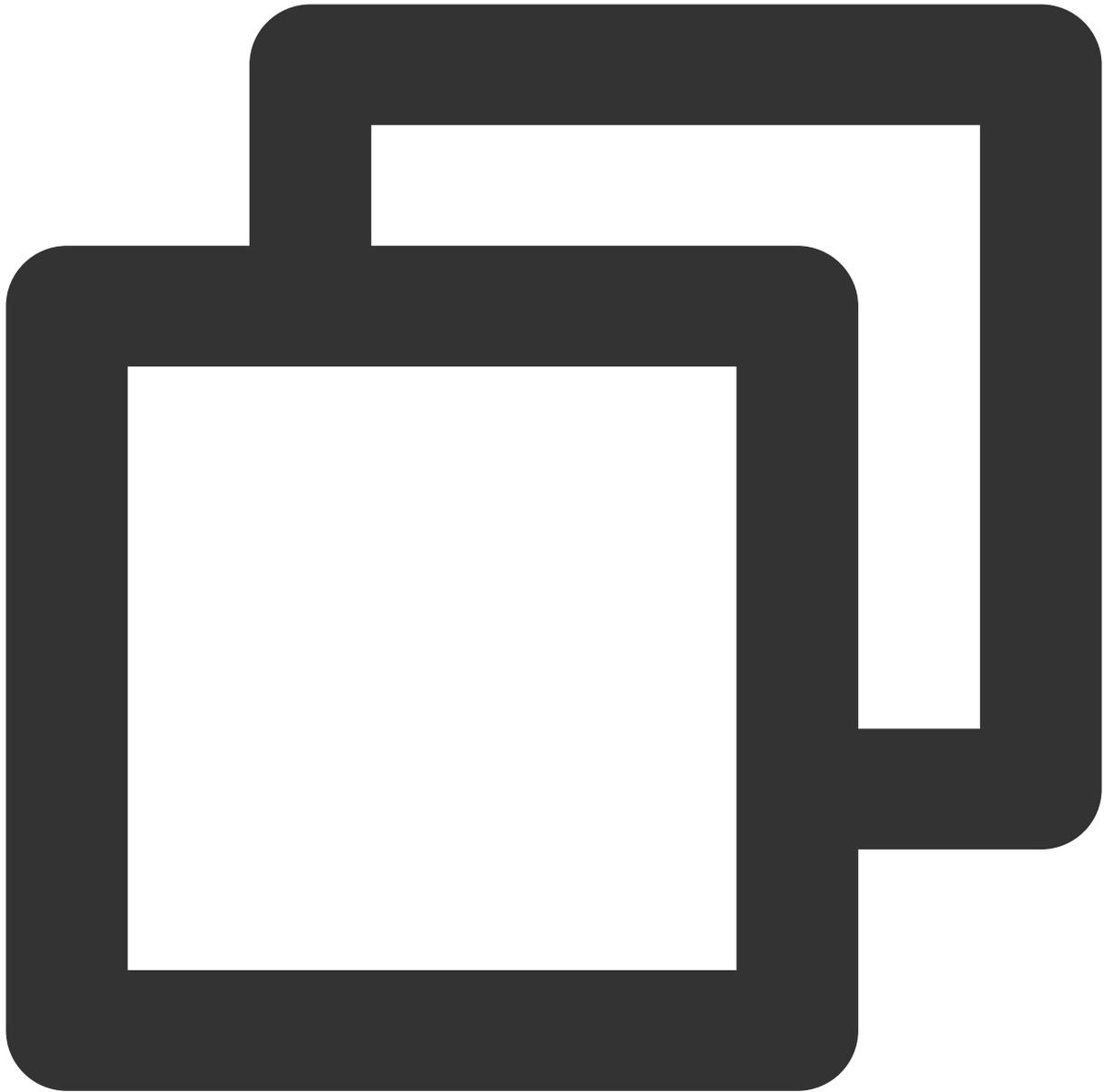
安装完成后，执行以下命令进行更新：



```
pip install coscmd -U
```

方式二：源码安装（不推荐）

源码下载地址：[单击此处](#)。



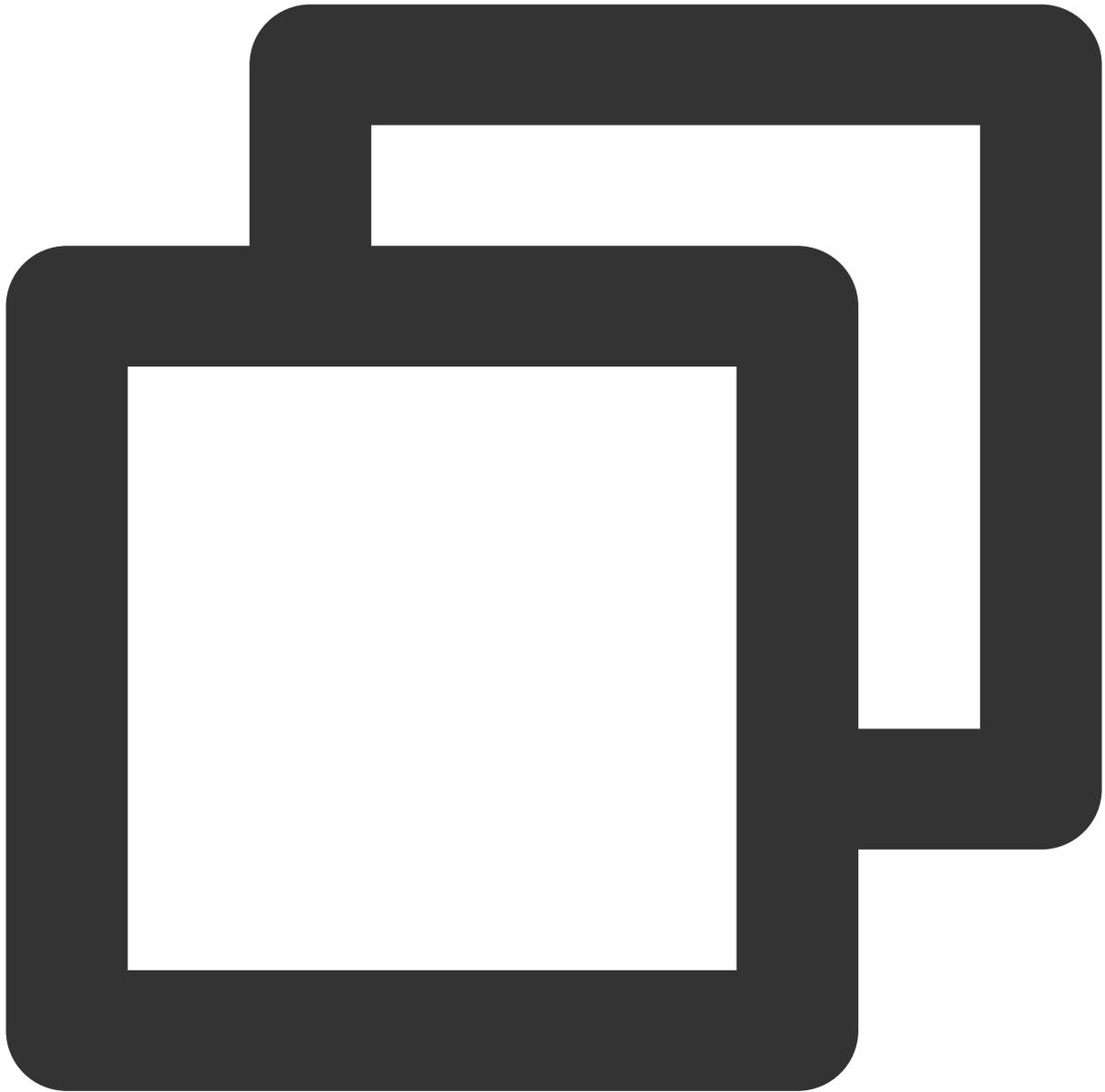
```
git clone https://github.com/tencentyun/coscmd.git
cd coscmd
python setup.py install
```

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。

方式三：离线安装**注意：**

请确保两台机器的 Python 版本保持一致，否则会出现安装失败的情况。



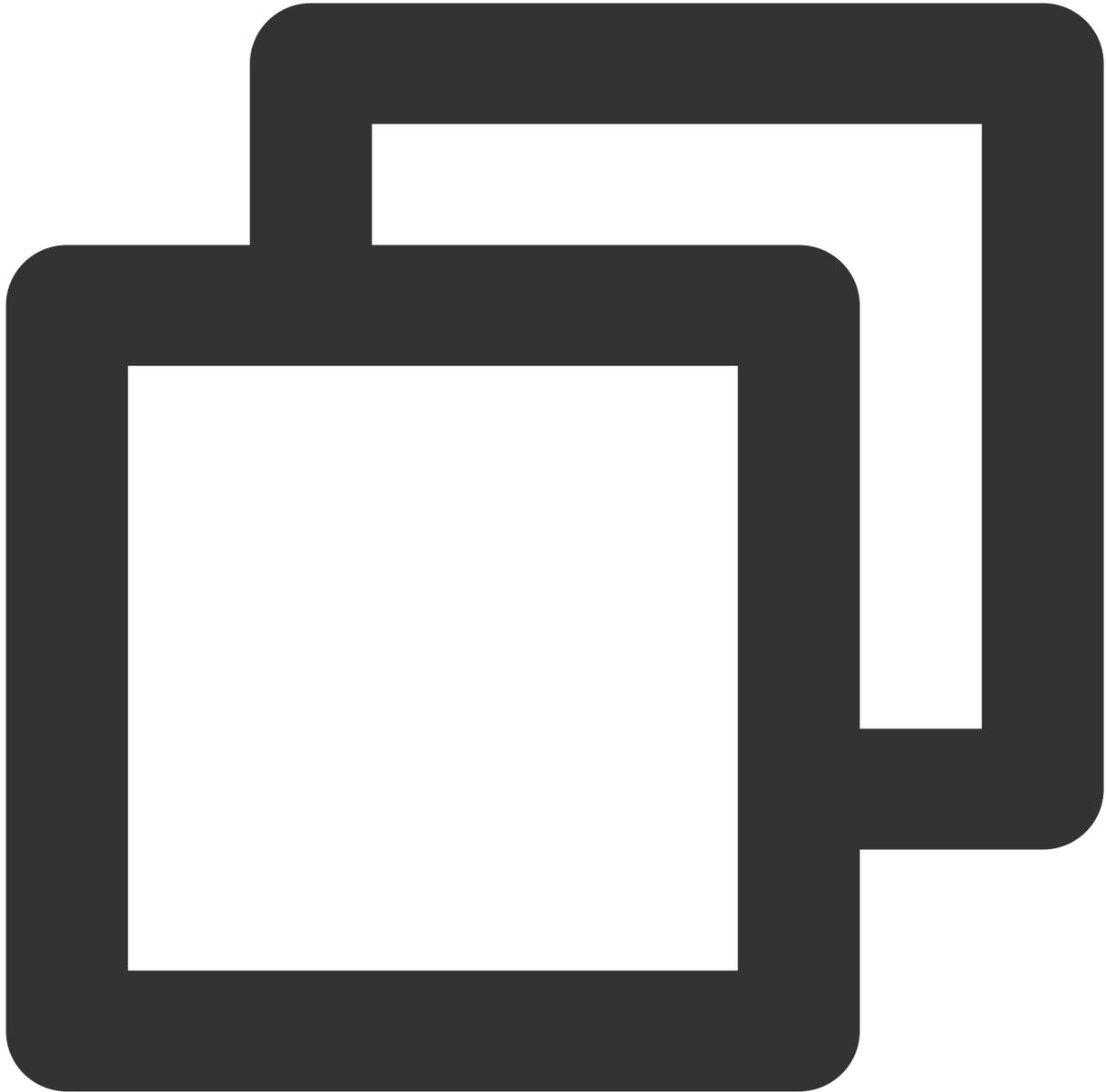
```
# 在有外网的机器下运行如下命令
mkdir coscmd-packages
pip download coscmd -d coscmd-packages
tar -czvf coscmd-packages.tar.gz coscmd-packages
```

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

配置参数

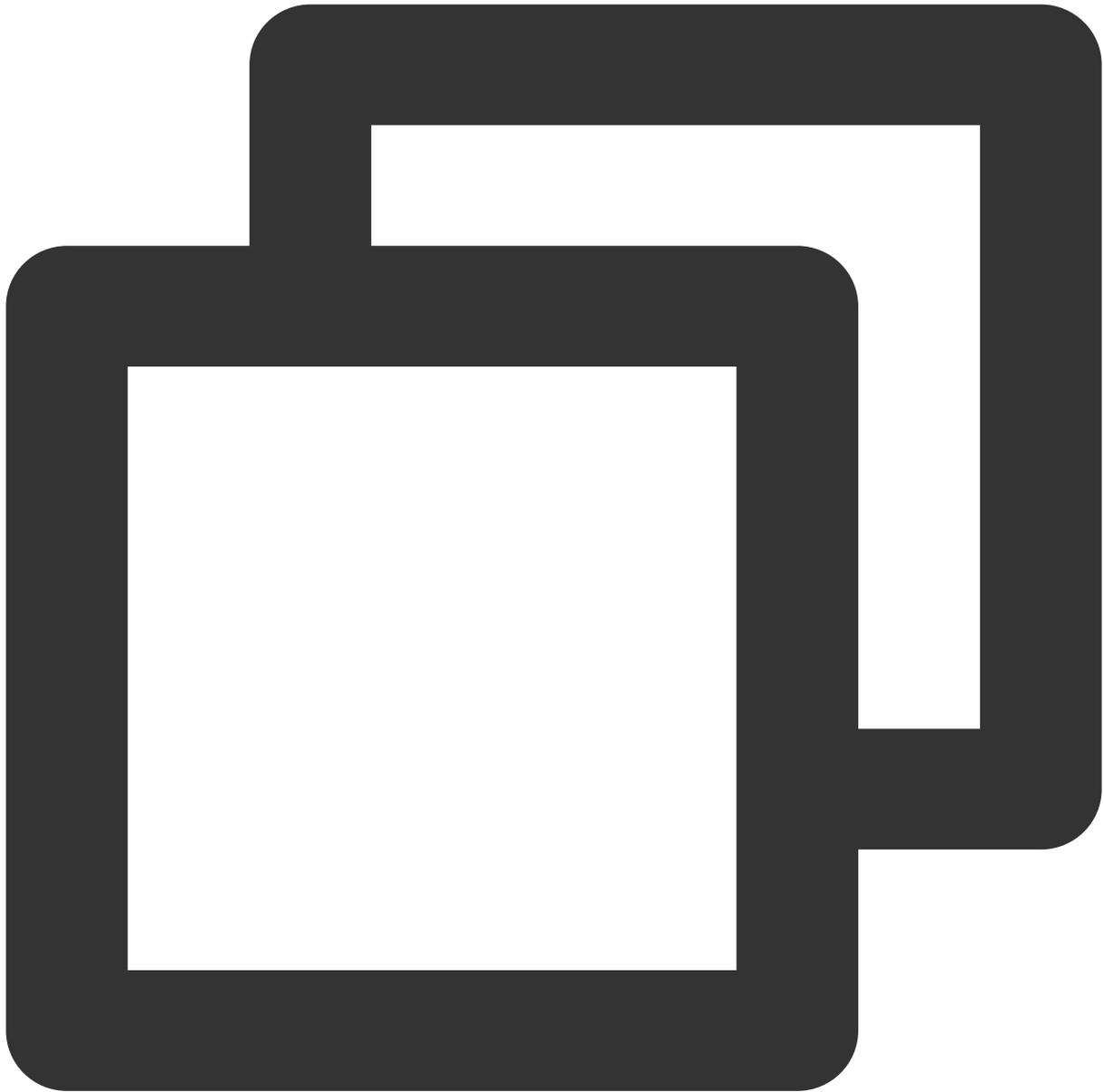
查看 help 选项

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



```
coscmd -h
```

help 信息如下所示：



```
usage: coscmd [-h] [-d] [-s] [-b BUCKET] [-r REGION] [-c CONFIG_PATH]
             [-l LOG_PATH] [--log_size LOG_SIZE]
             [--log_backup_count LOG_BACKUP_COUNT] [-v]
             {config,upload,download,delete,abort,copy,move,list,listparts,info,re
             ...
```

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

positional arguments:

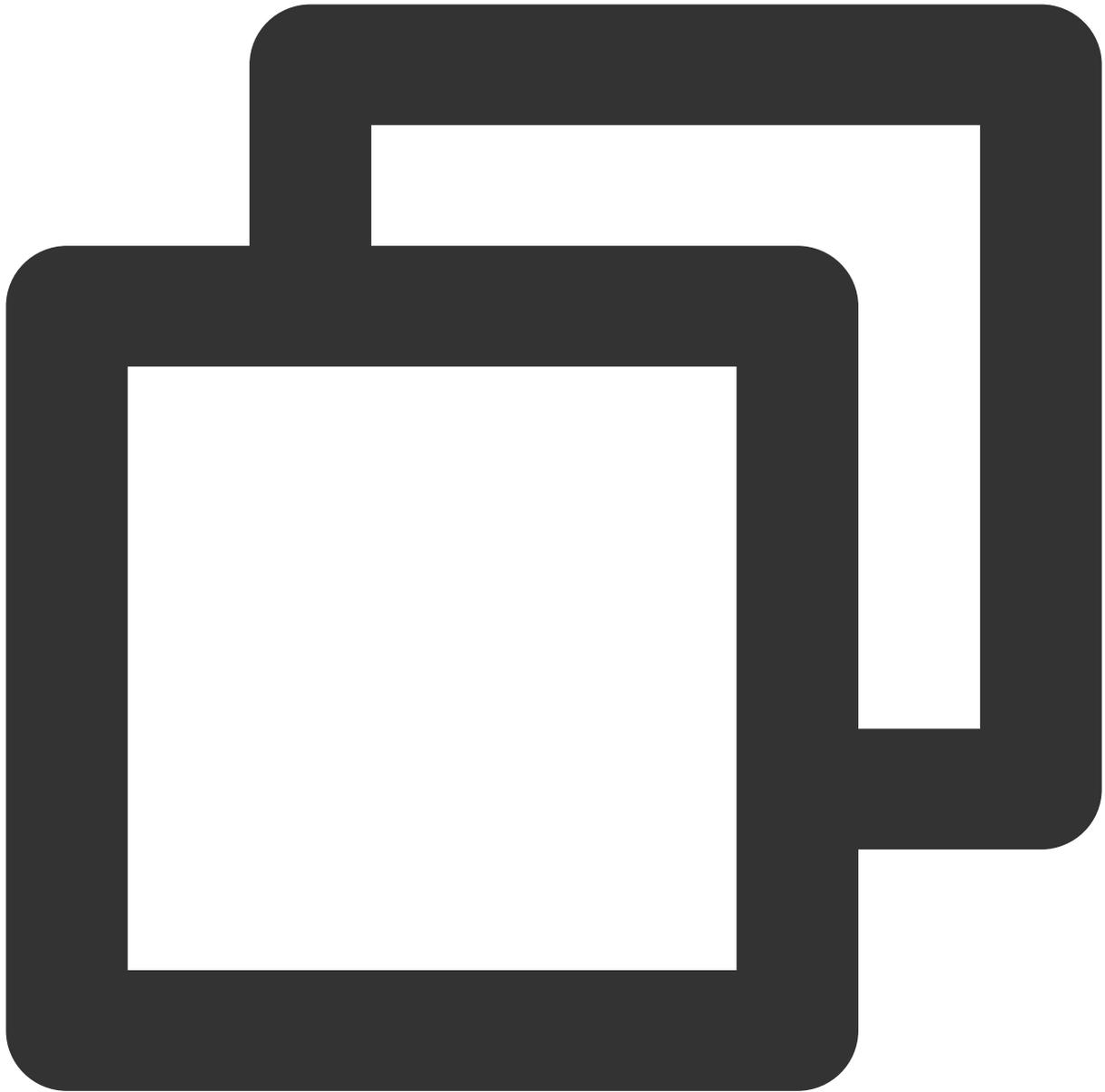
```

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

optional arguments:
  -h, --help          show this help message and exit
  -d, --debug         Debug mode
  -s, --silence       Silence mode
  -b BUCKET, --bucket BUCKET
                      Specify bucket
  -r REGION, --region REGION
                      Specify region
  -c CONFIG_PATH, --config_path CONFIG_PATH
                      Specify config_path
  -l LOG_PATH, --log_path LOG_PATH
                      Specify log_path
  --log_size LOG_SIZE specify max log size in MB (default 1MB)
  --log_backup_count LOG_BACKUP_COUNT
                      specify log backup num
  -v, --version       show program's version number and exit

```

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



```
coscmd upload -h //查看 upload 命令使用方法
```

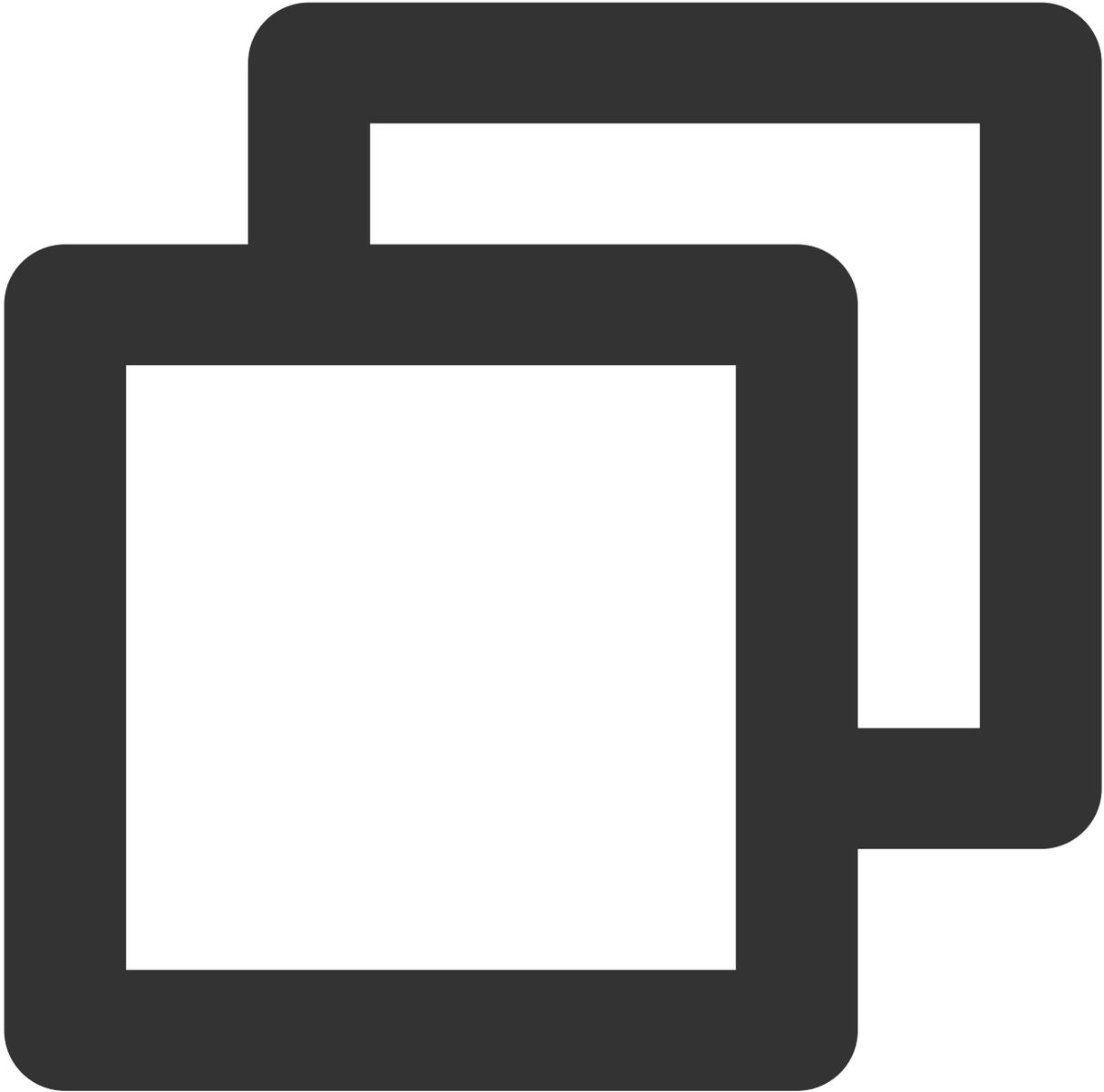
生成配置文件

COSCMD 工具在运行前会首先从配置文件中读取运行时所需的必要信息，COSCMD 会默认从 `~/.cos.conf` 中读取配置项。

说明：

配置前，您需要先在 COS 控制台创建一个用于配置参数的存储桶（例如 `configure-bucket-1250000000`），并创建密钥信息。

一个配置文件的示例如下所示：



```
[common]
secret_id = AKIDA6wUmImTMzvXZNbGLCgtusZ2E8mG****
secret_key = TghWBCyf5LIyTcXCoBdw1oRpytWk****
bucket = configure-bucket-1250000000
region = ap-chengdu
max_thread = 5
part_size = 1
retry = 5
timeout = 60
```

```
schema = https
verify = md5
anonymous = False
```

说明：

配置文件中 `timeout` 项，单位为秒，指读写数据的超时时间。

配置文件中 `schema` 项，可选值为 `http`、`https`，默认为 `https`。

配置文件中 `anonymous` 项，可选值为 `True`、`False`，表示是否使用匿名模式，即签名保持为空。

更多配置参数说明，请使用命令 `coscmd config -h` 查看。

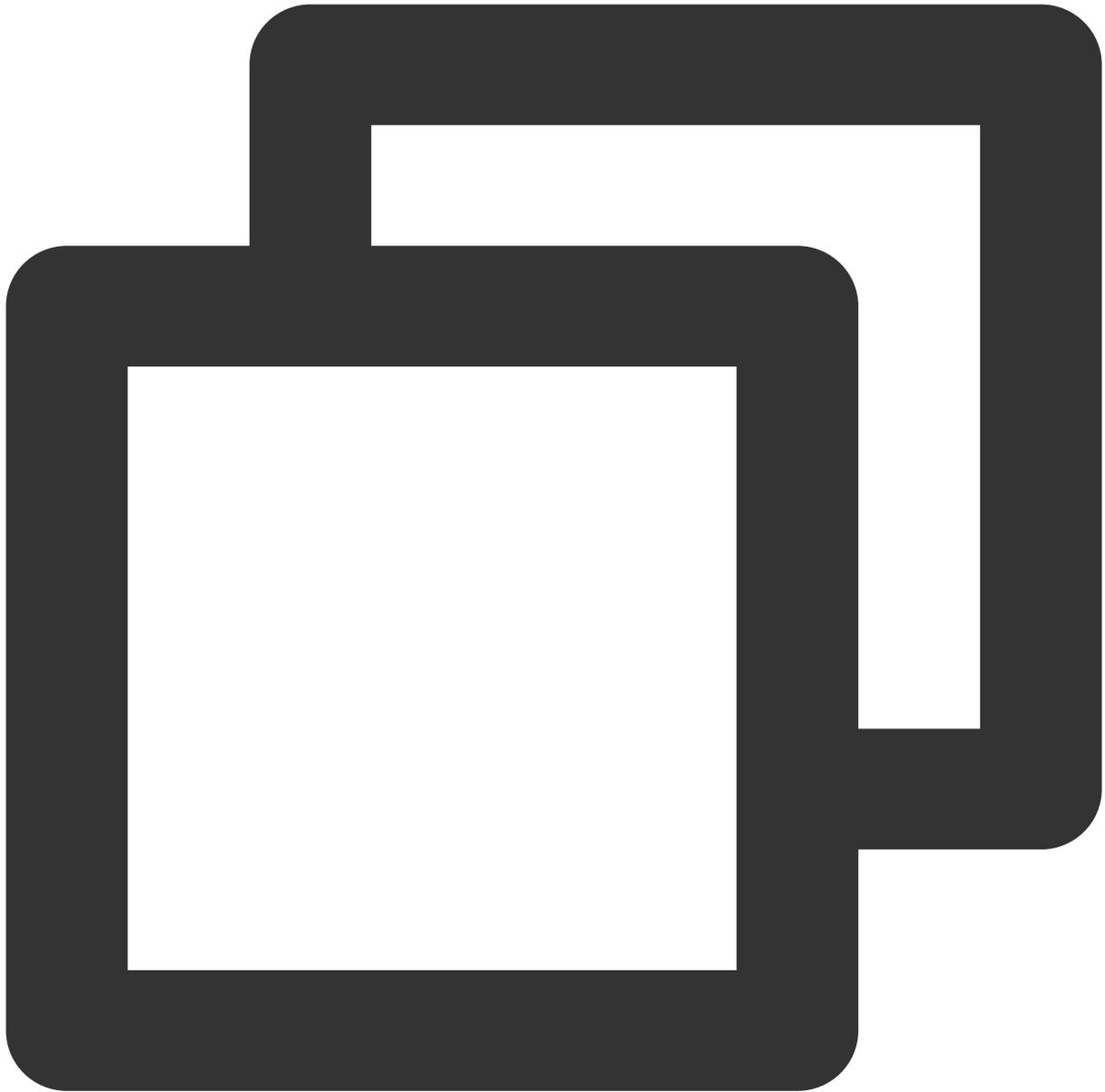
使用 `config` 命令生成配置文件

注意：

建议用户通过 [临时密钥](#) 使用工具，通过临时授权的方式进一步提高工具使用的安全性。申请临时密钥时，请遵循 [最小权限指引原则](#)，防止泄露目标存储桶或对象之外的资源。

如果您一定要使用永久密钥，建议遵循 [最小权限指引原则](#) 对永久密钥的权限范围进行限制。

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



```
coscmd config [OPTION]...<FILE>...
  [-h] --help
  [-a] <SECRET_ID>
  [-s] <SECRET_KEY>
  [-t] <TOKEN>
  [-b] <BucketName-APPID>
  [-r] <REGION> | [-e] <ENDPOINT>
  [-m] <MAX_THREAD>
  [-p] <PART_SIZE>
  [--do-not-use-ssl]
  [--anonymous]
```

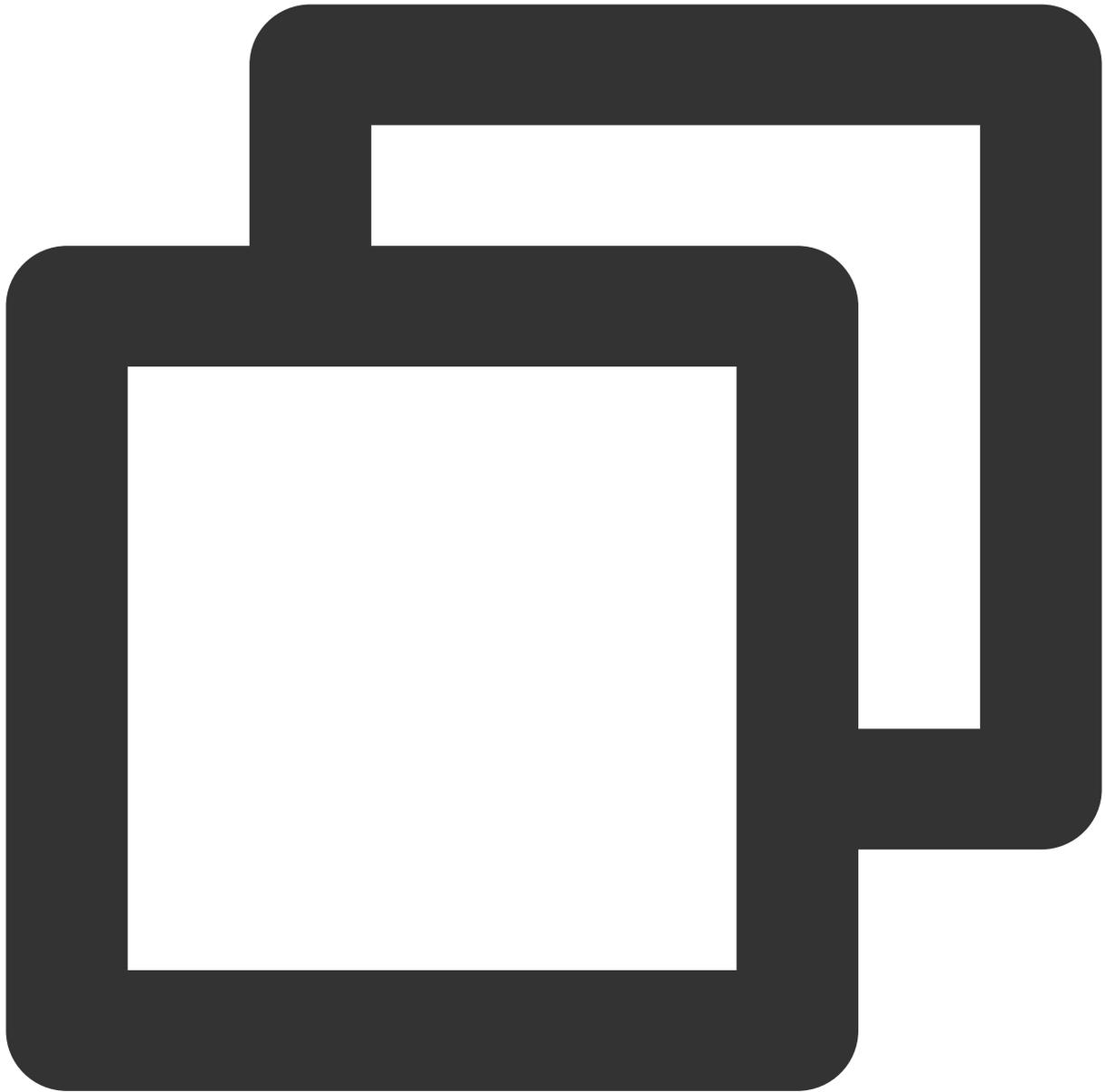
说明：

其中 "[]" 中的字段为选项， "<>" 的字段为需要填写的参数。

参数配置说明如下：

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

config 命令的使用示例如下：



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

通用命令

指定 Bucket 和 Region 的命令

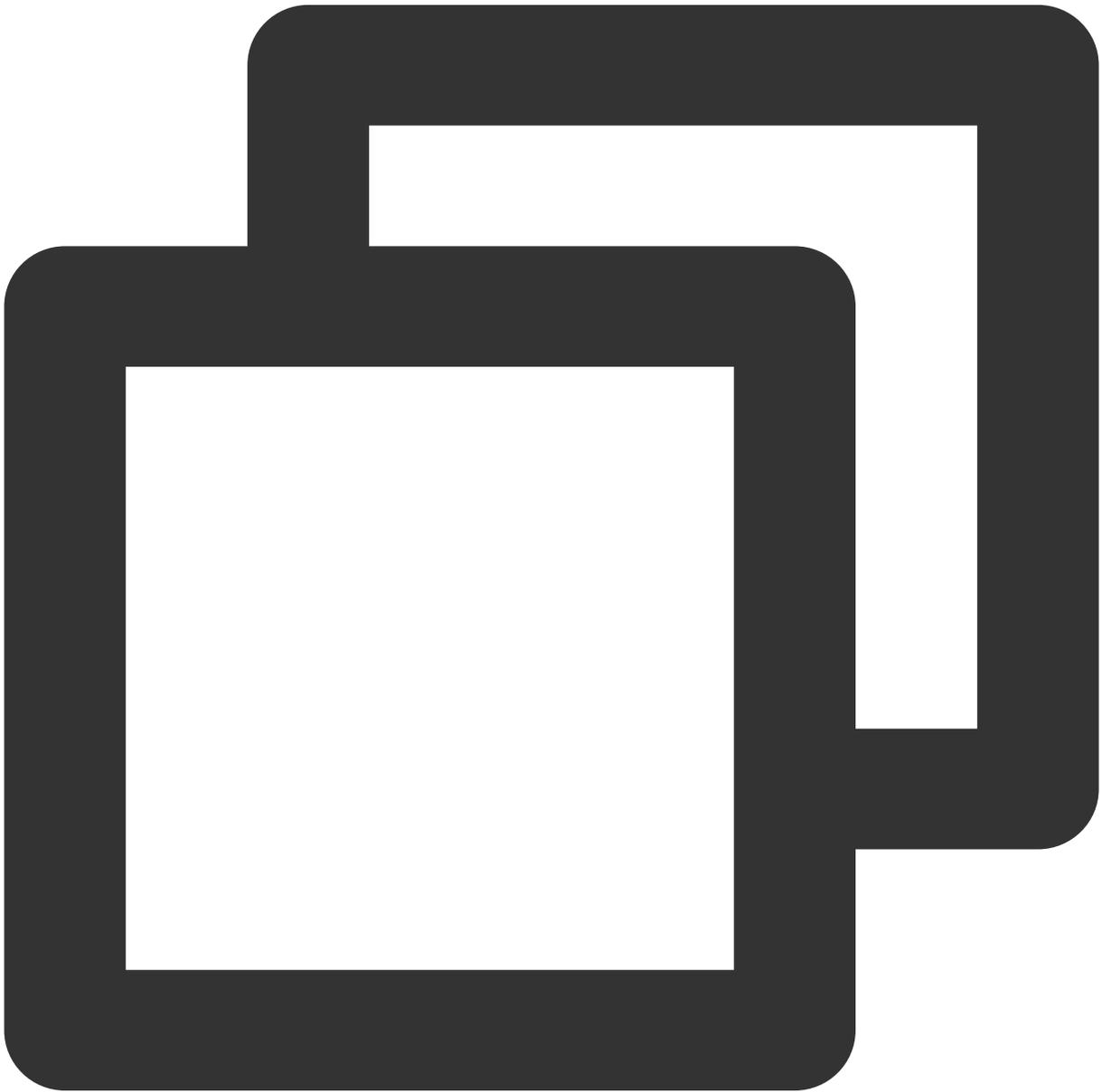
用户若不指定存储桶名称和所属地域执行命令，则默认对配置参数时所填的存储桶有效。当需要对不同存储桶执行操作时，需要指定存储桶名称和所属地域。

说明：

通过 `-b <BucketName-APPID>` 参数指定存储桶名称，存储桶的命名格式为 `BucketName-APPID`，此处填写的存储桶名称必须为此格式。

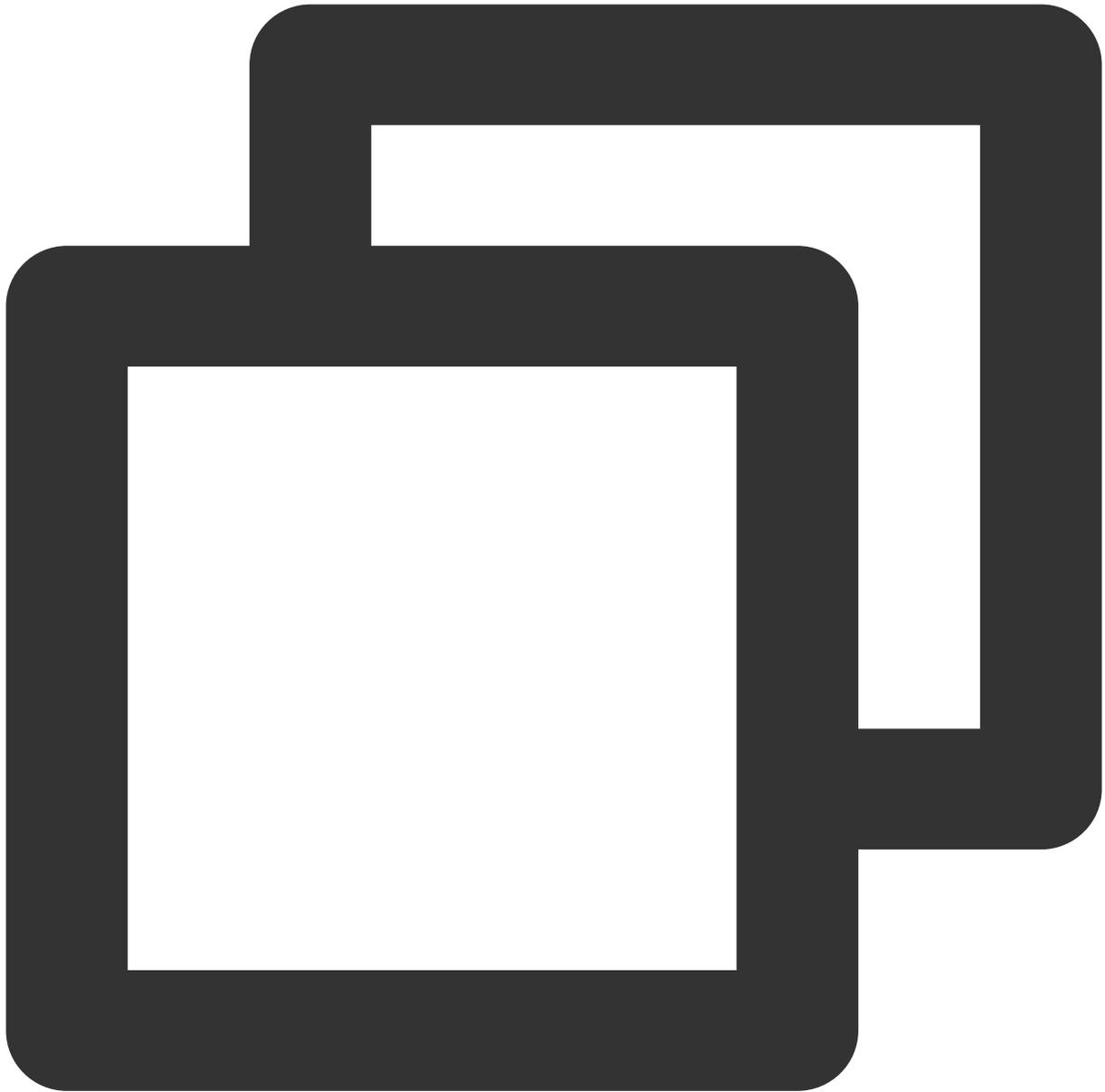
通过 `-r <region>` 指定 `Region`，可以指定存储桶的所属地域。

命令格式



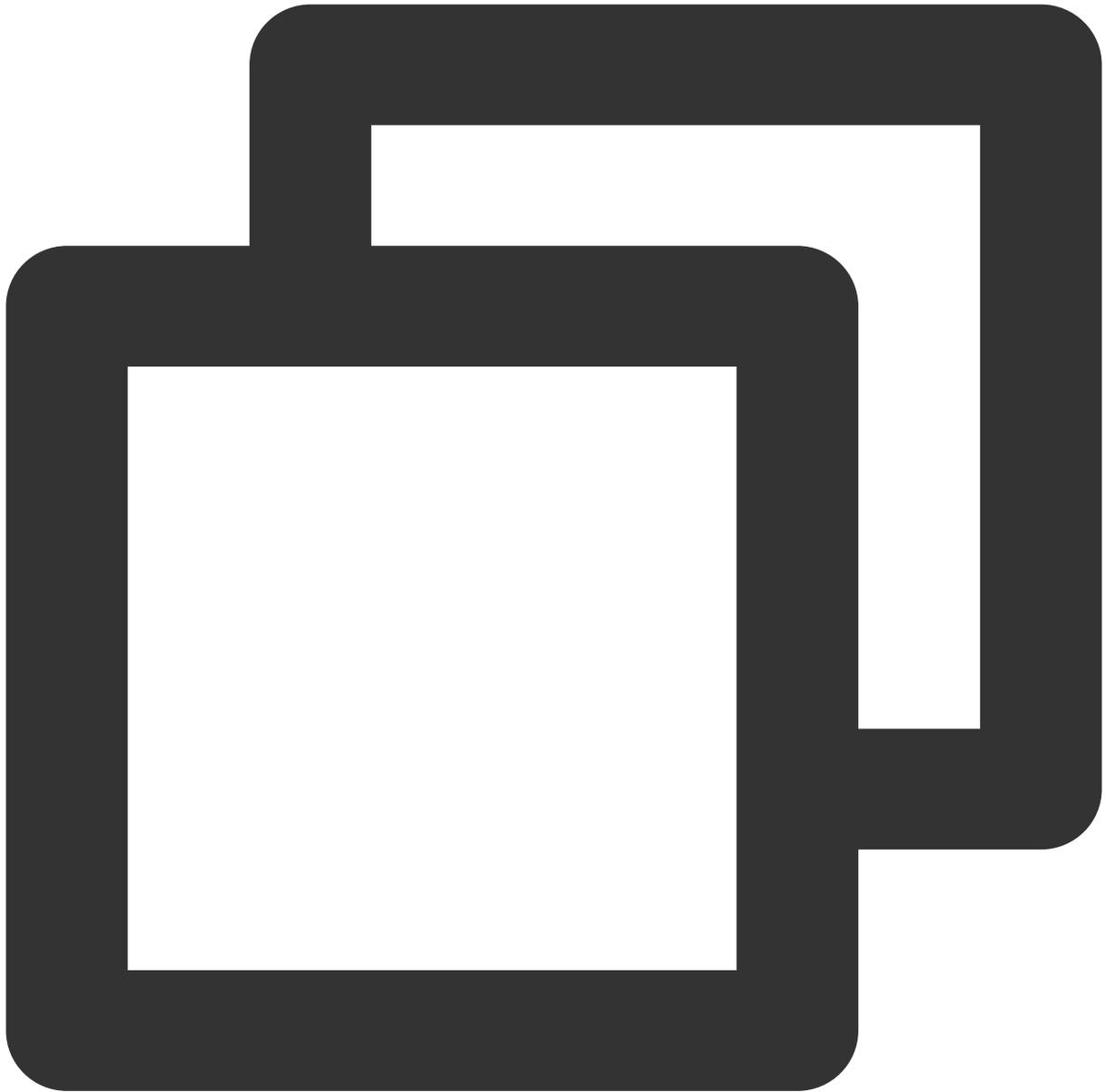
```
coscmd -b <BucketName-APPID> -r <region> <action> ...
```

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



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

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



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

指定配置文件和日志文件路径的命令

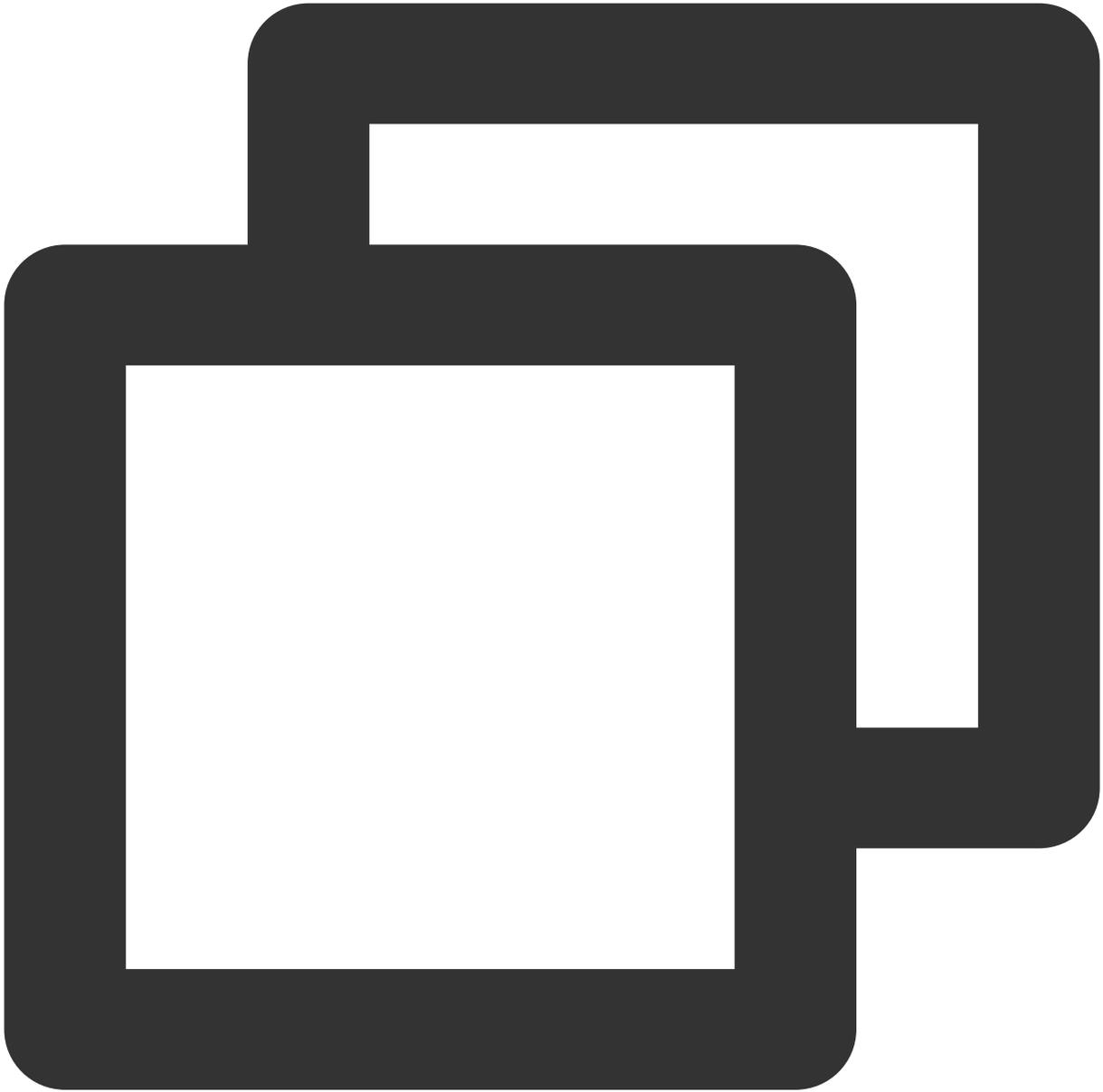
用户若不指定配置文件的路径，则会使用默认的配置文件的文件路径 `~/.cos.conf`。若不指定日志文件的路径，则会使用默认的日志文件的文件路径 `~/.cos.log`。

说明：

通过 `-c <conf_path>` 参数指定配置文件路径，COSCMD 在运行时从此路径读取配置信息。

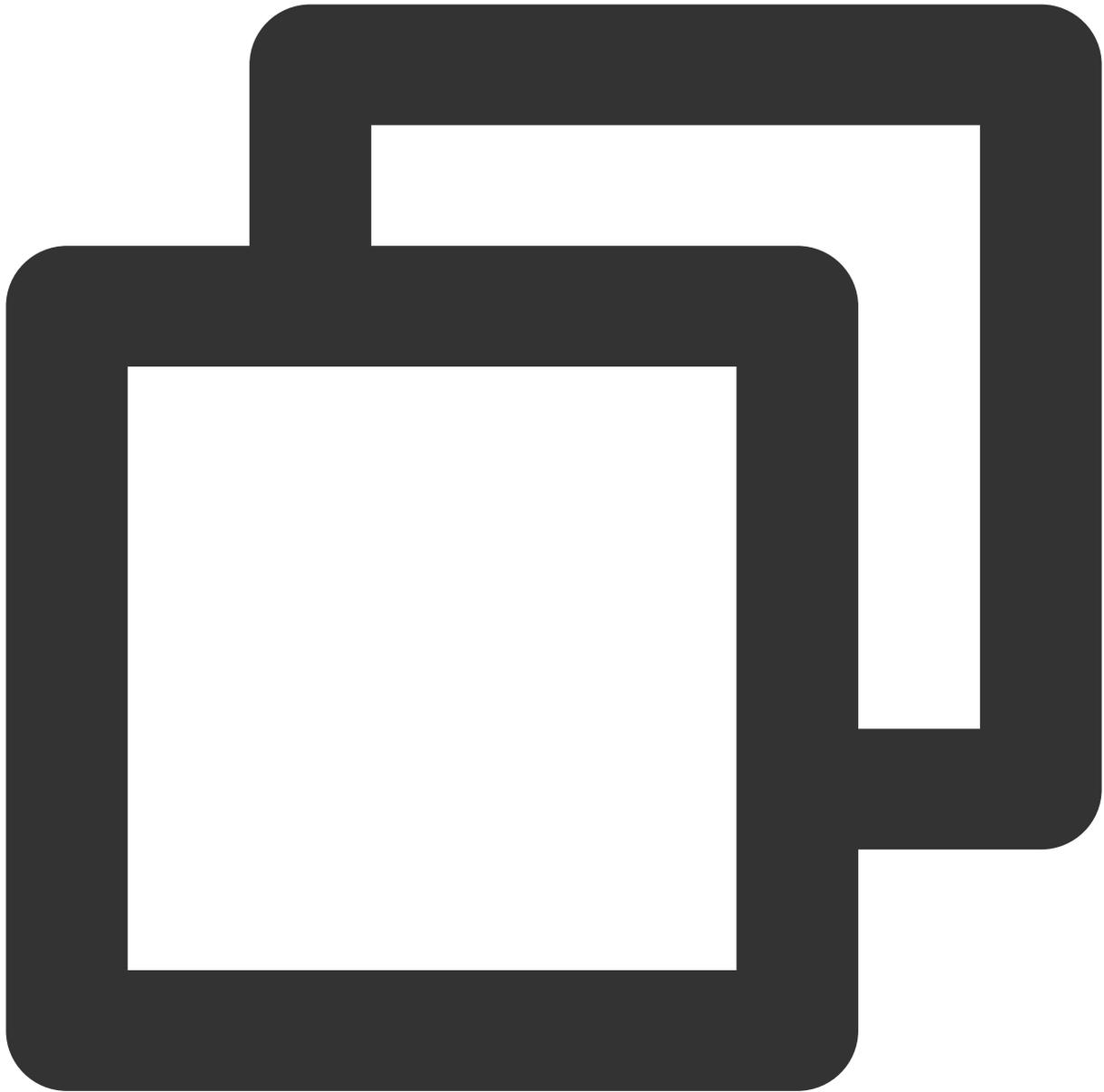
通过 `-l <log_conf>` 参数指定日志路径，COSCMD 会将运行过程中产生的日志输出到此路径下的日志文件中。

命令格式



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

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

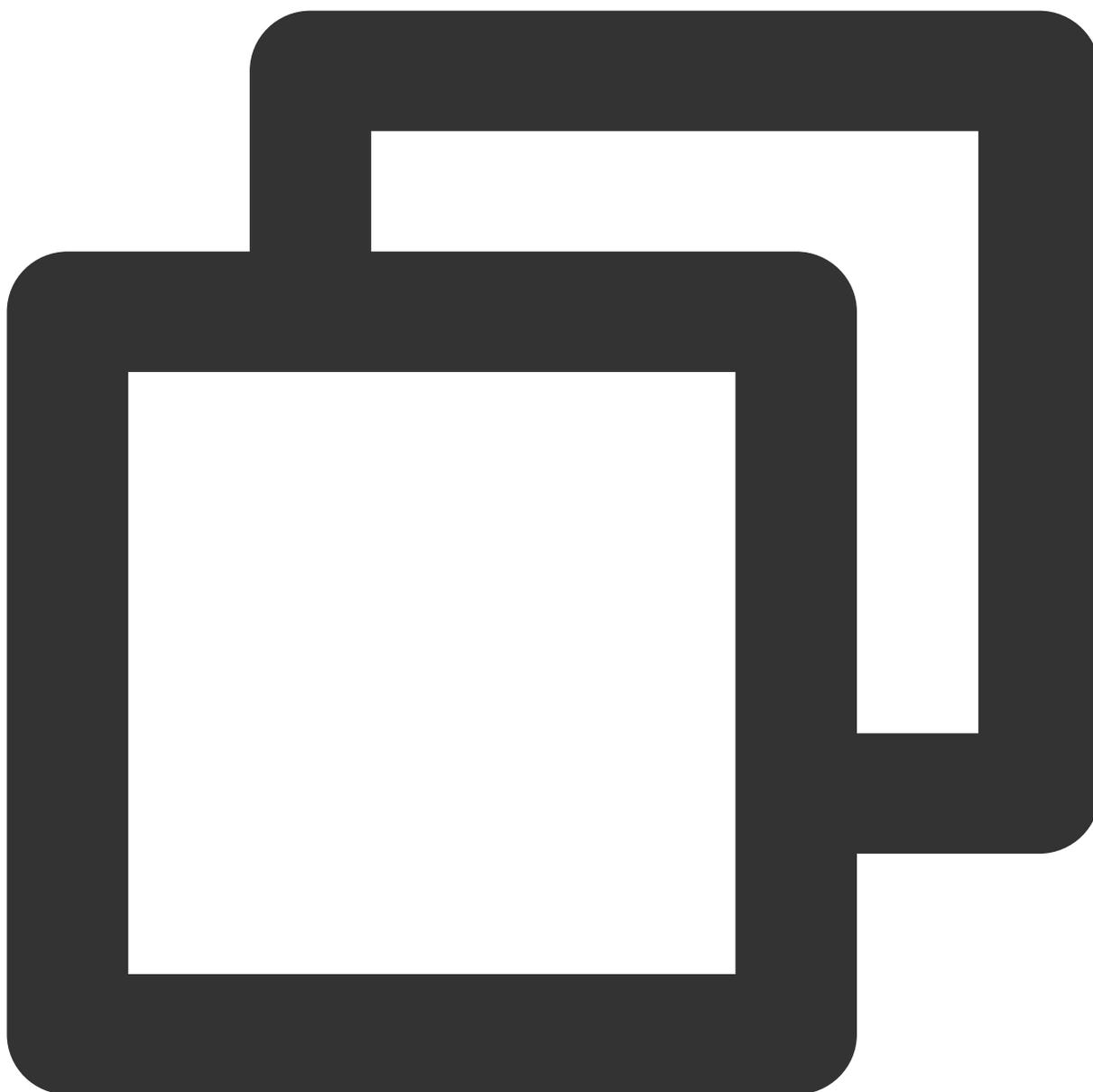


```
coscmd -c /data/home/cos_conf -l /data/home/cos_log -b examplebucket-1250000000 -r
```

Debug 模式执行命令

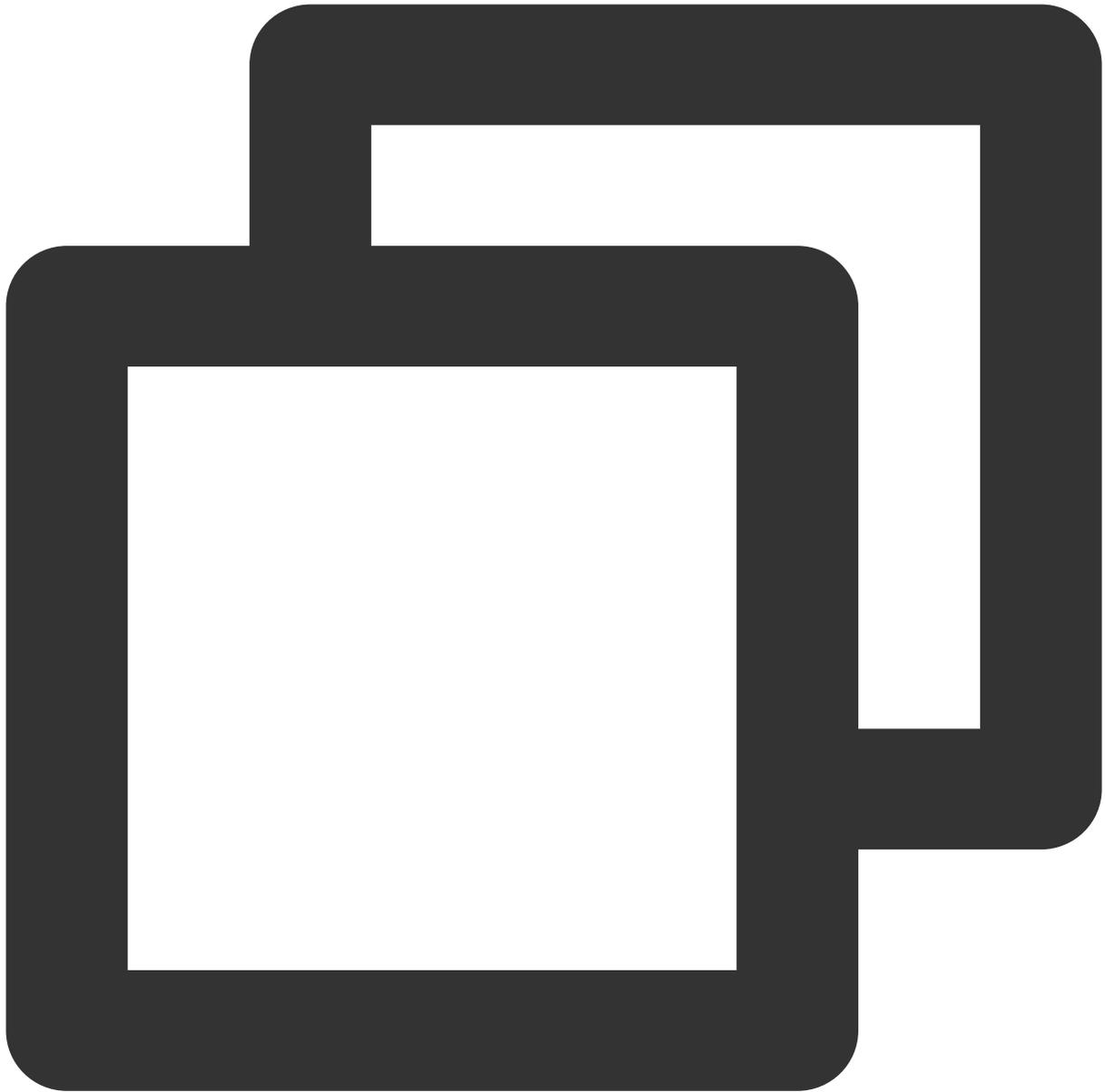
在各命令前加上 `-d` 或者 `--debug`，在命令执行的过程中，会显示详细的操作信息。示例如下：

命令格式



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

操作示例 - 上传时输出详细信息



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

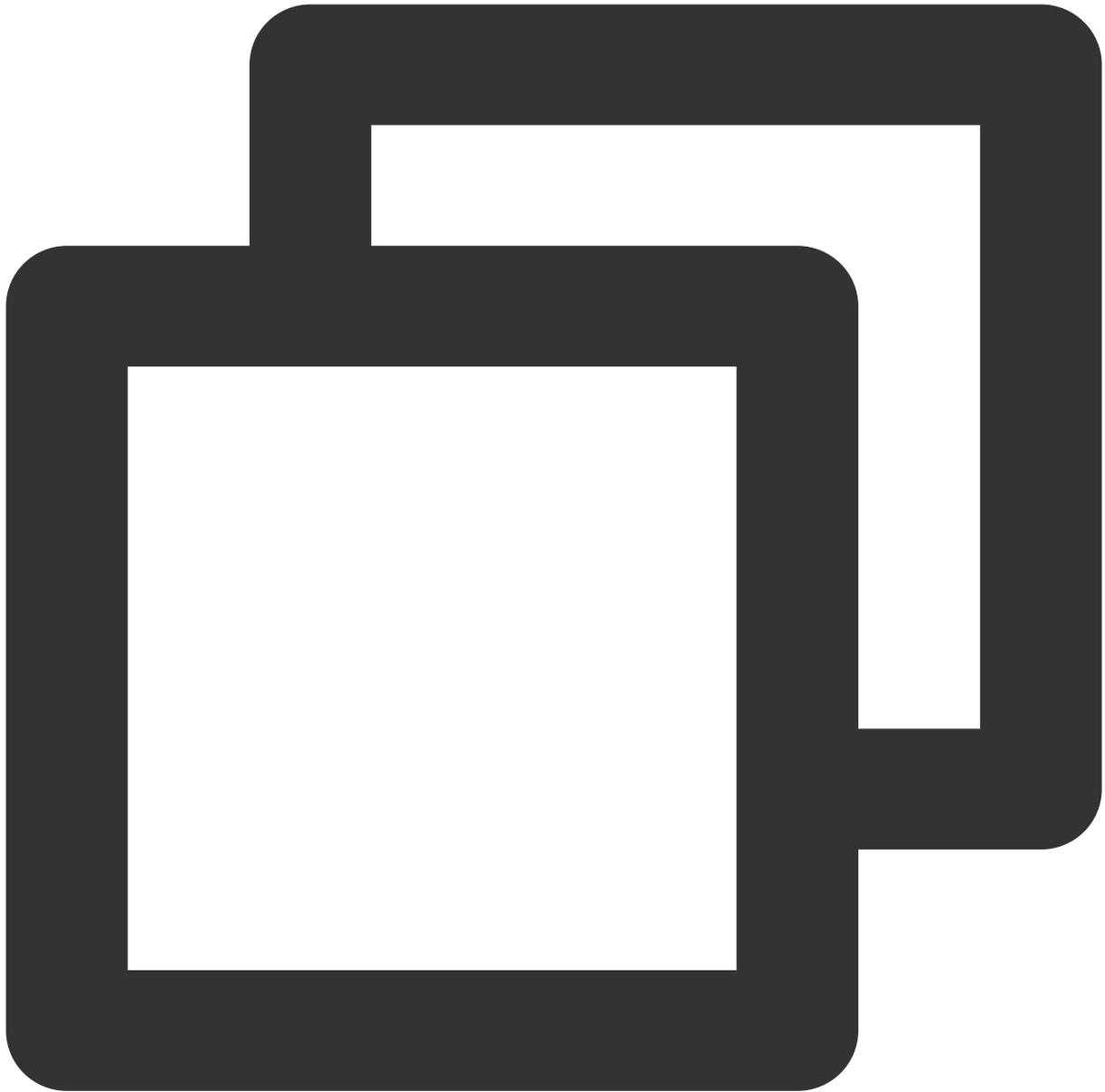
Silence 模式执行命令

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

说明：

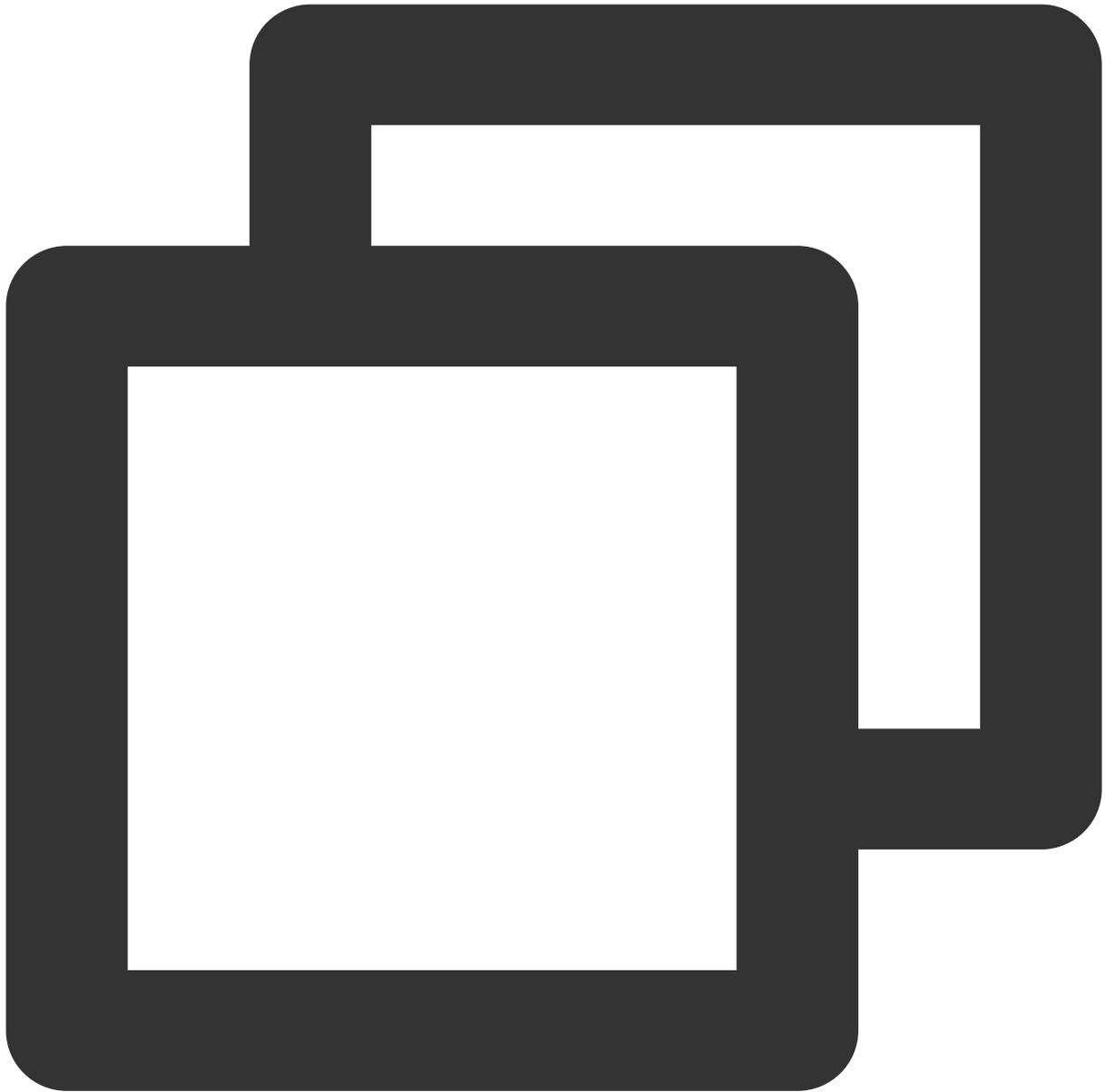
该命令需满足最低版本为1.8.6.24。

命令格式



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

操作示例



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

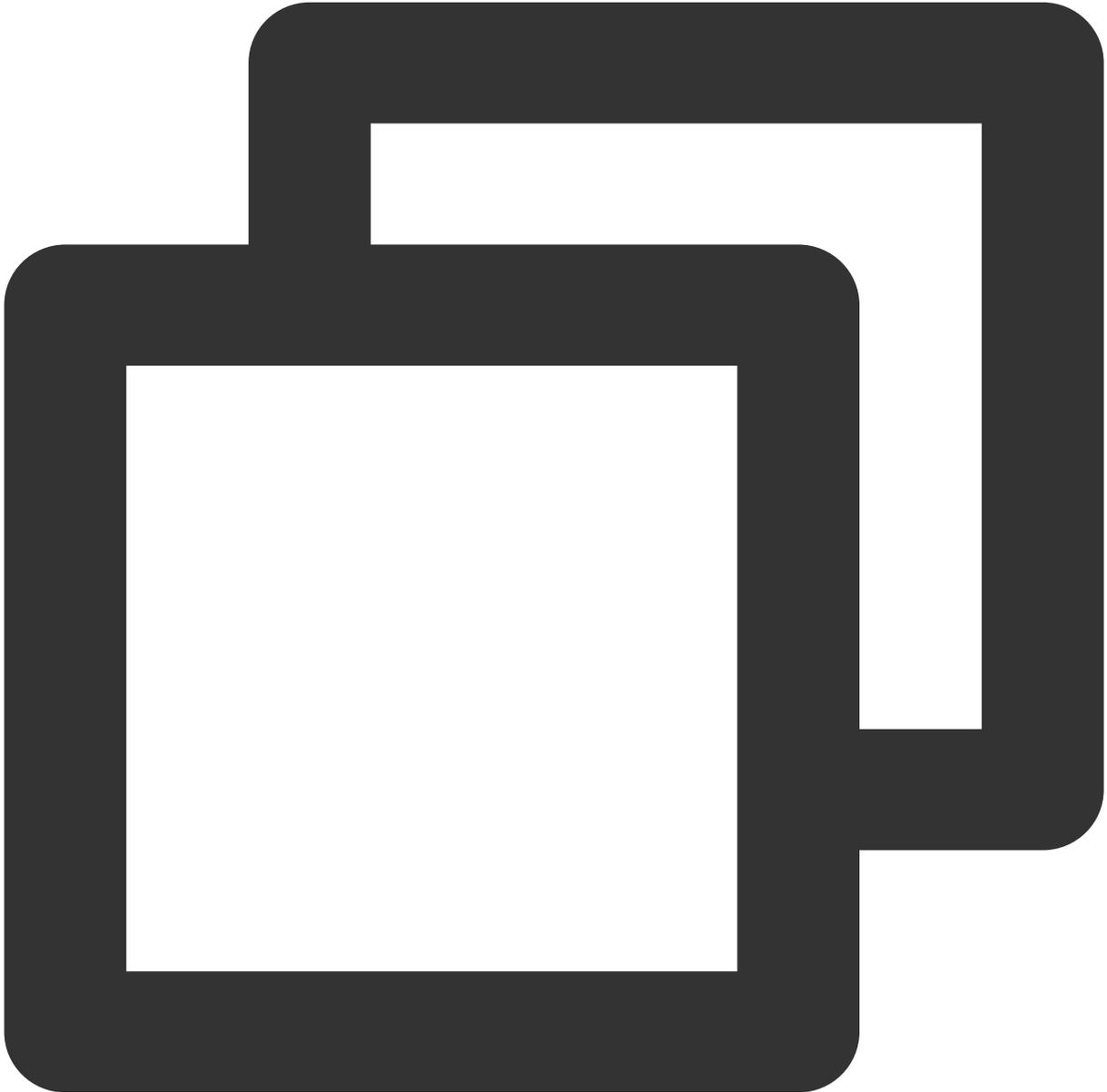
常用存储桶命令

创建存储桶

说明：

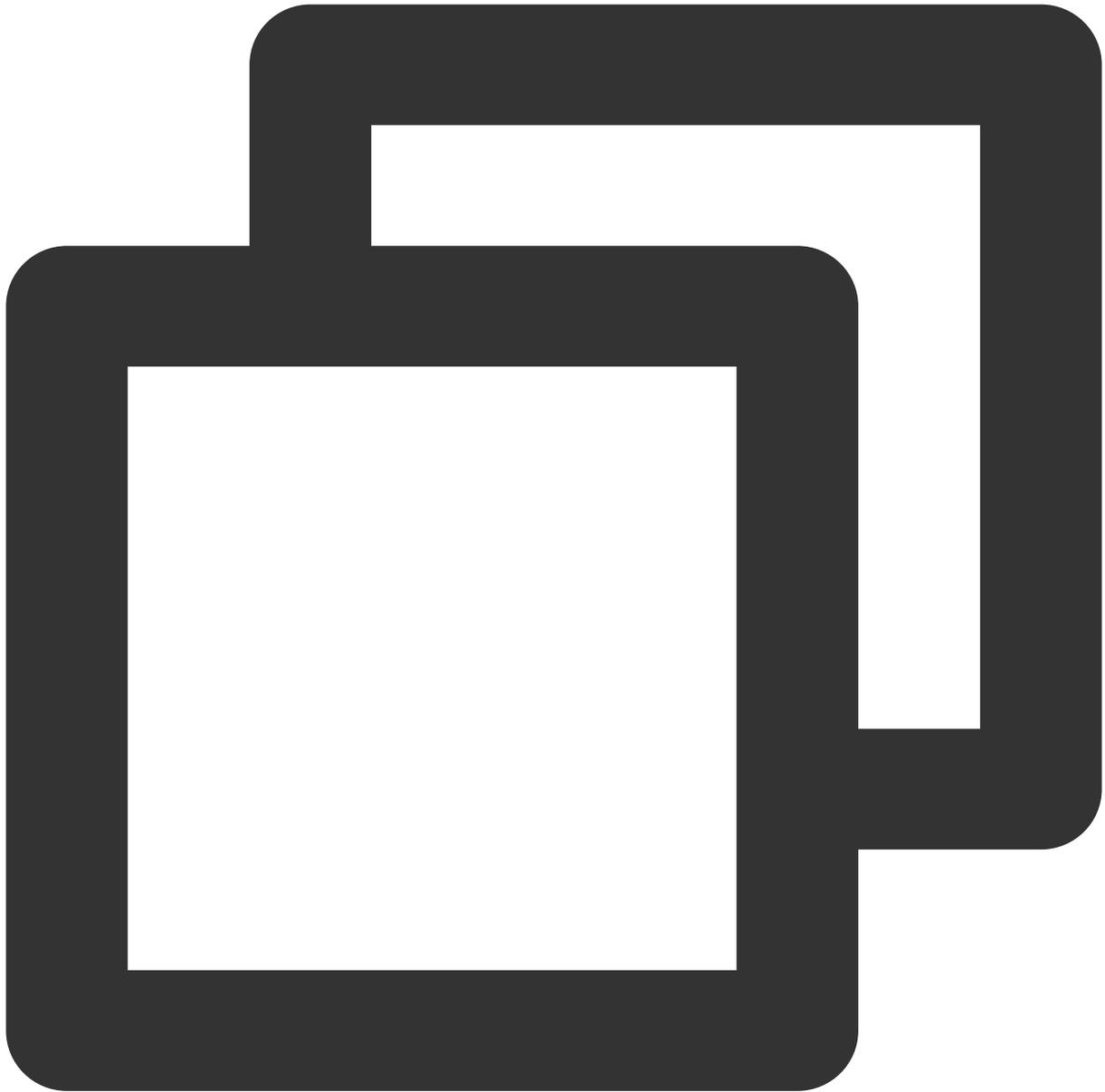
执行创建存储桶命令时，请携带参数 `-b <BucketName-APPID>` 指定存储桶名称和 `-r <Region>` 指定所属地域。若直接执行 `coscmd createbucket`，则会报错，原因是不指定存储桶名称和所属地域，则相当于对已存在的存储桶（即配置参数时所填的存储桶）进行创建。

命令格式



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

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



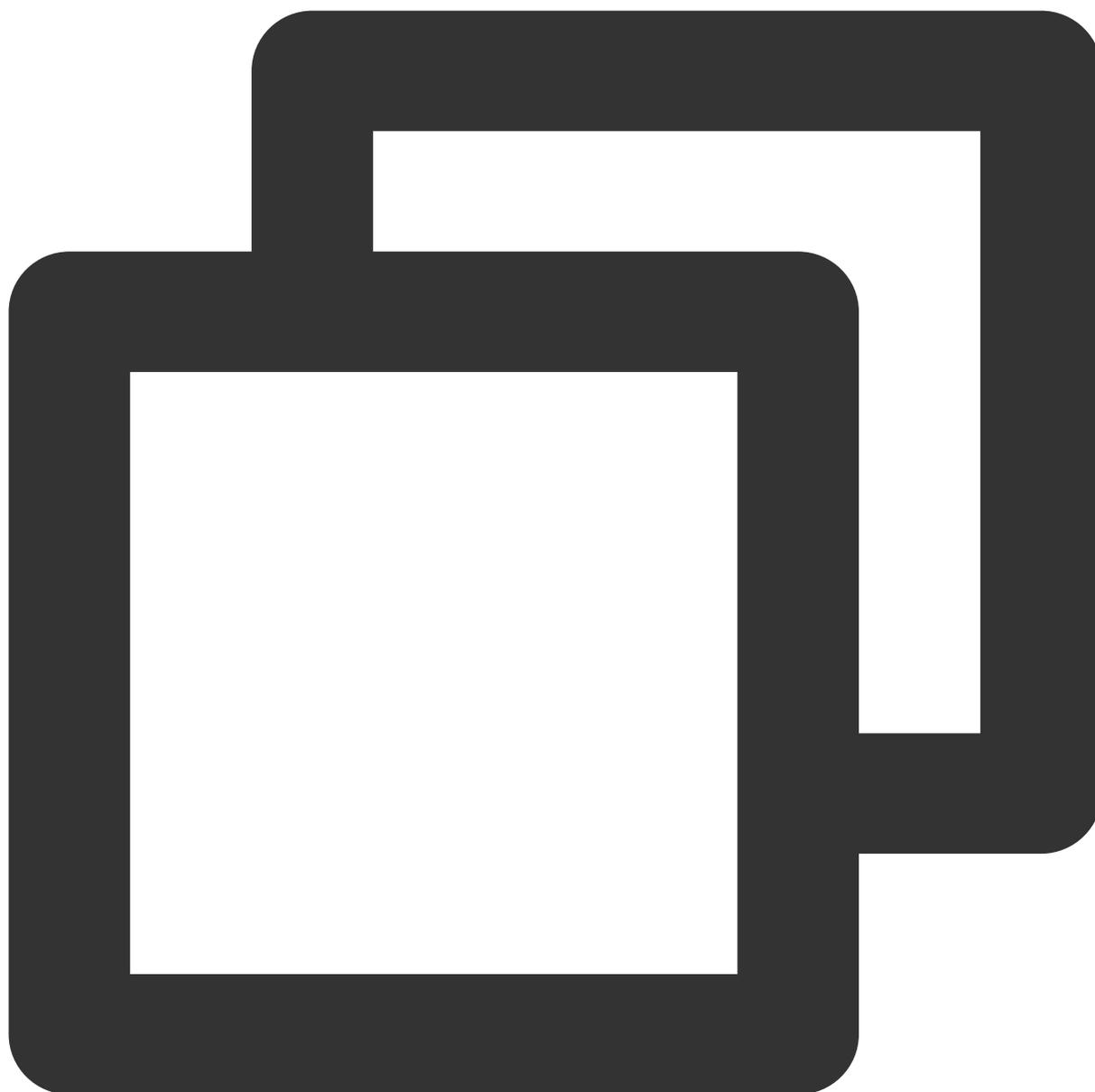
```
coscmd -b examplebucket-1250000000 -r ap-beijing createbucket
```

删除存储桶

说明：

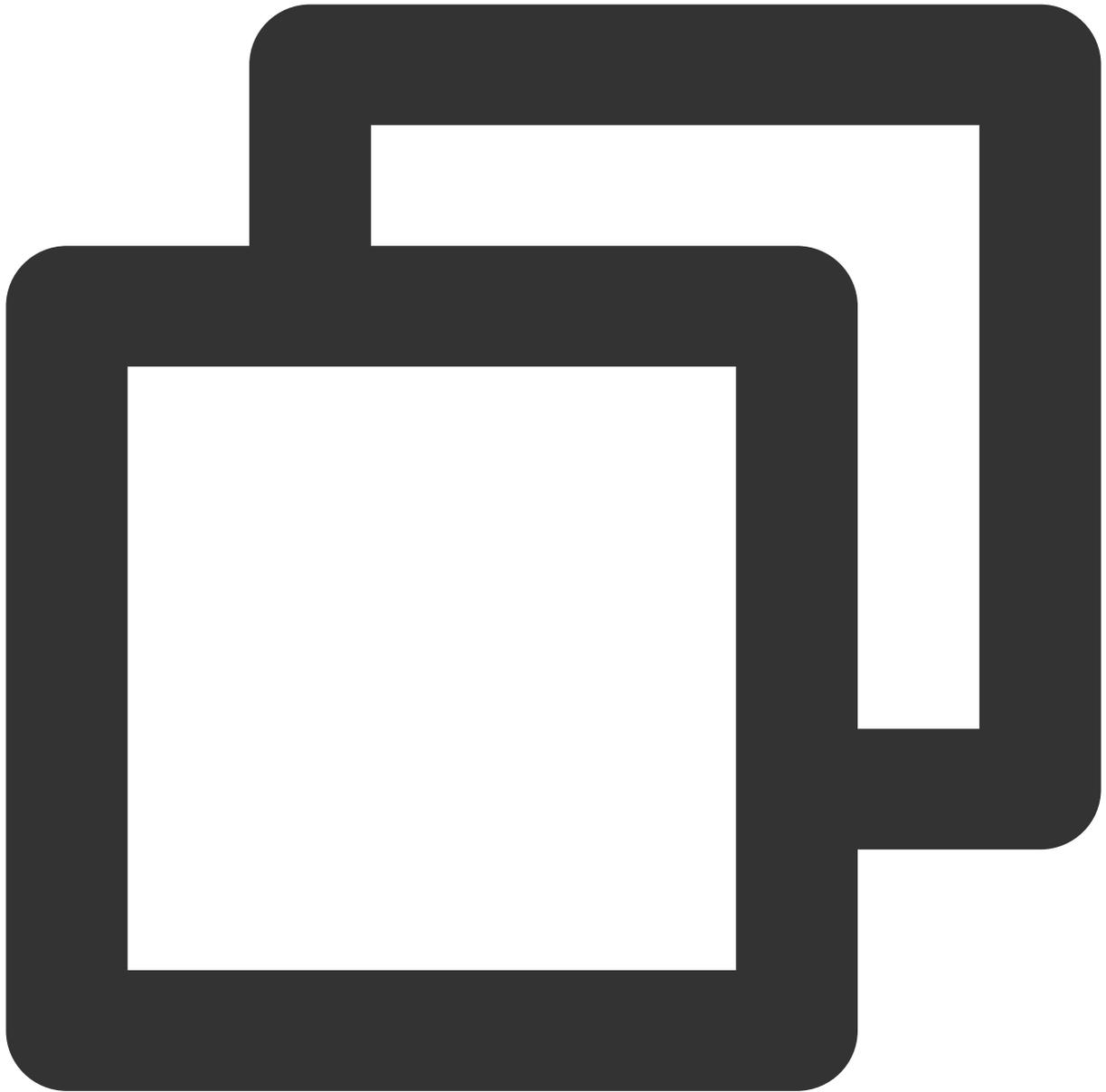
`coscmd deletebucket` 的用法仅对配置参数时的存储桶有效。建议配合 `-b <BucketName-APPID>` 指定 Bucket 和 `-r <region>` 指定 Region 使用。

命令格式



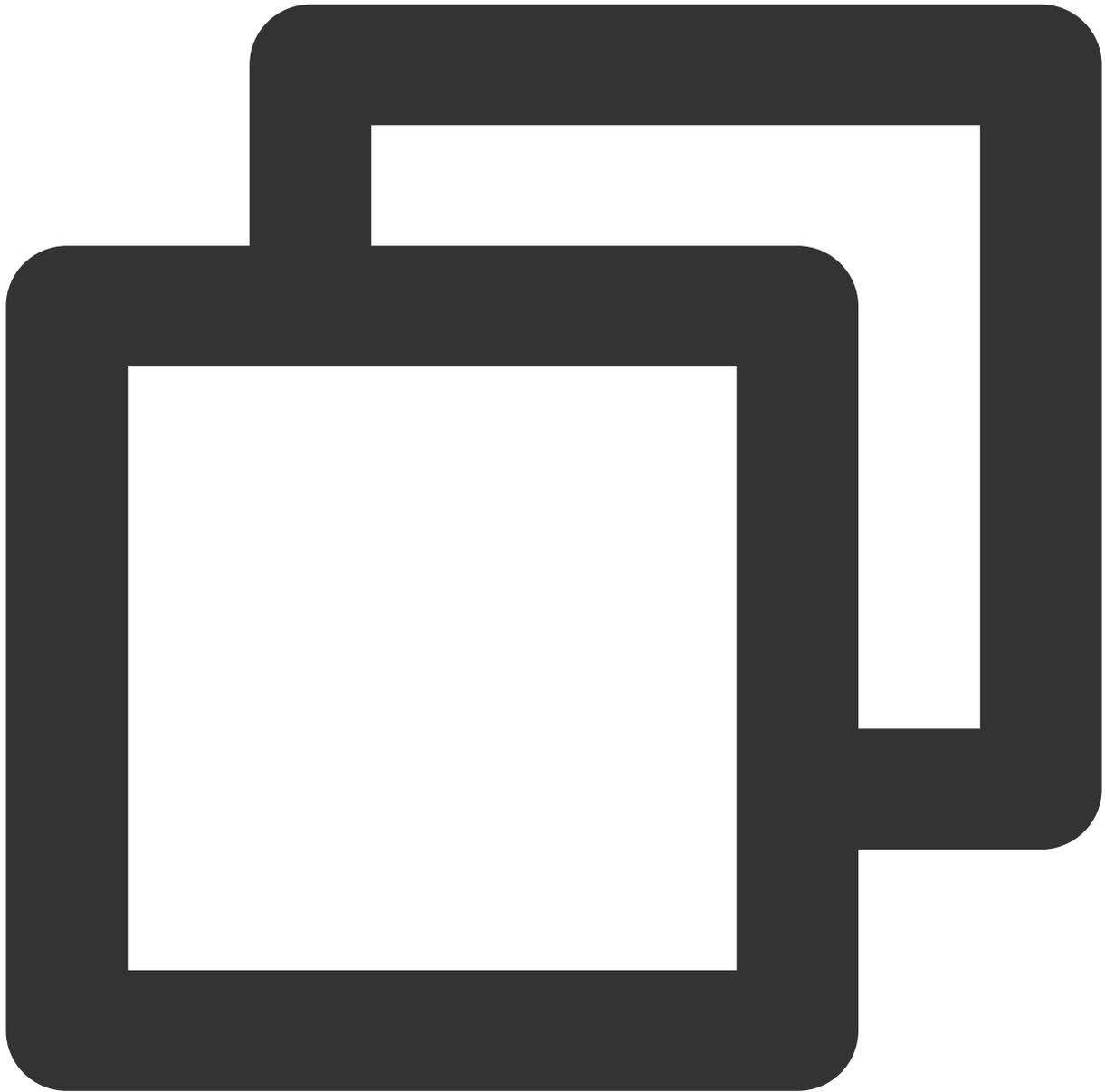
```
coscmd -b <BucketName-APPID> deletebucket
```

操作示例 - 删除空的存储桶



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

操作示例 - 强制删除非空的存储桶



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

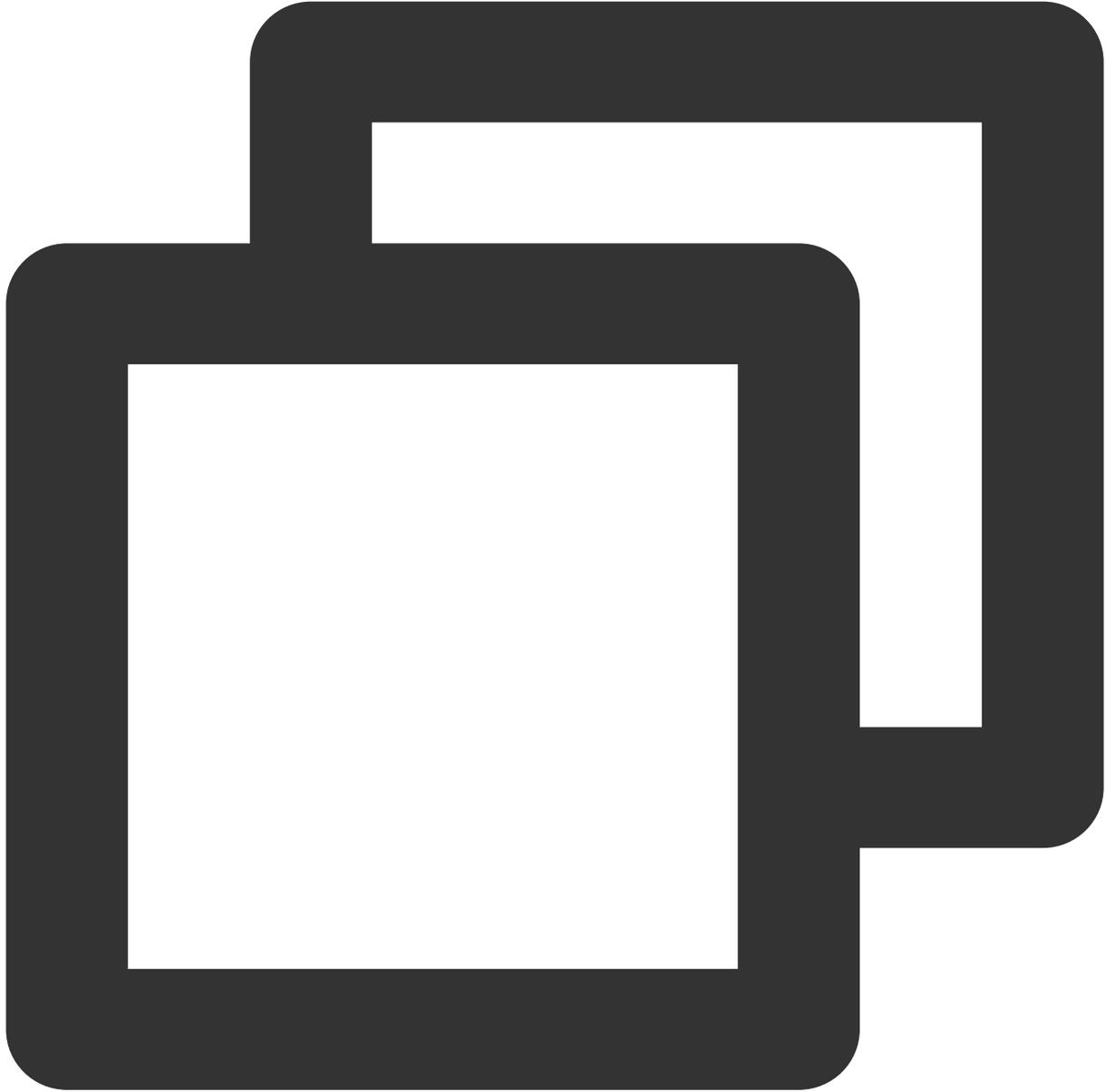
注意：

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

常用对象命令

上传文件

上传文件命令格式

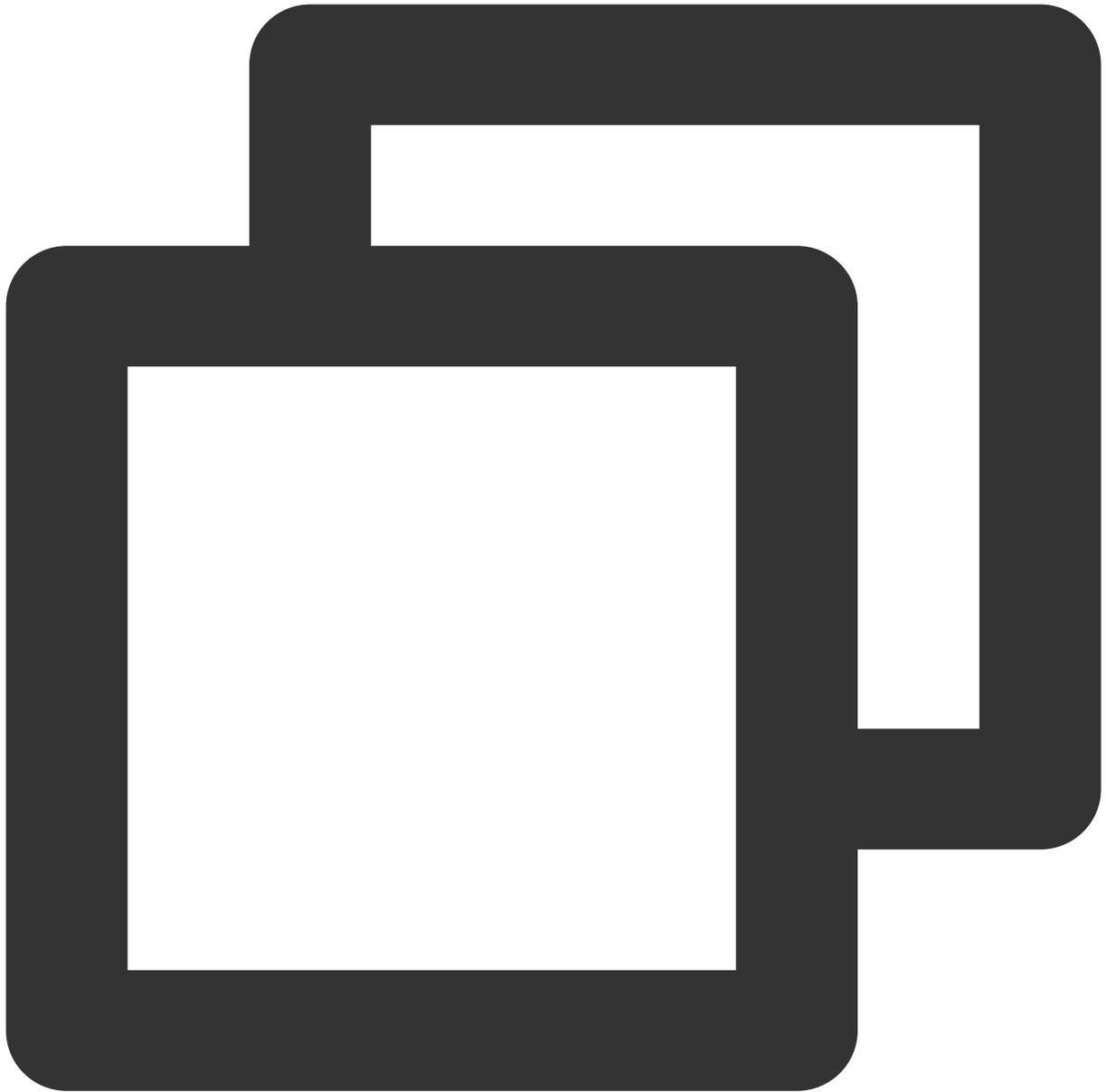


```
coscmd upload <localpath> <cospath>
```

注意：

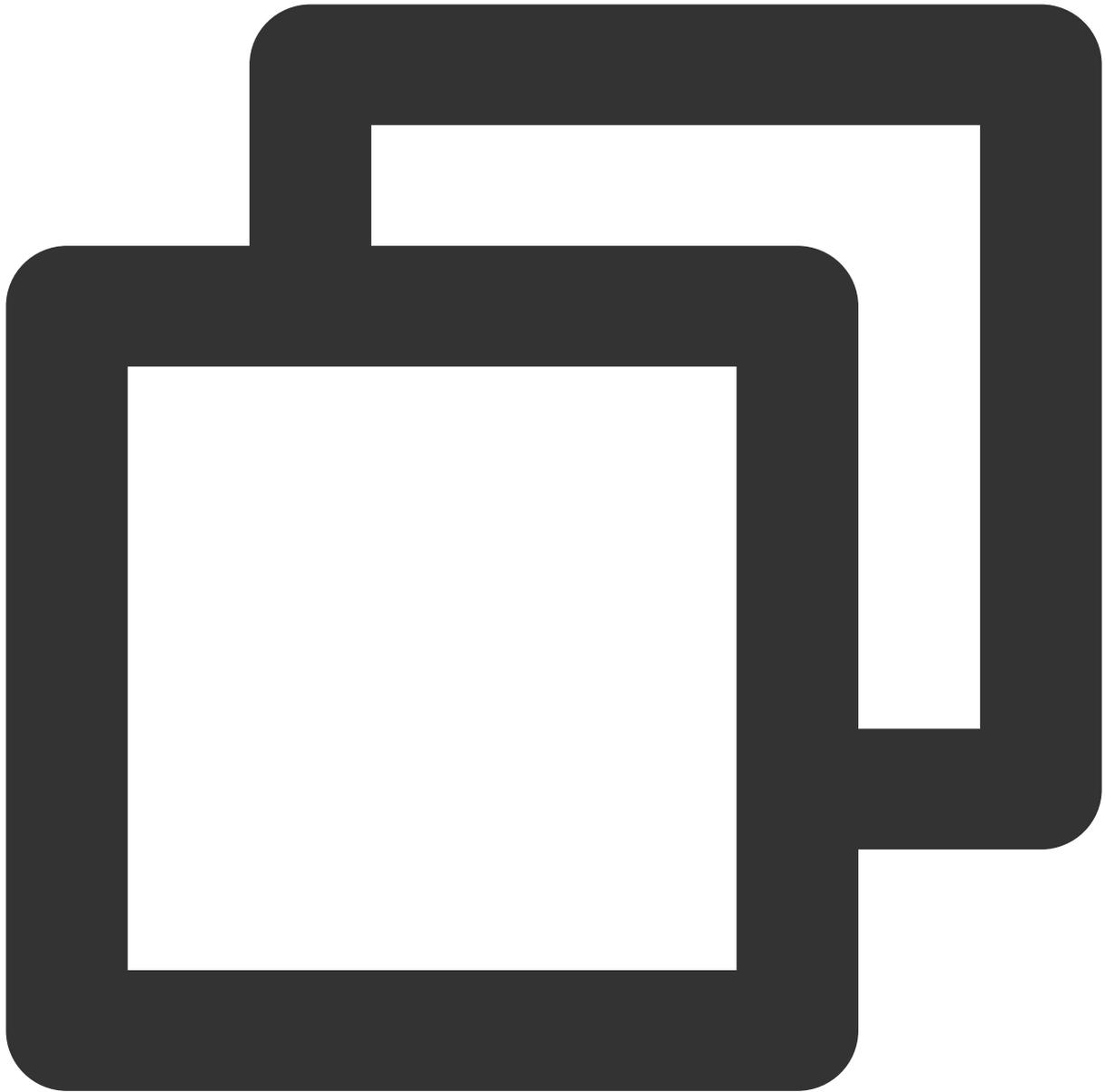
请将 "<>" 中的参数替换为您需要上传的本地文件路径（localpath），以及 COS 上存储的路径（cospath）。

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



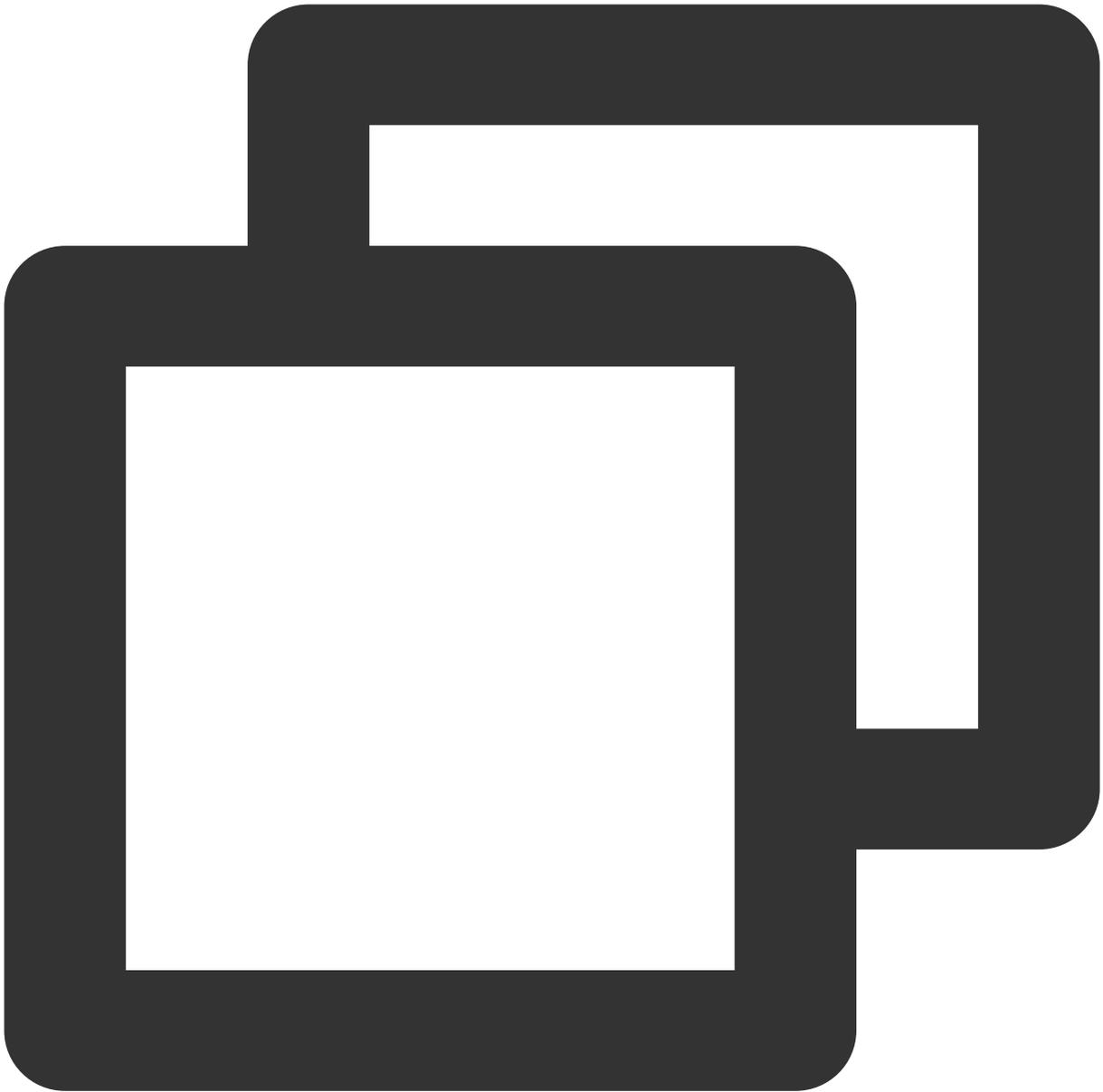
```
coscmd upload D:/picture.jpg doc/
```

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



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

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

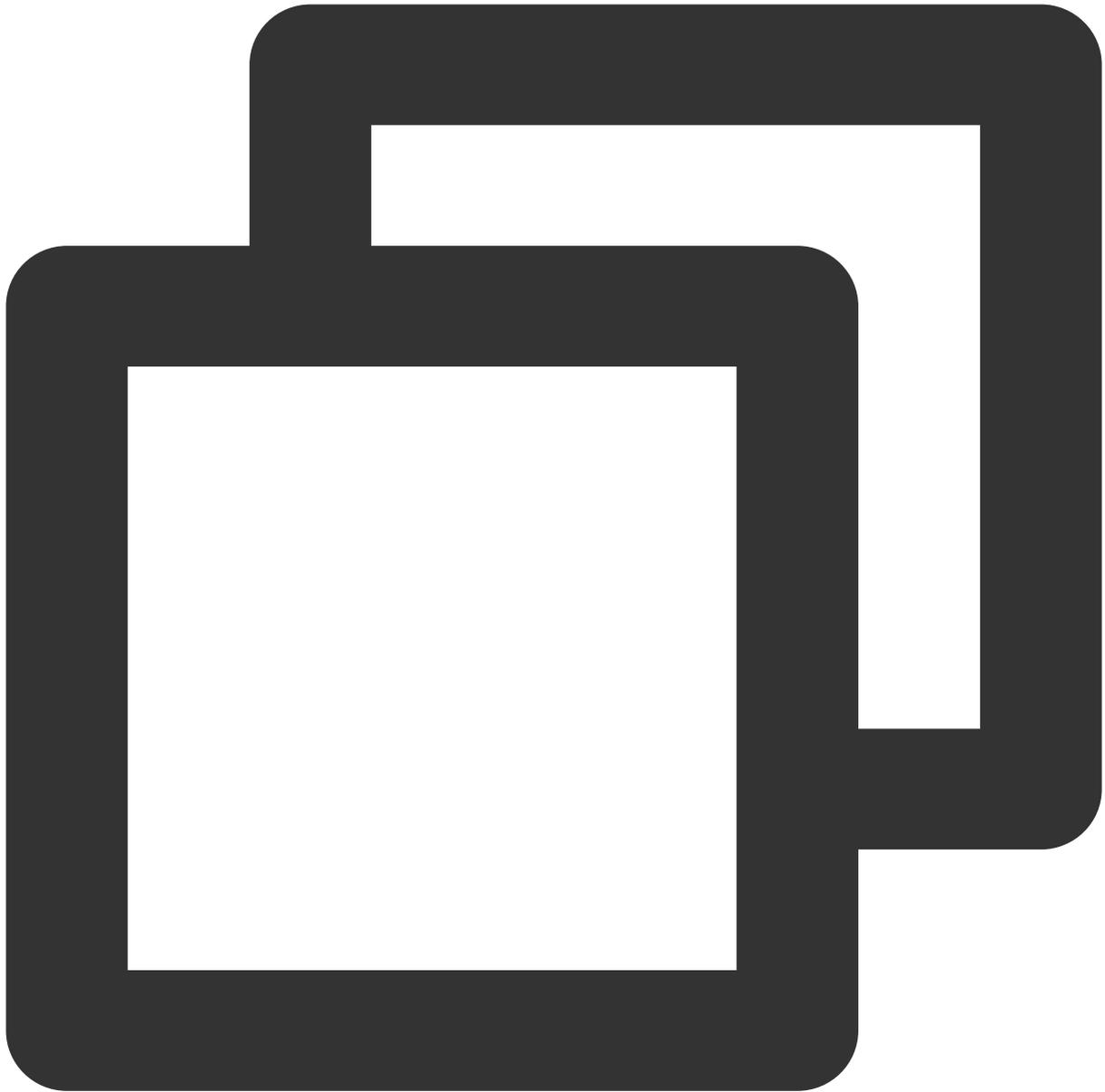


```
coscmd upload D:/picture.jpg doc/ -H '{"x-cos-storage-class':'Archive'}"
```

注意：

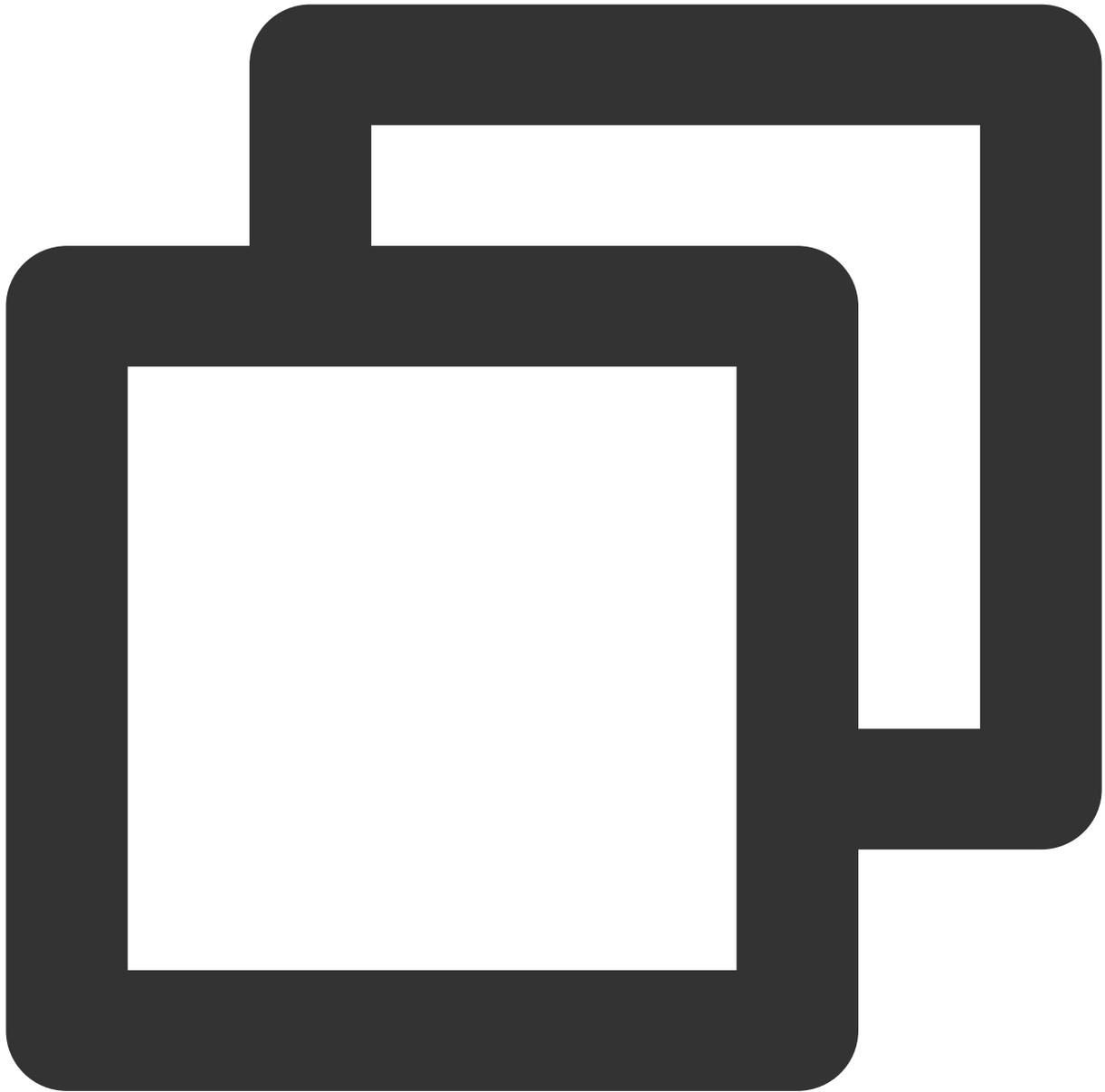
使用 `-H` 参数设置 HTTP header 时，请务必保证格式为 JSON，示例：`coscmd upload -H '{"x-cos-storage-class':'Archive','Content-Language':'zh-CN'}' <localpath> <cospath>`。更多头部可参见 [PUT Object](#) 文档。

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



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

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



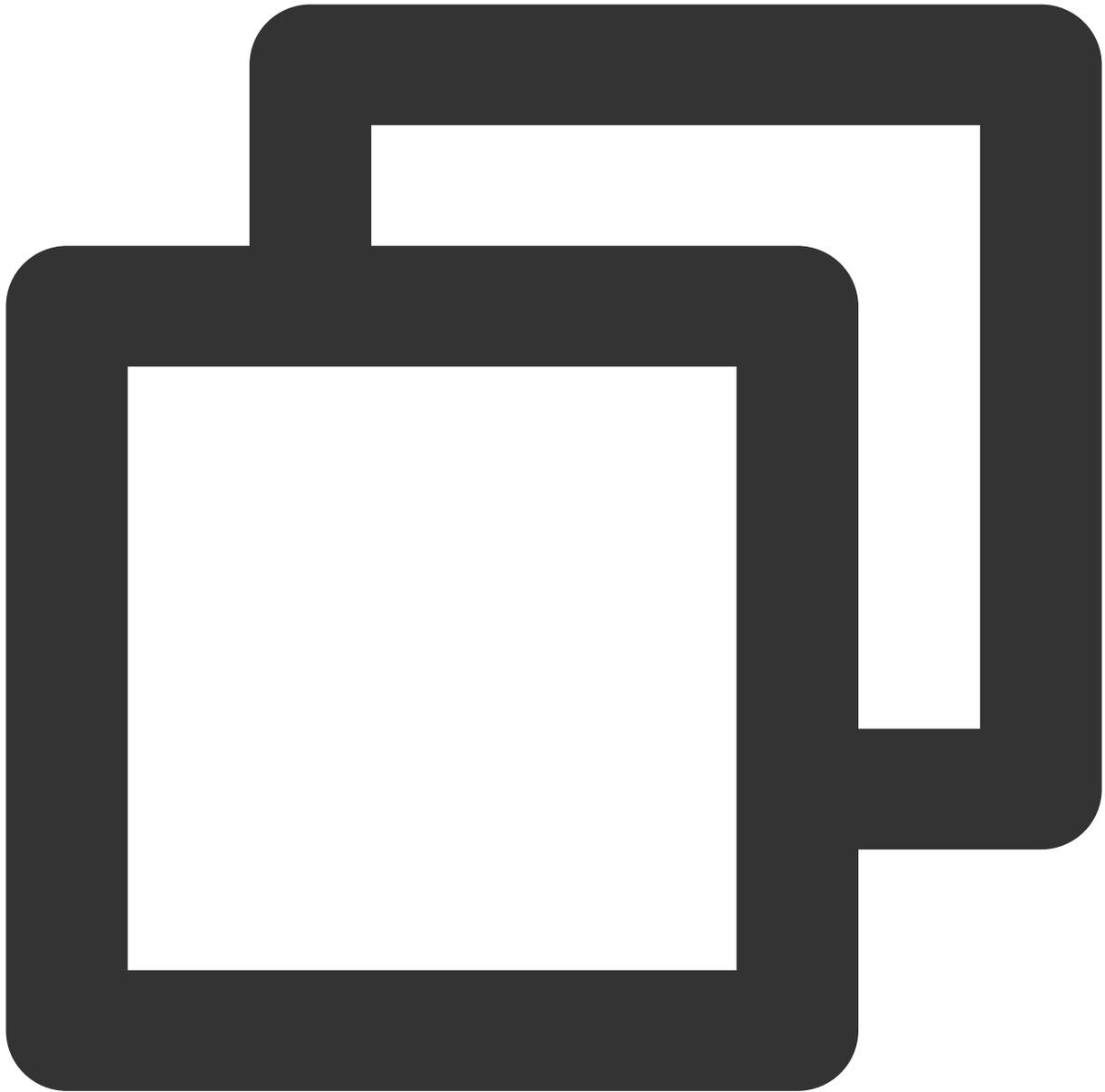
```
coscmd upload D:/doc/file.zip doc/ -H '{"x-cos-traffic-limit':'819200'}"
```

说明：

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

上传文件夹

上传文件夹命令格式

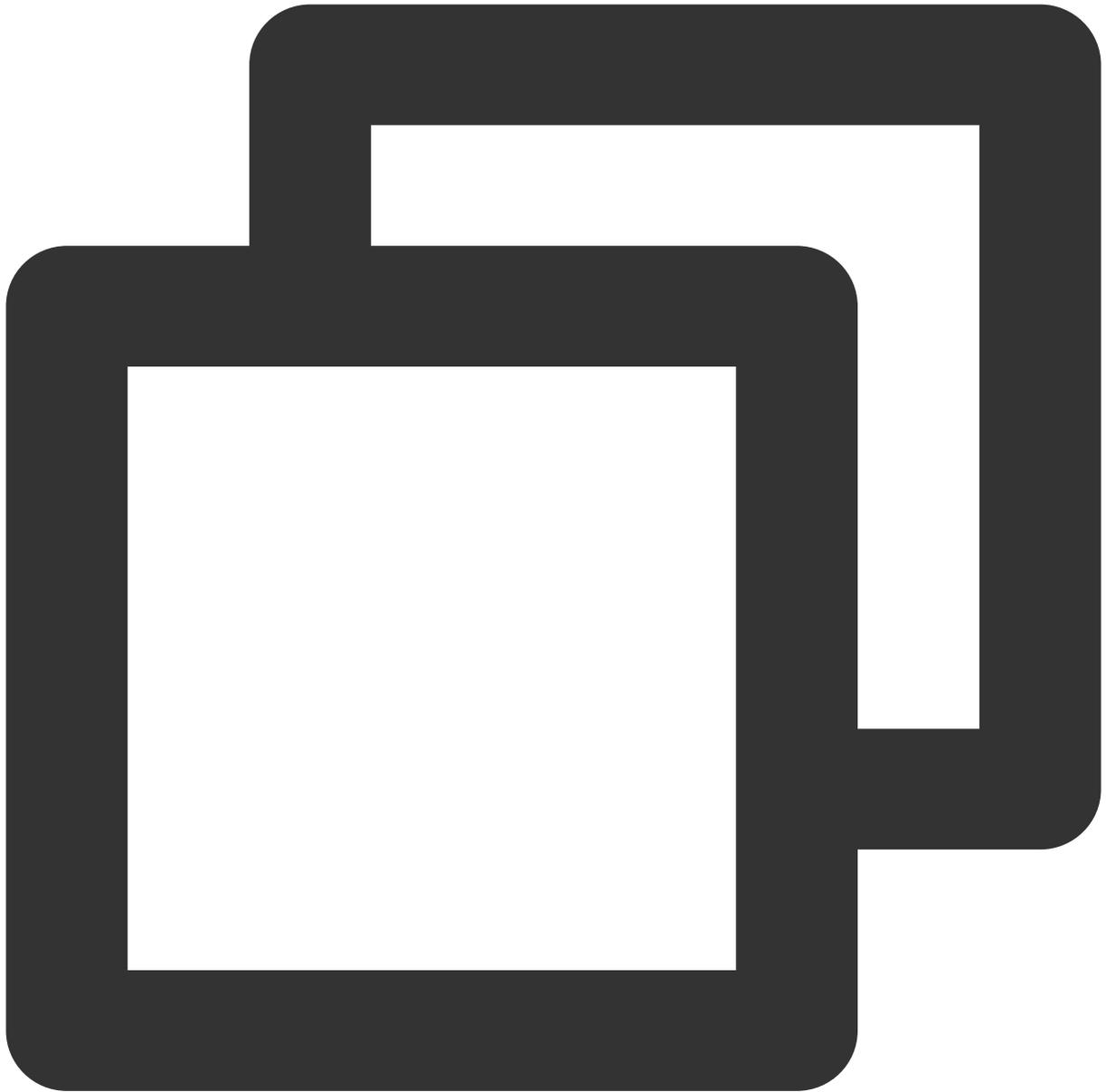


```
coscmd upload -r <localpath> <cospath>
```

注意：

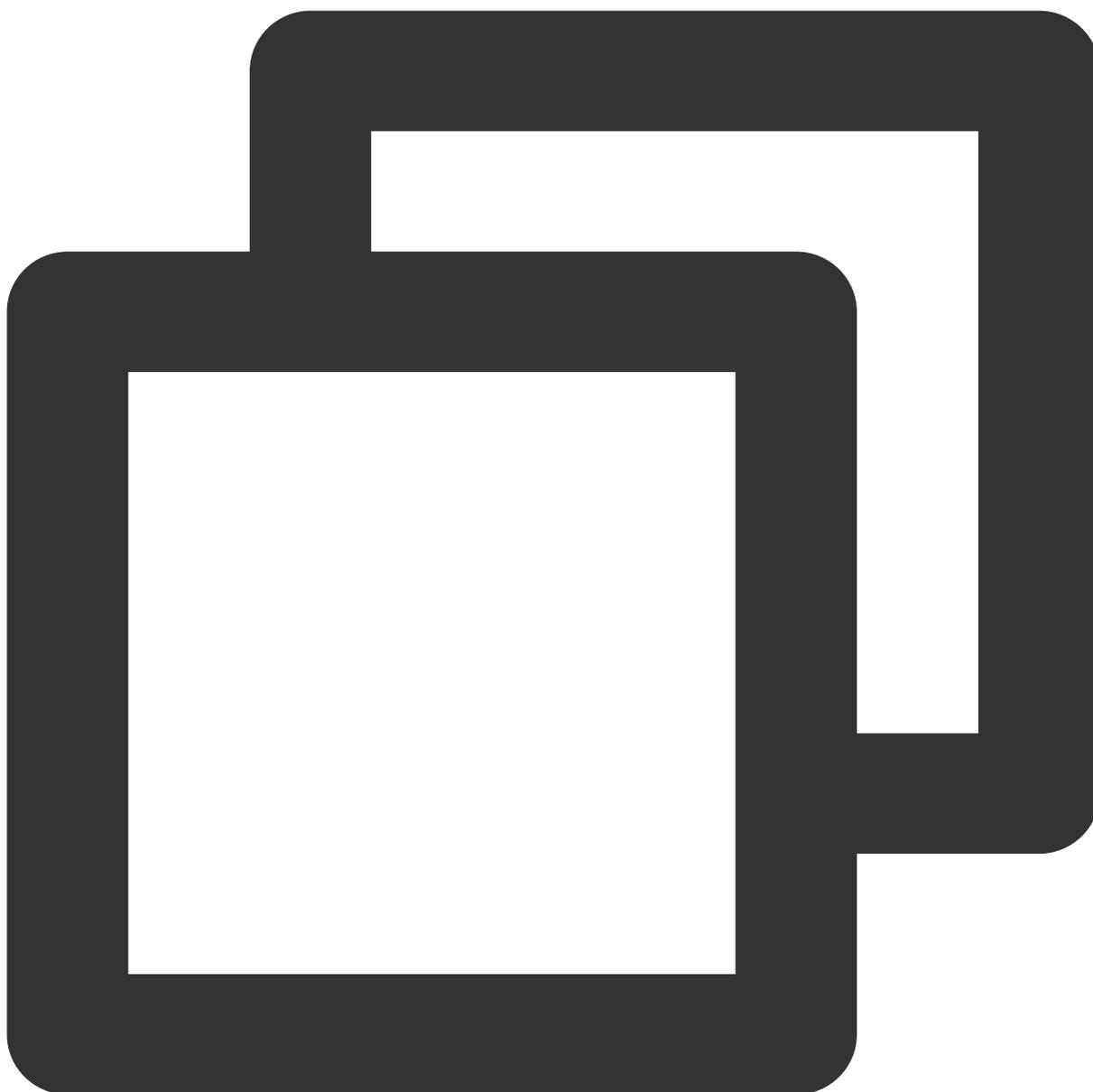
Windows 用户推荐在系统自带的 cmd 工具或 PowerShell 上使用 COSCMD 的 upload 命令，其他工具（如 git bash）对命令路径的解析策略与 PowerShell 不同，可能会导致用户的文件被上传到错误的路径上去。

操作示例 - 将 D 盘的 doc 文件夹及其文件上传到 COS 根路径



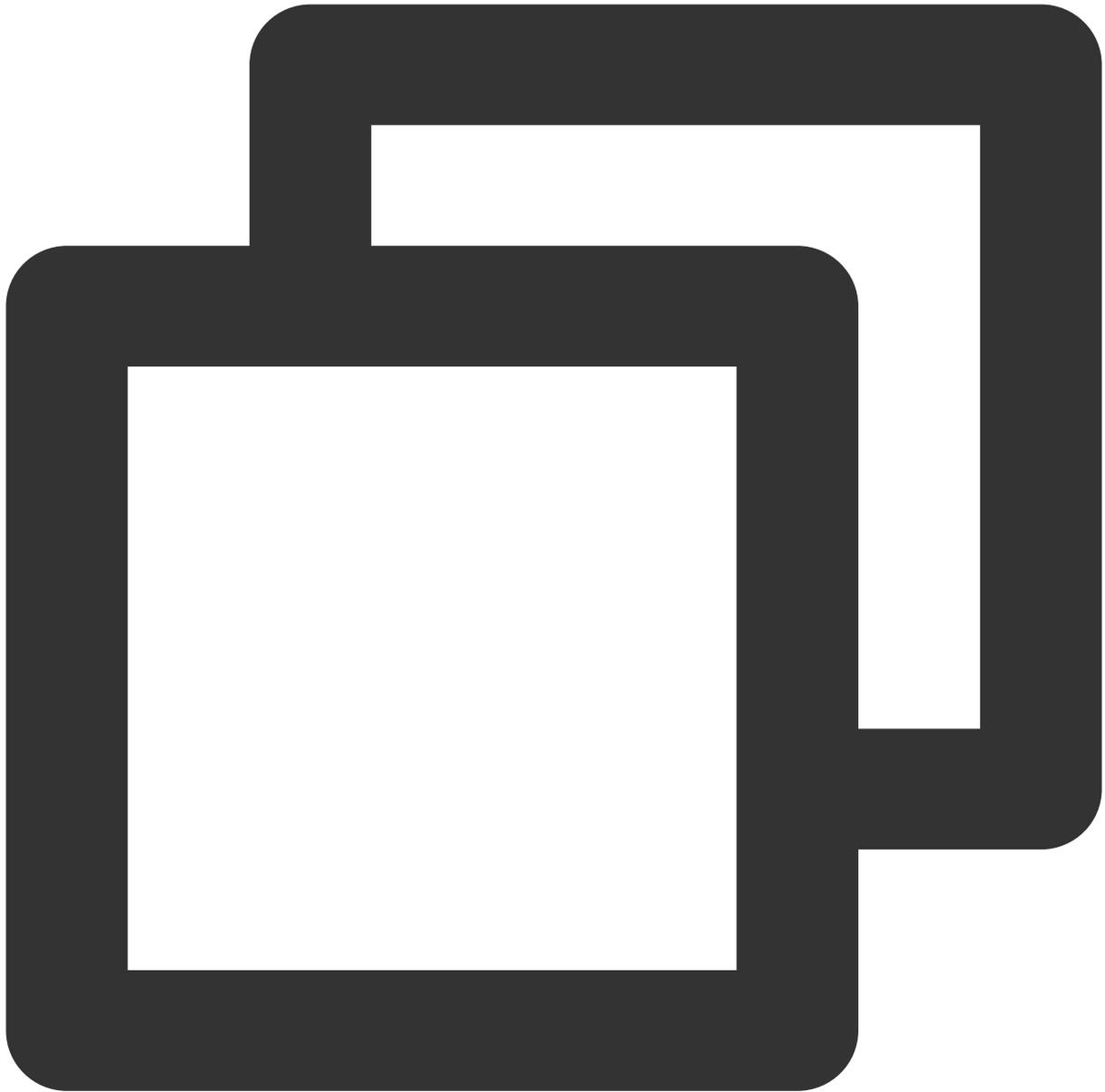
```
coscmd upload -r D:/doc /
```

操作示例 - 将 D 盘的 doc 文件夹及其文件上传到 COS 的 doc 路径



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

操作示例 - 同步上传，跳过 md5、文件大小相同的同名文件

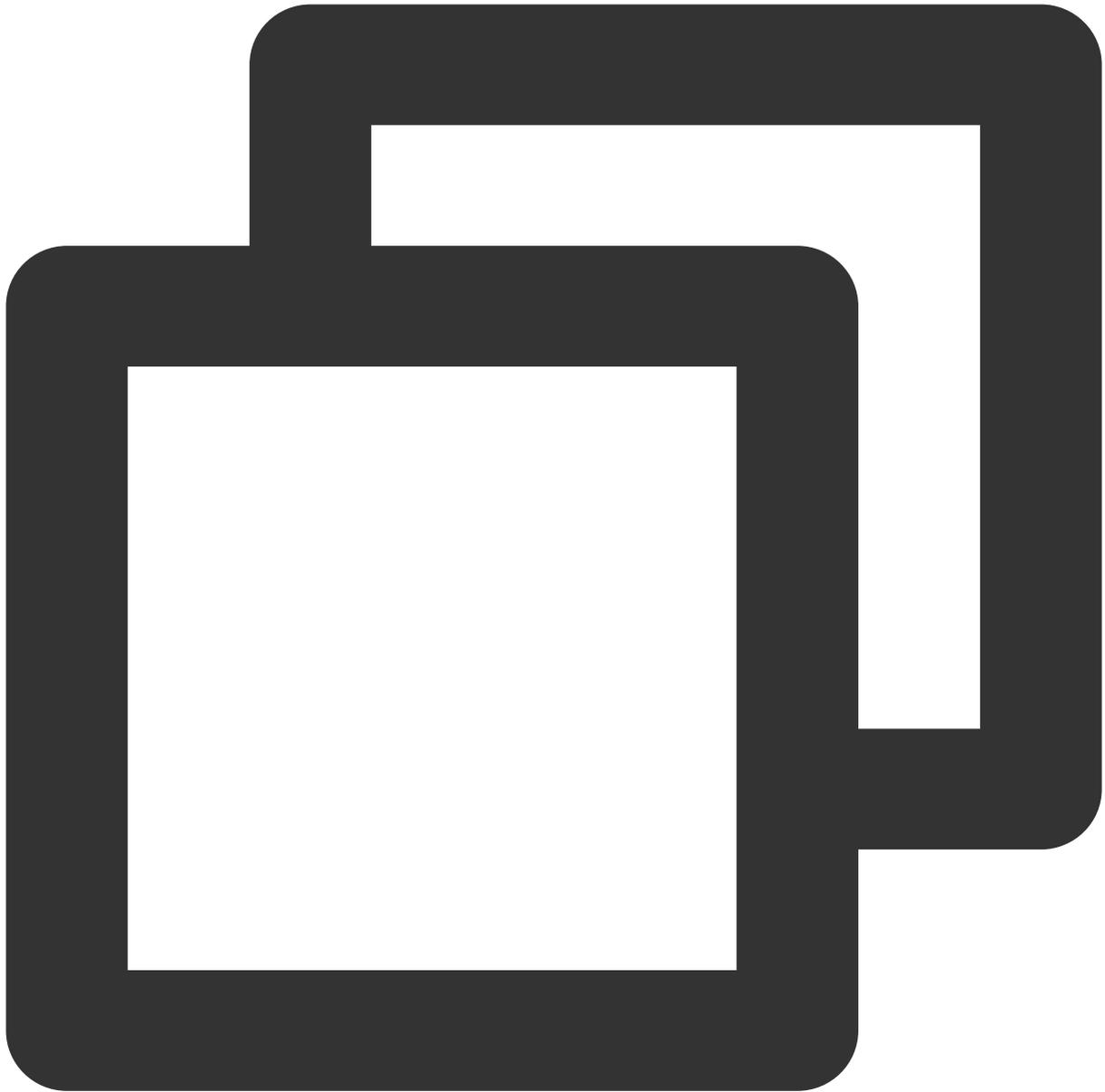


```
coscmd upload -rs D:/doc doc
```

注意：

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

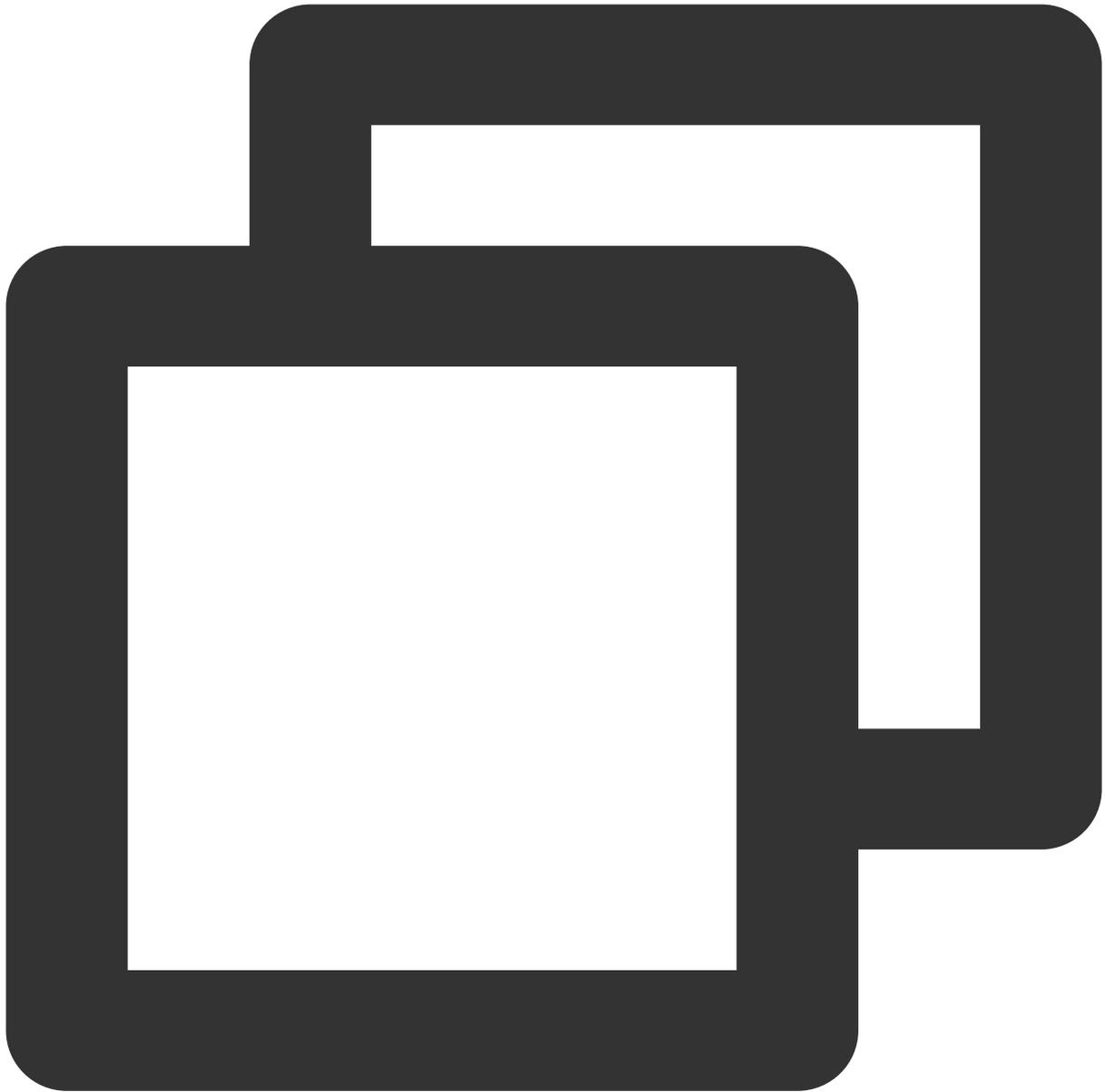
操作示例 - 同步上传，跳过文件大小相同的同名文件



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

注意：

使用 `-s` 参数可以使用同步上传，且带上 `--skipmd5` 参数后，将只对比同名文件的大小，如果大小相同则跳过上传。
操作示例 - 同步上传，并删除“D 盘 doc 文件夹中已经删除的文件”

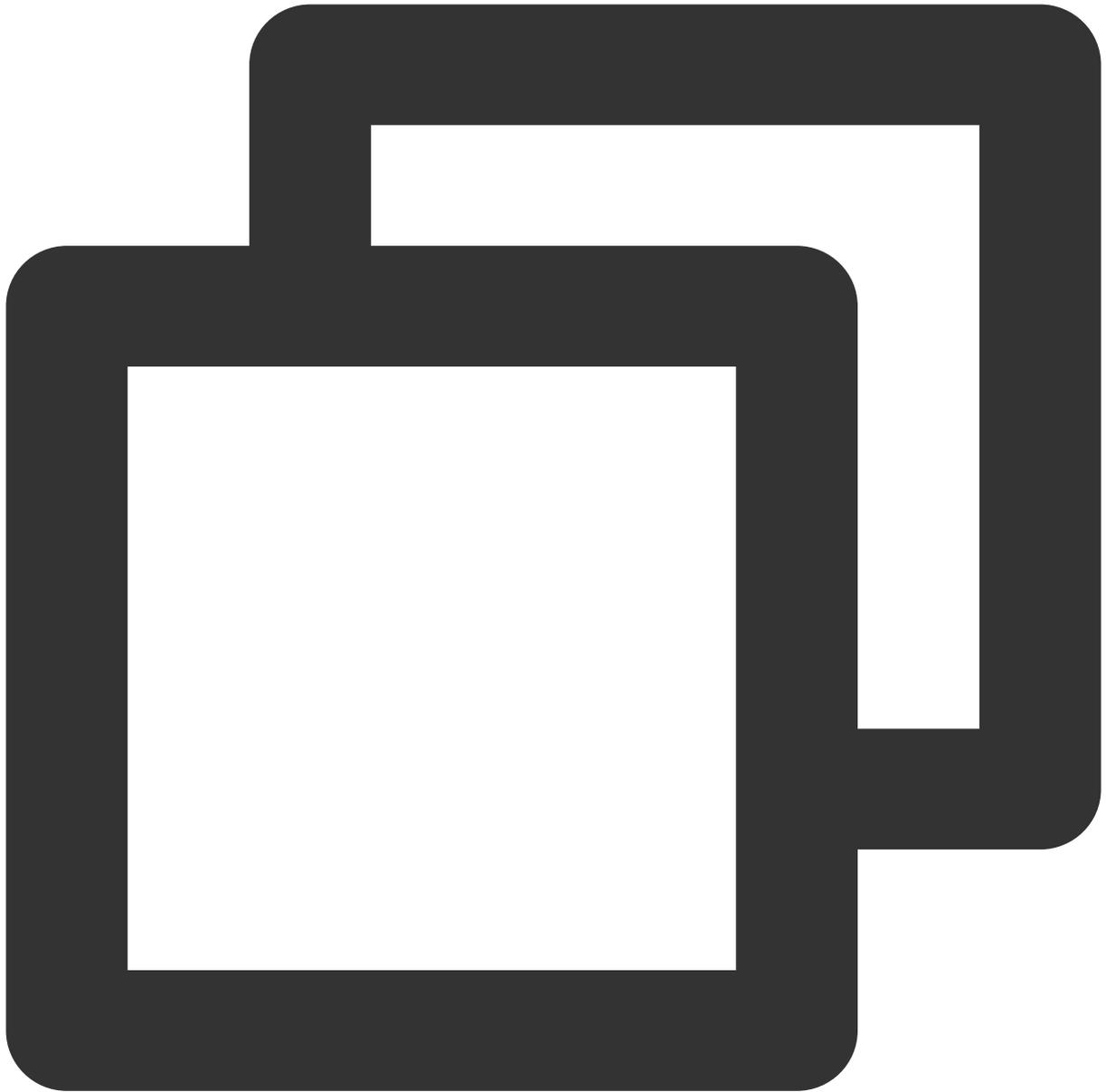


```
coscmd upload -rs --delete D:/doc /
```

说明：

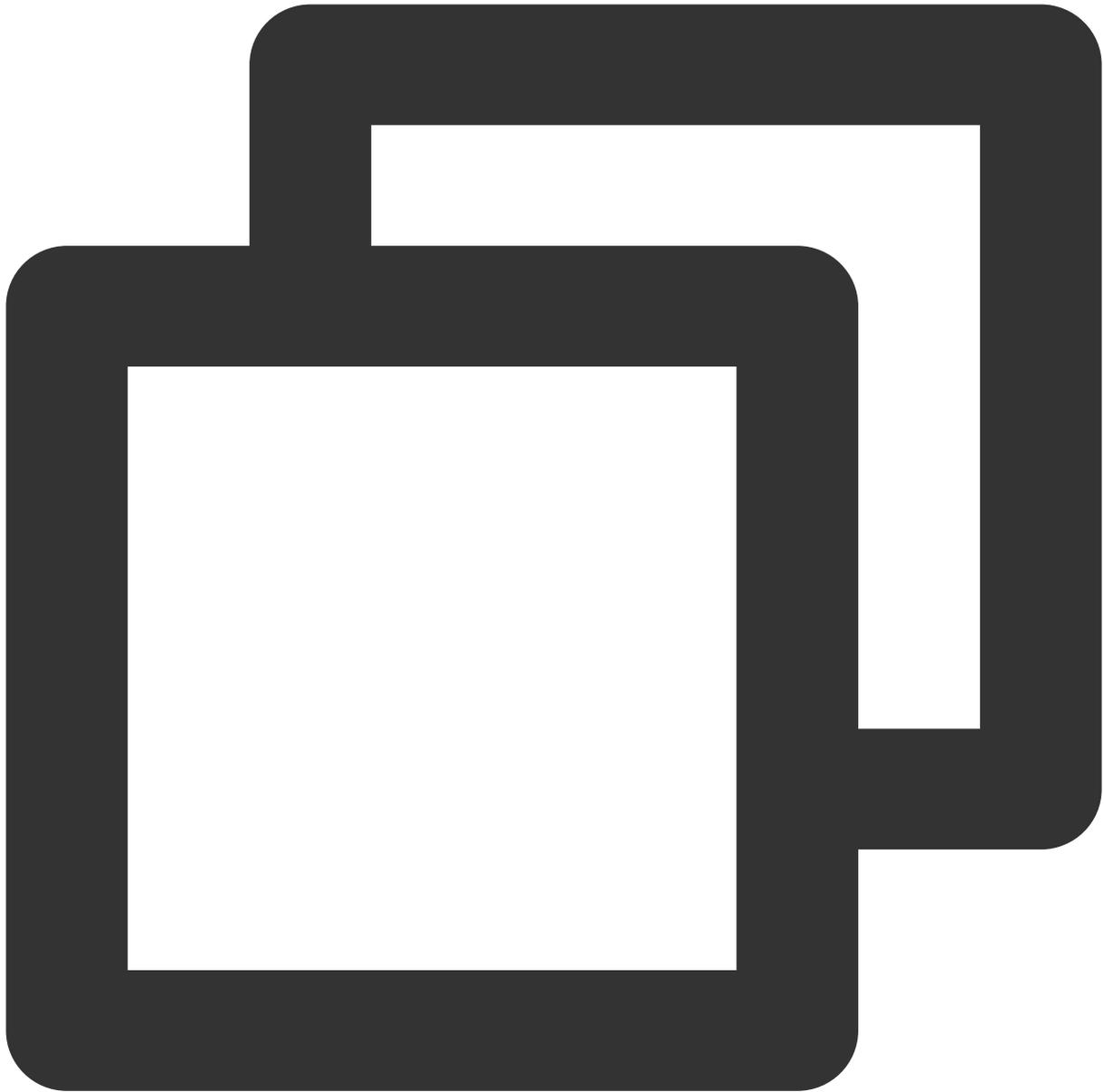
例如当天 D 盘 doc 文件夹中的文件与 COS 中 doc 路径下的文件已同步上传，源端与目标端一致。第二天用户把 D 盘 doc 文件夹中的 A 文件删除了，此时执行以上命令，会删除 COS 中 doc 路径下的 A 文件，最终实现 D 盘 doc 文件夹中的文件与 COS 中 doc 路径下的文件保持一致。

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



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

操作示例 - D 盘 doc 文件夹中 .txt 后缀的文件选择忽略上传



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

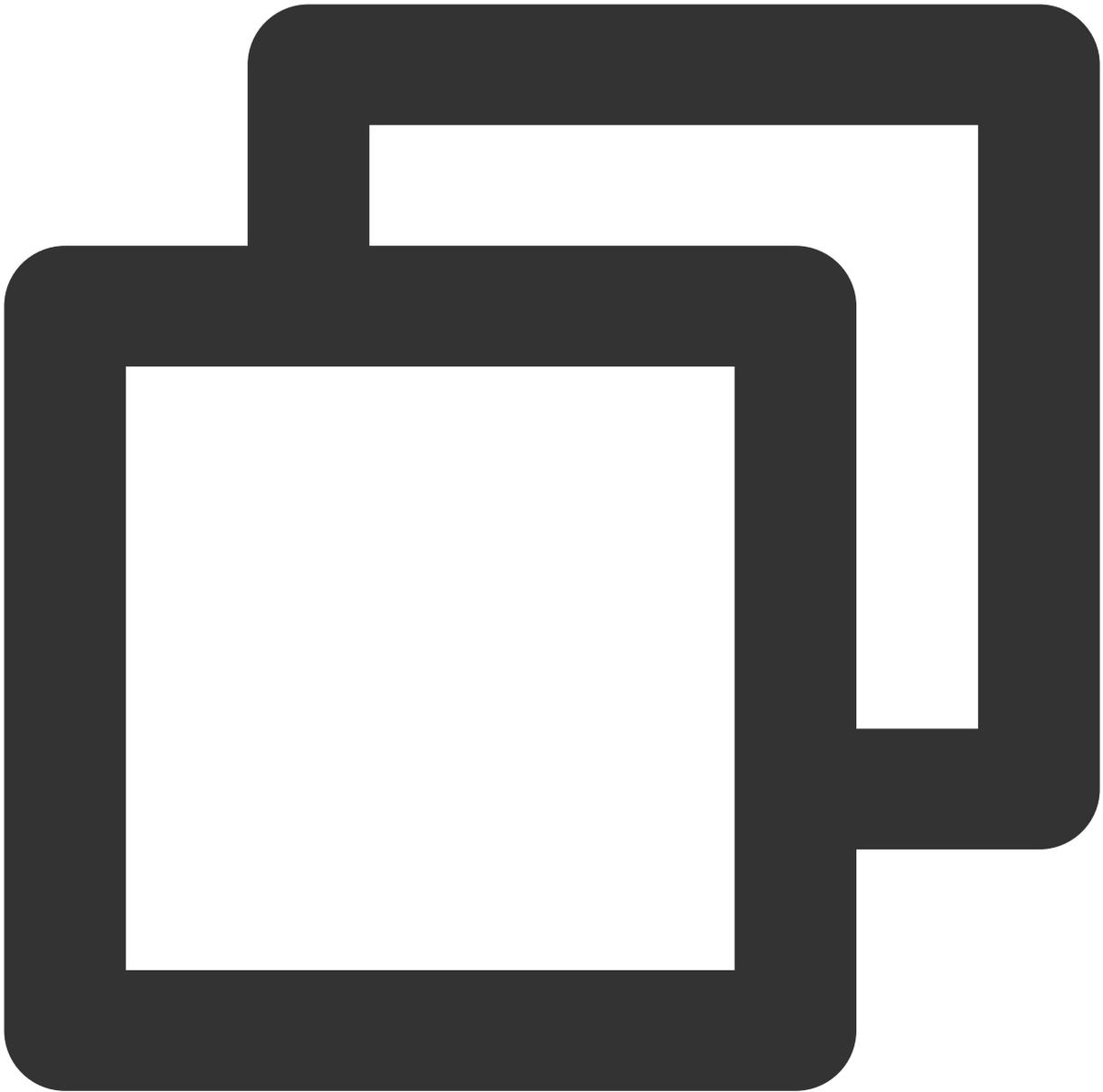
注意：

在上传文件夹时，使用 `--ignore` 参数可以忽略某一类文件，使用 `--include` 参数可以过滤某一类文件，支持 `shell` 通配规则，支持多条规则，用逗号 `,` 分隔。当忽略一类后缀时，必须最后要输入 `,` 或者加入 `"`。如果 `"` 中包含多条逗号分隔的规则，以第一条规则为准。

如果您希望使用 `--ignore` 过滤特定文件夹内的所有文件，需要使用绝对路径，并在路径前后加入 `"`。例如

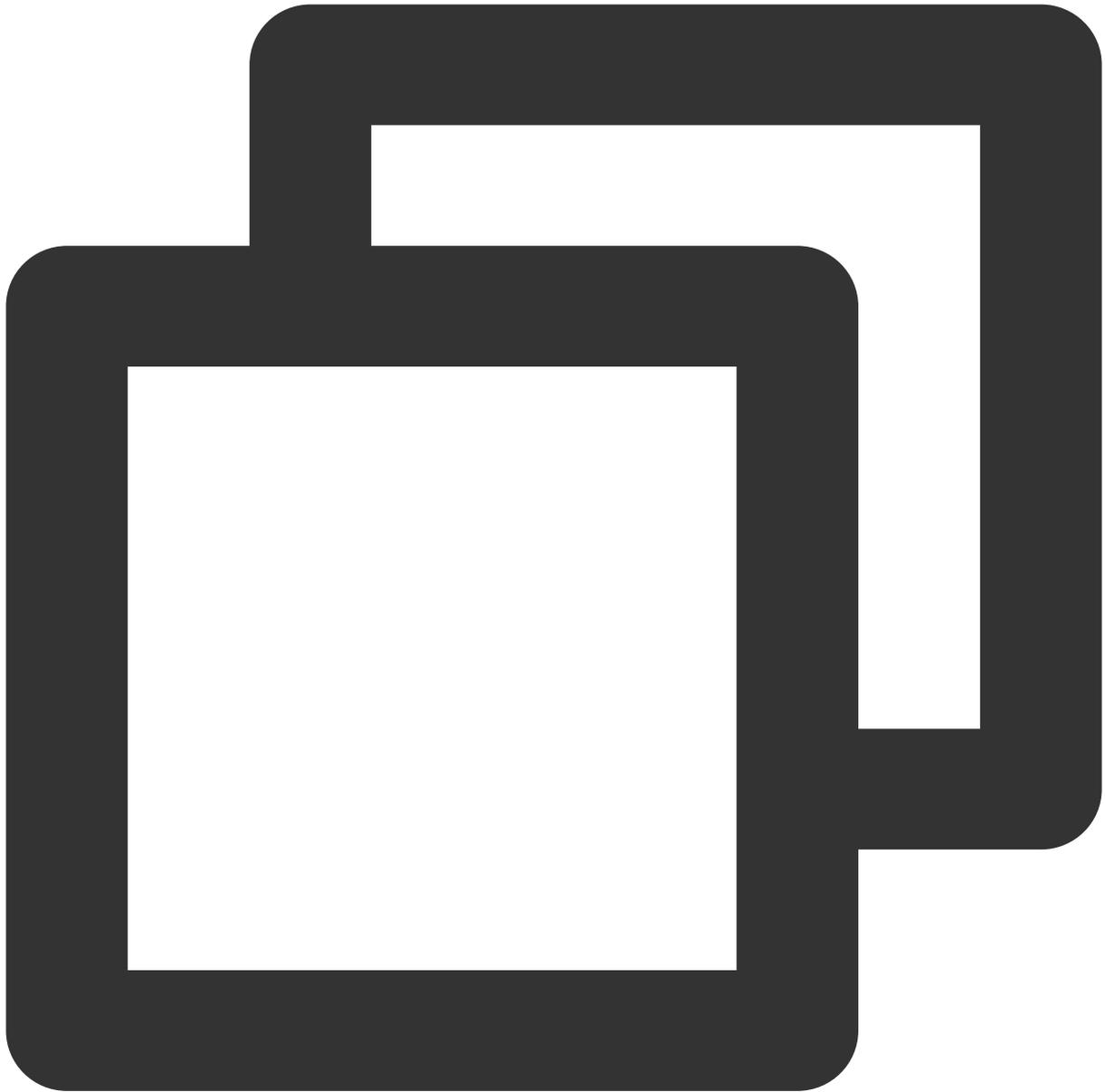
```
coscmd upload -rs D:/doc / --ignore "D:/doc/ignore_folder/*"
```

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



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

操作示例 - D 盘 doc 文件夹中 .txt 后缀的文件上传



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

注意：

当上传大于10MB的文件，COSCMD即采用分块上传方式，命令用法和简单上传一致，即 `coscmd upload <localpath> <cospath>`。

COSCMD支持大文件断点上传功能；当分块上传大文件失败时，重新上传该文件只会上传失败的分块，而不会从头开始（请保证重新上传的文件的目录以及内容和上传的目录保持一致）。

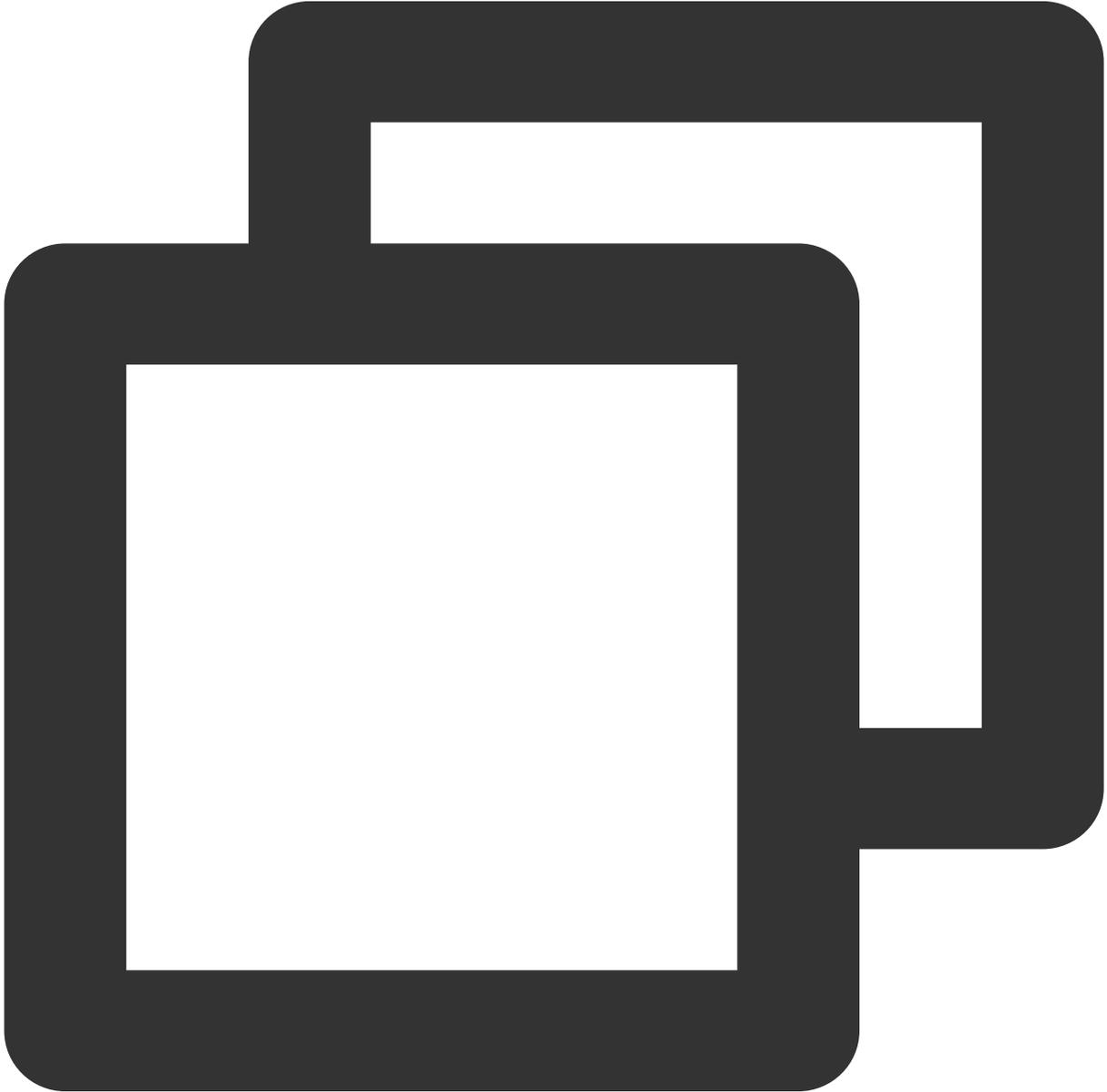
COSCMD分块上传时会对每一块进行MD5校验。

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

查询文件列表

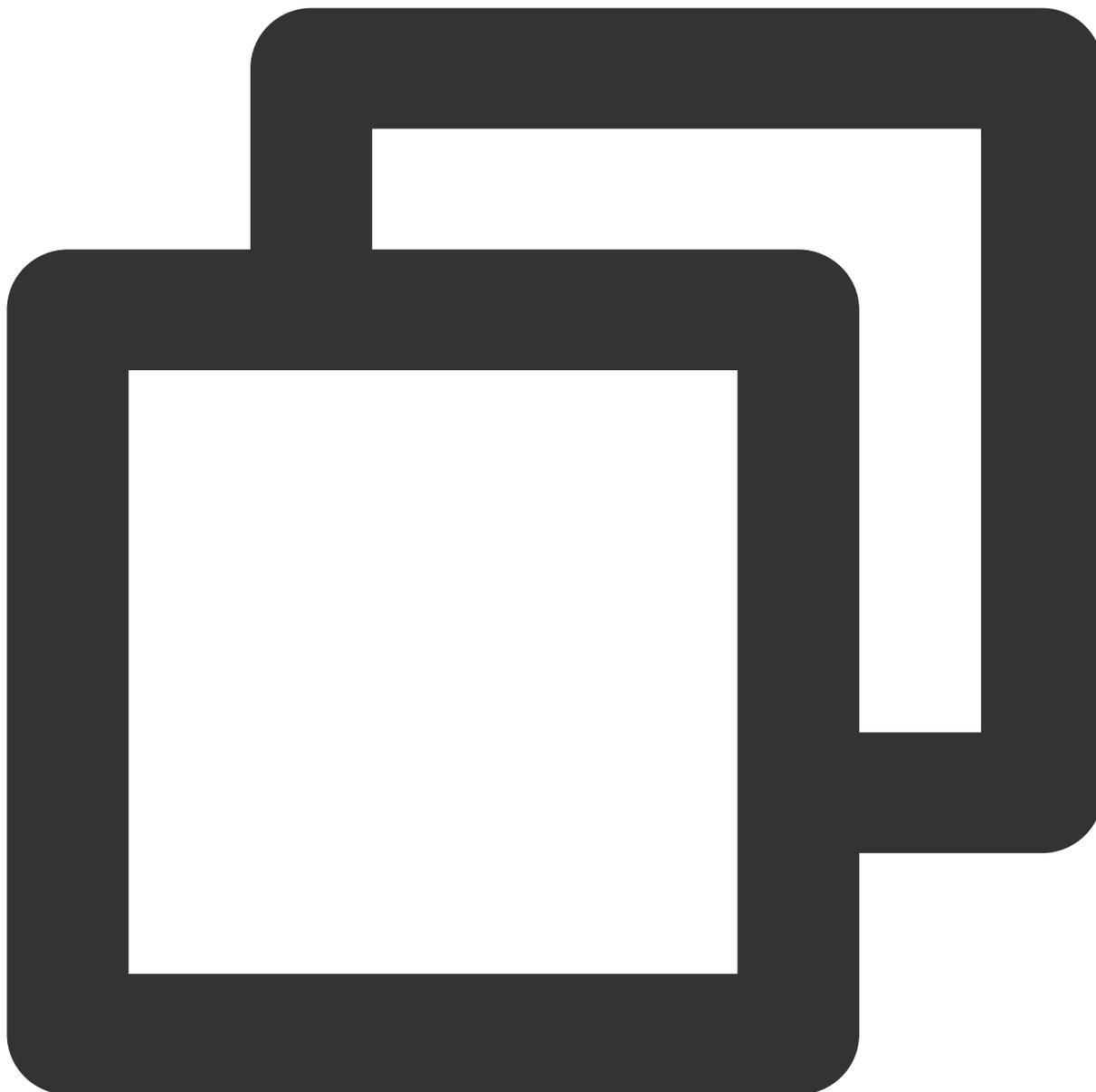
查询命令如下：

命令格式



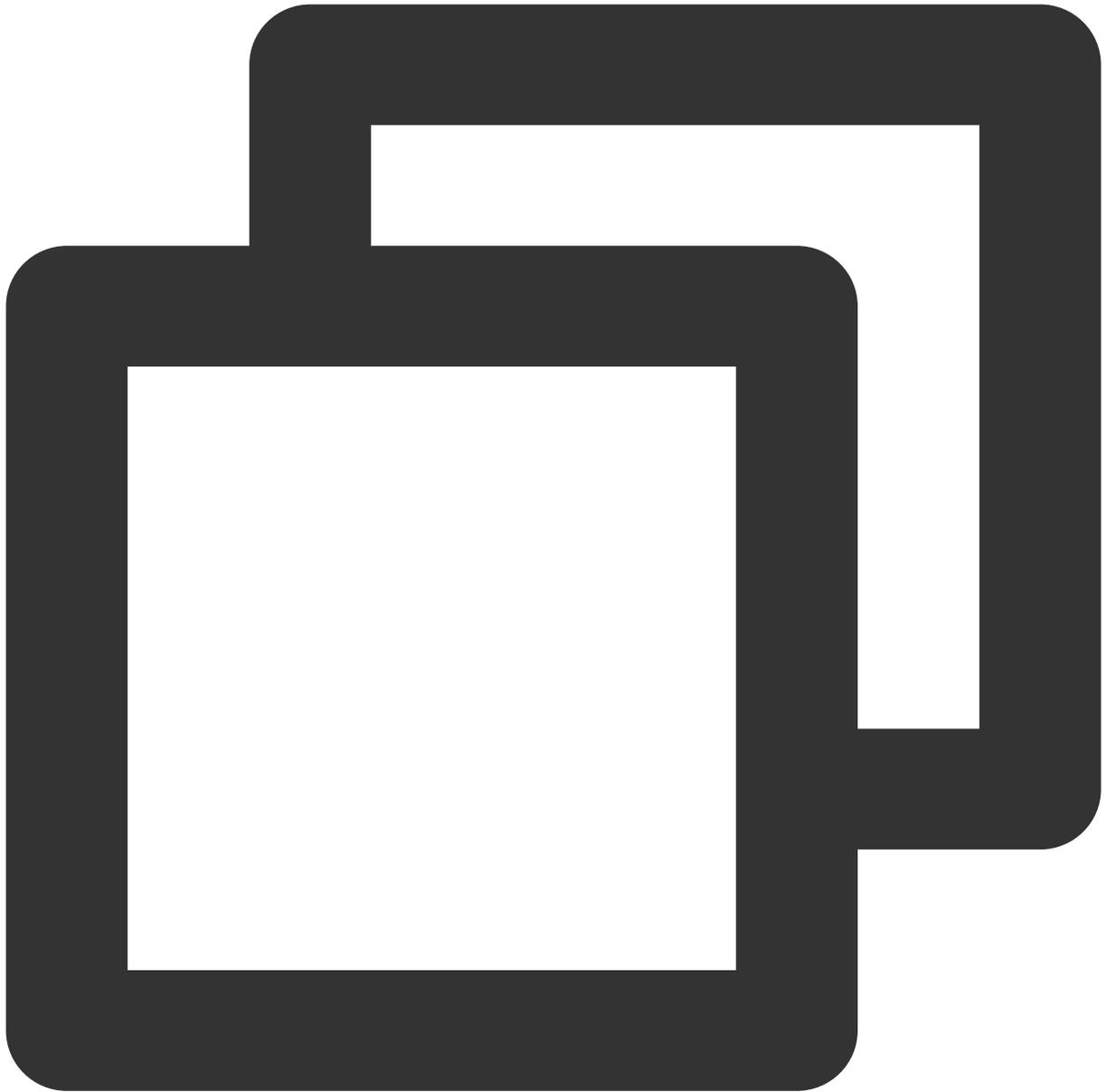
```
coscmd list <cospath>
```

操作示例 - 递归查询该存储桶下 doc/ 前缀下所有的文件列表



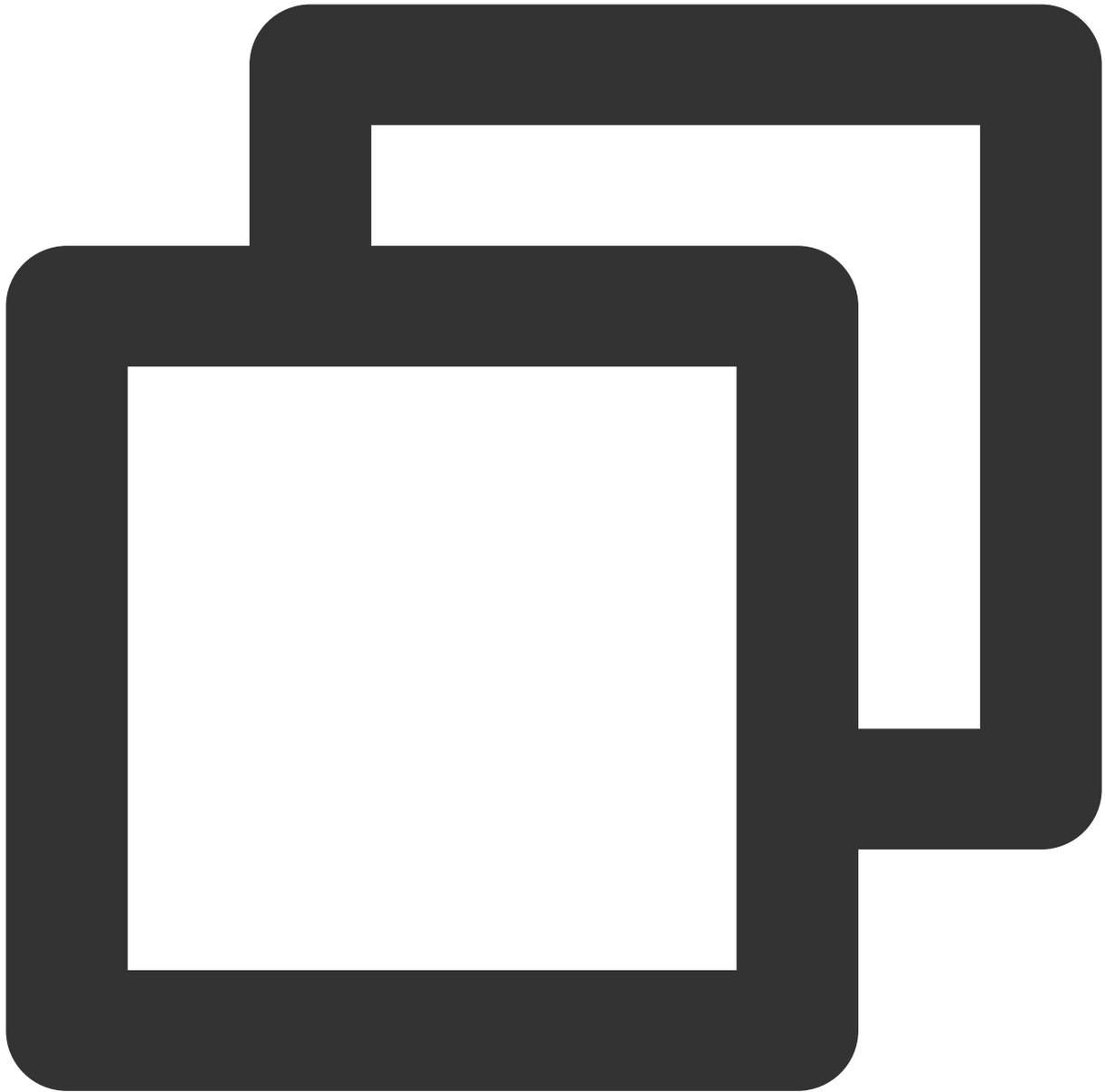
```
coscmd list doc/
```

操作示例 - 递归查询该存储桶下所有的文件列表、文件数量和文件大小



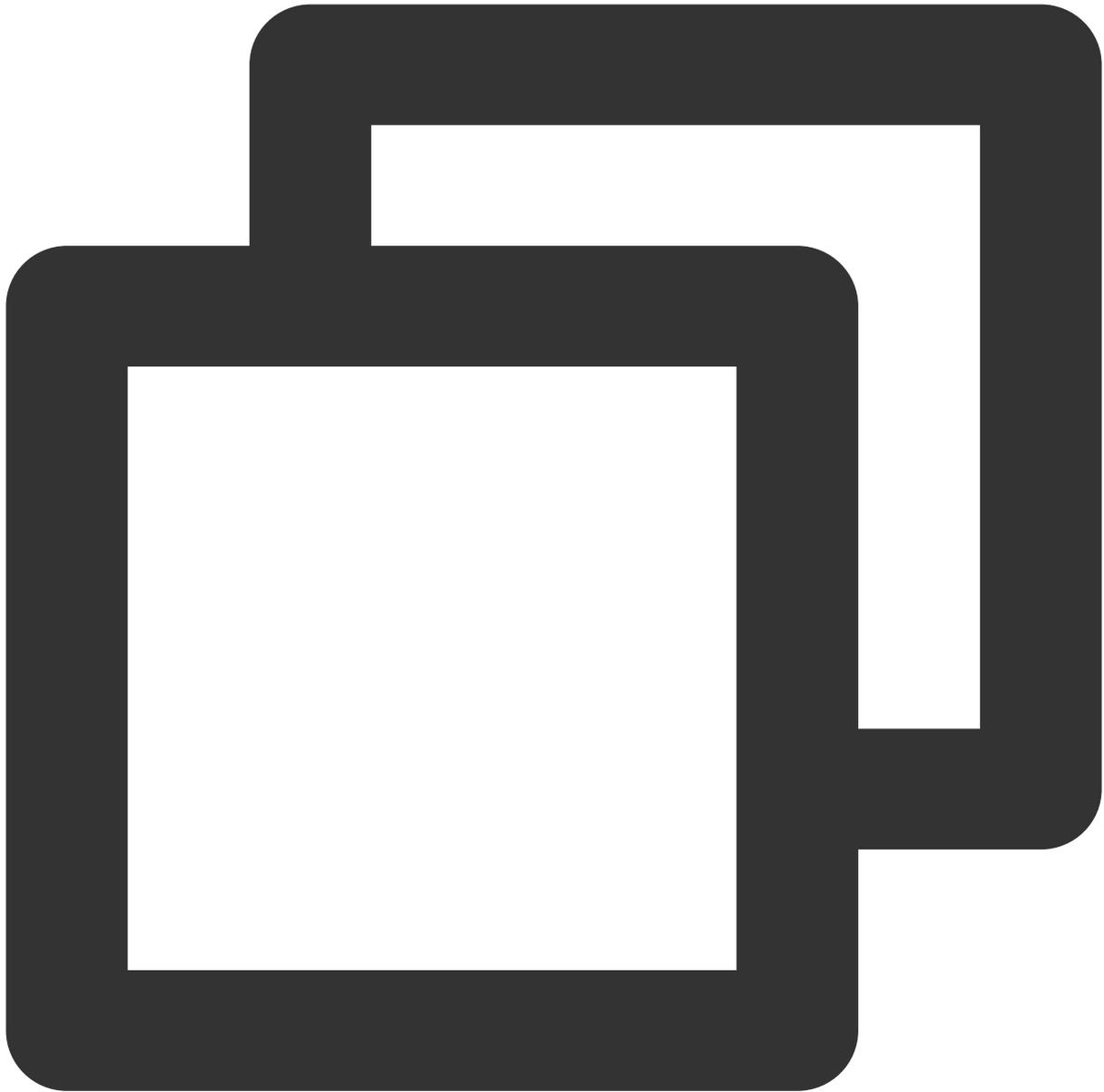
```
coscmd list -ar
```

操作示例 - 递归查询 `examplefolder` 前缀的所有文件列表



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

操作示例 - 查询该存储桶下所有文件的历史版本



```
coscmd list -v
```

说明：

请将 "<>" 中的参数替换为您需要查询文件列表的 COS 上文件的路径（`cospath`）。`<cospath>` 为空默认查询当前存储桶根目录。

使用 `-a` 查询全部文件。

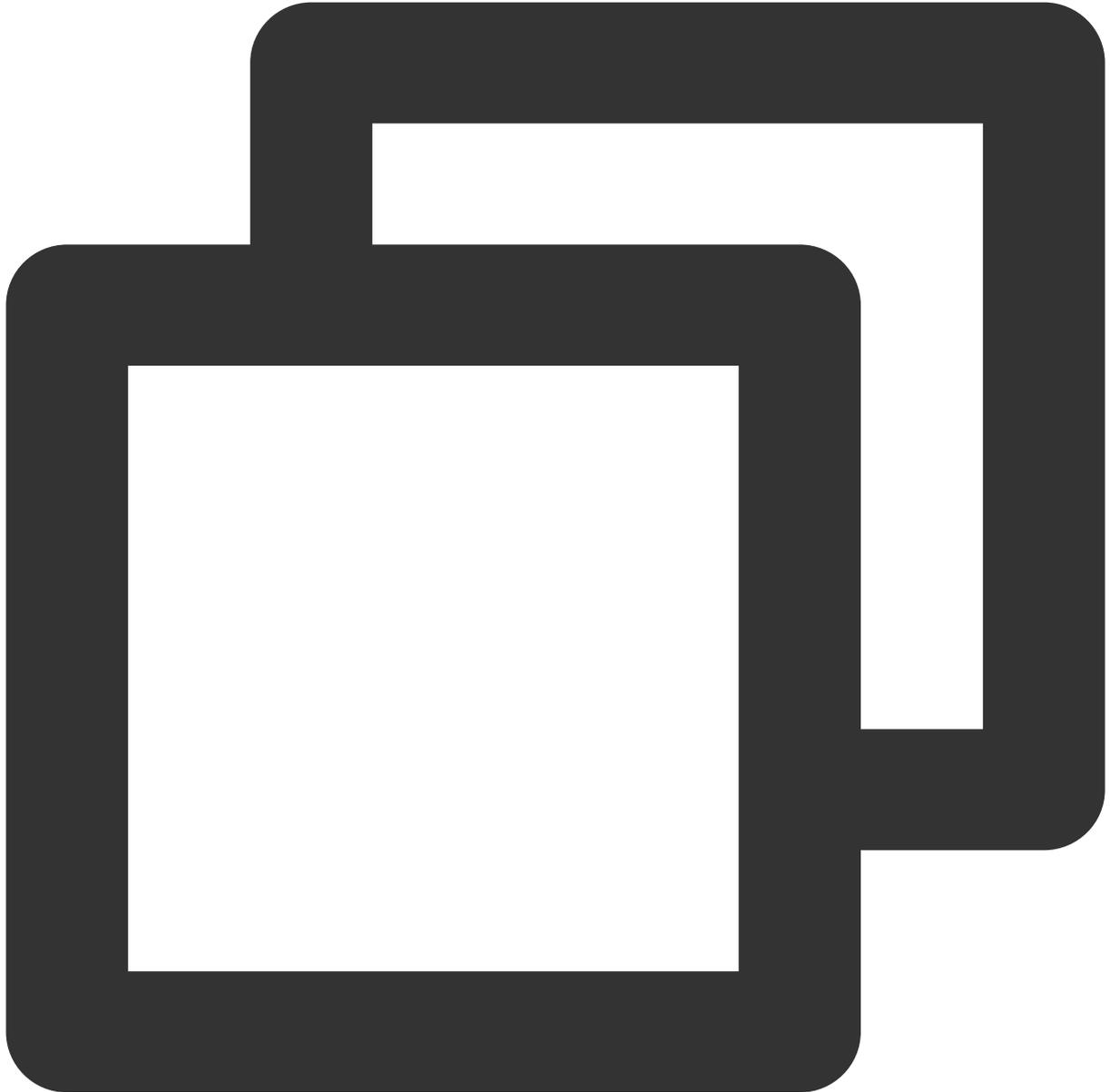
使用 `-r` 递归查询，并且会在末尾返回列出文件的数量和大小之和。

使用 `-n num` 设置查询数量的最大值。

查看文件信息

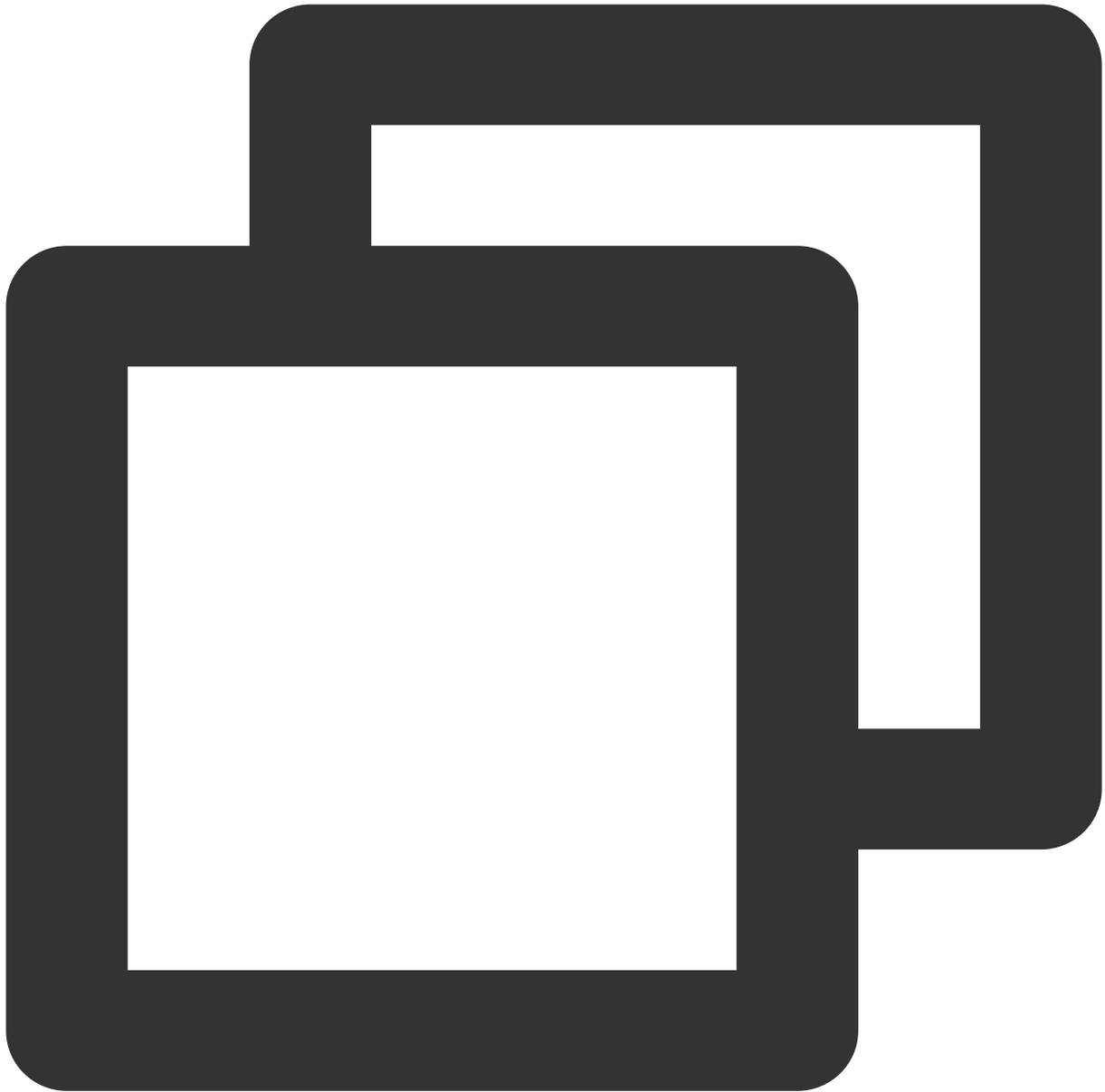
命令如下：

命令格式



```
coscmd info <cospath>
```

操作示例 - 查看 doc/picture.jpg 的元信息



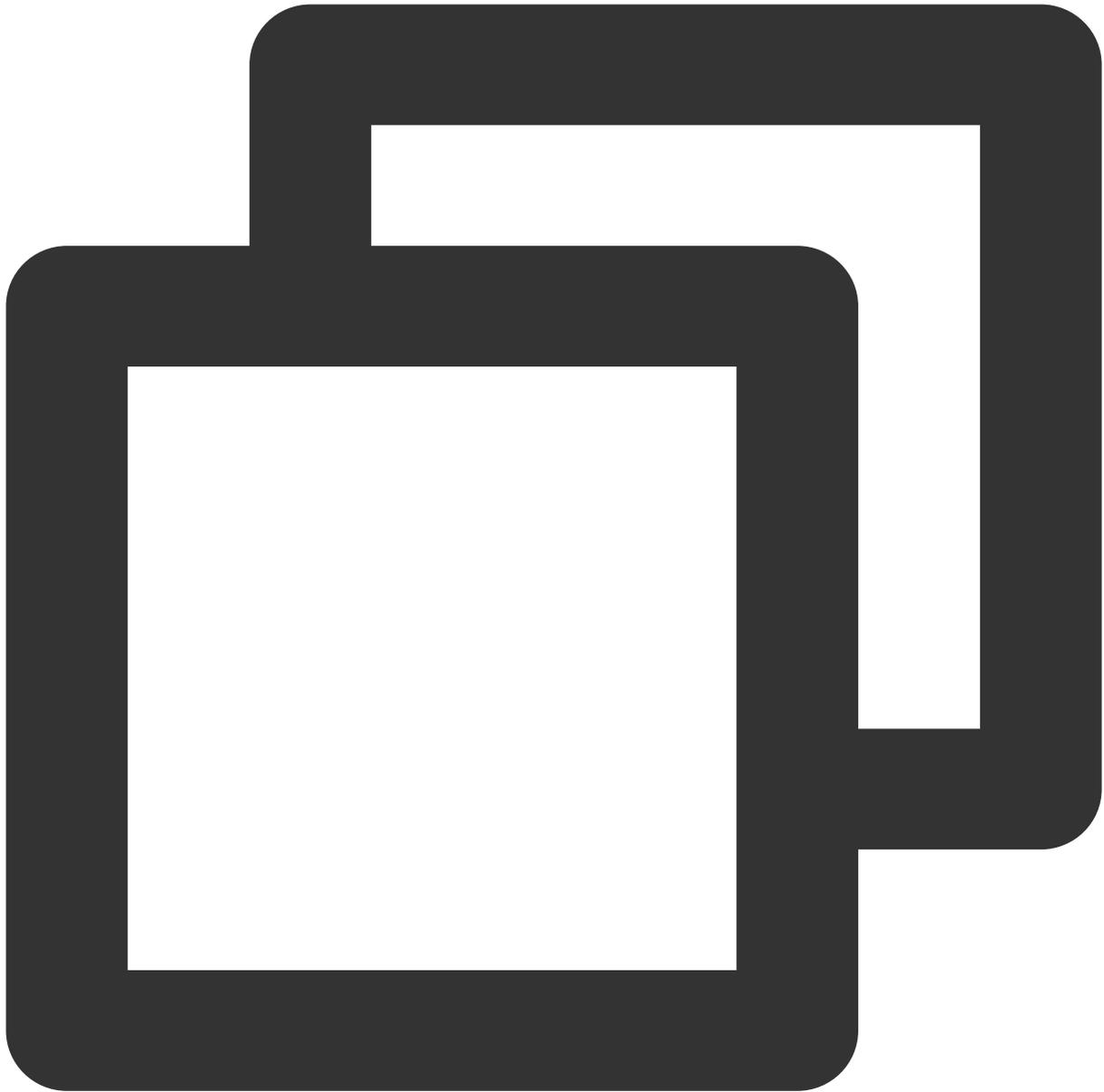
```
coscmd info doc/picture.jpg
```

说明：

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

下载文件或文件夹

下载文件命令格式

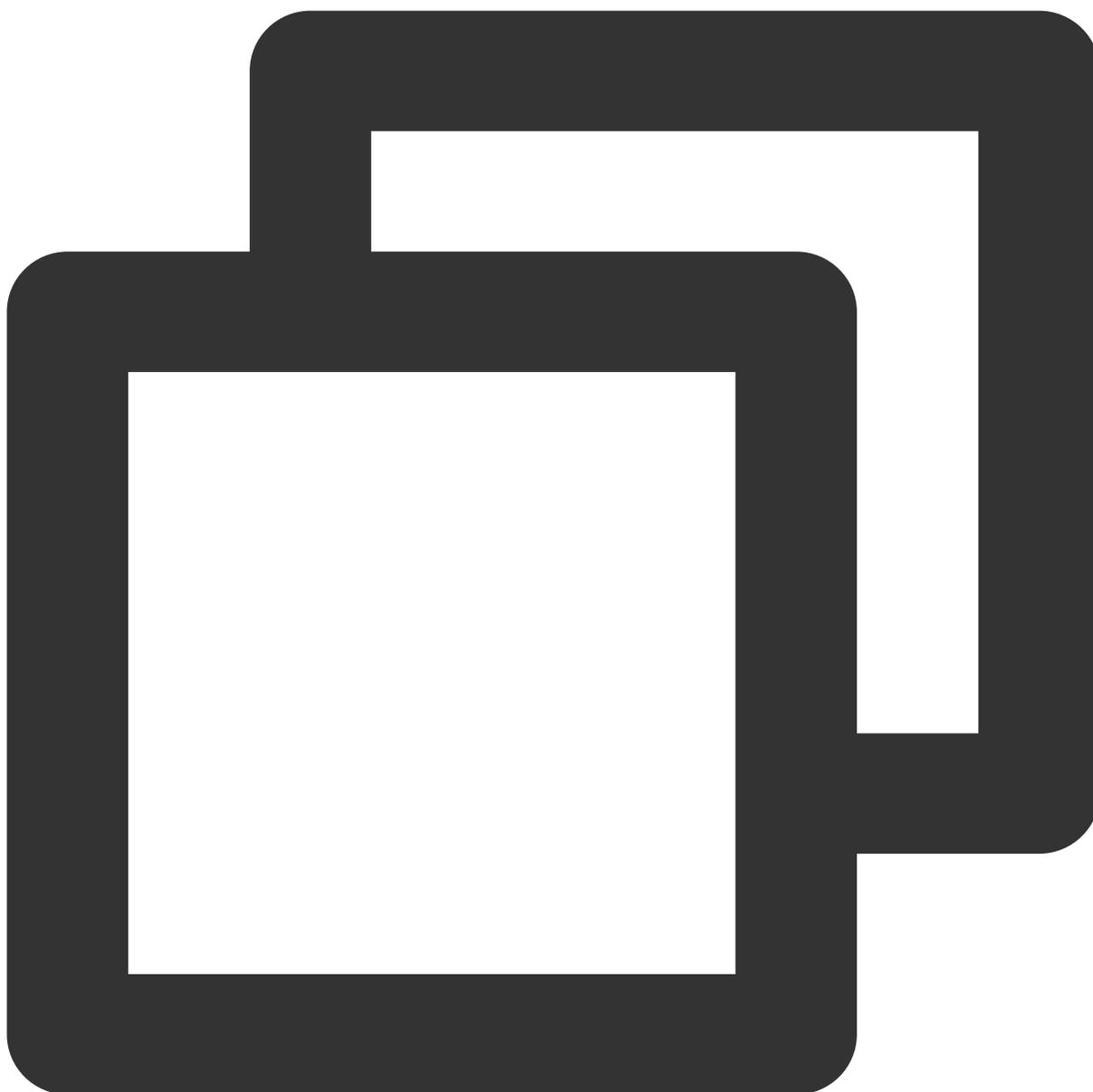


```
coscmd download <cospath> <localpath>
```

注意：

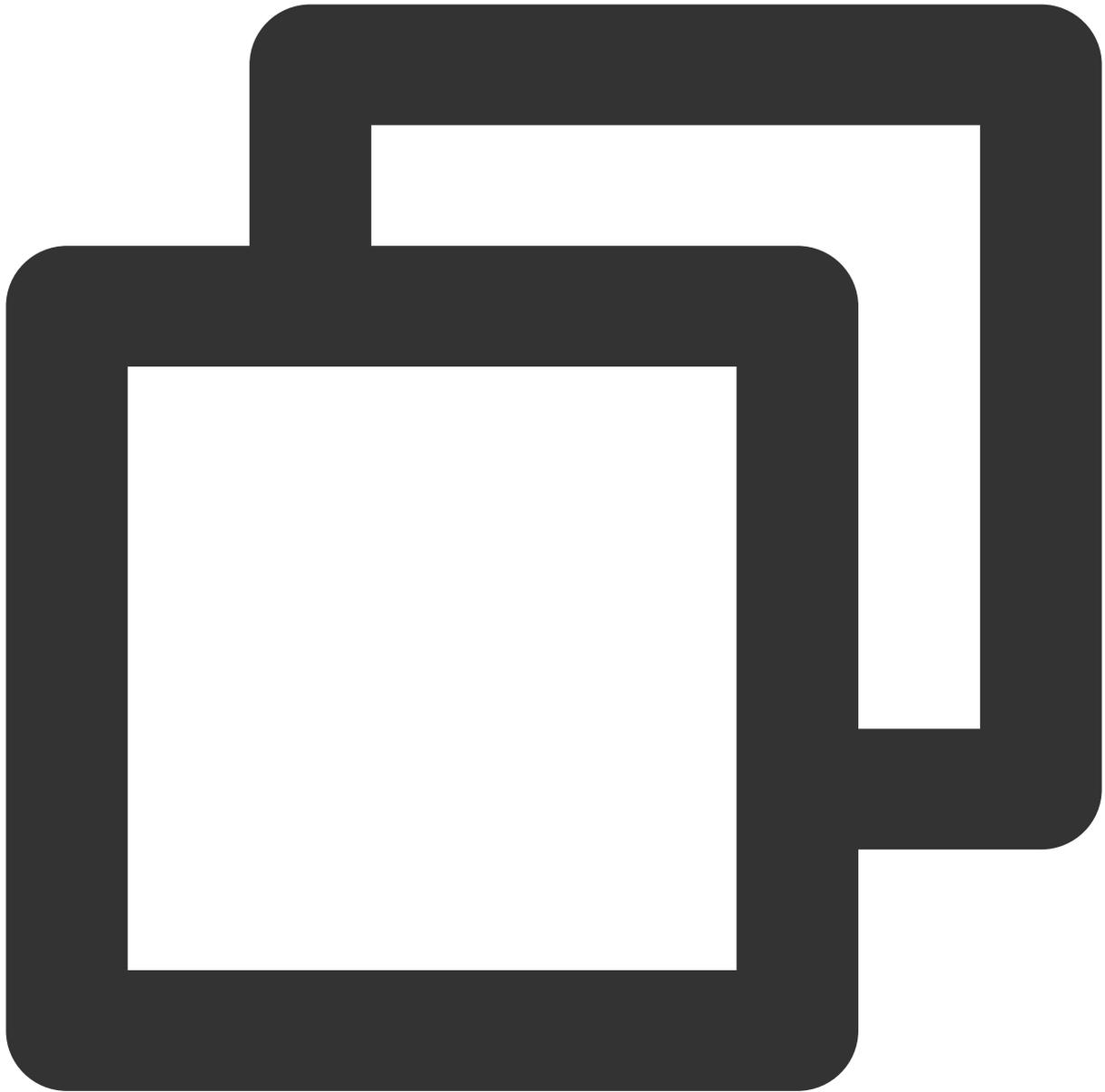
请将 "<>" 中的参数替换为您需要下载 COS 上文件的路径（cospath），以及本地存储路径（localpath）。

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



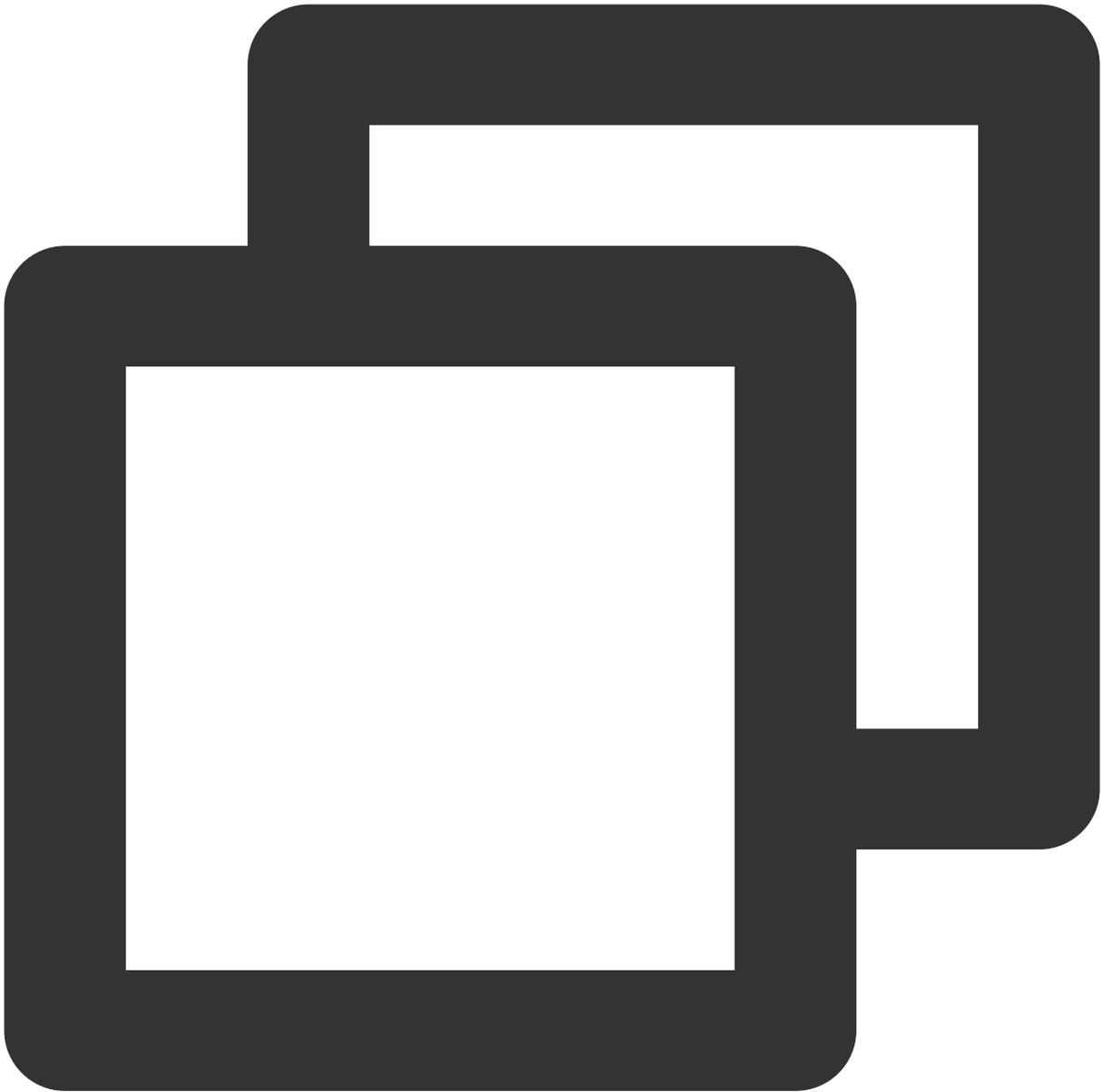
```
coscmd download doc/picture.jpg D:/picture.jpg
```

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



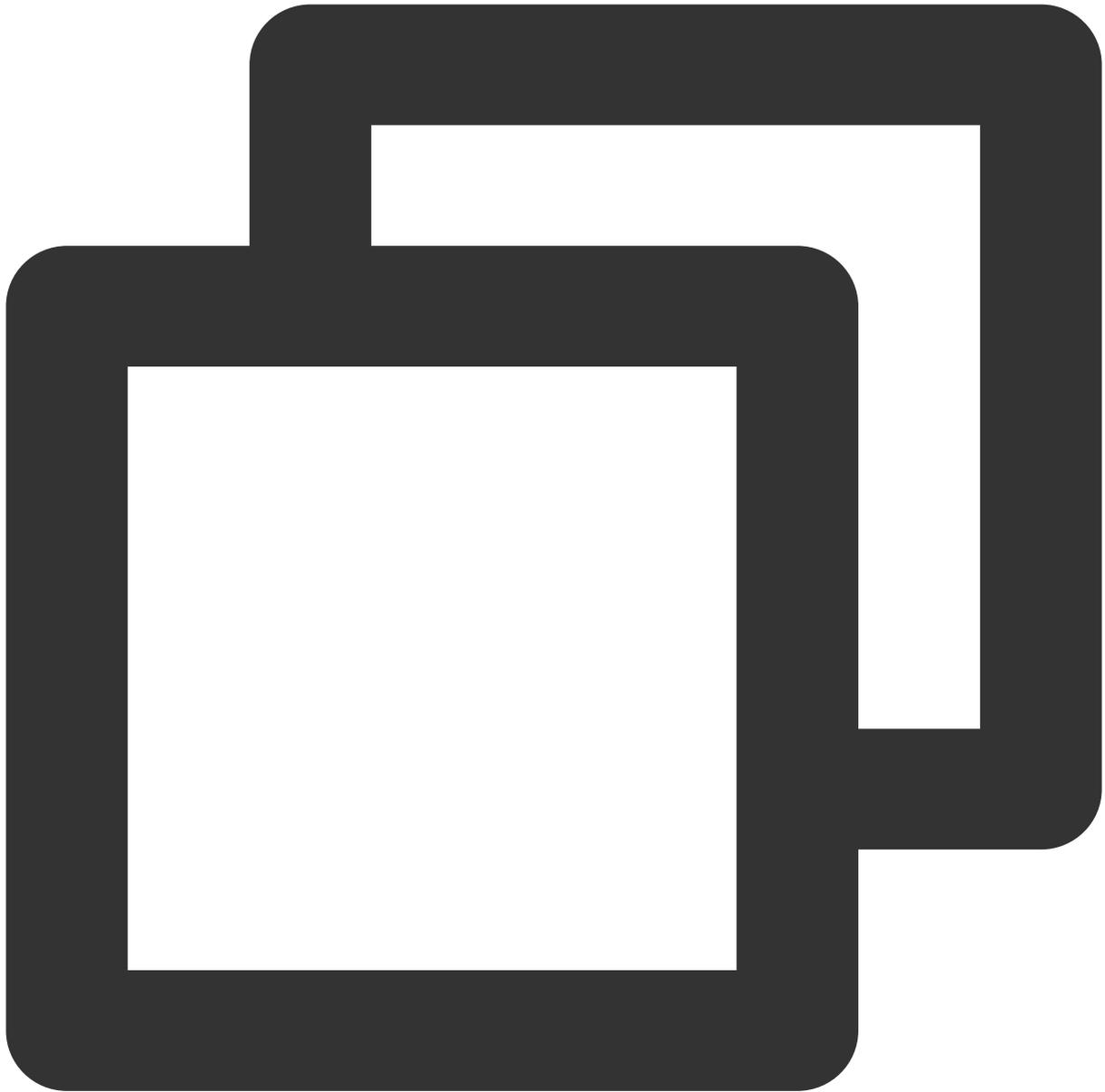
```
coscmd download doc/picture.jpg D:/
```

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



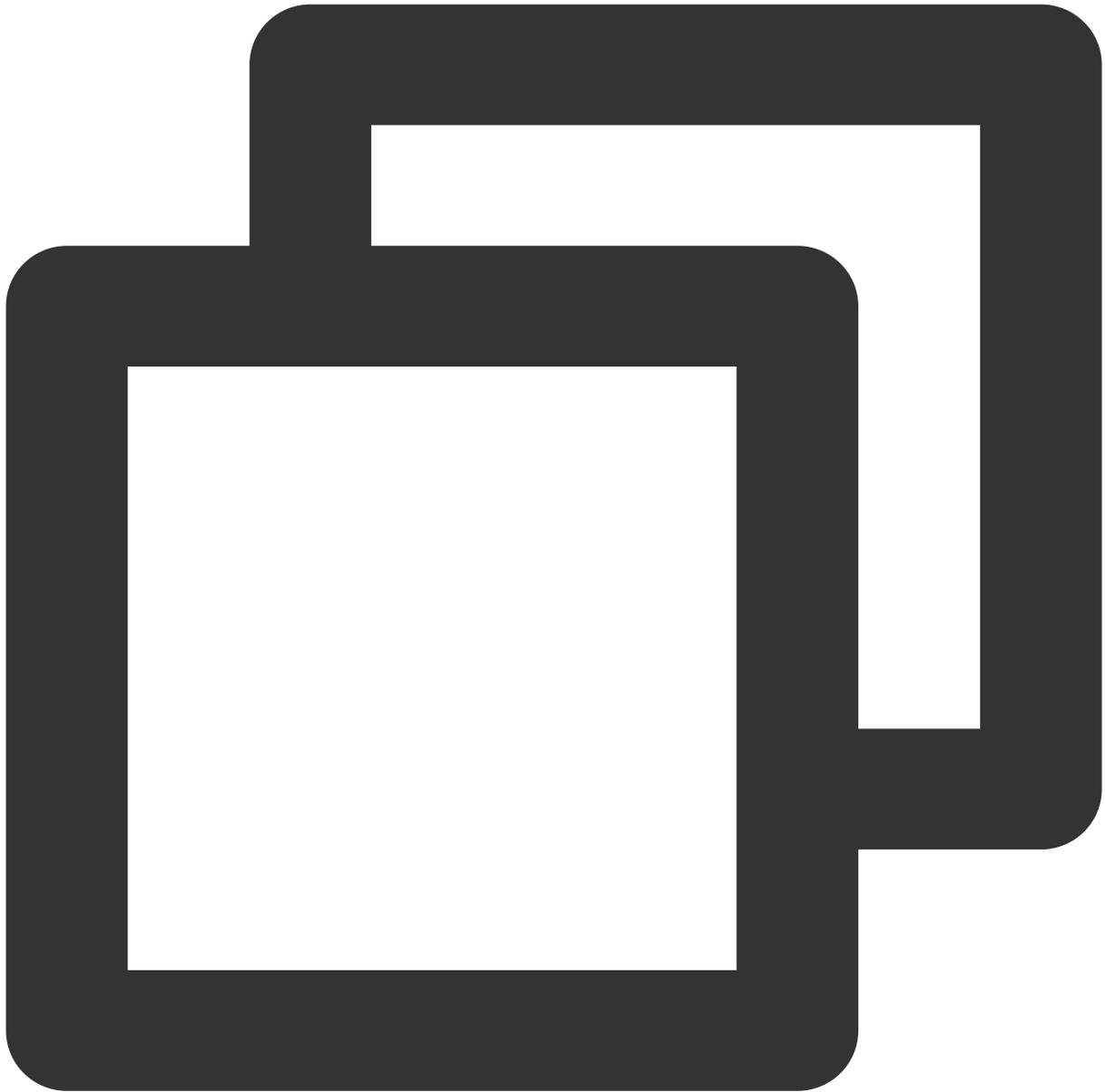
```
coscmd download picture.jpg --versionId MTg0NDUxMzc2OTM4NTExNTg7Tjg D:/
```

下载文件夹命令格式



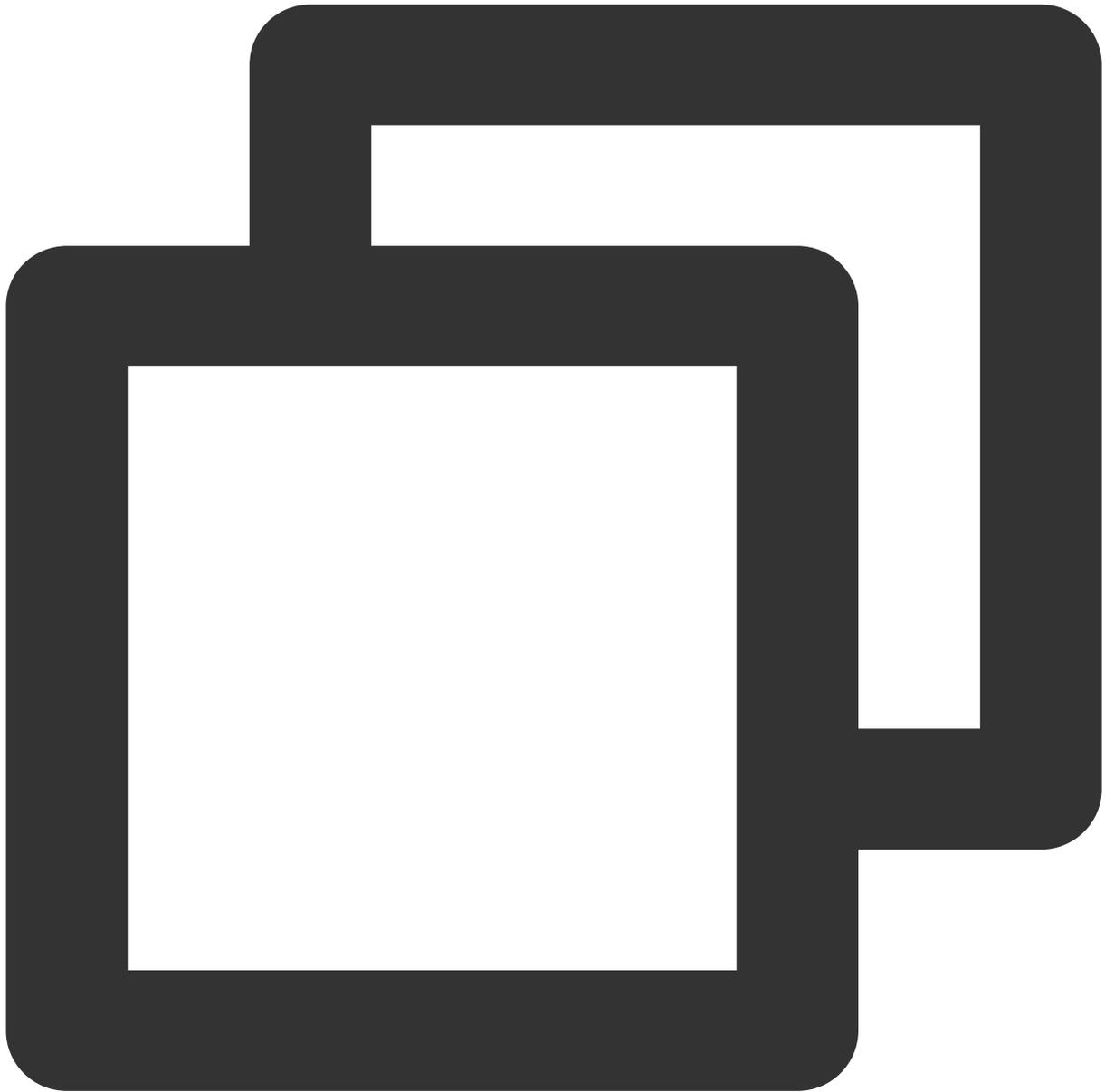
```
coscmd download -r <cospath> <localpath>
```

操作示例 - 下载 doc 目录为 D:/folder/doc



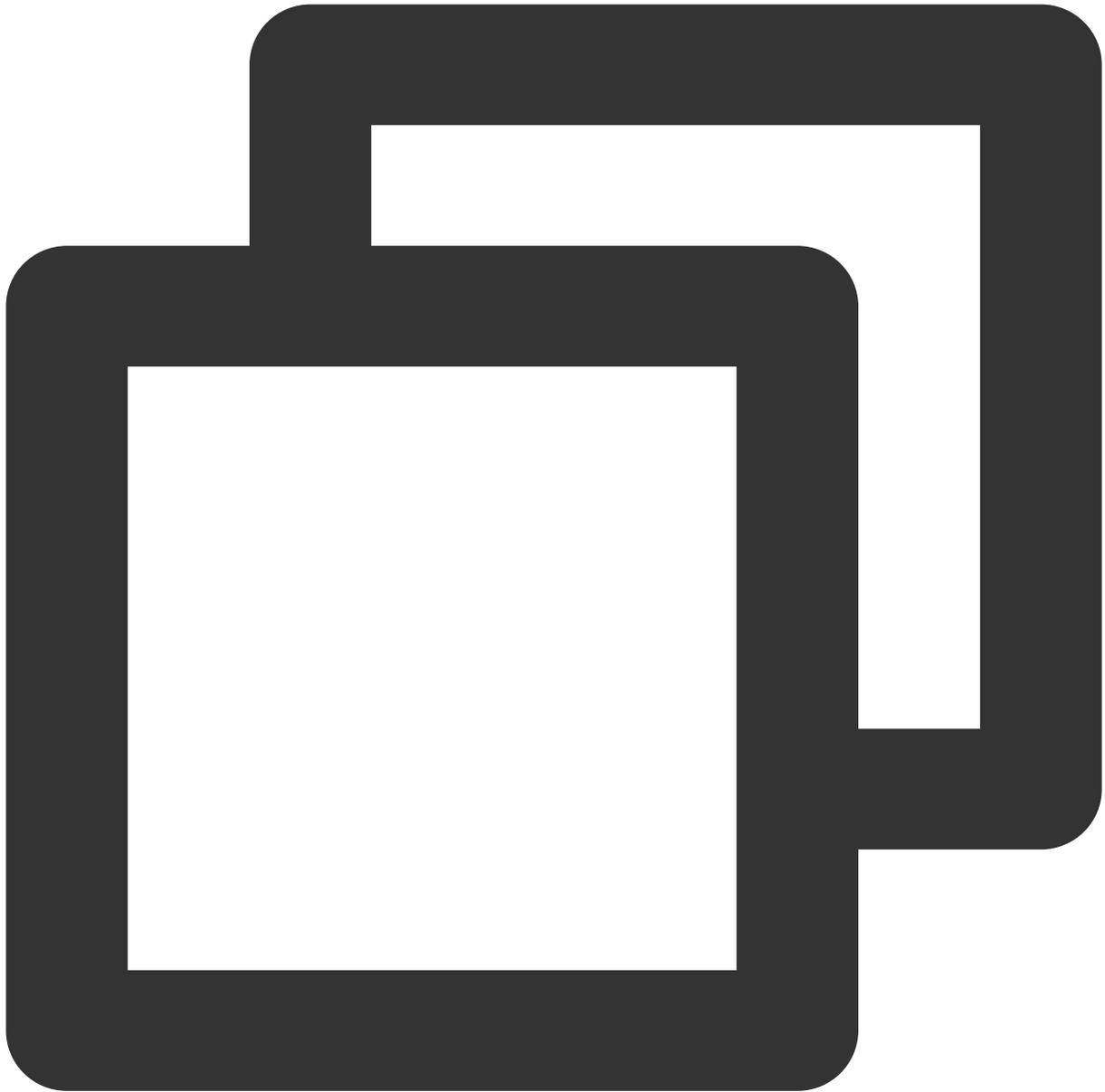
```
coscmd download -r doc D:/folder/
```

操作示例 - 下载根目录文件，但跳过根目录下的 doc 目录



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

操作示例 - 覆盖下载当前存储桶根目录下所有的文件

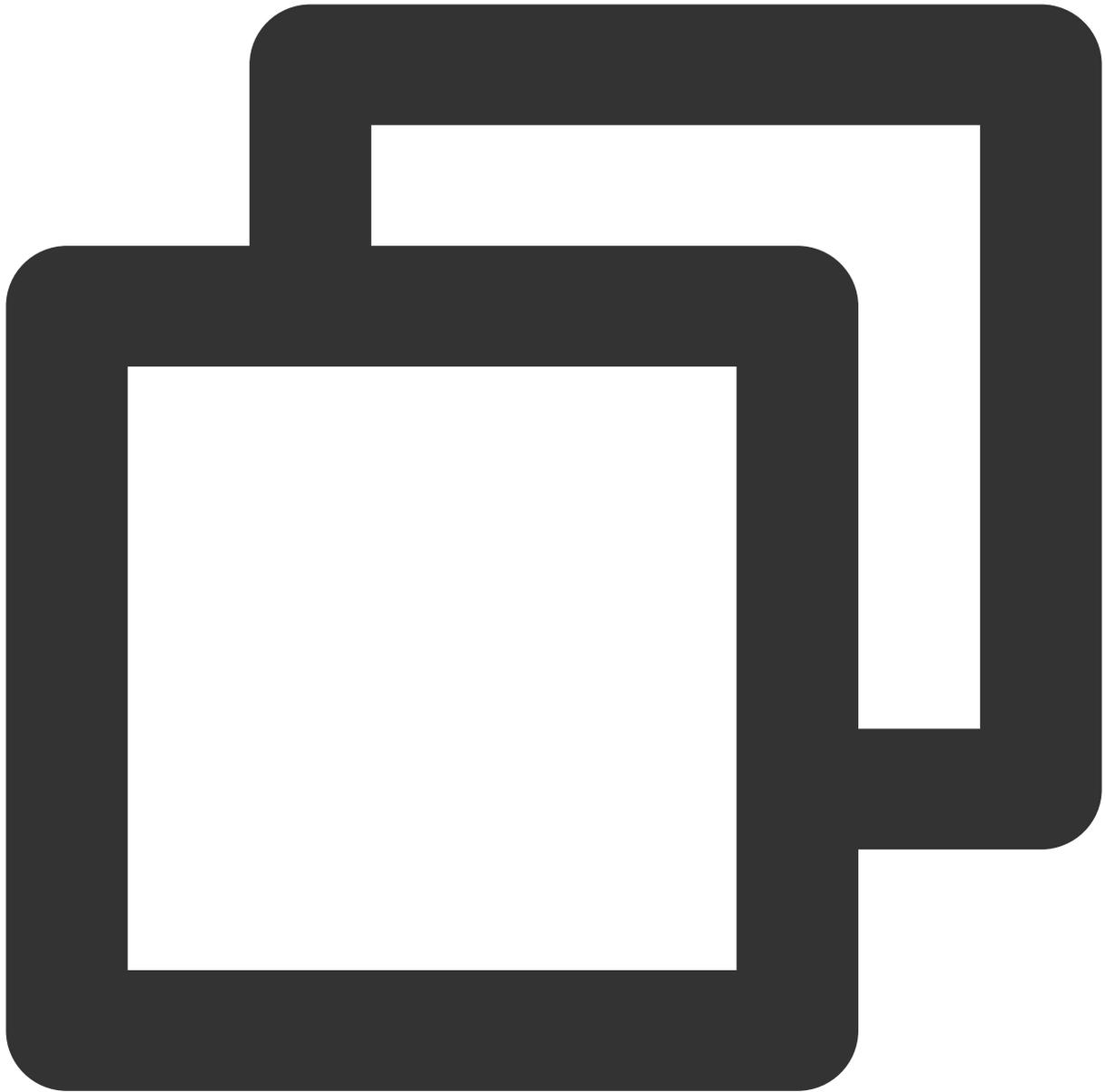


```
coscmd download -rf / D:/examplefolder/
```

注意：

若本地存在同名文件，则会下载失败，需要使用 `-f` 参数覆盖本地文件。

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

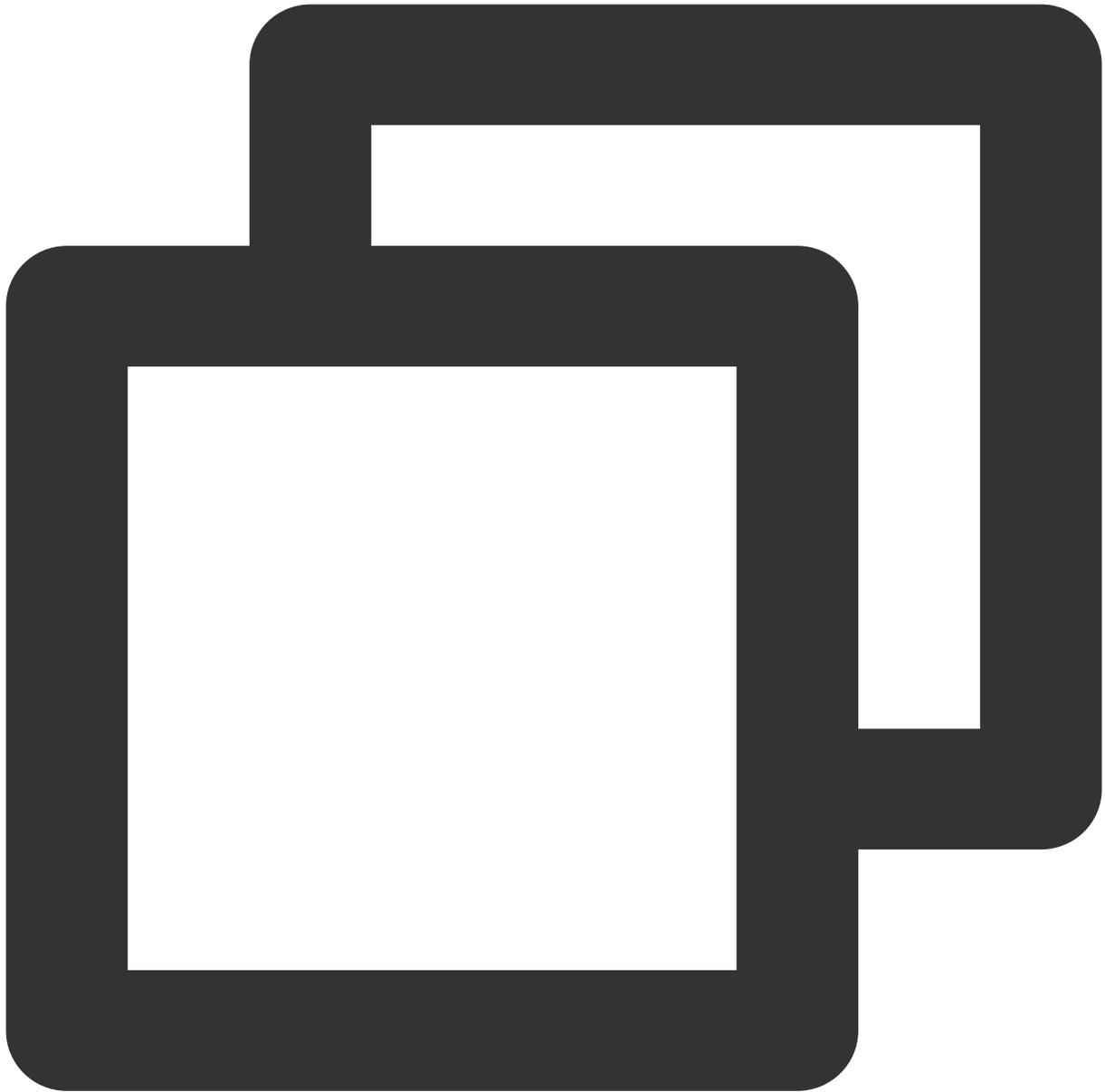


```
coscmd download -rs / D:/examplefolder
```

注意：

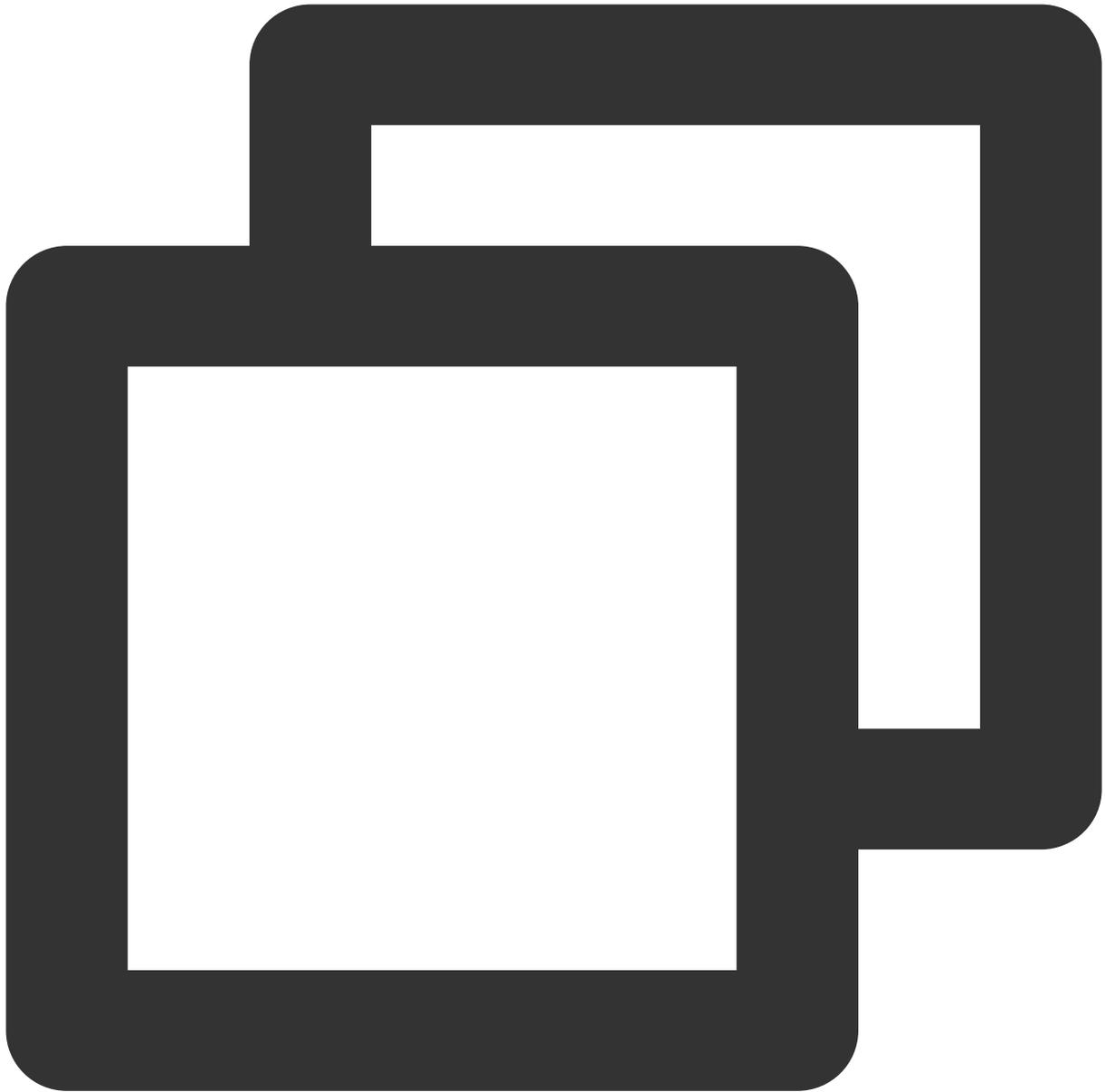
使用 `-s` 或者 `--sync` 参数，可以在下载文件夹时跳过本地已存在的相同文件（前提是下载的文件是通过 COSCMD 的 `upload` 接口上传的，文件携带有 `x-cos-meta-md5` 头部）。

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



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

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

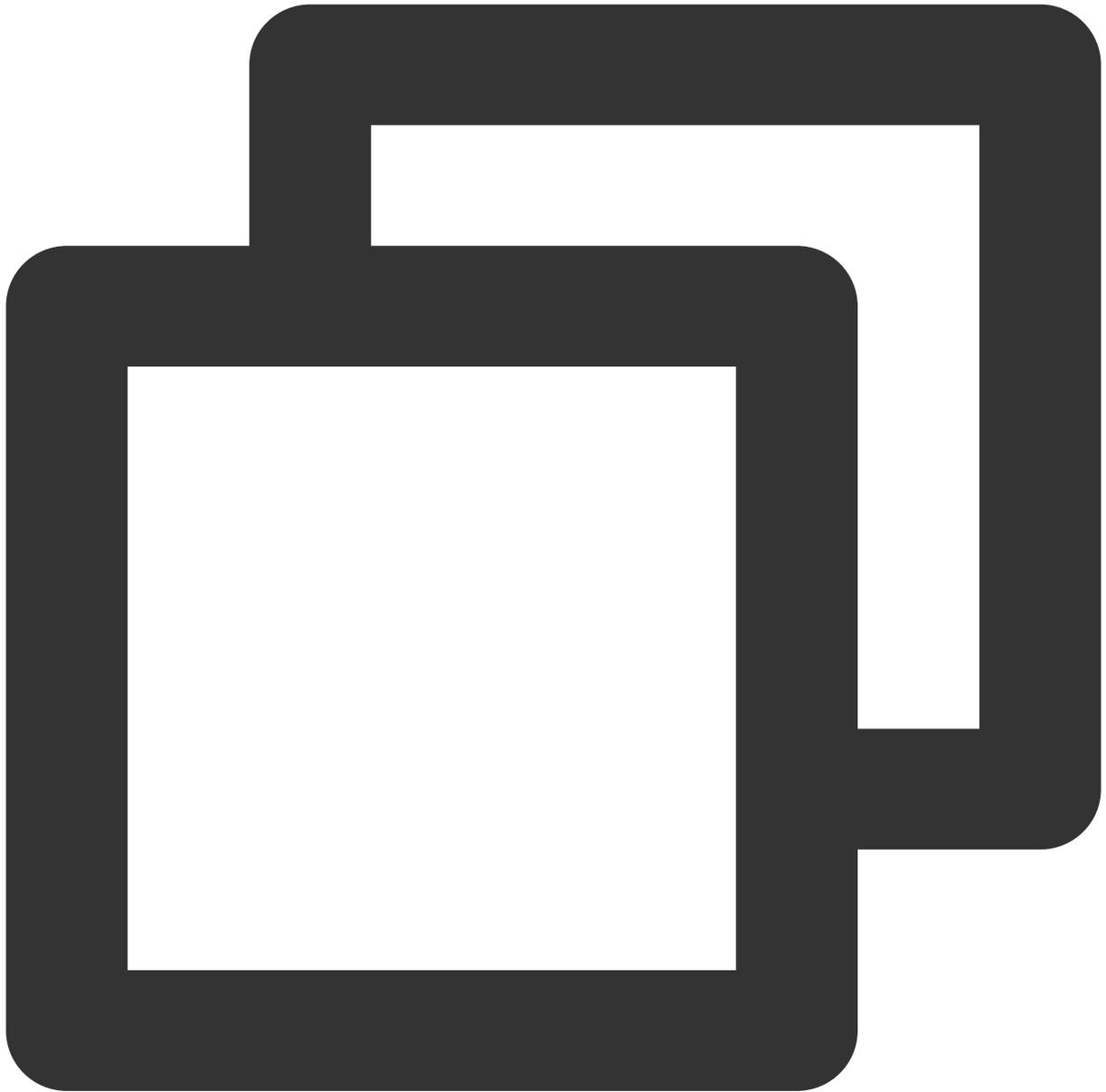


```
coscmd download -rs --delete / D:/doc /
```

说明：

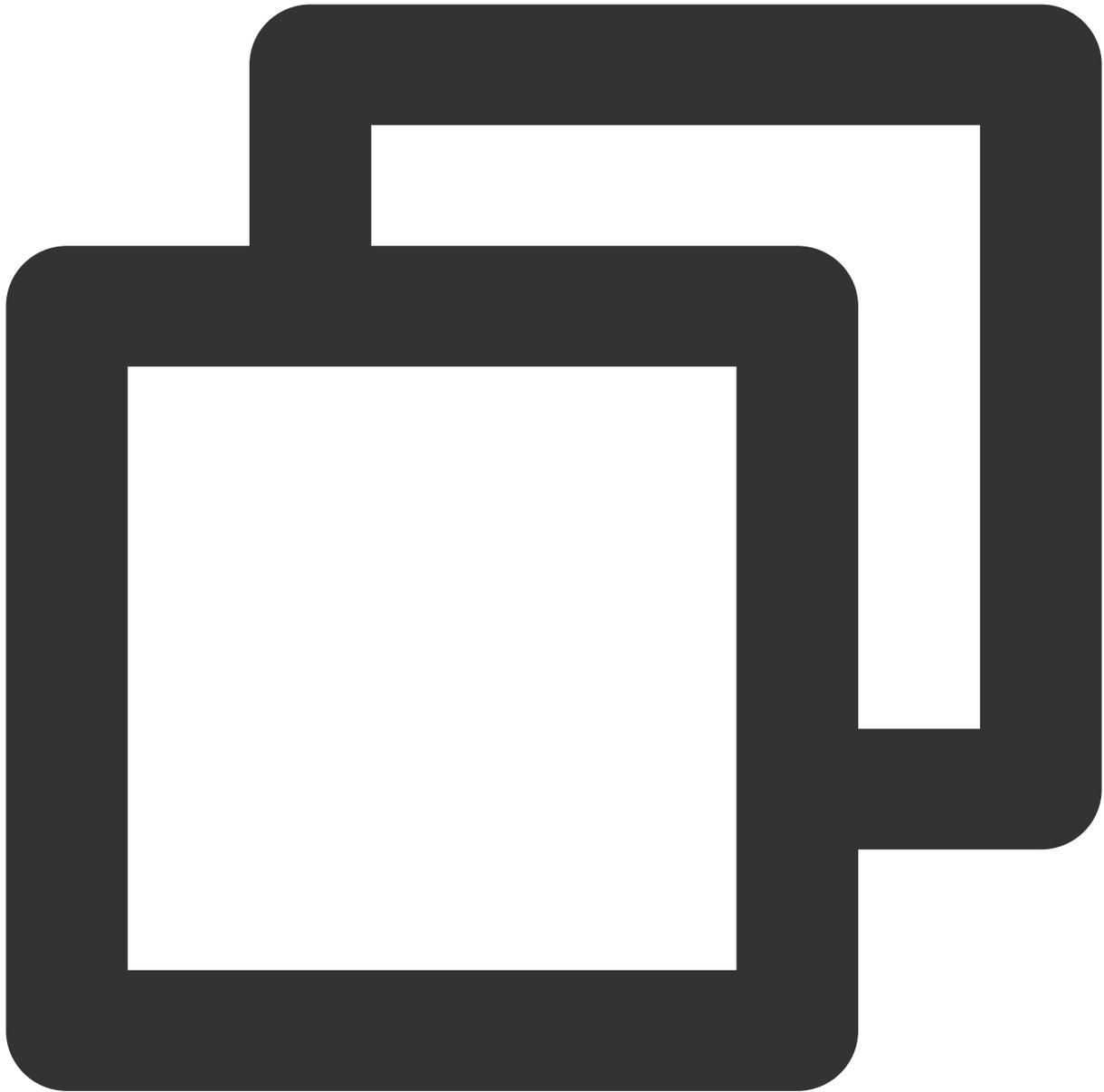
例如，当天 D 盘 doc 文件夹中的文件与 COS 中 doc 路径下的文件已同步上传，源端与目标端一致。第二天用户把 COS 中 doc 路径下的 A 文件删除了，此时执行以上命令，会删除 D 盘 doc 文件夹下的 A 文件，最终实现 D 盘 doc 文件夹中的文件与 COS 中 doc 路径下的文件保持一致。

操作示例 - 忽略 .txt 和 .doc 的后缀文件



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

操作示例 - 忽略 .txt 后缀的文件



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

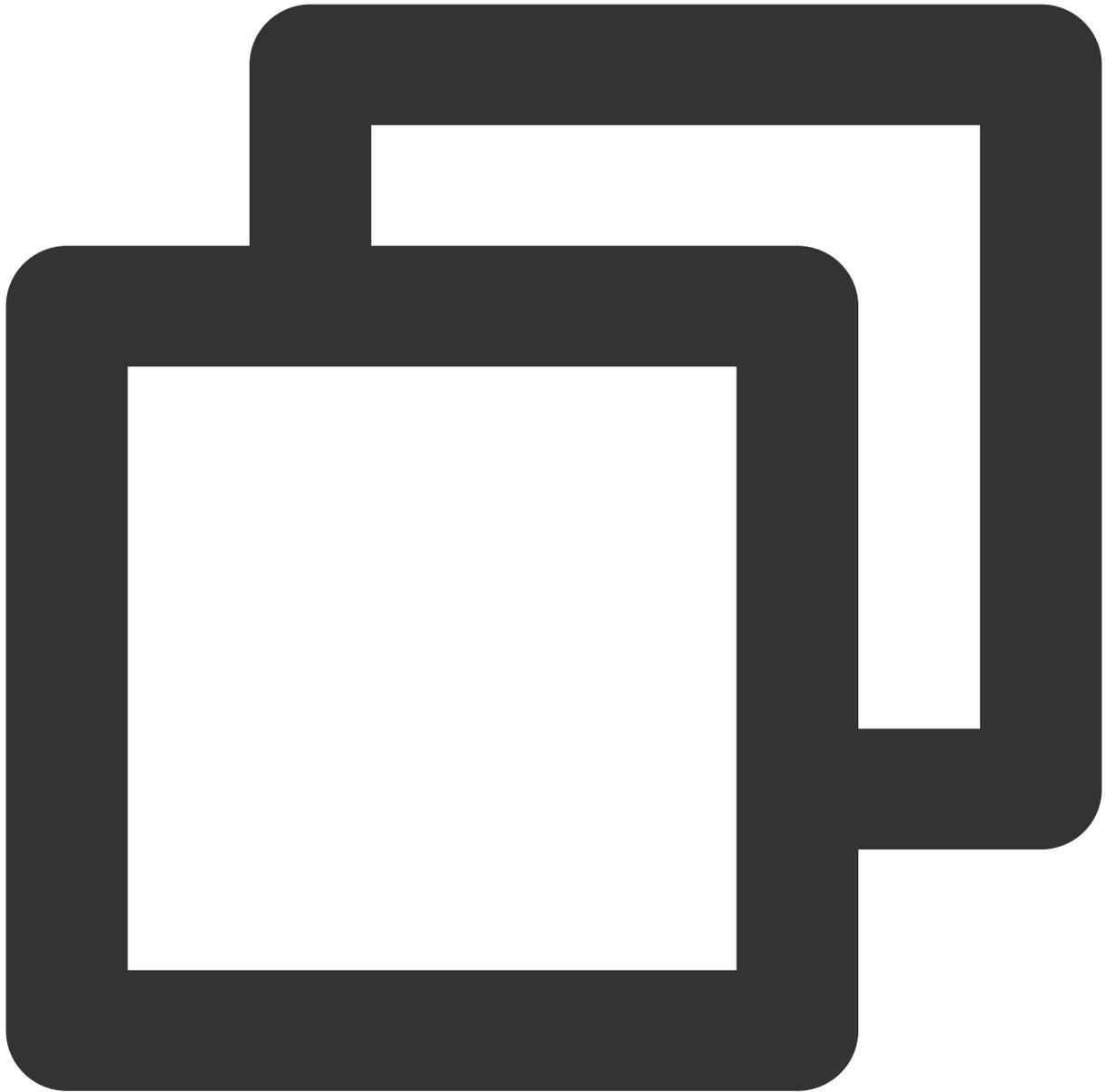
注意：

在上传文件夹时，使用 `--ignore` 参数可以忽略某一类文件，使用 `--include` 参数可以过滤某一类文件，支持 `shell` 通配规则，支持多条规则，用逗号 `,` 分隔。当忽略一类后缀时，必须最后要输入 `,` 或者加入 `"`。如果 `"` 中包含多条逗号分隔的规则，以第一条规则为准。

如果您希望使用 `--ignore` 过滤特定目录内的所有文件，需要使用绝对路径，并在路径前后加入 `"`。例如

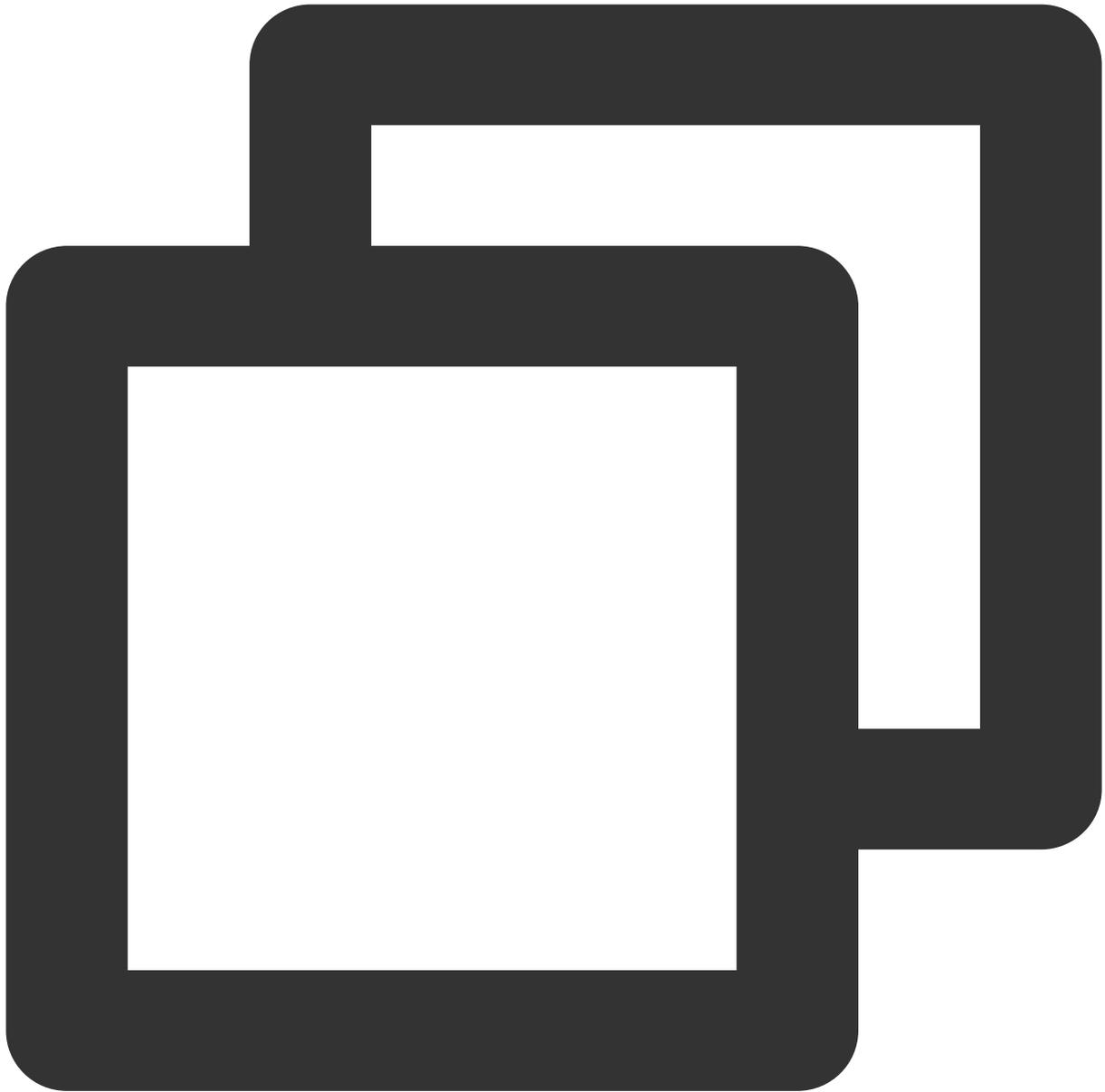
```
coscmd upload -rs D:/doc / --ignore "D:/doc/ignore_folder/*"
```

操作示例 - 过滤 `.txt` 和 `.doc` 的后缀文件



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

操作示例 - 过滤 .txt 后缀的文件



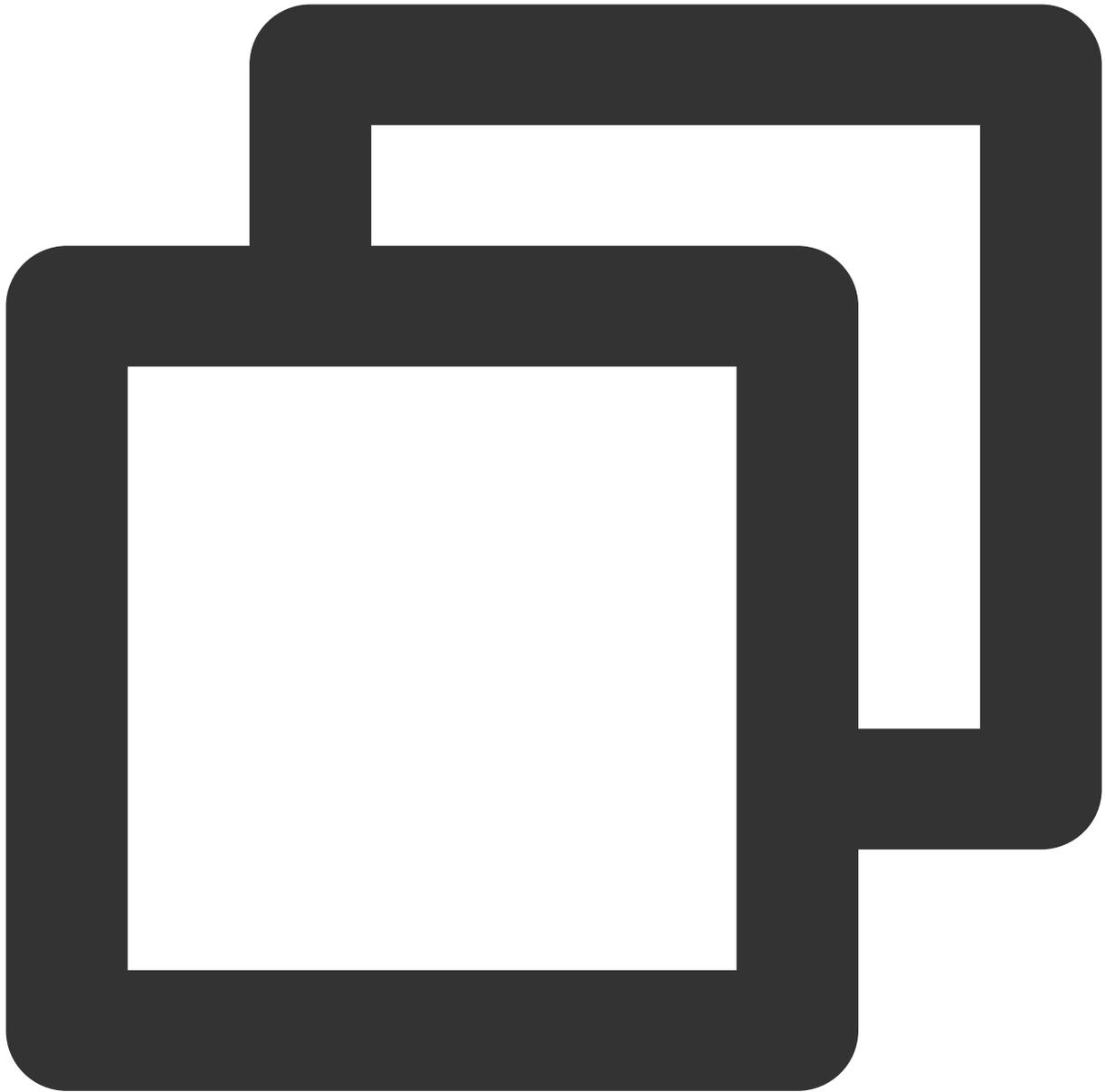
```
coscmd download -rs / D:/examplefolder --include "*.txt"
```

注意：

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

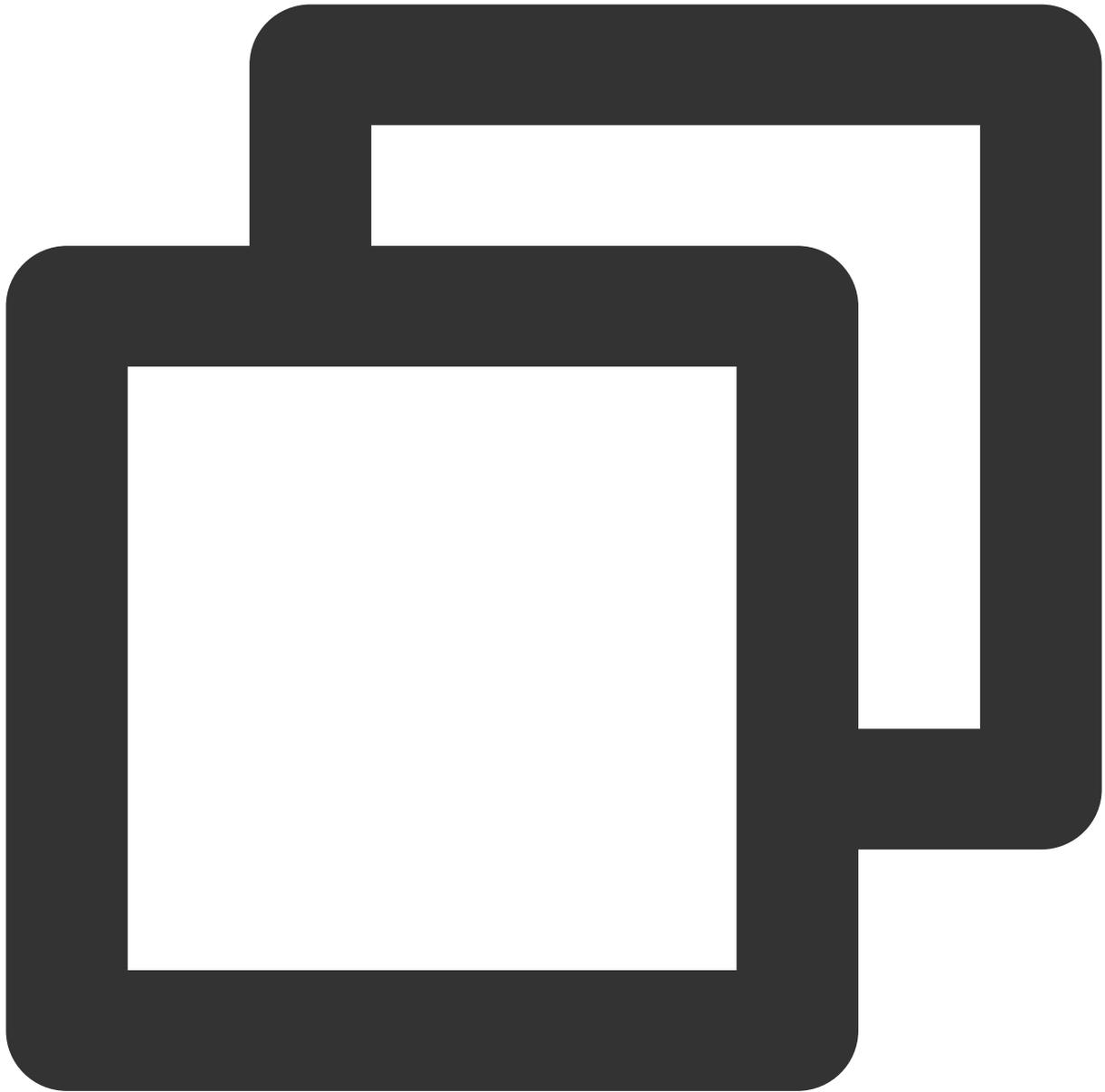
获取带签名的下载 URL

命令格式



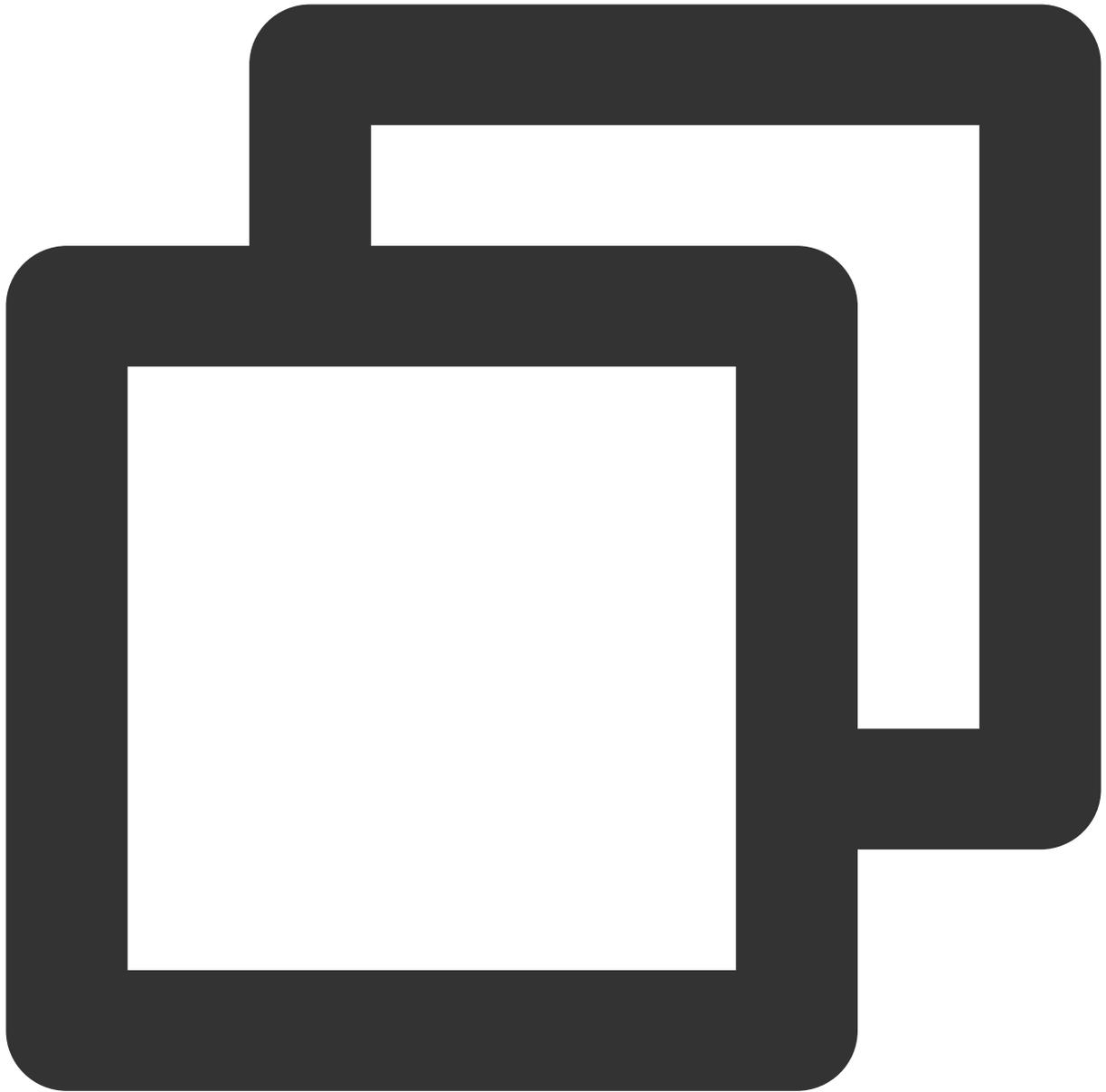
```
coscmd signurl <cospath>
```

操作示例 - 生成 doc/picture.jpg 路径带签名的 URL



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

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



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

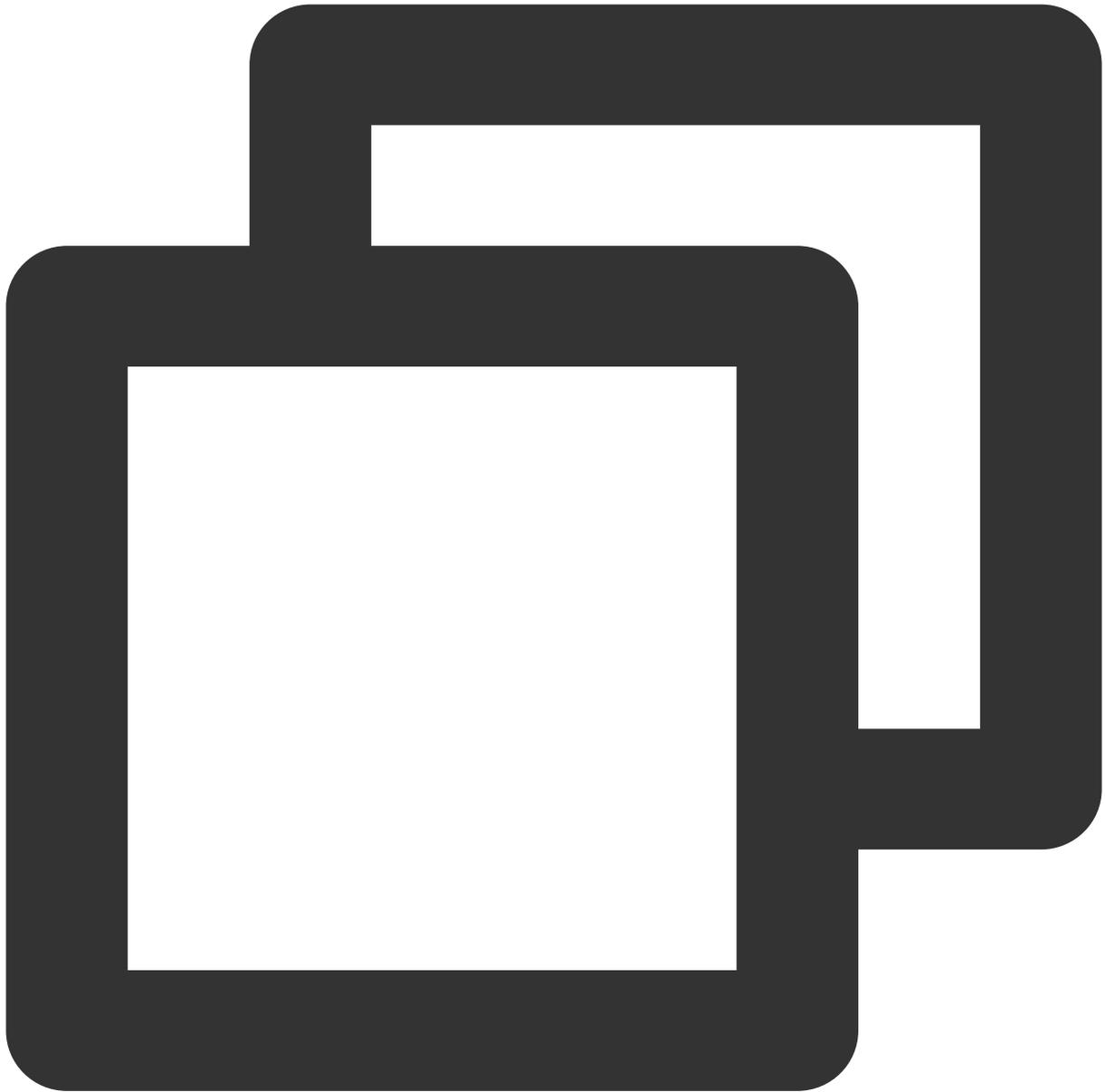
说明：

请将 "<>" 中的参数替换为您需要获取下载 URL 的 COS 上文件的路径（cospath）。

使用 `-t time` 设置该 URL 中签名的有效时间（单位为秒），默认为10000s。

删除文件或文件夹

删除文件命令格式

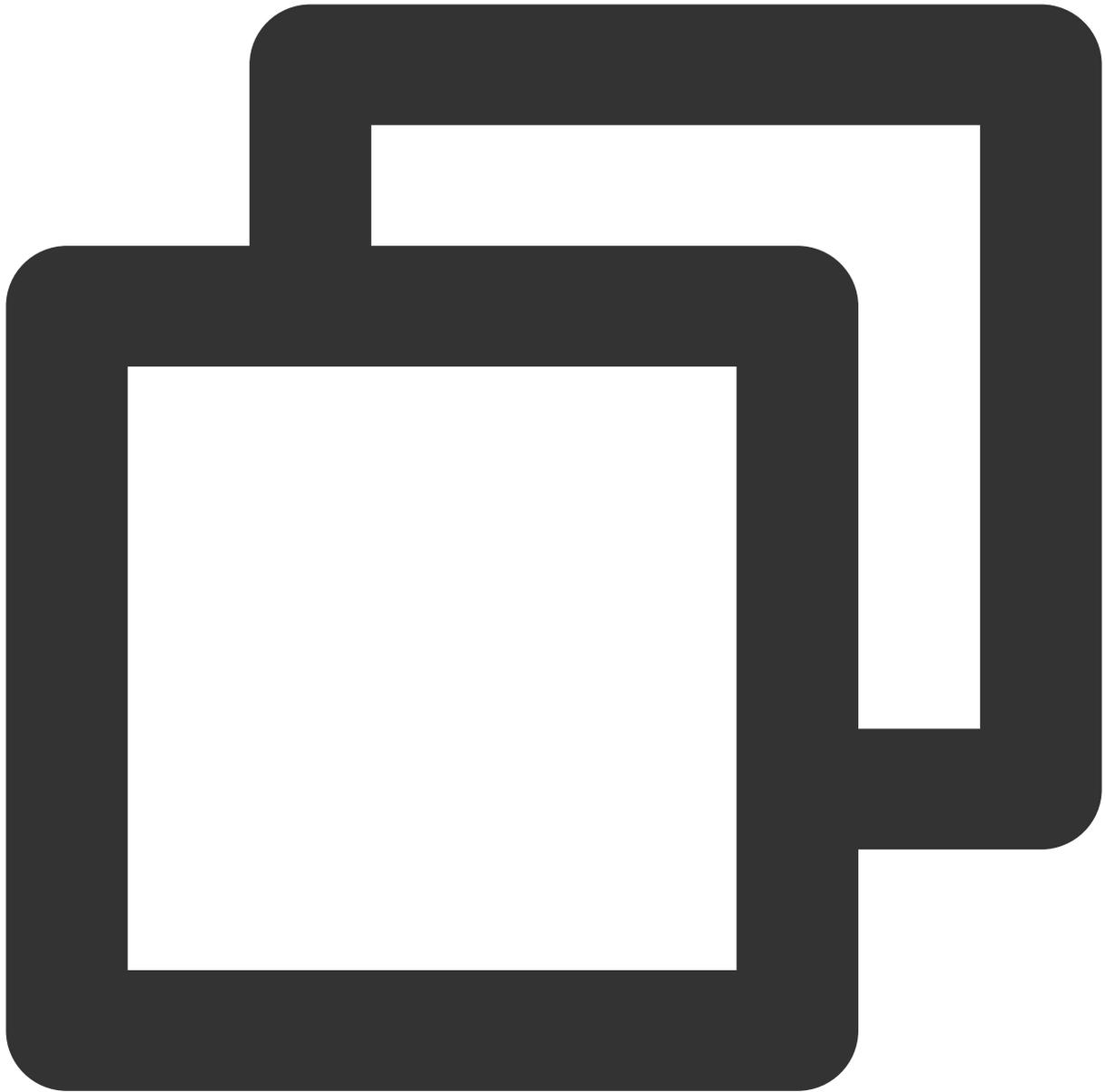


```
coscmd delete <cospath>
```

注意：

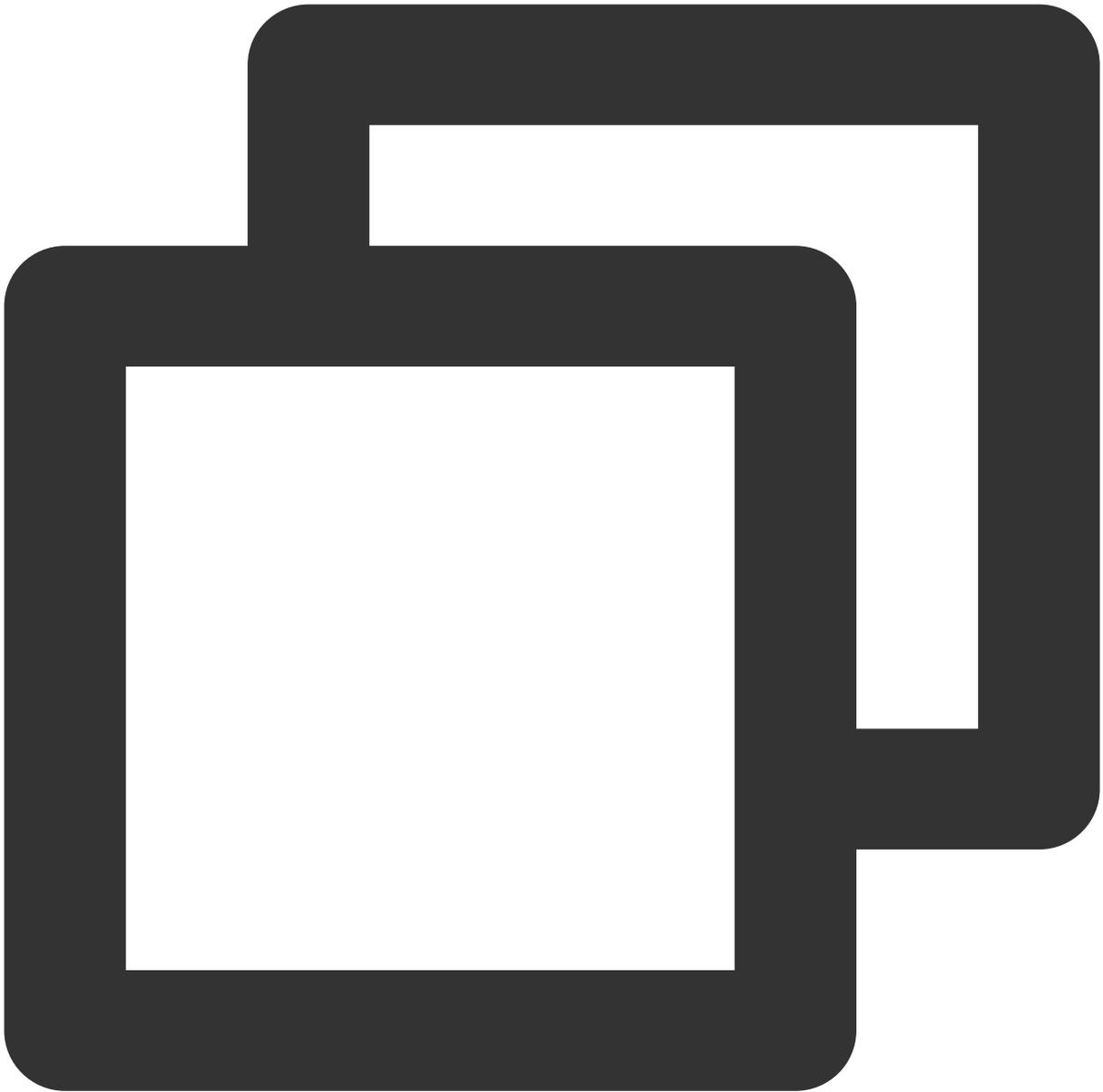
请将 "<>" 中的参数替换为您需要删除的 COS 上文件的路径（cospath），工具会提示用户是否确认进行删除操作。

操作示例 - 删除 doc/exampleobject.txt



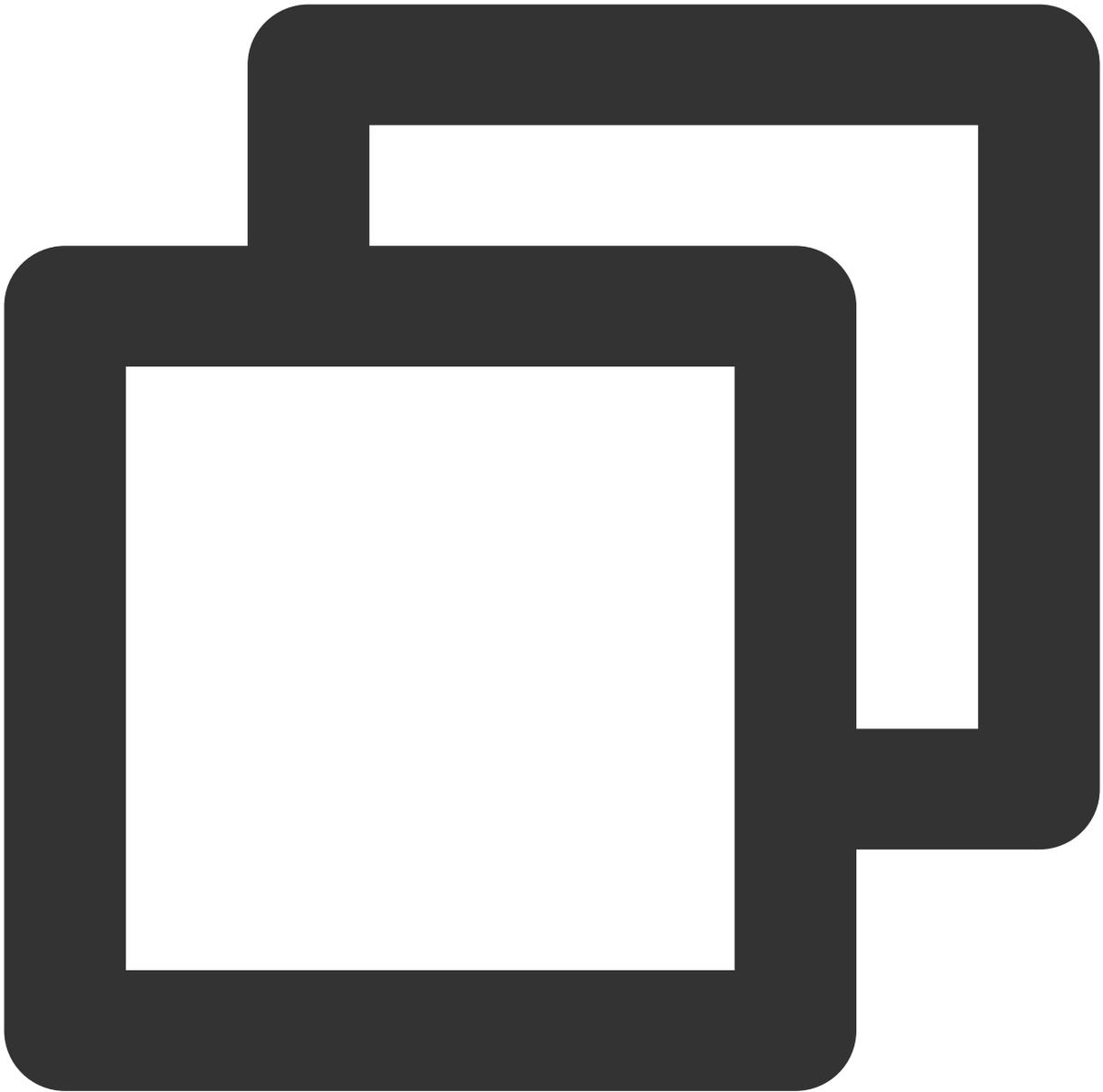
```
coscmd delete doc/exampleobject.txt
```

操作示例 - 删除带有版本 ID 的文件



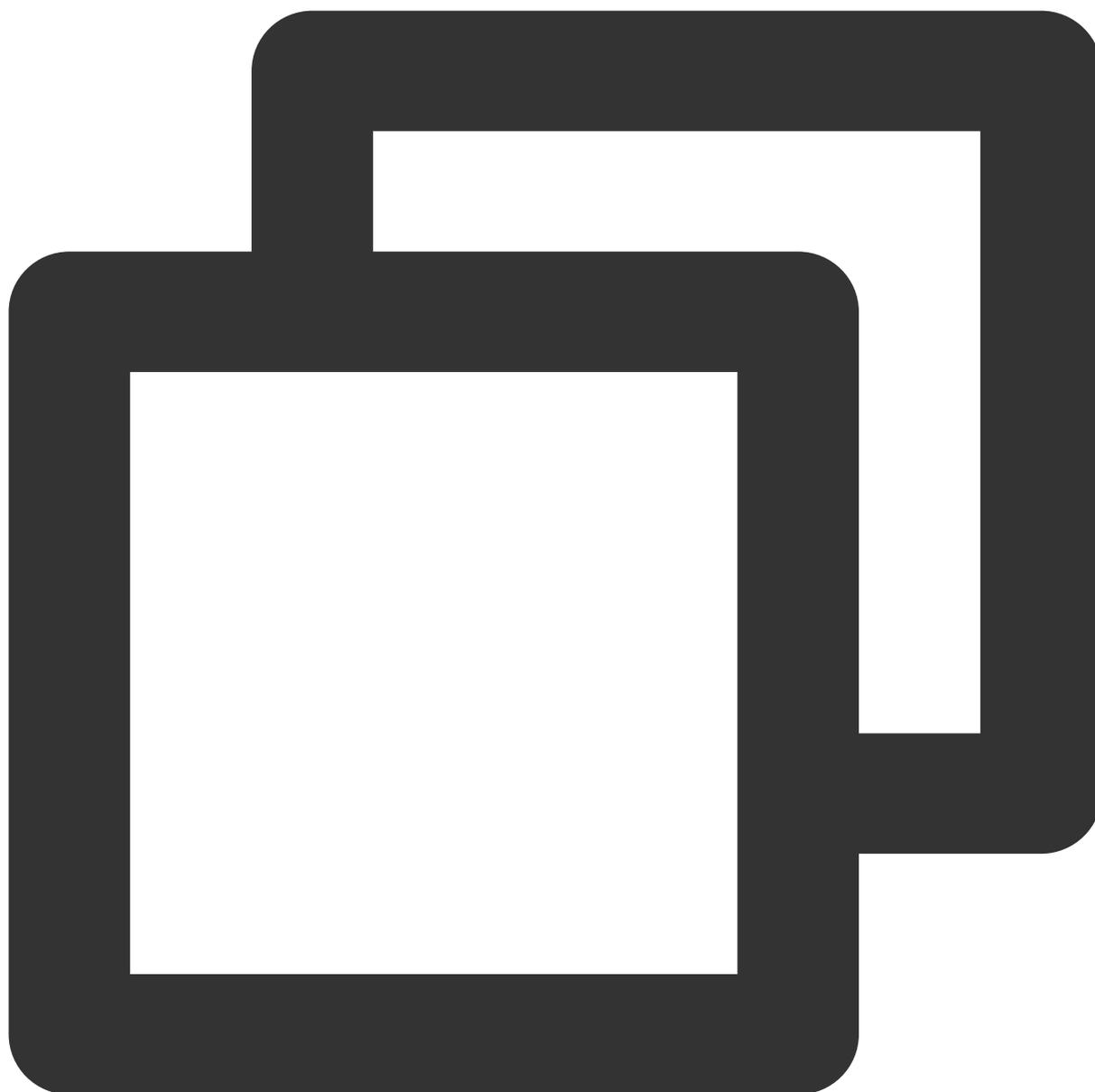
```
coscmd delete doc/exampleobject.txt --versionId MTg0NDUxMzc4ODA3NTgyMTErEWN
```

删除文件夹命令格式



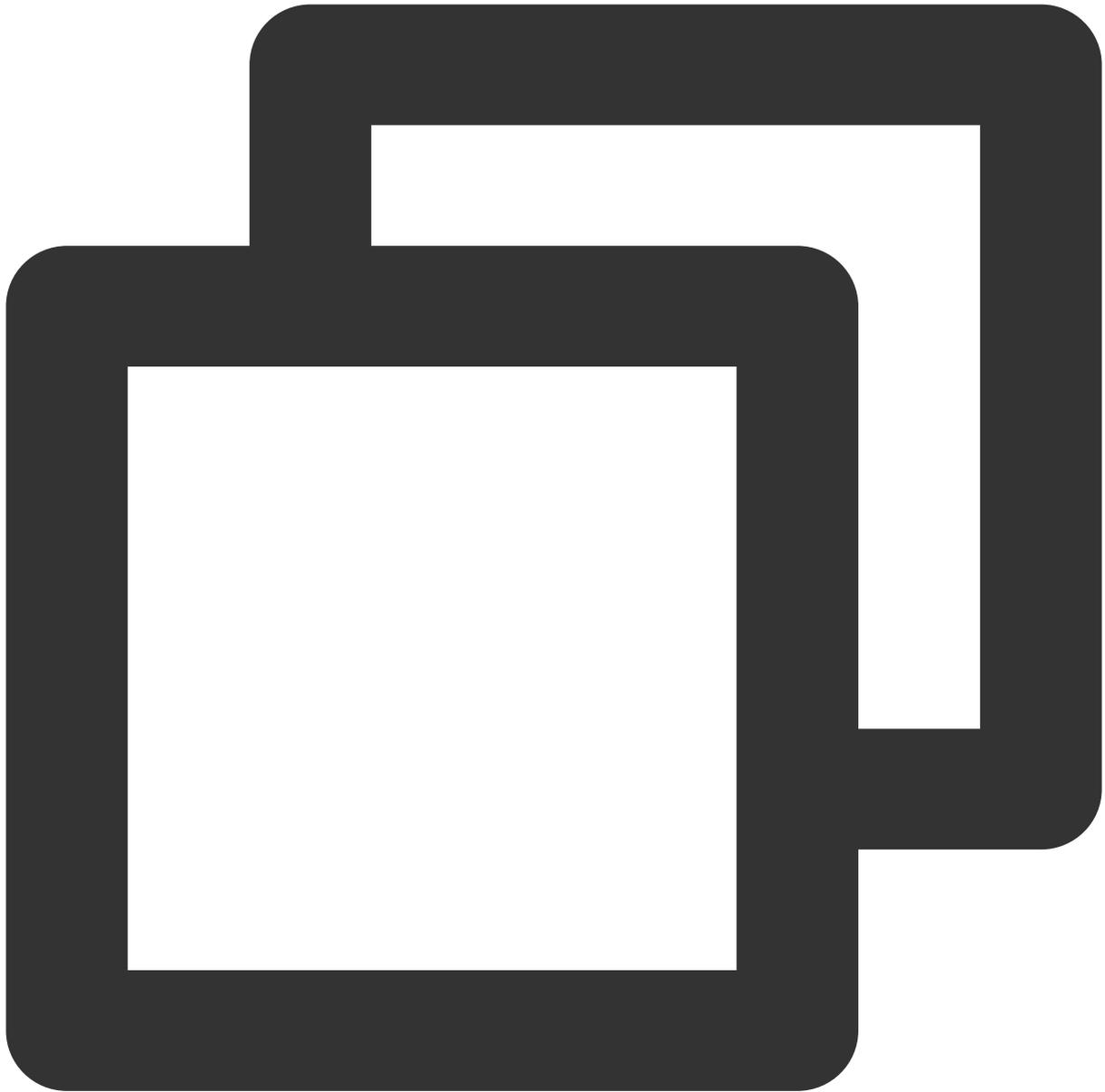
```
coscmd delete -r <cospath>
```

操作示例 - 删除 doc 目录



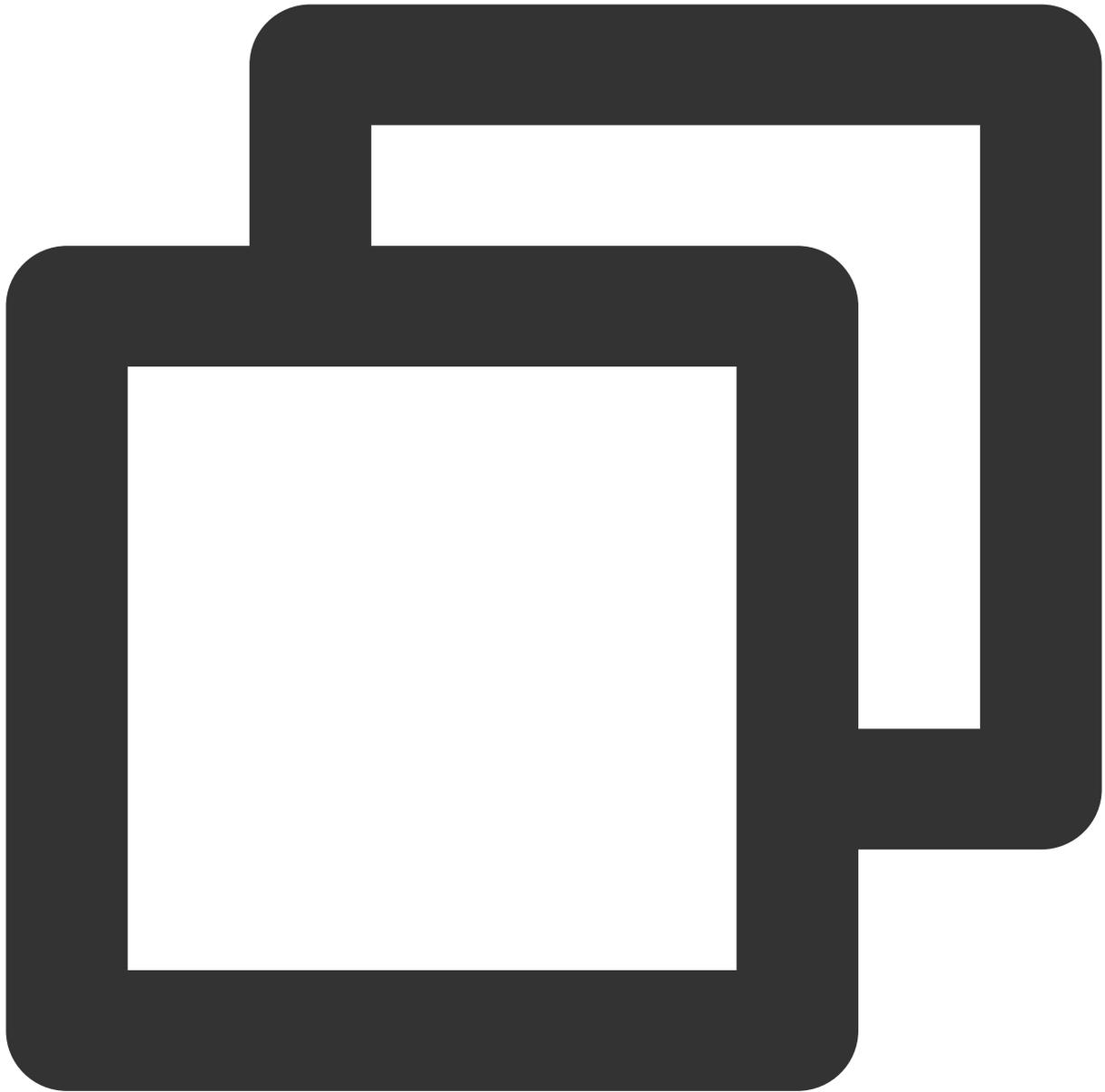
```
coscmd delete -r doc
```

操作示例 - 删除 folder/doc 目录



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

操作示例 - 删除 doc 文件夹下的所有版本控制文件



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

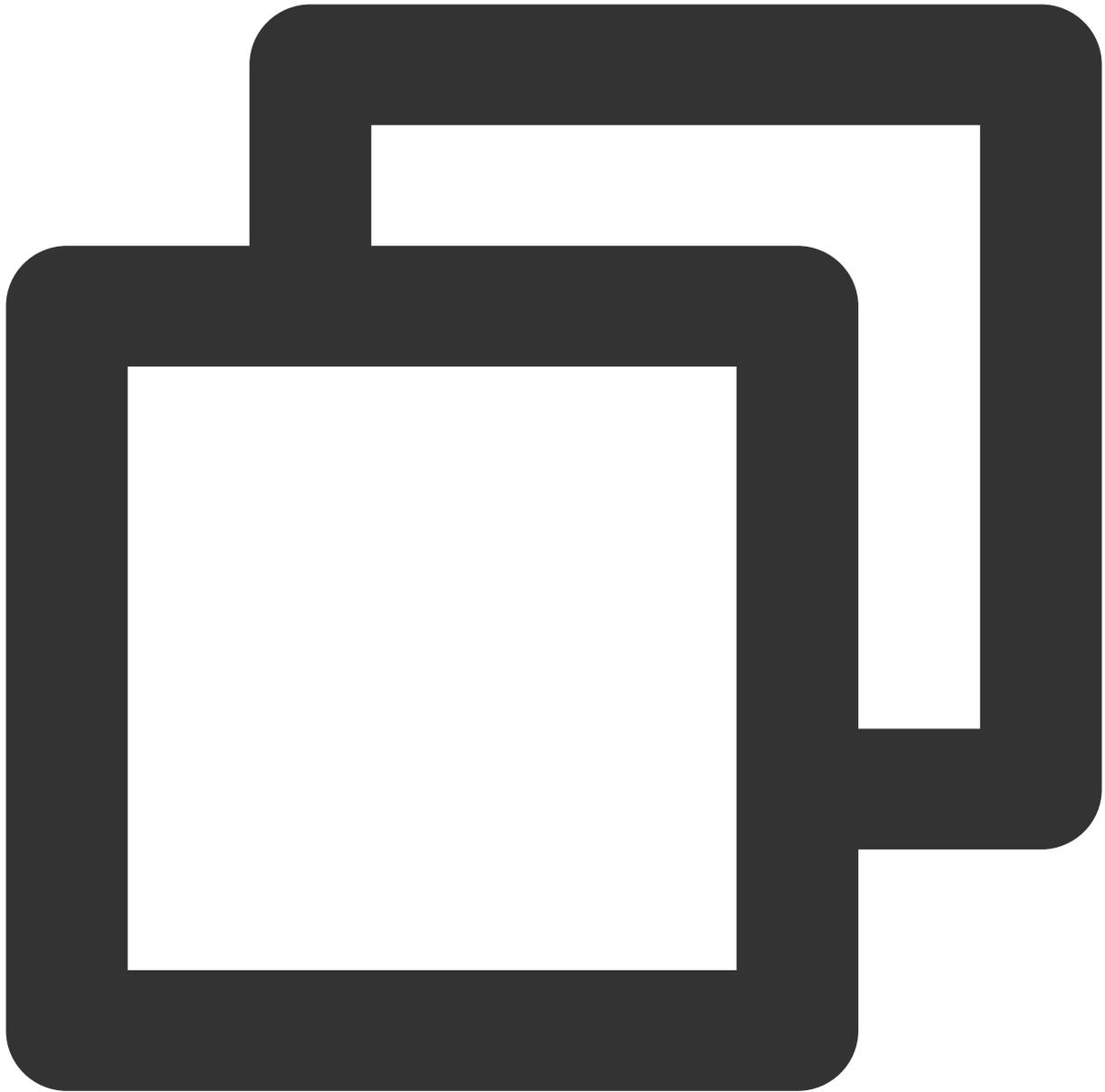
说明：

批量删除需要输入 `y` 确定，使用 `-f` 参数则可以跳过确认直接删除。

需注意，执行删除文件夹命令，将会删除当前文件夹及其文件。但删除版本控制的文件，需要指定版本 ID 进行删除。

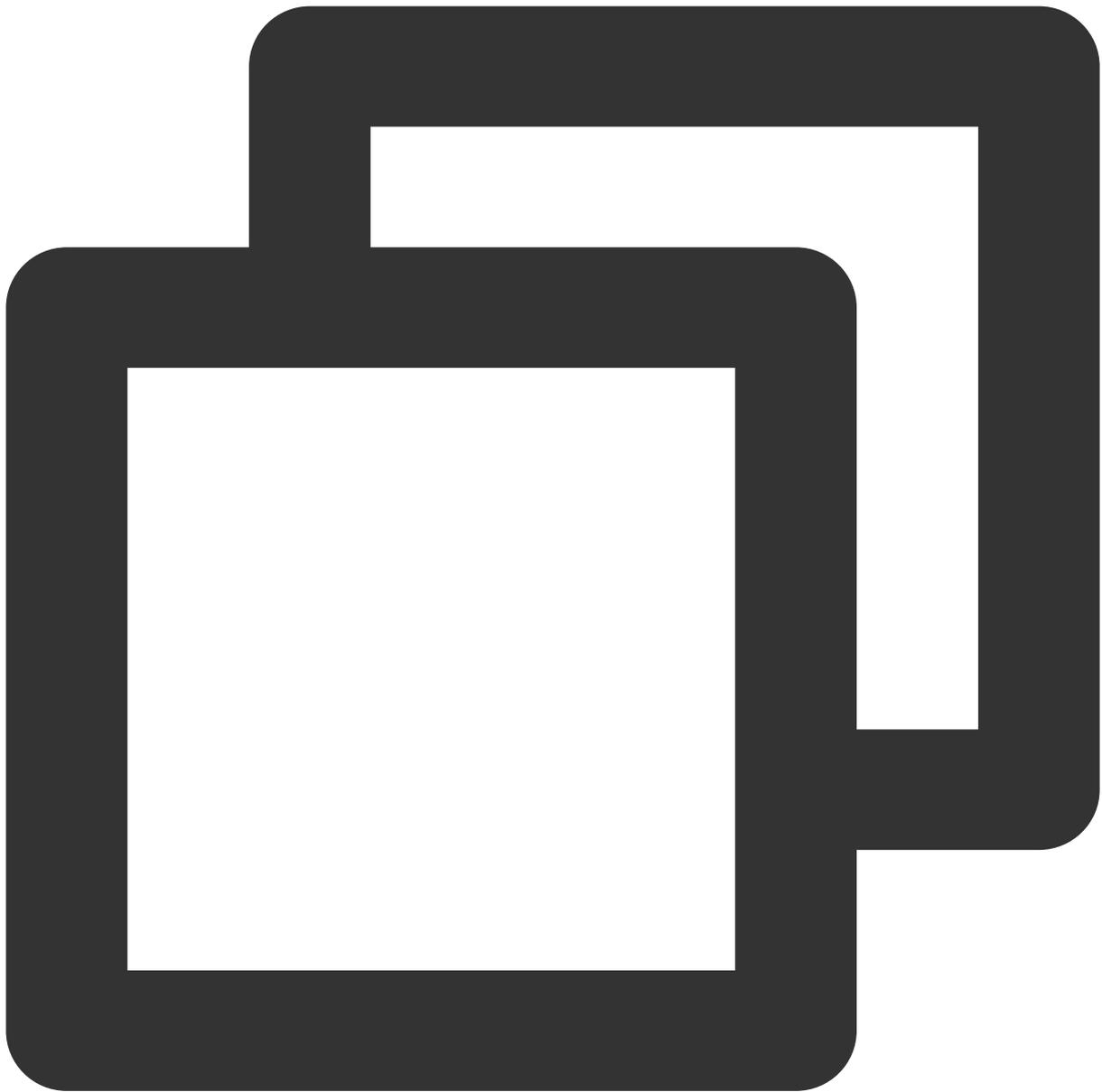
查询分块上传文件碎片

命令格式



```
coscmd listparts <cospath>
```

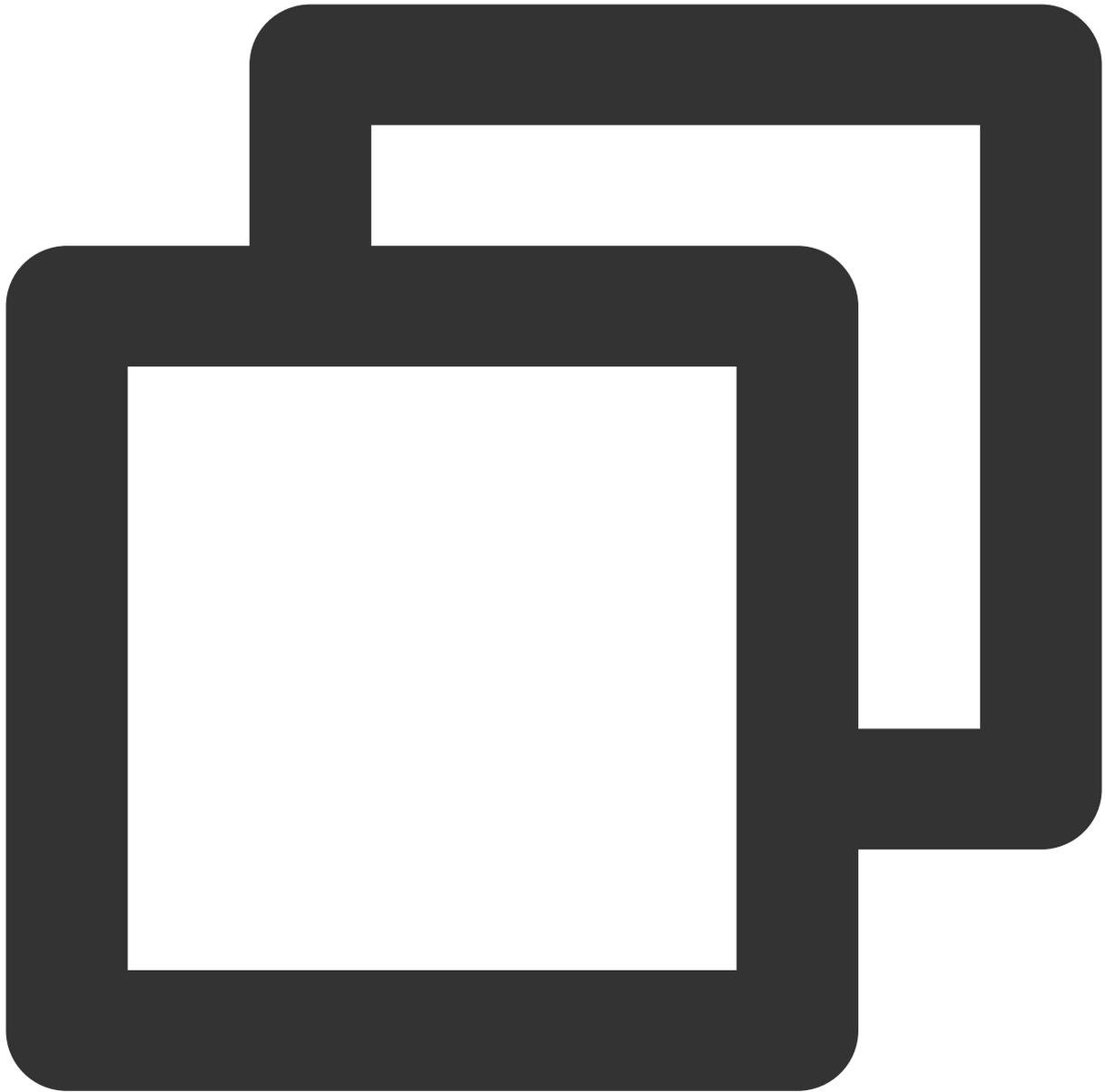
操作示例 - 查询 doc/ 前缀的文件碎片



```
coscmd listparts doc/
```

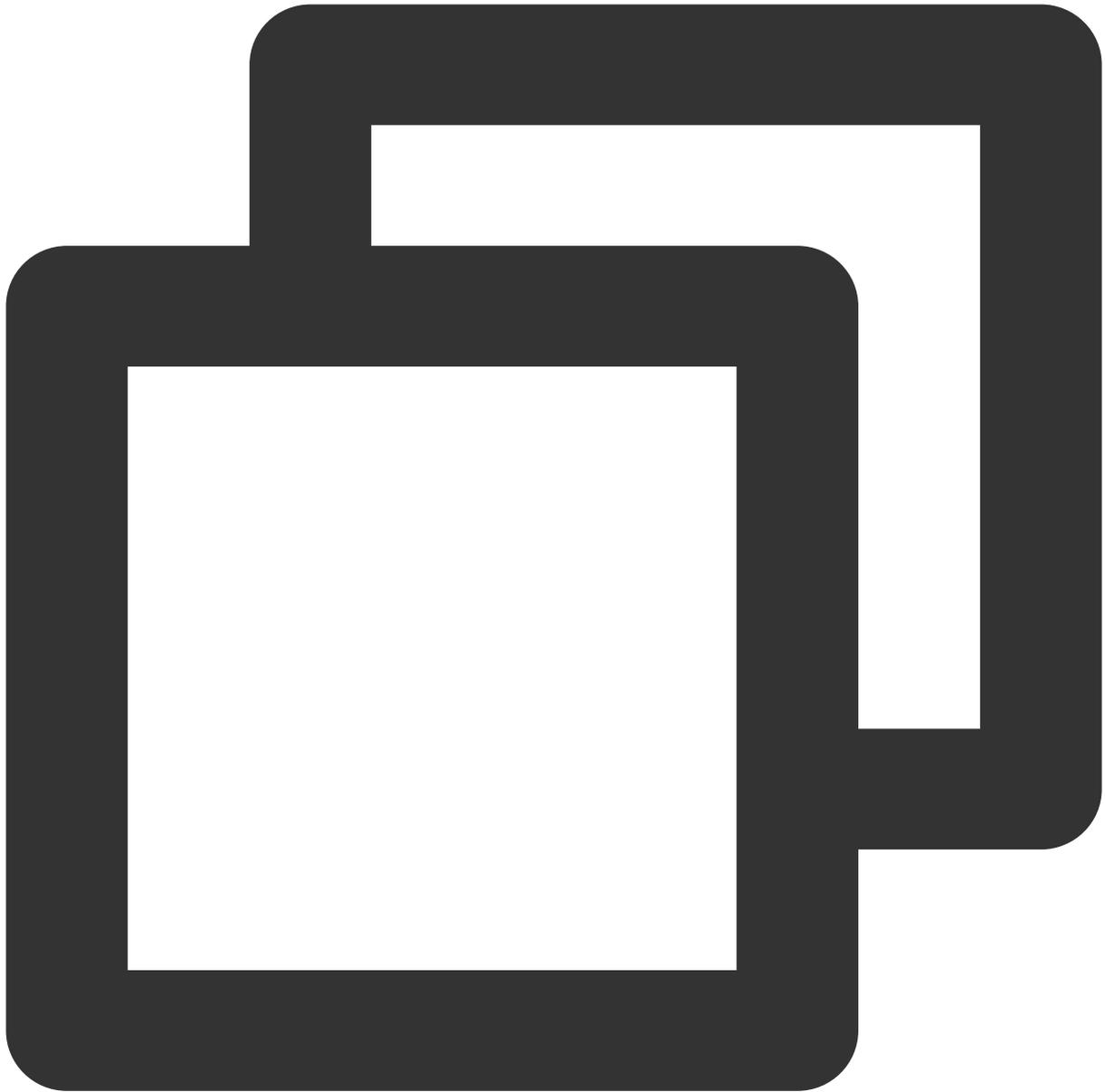
清除分块上传文件碎片

命令格式



```
coscmd abort
```

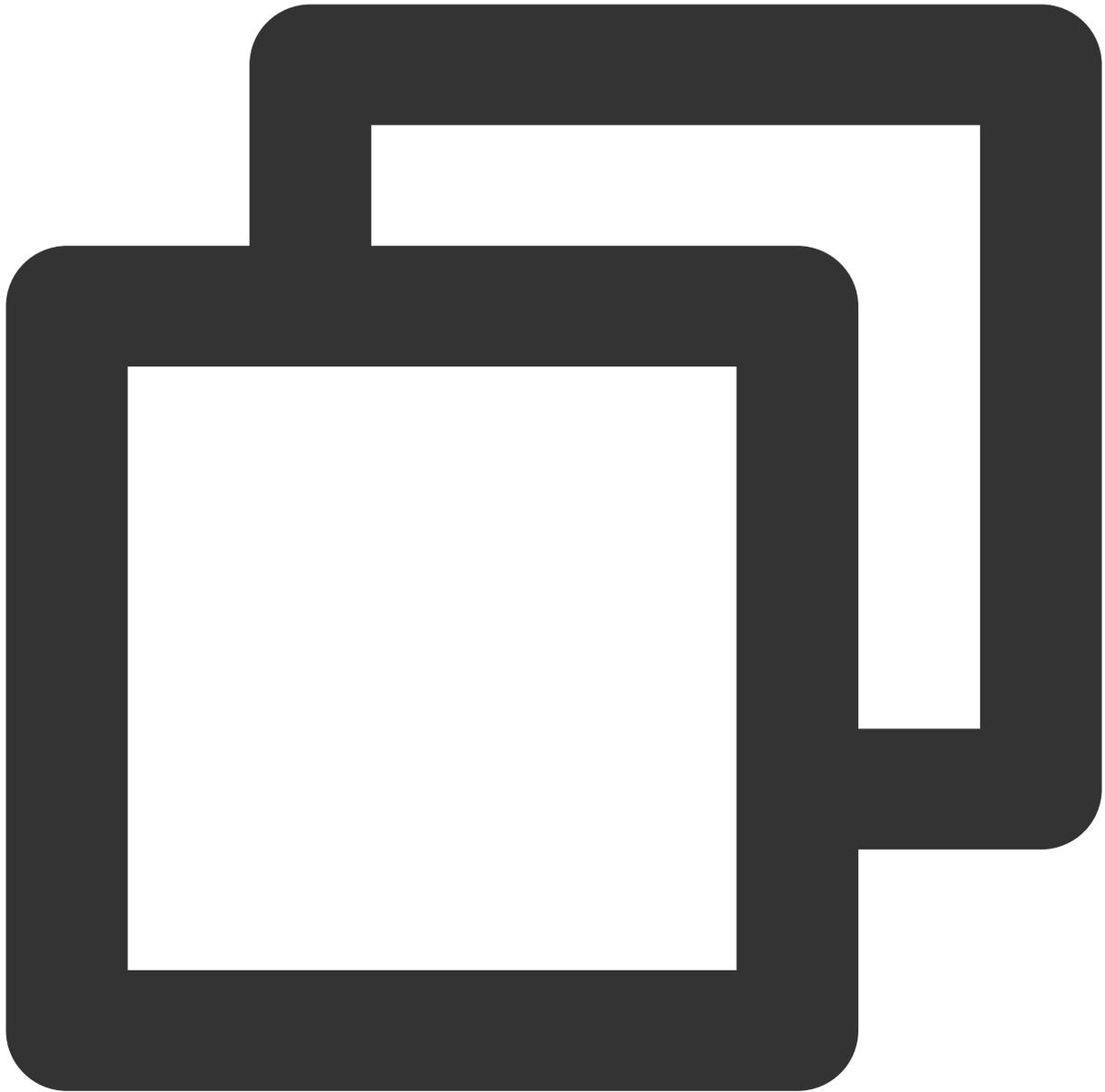
操作示例 - 消除所有上传文件碎片



```
coscmd abort
```

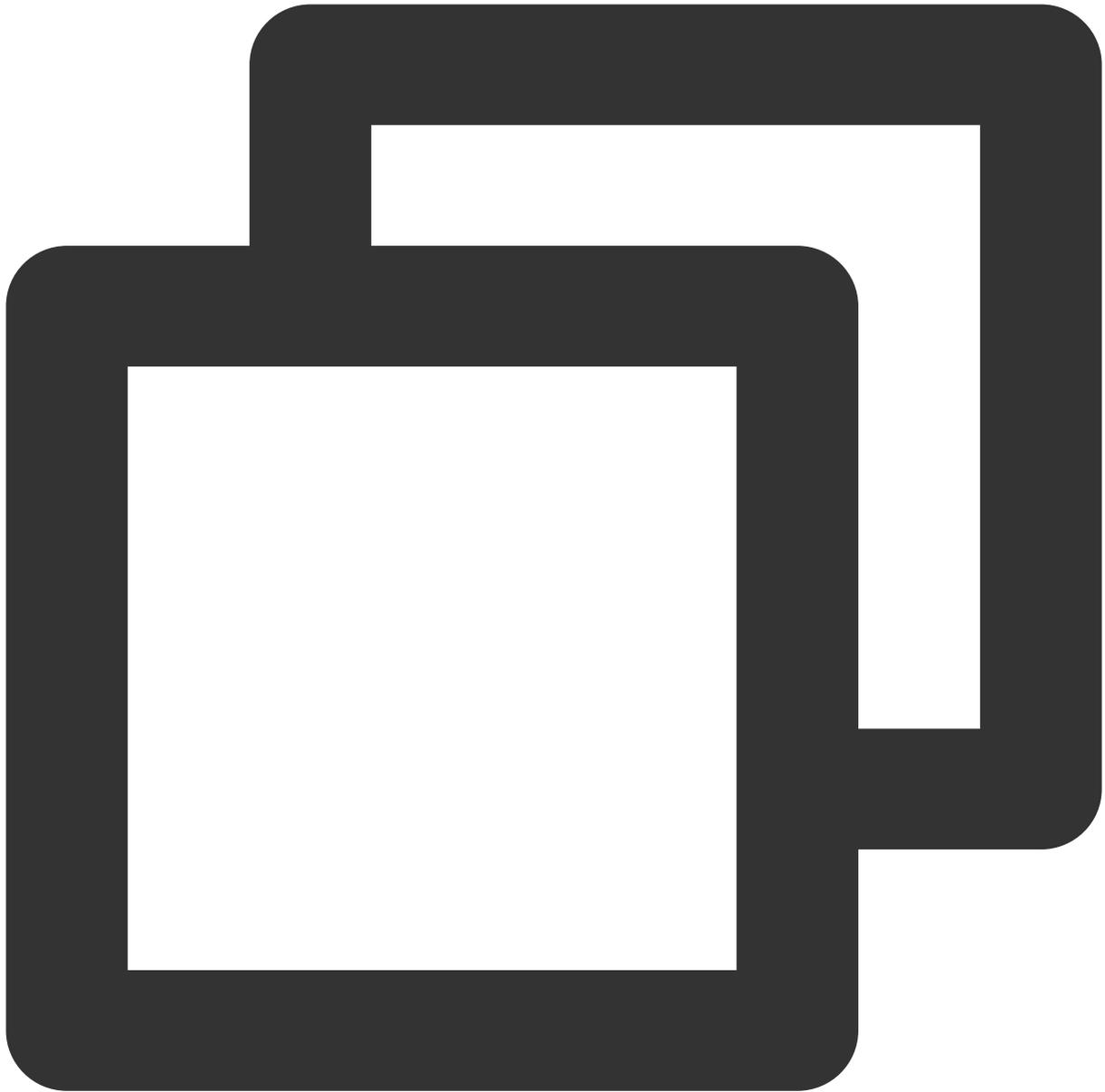
复制文件或文件夹

复制文件命令格式



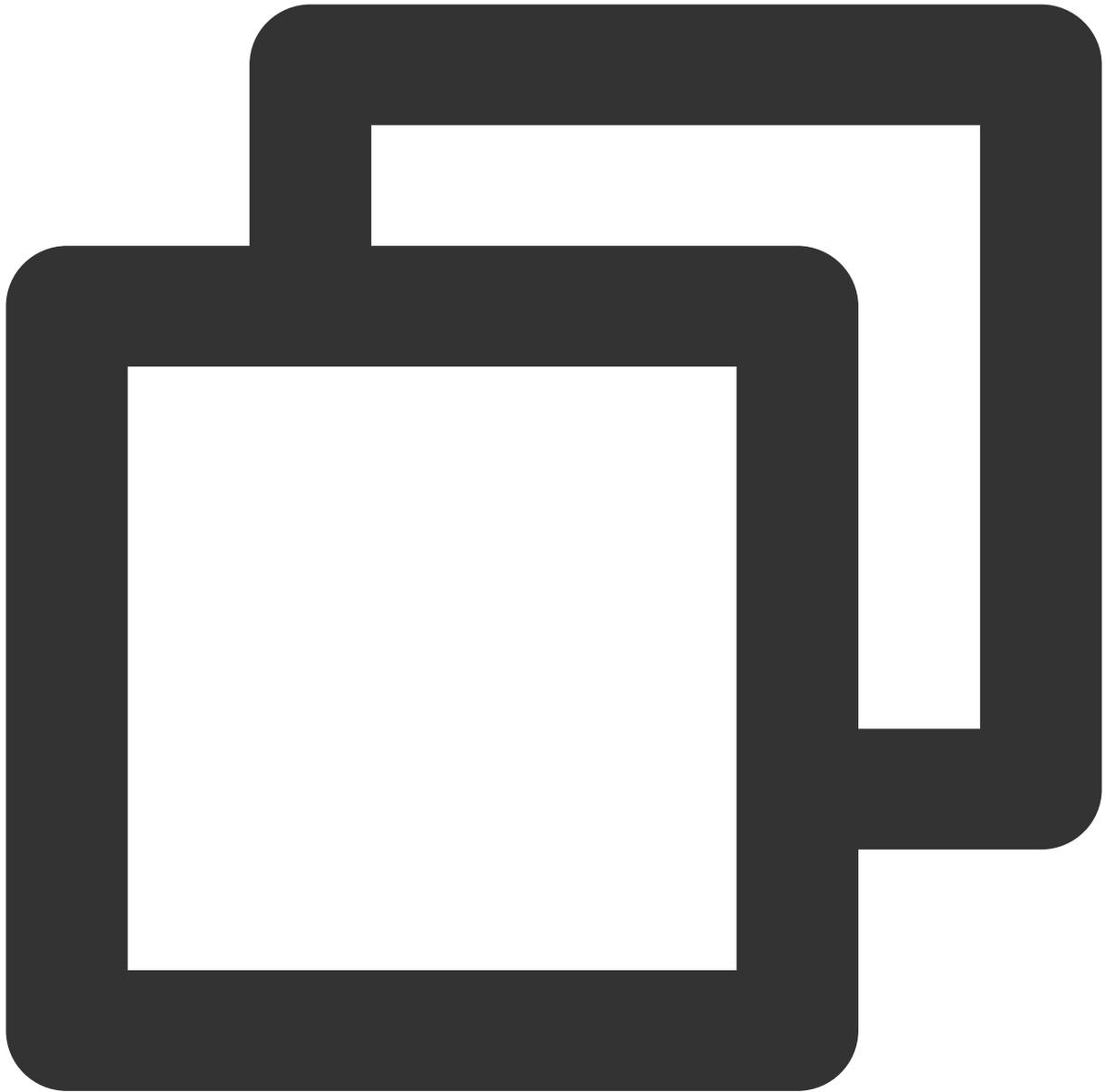
```
coscmd copy <sourcepath> <cospath>
```

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



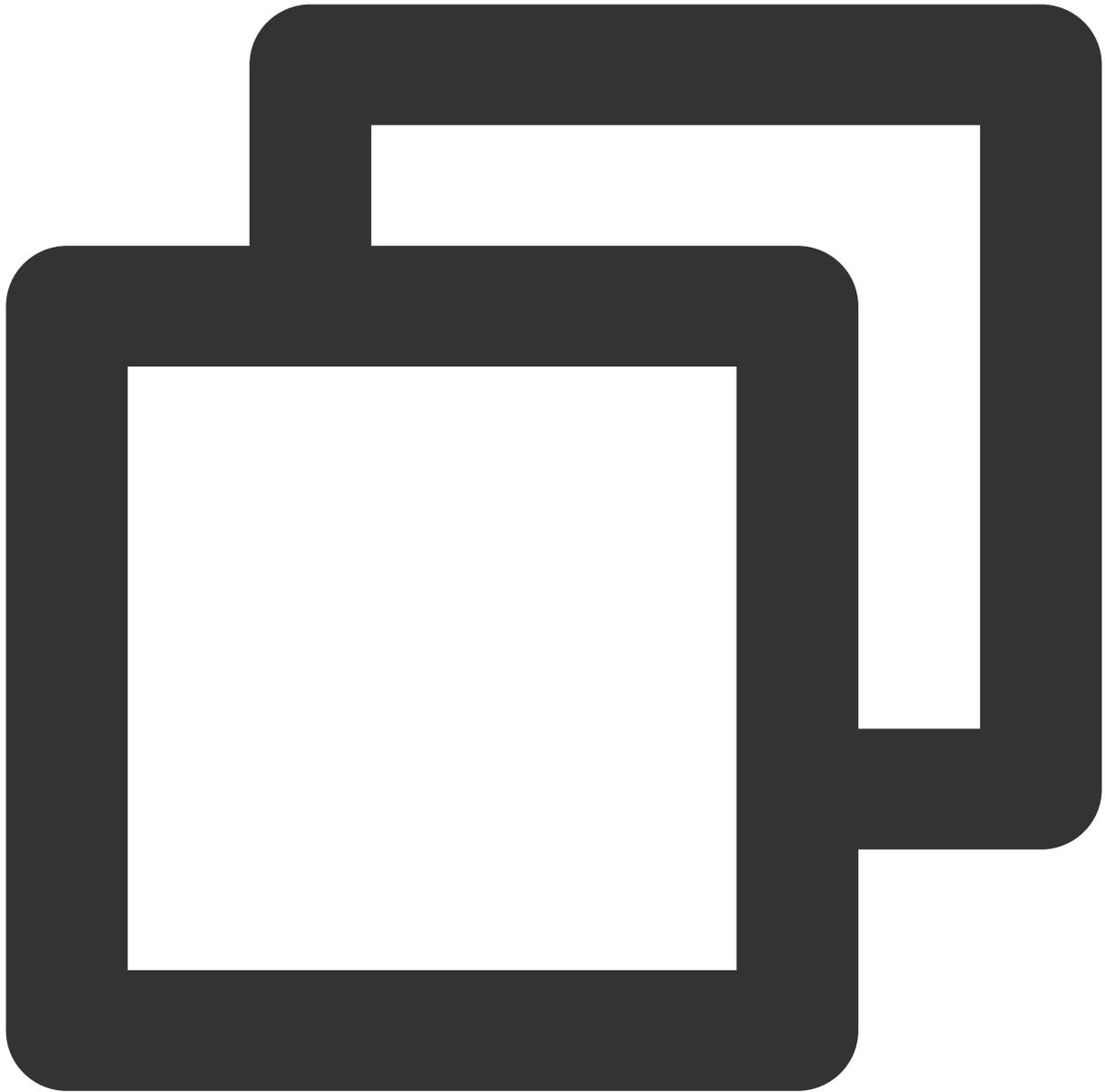
```
coscmd -b examplebucket-1250000000 -r ap-chengdu copy examplebucket-1250000000.ap-c
```

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



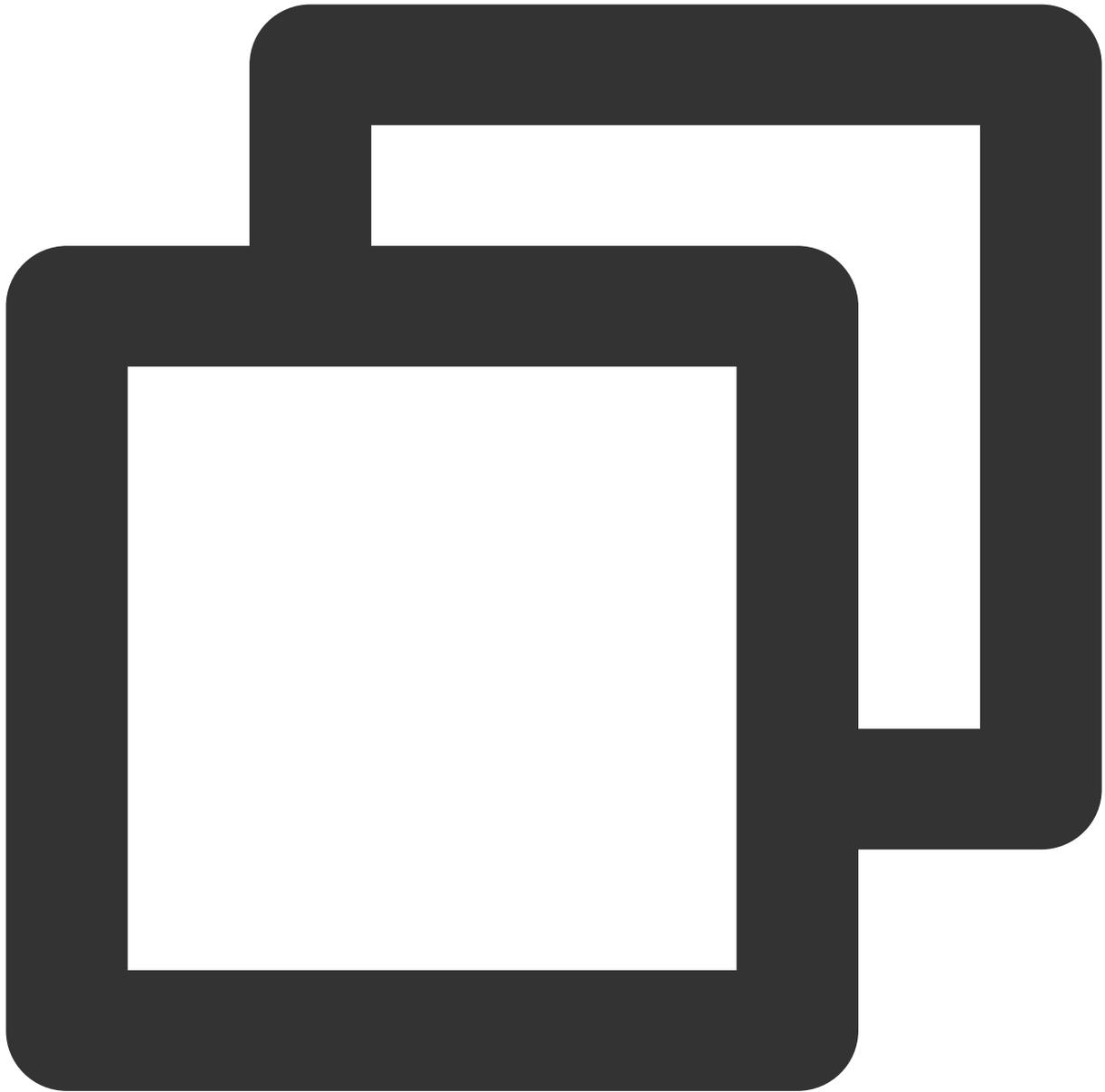
```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy examplebucket2-1250000000.
```

修改存储类型，将文件类型改为低频存储



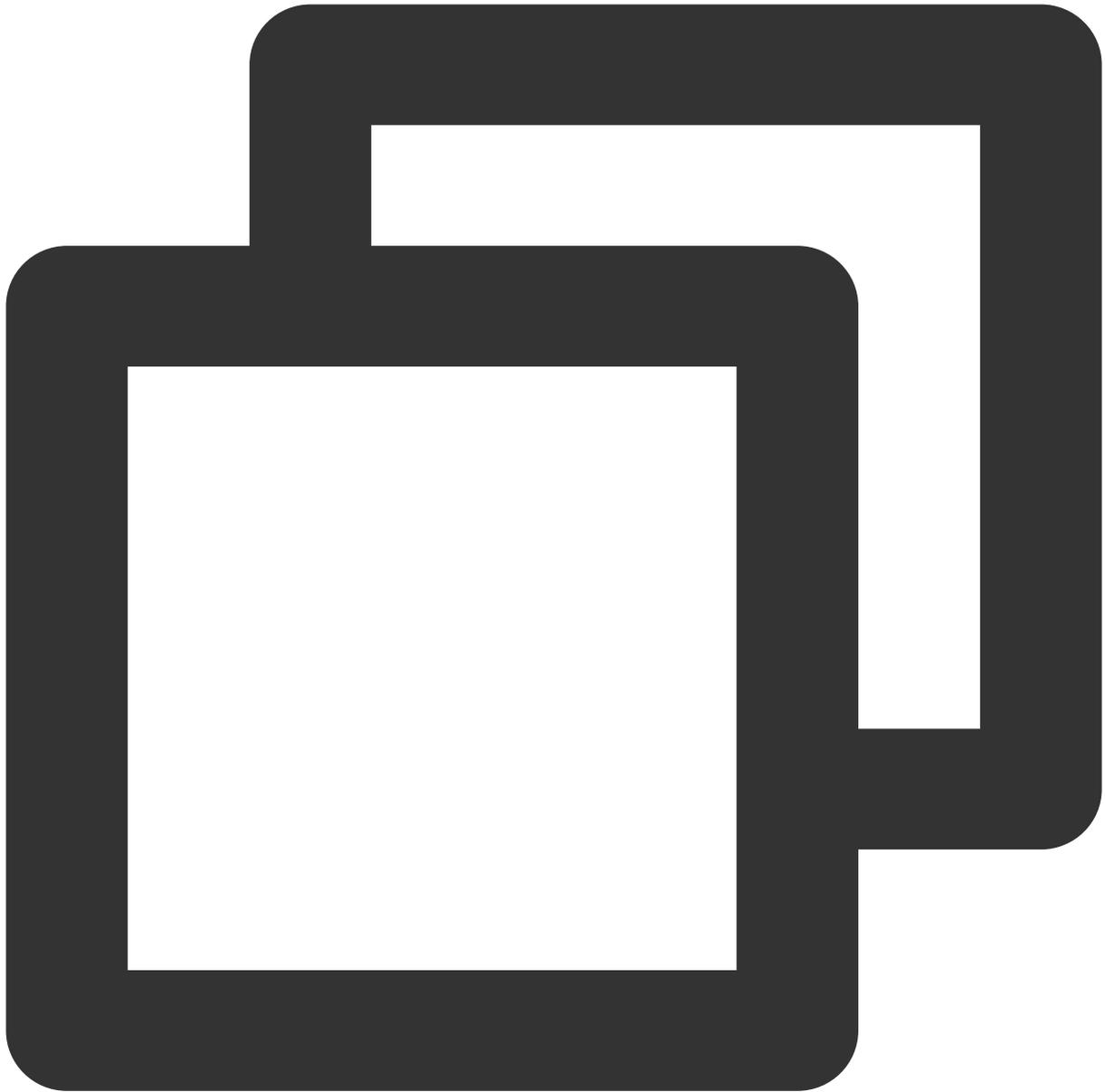
```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy examplebucket2-1250000000.
```

修改存储类型，将文件类型改为归档存储，并重命名为 photo.jpg



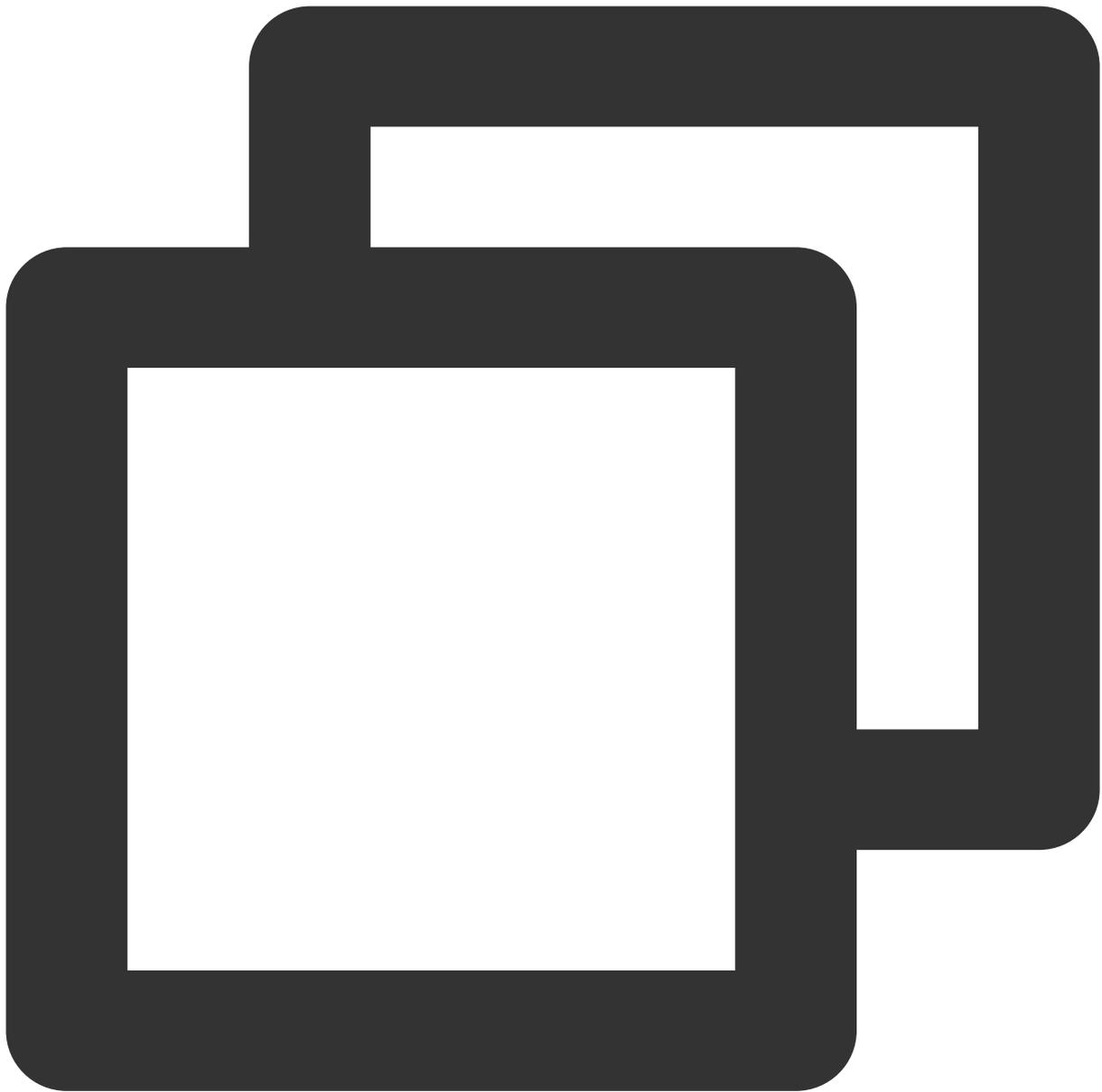
```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy examplebucket2-1250000000.
```

复制文件夹命令格式



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

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



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

说明：

请将 "<>" 中的参数替换为您需要复制的 COS 上文件的路径（sourcepath），和您需要复制到 COS 上文件的路径（cospath）。

sourcepath 的格式为：`<BucketName-APPID>.cos.<region>.myqcloud.com/<cospath>`。

使用 -d 参数可以设置 `x-cos-metadata-directive` 参数，可选值为 Copy 和 Replaced，默认为 Copy。

使用 -H 参数设置 HTTP header 时，请务必保证格式为 JSON，示例：`coscmd copy -H -d Replaced '{"x-cos-storage-class': 'Archive', 'Content-Language': 'zh-CN'}"` `<localpath>` `<cospath>`。更

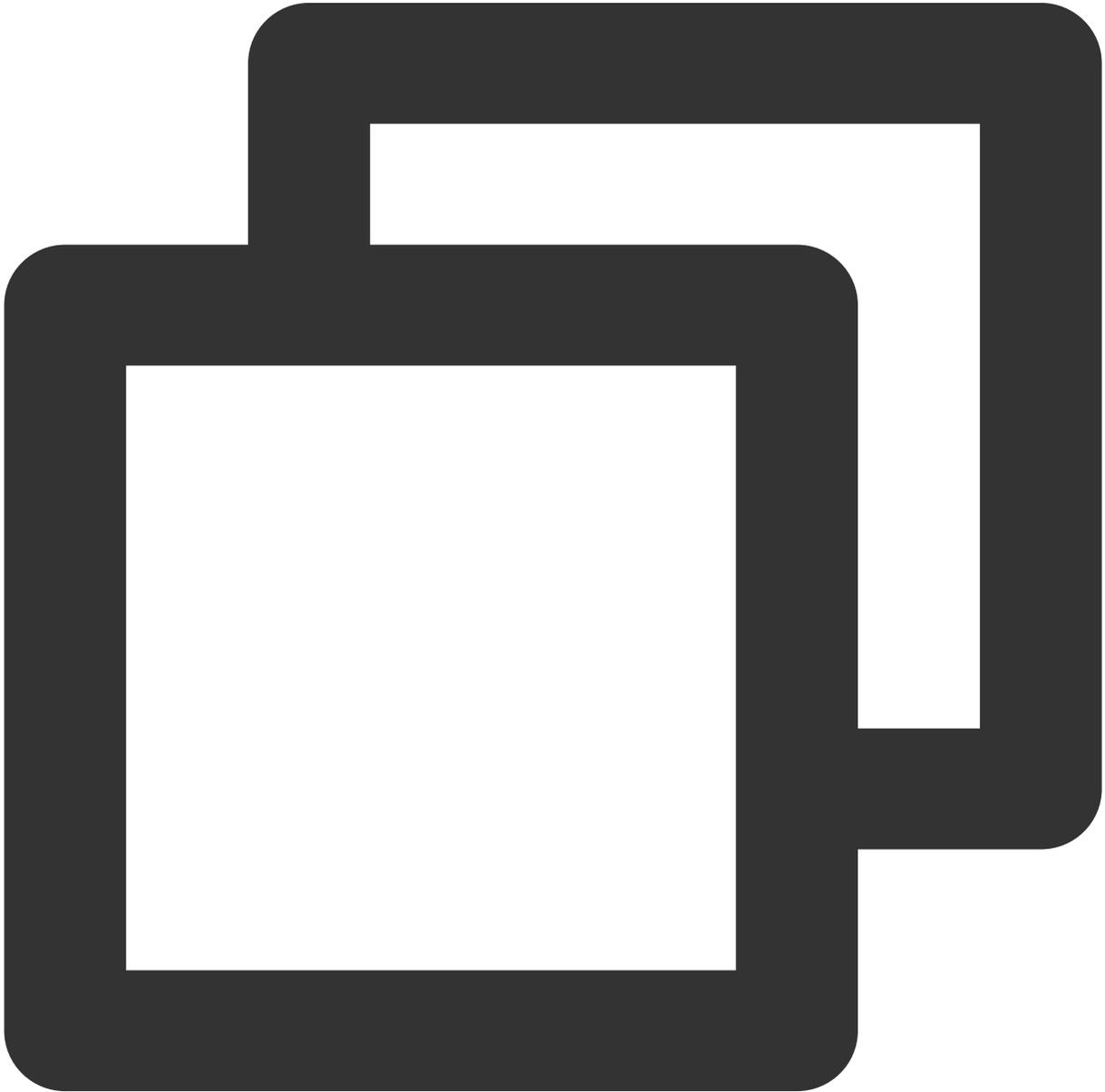
多头部请参见 [PUT Object - Copy](#) 文档。

移动文件或文件夹

注意：

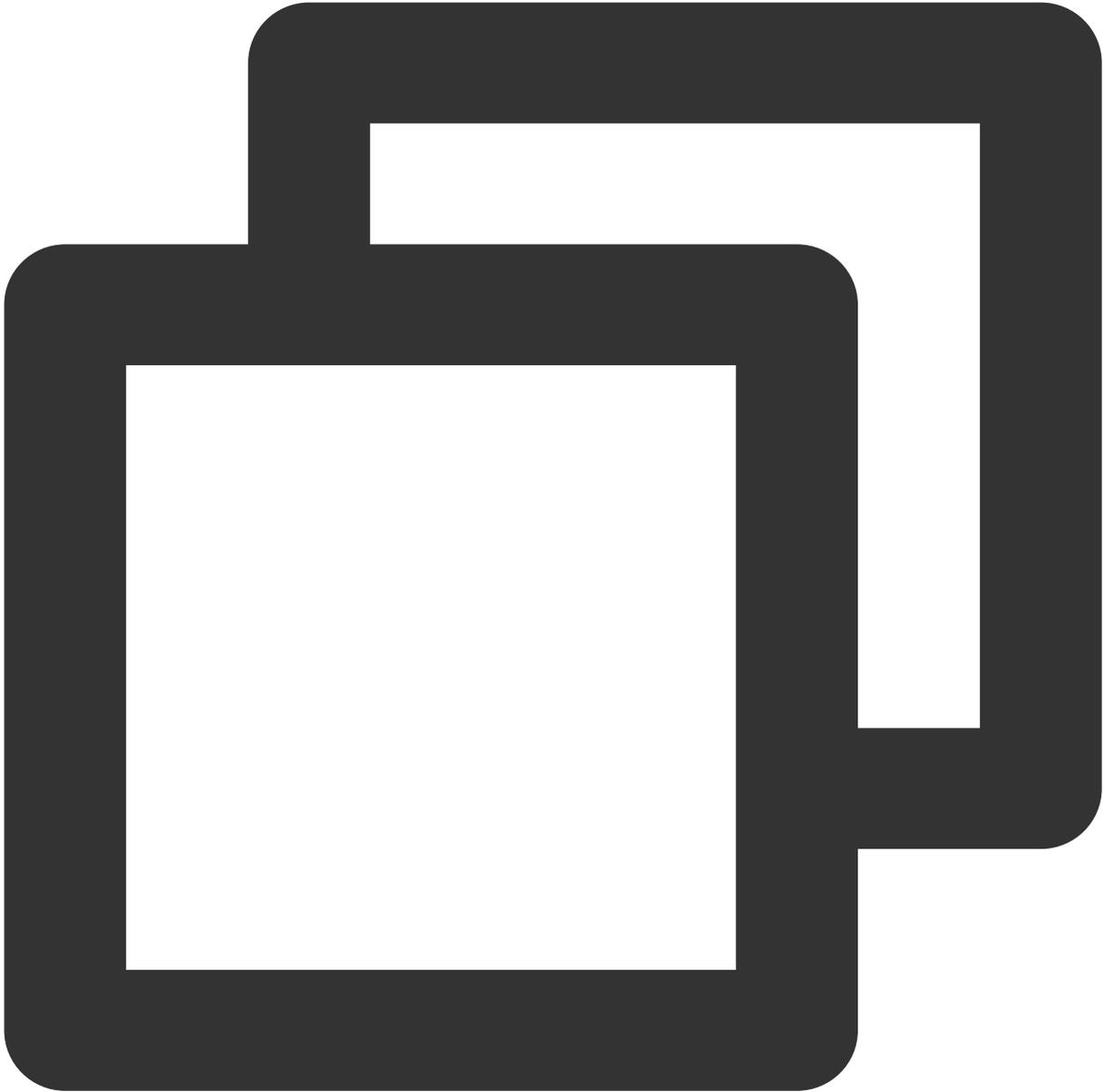
移动命令的 `<sourcepath>` 和 `<cospath>` 不能相同，否则会导致文件被删除。原因在于 `move` 命令会先复制，再删除，`<sourcepath>` 路径下的文件最终会被删除。

移动文件命令格式



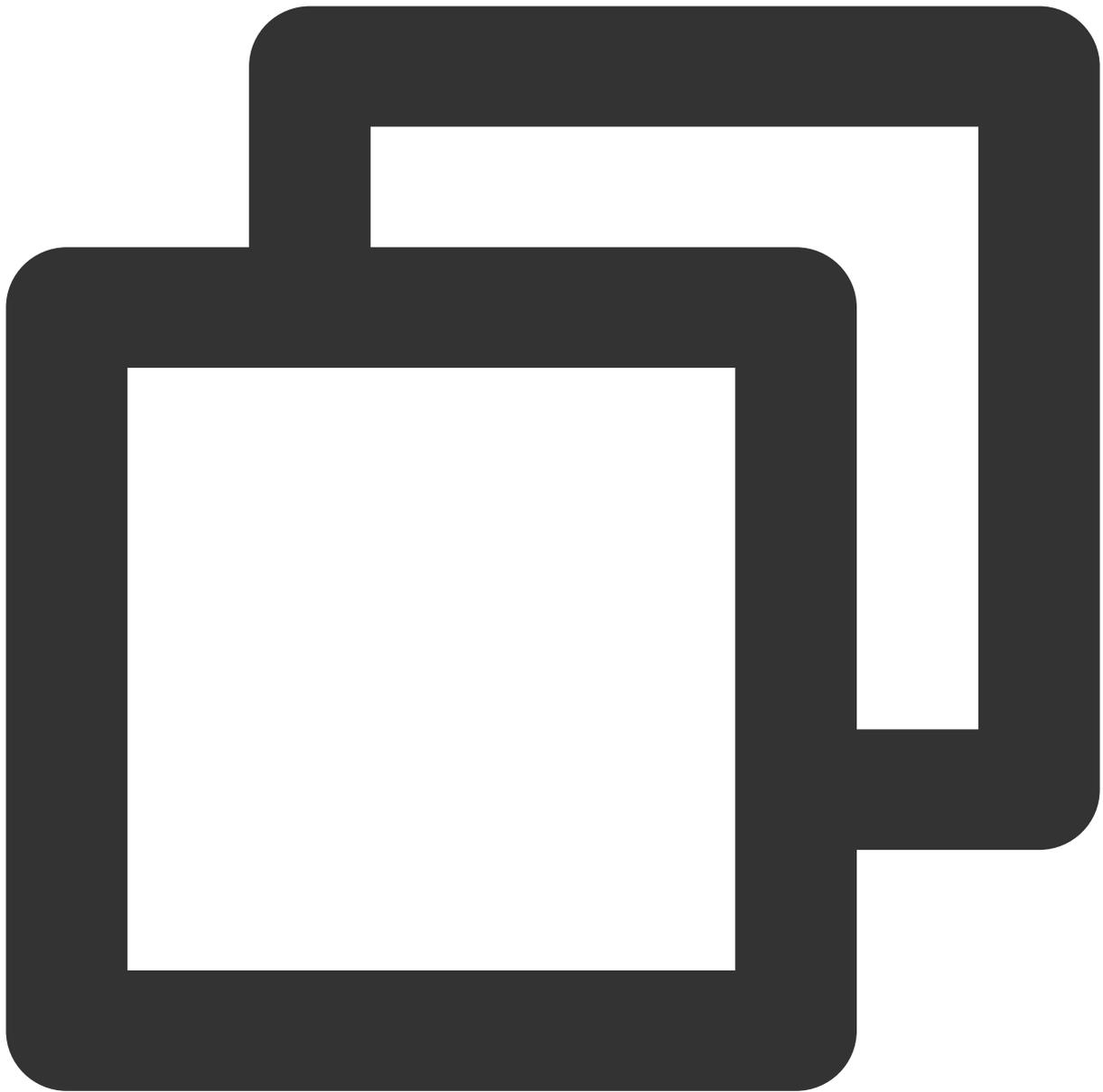
```
coscmd move <sourcepath> <cospath>
```

操作示例 - 在同一个存储桶内进行移动，将 examplebucket-1250000000 存储桶下的 picture.jpg 移动到 doc 文件夹下



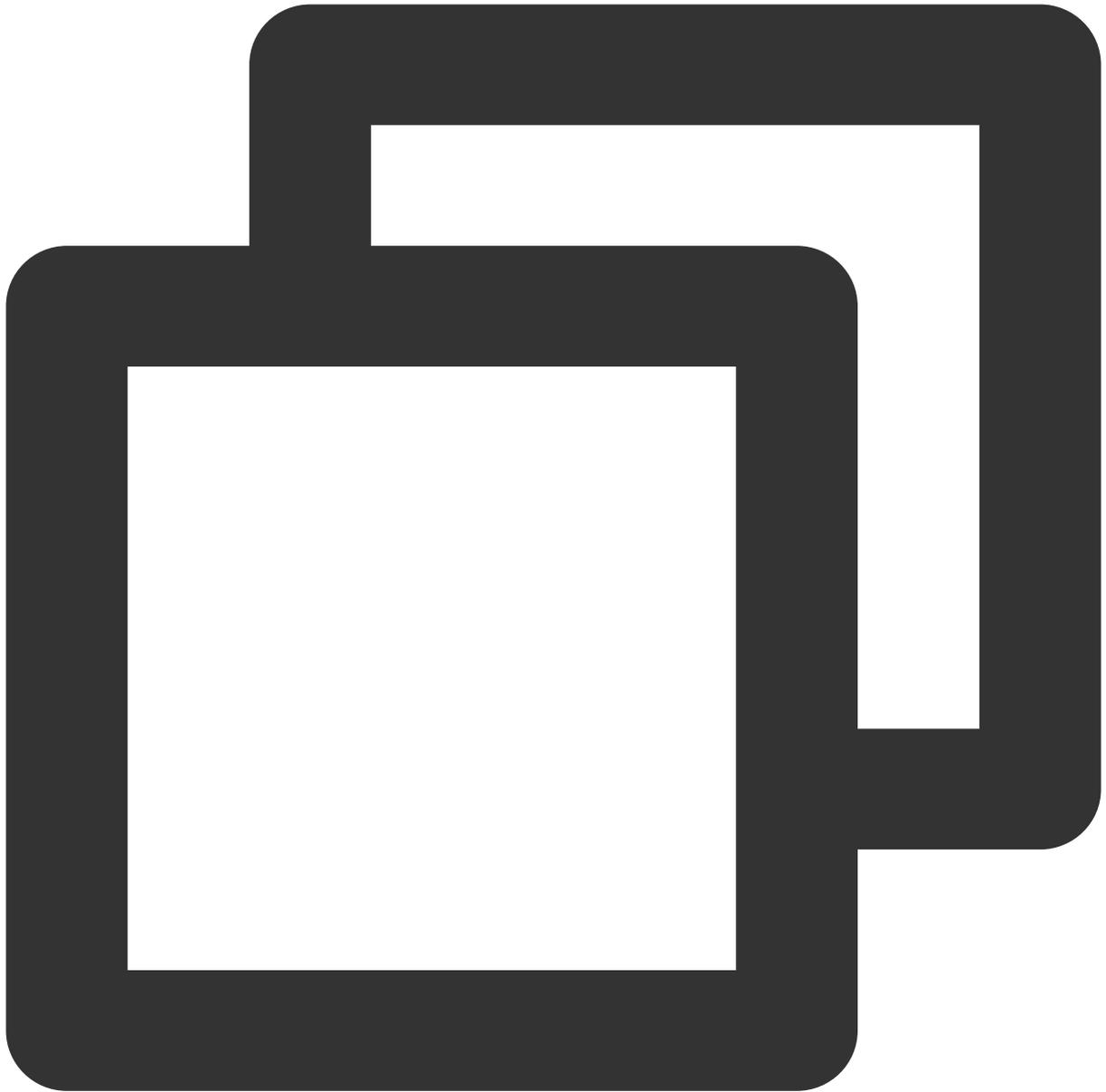
```
coscmd -b examplebucket-1250000000 -r ap-chengdu move examplebucket-1250000000.ap-c
```

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



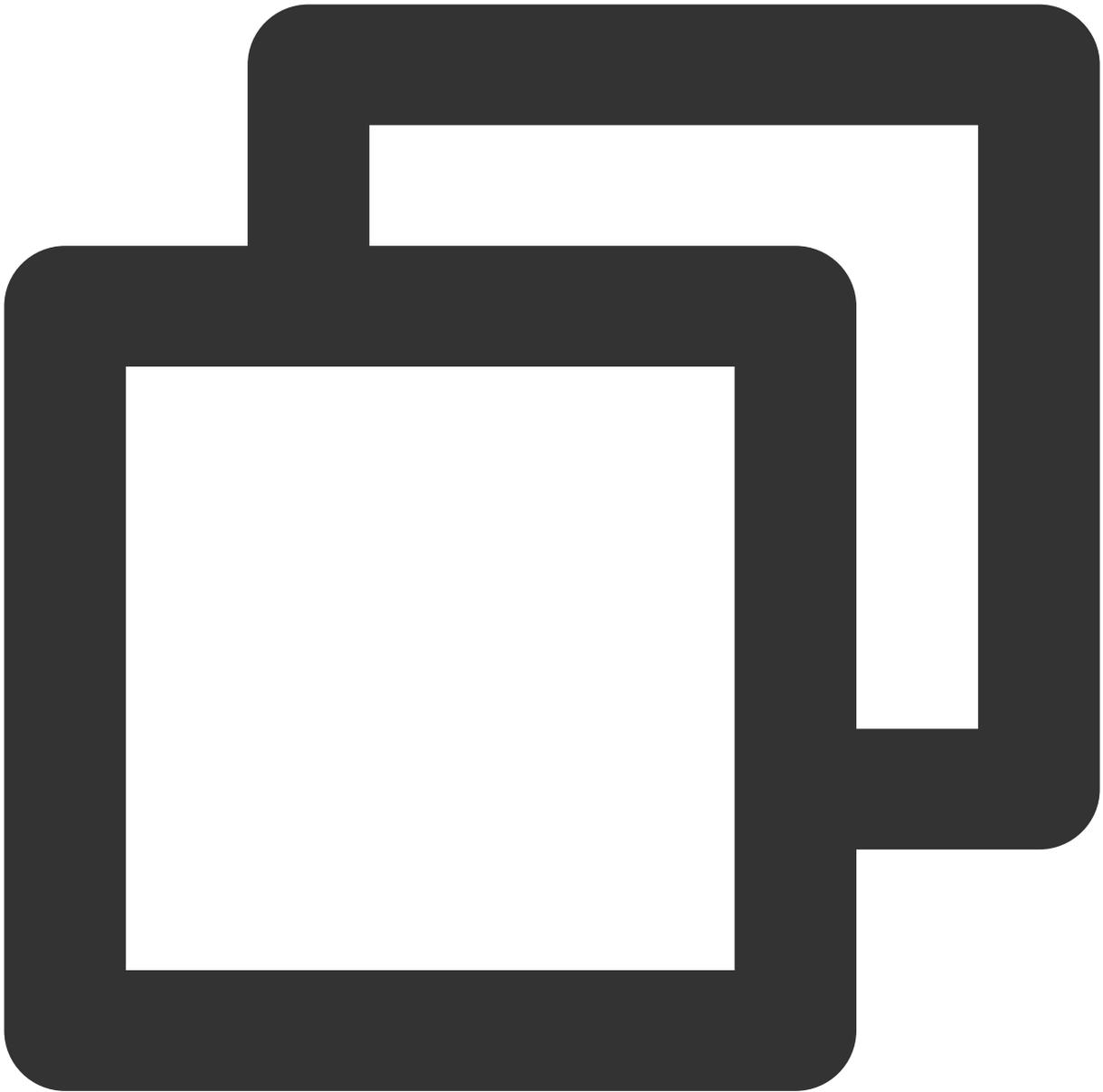
```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou move examplebucket2-1250000000.
```

操作示例 - 修改存储类型，将文件类型改为低频存储



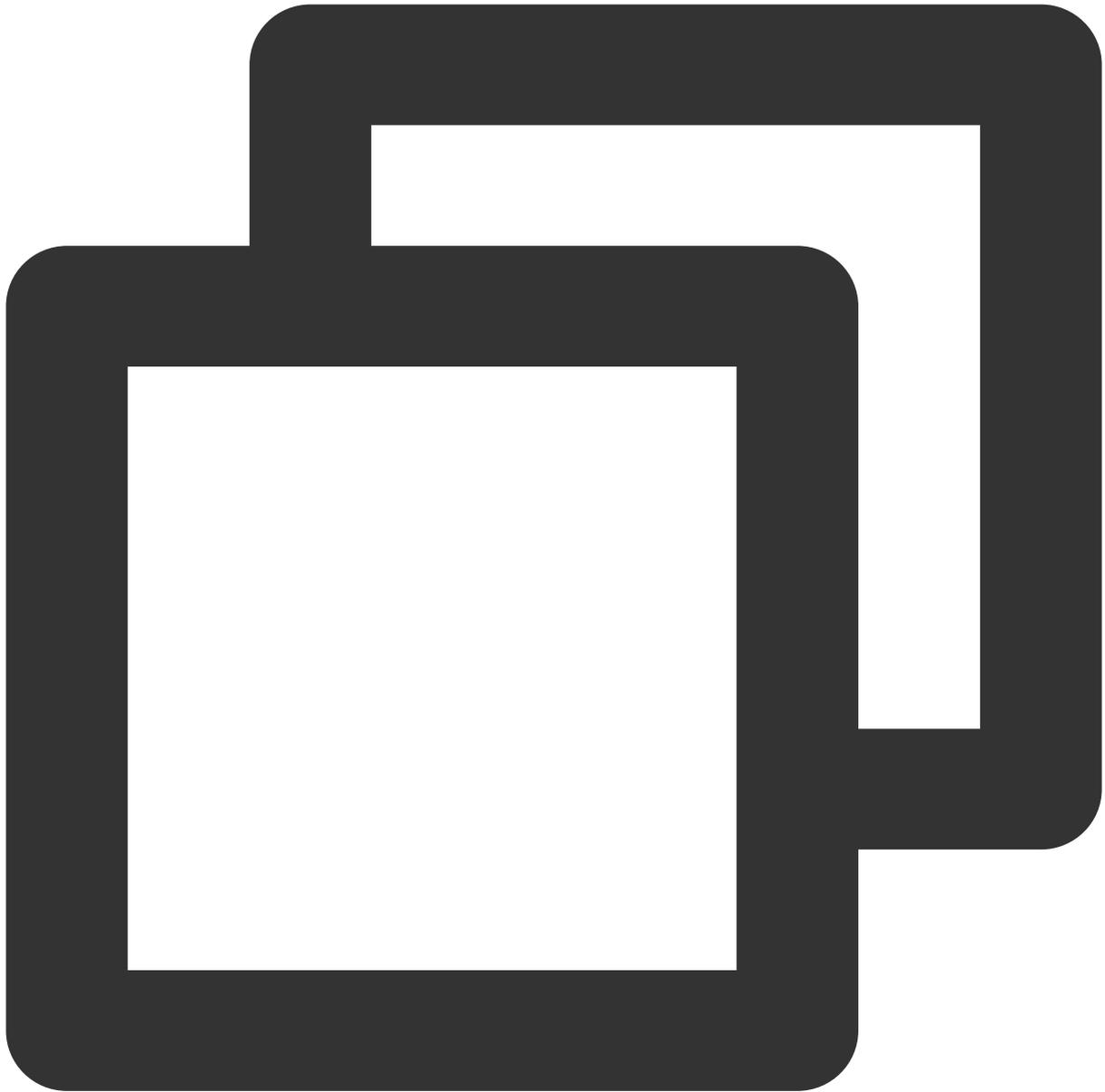
```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou move examplebucket2-1250000000.
```

操作示例 - 修改存储类型，将文件类型改为归档存储



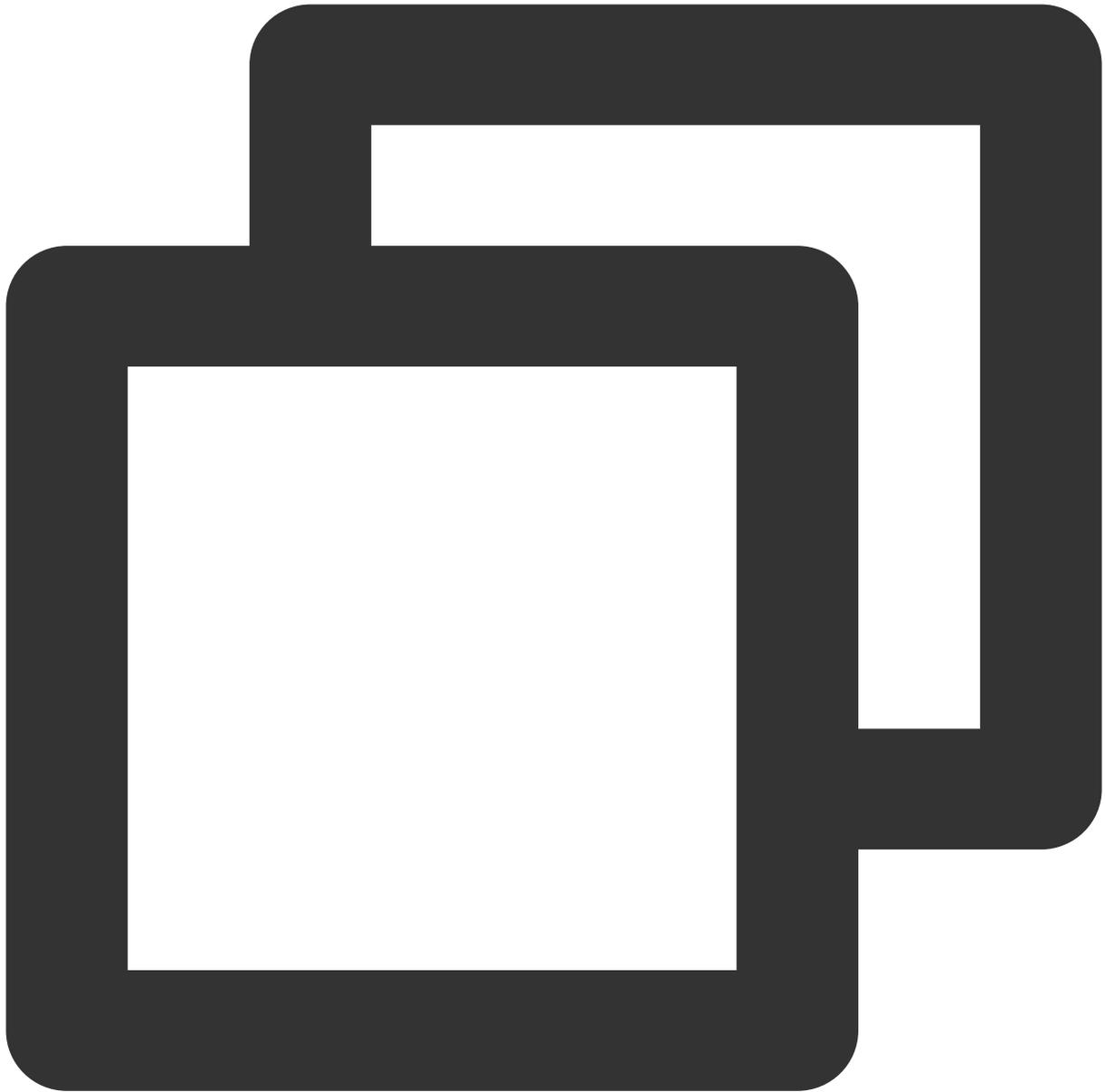
```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou move examplebucket2-1250000000.
```

移动文件夹命令格式



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

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



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

说明：

请将 "<>" 中的参数替换为您需要移动的 COS 上文件的路径（sourcepath），和您需要移动到 COS 上文件的路径（cospath）。

sourcepath 的格式为：`<BucketName-APPID>.cos.<region>.myqcloud.com/<cospath>`。

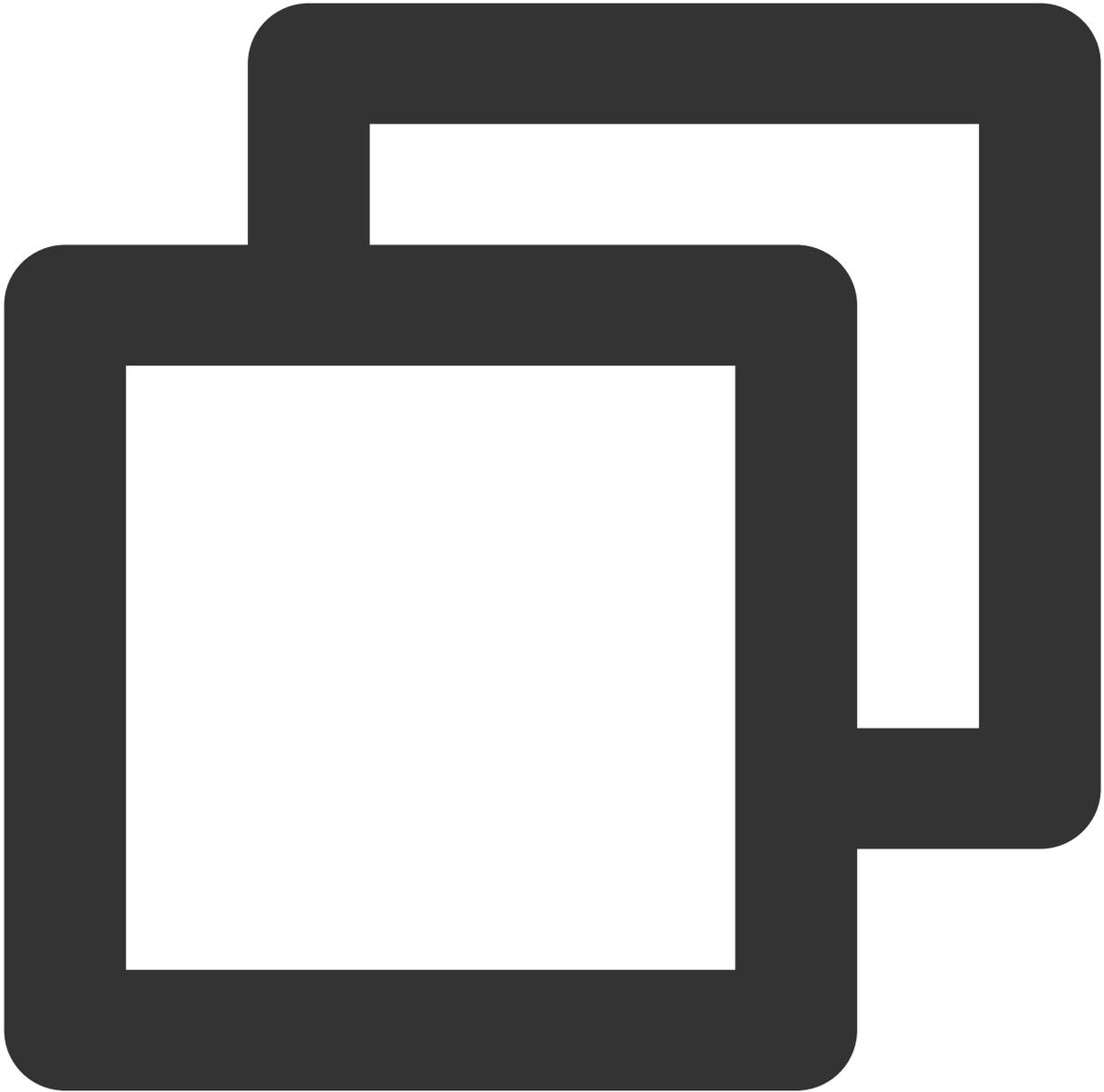
使用 -d 参数可以设置 `x-cos-metadata-directive` 参数，可选值为 Copy 和 Replaced，默认为 Copy。

使用 -H 参数设置 HTTP header 时，请务必保证格式为 JSON，示例：`coscmd move -H -d Replaced '{"x-cos-storage-class': 'Archive', 'Content-Language': 'zh-CN'}"` `<localpath>` `<cospath>`。更

多头部请参见 [PUT Object - copy](#) 文档。

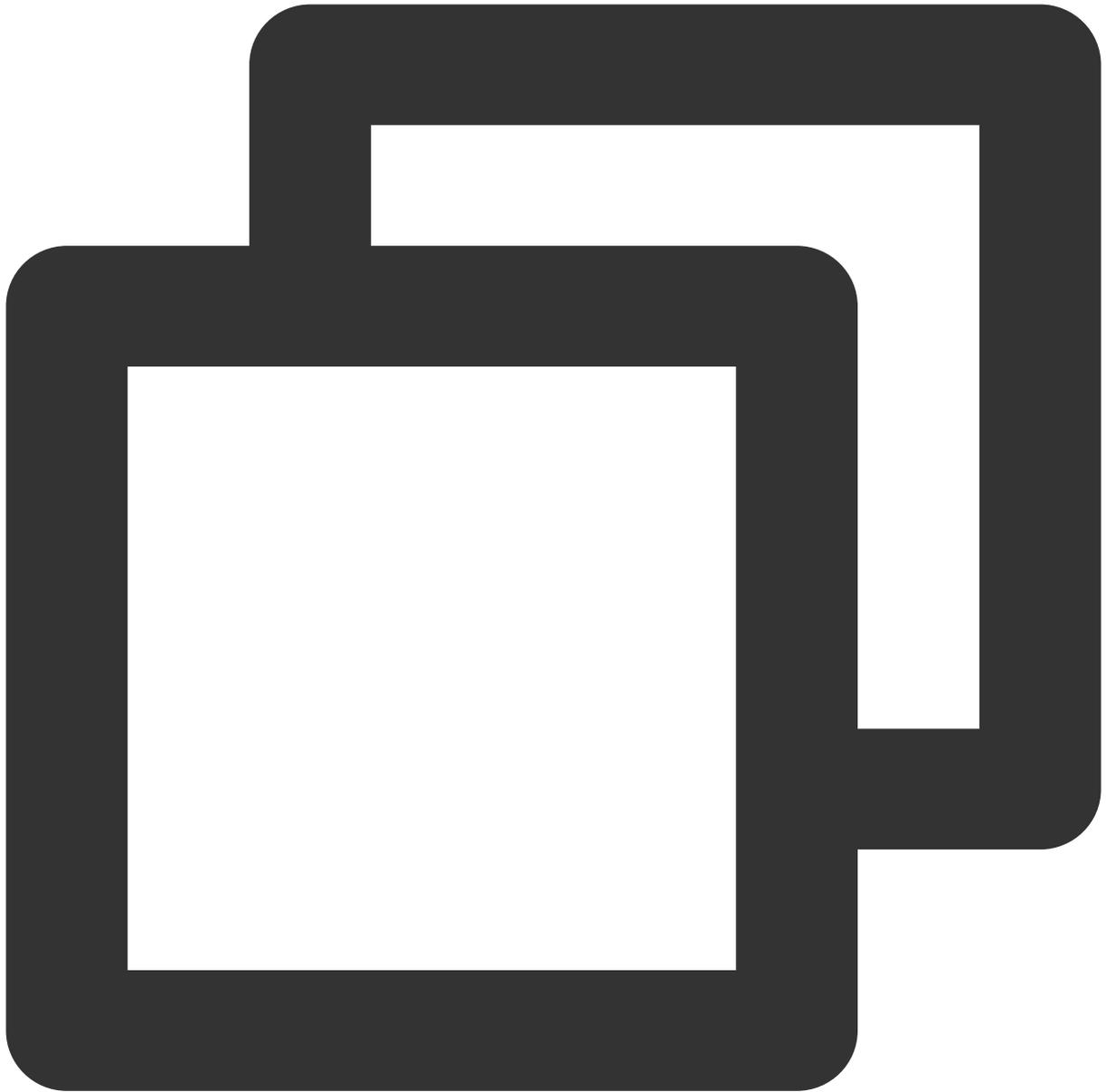
设置对象访问权限

命令格式



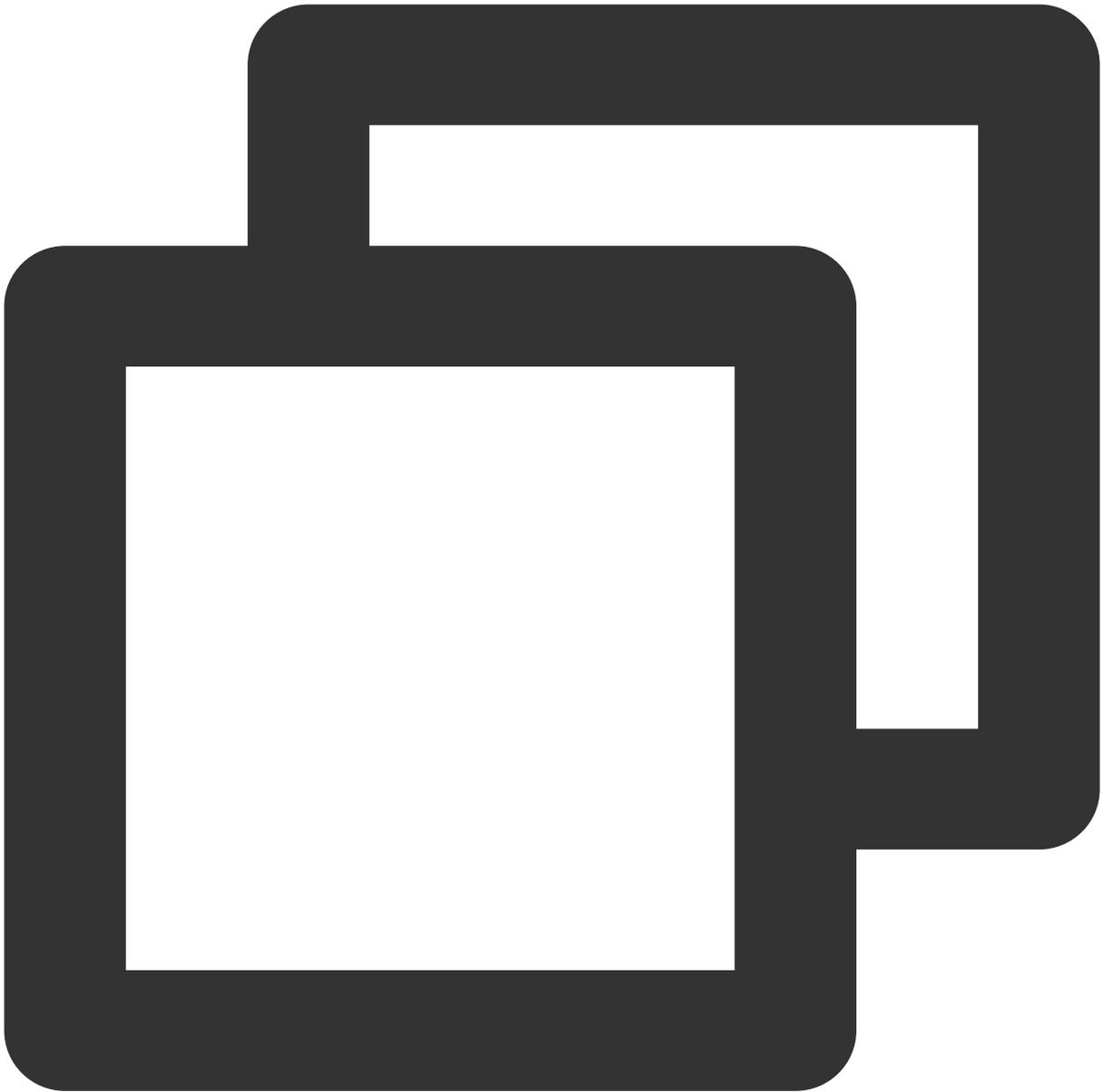
```
coscmd putobjectacl --grant-<permissions> <UIN> <cospath>
```

操作示例 - 授予账号 100000000001 拥有 picture.jpg 的读取权限



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

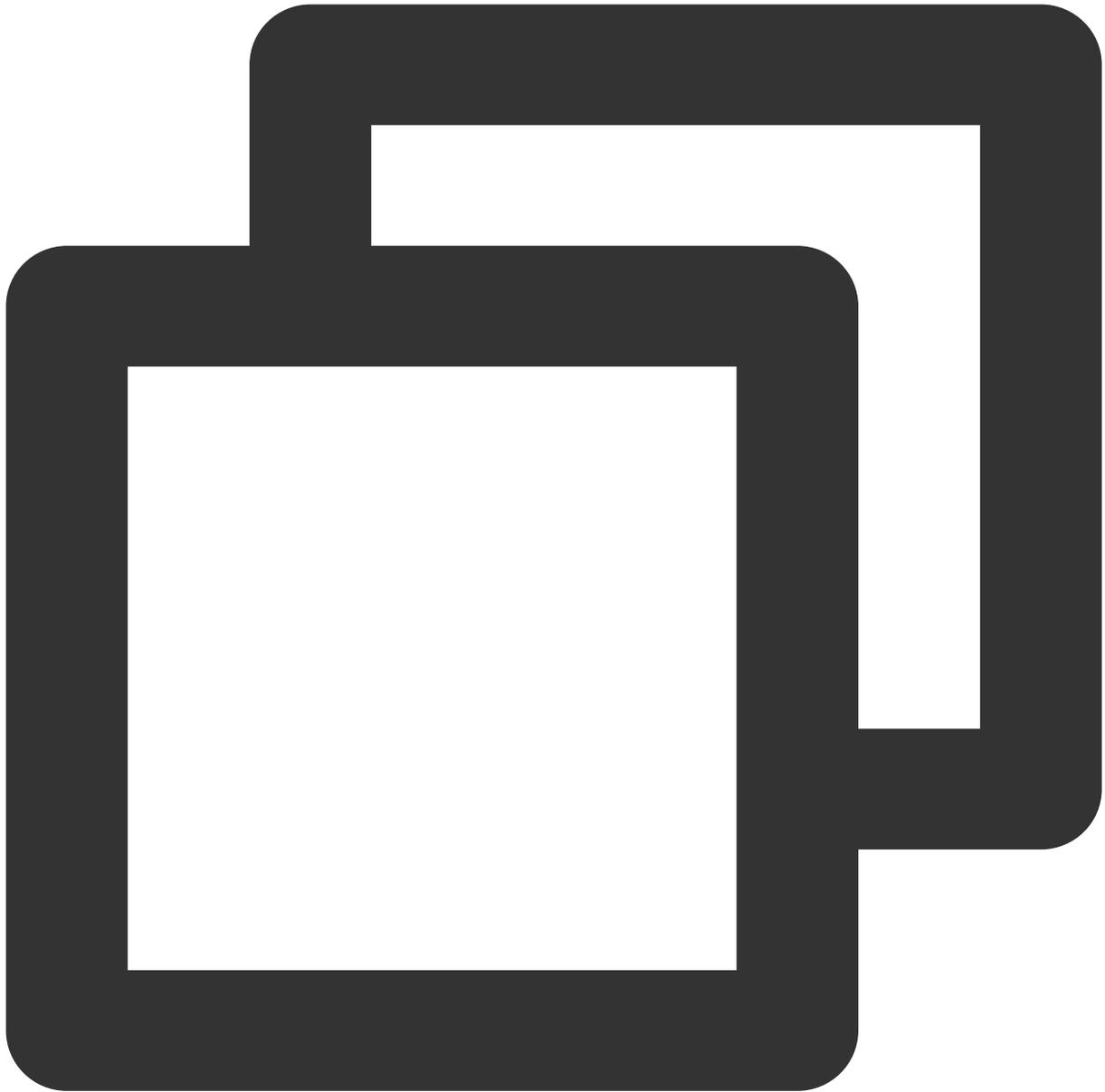
操作示例 - 查询文件的访问权限



```
coscmd getobjectacl picture.jpg
```

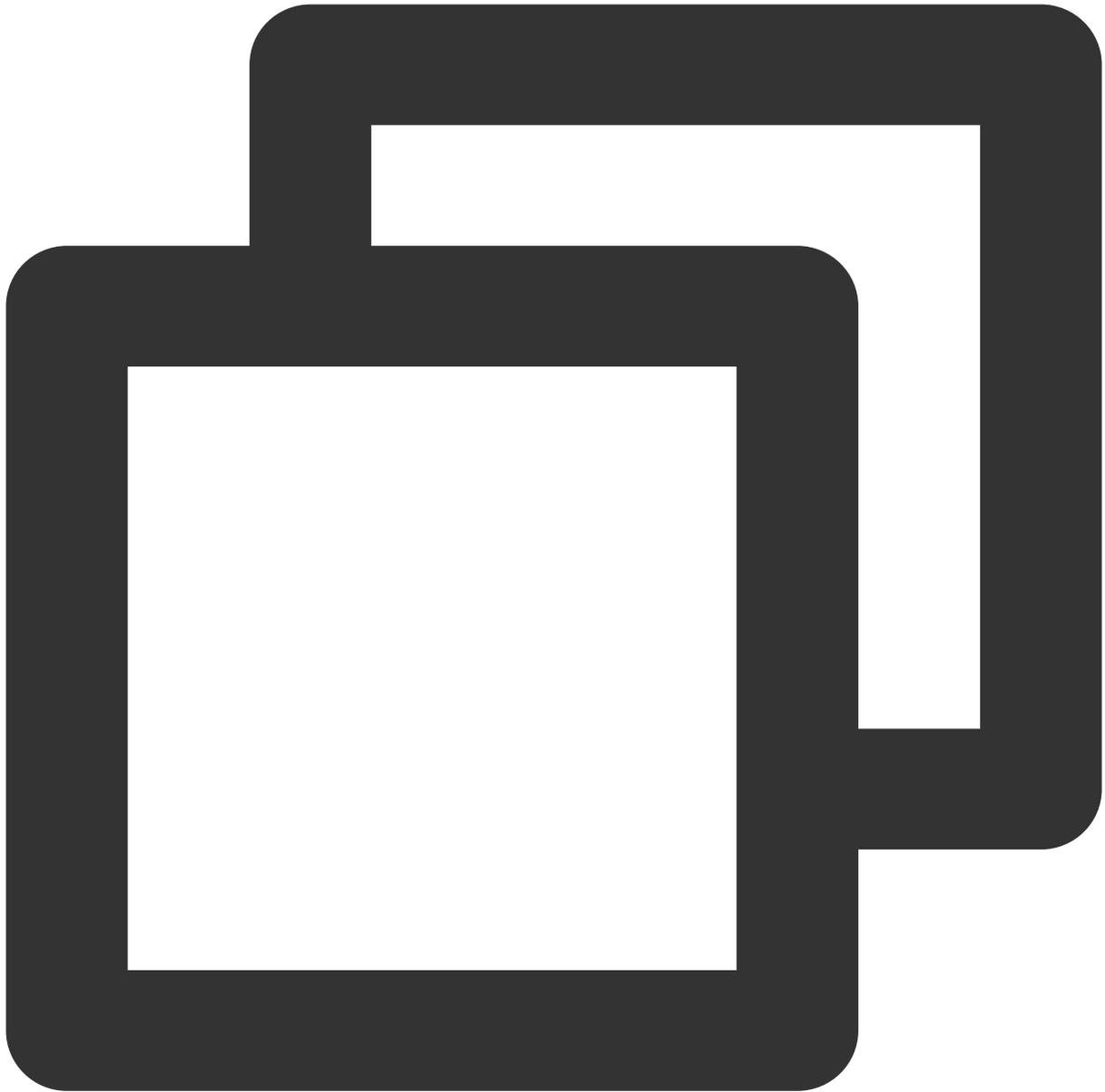
开启/暂停版本控制

命令格式



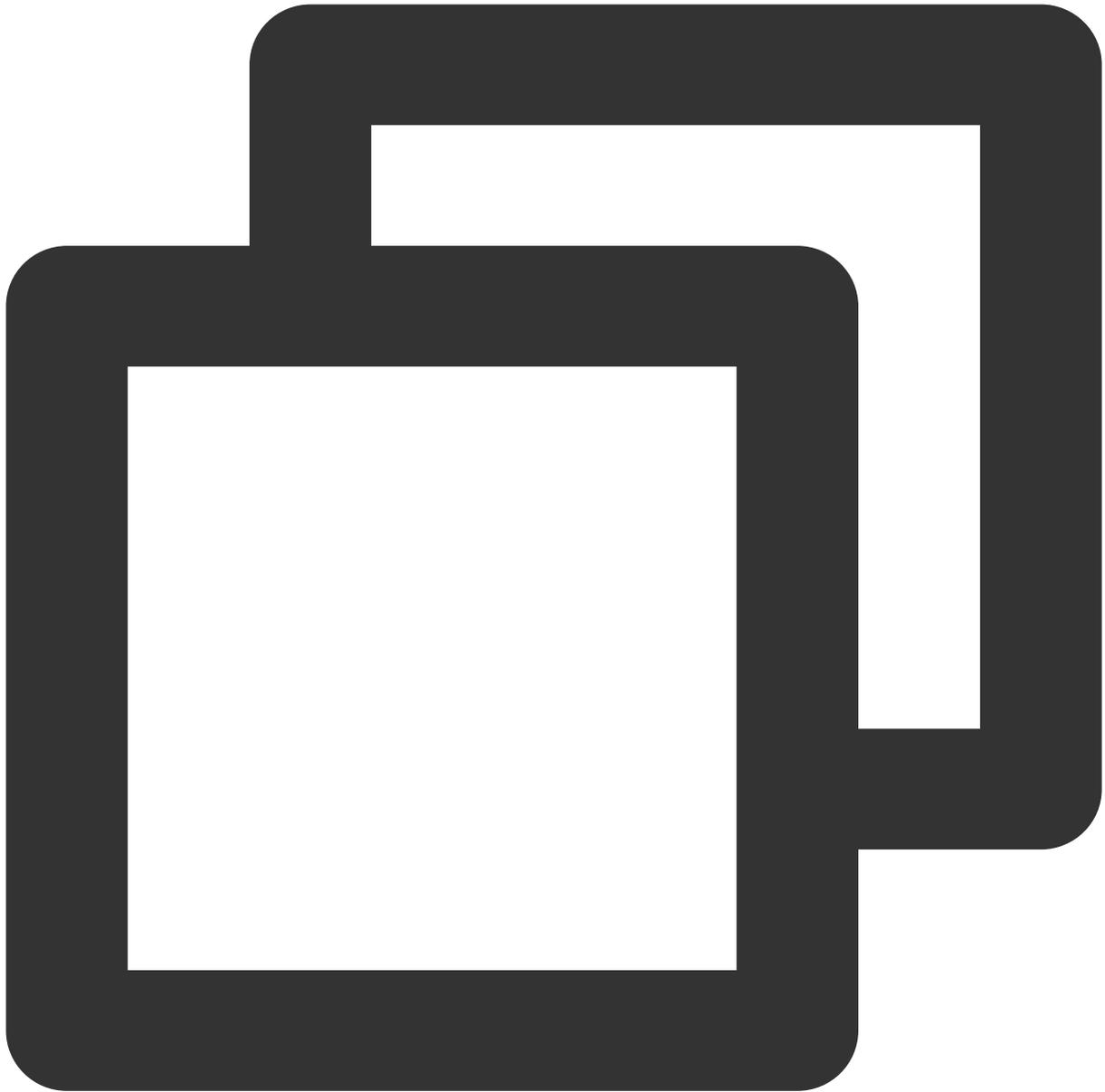
```
coscmd putbucketversioning <status>
```

操作示例 - 开启版本控制



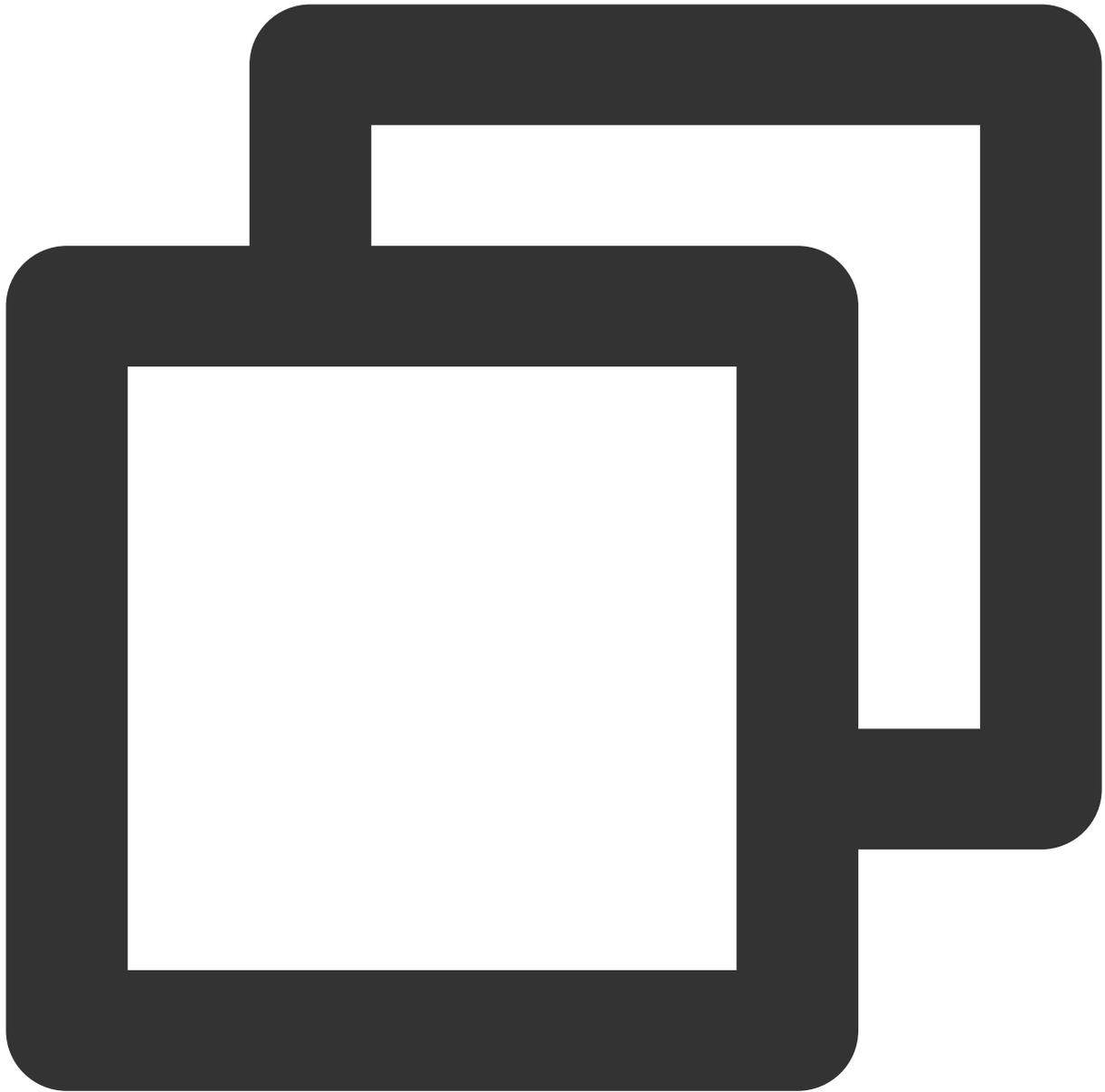
```
coscmd putbucketversioning Enabled
```

操作示例 - 暂停版本控制



```
coscmd putbucketversioning Suspended
```

操作示例 - 查询版本控制



```
coscmd getbucketversioning
```

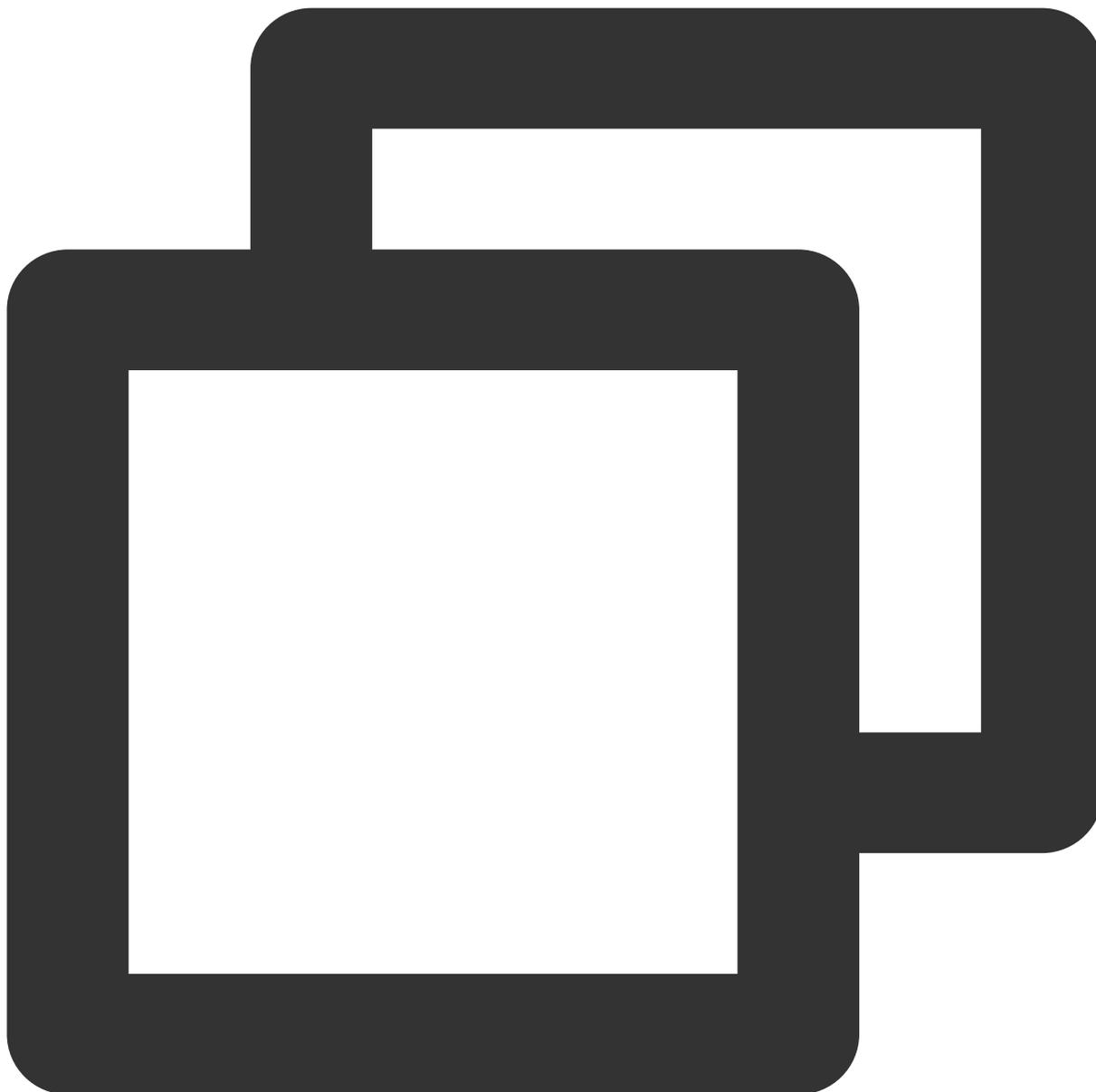
注意

请将 "<>" 中的参数替换为您需要版本控制状态 (*status*)。

一旦您对存储桶启用了版本控制，它将无法返回到未启用版本控制状态（初始状态）。但是，您可以对该存储桶暂停版本控制，这样后续上传的对象将不会产生多个版本。

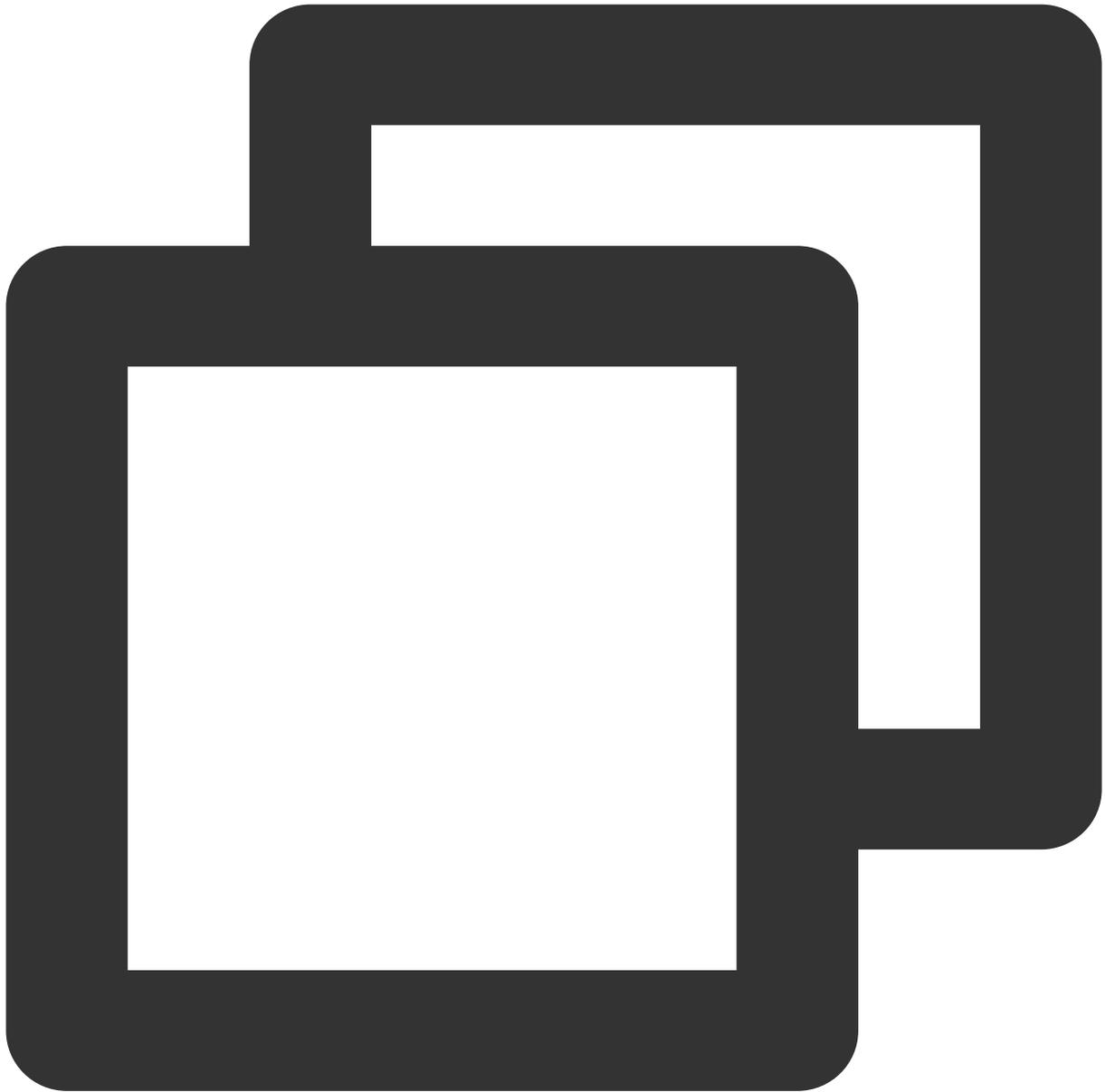
恢复归档文件

恢复归档文件命令格式



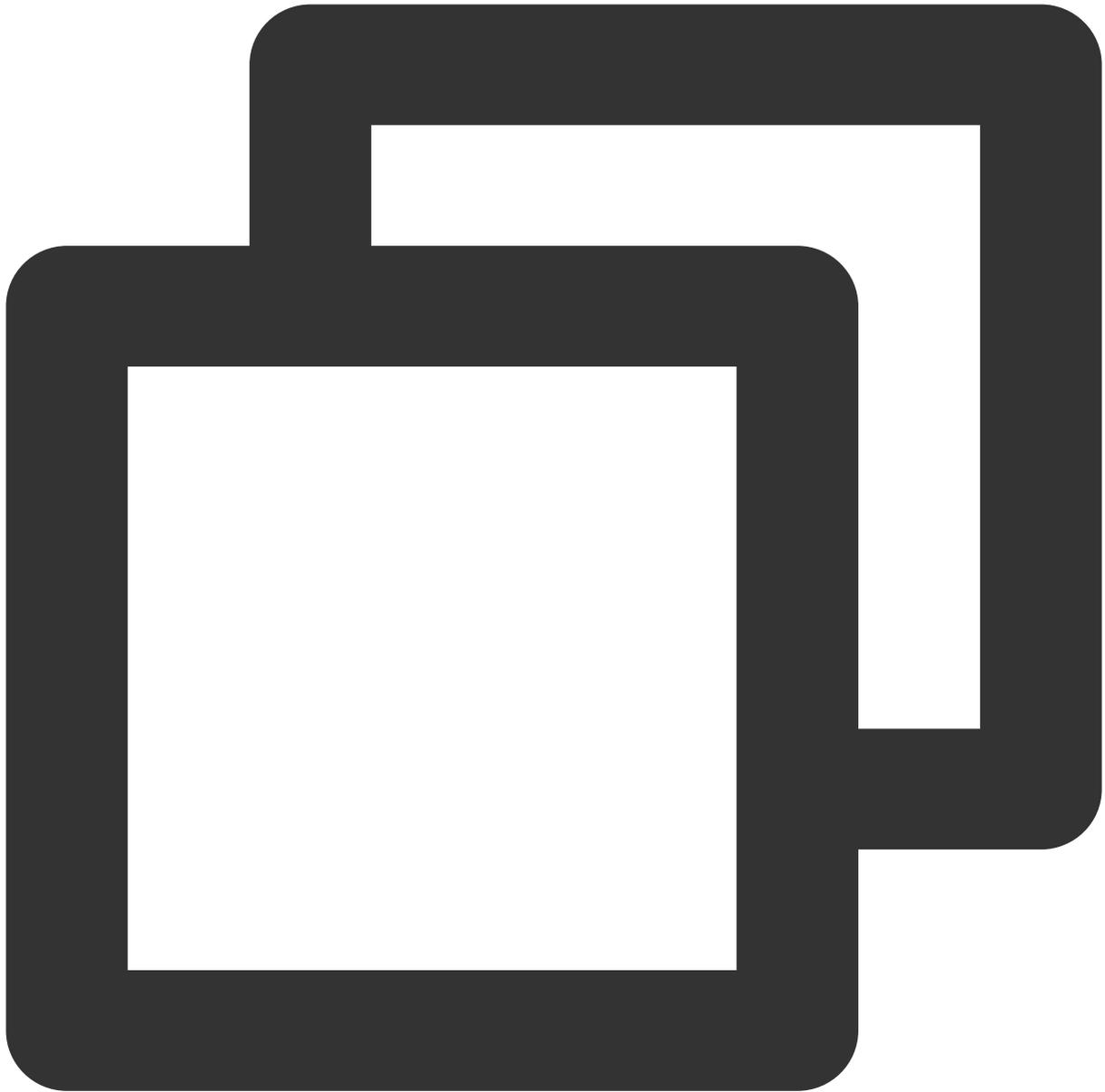
```
coscmd restore <cospath>
```

操作示例 - 快速取回模式回热 picture.jpg ，有效期3天



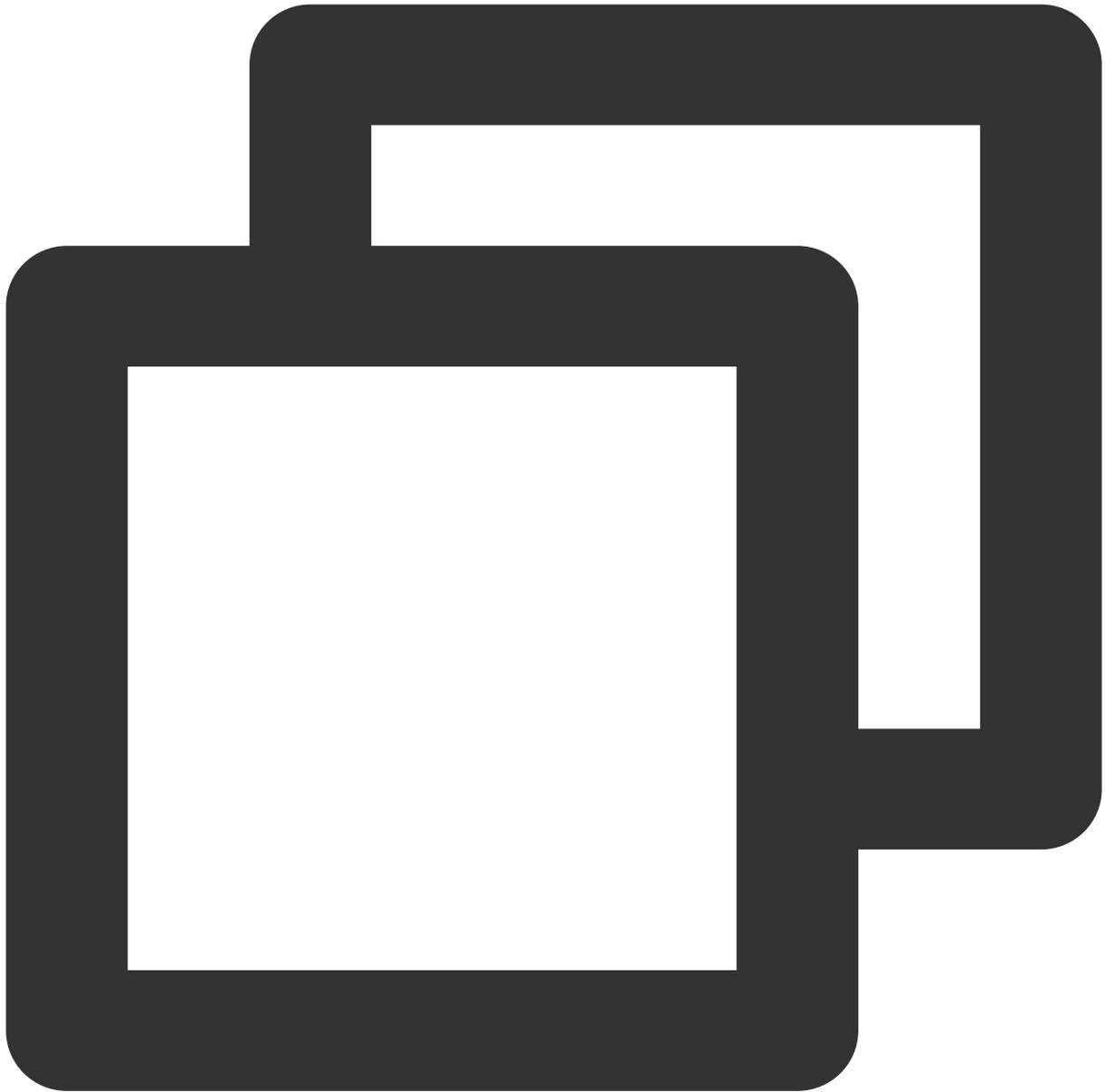
```
coscmd restore -d 3 -t Expedited picture.jpg
```

批量恢复归档文件命令格式



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

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



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

说明：

请将 "<>" 中的参数替换为您需要查询文件列表的 COS 上文件的路径（`cospath`）。

使用 `-d <day>` 设置临时副本的过期时间，默认值：7。

使用 `-t <tier>` 指定恢复模式，枚举值：`Expedited`（快速取回模式），`Standard`（标准取回模式），`Bulk`（批量取回模式），默认值：`Standard`。

常见问题

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

COS Migration 工具

最近更新时间：2024-04-30 16:43:10

功能说明

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

断点续传：工具支持上传时断点续传。对于一些大文件，如果中途退出或者因为服务故障，可重新运行工具，会对未上传完成的文件进行续传。

分块上传：将对象按照分块的方式上传到 COS。

并行上传：支持多个对象同时上传。

注意：

COS Migration 的编码格式只支持 UTF-8 格式。

使用该工具上传同名文件，默认会覆盖较旧的同名文件，需要额外设置以跳过同名文件。

除本地数据迁移之外的场景请优先使用迁移服务平台。

COS Migration 是用来做**一次性**迁移服务的，不适合于持续同步的场景。例如本地每天新增文件，需要持续同步至 COS 中，COS Migration 为了避免重复迁移任务，会保存迁移成功的记录，持续同步后，扫描记录时间会持续增大。此种场景建议使用 [文件同步](#)。

使用环境

系统环境

Windows 和 Linux 系统。

软件依赖

JDK 1.8 X64或以上，有关 JDK 的安装与配置请参见 [Java 安装与配置](#)。

Linux 环境需要 IFUNC 支持，确保环境 binutils 版本大于 2.20。

使用方法

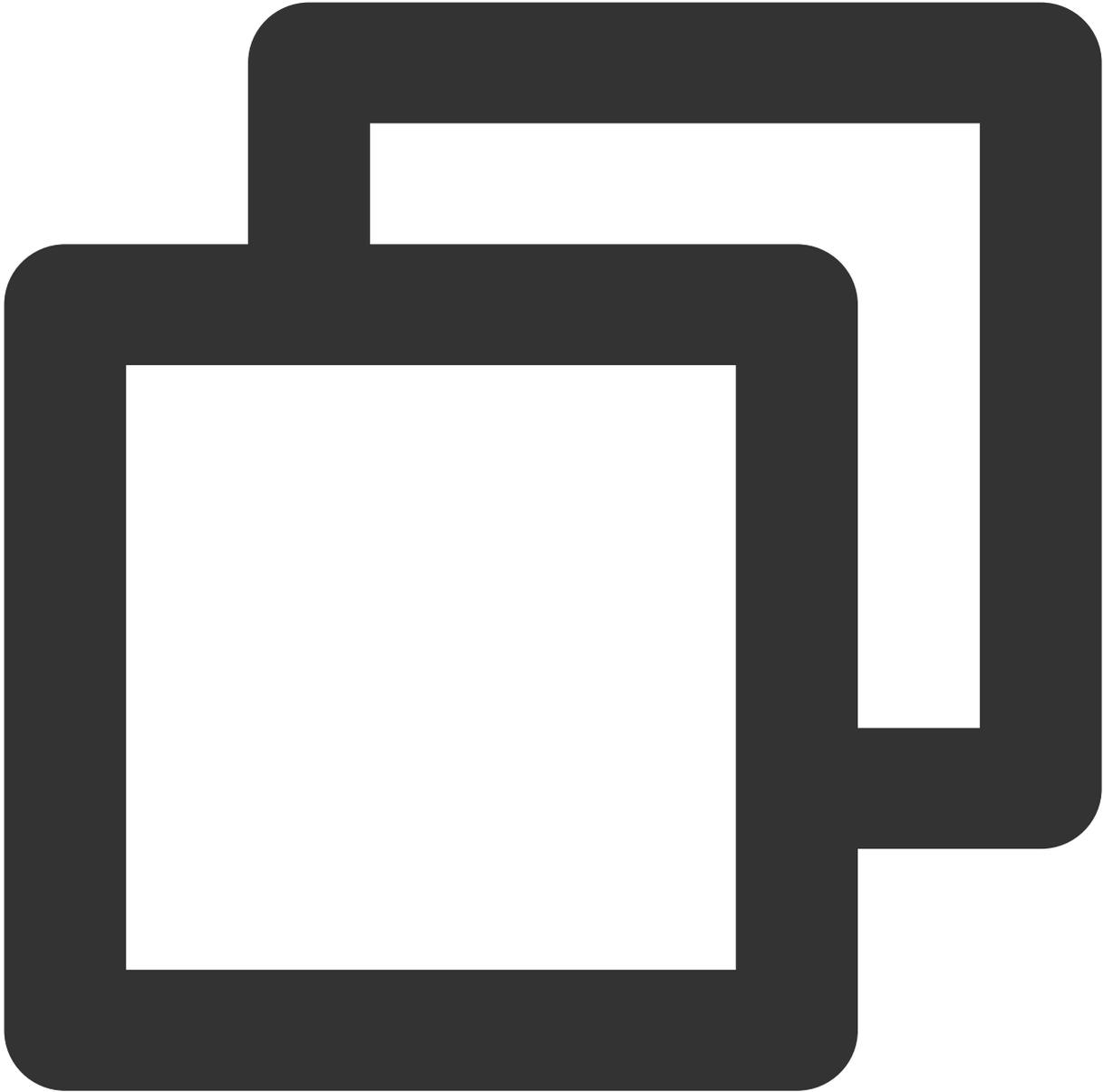
1. 获取工具

前往下载 [COS Migration 工具](#)。

2. 解压缩工具包

Windows

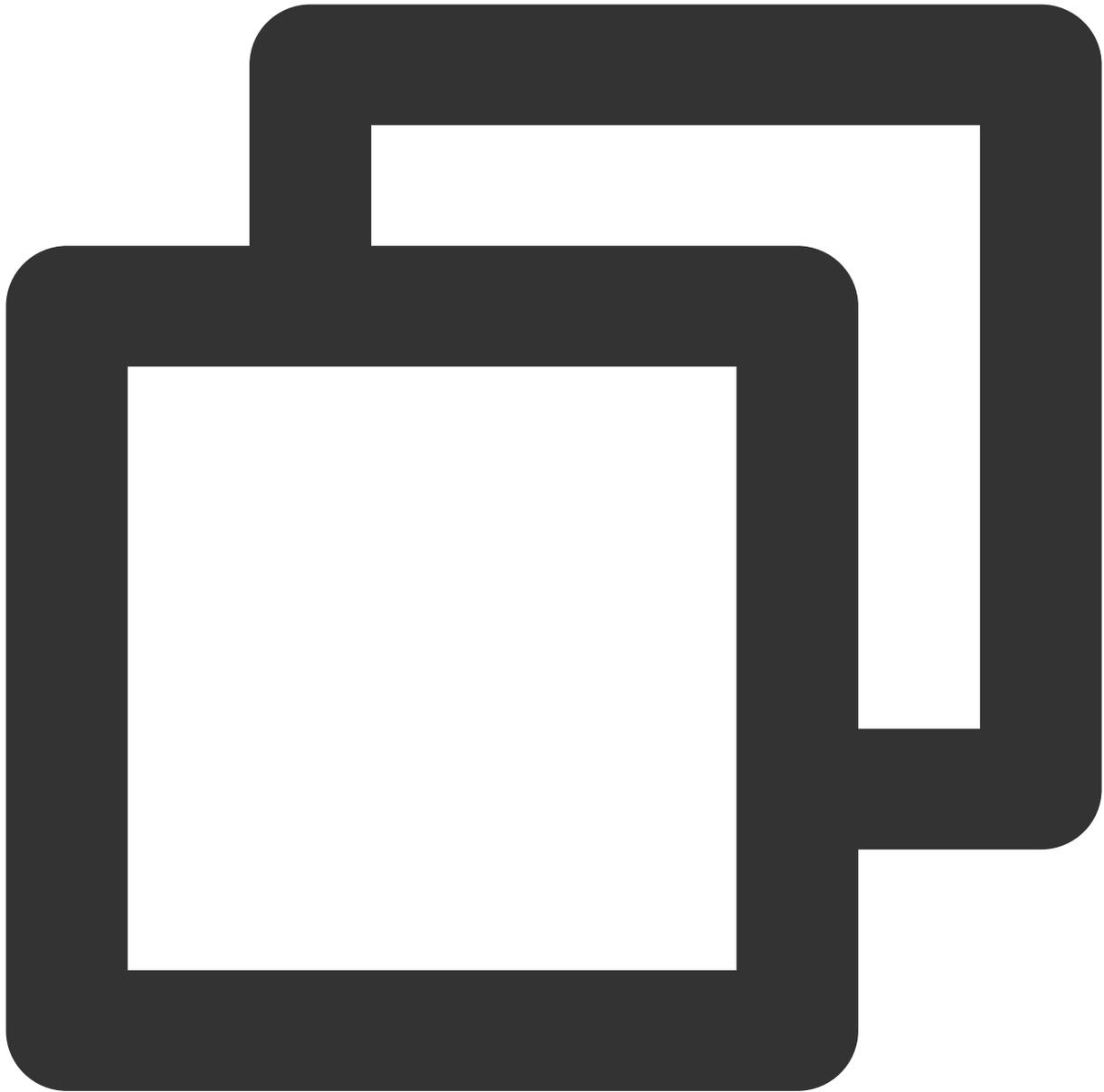
解压并保存到某个目录，例如：



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

Linux

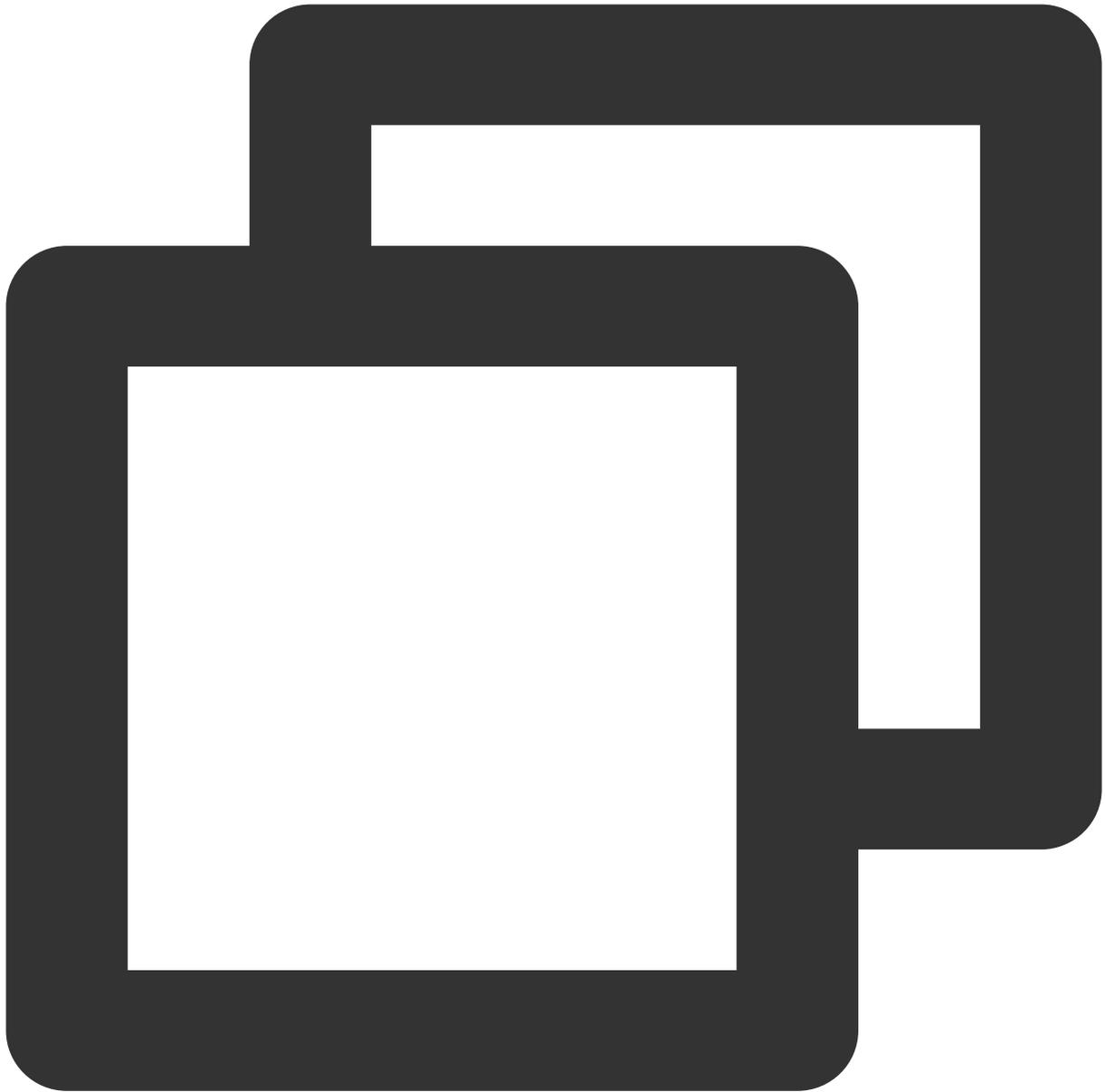
解压并保存到某个目录：



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

迁移工具结构

正确解压后的 COS Migration 工具目录结构如下所示：



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

```
|—pom.xml #项目配置文件
|—README #说明文档
|—start_migrate.sh #Linux 下迁移启动脚本
|—start_migrate.bat #Windows 下迁移启动脚本
```

说明

db 目录主要记录工具迁移成功的文件标识，每次迁移任务会优先对比 db 中的记录，若当前文件标识已被记录，则会跳过当前文件，否则进行文件迁移。

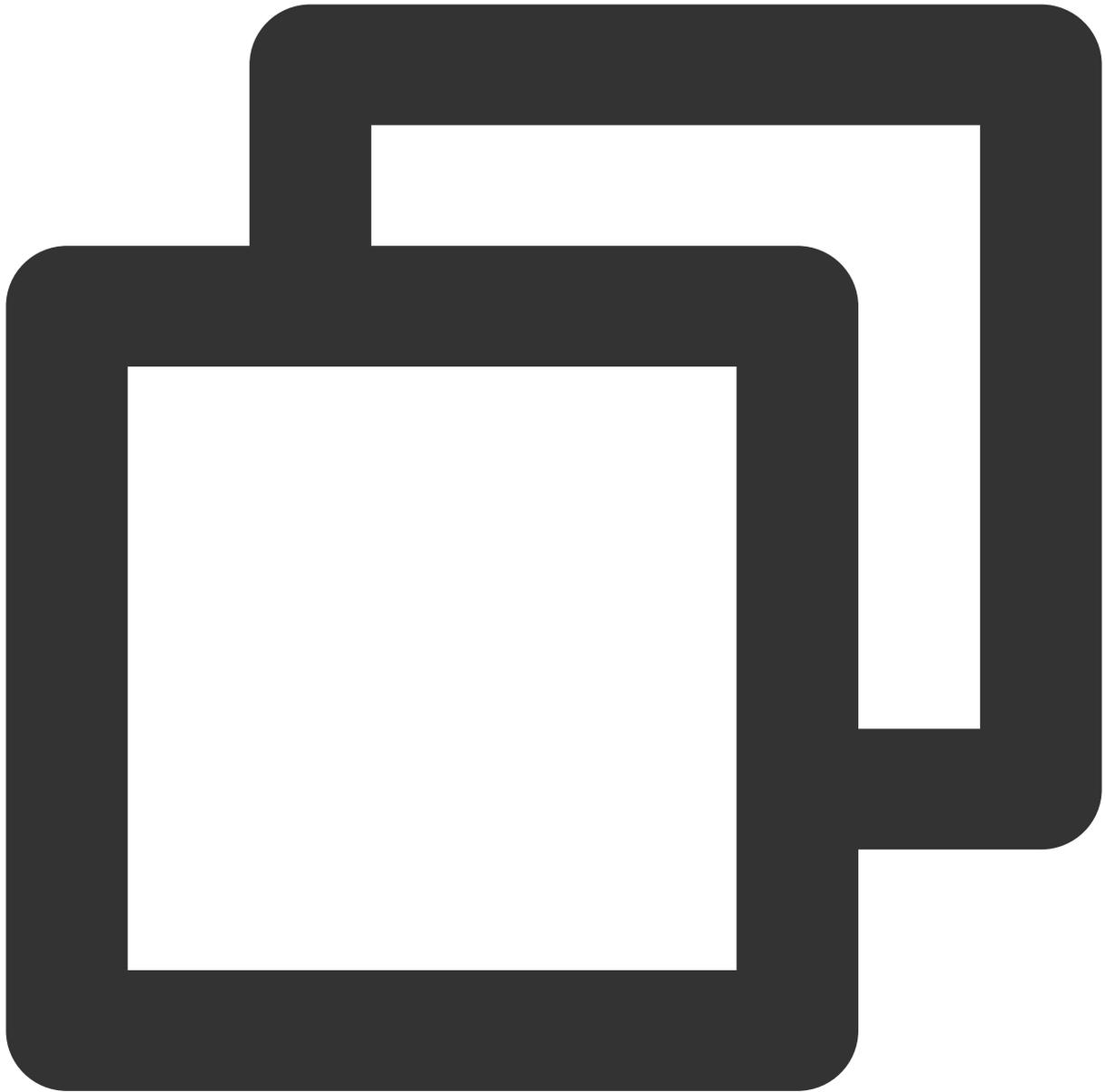
log 目录记录着工具迁移时的所有日志，若在迁移过程中出现错误，请先查看该目录下的 error.log。

3. 修改 config.ini 配置文件

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

3.1 配置迁移类型

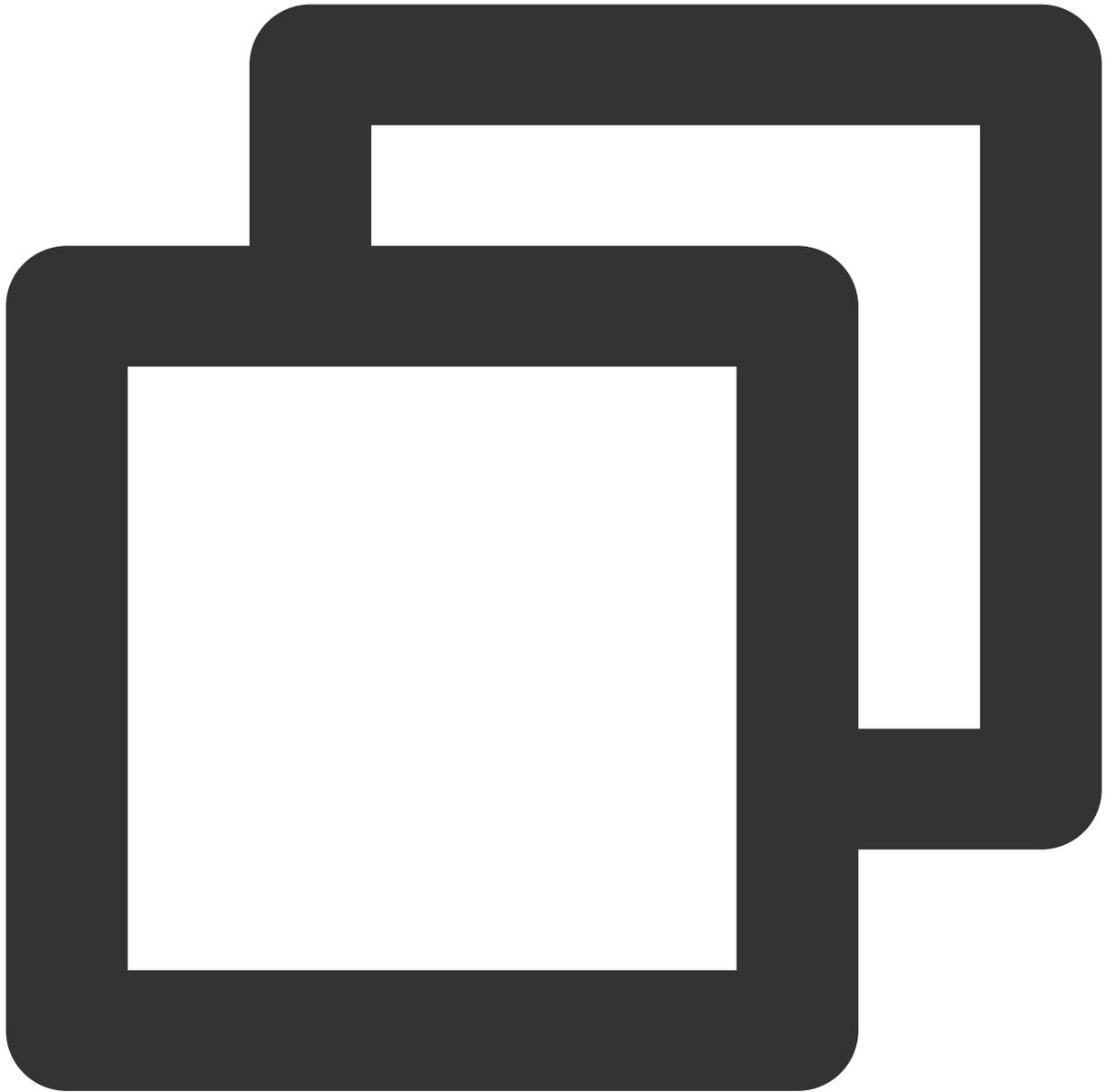
type 表示迁移类型，固定填写 `type=migrateLocal`。



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

3.2 配置迁移任务

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



迁移工具的公共配置分节，包含了需要迁移到目标 COS 的账户信息。

```
[common]
secretId=COS_SECRETID
secretKey=COS_SECRETKEY
bucketName=examplebucket-1250000000
region=ap-guangzhou
storageClass=Standard
cosPath=/
https=off
tmpFolder=./tmp
smallFileThreshold=5242880
```

```

smallFileExecutorNum=64
bigFileExecutorNum=8
entireFileMd5Attached=on
executeTimeWindow=00:00,24:00
outputFinishedFileFolder=./result
resume=false
skipSamePath=false
requestTryCount=5

```

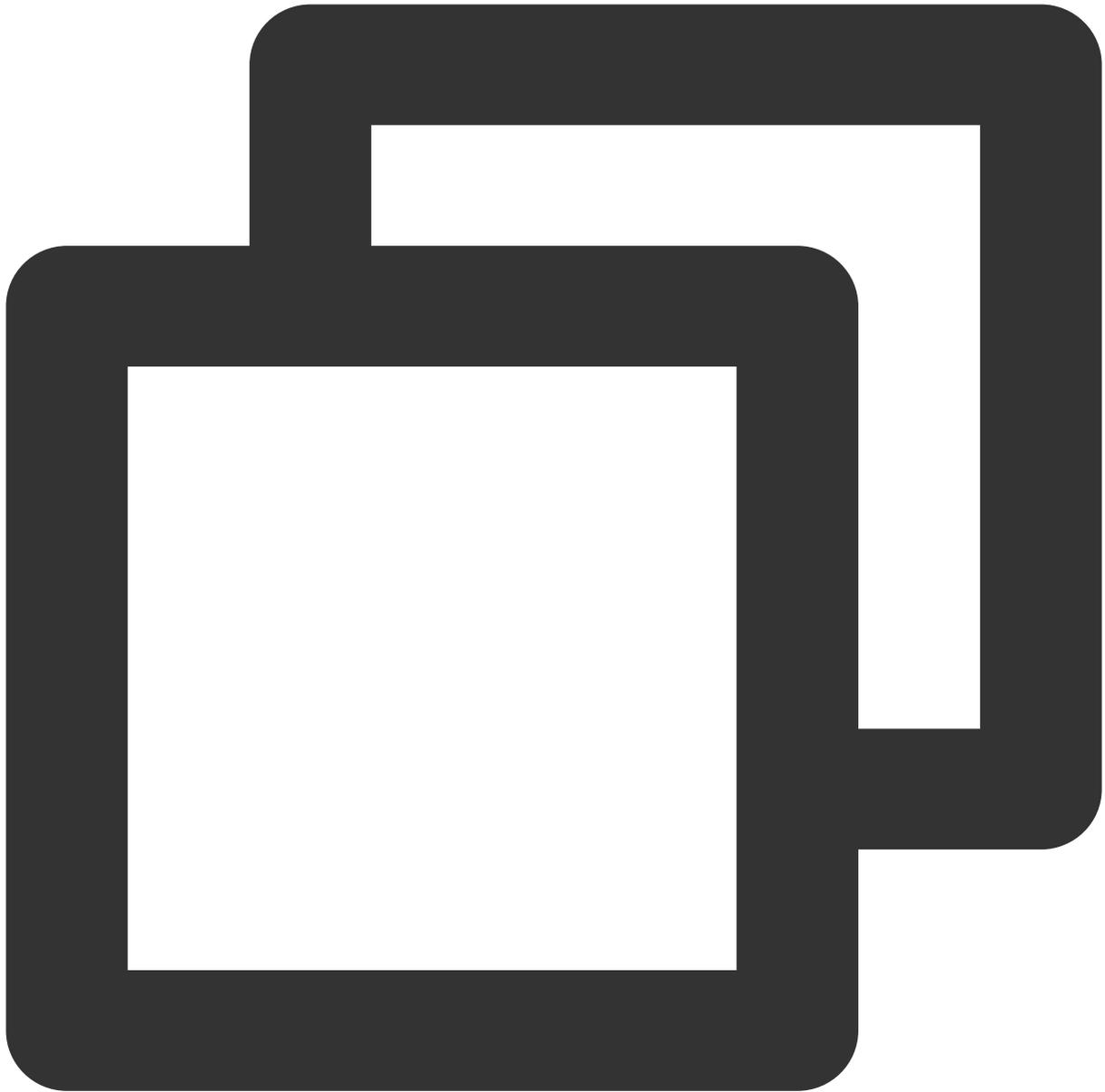
名称	描述	默认值
secretId	用户密钥 SecretId, 请将 COS_SECRETID 替换为您的真实密钥信息。可前往 访问管理控制台 中的云 API 密钥页面查看获取	-
secretKey	用户密钥 SecretKey, 请将 COS_SECRETKEY 替换为您的真实密钥信息。可前往 访问管理控制台 中的云 API 密钥页面查看获取	-
bucketName	目的 Bucket 的名称, 命名格式为 <BucketName-APPID>, 即 Bucket 名必须包含 APPID, 例如 examplebucket-1250000000	-
region	目的 Bucket 的 Region 信息。COS 的地域简称请参照 地域和访问域名	-
storageClass	数据迁移后的存储类型, 可选值为 Standard (标准存储)、Standard_IA (低频存储)、Archive (归档存储)、Maz_Standard (标准存储多 AZ)、Maz_Standard_IA (低频存储多 AZ), 相关介绍请参见 存储类型概述	Standard
cosPath	要迁移到的 COS 路径。 / 表示迁移到 Bucket 的根路径下, /folder/doc/ 表示要迁移到 Bucket 的 /folder/doc/ 下, 若 /folder/doc/ 不存在, 则会自动创建路径	/
https	是否使用 HTTPS 传输: on 表示开启, off 表示关闭。开启传输速度较慢, 适用于对传输安全要求高的场景	off
tmpFolder	从其他云存储迁移至 COS 的过程中, 用于存储临时文件的目录, 迁移完成后会删除。要求格式为绝对路径: Linux 下分隔符为单斜杠, 例如 /a/b/c Windows 下分隔符为两个反斜杠, 例如 E:\\\\a\\\\b\\\\c 默认为工具所在路径下的 tmp 目录	./tmp
smallFileThreshold	小文件阈值的字节, 大于等于这个阈值使用分块上传, 否则使用简单上传, 默认5MB (5242880 Byte)	5242880
smallFileExecutorNum	小文件 (文件小于 smallFileThreshold) 的并发度, 使用简单上传。如果是通过外网来连接 COS, 且带宽较小, 请减小该并发度	64

bigFileExecutorNum	大文件（文件大于等于 smallFileThreshold）的并发度，使用分块上传。如果是通过外网来连接 COS，且带宽较小，请减小该并发度	8
entireFileMd5Attached	表示迁移工具将全文的 MD5 计算后，存入文件的自定义头部 x-cos-meta-md5 中，用于后续的校验，因为 COS 的分块上传的大文件的 etag 不是全文的 MD5	on
executeTimeWindow	执行时间窗口，时刻粒度为分钟，该参数定义迁移工具每天执行的时间段。例如： 参数 03:30,21:00，表示在凌晨 03:30 到晚上 21:00 之间执行任务，其他时间则会进入休眠状态，休眠状态暂停迁移会保留迁移进度，直到下一个时间窗口自动继续执行。注意后面的时间点必须大于前面的时间点。	00:00,24:00
outputFinishedFileFolder	这个目录保存迁移成功的结果，结果文件会按照日期命名，例如 <code>./result/2021-05-27.out</code> ，其中 <code>./result</code> 为已创建的目录。文件内容每一行的格式为：绝对路径\t文件大小\t最后修改时间。设置为空，则不输出结果。	./result
resume	是否接着最后一次运行的结果，继续往下遍历源的文件列表。默认从头开始。	false
skipSamePath	如果 COS 上已经有相同的文件名，是否直接跳过。默认不跳过，即覆盖原有文件。	false
requestTryCount	每个文件上传总的尝试次数。	5

3.3 配置数据源信息

3.3.1 配置本地数据源 migrateLocal

若从本地迁移至 COS，则进行该部分配置，具体配置项及说明如下：



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

配置项	描述
localPath	本地目录，要求格式为绝对路径： Linux 下分隔符为单斜杠，例如 <code>/a/b/c</code>

	<p>Windows 下分隔符为两个反斜杠，例如 <code>E:\\\\a\\\\b\\\\c</code></p> <p>注意：此参数只能填目录的路径，不能填具体文件的路径，否则会导致目标对象名解析错误，在 <code>cosPath=/</code> 情况下，还会错误地解析成创桶请求。</p>
excludes	<p>要排除的目录或者文件的绝对路径，表示将 <code>localPath</code> 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 <code>localPath</code> 下面的全部迁移。</p>
ignoreModifiedTimeLessThanSeconds	<p>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 <code>lastmodified</code> 时间进行筛选，适用于客户在更新文件的同时又在运行迁移工具，并要求不把正在更新的文件迁移上传到 COS，例如设置为300，表示只上传更新了5分钟以上的文件。</p>

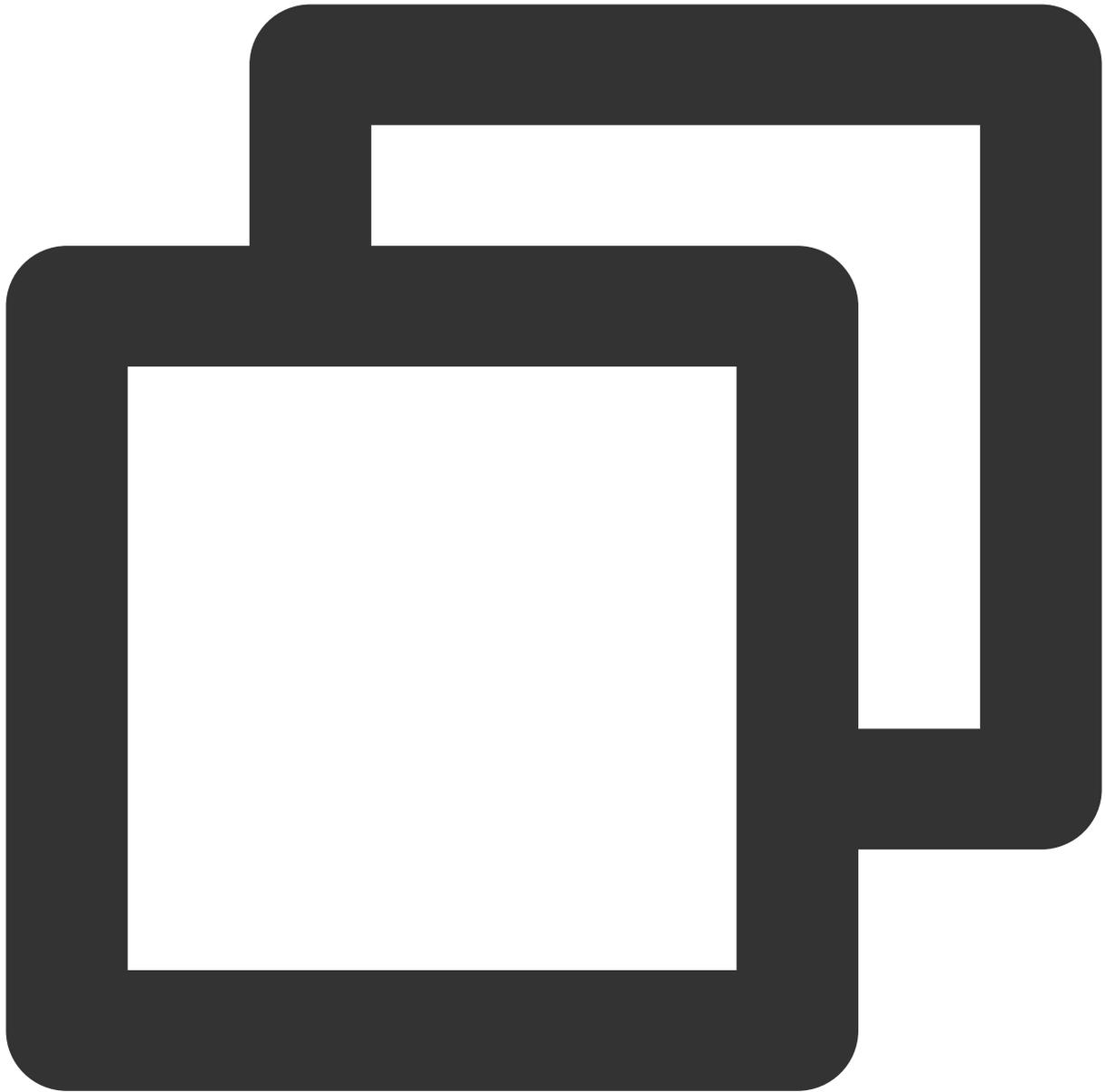
4. 运行迁移工具

Windows

双击 `start_migrate.bat` 即可运行。

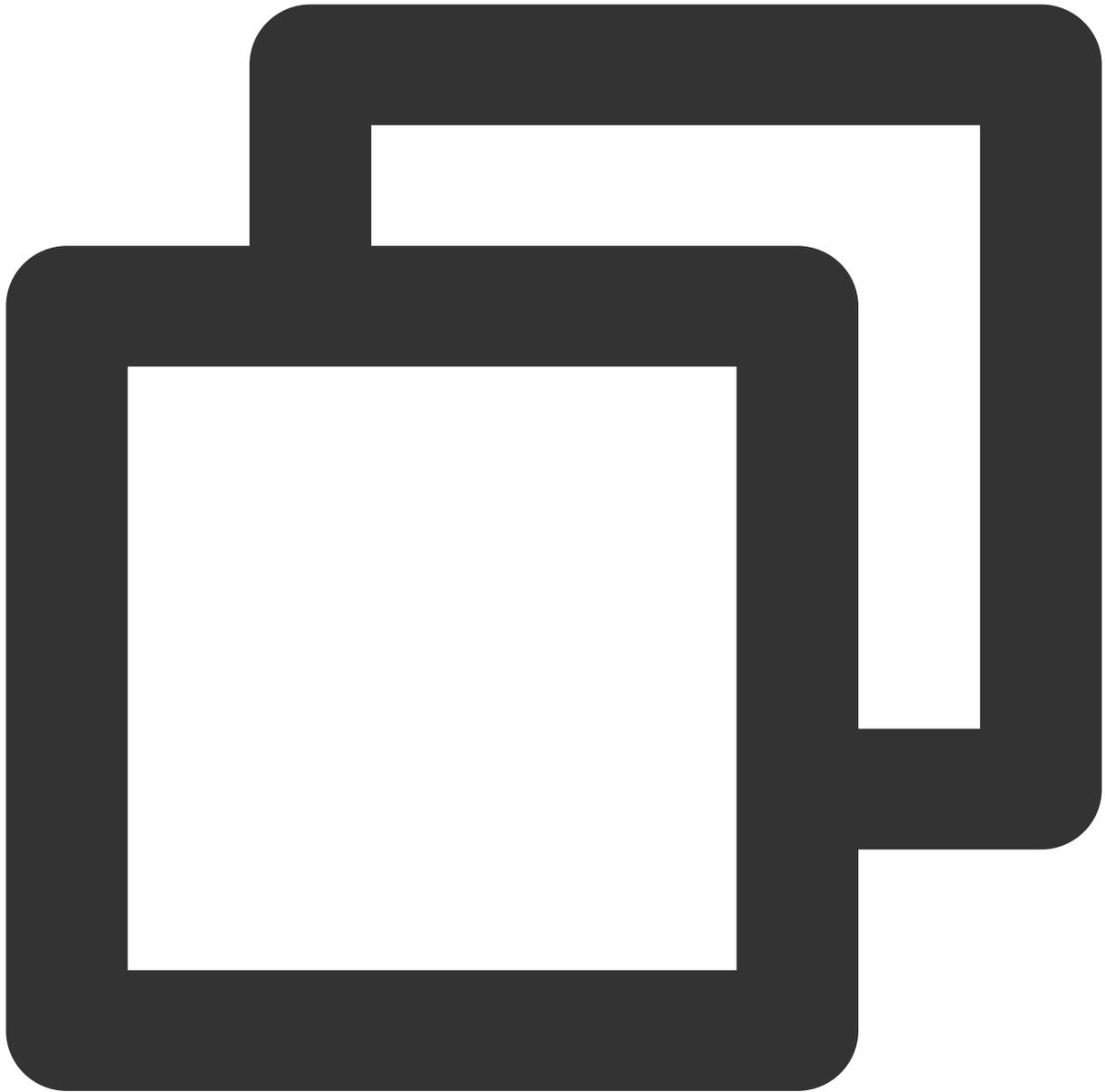
Linux

1.从 `config.ini` 配置文件读入配置，运行命令为：



```
sh start_migrate.sh
```

2.部分参数从命令行读入配置，运行命令为：



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

说明：

工具支持配置项读取方式有两种：命令行读取或配置文件读取。

命令行优先级高于配置文件，即相同配置选项会优先采用命令行里的参数。

命令行中读取配置项的形式方便用户同时运行不同的迁移任务，但前提是两次任务中的关键配置项不完全一样，例如 Bucket 名称，COS 路径，要迁移的源路径等。因为不同的迁移任务写入的是不同的 db 目录，可以保证并发迁移。请参照前文中的工具结构中的 db 信息。

配置项的形式为 `-D{sectionName}.{sectionKey}={sectionValue}` 的形式。其中 `sectionName` 是配置文件的分节名称，`sectionKey` 表示分节中配置项名称，`sectionValue` 表示分节中配置项值。如设置要迁移到的 COS 路径，则以 `-Dcommon.cosPath=/bbb/ddd` 表示。

迁移机制及流程

迁移机制原理

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

迁移流程步骤

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

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

结语

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

FTP Server 工具

最近更新时间：2024-01-06 10:43:44

功能概述

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

功能说明

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

下载机制：直接流式返回给客户端。

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

多 bucket 绑定：支持同时绑定多个 bucket。

说明

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

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

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

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

说明

FTP Server 工具暂时不支持断点续传功能。

开始使用

系统环境

操作系统：Linux，推荐使用腾讯 CentOS 系列 [云服务器](#)，暂时不支持 Windows 系统。

psutil 依赖的 Linux 系统包：python-devel（或 python-dev，依据不同的 Linux 发行版名字不同），通过 Linux 下的包管理工具添加，例如 `yum install python-devel` 或 `aptitude install python-dev`。

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

说明

FTP Server 工具不支持 Python 3。

依赖包：

[cos-python-sdk-v5](#) (≥1.6.5)

[pyftplib](#) (≥1.5.2)

[psutil](#) (≥5.6.1)

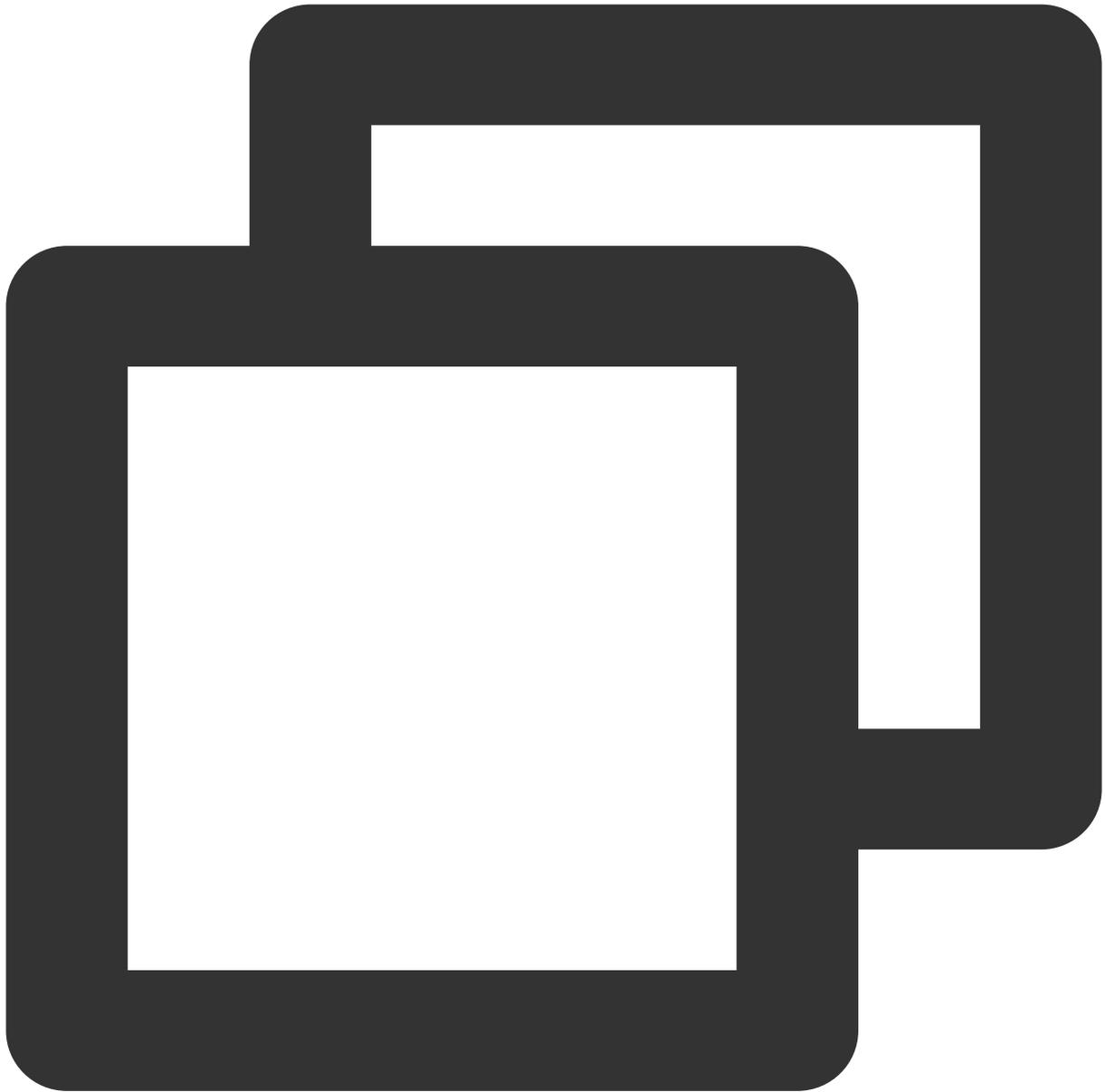
使用限制

适用于 COS XML 版本。

安装运行

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

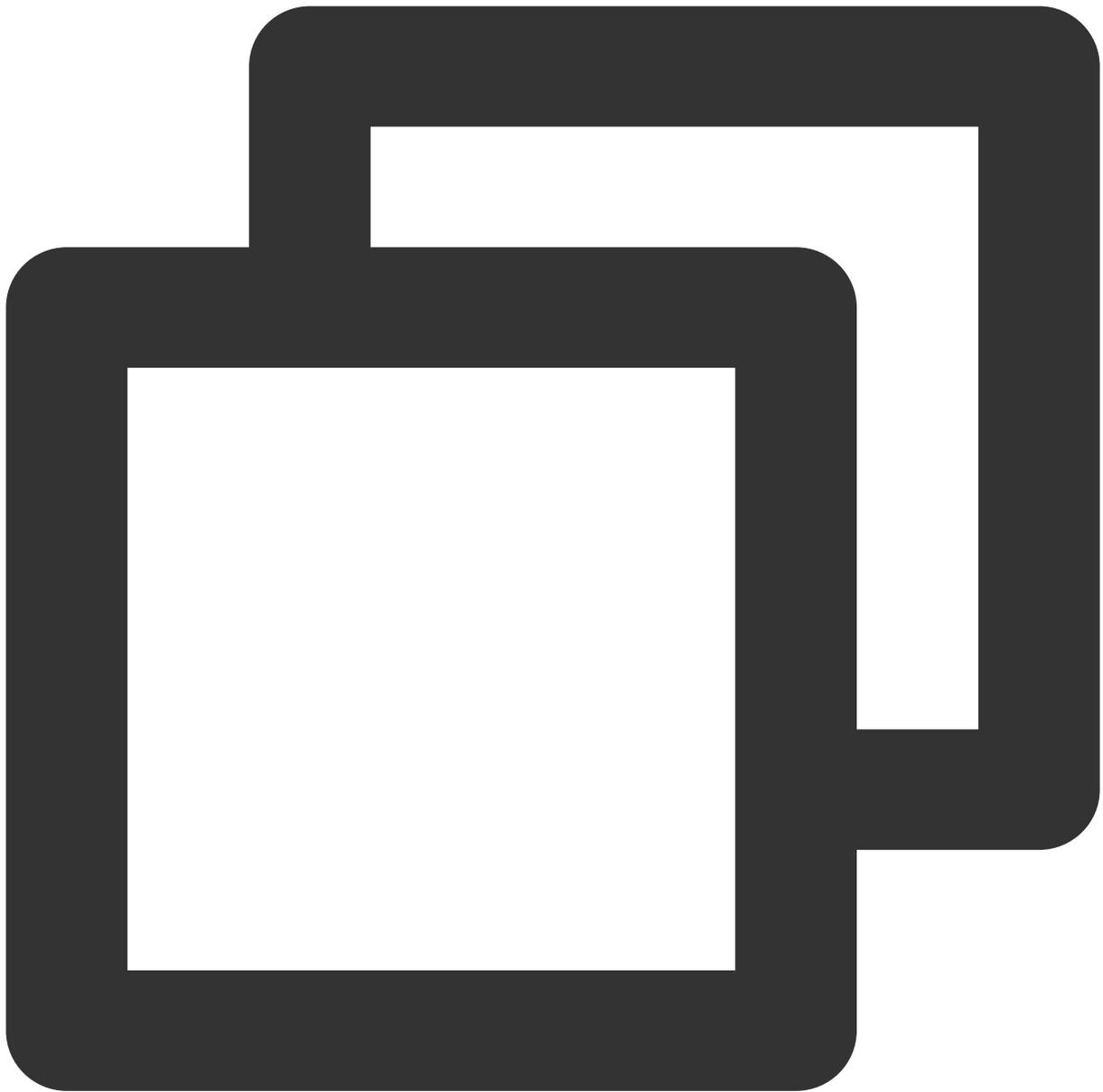
1. 进入 FTP Server 目录，运行 `setup.py` 安装 FTP Server 及其相关的依赖库（需要联网）：



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

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

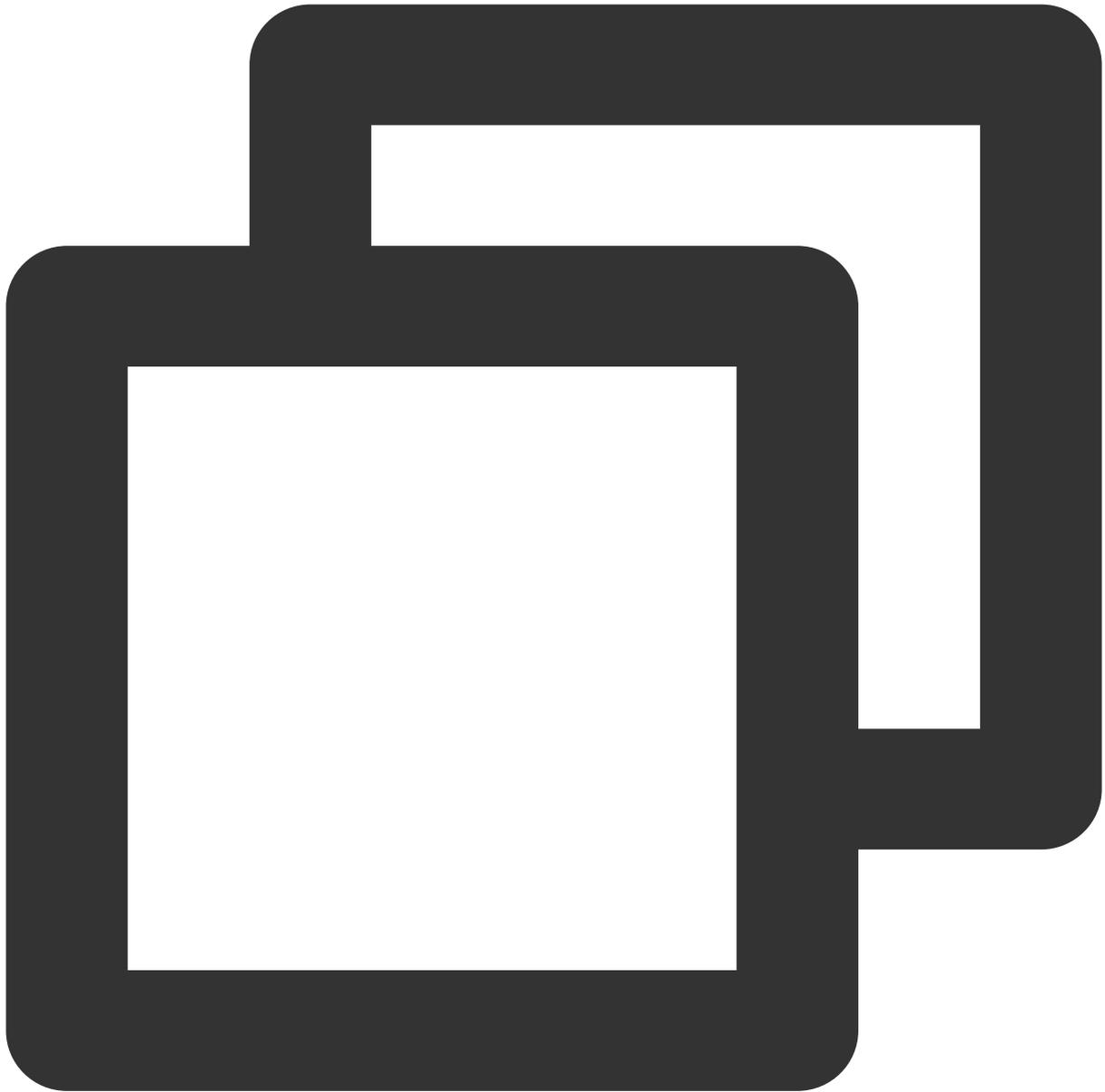
3. 运行 `ftp_server.py` 启动 FTP Server：



```
python ftp_server.py
```

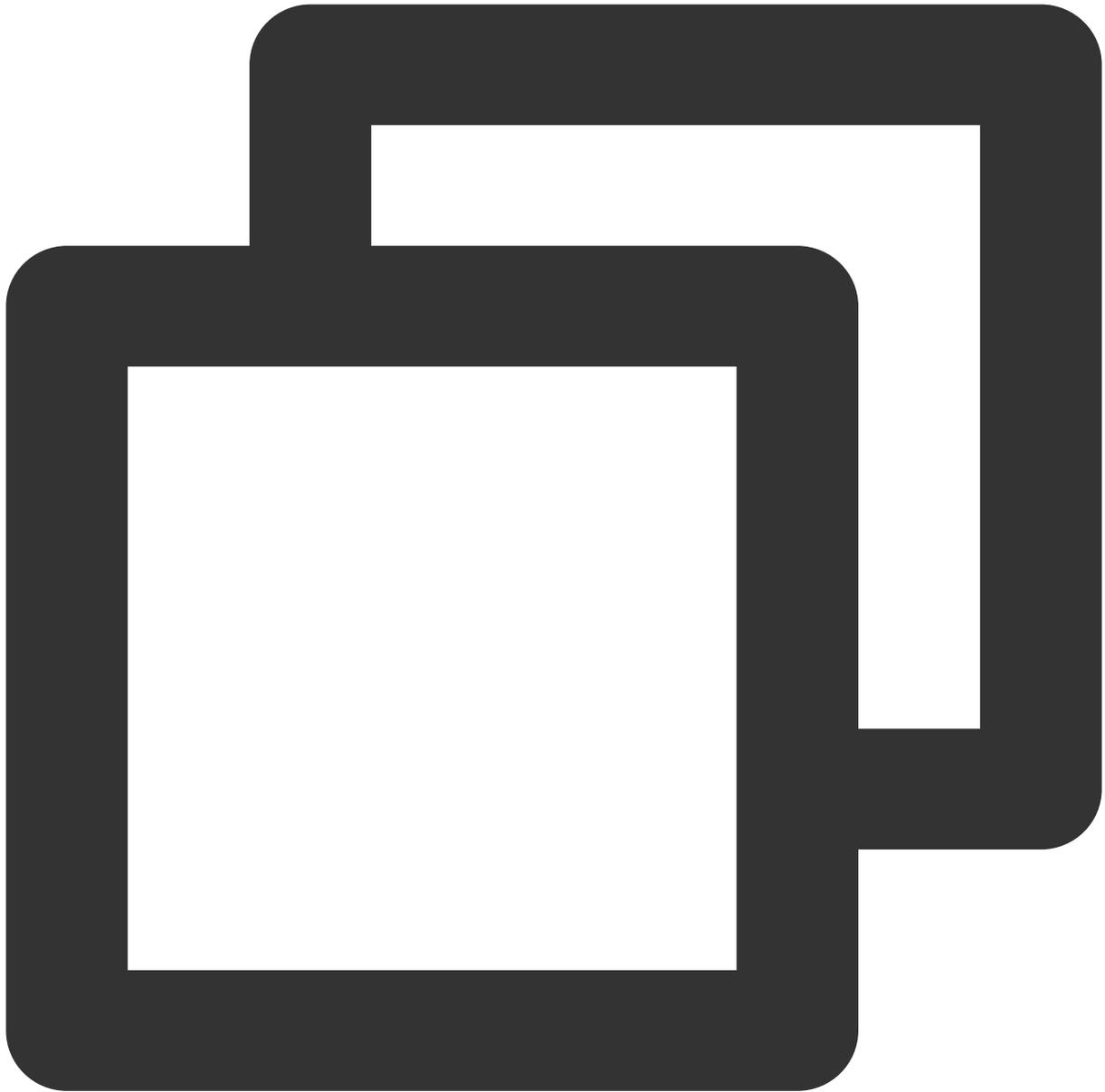
此外还有两种方式启动 FTP Server，如下：

使用 `nohup` 命令，以后台进程方式启动：



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

使用 `screen` 命令放入后台运行（需要安装 `screen` 工具）：

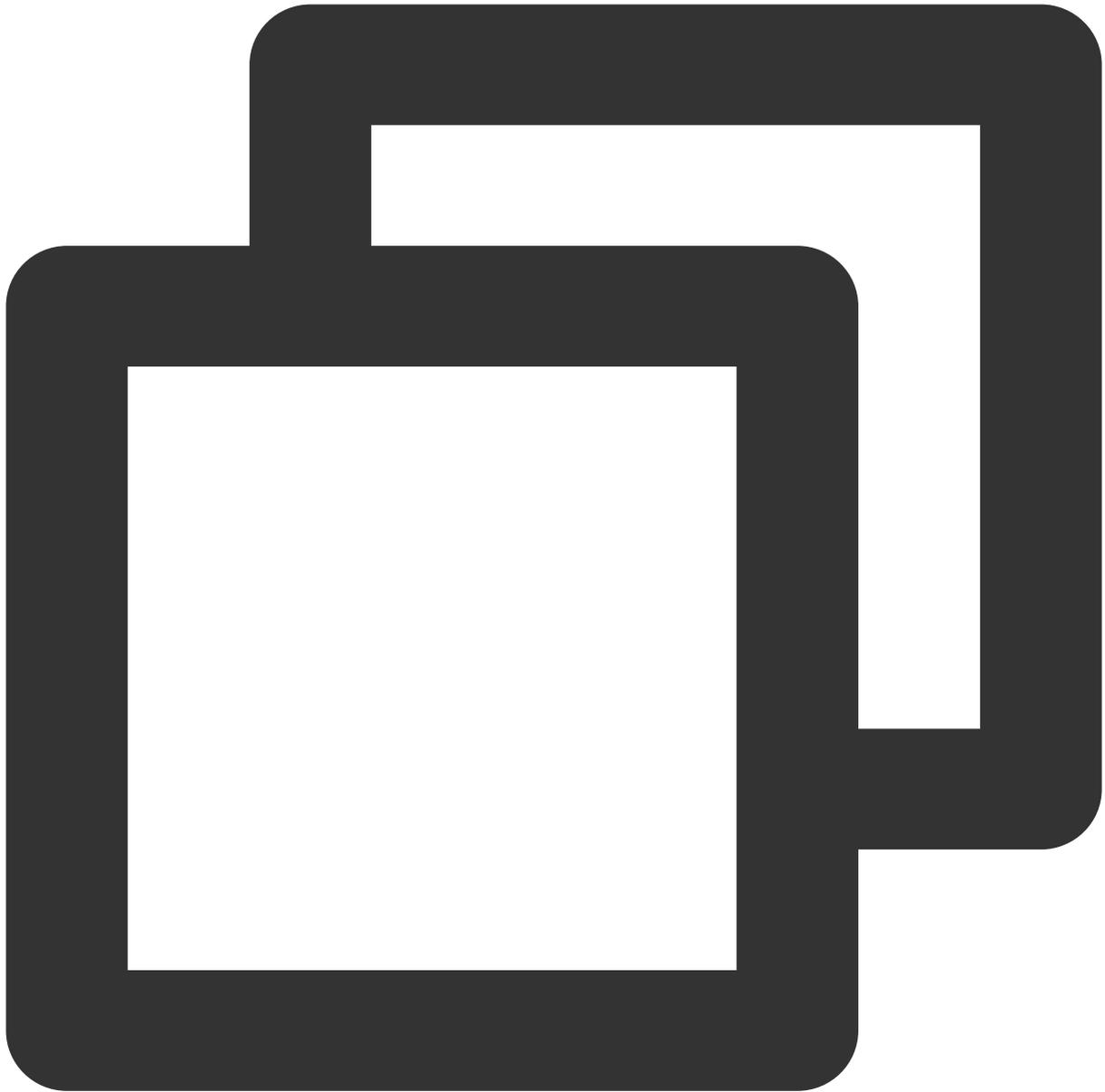


```
screen -dmS ftp
screen -r ftp
python ftp_server.py
#使用快捷键 Ctrl+A+D , 切回主 screen 即可。
```

停止运行

若您直接运行，或 `screen` 方式放在后台运行的 FTP Server，您可以使用快捷键 `Ctrl+C` 停止 FTP Server 运行。

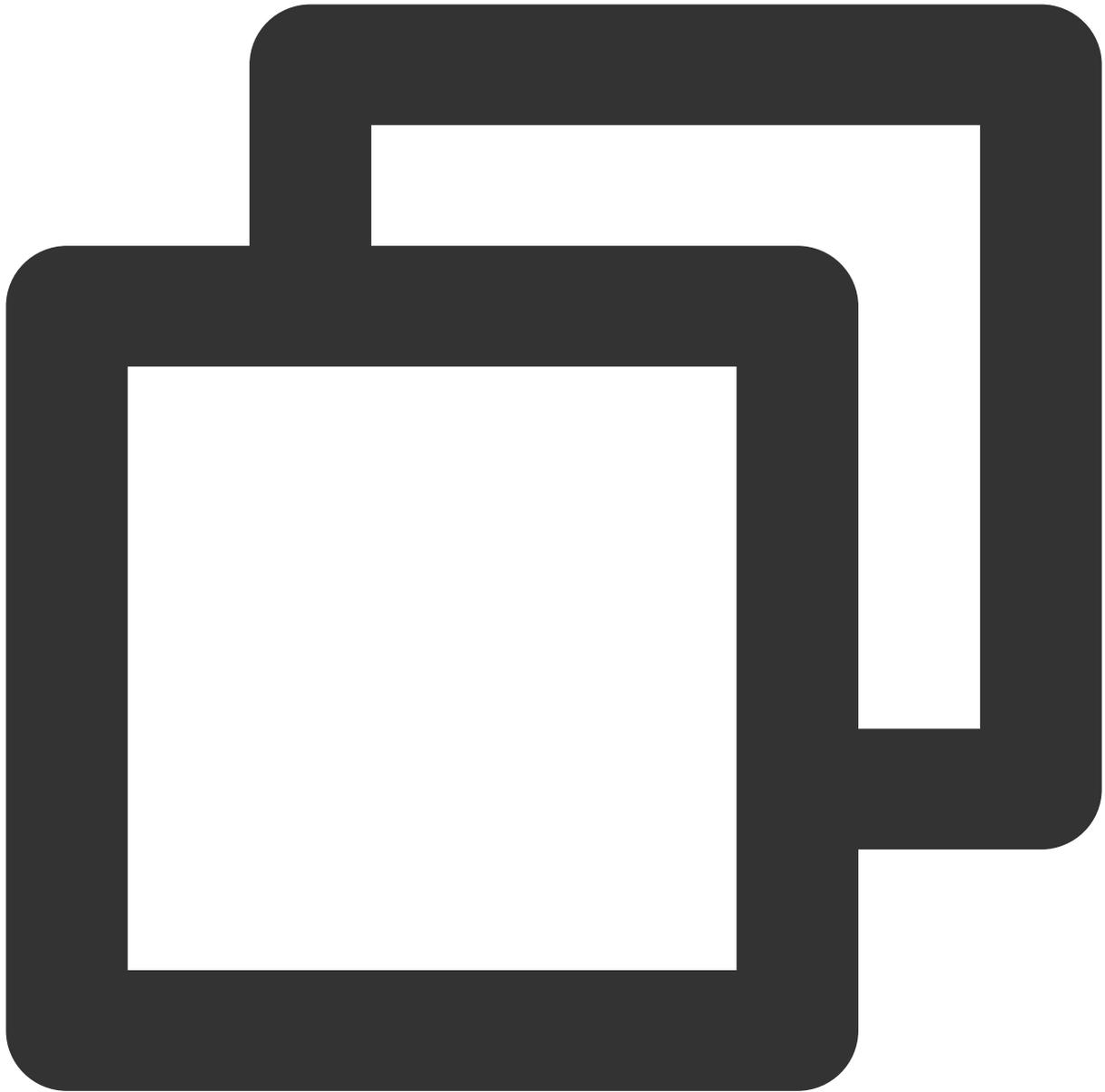
若您是通过 `nohup` 命令启动，可以使用下面方式停止：



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

配置文件

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



```
[COS_ACCOUNT_0]
cos_secretid = COS_SECRETID      # 替换为您的 SECRETID
cos_secretkey = COS_SECRETKEY    # 替换为您的 SECRETKEY
cos_bucket = examplebucket-1250000000
cos_region = region             # 替换为您的存储桶地域
cos_protocol = https
#cos_endpoint = region.myqcloud.com
home_dir = /home/user0          # 替换为您希望 FTP 挂载到的本地路径（需设置为机器上实际存在的路径）
ftp_login_user_name=user0       # 替换为用户自定义的账号
ftp_login_user_password=pass0   # 替换为用户自定义的密码
authority=RW                     # 设置该用户的读写权限，R 表示读权限，W 表示写权限，RW 表示同时具
```

```
delete_enable=true # true 为允许该 ftp 用户进行删除操作（默认），false 为禁

[COS_ACCOUNT_1]
cos_secretid = COS_SECRETID # 替换为您的 SECRETID
cos_secretkey = COS_SECRETKEY # 替换为您的 SECRETKEY
cos_bucket = examplebucket-1250000000
cos_region = region # 替换为您的存储桶地域
cos_protocol = https
#cos_endpoint = region.myqcloud.com
home_dir = /home/user1 # 替换为您希望 FTP 挂载到的本地路径（需设置为机器上实际存在的路径，
ftp_login_user_name=user1 # 替换为用户自定义的账号
ftp_login_user_password=pass1 #替换为用户自定义的密码
authority=RW # 设置该用户的读写权限，R 表示读权限，W 表示写权限，RW 表示同时具
delete_enable=false # true 为允许该 ftp 用户进行删除操作（默认），false 为禁止该用户

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

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

[OPTIONAL]
# 以下设置，如无特殊需要，建议保留 default 设置。如需设置，请合理填写一个整数
min_part_size = default
upload_thread_num = default
max_connection_num = 512
max_list_file = 10000 # ls 命令最大可列出的文件数目，建议不要设置太大
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
```

说明

如果要将每个用户绑定到不同的 bucket 上，则只需要添加 [COS_ACCOUNT_X] 的 section 即可。

针对每个不同的 COS_ACCOUNT_X 的 section 有如下说明：

每个 ACCOUNT 下的用户名（ftp_login_user_name）和用户的主目录（home_dir）必须各不相同，并且主目录必须是系统中真实存在的目录。

每个 COS FTP Server 允许同时登录的用户数目不能超过100。

endpoint 和 region 不会同时生效，使用公有云 COS 服务只需要正确填写 region 字段即可，endpoint 常用于私有化部署环境中。当同时填写了 region 和 endpoint，则 endpoint 会优先生效。

配置文件中的 `OPTIONAL` 选项是提供给高级用户用于调整上传性能的可选项，根据机器的性能合理地调整上传分片的大小和并发上传的线程数，可以获得更好的上传速度，一般用户不需要调整，保持默认值即可。

同时，提供最大连接数的限制选项。这里如果不想限制最大连接数，可以填写0，即表示不限制最大连接数目（不过需要根据您机器的性能合理评估）。

在配置文件的 `masquerade_address` 配置项中，一般建议指定为客户端连接 COS FTP Server 所使用的 IP 地址。如您对此有疑问，可参见 [FTP Server 工具 常见问题文档](#)。

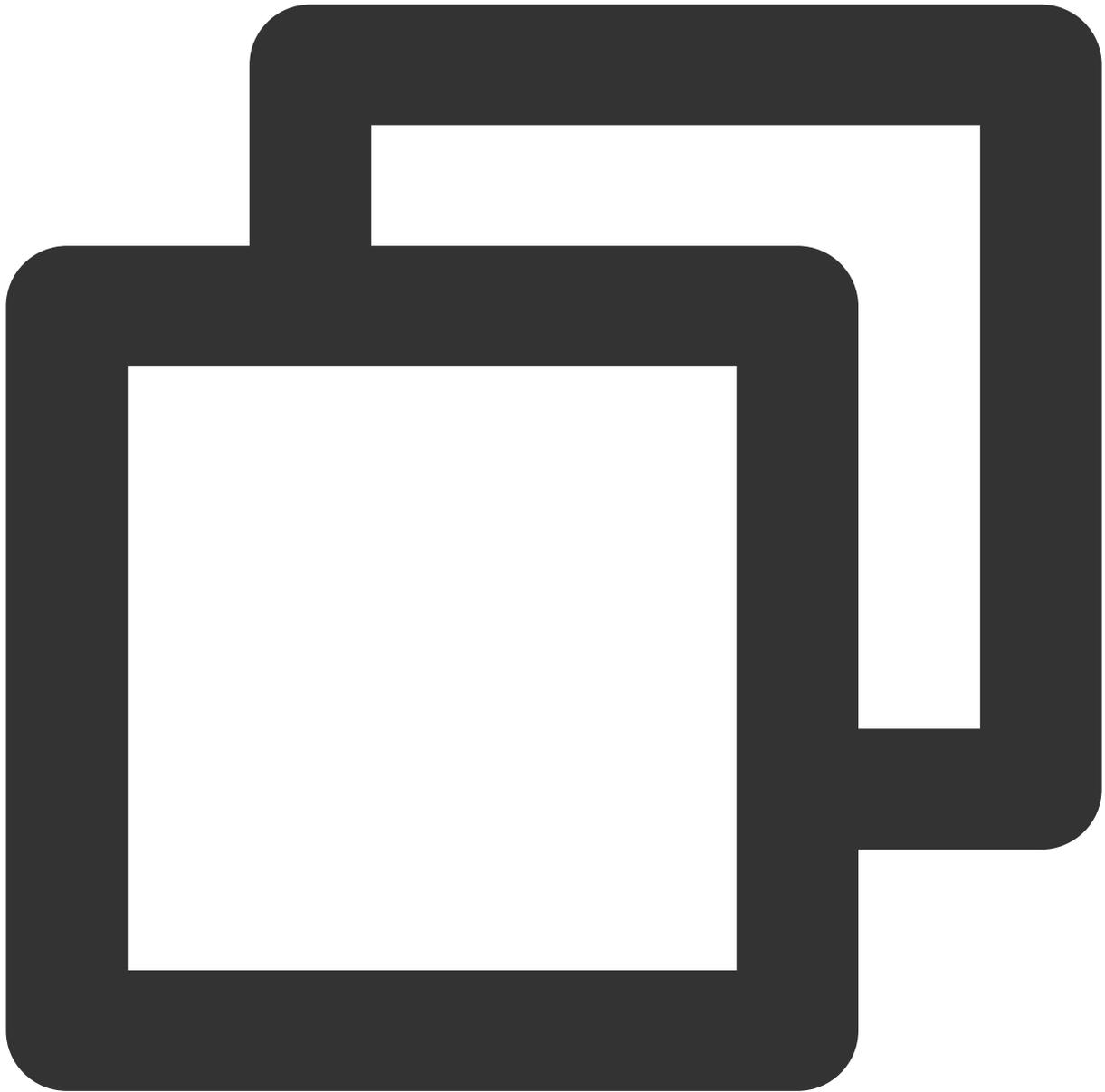
当 FTP Server 有多个 IP 地址时，执行 `ifconfig` 命令后，得到映射到外网的网卡 IP 为 `10.xxx.xxx.xxx`，它映射的外网 IP 假设为 `119.xxx.xxx.xxx`。此时，若 FTP Server 未显式配置 `masquerade_address` 为客户端访问 server 时的外网 IP（`119.xxx.xxx.xxx`），则 FTP Server 在 `Passive` 模式下，给客户端回包可能会使用内网地址（`10.xxx.xxx.xxx`）。这时就会出现客户端能够连上 FTP Server，但却不能正常给客户端返回数据包的情况。因此，通常情况下，建议用户将 `masquerade_address` 配置为客户端连接 Server 时所使用的那个 IP 地址。

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

快速实践

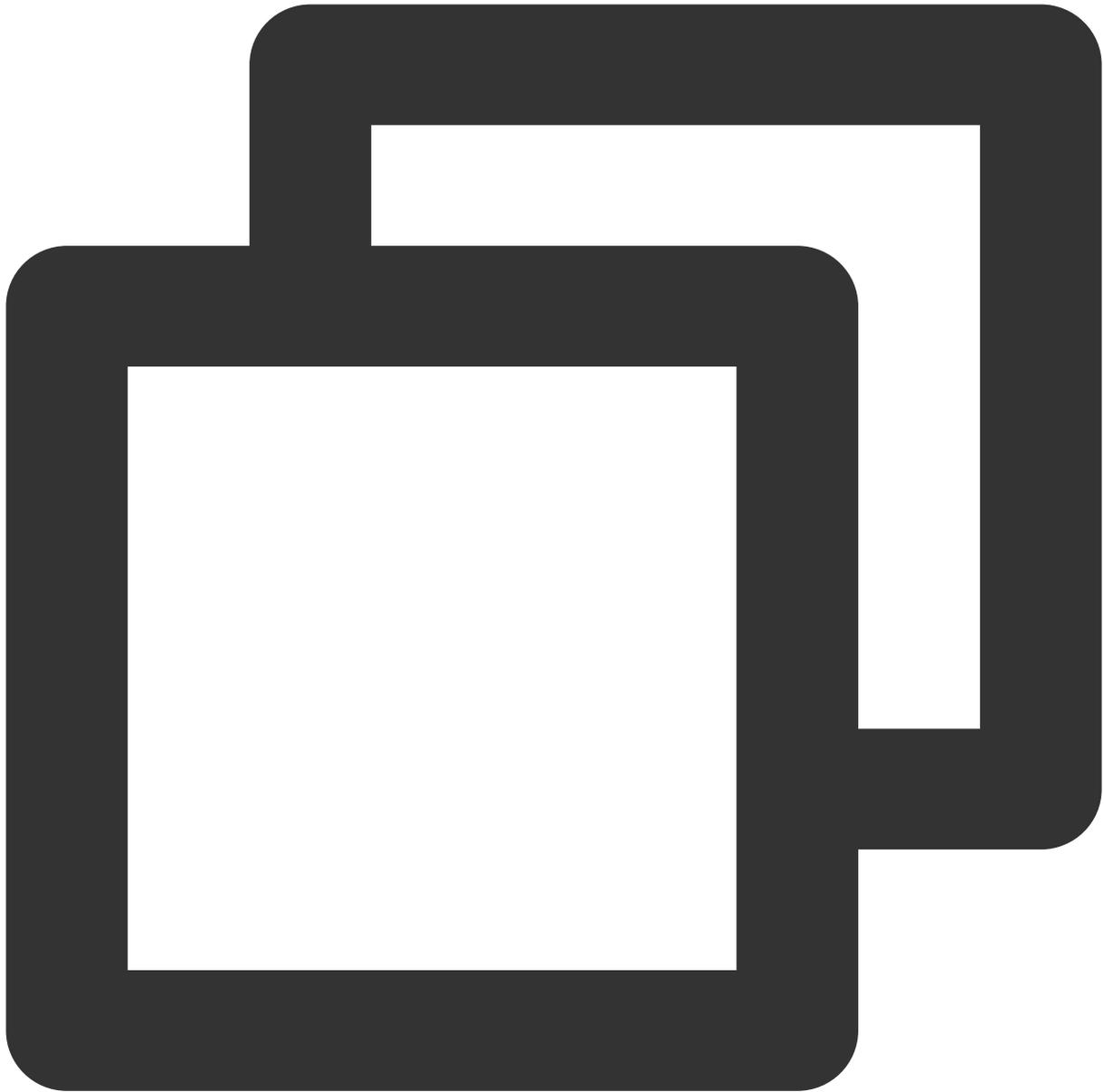
使用 Linux ftp 命令 访问 COS FTP Server

1. 安装 Linux ftp 客户端



```
yum install -y ftp
```

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



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

ftp 命令中，IP 的设置对应配置示例文件 `conf/vsftpd.conf.example` 中的 **masquerade_address** 配置项。在本例中 IP 设置为 192.xxx.xx.103。

ftp 命令中，端口的设置对应配置示例文件 `conf/vsftpd.conf.example` 中的 **listen_port** 配置项。在本例中设置为2121。

3. 运行上述命令后，出现 **Name**和 **Password** 待输入项，输入 COS FTP Server 配置项 `ftp_login_user_name` 和 `ftp_login_user_password` 中配置的内容，即可连接成功。

Name：对应配置示例文件 `conf/vsftpd.conf.example` 中的 `ftp_login_user_name` 配置项（需要进行配置）。

Password：对应配置示例文件 `conf/vsftpd.conf.example` 中的 `ftp_login_user_password` 配置项（需要进行配置）。

使用 FileZilla 访问 COS FTP Server

1. 下载 [FileZilla 客户端](#) 并安装。
2. 在 FileZilla 客户端配置 COS FTP Server 的访问信息后，单击[快速连接](#)。

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

注意

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

用户名 (U)：对应配置示例文件 `conf/vsftpd.conf.example` 中的 `ftp_login_user_name` 配置项（需要进行配置）。

密码 (W)：对应配置示例文件 `conf/vsftpd.conf.example` 中的 `ftp_login_user_password` 配置项（需要进行配置）。

端口 (P)：对应配置示例文件 `conf/vsftpd.conf.example` 中的 `listen_port` 配置项。在本例中设置为 2121。

常见问题

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

Hadoop 工具

最近更新时间：2024-01-06 10:43:44

功能说明

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

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

使用环境

系统环境

支持 Linux、Windows 和 macOS 系统。

软件依赖

Hadoop-2.6.0及以上版本。

说明

目前 Hadoop-COS 已经正式被 Apache Hadoop-3.3.0 [官方集成](#)。

在 Apache Hadoop-3.3.0 之前版本或 CDH 集成 Hadoop-cos jar 包后，需要重启 NodeManager 才能加载到 jar 包。

需要编译具体 Hadoop 版本的 jar 包时，可更改 pom 文件中 hadoop.version 进行编译。

下载与安装

获取 Hadoop-COS 分发包及其依赖

下载地址：[Hadoop-COS release](#)。

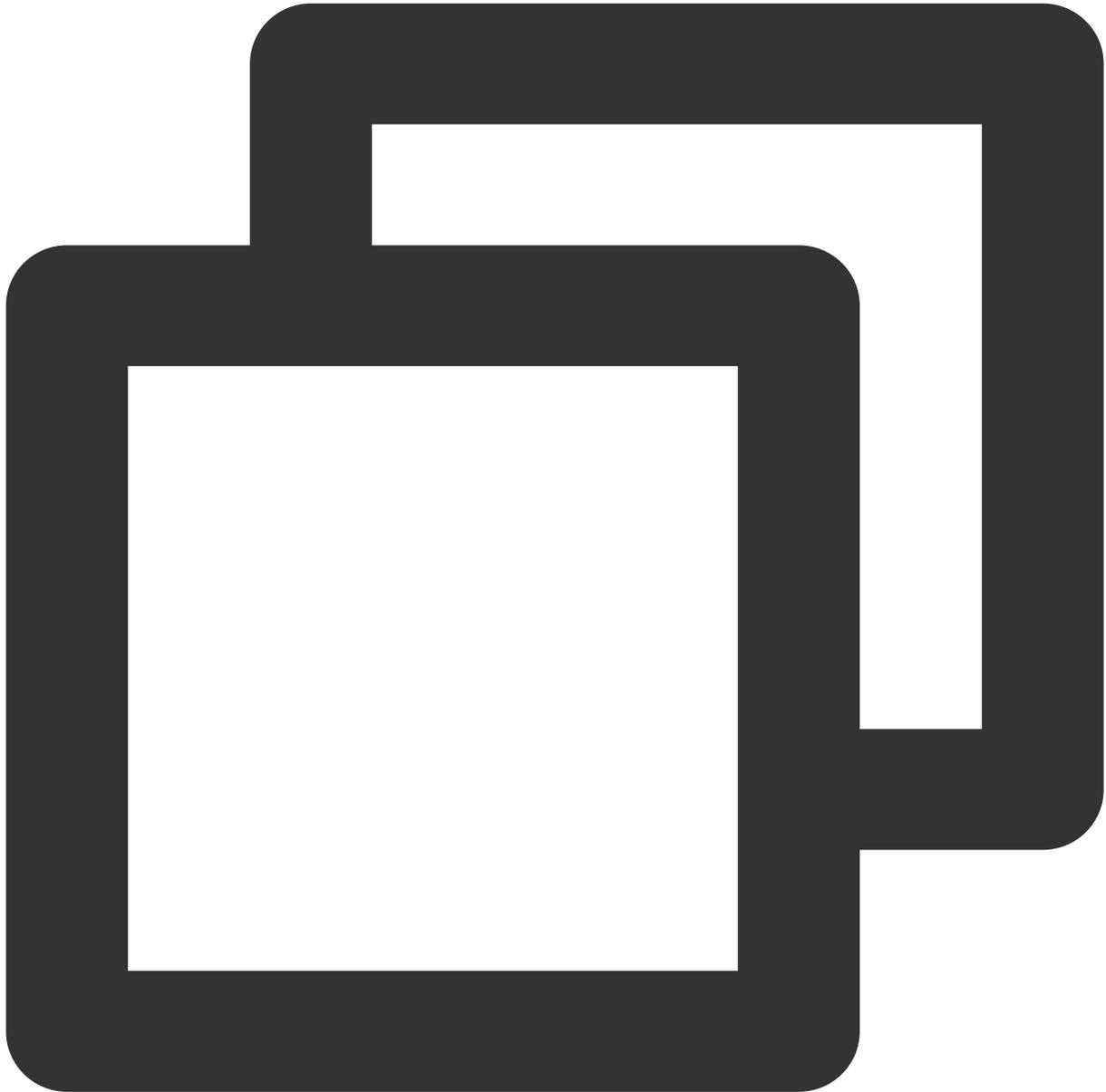
安装 Hadoop-COS 插件

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

说明

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

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



```
for f in $HADOOP_HOME/share/hadoop/tools/lib/*.jar; do
  if [ "$HADOOP_CLASSPATH" ]; then
    export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
  else
    export HADOOP_CLASSPATH=$f
  fi
done
```

配置方法

配置项说明

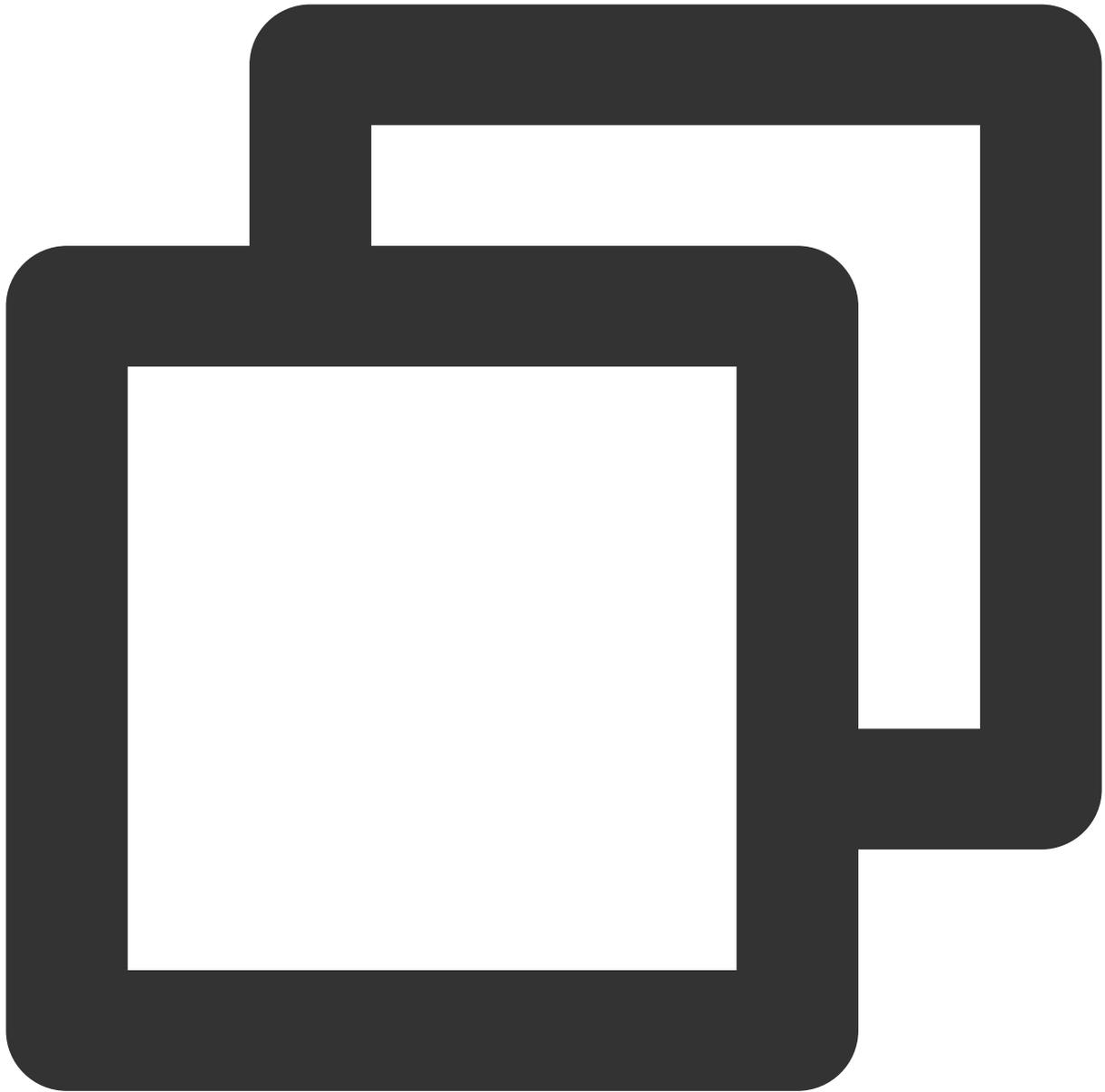
属性键	说明	默
fs.cosn.userinfo.secretId/secretKey	填写您账户的 API 密钥信息。可登录 访问管理控制台 查看云 API 密钥。	无
fs.cosn.credentials.provider	<p>配置 SecretId 和 SecretKey 的获取方式。当前支持如下获取方式：</p> <ol style="list-style-type: none"> org.apache.hadoop.fs.auth.SessionCredentialProvider：从请求 URI 中获取 secret id 和 secret key。其格式为：<code>cosn://{secretId}:{secretKey}@examplebucket-1250000000/</code> org.apache.hadoop.fs.auth.SimpleCredentialProvider：从 core-site.xml 配置文件中读取 fs.cosn.userinfo.secretId 和 fs.cosn.userinfo.secretKey 来获取 SecretId 和 SecretKey。 org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider：从系统环境变量 COS_SECRET_ID 和 COS_SECRET_KEY 中获取。 org.apache.hadoop.fs.auth.SessionTokenCredentialProvider：使用 临时密钥形式 访问。 org.apache.hadoop.fs.auth.CVMInstanceCredentialsProvider：利用腾讯云云服务器（CVM）绑定的角色，获取访问 COS 的临时密钥。 org.apache.hadoop.fs.auth.CPMInstanceCredentialsProvider：利用腾讯云黑石物理机（CPM）绑定的角色，获取访问 COS 的临时密钥。 org.apache.hadoop.fs.auth.EMRInstanceCredentialsProvider：利用腾讯云 EMR 实例绑定的角色，获取访问 COS 的临时密钥。 org.apache.hadoop.fs.auth.RangerCredentialsProvider 使用 ranger 进行获取密钥。 	默 如 以 1. 2. 3. or 4. 5. 6. 7.
fs.cosn.useHttps	配置是否使用 HTTPS 作为与 COS 后端的传输协议。	true
fs.cosn.impl	cosn 对 FileSystem 的实现类，固定为 org.apache.hadoop.fs.CosFileSystem。	无
fs.AbstractFileSystem.cosn.impl	cosn 对 AbstractFileSystem 的实现类，固定为 org.apache.hadoop.fs.CosN。	无
fs.cosn.bucket.region	请填写待访问存储桶的地域信息，枚举值请参见 地域和访问域名 中的地域简称，	无

	例如：ap-beijing、ap-guangzhou 等。兼容原有配置： fs.cosn.userinfo.region。	
fs.cosn.bucket.endpoint_suffix	指定要连接的 COS endpoint，该项为非必填项目。对于公有云 COS 用户而言，只需要正确填写上述的 region 配置即可。兼容原有配置：fs.cosn.userinfo.endpoint_suffix。配置该项时请删除 fs.cosn.bucket.region 配置项 endpoint 才能生效。	无
fs.cosn.tmp.dir	请设置一个实际存在的本地目录，运行过程中产生的临时文件会暂时放于此处。	/tr
fs.cosn.upload.part.size	CosN 文件系统每个 block 的大小，也是分块上传的每个 part size 的大小。由于 COS 的分块上传最多只能支持10000块，因此需要预估最大可能使用到的单文件大小。 例如，part size 为8MB时，最大能够支持78GB的单文件上传。part size 最大可以支持到2GB，即单文件最大可支持19TB。	83
fs.cosn.upload.buffer	CosN 文件系统上传时依赖的缓冲区类型。当前支持三种类型的缓冲区：非直接内存缓冲区（non_direct_memory），直接内存缓冲区（direct_memory），磁盘映射缓冲区（mapped_disk）。非直接内存缓冲区使用的是 JVM 堆内存，直接内存缓冲区使用的是堆外内存，而磁盘映射缓冲区则是基于内存文件映射得到的缓冲区。	m
fs.cosn.upload.buffer.size	CosN 文件系统上传时依赖的缓冲区大小，如果指定为-1，则表示不限制缓冲区。若不限制缓冲区大小，则缓冲区的类型必须为 mapped_disk。如果指定大小大于0，则要求该值至少大于等于一个 block 的大小。兼容原有配置 fs.cosn.buffer.size。	-1
fs.cosn.block.size	CosN 文件系统 block size。	13
fs.cosn.upload_thread_pool	文件流式上传到 COS 时，并发上传的线程数目。	10
fs.cosn.copy_thread_pool	目录拷贝操作时，可用于并发拷贝和删除文件的线程数目。	3
fs.cosn.read.ahead.block.size	预读块的大小。	10
fs.cosn.read.ahead.queue.size	预读队列的长度。	8
fs.cosn.maxRetries	访问 COS 出现错误时，最多重试的次数。	20
fs.cosn.retry.interval.seconds	每次重试的时间间隔。	3
fs.cosn.server-side-encryption.algorithm	配置 COS 服务端加密算法，支持 SSE-C 和 SSE-COS，默认为空，不加密。	无
fs.cosn.server-side-encryption.key	当开启 COS 的 SSE-C 服务端加密算法时，必须配置 SSE-C 的密钥，密钥格式为 base64 编码的 AES-256 密钥，默认为空，不加	无

	密。	
fs.cosn.client-side-encryption.enabled	是否开启客户端加密，默认不开启。开启后必须配置客户端加密的公钥和私钥。此时无法使用 <code>append</code> 、 <code>truncate</code> 接口。	fa
fs.cosn.client-side-encryption.public.key.path	客户端加密公钥文件的绝对路径	无
fs.cosn.client-side-encryption.private.key.path	客户端加密私钥文件的绝对路径	无
fs.cosn.crc64.checksum.enabled	是否开启 CRC64 校验。默认不开启，此时无法使用 <code>hadoop fs -checksum</code> 命令获取文件的 CRC64 校验值。	fa
fs.cosn.crc32c.checksum.enabled	是否开启 CRC32C 校验。默认不开启，此时无法使用 <code>hadoop fs -checksum</code> 命令获取文件的 CRC32C 校验值，只能开启一种校验方式： <code>crc32c</code> 或 <code>crc64</code> 。	fa
fs.cosn.traffic.limit	上传带宽的控制选项，819200 - 838860800 bits/s，默认值为-1，默认表示不限制。	无

Hadoop 配置

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



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

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

```
    1.org.apache.hadoop.fs.auth.SessionCredentialProvider: Obtain the secre
```

2.org.apache.hadoop.fs.auth.SimpleCredentialProvider: Obtain the secret from fs.cosn.userinfo.secretId and fs.cosn.userinfo.secretKey in core-s

3.org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider: Obtain from system environment variables named COS_SECRET_ID and COS_SECRET_KE

If unspecified, the default order of credential providers is:

1. org.apache.hadoop.fs.auth.SessionCredentialProvider
2. org.apache.hadoop.fs.auth.SimpleCredentialProvider
3. org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider
4. org.apache.hadoop.fs.auth.SessionTokenCredentialProvider
5. org.apache.hadoop.fs.auth.CVMInstanceCredentialsProvider
6. org.apache.hadoop.fs.auth.CPMInstanceCredentialsProvider
7. org.apache.hadoop.fs.auth.EMRInstanceCredentialsProvider

</description>

</property>

<property>

<name>fs.cosn.userinfo.secretId</name>

<value>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</value>

<description>Tencent Cloud Secret Id</description>

</property>

<property>

<name>fs.cosn.userinfo.secretKey</name>

<value>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</value>

<description>Tencent Cloud Secret Key</description>

</property>

<property>

<name>fs.cosn.bucket.region</name>

<value>ap-xxx</value>

<description>The region where the bucket is located.</description>

</property>

<property>

<name>fs.cosn.bucket.endpoint_suffix</name>

<value>cos.ap-xxx.myqcloud.com</value>

<description>

COS endpoint to connect to.

For public cloud users, it is recommended not to set this option, and onl

</description>

</property>

<property>

<name>fs.cosn.impl</name>

<value>org.apache.hadoop.fs.CosFileSystem</value>

<description>The implementation class of the CosN Filesystem.</description>

```
</property>

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

<property>
  <name>fs.cosn.tmp.dir</name>
  <value>/tmp/hadoop_cos</value>
  <description>Temporary files will be placed here.</description>
</property>

<property>
  <name>fs.cosn.upload.buffer</name>
  <value>mapped_disk</value>
  <description>The type of upload buffer. Available values: non_direct_memory
</property>

<property>
  <name>fs.cosn.upload.buffer.size</name>
  <value>134217728</value>
  <description>The total size of the upload buffer pool. -1 means unlimited.<
</property>

<property>
  <name>fs.cosn.upload.part.size</name>
  <value>8388608</value>
  <description>Block size to use cosn filesystem, which is the part size for
Considering the COS supports up to 10000 blocks, user should estimate the m
For example, 8MB part size can allow writing a 78GB single file.</descript
</property>

<property>
  <name>fs.cosn.maxRetries</name>
  <value>3</value>
  <description>
    The maximum number of retries for reading or writing files to
    COS, before we signal failure to the application.
  </description>
</property>

<property>
  <name>fs.cosn.retry.interval.seconds</name>
  <value>3</value>
  <description>The number of seconds to sleep between each COS retry.</descript
```

```
</property>

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

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

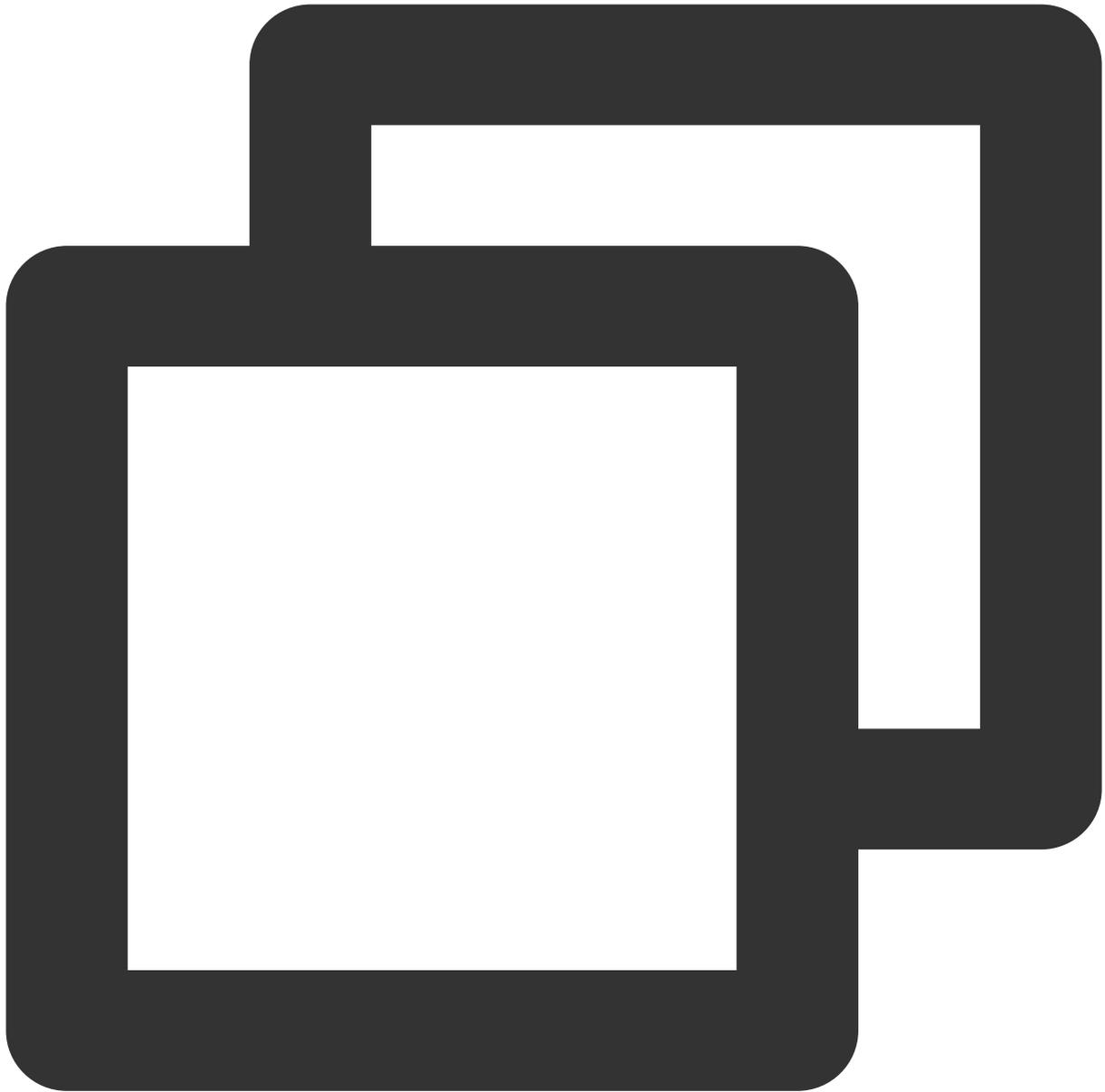
<property>
  <name>fs.cosn.client-side-encryption.enabled</name>
  <value></value>
  <description>Enable or disable the client encryption function</description>
</property>

<property>
  <name>fs.cosn.client-side-encryption.public.key.path</name>
  <value>/xxx/xxx.key</value>
  <description>The direct path to the public key</description>
</property>

<property>
  <name>fs.cosn.client-side-encryption.private.key.path</name>
  <value>/xxx/xxx.key</value>
  <description>The direct path to the private key</description>
</property>

</configuration>
```

其中 `fs.defaultFS` 不建议在生产环境进行配置，若您需要用于部分测试场景（例如 `hive-testbench` 等），可添加如下配置信息：



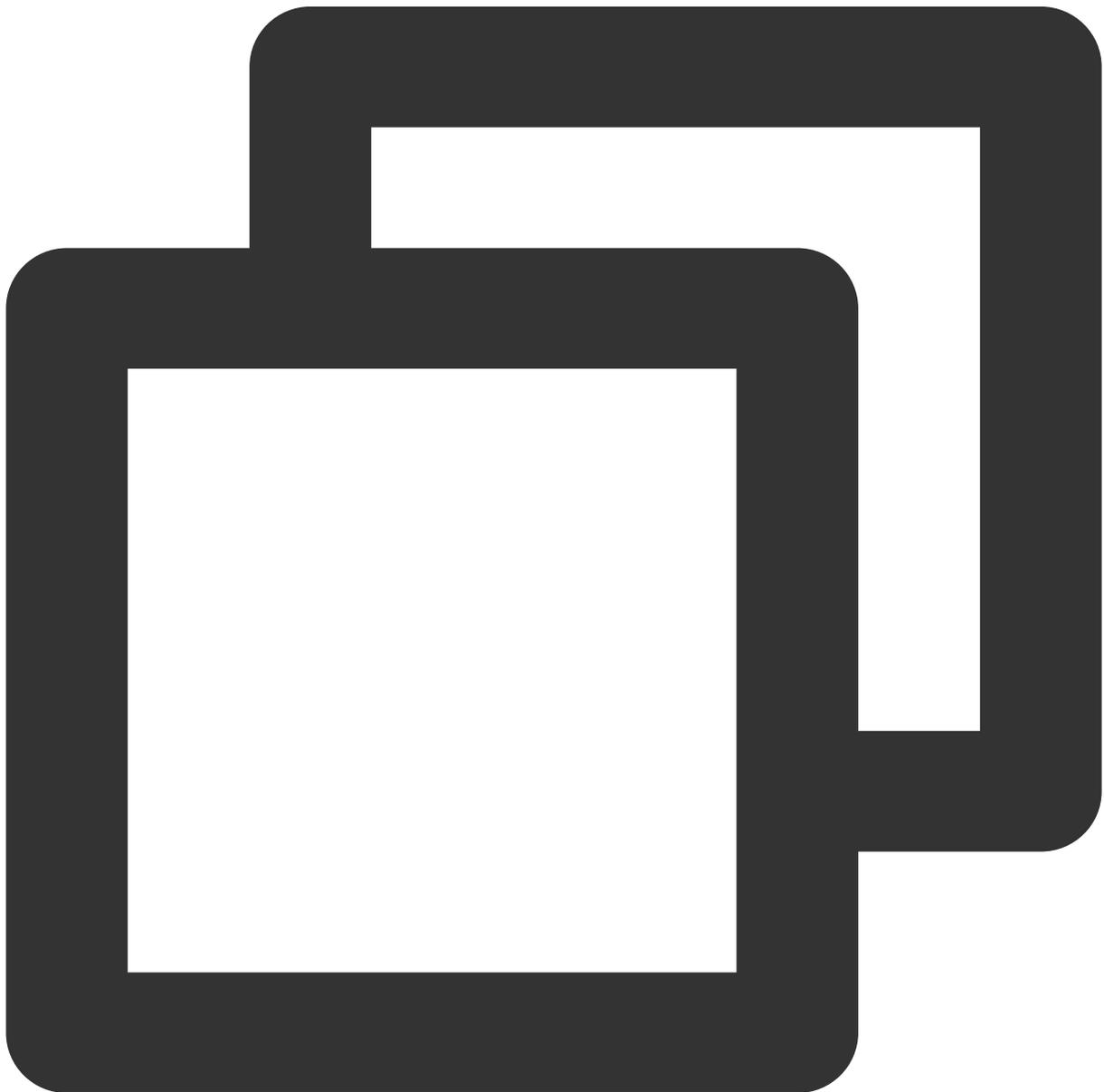
```
<property>
  <name>fs.defaultFS</name>
  <value>cosn://examplebucket-1250000000</value>
  <description>
    This option is not advice to config, this only used for some special t
  </description>
</property>
```

服务端加密

Hadoop-COS 支持服务端加密，目前提供两种加密方式：COS 托管密钥方式（SSE-COS）和用户自定义密钥方式（SSE-C），Hadoop-COS 的加密功能默认为关闭状态，用户可以选择开启，通过以下方式进行配置。

SSE-COS 加密

SSE-COS 加密即 COS 托管密钥的服务端加密，由腾讯云 COS 托管主密钥和管理数据。当使用 Hadoop-COS 时，用户可以在 `$HADOOP_HOME/etc/hadoop/core-site.xml` 文件中，增加以下配置来进行实现 SSE-COS 加密。



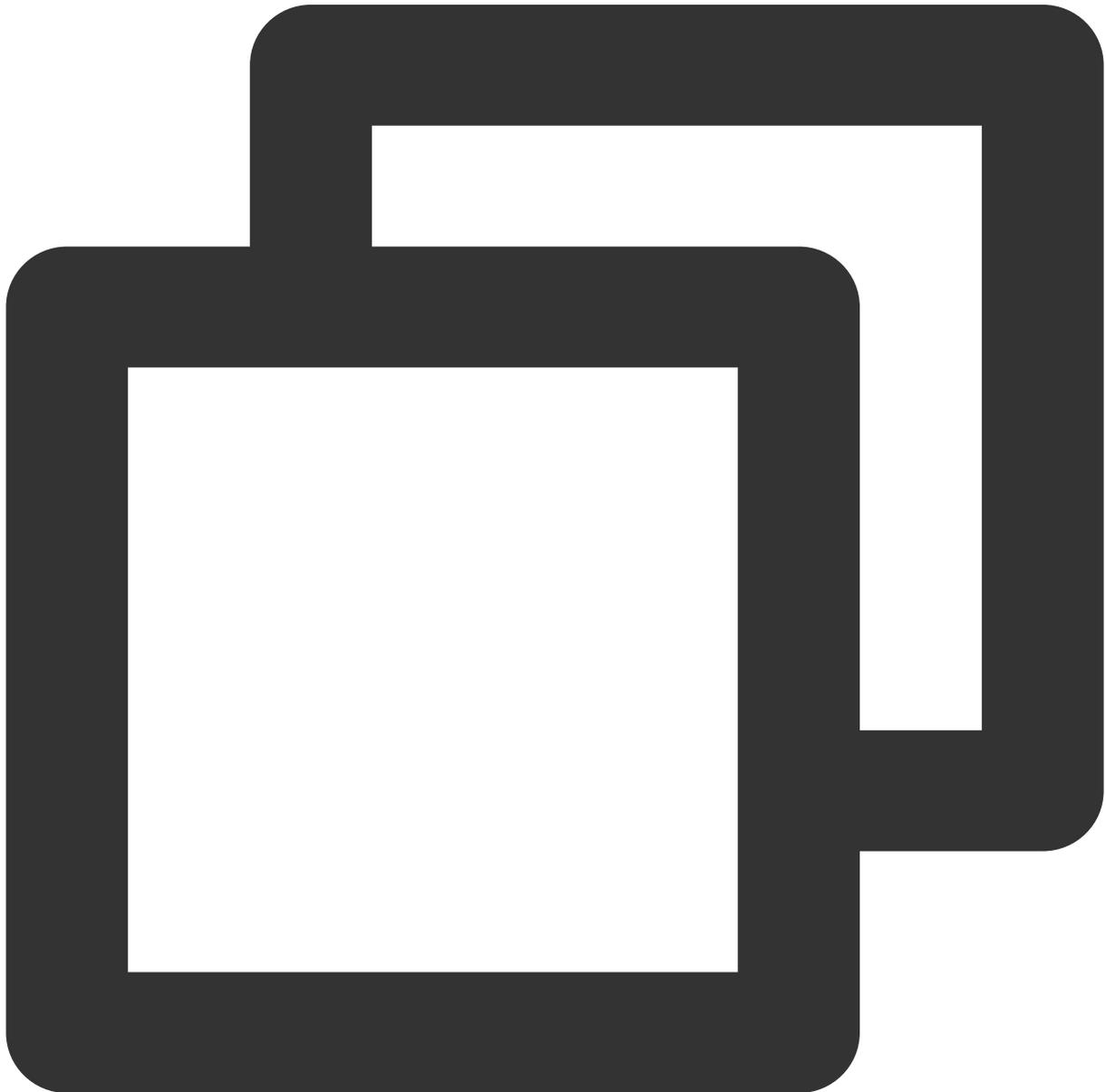
```
<property>
  <name>fs.cosn.server-side-encryption.algorithm</name>
```

```
<value>SSE-COS</value>
<description>The server side encryption algorithm.</description>
</property>
```

SSE-C 加密

SSE-C 加密即用户自定义密钥的服务端加密。加密密钥由用户自己提供，用户在上传对象时，COS 将使用用户提供的加密密钥对用户的数据进行 AES-256 加密。当使用 Hadoop-COS 时，用户可以

在 `$SHADOOP_HOME/etc/hadoop/core-site.xml` 文件中，增加以下配置来进行实现 SSE-C 加密。



```
<property>
```

```
<name>fs.cosn.server-side-encryption.algorithm</name>
<value>SSE-C</value>
<description>The server side encryption algorithm.</description>
</property>
<property>
  <name>fs.cosn.server-side-encryption.key</name>
  <value>MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNERUY=</value> #用户需要自行配置
  <description>The SSE-C server side encryption key.</description>
</property>
```

注意

Hadoop-COS 的 SSE-C 服务端加密依赖于 COS 的 SSE-C 服务端加密。因此，Hadoop-COS 不存储用户提供的加密密钥。同时需要值得注意的是，COS 的 SSE-C 服务端加密方式不存储用户提供的加密密钥，而是存储加密密钥添加了随机数据的 HMAC 值，该值用于验证用户访问对象的请求。COS 无法使用随机数据的 HMAC 值来推导出加密密钥的值或解密加密对象的内容。因此，如果用户丢失了加密密钥，则无法再次获取到该对象。

Hadoop-COS 配置了 SSE-C 服务端加密算法时，必须在 `fs.cosn.server-side-encryption.key` 配置项中配置 SSE-C 的密钥，密钥格式为 base64 编码的 AES-256 密钥。

客户端加密

COSN 客户端加密采用 RSA 加密方式，密钥分为公钥和私钥，其中公钥用于文件加密过程，私钥用于文件解密过程。在上传文件时，COSN 会生成一个随机密钥，并用该密钥对文件进行对称加密。公钥会对该密钥进行加密，并将加密后的信息保存在文件元数据中。在下载文件时，COSN 会使用私钥从文件元数据中得到加密随机密钥进行解密，再使用解密后的随机密钥对文件进行对此解密。公钥和私钥只参与客户端本地计算，不会在网络上进行传输或保存在服务端，以保证主密钥的数据安全。

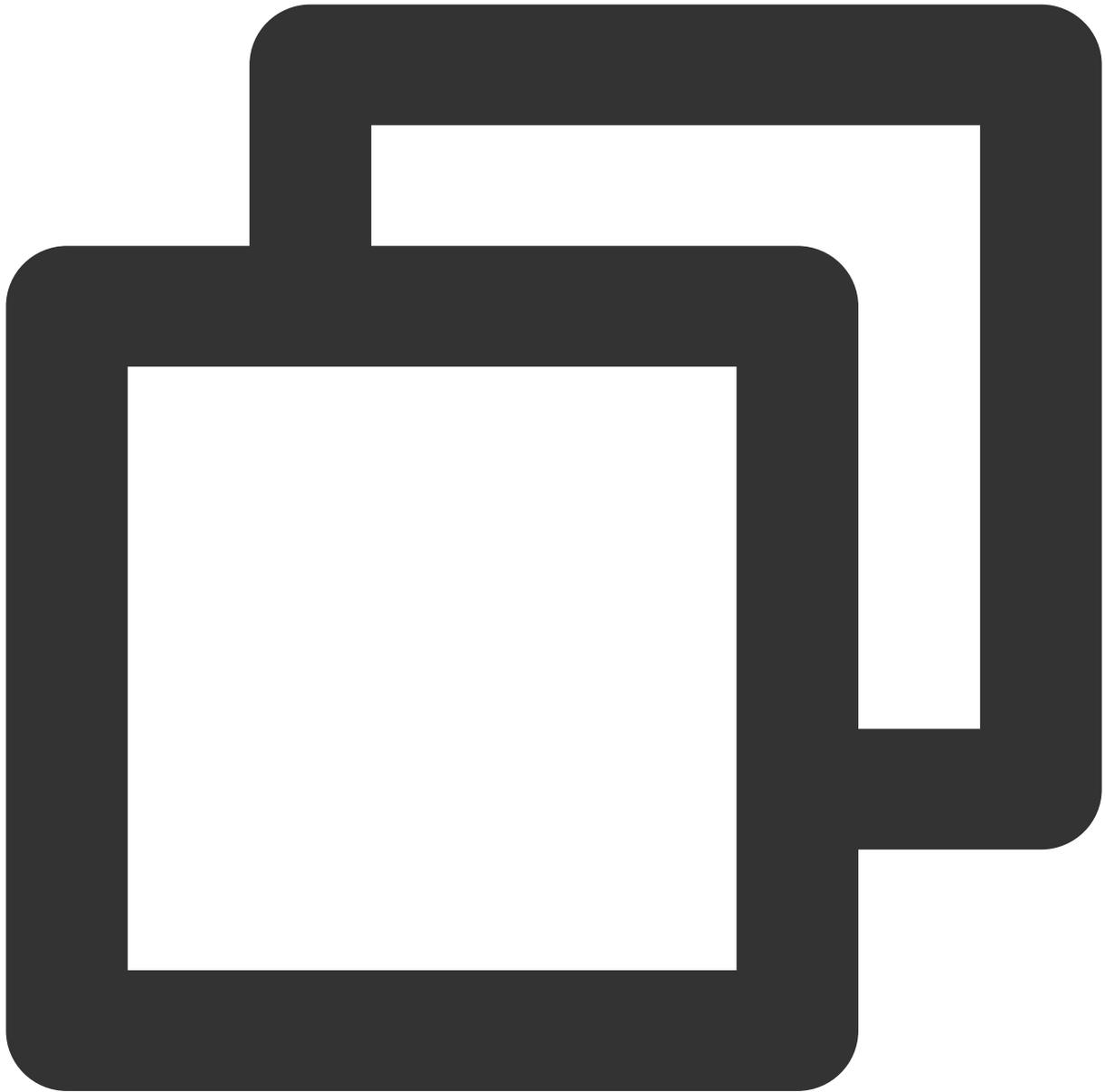
使用客户端加密功能时，您需要对主密钥的完整性和正确性负责。在对加密数据进行复制或者迁移时，您需要对加密元信息的完整性和正确性负责。因您维护不当导致主密钥用错或丢失，加密元信息出错或丢失，从而导致加密数据无法解密所引起的一切损失和后果均由您自行承担。

开启客户端加密后，不再支持 `append`、`truncate` 接口。

使用关闭了客户端加密功能的客户端对加密文件进行 `hadoop fs -cp` 命令，会丢失加密信息。

开启客户端加密后，默认关闭 CRC 文件校验，默认关闭异步文件分块上传。

当使用 Hadoop-COS 时，用户可以在 `$HADOOP_HOME/etc/hadoop/core-site.xml` 文件中，增加以下配置来进行实现 SSE-COS 加密。

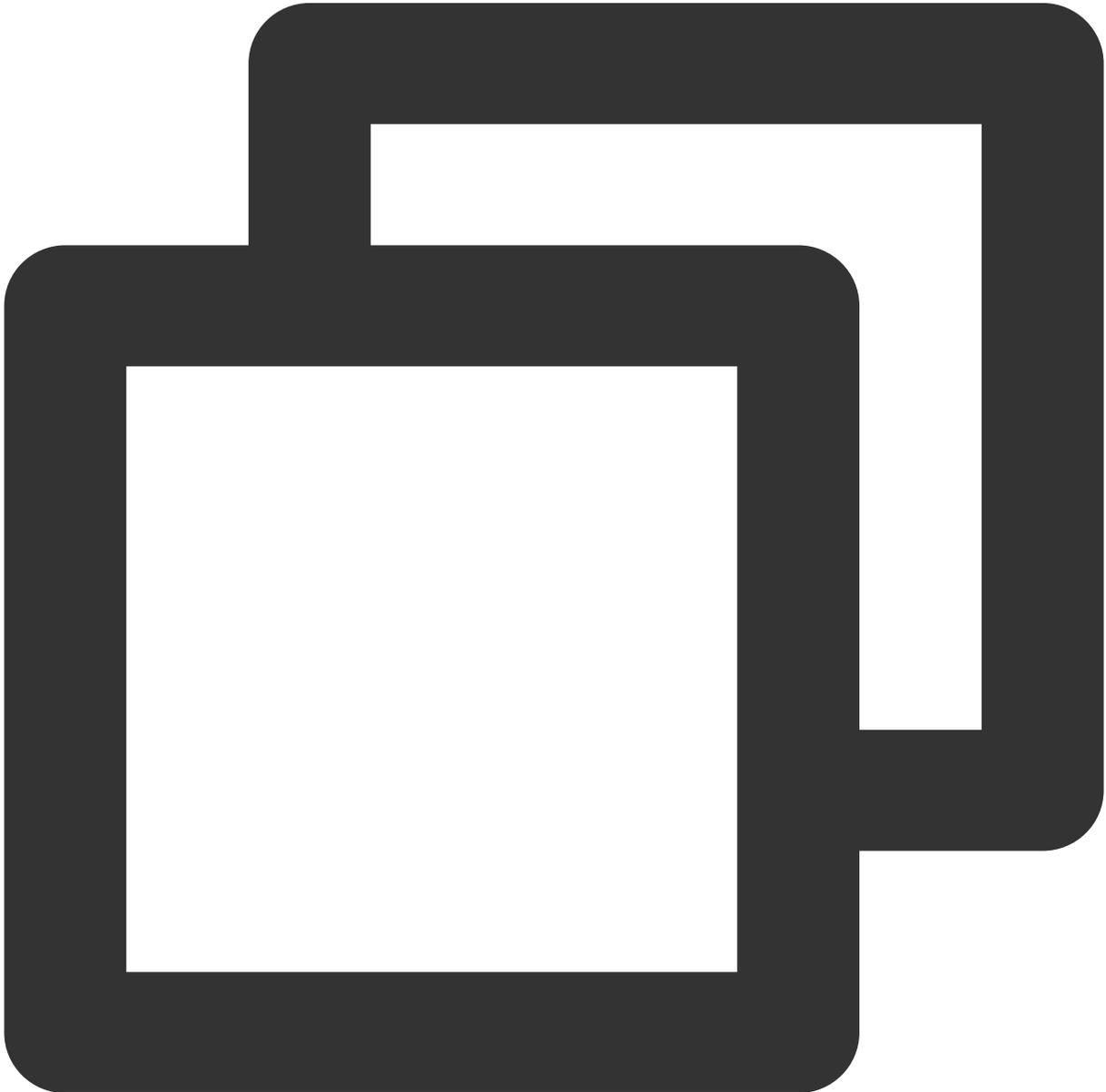


```
<property>
  <name>fs.cosn.client-side-encryption.enabled</name>
  <value>true</value>
  <description>Enable or disable the client encryption function</description>
</property>

<property>
  <name>fs.cosn.client-side-encryption.public.key.path</name>
  <value>/xxx/xxx.key</value>
  <description>The direct path to the public key</description>
</property>
```

```
<property>
  <name>fs.cosn.client-side-encryption.private.key.path</name>
  <value>/xxx/xxx.key</value>
  <description>The direct path to the private key</description>
</property>
```

可使用以下代码生成密钥：



```
import java.io.FileOutputStream;
import java.io.IOException;
import java.security.KeyPair;
```

```
import java.security.KeyPairGenerator;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;

// 使用非对称密钥 RSA 加密每次生成的随机对称密钥
public class BuildKey {
    private static final SecureRandom srand = new SecureRandom();
    private static void buildAndSaveAsymKeyPair(String pubKeyPath, String priKeyPath) {
        KeyPairGenerator keyGenerator = KeyPairGenerator.getInstance("RSA");
        keyGenerator.initialize(1024, srand);
        KeyPair keyPair = keyGenerator.generateKeyPair();
        PrivateKey privateKey = keyPair.getPrivate();
        PublicKey publicKey = keyPair.getPublic();

        X509EncodedKeySpec x509EncodedKeySpec = new X509EncodedKeySpec(publicKey.getEncoded());
        FileOutputStream fos = new FileOutputStream(pubKeyPath);
        fos.write(x509EncodedKeySpec.getEncoded());
        fos.close();

        PKCS8EncodedKeySpec pkcs8EncodedKeySpec = new PKCS8EncodedKeySpec(privateKey.getEncoded());
        fos = new FileOutputStream(priKeyPath);
        fos.write(pkcs8EncodedKeySpec.getEncoded());
        fos.close();
    }

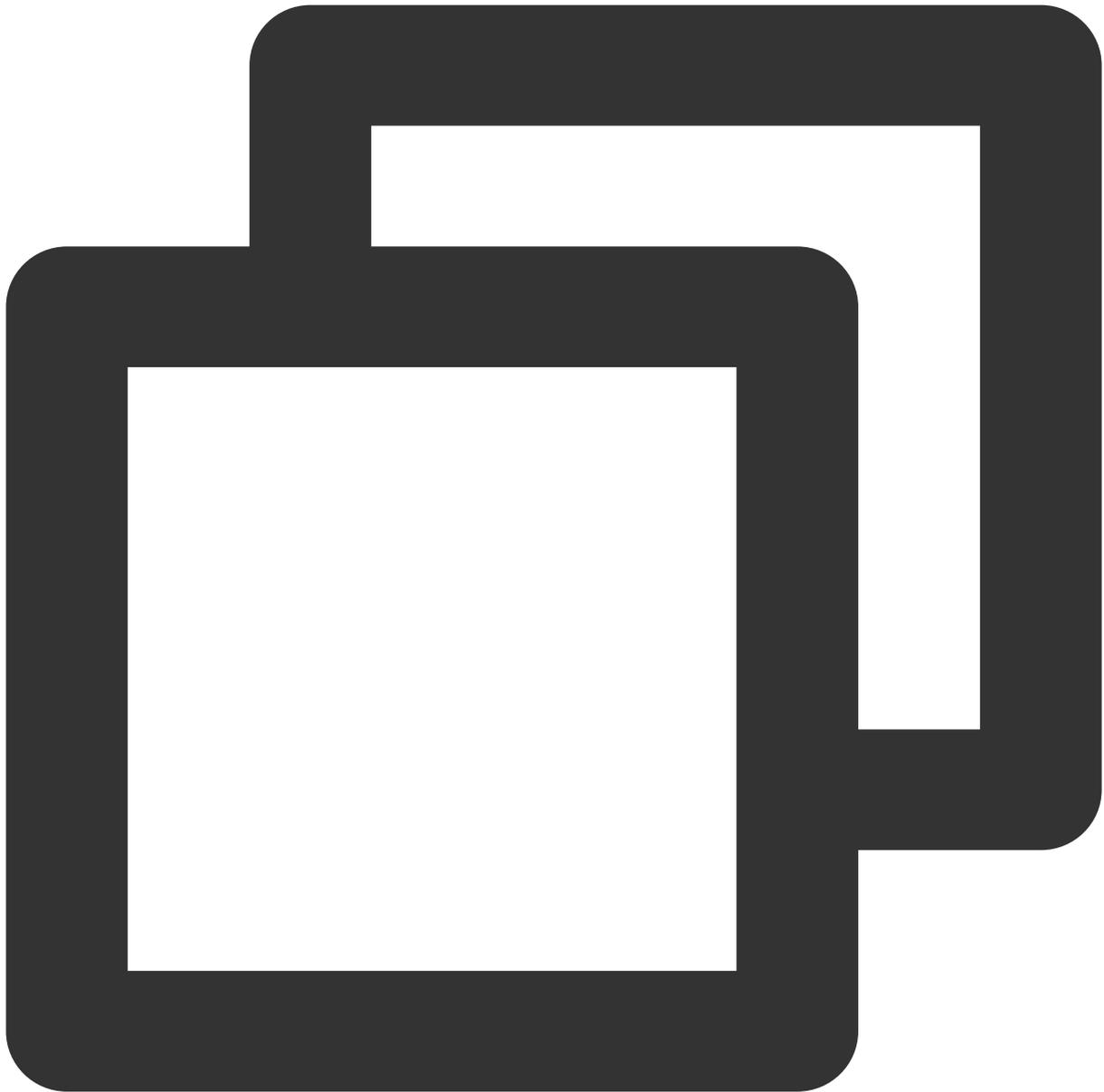
    public static void main(String[] args) throws Exception {

        String pubKeyPath = "pub.key";
        String priKeyPath = "pri.key";
        buildAndSaveAsymKeyPair(pubKeyPath, priKeyPath);
    }
}
```

使用方法

使用示例

命令格式为 `hadoop fs -ls -R cosn://<BucketName-APPID>/<路径>`，或 `hadoop fs -ls -R /<路径>`（需要配置 `fs.defaultFS` 选项为 `cosn://BucketName-APPID`），下例中以名称为 `examplebucket-1250000000` 的 bucket 为例，可在其后面加上具体路径。



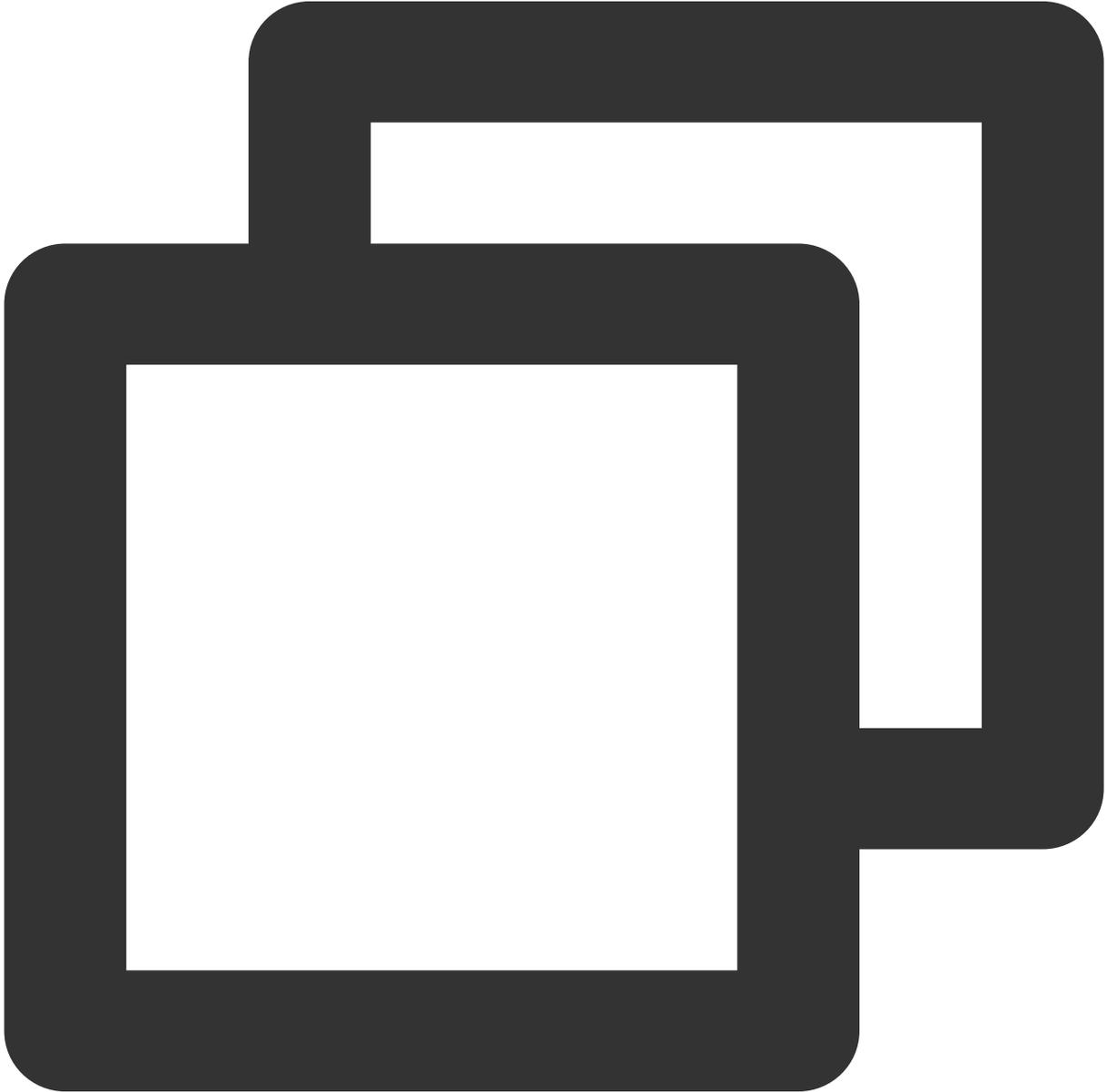
```
hadoop fs -ls -R cosn://examplebucket-1250000000/
-rw-rw-rw-  1 root root      1087 2018-06-11 07:49 cosn://examplebucket-125000000
drwxrwxrwx  - root root         0 1970-01-01 00:00 cosn://examplebucket-125000000
drwxrwxrwx  - root root         0 1970-01-01 00:00 cosn://examplebucket-125000000
-rw-rw-rw-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

```
-rw-rw-rw- 1 root root 1087 2018-06-11 07:32 cosn://examplebucket-125000000
-rw-rw-rw- 1 root root 2386 2018-06-11 07:29 cosn://examplebucket-125000000
```

运行 MapReduce 自带的 wordcount，执行以下命令。

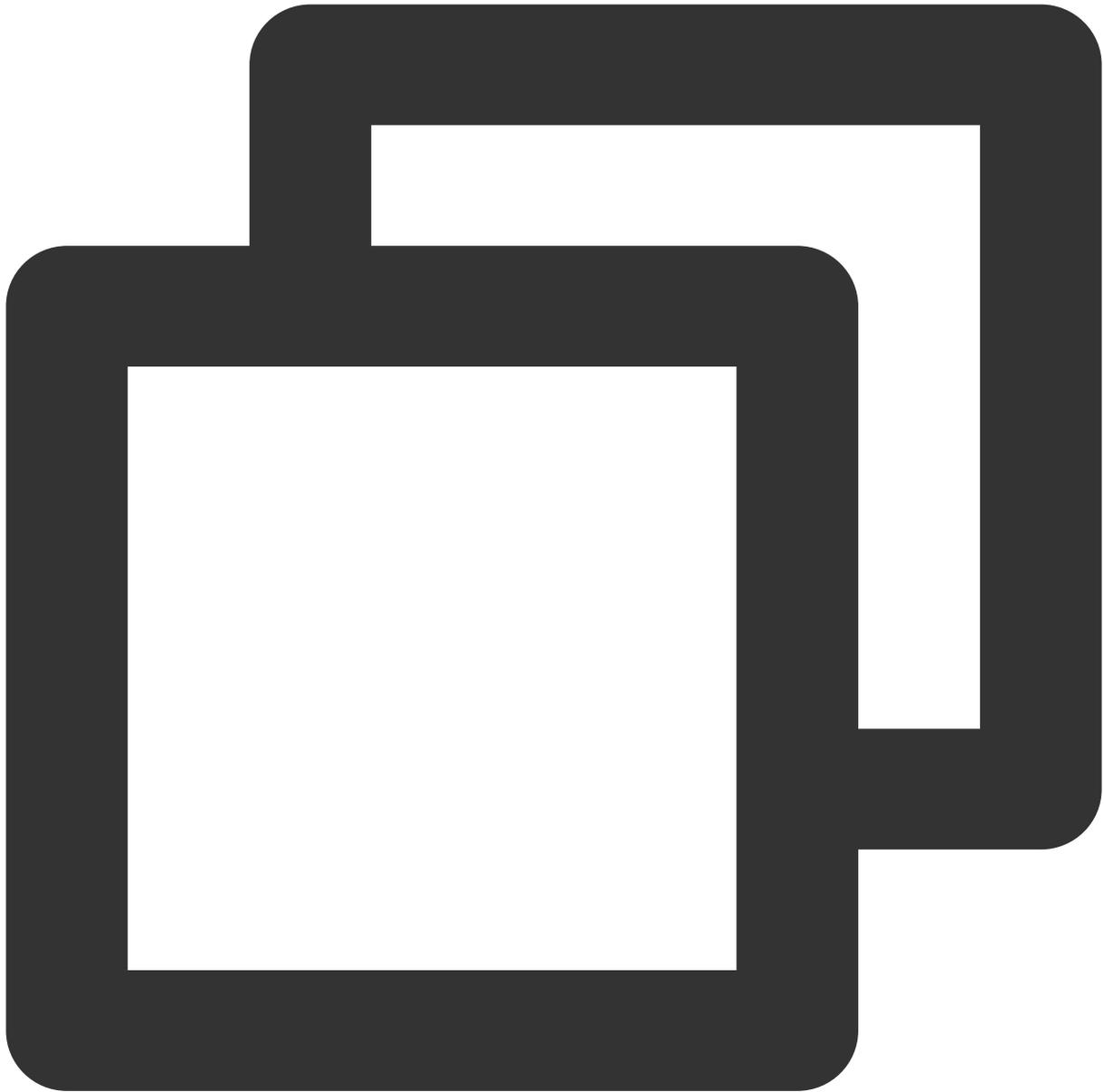
注意

以下命令中 `hadoop-mapreduce-examples-2.7.2.jar` 是以2.7.2版本为例，若版本不同，请修改成对应的版本号。



```
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount
```

执行成功会返回统计信息，示例如下：

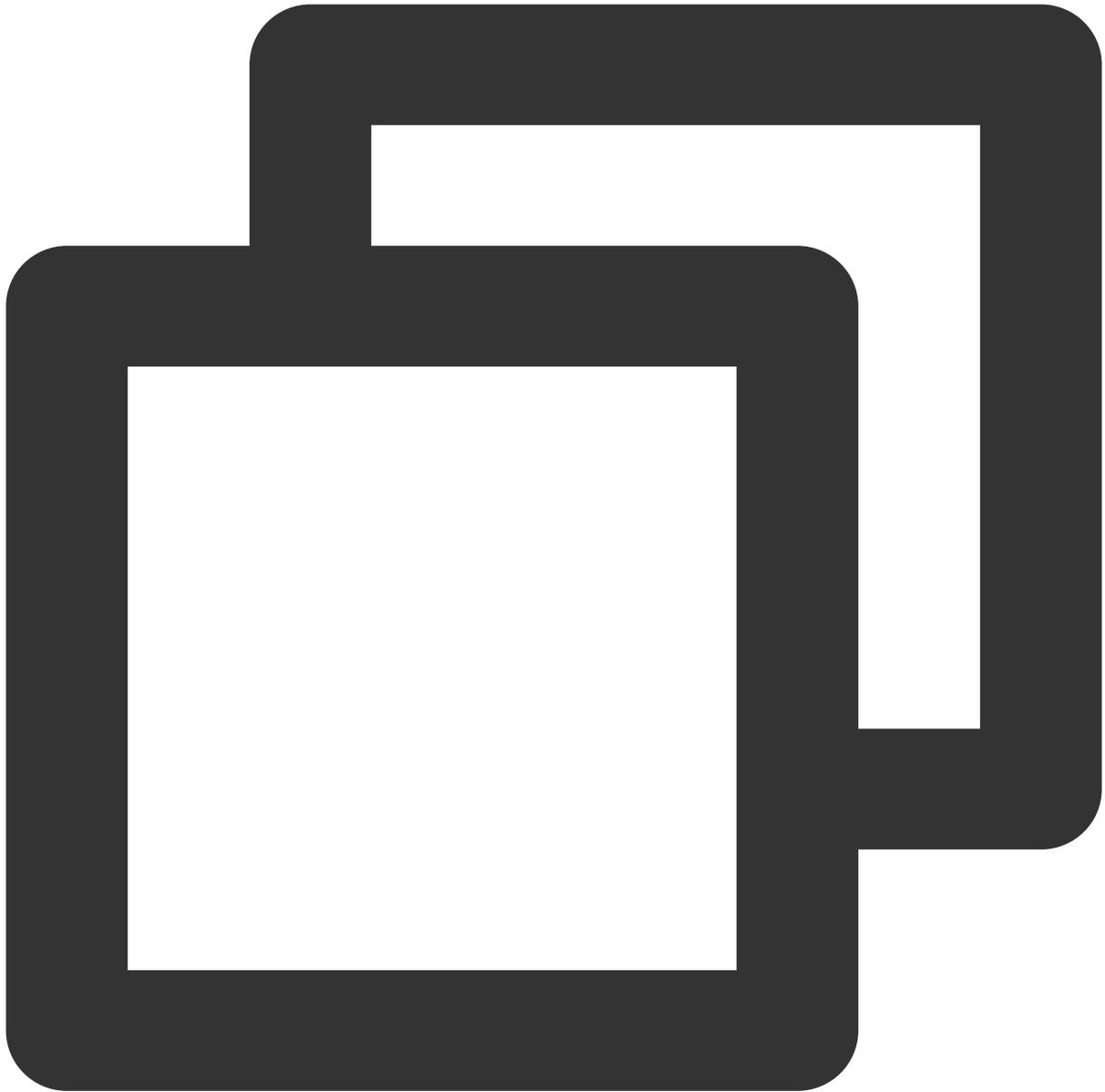


File System Counters

```
COSN: Number of bytes read=72
COSN: Number of bytes written=40
COSN: Number of read operations=0
COSN: Number of large read operations=0
COSN: Number of write operations=0
FILE: Number of bytes read=547350
FILE: Number of bytes written=1155616
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
```

```
HDFS: Number of bytes read=0
HDFS: Number of bytes written=0
HDFS: Number of read operations=0
HDFS: Number of large read operations=0
HDFS: Number of write operations=0
Map-Reduce Framework
Map input records=5
Map output records=7
Map output bytes=59
Map output materialized bytes=70
Input split bytes=99
Combine input records=7
Combine output records=6
Reduce input groups=6
Reduce shuffle bytes=70
Reduce input records=6
Reduce output records=6
Spilled Records=12
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=0
Total committed heap usage (bytes)=653262848
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=36
File Output Format Counters
Bytes Written=40
```

通过 Java 代码访问 COSN



```
package com.qcloud.chdfs.demo;

import org.apache.commons.io.IOUtils;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FSDataOutputStream;
import org.apache.hadoop.fs.FileChecksum;
import org.apache.hadoop.fs.FileStatus;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
```

```
import java.io.IOException;
import java.net.URI;
import java.nio.ByteBuffer;

public class Demo {
    private static FileSystem initFS() throws IOException {
        Configuration conf = new Configuration();
        // COSN 的配置项可参见 https://cloud.tencent.com/document/product/436/6884#ha
        // 以下配置是必填项
        conf.set("fs.cosn.impl", "org.apache.hadoop.fs.CosFileSystem");
        conf.set("fs.AbstractFileSystem.cosn.impl", "org.apache.hadoop.fs.CosN");
        conf.set("fs.cosn.tmp.dir", "/tmp/hadoop_cos");
        conf.set("fs.cosn.bucket.region", "ap-guangzhou");
        conf.set("fs.cosn.userinfo.secretId", "AKXXXXXXXXXXXXXXXXXXXX");
        conf.set("fs.cosn.userinfo.secretKey", "XXXXXXXXXXXXXXXXXXXX");
        conf.set("fs ofs.user.appid", "XXXXXXXXXXXX");
        // 其他配置参考官网文档 https://cloud.tencent.com/document/product/436/6884#ha
        // 是否开启 CRC64 校验。默认不开启, 此时无法使用 hadoop fs -checksum 命令获取文件的
        conf.set("fs.cosn.crc64.checksum.enabled", "true");
        String cosnUrl = "cosn://f4mxxxxxxxx-125xxxxxxxx";
        return FileSystem.get(URI.create(cosnUrl), conf);
    }

    private static void mkdir(FileSystem fs, Path filePath) throws IOException {
        fs.mkdirs(filePath);
    }

    private static void createFile(FileSystem fs, Path filePath) throws IOException {
        // 创建一个文件 (如果存在则将其覆盖)
        // if the parent dir does not exist, fs will create it!
        FSDataOutputStream out = fs.create(filePath, true);
        try {
            // 写入一个文件
            String content = "test write file";
            out.write(content.getBytes());
        } finally {
            IOUtils.closeQuietly(out);
        }
    }

    private static void readFile(FileSystem fs, Path filePath) throws IOException {
        FSDataInputStream in = fs.open(filePath);
        try {
            byte[] buf = new byte[4096];
            int readLen = -1;
            do {
                readLen = in.read(buf);
            }
        }
    }
}
```

```
        } while (readLen >= 0);
    } finally {
        IOUtils.closeQuietly(in);
    }
}

private static void queryFileOrDirStatus(FileSystem fs, Path path) throws IOExc
    FileStatus fileStatus = fs.getFileStatus(path);
    if (fileStatus.isDirectory()) {
        System.out.printf("path %s is dir\\n", path);
        return;
    }
    long fileLen = fileStatus.getLen();
    long accessTime = fileStatus.getAccessTime();
    long modifyTime = fileStatus.getModificationTime();
    String owner = fileStatus.getOwner();
    String group = fileStatus.getGroup();

    System.out.printf("path %s is file, fileLen: %d, accessTime: %d, modifyTime
        path, fileLen, accessTime, modifyTime, owner, group);
}

private static void getFileChecksum(FileSystem fs, Path path) throws IOExceptio
    FileChecksum checksum = fs.getFileChecksum(path);
    System.out.printf("path %s, checksumType: %s, checksumCrcVal: %d\\n",
        path, checksum.getAlgorithmName(), ByteBuffer.wrap(checksum.getByte

}

private static void copyFileFromLocal(FileSystem fs, Path cosnPath, Path localP
    fs.copyFromLocalFile(localPath, cosnPath);
}

private static void copyFileToLocal(FileSystem fs, Path cosnPath, Path localPat
    fs.copyToLocalFile(cosnPath, localPath);
}

private static void renamePath(FileSystem fs, Path oldPath, Path newPath) throw
    fs.rename(oldPath, newPath);
}

private static void listDirPath(FileSystem fs, Path dirPath) throws IOExceptio
    FileStatus[] dirMemberArray = fs.listStatus(dirPath);

    for (FileStatus dirMember : dirMemberArray) {
        System.out.printf("dirMember path %s, fileLen: %d\\n", dirMember.getPat
    }
}
```

```
// 递归删除标志用于删除目录
// 如果递归为 false 并且 dir 不为空, 则操作将失败
private static void deleteFileOrDir(FileSystem fs, Path path, boolean recursive) {
    fs.delete(path, recursive);
}

private static void closeFileSystem(FileSystem fs) throws IOException {
    fs.close();
}

public static void main(String[] args) throws IOException {
    // 初始化文件
    FileSystem fs = initFS();

    // 创建文件
    Path cosnFilePath = new Path("/folder/exampleobject.txt");
    createFile(fs, cosnFilePath);

    // 读取文件
    readFile(fs, cosnFilePath);

    // 查询文件或目录
    queryFileOrDirStatus(fs, cosnFilePath);

    // 获取文件校验和
    getFileChecksum(fs, cosnFilePath);

    // 从本地复制文件
    Path localFilePath = new Path("file:///home/hadoop/ofs_demo/data/exampleobj");
    copyFileFromLocal(fs, cosnFilePath, localFilePath);

    // 获取文件到本地
    Path localDownFilePath = new Path("file:///home/hadoop/ofs_demo/data/exempl");
    copyFileToLocal(fs, cosnFilePath, localDownFilePath);

    listDirPath(fs, cosnFilePath);
    // 重命名
    mkdir(fs, new Path("/doc"));
    Path newPath = new Path("/doc/example.txt");
    renamePath(fs, cosnFilePath, newPath);

    // 删除文件
    deleteFileOrDir(fs, newPath, false);

    // 创建目录
    Path dirPath = new Path("/folder");
```

```
mkdir(fs, dirPath);

// 在目录中创建文件
Path subFilePath = new Path("/folder/exampleobject.txt");
createFile(fs, subFilePath);

// 列出目录
listDirPath(fs, dirPath);

// 删除目录
deleteFileOrDir(fs, dirPath, true);
deleteFileOrDir(fs, new Path("/doc"), true);

// 关闭文件系统
closeFileSystem(fs);
}
}
```

常见问题

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

COSDistCp 工具

最近更新时间：2024-01-06 10:43:44

功能说明

COSDistCp 是一款基于 MapReduce 的分布式文件拷贝工具，主要用于 HDFS 和 COS 之间的数据拷贝，它主要具有以下功能点：

根据长度、CRC 校验和，进行文件的增量迁移、数据校验

对源目录中的文件进行正则表达式过滤

对源目录中的文件进行解压缩，并转换为预期的压缩格式

基于正则表达式，对文本文件进行聚合

保留源文件和源目录的用户、组、扩展属性和时间

配置告警和 Prometheus 监控

统计文件大小分布

对读取带宽进行限速

使用环境

系统环境

支持 Linux 系统。

软件依赖

Hadoop-2.6.0及以上版本、Hadoop-COS 插件 5.9.3 及以上版本。

下载与安装

获取 COSDistCp jar 包

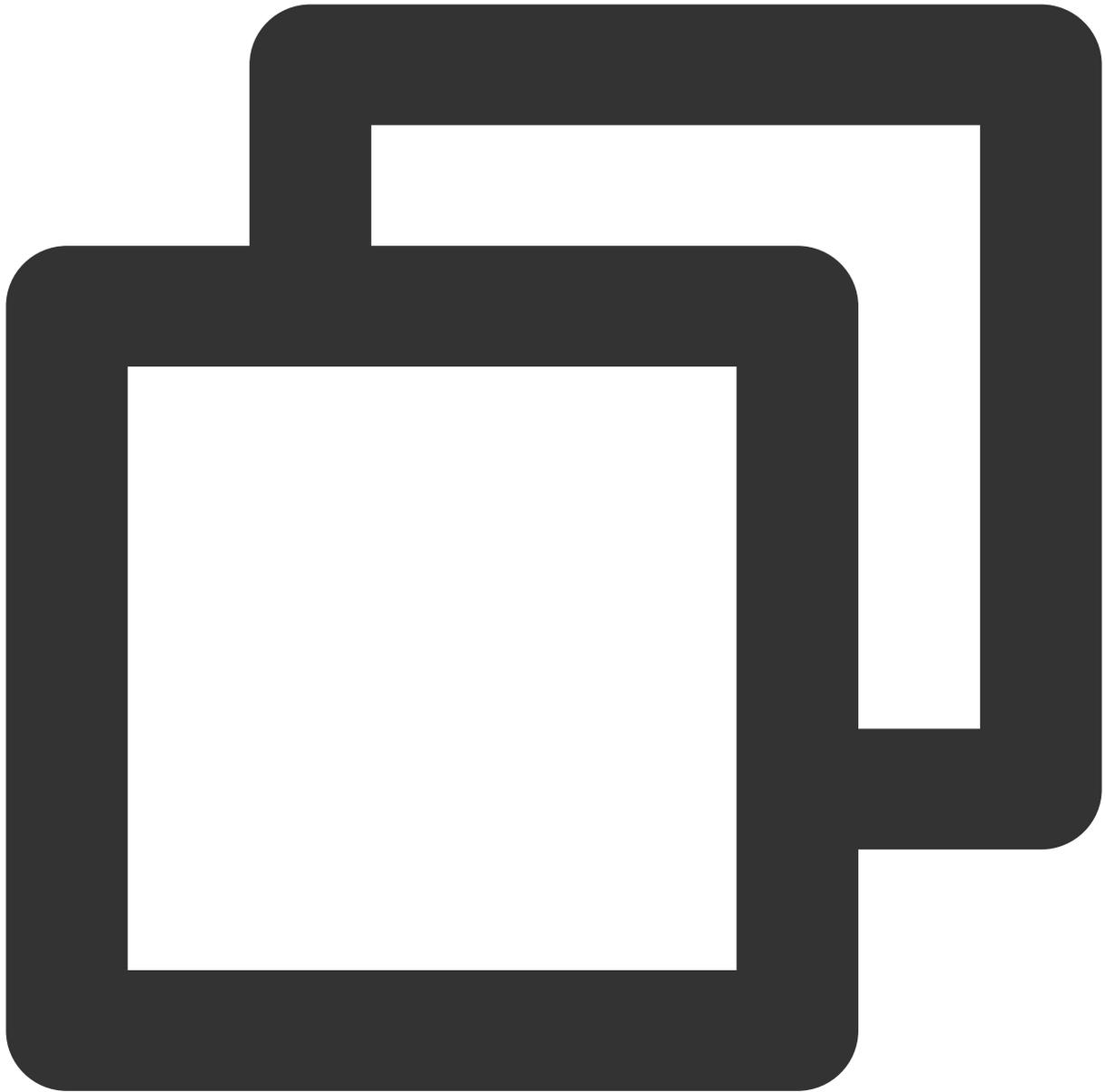
Hadoop 2.x 用户可下载 [cos-distcp-1.13-2.8.5.jar 包](#)，根据 jar 包的 MD5 校验值 确认下载的 jar 包是否完整。

Hadoop 3.x 用户可下载 [cos-distcp-1.13-3.1.0.jar 包](#)，根据 jar 包的 MD5 校验值 确认下载的 jar 包是否完整。

安装说明

在 Hadoop 环境下，安装 [Hadoop-COS](#) 后，即可直接运行 COSDistCp 工具。

对于环境中未安装和配置 Hadoop-COS 插件的用户，根据 Hadoop 版本，下载对应版本的 COSDistCp jar、Hadoop-COS jar 和 cos_api-bundle jar 包后(相关 jar 包下载地址见上文)，指定 Hadoop-COS 相关参数执行拷贝任务，其中 jar 包地址需填本地 jar 所在地址：



```
hadoop jar cos-distcp-${version}.jar \<\  
-libjars cos_api-bundle-${version}.jar,hadoop-cos-${version}.jar \<\  
-Dfs.cosn.credentials.provider=org.apache.hadoop.fs.auth.SimpleCredentialProvider \<\  
-Dfs.cosn.userinfo.secretId=COS_SECRETID \<\  
-Dfs.cosn.userinfo.secretKey=COS_SECRETKEY \<\  
-Dfs.cosn.bucket.region=ap-guangzhou \<\  
-Dfs.cosn.impl=org.apache.hadoop.fs.CosFileSystem \<\  
-Dfs.AbstractFileSystem.cosn.impl=org.apache.hadoop.fs.CosN \<\  
--src /data/warehouse \<\  
--dest cosn://examplebucket-1250000000/warehouse
```

原理说明

COSDistCp 基于 MapReduce 框架实现，为多进程+多线程的架构，可以对文件进行拷贝、数据校验、压缩、文件属性保留以及拷贝重试等工作。COSDistCp 默认会覆盖目标端已经存在的同名文件，当文件迁移或校验失败的时候，对应的文件会拷贝失败，并会在临时目录下记录迁移失败的文件信息。当您的源目录有文件新增或文件内容发生变化时，您可通过 skipMode 或 diffMode 模式，通过对比文件的长度或 CRC 校验值，进行数据校验和进行文件的增量迁移。

参数说明

您可在hadoop用户下，使用命令 `hadoop jar cos-distcp-${version}.jar --help` 查看 COSDistCp 支持的参数选项，其中 `${version}` 为版本号，以下为当前版本 COSDistCp 的参数说明：

属性键	说明	默认值
<code>--help</code>	输出 COSDistCp 支持的参数选项 示例： <code>--help</code>	无
<code>--src=LOCATION</code>	指定拷贝的源目录，可以是 HDFS 或者 COS 路径 示例： <code>--src=hdfs://user/logs/</code>	无
<code>--dest=LOCATION</code>	指定拷贝的目标目录，可以是 HDFS 或者 COS 路径 示例： <code>--dest=cosn://examplebucket-1250000000/user/logs</code>	无
<code>--srcPattern=PATTERN</code>	指定正则表达式对源目录中的文件进行过滤 示例： <code>--srcPattern='.*\\.log\$'</code> 注意：您需要将参数使用单引号包围，以避免符号 * 被 shell 解释	无
<code>--taskNumber=VALUE</code>	指定拷贝进程数，示例： <code>--taskNumber=10</code>	10
<code>--workerNumber=VALUE</code>	指定拷贝线程数，COSDistCp 在每个拷贝进程中创建该参数大小的拷贝线程池 示例： <code>--workerNumber=4</code>	4
<code>--filesPerMapper=VALUE</code>	指定每个 Mapper 输入文件的行数 示例： <code>--filesPerMapper=10000</code>	500000

<code>--groupBy=PATTERN</code>	指定正则表达式对文本文件进行聚合 示例： <code>--groupBy='.*group-input/(\d+)-(\d+).*'</code>	无
<code>--targetSize=VALUE</code>	指定目标文件的大小，单位：MB，与 <code>--groupBy</code> 一起使用 示例： <code>--targetSize=10</code>	无
<code>--outputCodec=VALUE</code>	指定输出文件的压缩方式，可选 <code>gzip</code> 、 <code>lzo</code> 、 <code>snappy</code> 、 <code>none</code> 和 <code>keep</code> ，其中： 1. <code>keep</code> 保持原有文件的压缩方式 1. <code>none</code> 则根据文件后缀对文件进行解压，示例： <code>--outputCodec=gzip</code> 注意：如果存在文件 <code>/dir/test.gzip</code> 和 <code>/dir/test.gz</code>，指定输出格式为 <code>lzo</code>，最终只会保留一个文件 <code>/dir/test.lzo</code>	<code>keep</code>
<code>--deleteOnSuccess</code>	指定源文件拷贝到目标目录成功时，立即删除源文件 示例： <code>--deleteOnSuccess</code> ， 注意：1.7 及以上版本不再提供该参数，建议数据迁移成功并使用 <code>--diffMode</code> 校验后，再删除源文件系统的数据	<code>false</code>
<code>--multipartUploadChunkSize=VALUE</code>	指定 Hadoop-COS 插件传输文件到 COS 时分块的大小，COS 支持的最大分块数为 10000，您可根据文件大小，调整分块大小，单位：MB，默认为 8MB 示例： <code>--multipartUploadChunkSize=20</code>	8MB
<code>--cosServerSideEncryption</code>	指定文件上传到 COS 时，使用 SSE-COS 作为加解密算法 示例： <code>--cosServerSideEncryption</code>	<code>false</code>
<code>--outputManifest=VALUE</code>	指定拷贝完成的时候，在目标目录下生成本次拷贝的目标文件信息列表（GZIP 压缩） 示例： <code>--outputManifest=manifest.gz</code>	无
<code>--requirePreviousManifest</code>	要求指定 <code>--previousManifest=VALUE</code> 参数，以进行增量拷贝 示例： <code>--requirePreviousManifest</code>	<code>false</code>
<code>--previousManifest=LOCATION</code>	前一次拷贝生成的目标文件信息 示例： <code>--previousManifest=cosn://examplebucket-1250000000/big-data/manifest.gz</code>	无
<code>--copyFromManifest</code>	和 <code>--previousManifest=LOCATION</code> 一起使	<code>false</code>

	<p>用，可将 <code>--previousManifest</code> 中的文件，拷贝到目标文件系统</p> <p>示例：<code>--copyFromManifest</code></p>	
<code>--storageClass=VALUE</code>	<p>指定对象存储类型，可选值为 <code>STANDARD</code>、<code>STANDARD_IA</code>、<code>ARCHIVE</code>、<code>DEEP_ARCHIVE</code>、<code>INTELLIGENT_TIERING</code>，关于更多支持的存储类型和介绍，请参见 存储类型概述</p>	无
<code>--srcPrefixesFile=LOCATION</code>	<p>指定本地文件，该文件中每行包含一个需要拷贝的源目录</p> <p>示例：<code>--srcPrefixesFile=file:///data/migrate-folders.txt</code></p>	无
<code>--skipMode=MODE</code>	<p>拷贝文件前，校验源文件和目标文件是否相同，相同则跳过，可选 <code>none</code>（不校验）、<code>length</code>（长度）、<code>checksum</code>（CRC值）、<code>length-mtime</code>（长度+mtime值）和 <code>length-checksum</code>（长度 + CRC 值）</p> <p>示例：<code>--skipMode=length</code></p>	<code>length-checksum</code>
<code>--checkMode=MODE</code>	<p>当文件拷贝完成的时候，校验源文件和目标文件是否相同，可选 <code>none</code>（不校验）、<code>length</code>（长度）、<code>checksum</code>（CRC值）、<code>length-mtime</code>（长度+mtime值）和 <code>length-checksum</code>（长度 + CRC 值）</p> <p>示例：<code>--checkMode=length-checksum</code></p>	<code>length-checksum</code>
<code>--diffMode=MODE</code>	<p>指定获取源和目的目录的差异文件列表，可选 <code>length</code>（长度）、<code>checksum</code>（CRC 值）、<code>length-mtime</code>（长度+mtime值）和 <code>length-checksum</code>（长度 + CRC 值）</p> <p>示例：<code>--diffMode=length-checksum</code></p>	无
<code>--diffOutput=LOCATION</code>	<p>指定 <code>diffMode</code> 的 HDFS 输出目录，该输出目录必须为空</p> <p>示例：<code>--diffOutput=/diff-output</code></p>	无
<code>--cosChecksumType=TYPE</code>	<p>指定 Hadoop-COS 插件使用的 CRC 算法，可选值为 <code>CRC32C</code> 和 <code>CRC64</code></p> <p>示例：<code>--cosChecksumType=CRC32C</code></p>	<code>CRC32C</code>
<code>--preserveStatus=VALUE</code>	<p>指定是否将源文件的 <code>user</code>、<code>group</code>、<code>permission</code>、<code>xattr</code> 和 <code>timestamps</code> 元信息拷贝到目标文件，可选值为 <code>ugpxt</code>（即为 <code>user</code>、</p>	无

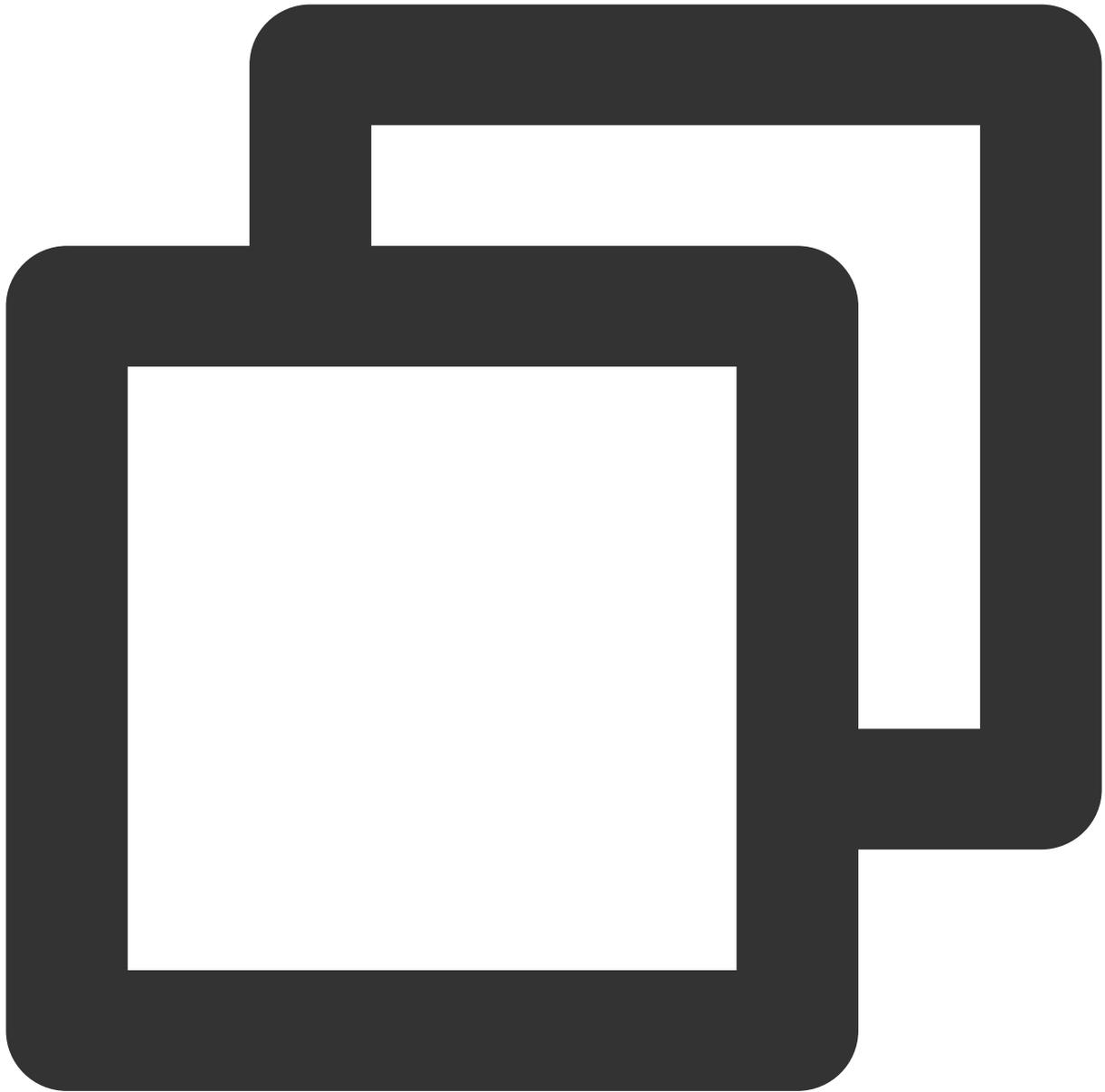
	group、permission、xattr 和 timestamps 的英文首字母) 示例：--preserveStatus=ugpt	
--ignoreSrcMiss	忽略存在于文件清单中，但拷贝时不存在的文件	false
--promGatewayAddress=VALUE	指定 MapReduce 任务运行的 Counter 数据推送到的 Prometheus PushGateway 的地址和端口	无
--promGatewayDeleteOnFinish=VALUE	指定任务完成时，删除 Prometheus PushGateway 中 JobName 的指标集合 示例：--promGatewayDeleteOnFinish=true	true
--promGatewayJobName=VALUE	指定上报给 Prometheus PushGateway 的 JobName 示例：--promGatewayJobName=cos-distcp-hive-backup	无
--promCollectInterval=VALUE	指定收集 MapReduce 任务 Counter 信息的间隔，单位：ms 示例：--promCollectInterval=5000	5000
--promPort=VALUE	指定将 Prometheus 指标暴露给外部的 Server 端口 示例：--promPort=9028	无
--enableDynamicStrategy	指定开启任务动态分配策略，使迁移速度快的任务迁移更多的文件。 注意：该模式存在一定局限性，例如任务计数器在进程异常的情况下计数不准确，请迁移完成后用 --diffMode 进行数据校验 示例：--enableDynamicStrategy	false
--splitRatio=VALUE	指定 Dynamic Strategy 的切分比例，splitRatio 值越大，则任务粒度越小 示例：--splitRatio=8	8
--localTemp=VALUE	指定 Dynamic Strategy 生成的任务信息文件所在的本地文件夹 示例：--localTemp=/tmp	/tmp
--taskFilesCopyThreadNum=VALUE	指定 Dynamic Strategy 任务信息文件拷贝到 HDFS 上的并发度 示例：--taskFilesCopyThreadNum=32	32

<code>--statsRange=VALUE</code>	指定统计的区间范围 示例： <code>---</code> <code>statsRange=0,1mb,10mb,100mb,1gb,10gb,inf</code>	0,1mb,10mb,100mb,
<code>--printStatsOnly</code>	只统计待迁移文件大小的分布信息，不迁移数据 示例： <code>--printStatsOnly</code>	无
<code>--bandWidth</code>	限制读取每个迁移文件的带宽，单位为： MB/s ，默认-1，不限制读取带宽。 示例： <code>--bandWidth=10</code>	无
<code>--jobName</code>	指定迁移任务的名称。 示例： <code>--jobName=cosdistcp-to-warehouse</code>	无
<code>--compareWithCompatibleSuffix</code>	使用 <code>--skipMode</code> 和 <code>--diffMode</code> 参数时，是否将源文件的后缀 <code>gzip</code> 转换为 <code>gz</code> ， <code>lzop</code> 文件后缀转换为 <code>lzo</code> ，进行判断。 示例： <code>--compareWithCompatibleSuffix</code>	无
<code>--delete</code>	保证源目录和目标目录文件的一致性，将源目录中没有而目标目录中有的文件，移动到独立的 <code>trash</code> 目录下，同时生成文件清单。 注意：不能同时使用 <code>--diffMode</code> 参数	无
<code>--deleteOutput</code>	指定 <code>delete</code> 的 HDFS 输出目录，该目录必须为空 示例： <code>--deleteOutput=/dele-output</code>	无

使用示例

查看 help 选项

以参数 `--help` 执行命令，查看 COSDistCp 支持的参数，示例如下：

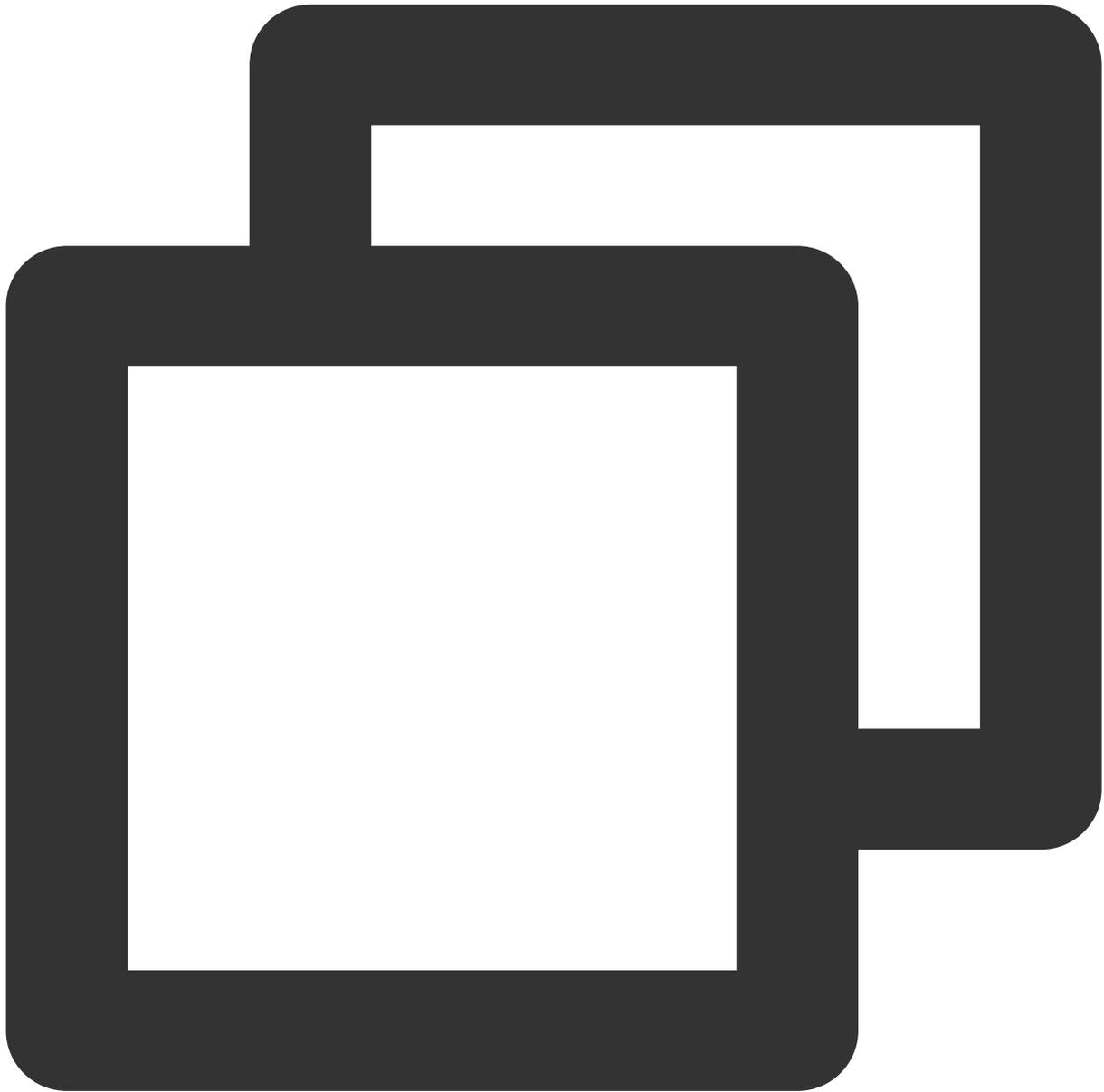


```
hadoop jar cos-distcp-${version}.jar --help
```

以上命令中，`${version}` 为 COSDistCp 版本号，例如 1.0 版本的 COSDistCp jar 包名为 `cos-distcp-1.0.jar`。

统计待迁移的文件大小分布信息

以参数 `--printStatsOnly` 和 `--statsRange=VALUE` 执行命令，输出待迁移文件的大小分布信息：



```
hadoop jar cos-distcp-${version}.jar --src /wookie/data --dest cosn://examplebucket
```

Copy File Distribution Statistics:

Total File Count: 4

Total File Size: 1190133760

SizeRange	TotalCount	TotalSize
0MB ~ 1MB	0 (0.00%)	0 (0.00%)
1MB ~ 10MB	1 (25.00%)	1048576 (0.09%)
10MB ~ 100MB	1 (25.00%)	10485760 (0.88%)
100MB ~ 1024MB	1 (25.00%)	104857600 (8.81%)
1024MB ~ 10240MB	1 (25.00%)	1073741824 (90.22%)

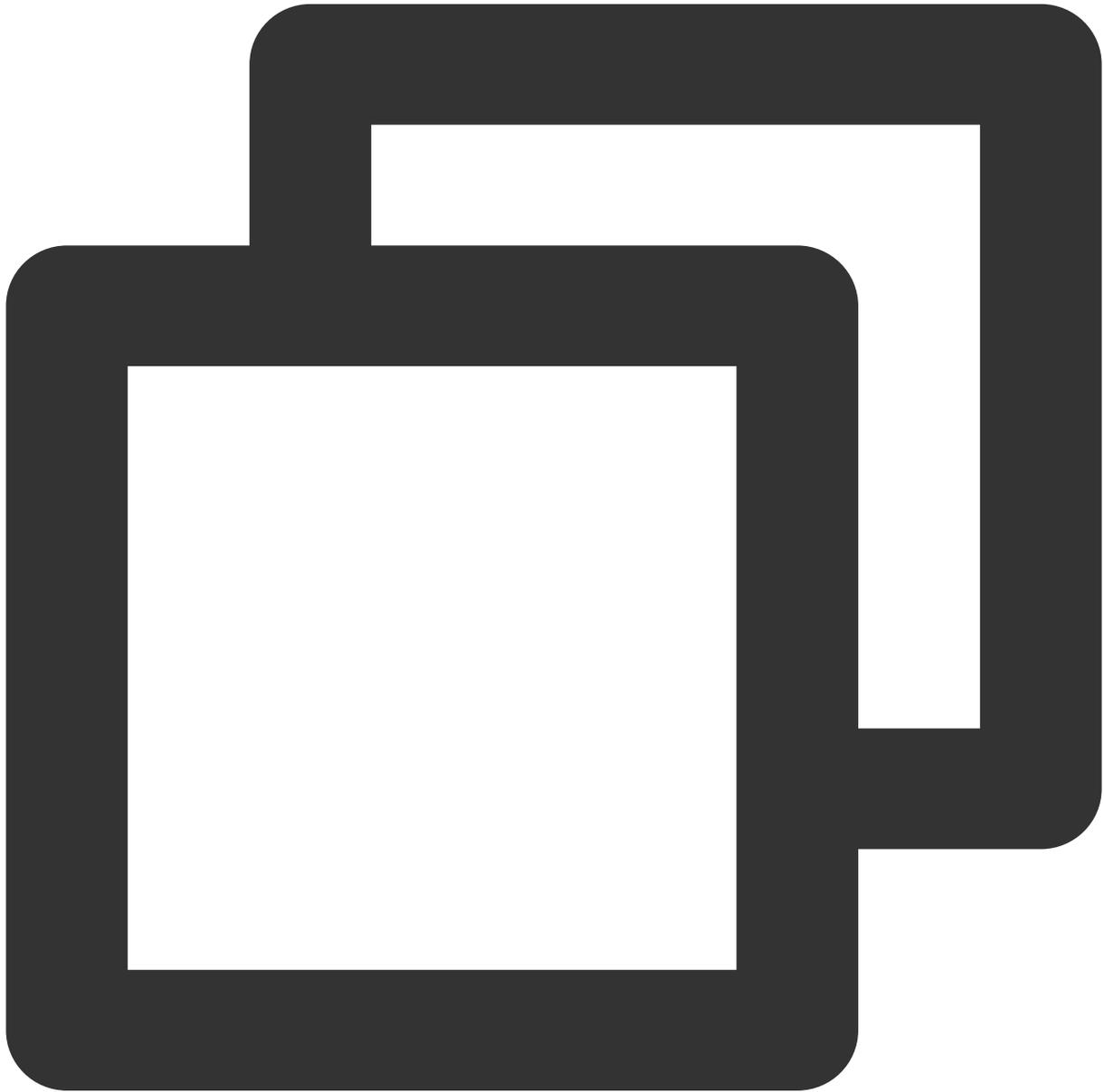
| 10240MB ~ LONG_MAX| 0(0.00%)

| 0(0.00%)

|

指定待迁移文件的源目录和目标目录

以参数 `--src` 和 `--dest` 执行命令，示例如下：



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

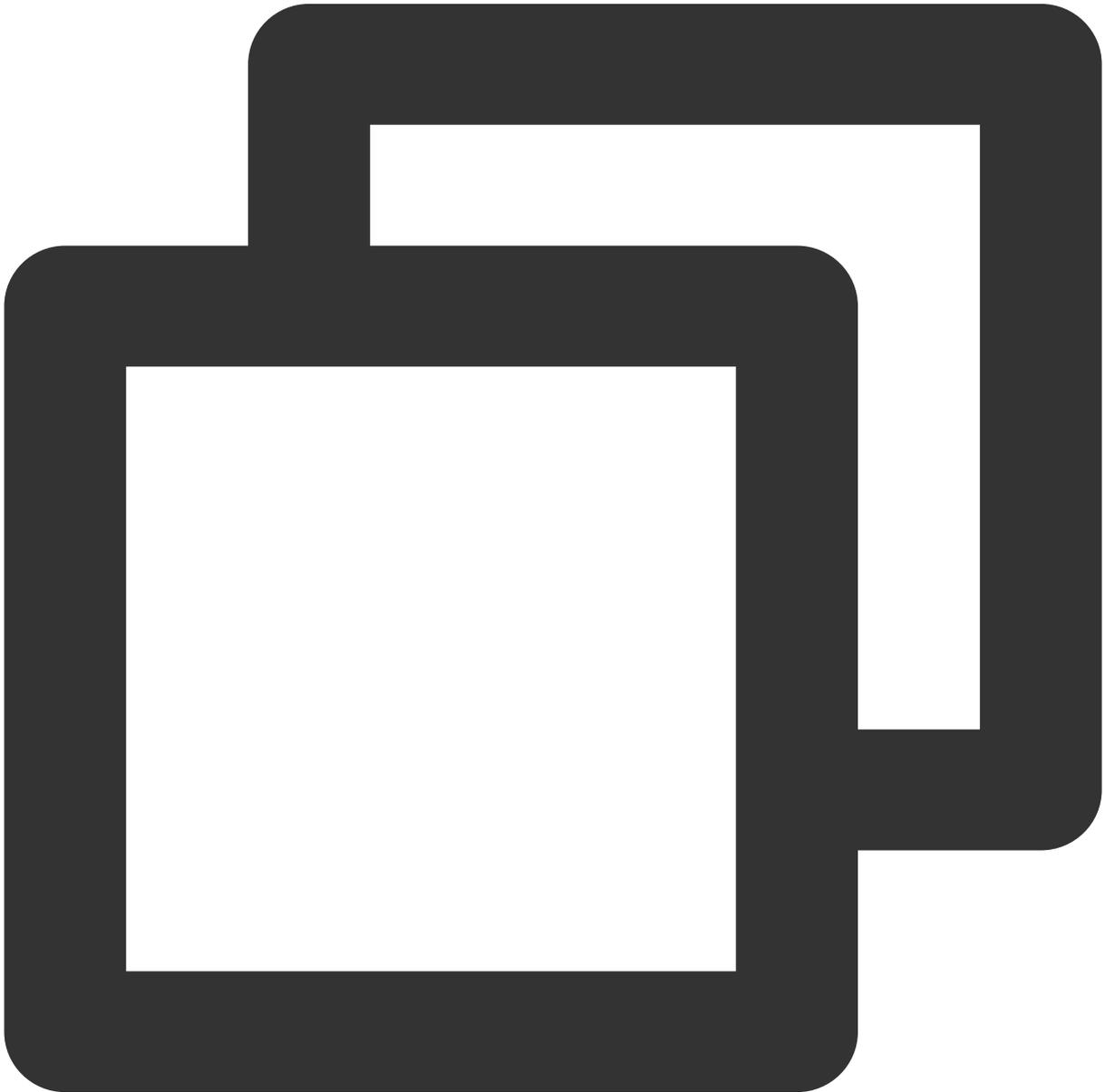
COSDistCp 默认会对拷贝失败的文件重试5次，如果仍然失败，则会将失败文件信息写入 `/tmp/${randomUUID}/output/failed/` 目录下，其中，`${randomUUID}` 为随机字符串。记录失败文件信息后，COSDistCp 会继续迁移剩余文件，迁移任务并不会因为部分文件迁移失败而失败。在迁移任务完成的时候，

COSDistcp 会输出计数器信息（请确保您的任务提交机器，配置了 MapReduce 任务的提交端 INFO 日志输出），并判断是否存在文件迁移失败，如果存在，则在提交任务的客户端抛出异常。

以下类型的源文件信息包含在输出文件中：

1. 存在源文件的清单中，但拷贝时源文件不存在，记录为 SRC_MISS
2. 其他原因导致的拷贝失败，统一记录为 COPY_FAILED

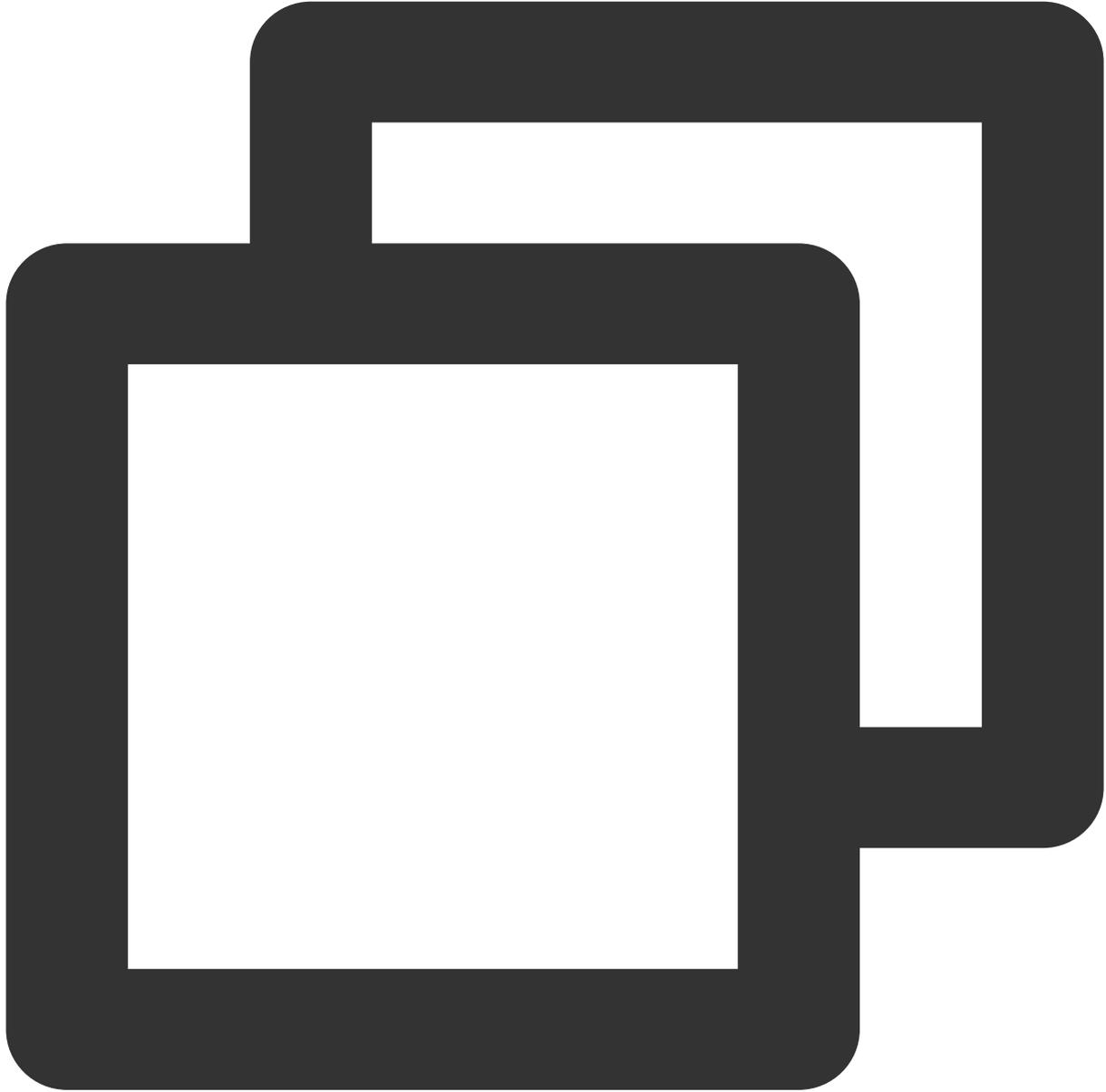
您可以重新运行拷贝命令实现增量迁移，通过如下命令，获取 MapReduce 任务的日志，确定文件拷贝失败的原因，其中 application_1610615435237_0021 为应用 ID：



```
yarn logs -applicationId application_1610615435237_0021 > application_1610615435237
```

查看 Counters

在拷贝任务结束时，会输出文件拷贝的统计信息，相关计数器如下：



CosDistCp Counters

```
BYTES_EXPECTED=10198247
BYTES_SKIPPED=10196880
FILES_COPIED=1
FILES_EXPECTED=7
FILES_FAILED=1
FILES_SKIPPED=5
```

文件拷贝的统计信息说明如下：

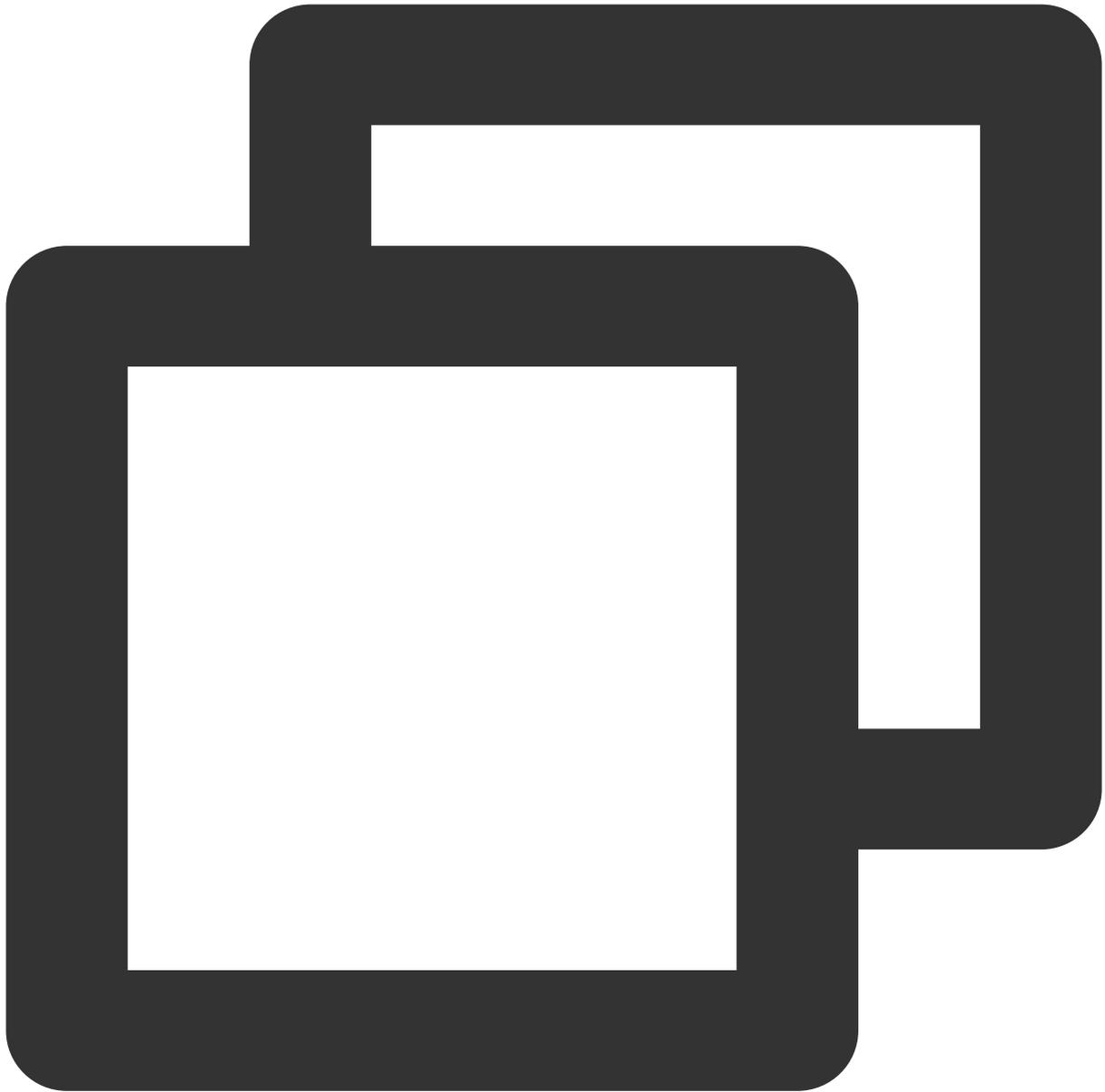
统计项	说明
BYTES_EXPECTED	根据源目录统计的需拷贝的文件总大小，单位：字节
FILES_EXPECTED	根据源目录统计的需拷贝文件数，包含目录文件
BYTES_SKIPPED	长度或校验和值相等，不拷贝的文件总大小，单位：字节
FILES_SKIPPED	长度或校验和值相等，不拷贝的源文件数
FILES_COPIED	拷贝成功的源文件数
FILES_FAILED	拷贝失败的源文件数
FOLDERS_COPIED	拷贝成功的目录数
FOLDERS_SKIPPED	跳过的目录数

指定拷贝进程数以及每个拷贝进程内的拷贝线程数

以参数 `--taskNumber` 和 `--workersNumber` 执行命令，COSDistCp 采用多进程+多线程的拷贝架构，您可以：

通过 `--taskNumber` 指定拷贝进程数目

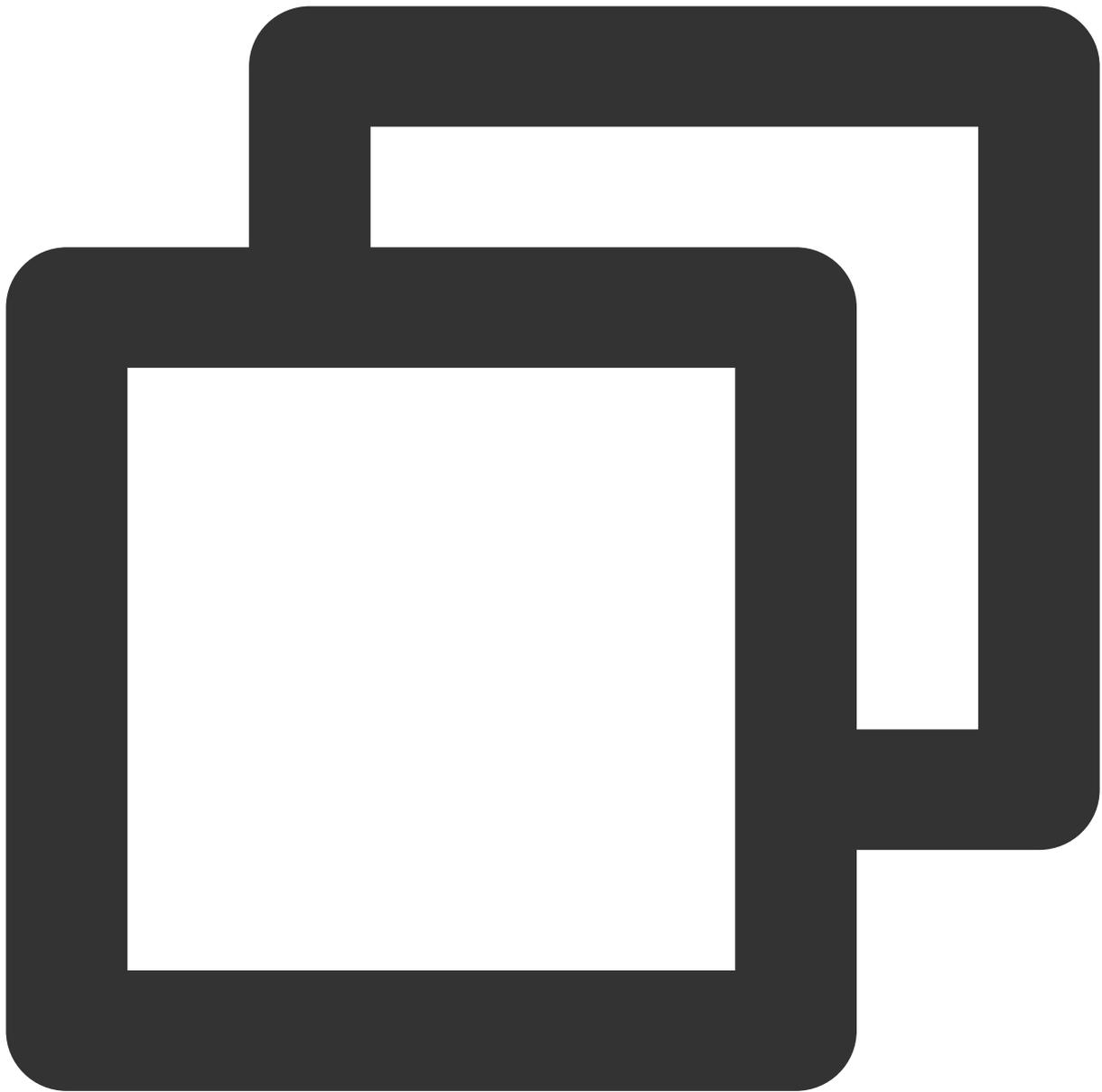
通过 `--workerNumber` 指定每个拷贝进程内的拷贝线程数



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

跳过具有相同校验值文件，进行增量迁移

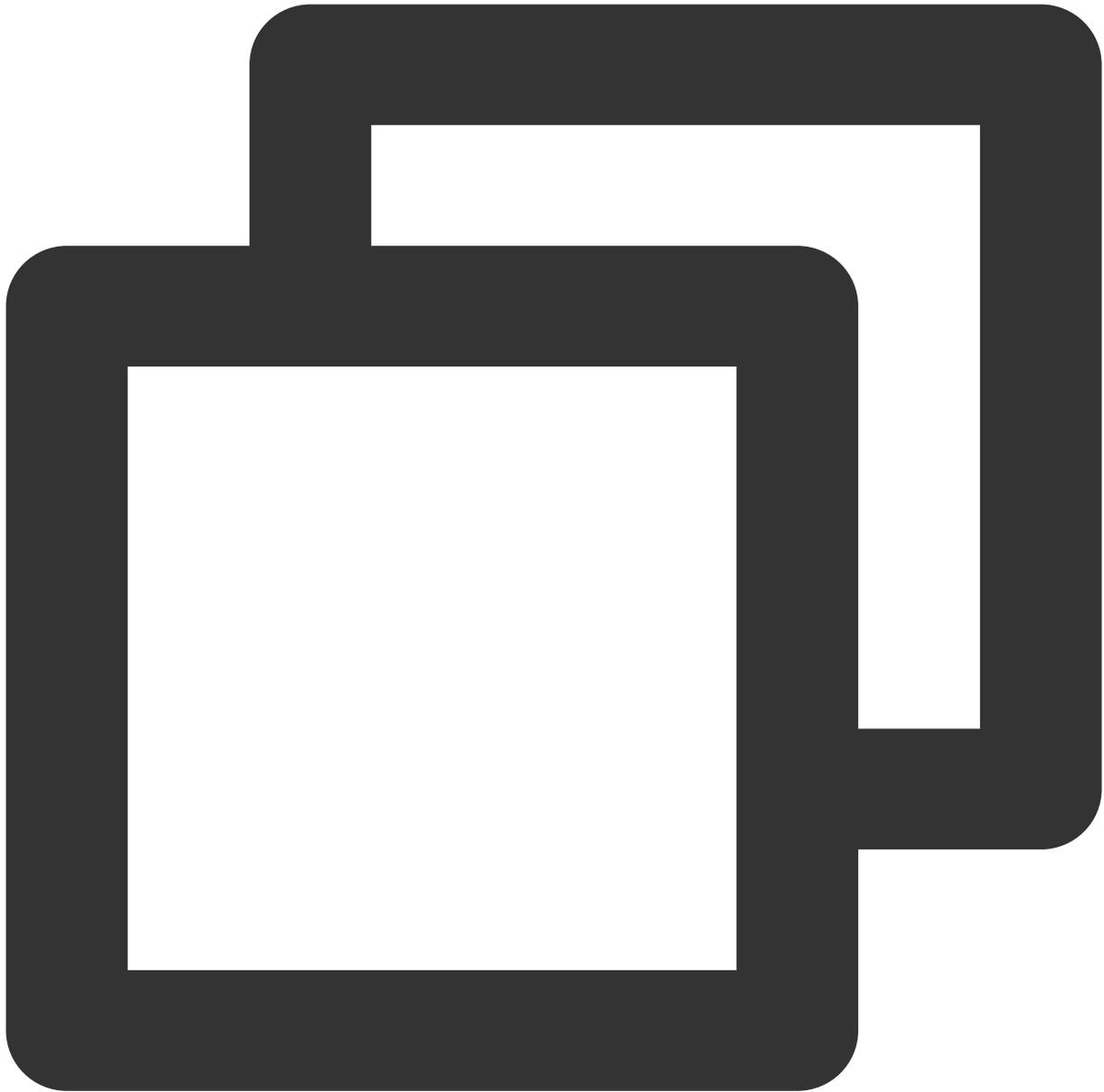
以参数 `--skipMode` 执行命令。跳过源和目标具有相同长度和校验和文件的拷贝，默认值 `length-checksum`：



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

`--skipMode` 选项用于在拷贝文件前，校验源文件和目标文件是否相同，若相同则跳过，可选 `none`（不校验）、`length`（长度）、`checksum`（CRC 值）和 `length-checksum`（长度 + CRC 值）。

如果源和目标文件系统的校验和算法不同，则会读取源端文件计算新的校验和。如果您的源是 HDFS，您可以通过如下方式，确定 HDFS 源是否支持 COMPOSITE-CRC32C 校验算法：



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

迁移完成后的数据校验及增量迁移

以参数 `--diffMode` 和 `--diffOutput` 执行命令：

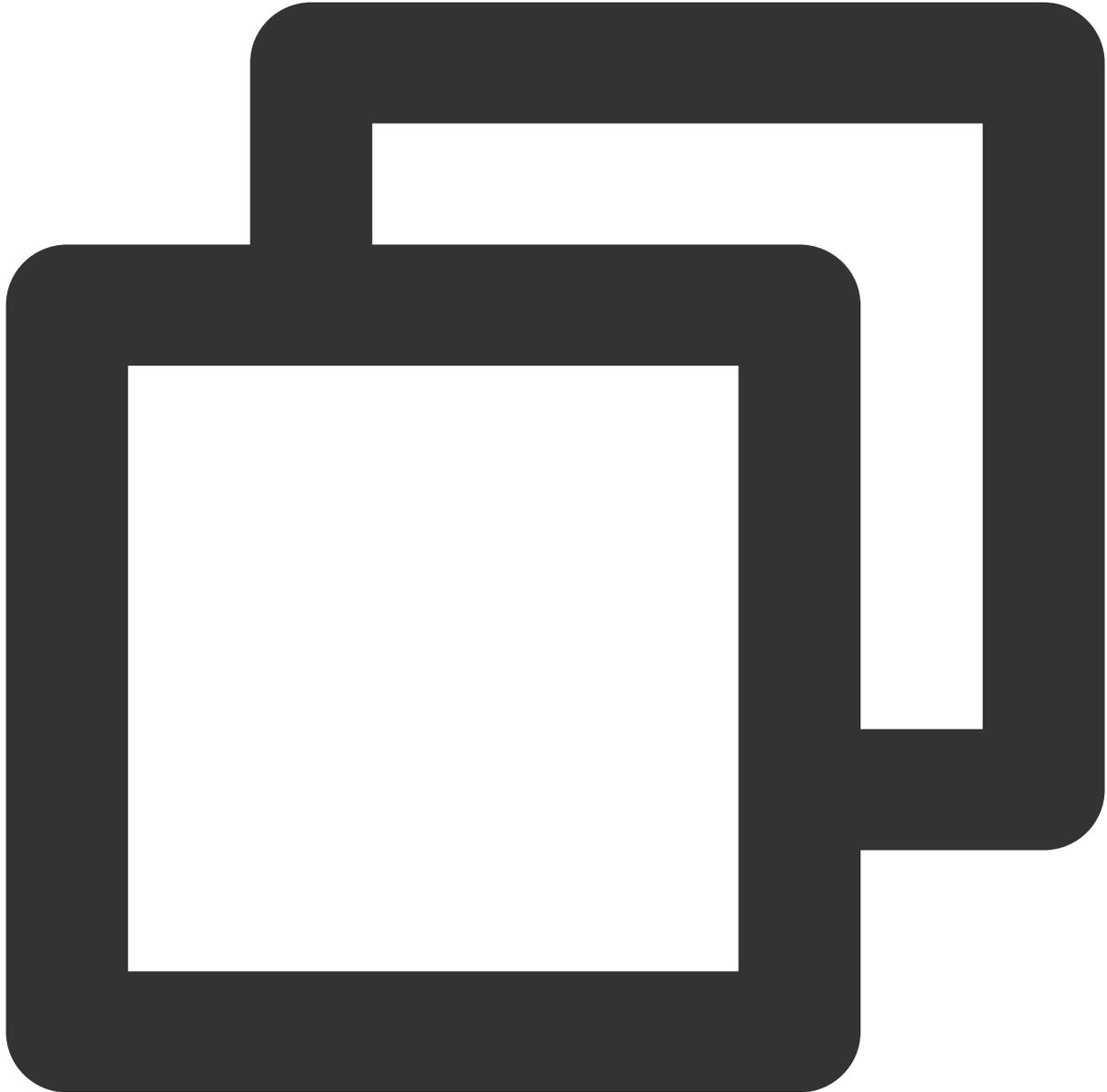
`--diffMode` 可选值为 `length` 和 `length-checksum`。

`--diffMode=length` 表示根据文件大小是否相同，获取差异文件列表。

`--diffMode=length-checksum`，根据文件大小和 CRC 检验和是否相同，获取差异文件列表。

`--diffOutput` 指定 diff 操作的输出目录。

如果目标文件系统为 COS，且源文件系统的 CRC 算法与之不同，则 COSDistCp 会拉取源文件计算目的文件系统的 CRC，以进行相同 CRC 算法值的对比。以下示例中，在迁移完成后，使用 `--diffMode` 参数，根据文件大小和 CRC 值，校验源和目标文件是否相同：

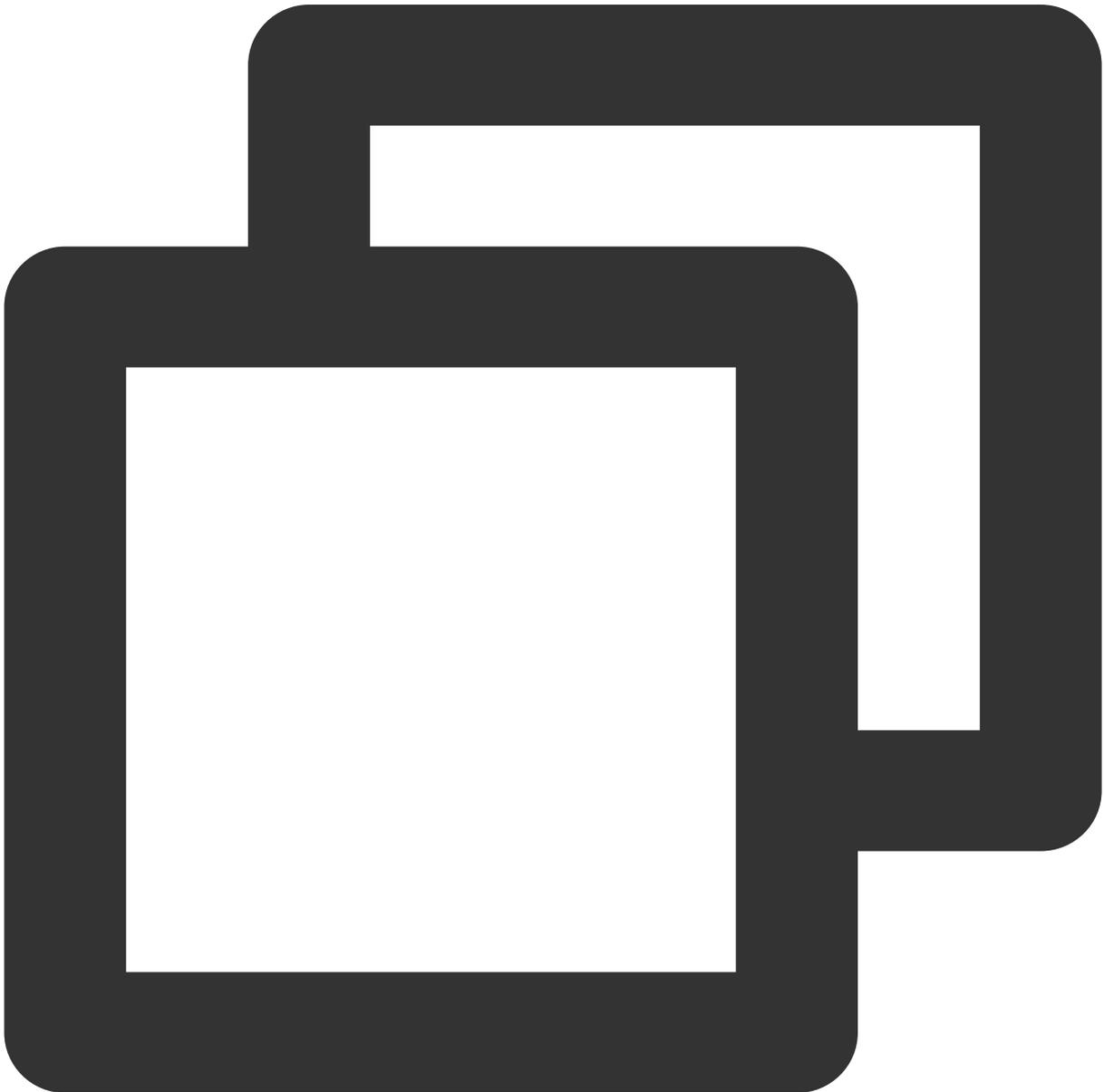


```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebuc
```

以上命令执行成功后，会输出以源文件系统文件列表为基准的计数器信息（请确保您的任务提交机器，配置了 MapReduce 任务的提交端 INFO 日志输出），您可以根据计数器信息，分析源和目的是否相同，计数器信息说明如下：

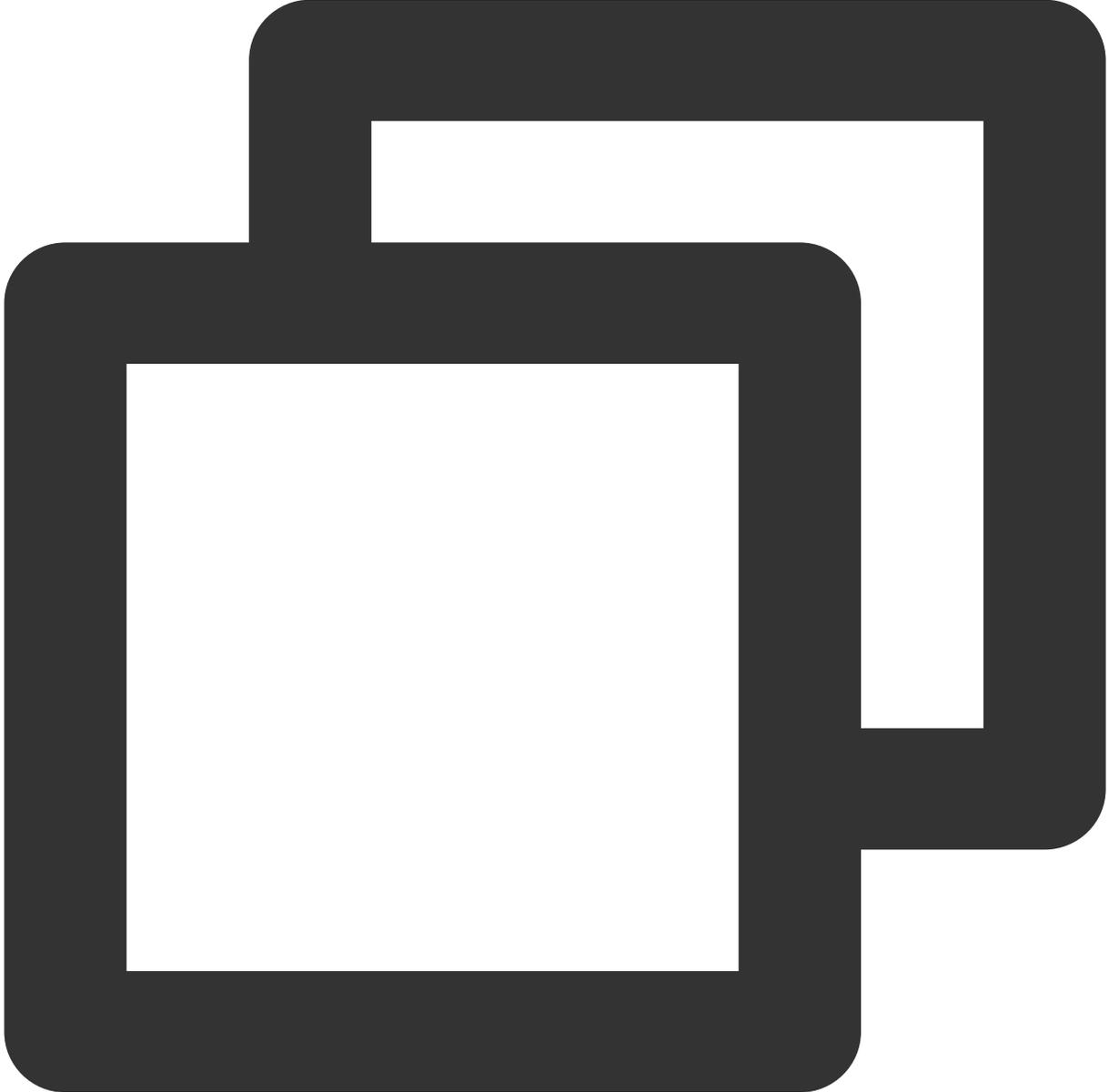
1. 源和目的文件相同，记录为 SUCCESS
2. 目标文件不存在，记录为 DEST_MISS
3. 存在源目录的清单中，但是校验时源文件不存在，记录为 SRC_MISS
4. 源文件和目标文件大小不同，记录为：LENGTH_DIFF
5. 源文件和目标文件 CRC 算法值不同，记录为：CHECKSUM_DIFF
6. 由于读取权限不够等因素导致 diff 操作失败，记录为：DIFF_FAILED
7. 源为目录，目的为文件，记录为：TYPE_DIFF

此外，COSDistcp 会在 HDFS 的 `/tmp/diff-output/failed` 目录下（1.0.5 及之前版本为 `/tmp/diff-output`），生成差异文件列表，您可以通过如下命令，获取除 SRC_MISS 以外的差异文件列表：



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

执行如下命令，根据差异文件列表进行增量迁移：



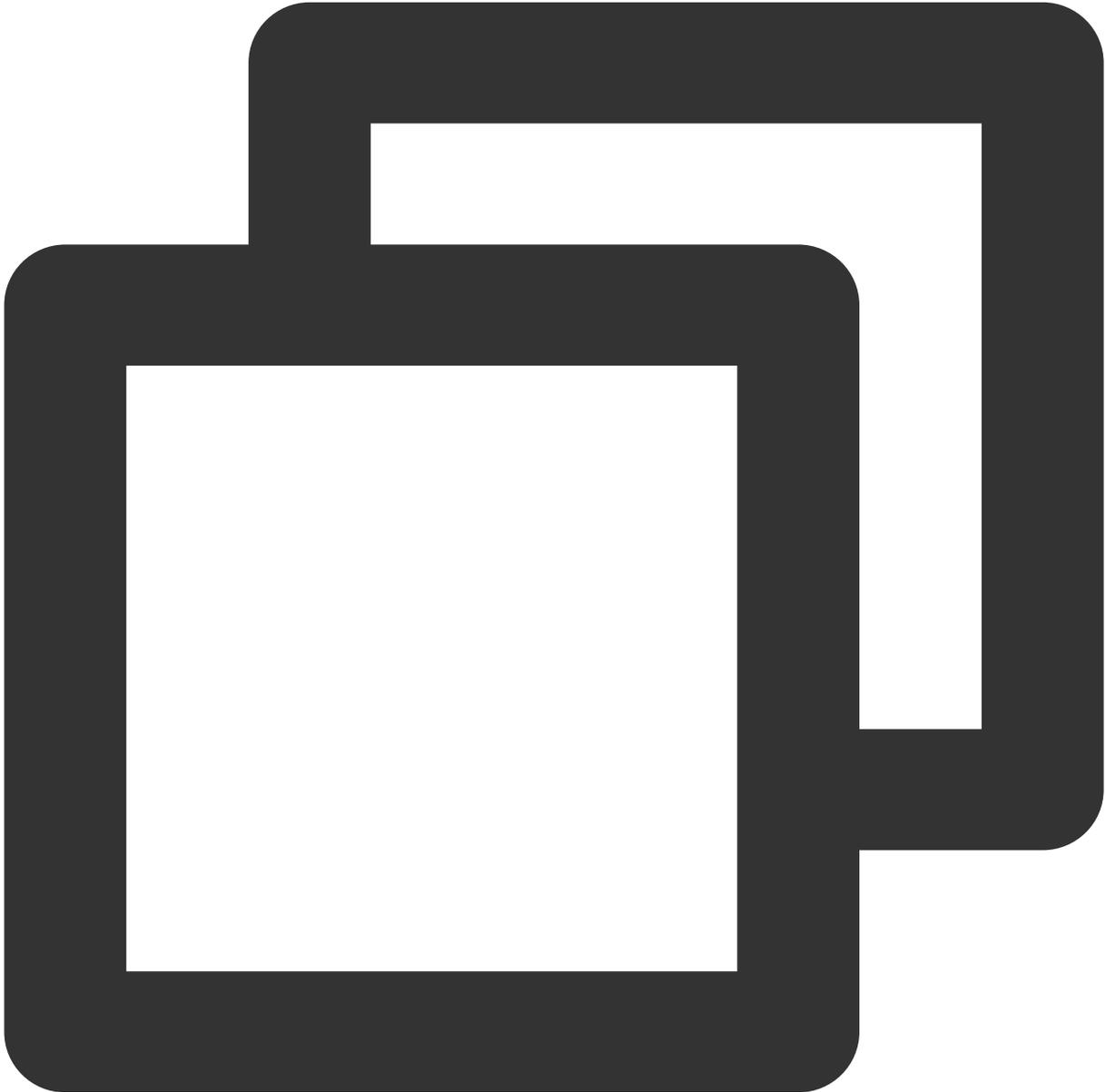
```
hadoop jar cos-distcp-${version}.jar --taskNumber=20 --src /data/warehouse --dest
```

增量迁移完成后，再次运行带 `--diffMode` 参数的命令，校验文件是否完全一致。

校验源文件和目标文件是否具有相同 CRC

以参数 `--checkMode` 执行命令，文件拷贝完成时，校验源文件和目标文件长度及校验和是否一致，默认值 `length-checksum`。

从非 COS 文件系统同步到 COS 时，如果源的 CRC 算法和 Hadoop-COS 的 CRC 算法不一致，则拷贝时计算 CRC，并在拷贝完成后，获取目标 COS 文件的 CRC，和计算得到的源文件 CRC 对比校验：



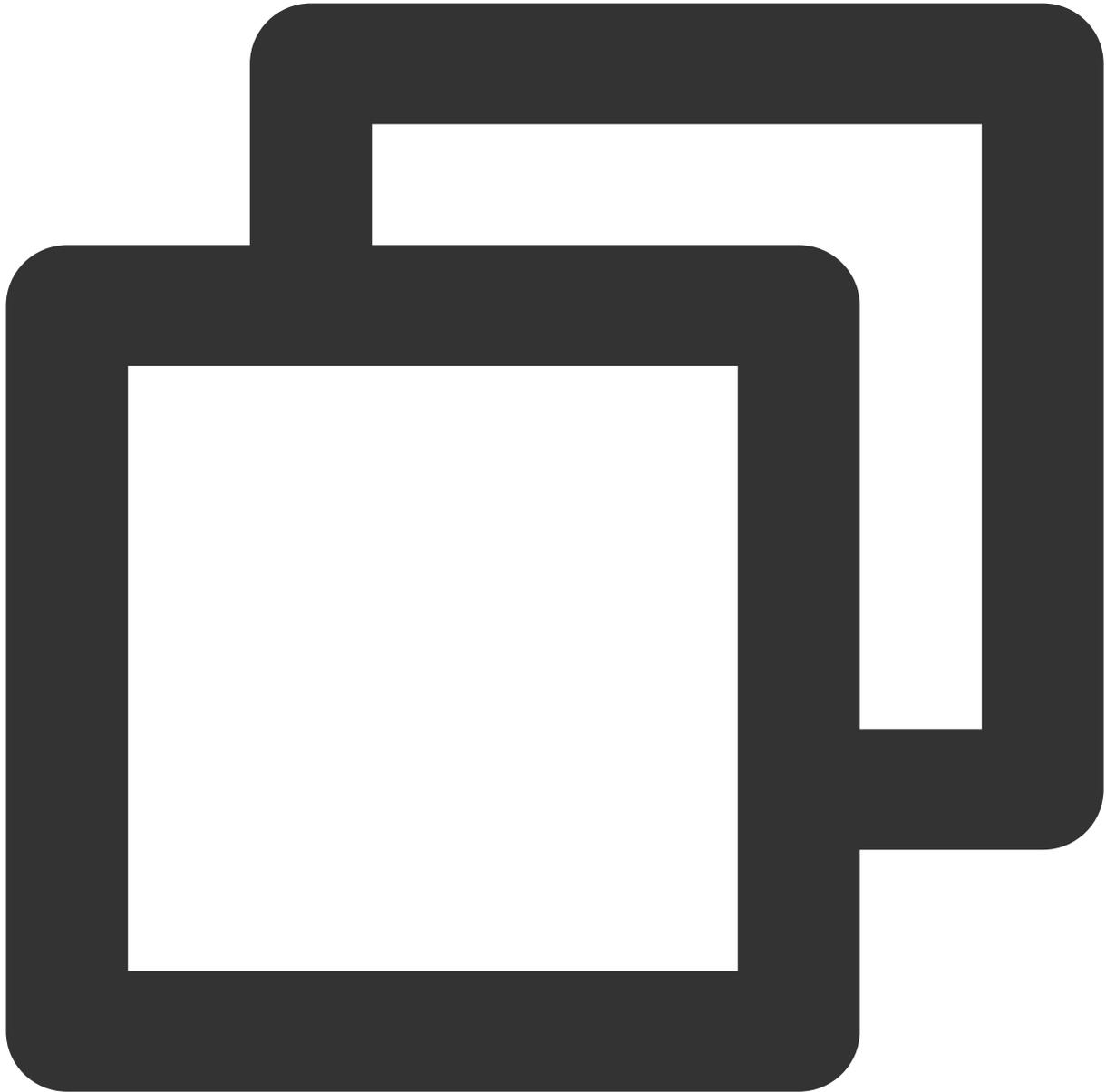
```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebuc
```

注意

在未指定 `--groupBy`，且 `--outputCodec` 为默认值时生效。

限制单文件读取带宽

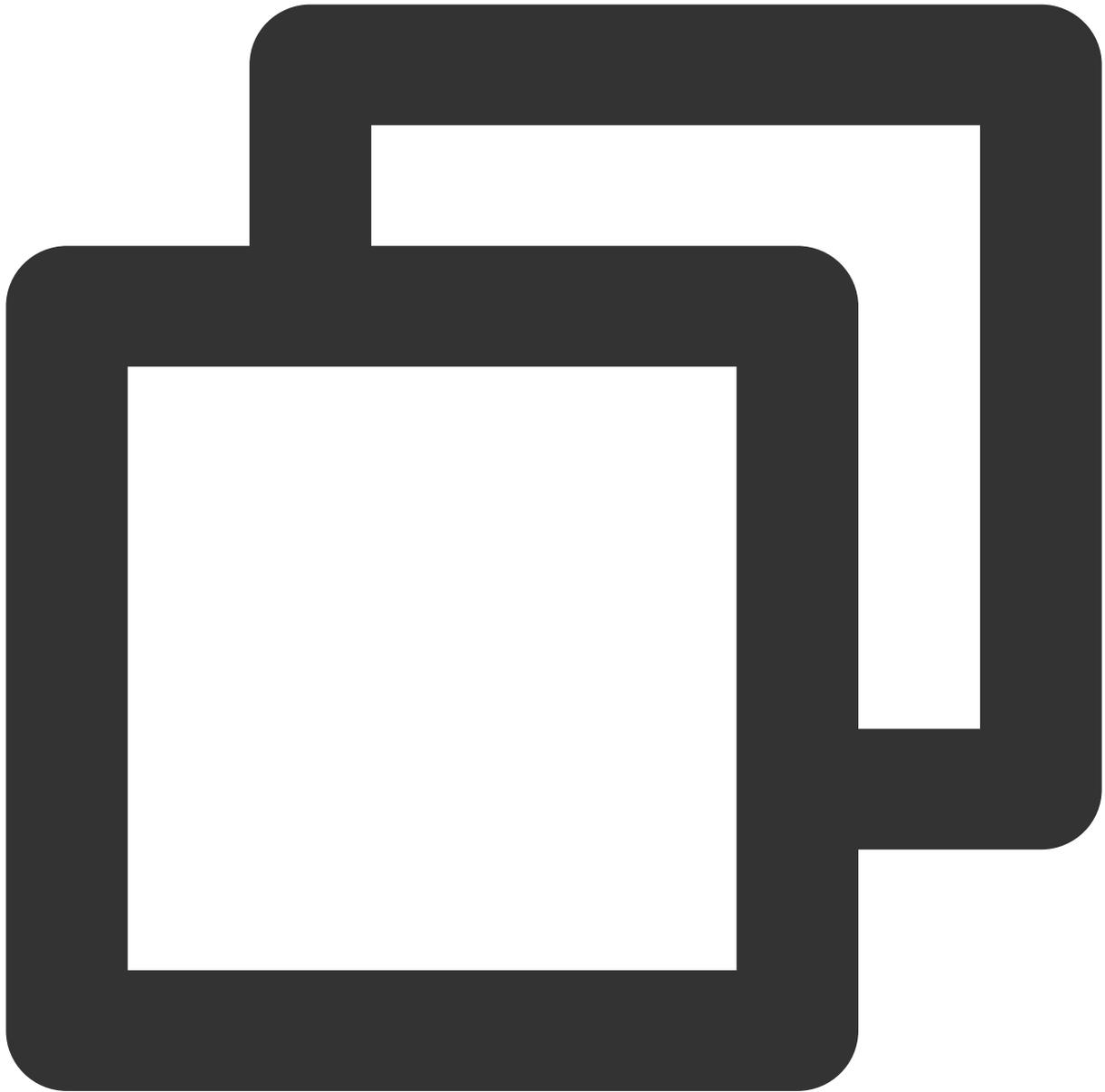
以参数 `--bandwidth` 执行命令，数值单位为MB。限制每个迁移文件的读取带宽为10MB/s，示例如下：



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

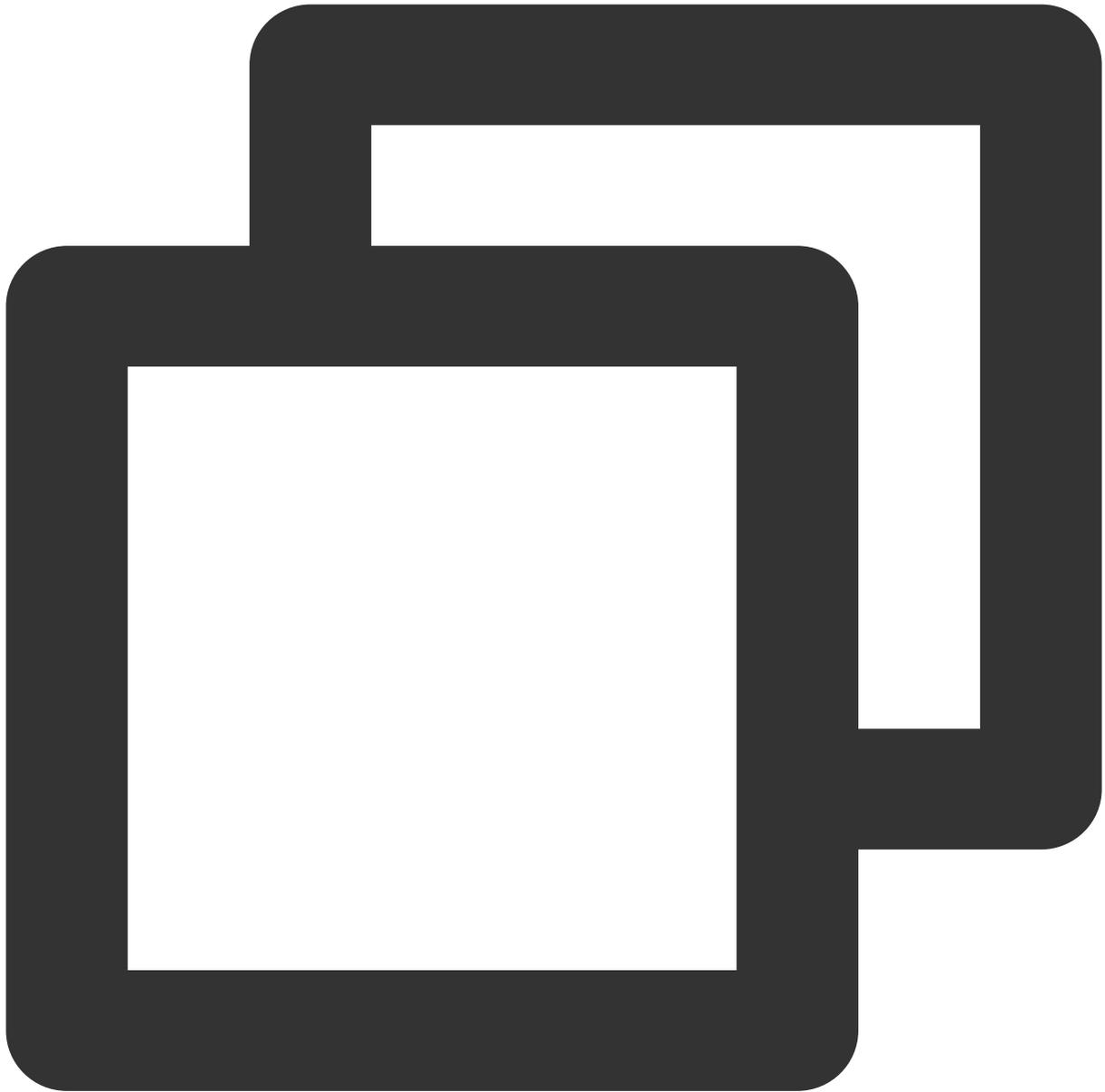
多目录同步

新建一个本地文件（例如 `srcPrefixes.txt`），在该文件中添加需要迁移的多个目录的绝对路径，这些目录之间需要不存在父子关系，添加之后，可通过 `cat` 命令查看，示例如下：



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

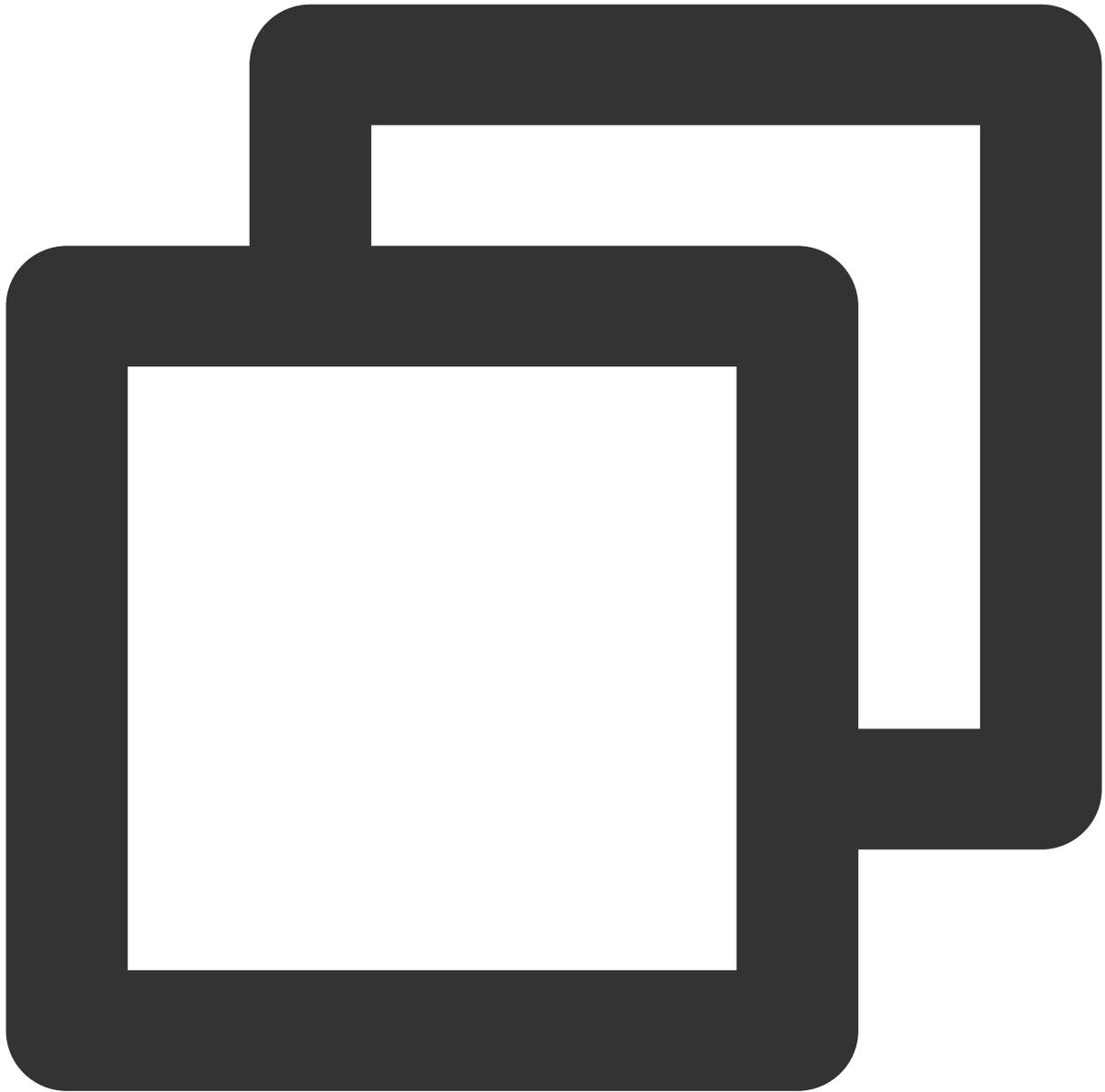
使用 `--srcPrefixesFile` 参数指定该文件，执行迁移命令：



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

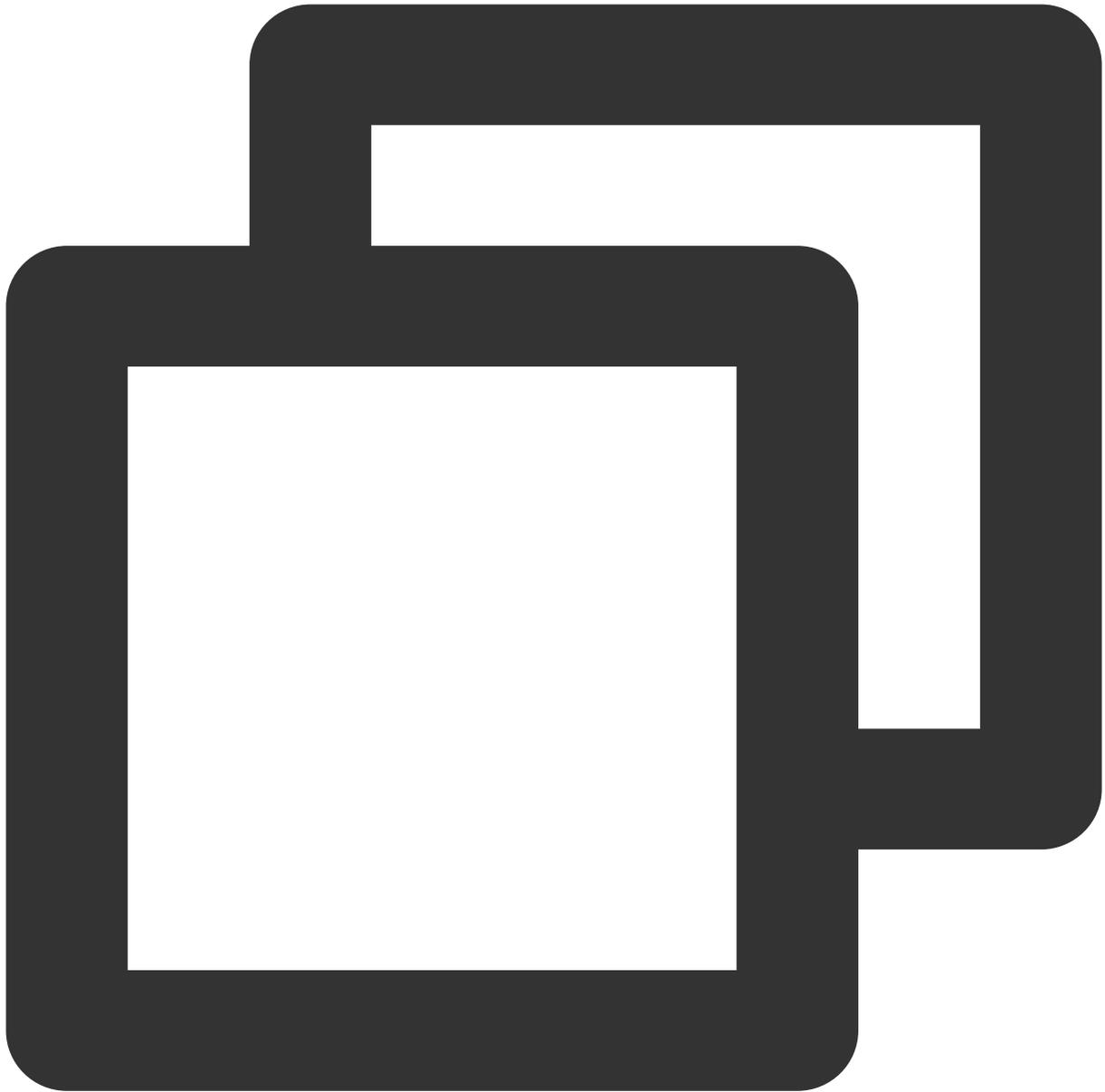
对输入文件进行正则表达式过滤

以参数 `--srcPattern` 执行命令，只同步 `/data/warehouse/` 目录下，以 `.log` 结尾的日志文件，示例如下：



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

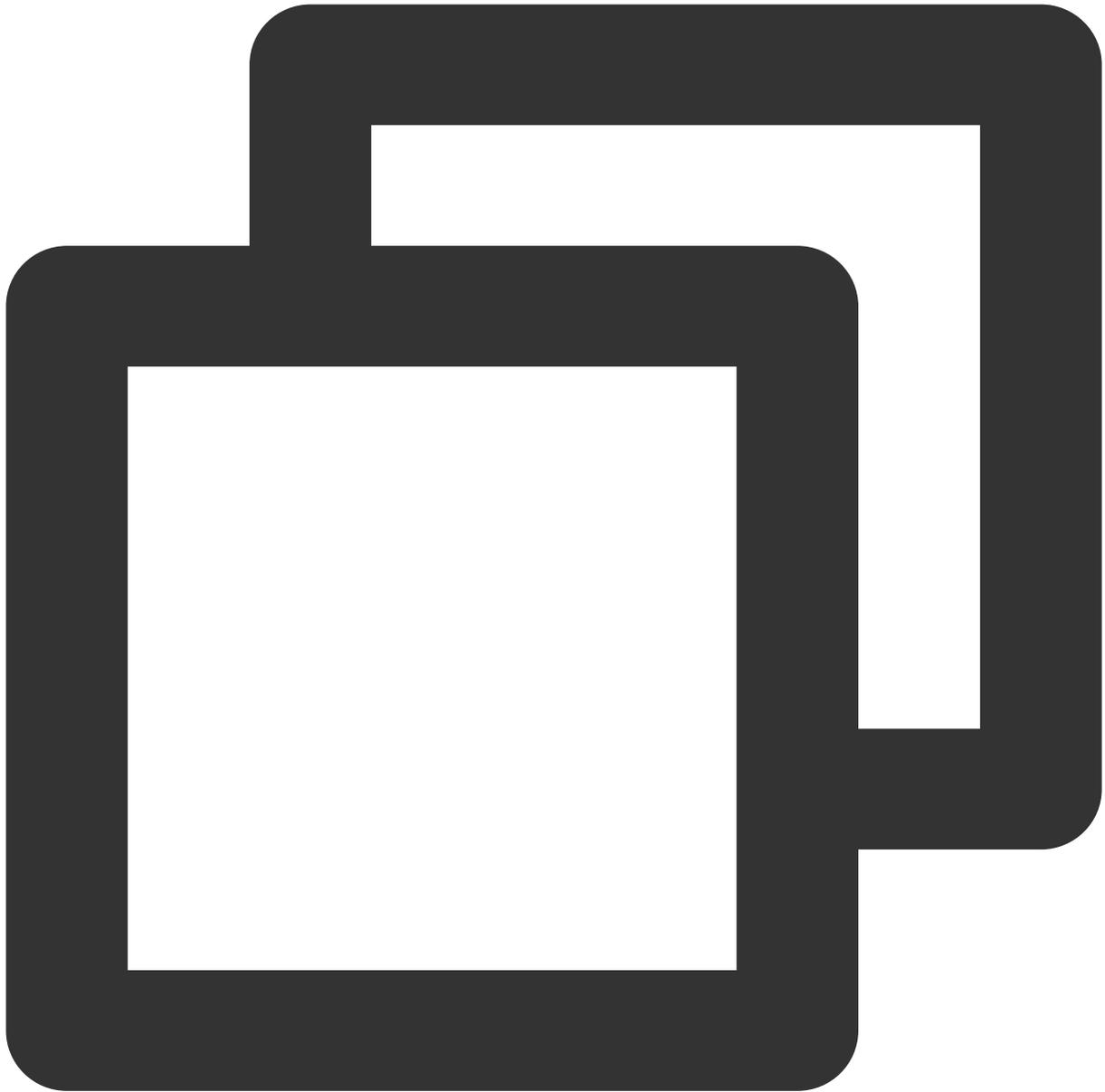
不迁移以 .temp 或 .tmp 结尾的文件：



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

指定 Hadoop-COS 的文件检验和类型

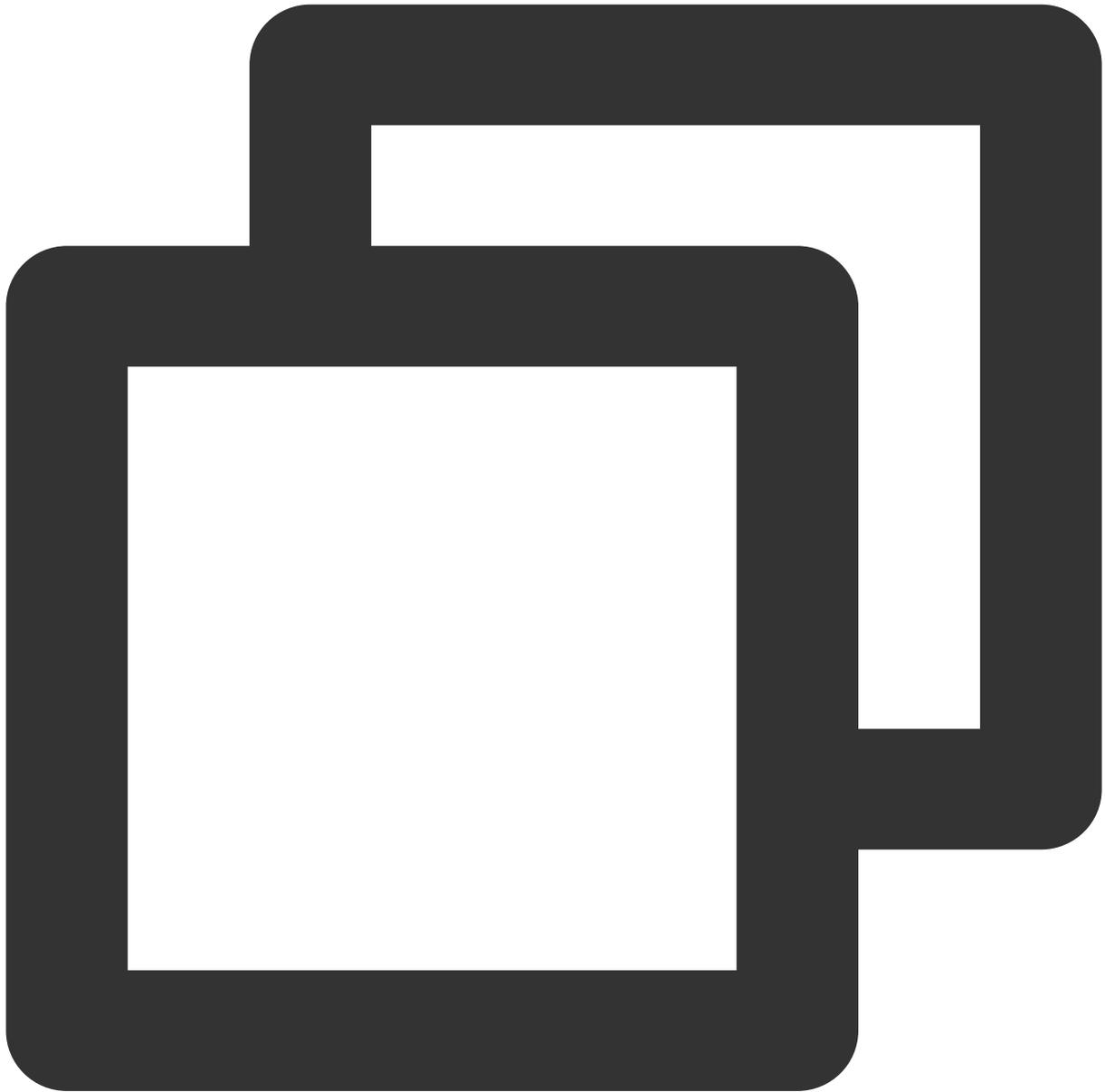
以参数 `--cosChecksumType` 执行命令，默认 CRC32C，可选 CRC32C 和 CRC64。



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

指定 COS 文件的存储类型

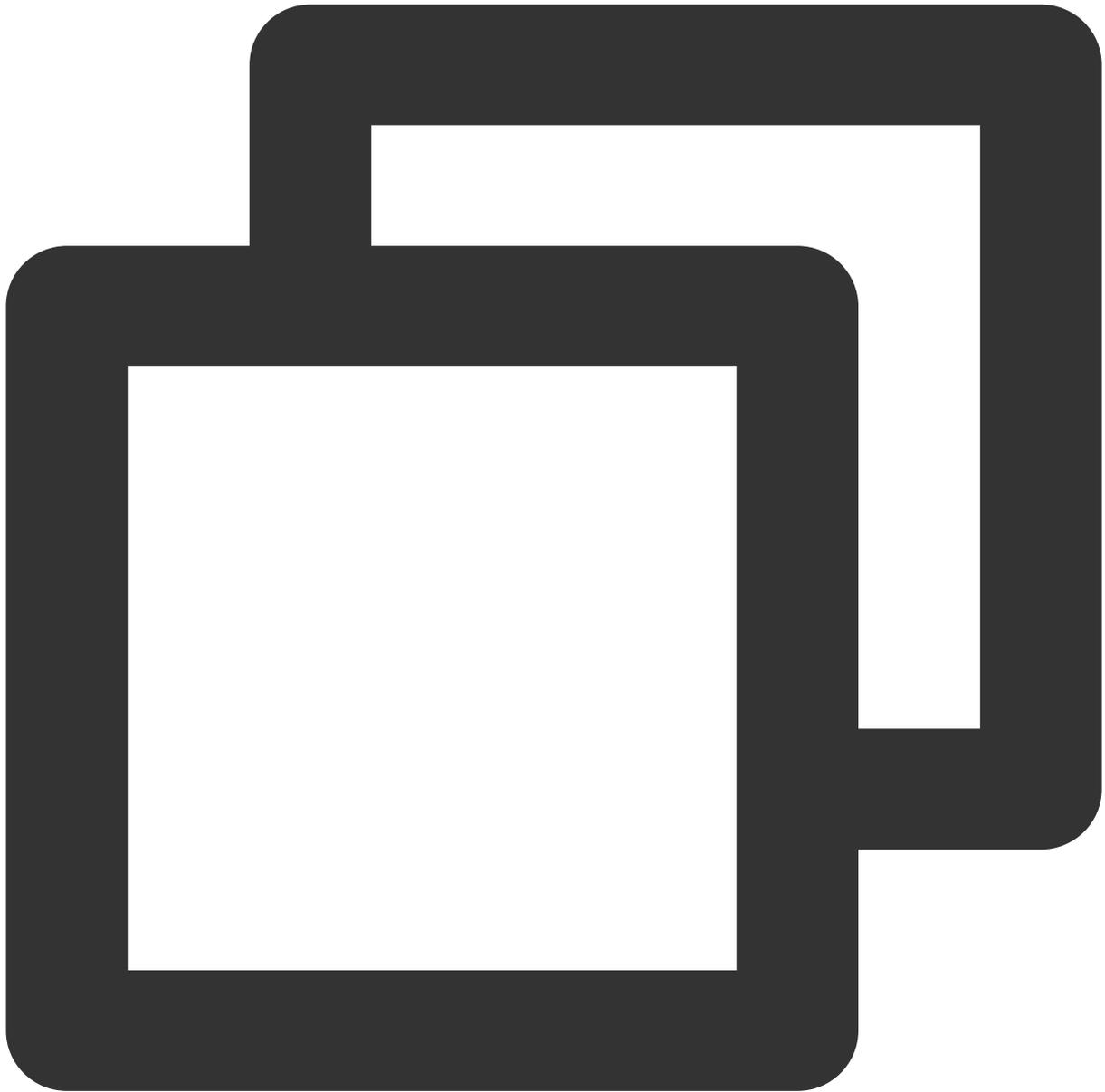
以参数 `--storageClass` 执行命令，示例如下：



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

指定目标文件的压缩类型

以参数 `--outputCodec` 执行命令，您可通过该参数，将 HDFS 中的数据实时压缩备份到 COS，节省存储成本。参数可选值为：`keep`、`none`、`gzip`、`lzop`、`snappy`，`none` 选项保存的目标文件为未压缩状态，`keep` 保持原来文件的压缩状态。示例如下：



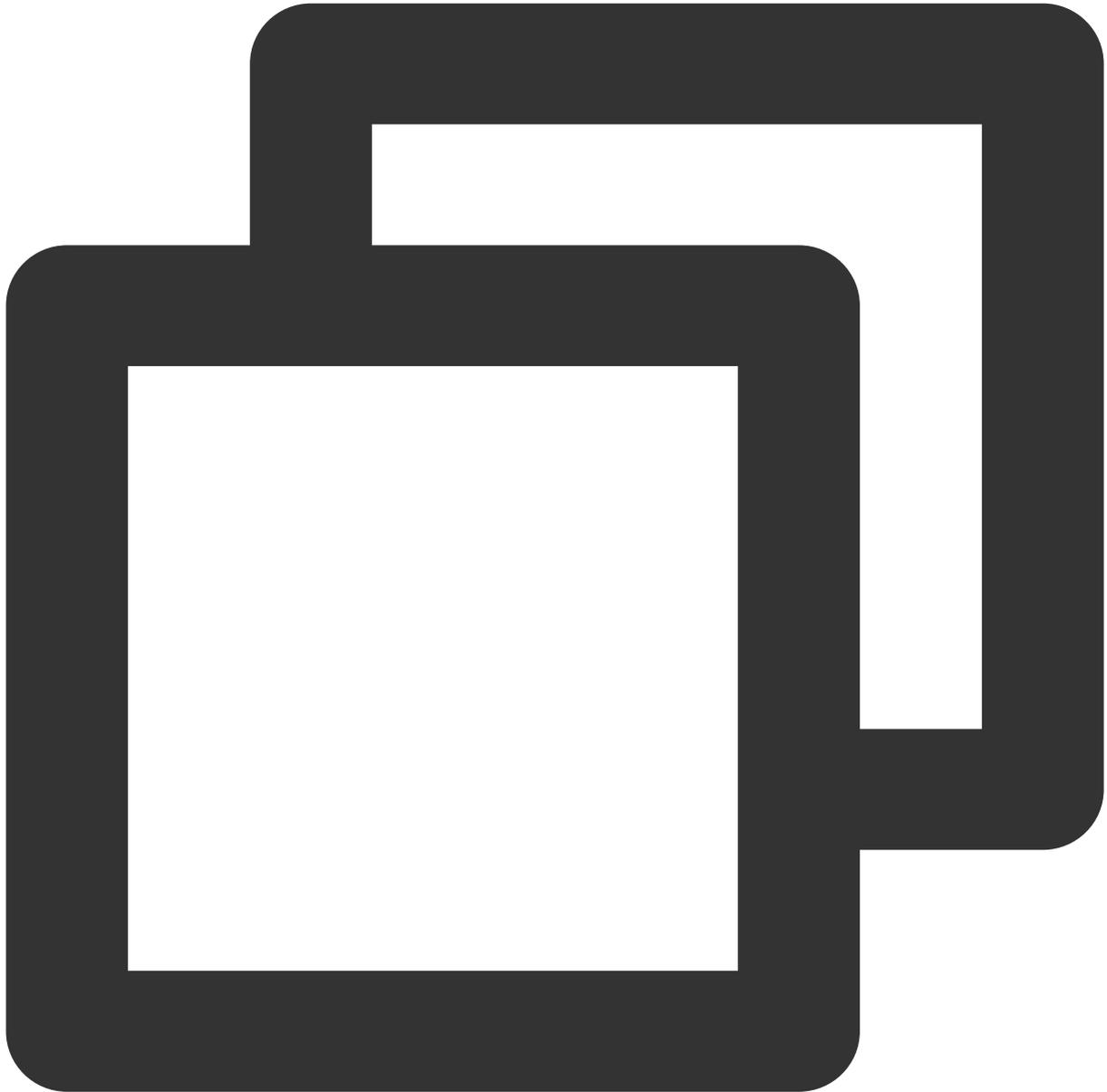
```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/logs --dest cosn://examp
```

注意

其中除 `keep` 选项外，皆会先对文件先解压，随后转换为目标压缩类型。因此，除 `keep` 选项外，可能会由于压缩参数等不一致，导致目标文件和源文件不一致，但解压后的文件一致；在未指定 `--groupBy`，且 `--outputCodec` 为默认值时，可通过 `--skipMode` 进行增量迁移，`--checkMode` 进行数据校验。

删除源文件

以参数 `--deleteOnSuccess` 执行命令，将 `/data/warehouse` 目录下文件从 HDFS 同步到 COS 后，立即删除源目录中的对应文件：



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

注意

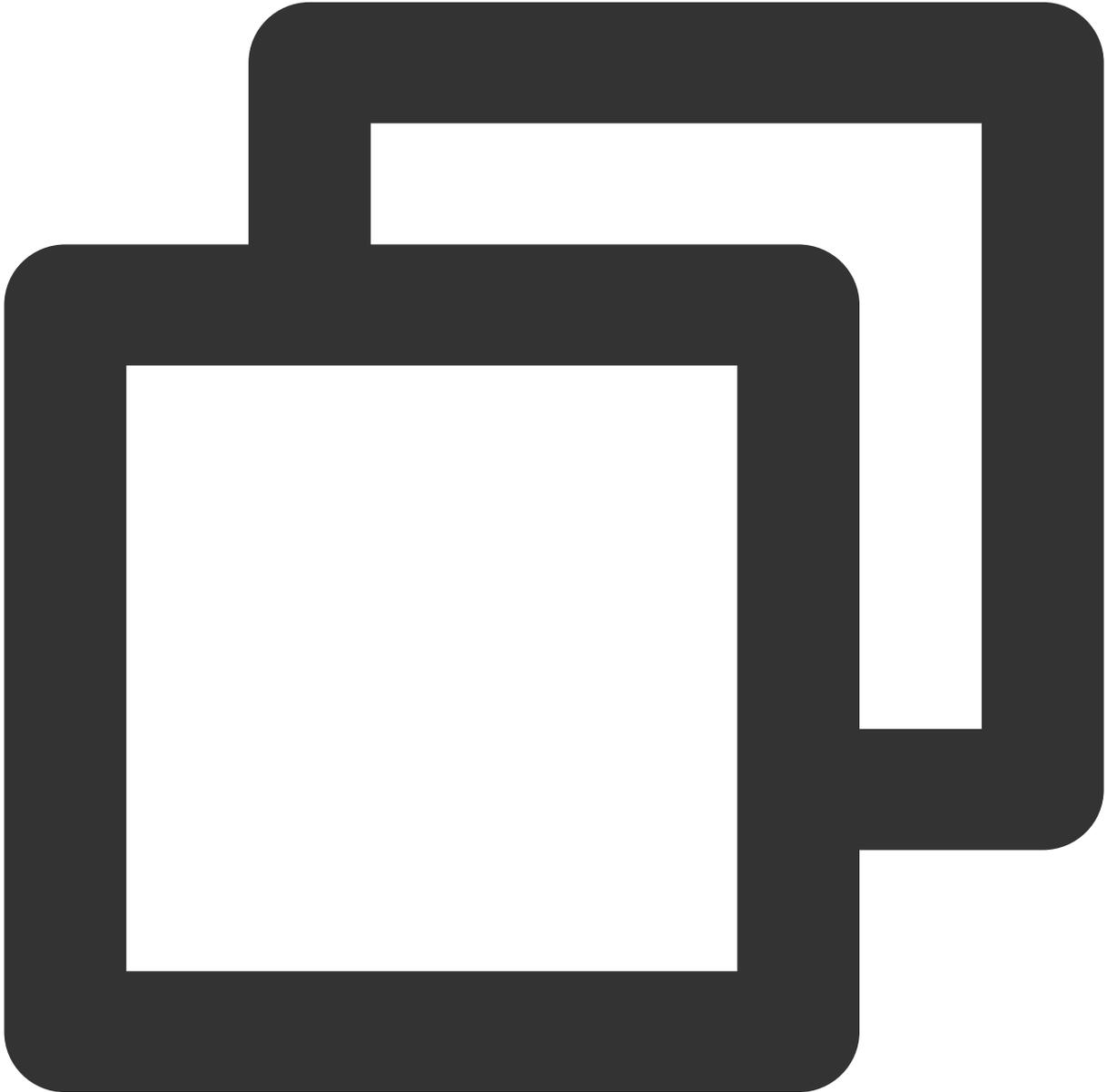
指定该选项后每迁移完一个文件，立即删除对应的源文件，并非整个迁移完成后，再删除源文件，请谨慎使用，1.7 及其以后版本不再提供该参数。

生成目标清单文件和指定上一次清单输出文件

以参数 `--outputManifest` 和 `--previousManifest` 执行命令。

`--outputManifest` 该选项首先会在本地生成一个 gzip 压缩的 `manifest.gz`，并在迁移成功时，移动到 `--dest` 所指定的目录下。

`--previousManifest` 指定上一次 `--outputManifest` 输出文件，COSDistCp 会跳过相同长度大小的文件：



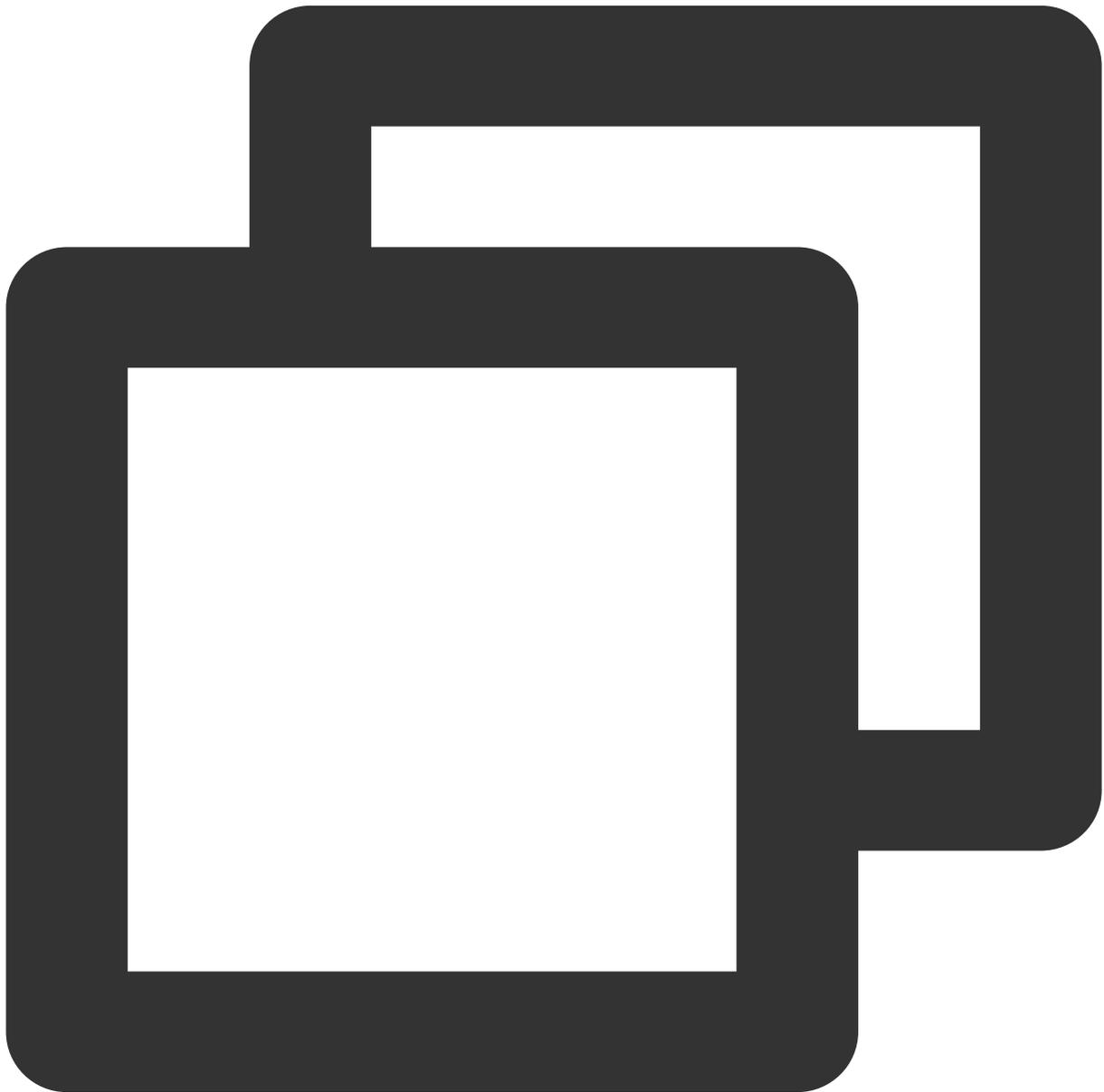
```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebuc
```

注意

上述命令的增量迁移，只能同步文件大小变化的文件，无法同步文件内容发生变化的文件。如果文件内容可能发生变化，请参考 `--diffMode` 使用示例，根据文件的 CRC 确定文件发生变化的文件列表。

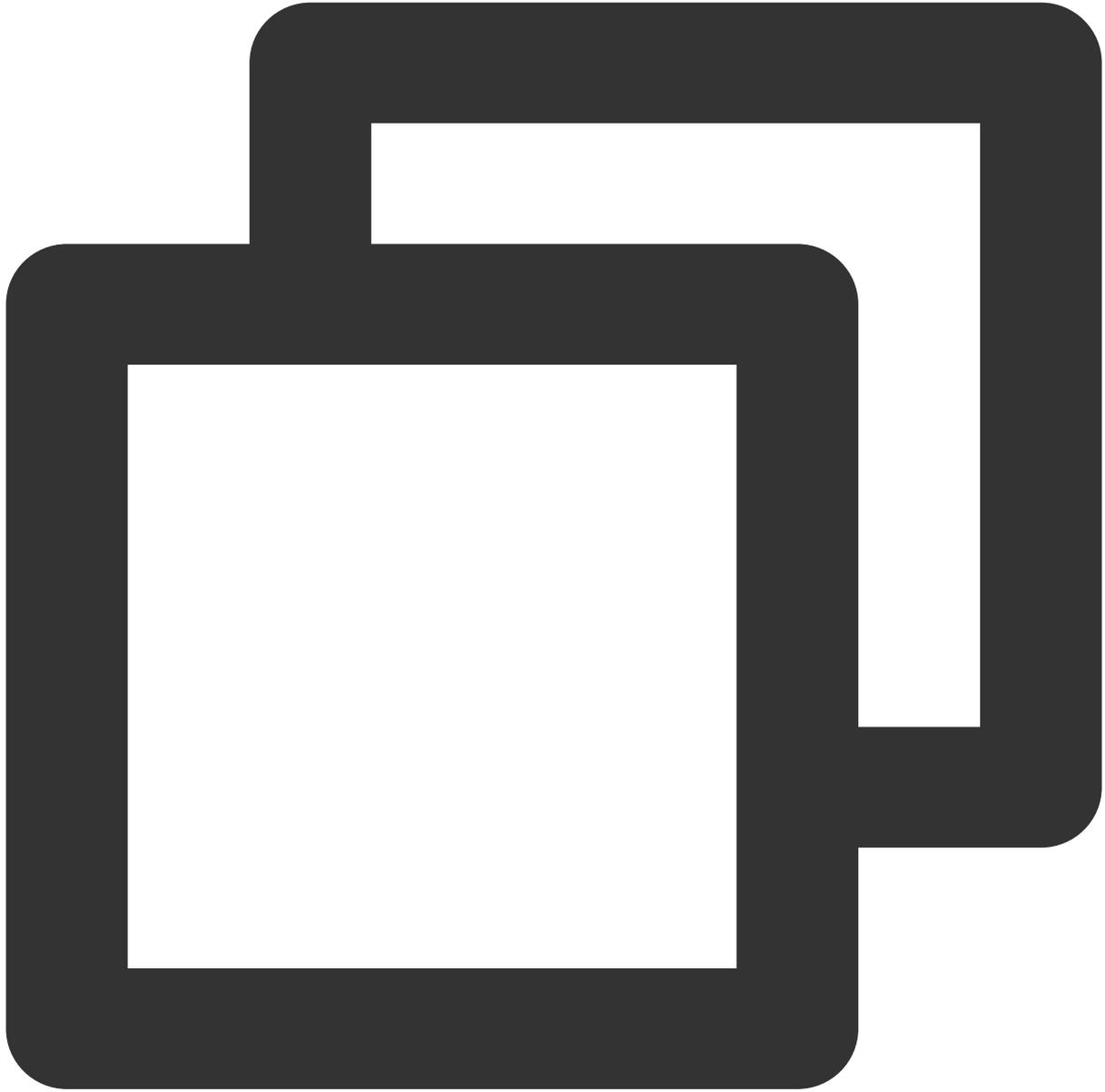
指定迁移任务分配策略为动态分配

如果您的文件分布情况是文件大小分化剧烈，例如极少量超大文件，大量小文件或者迁移机器负载不一。您可以通过 `--enableDynamicStrategy` 开启任务动态分配策略，使执行速度快的任务，迁移更多的文件，从而减少任务的执行时间：



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

迁移完成后对迁移的数据进行校验：



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

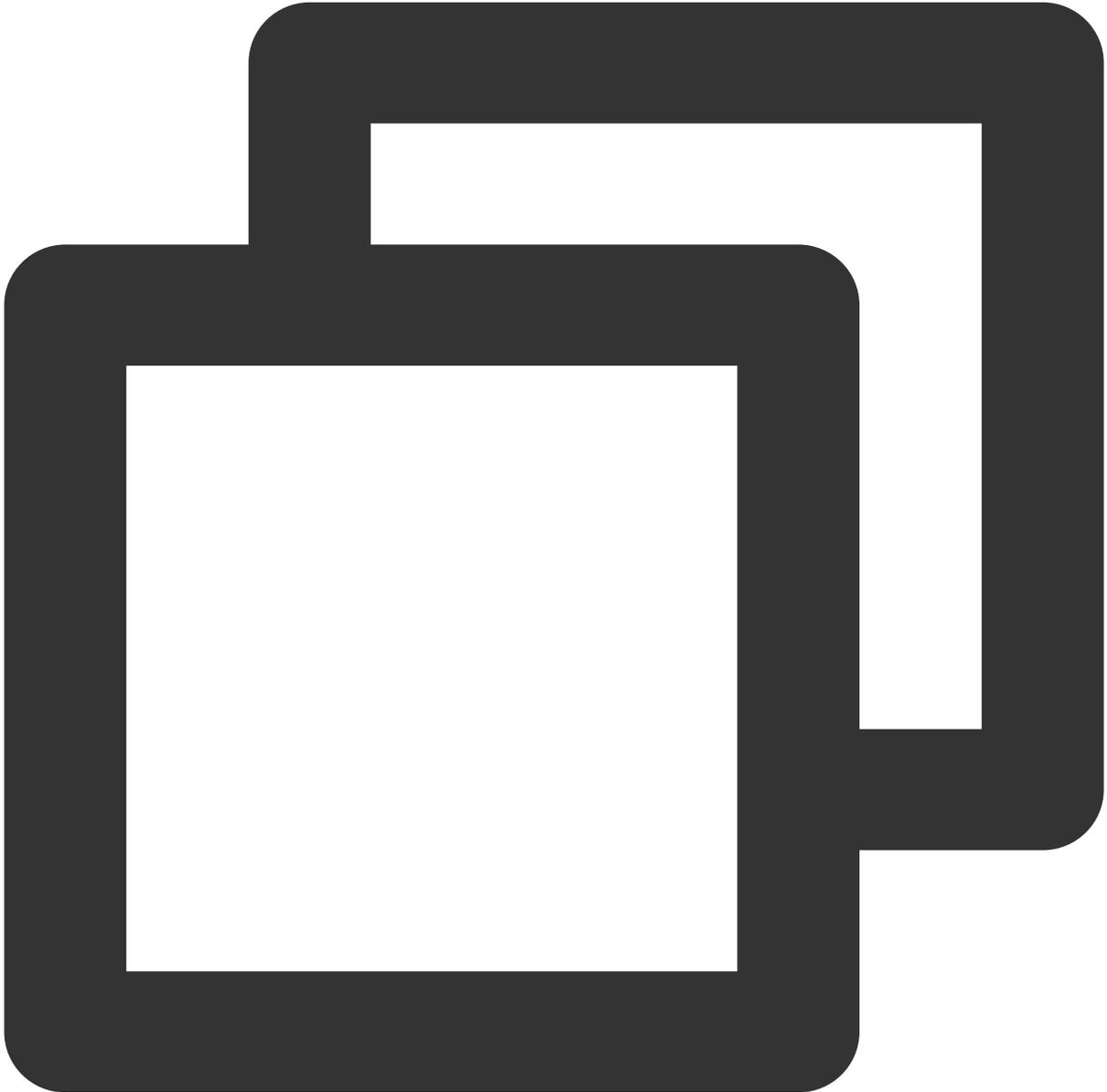
注意

该模式存在一定局限性，例如任务计数器在进程异常的情况下，可能计数不准确，请迁移完成后用 `--diffMode` 对数据进行校验。

拷贝文件的元信息

以参数 `--preserveStatus` 执行命令，将源文件或源目录的 `user`、`group`、`permission` 和 `timestamps`（`modification time` 和 `access time`）拷贝到目标文件或目标目录，该参数在将文件从 HDFS 拷贝到 CHDFS 时生效。

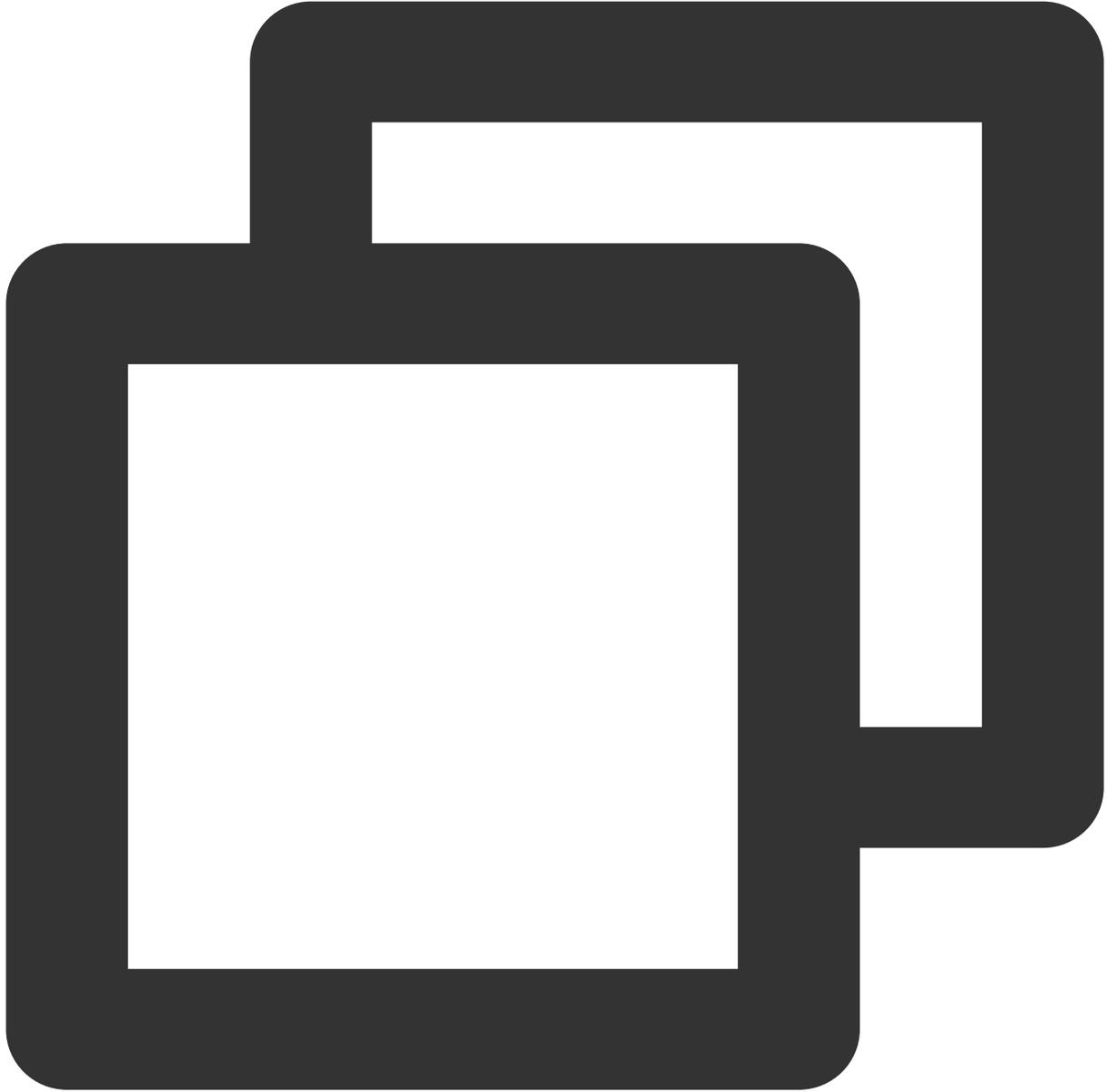
示例如下：



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

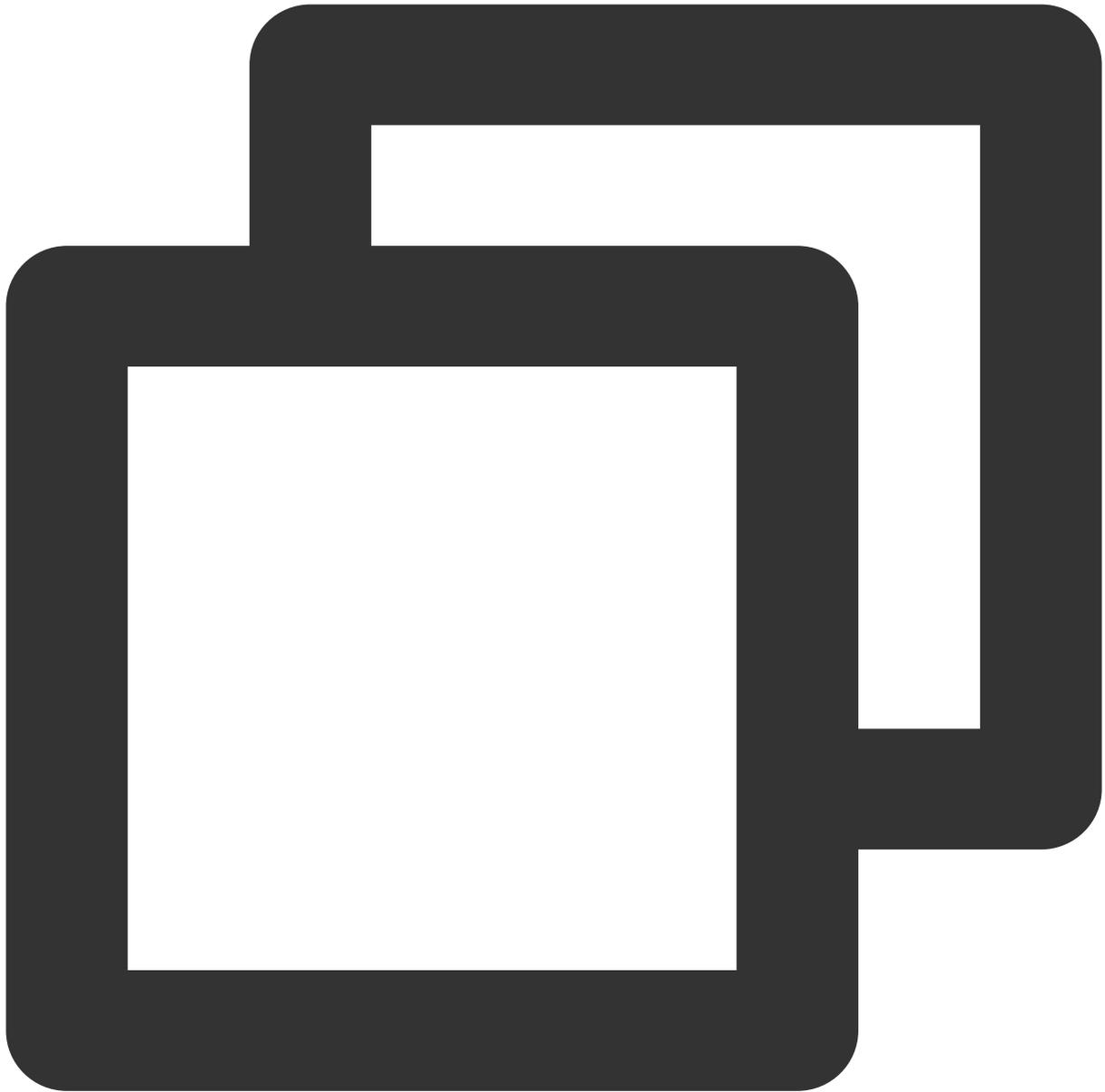
配置 Prometheus 监控

您可以在 Yarn 的管理页面，查看 COSDistcp 迁移任务的计数器，包括当前已经迁移完成的文件数、字节数等信息。为了方便地查看迁移任务计数器的曲线变化，您只需要简单的配置，即可将 COSDistcp 迁移任务的计数器展示在 Prometheus + Grafana 监控体系中，配置 prometheus.yml，添加抓取任务：



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

以参数 `--promPort=VALUE` 执行命令，将当前 MapReduce 任务的计数器暴露到外部：



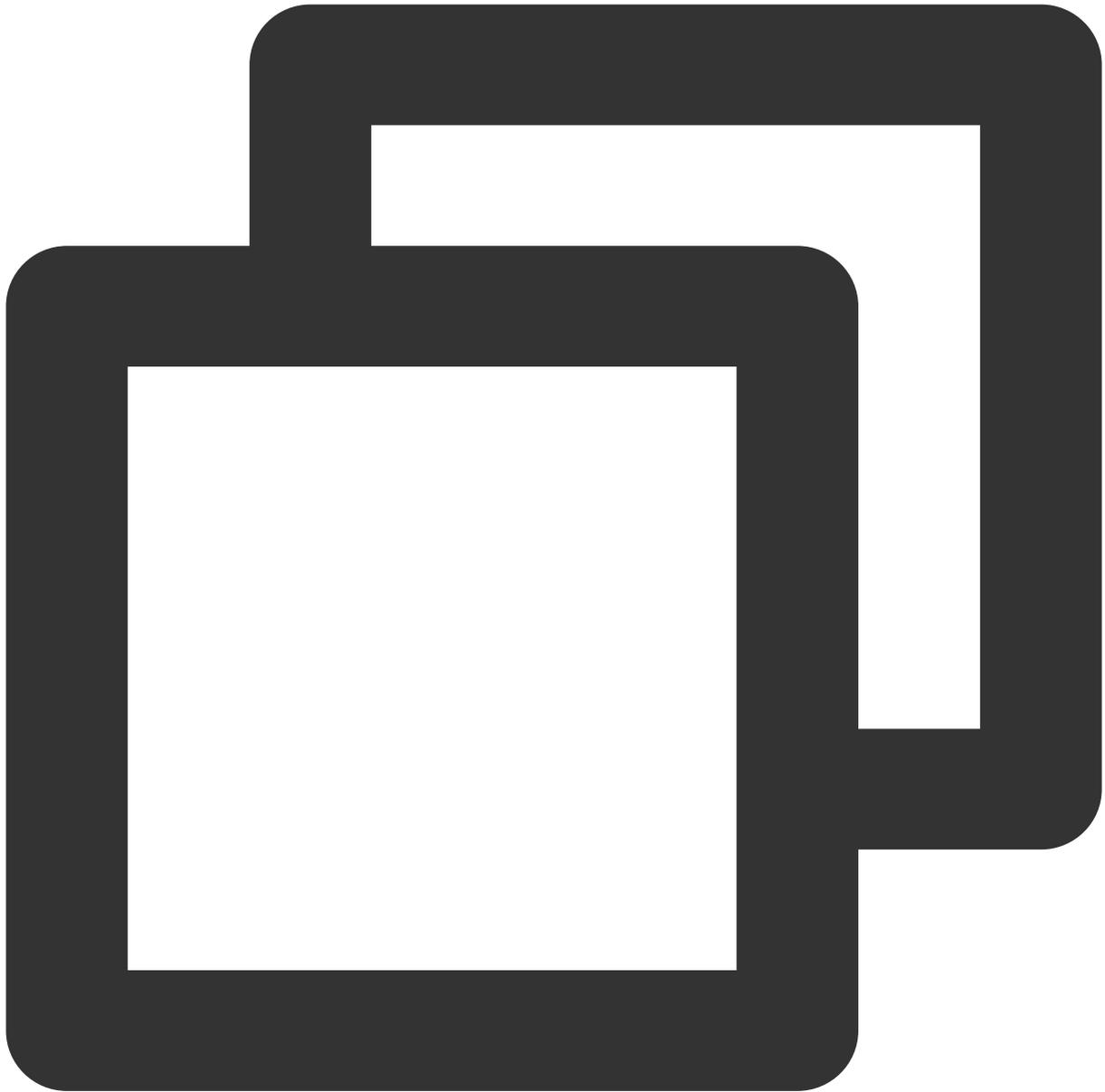
```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest cosn://examplebuc
```

下载示例 [Grafana Dashboard](#) 并导入，Grafana 展示如下：



拷贝文件失败时告警

以参数 `--completionCallbackClass` 指定回调类路径执行命令，COSDistCp 会在拷贝任务完成的时候，将收集的任务信息作为参数执行回调函数。用户自定义的回调函数，需要实现如下接口，前往 [下载回调示例代码](#)：

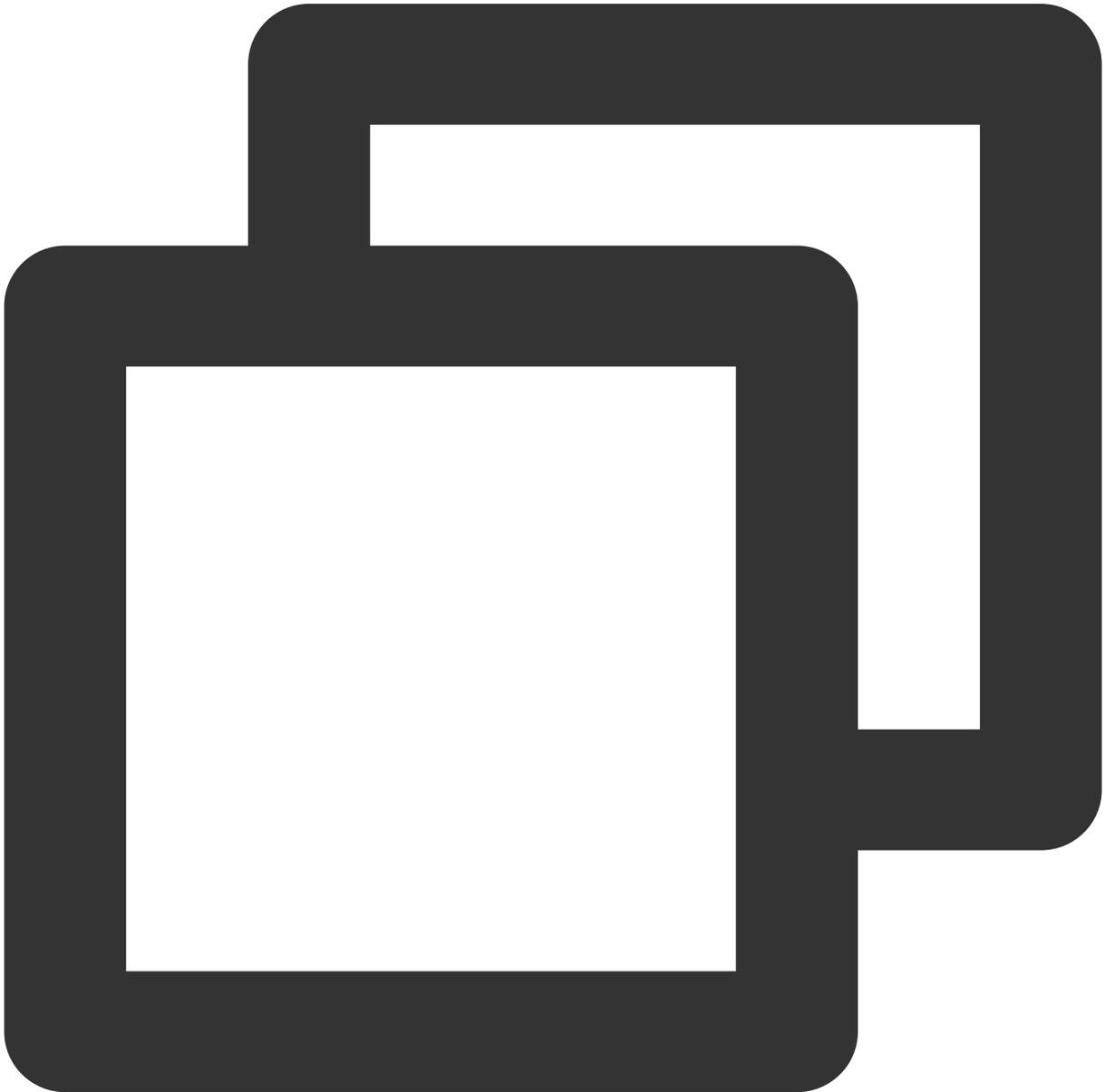


```
package com.qcloud.cos.distcp;
import java.util.Map;
public interface TaskCompletionCallback {
/**
 * @description: When the task is completed, the callback function is executed
 * @param jobType Copy or Diff
 * @param jobStartTime the job start time
 * @param errorMsg the exception error msg
 * @param applicationId the MapReduce application id
 * @param: cosDistCpCounters the job
 */
}
```

```
void doTaskCompletionCallback(String jobType, long jobStartTime, String errorMsg, S

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

COSDistCp 内部集成了腾讯云监测系统的告警，在任务出现异常及存在文件拷贝失败的时候，执行告警：



```
export alarmSecretId=SECRET-ID
```

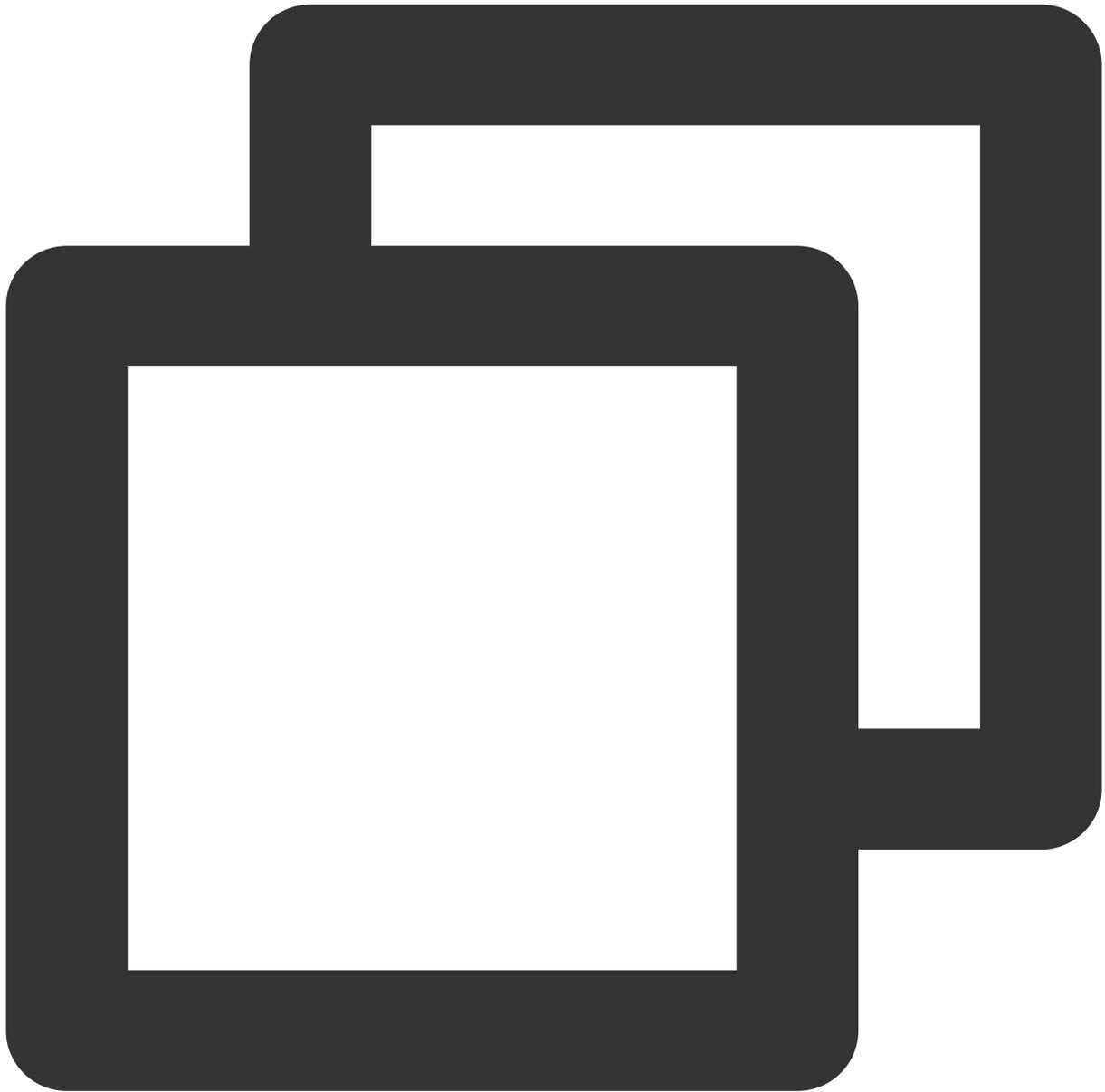
```
export alarmSecretKey=SECRET-KEY
export alarmRegion=ap-guangzhou
export alarmModule=module
export alarmPolicyId=cm-xxx
hadoop jar cos-distcp-1.4-2.8.5.jar \
-Dfs.cosn.credentials.provider=org.apache.hadoop.fs.auth.SimpleCredentialProvider \
-Dfs.cosn.userinfo.secretId=SECRET-ID \
-Dfs.cosn.userinfo.secretKey=SECRET-KEY \
-Dfs.cosn.bucket.region=ap-guangzhou \
-Dfs.cosn.impl=org.apache.hadoop.fs.CosFileSystem \
-Dfs.AbstractFileSystem.cosn.impl=org.apache.hadoop.fs.CosN \
--src /data/warehouse \
--dest cosn://examplebucket-1250000000/data/warehouse/ \
--checkMode=checksum \
--completionCallbackClass=com.qcloud.cos.distcp.DefaultTaskCompletionCallback
```

以上命令中 `alarmPolicyId` 为腾讯云监测系统告警策略，可进入腾讯云监测系统控制台进行创建和配置（告警管理 > 告警配置 > 自定义消息）。

常见问题

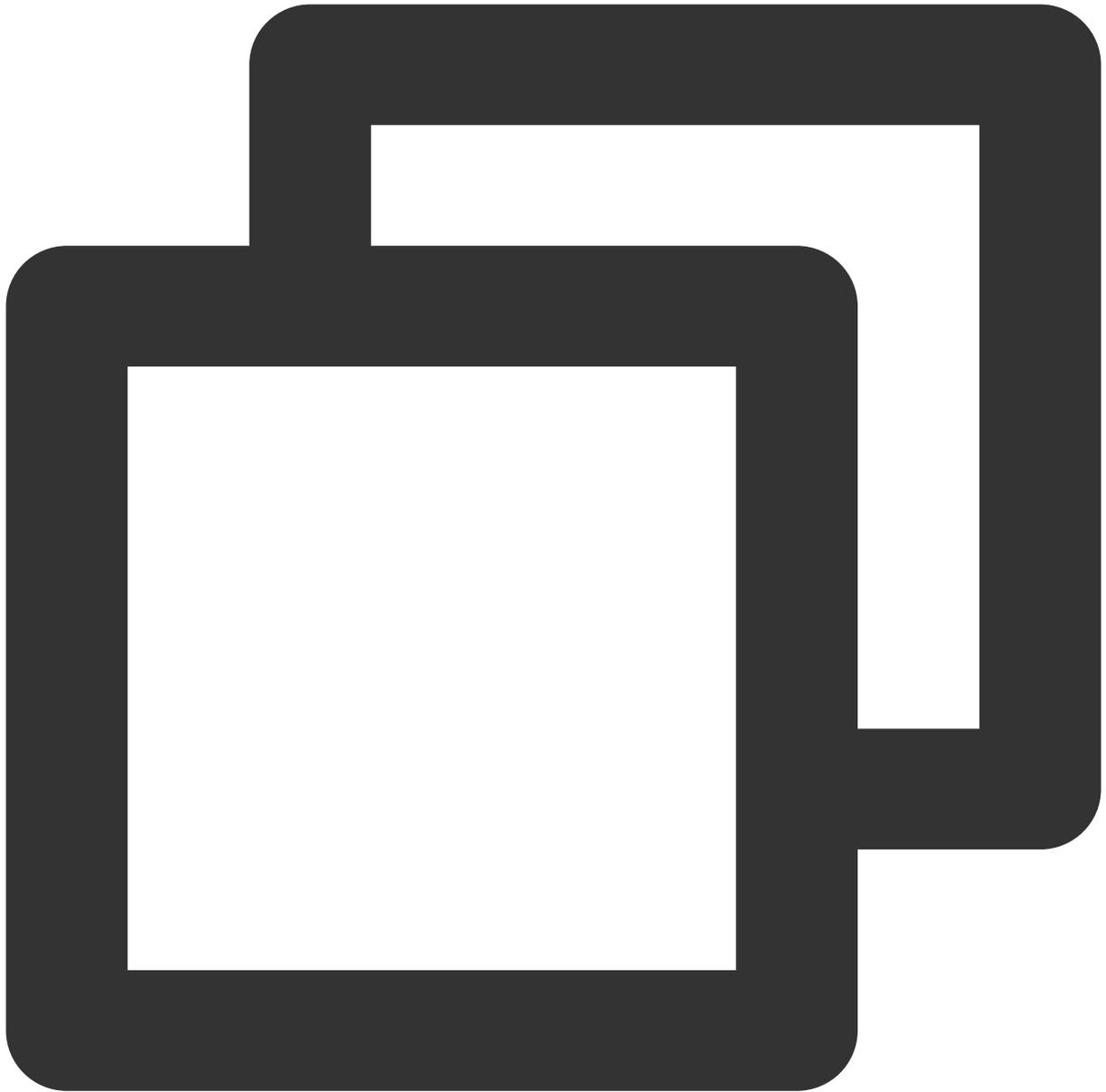
使用 COSDistcp 迁移 HDFS 数据包含哪些阶段，如何调整迁移性能和保障数据的正确性？

COSDistcp 每迁移完成一个文件，都会根据 `checkMode` 对迁移的文件进行校验：



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

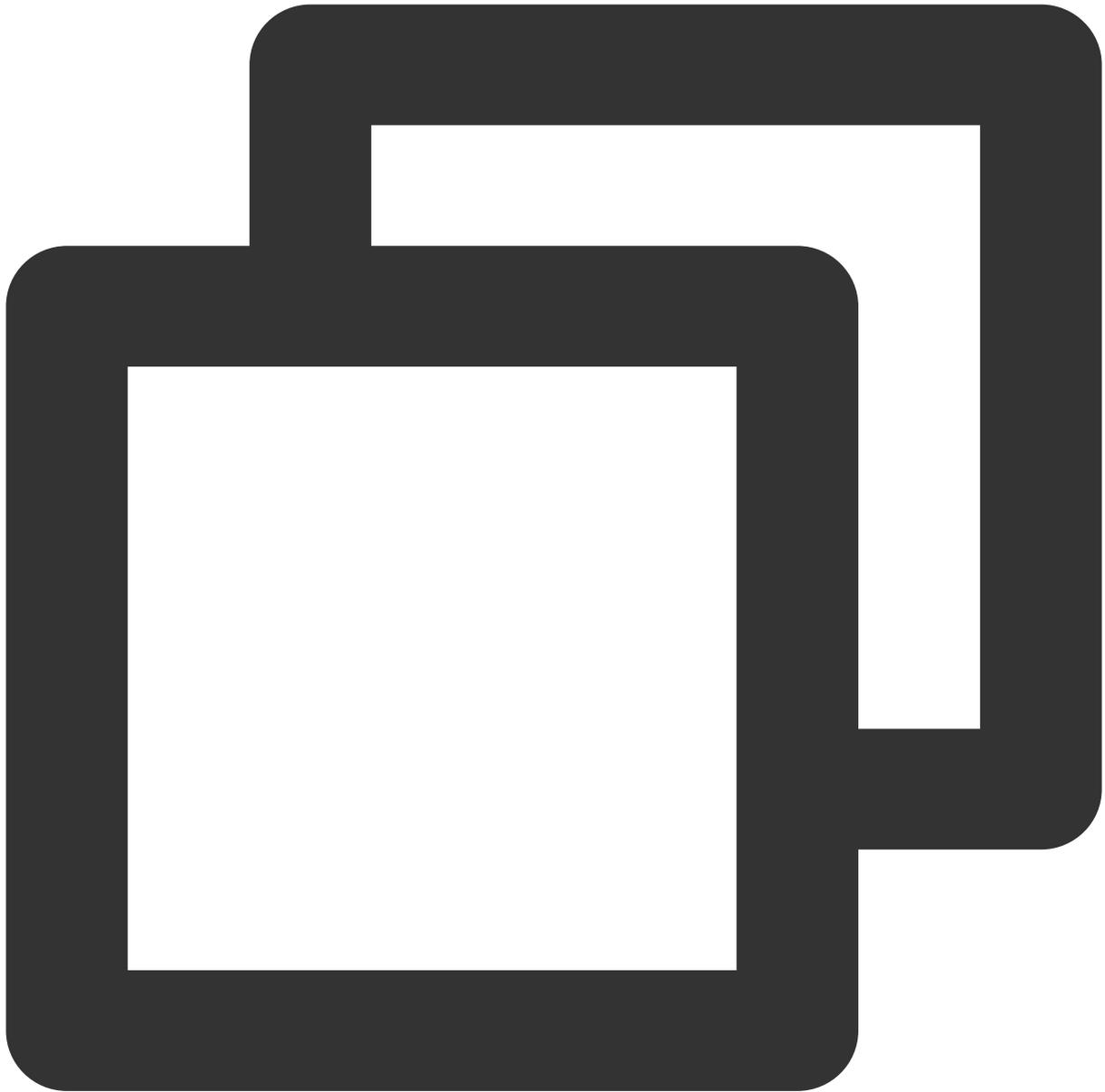
此外，您也可以迁移完成后，执行如下的命令，查看源和目的的差异文件列表：



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

环境中未配置 Hadoop-COS, 如何运行 COSDistCp?

对于环境中未配置 Hadoop-COS 插件的用户, 根据 Hadoop 版本, 下载对应版本的 COSDistCp jar 包后, 指定 Hadoop-COS 相关参数执行拷贝任务:



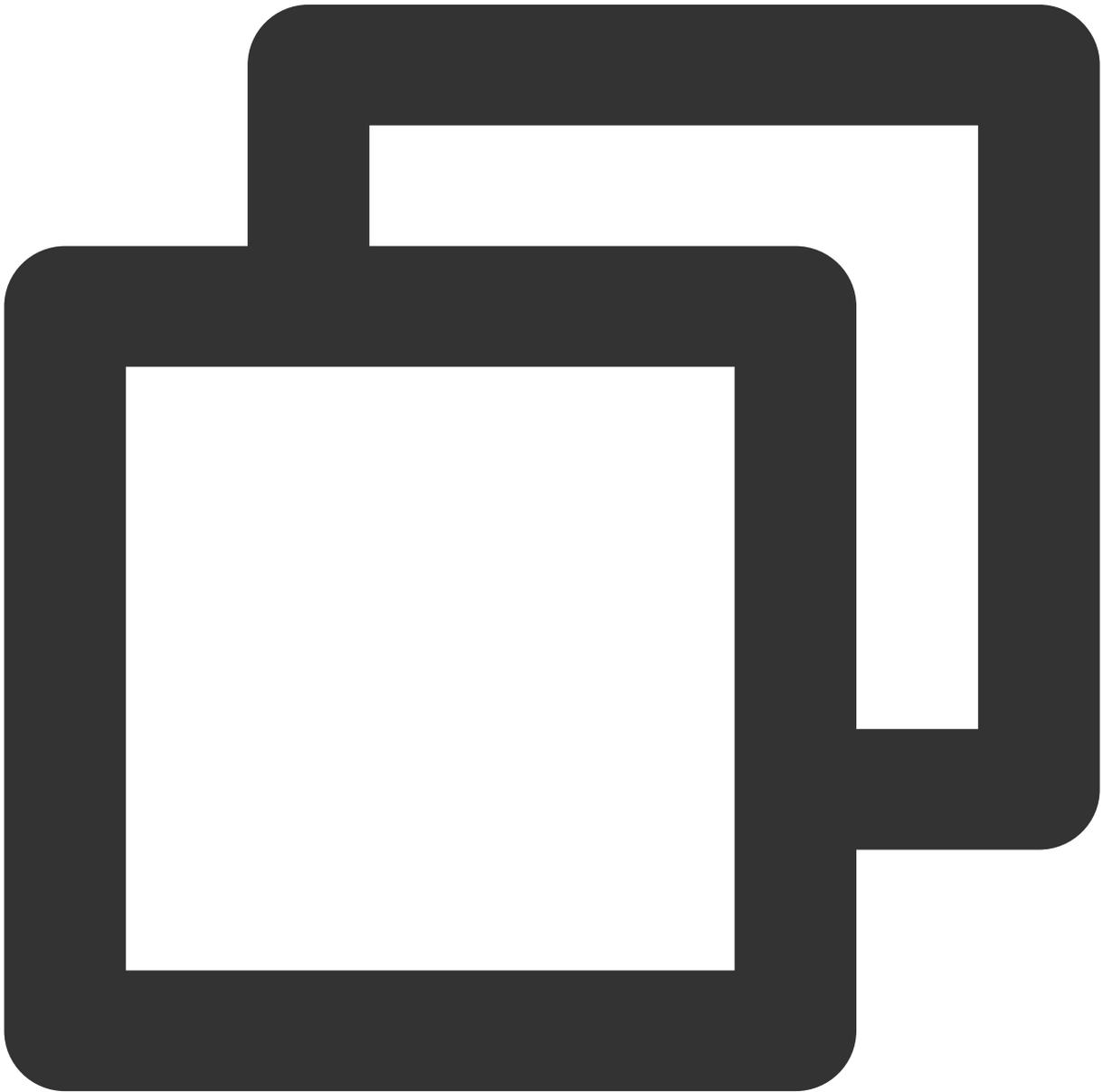
```
hadoop jar cos-distcp-${version}.jar \<\  
-Dfs.cosn.credentials.provider=org.apache.hadoop.fs.auth.SimpleCredentialProvider \<\  
-Dfs.cosn.userinfo.secretId=COS_SECRETID \<\  
-Dfs.cosn.userinfo.secretKey=COS_SECRETKEY \<\  
-Dfs.cosn.bucket.region=ap-guangzhou \<\  
-Dfs.cosn.impl=org.apache.hadoop.fs.CosFileSystem \<\  
-Dfs.AbstractFileSystem.cosn.impl=org.apache.hadoop.fs.CosN \<\  
--src /data/warehouse \<\  
--dest cosn://examplebucket-1250000000/warehouse
```

拷贝结果显示部分文件拷贝失败，如何处理？

COSDistCp 会对文件拷贝过程中出现的 `IOException` 重试五次，五次拷贝仍然失败，会将失败的文件信息写入 `/tmp/${randomUUID}/output/failed/` 目录下，其中，`${randomUUID}` 为随机字符串。常见的拷贝失败原因包括：

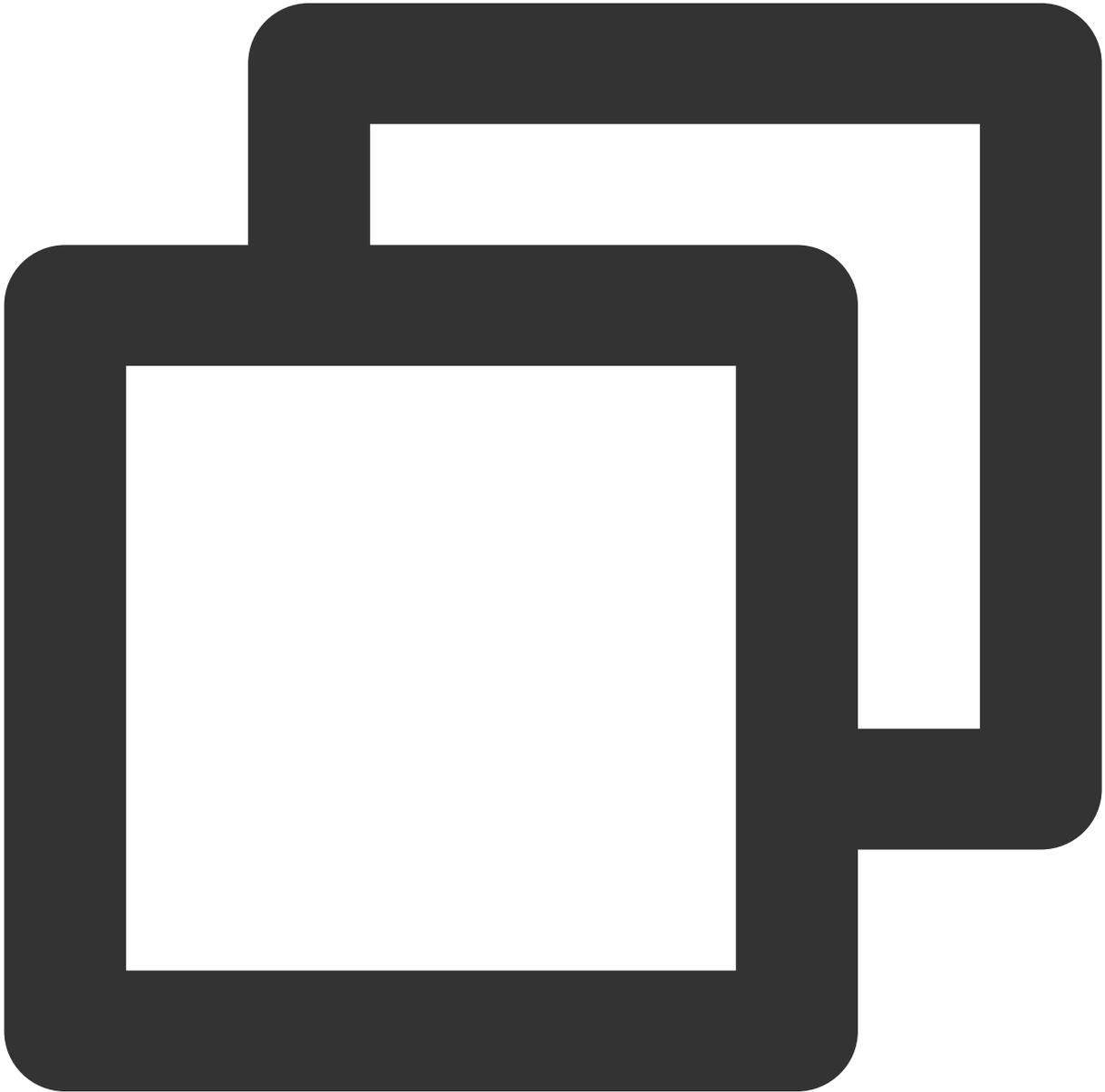
1. 源文件存在拷贝清单中，但是拷贝时源文件不存在，记录为 `SRC_MISS`。
2. 任务发起的用户，不具备读取源文件或写入目标文件的权限，以及其他原因，记录为 `COPY_FAILED`。

如果日志信息记录源文件不存在，且源文件确实可以忽略，您可以通过如下命令，获取除 `SRC_MISS` 以外的差异文件列表：



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

如果存在除 SRC_MISS 以外的失败文件，您可以根据汇总在 `/tmp/${randomUUID}/output/logs/` 目录下的异常日志信息和拉取应用日志诊断原因，例如拉取 yarn 应用的日志，可使用如下命令：



```
yarn logs -applicationId application_1610615435237_0021 > application_1610615435237
```

其中 `application_1610615435237_0021` 为应用 ID。

COSDistCp 是否会在网络等异常情况下，拷贝生成不完整文件？

在网络异常、源文件缺失和权限不足等情况下，COSDistCp 无法在目标端生成和源端同样大小的文件。

对于 COSDistCp 1.5 以下版本，COSDistCp 会尝试删除生成在目标端文件。如果删除失败，则需要您重新执行拷贝任务覆盖这些文件，或者手动删除这些不完整的文件。

对于 COSDistCp 1.5 及以上版本，且运行环境的 Hadoop COS 插件版本在 5.9.3 及以上版本时，如果拷贝到 COS 拷贝失败，COSDistCp 会调用 abort 接口终止正在上传的请求。因此，即使遇到异常情况，也不会生成不完整的文件。

对于 COSDistCp 1.5 及以上版本，如果运行环境的 Hadoop COS 插件版本低于 5.9.3，建议升级到 5.9.3 及其以上版本。

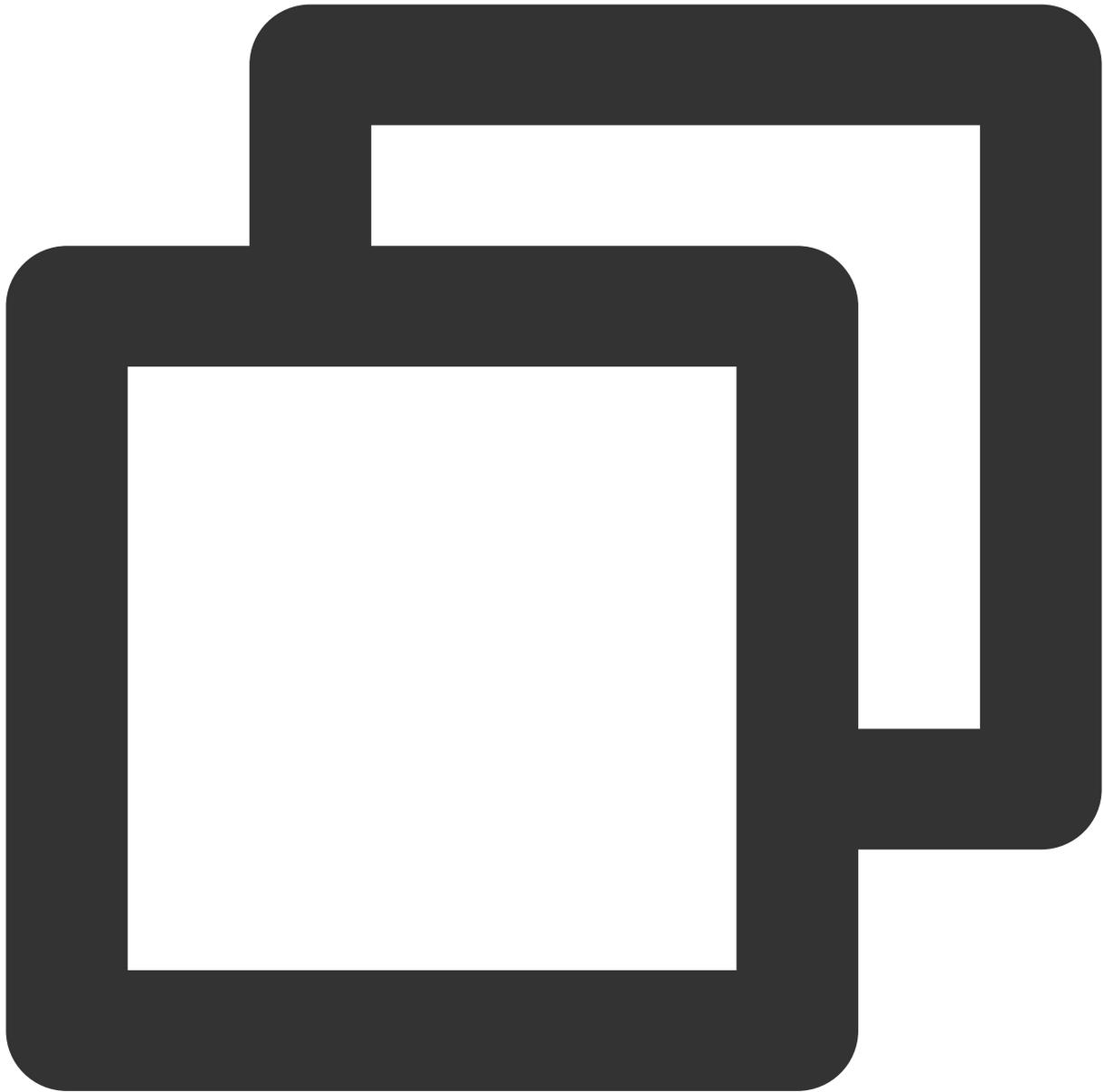
对于非 COS 的目标端，COSDistCp 会尝试删除目标端文件。

COS 存储桶中存在一些看不见的未完成上传文件，占用存储空间，如何处理？

由于机器异常、进程被 Kill 等因素，可能导致 COS 存储桶中存在一些碎片文件占用存储空间，您可参考官网 [生命周期文档](#) 配置碎片删除规则，进行清理。

迁移过程中，出现内存溢出和任务超时，如何进行参数调优？

在迁移过程中，COSDistcp 和访问 COS 和 CHDFS 的工具，基于自身逻辑，都会占用一些内存。为避免内存溢出和任务超时，您可以进行一些 MapReduce 任务的参数调整，例如：



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

其中，将任务的超时时间 `mapreduce.task.timeout` 调整为18000秒，避免拷贝超大型文件时，出现任务超时；将 Reduce 进程的内存空间 `mapreduce.reduce.memory.mb` 大小调整为8GB，避免内存溢出。

通过专线迁移，如何控制迁移任务的迁移带宽？

COSDistcp 迁移的总带宽限制计算公式为： $\text{taskNumber} * \text{workerNumber} * \text{bandWidth}$ ，您可以将 `workerNumber` 设置为 1，通过参数 `taskNumber` 控制迁移并发数，以及参数 `bandWidth` 控制单个并发的带宽。

Hadoop-cos-DistChecker 工具

最近更新时间：2024-01-06 10:43:44

功能说明

Hadoop-cos-DistChecker 是一个校验迁移目录完整性的工具。用户在使用 `hadoop distcp` 命令从 HDFS 迁移数据到 COS 上后，基于 MapReduce 的并行能力，Hadoop-cos-DistChecker 工具可以快速地进行源目录和目标目录的校验比对。

使用环境

Hadoop-cos v5.8.2以上，详见 [hadoop-cos release](#)。

Hadoop MapReduce 的运行环境。

注意

如果是自建 Hadoop 集群，则 Hadoop-cos 依赖需要选择最新版本（GitHub release 版本为5.8.2以上）才能支持 CRC64 校验码的获取。

如果是使用腾讯云 EMR 套件，则2020年5月8日后创建的集群才包含该 Hadoop-cos 版本，早于这个时间创建的集群，请 [联系我们](#)。

使用说明

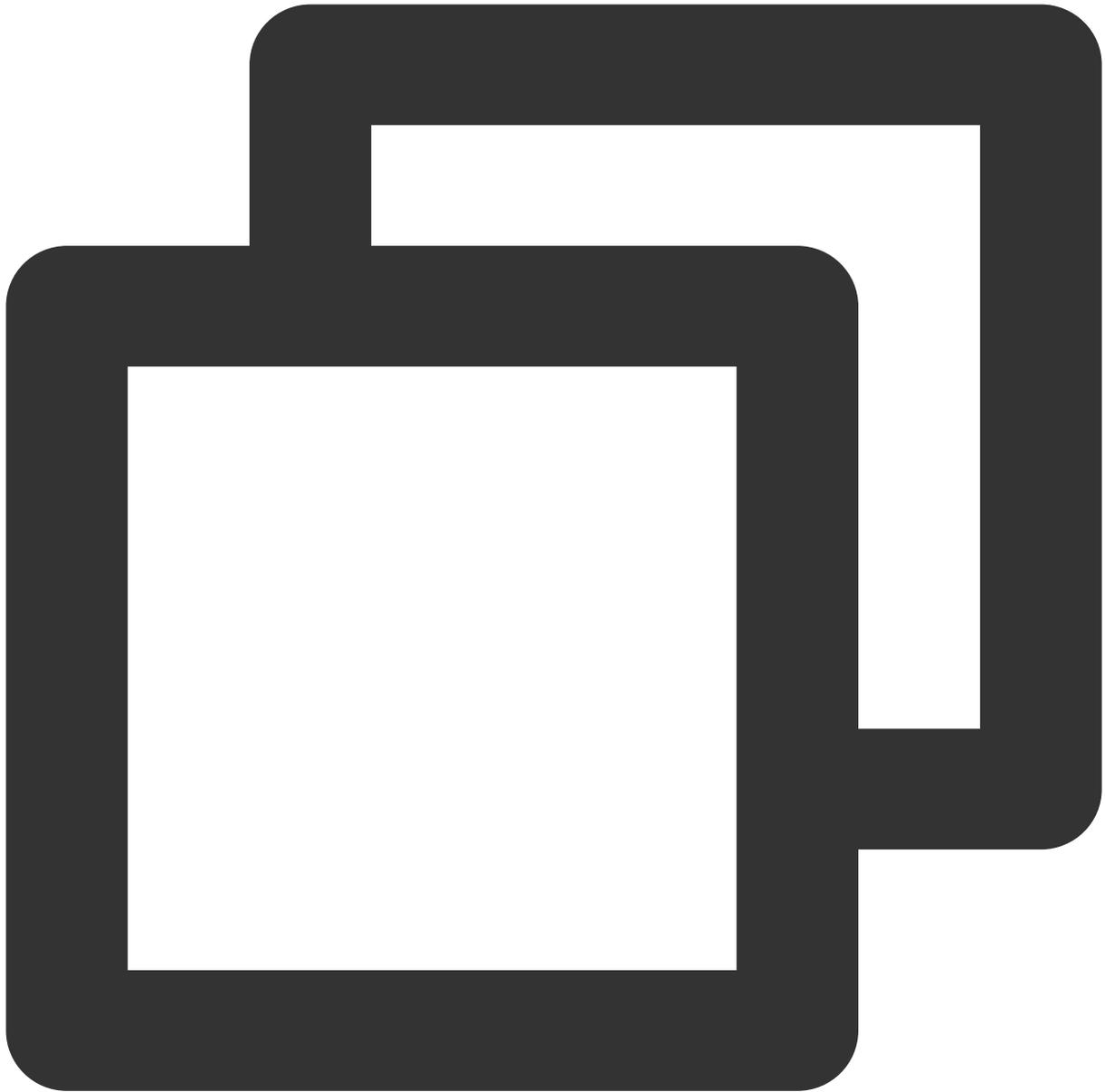
由于 Hadoop-cos-distchecker 需要获取 Hadoop-cos（CosN 文件系统）中的文件 CRC64 校验值，因此，在运行该工具以前，需要将配置项 `fs.cosn.crc64.checksum.enabled` 置为 `true` 以支持获取 Hadoop-cos 文件的 CRC64 校验和，待工具运行完成后，再将该选项置回 `false` 以关闭 CRC64校验和的获取。

注意

由于 Hadoop-COS 支持的 CRC64 校验和与 HDFS 文件系统的 CRC32C 校验和无法兼容，因此在使用完该工具以后，务必将上述配置项恢复为关闭状态，否则可能会导致 Hadoop-cos 在某些调用文件系统 `getFileChecksum` 接口的场景下运行失败。

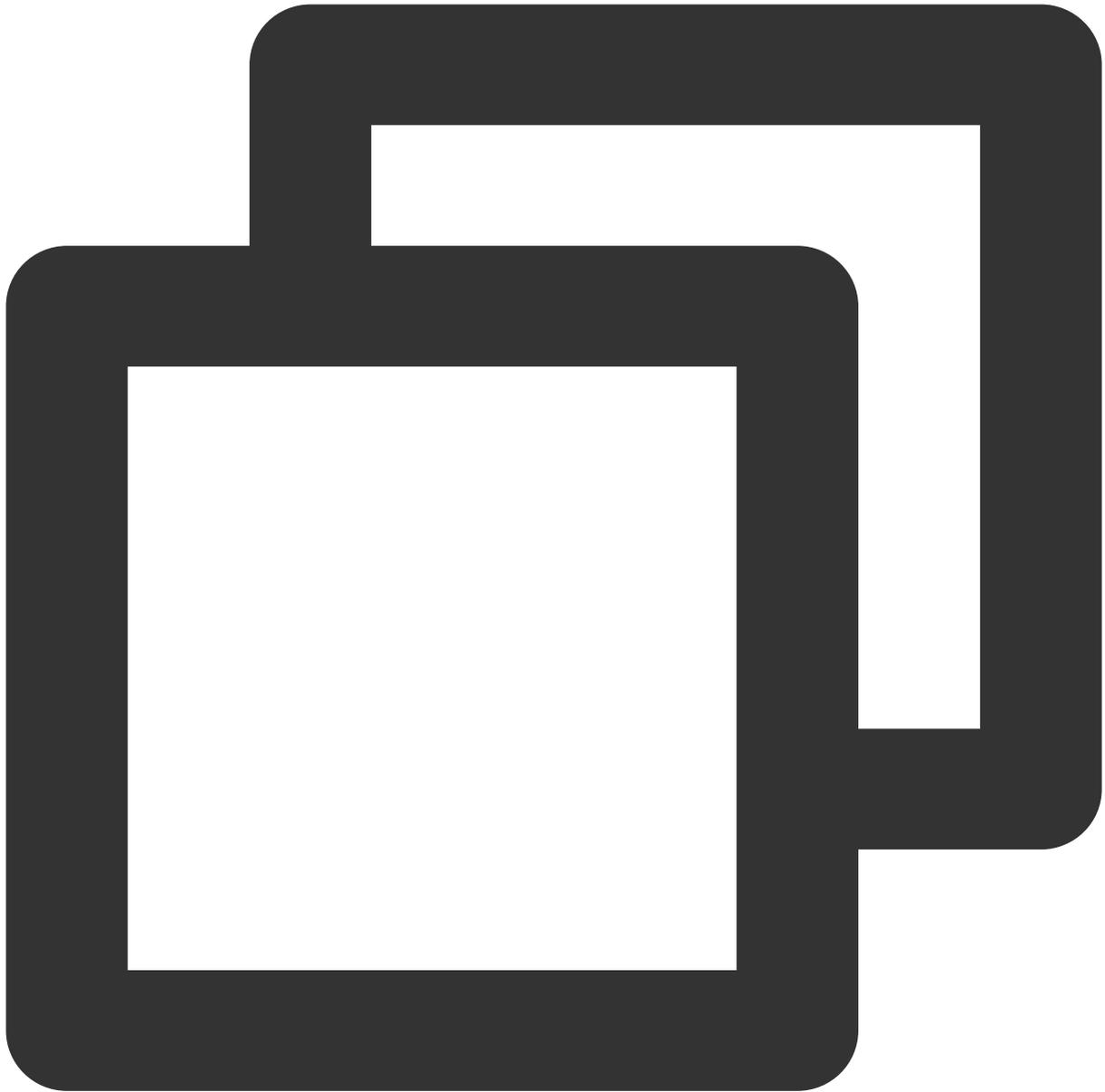
参数概念

源文件列表：源文件路径列表是用户执行以下命令，导出的待检查的子目录和文件列表。



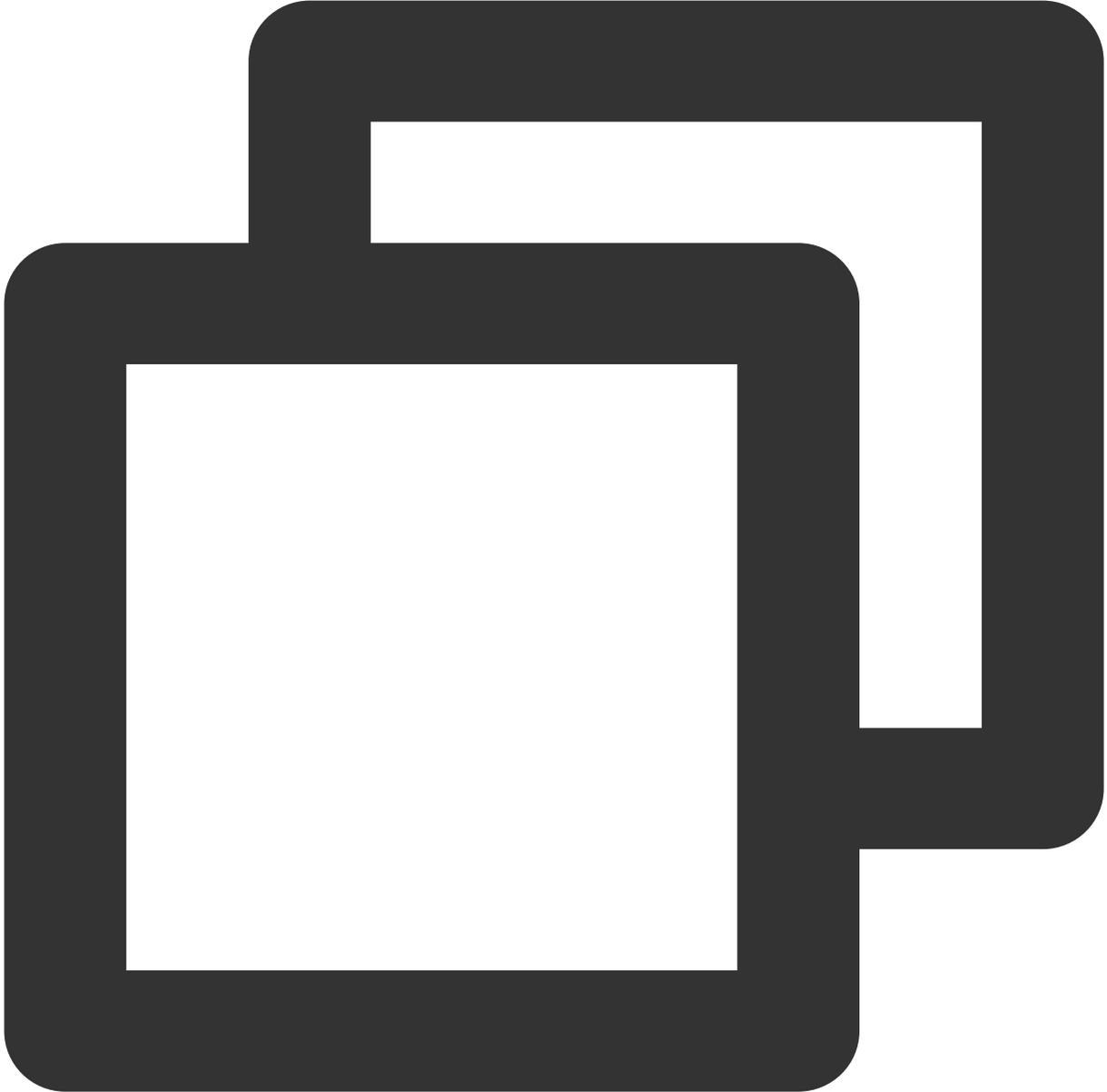
```
hadoop fs -ls -R hdfs://host:port/{source_dir} | awk '{print $8}' > check_list.txt
```

示例格式如下：



```
/benchmarks/TestDFSIO
/benchmarks/TestDFSIO/io_control
/benchmarks/TestDFSIO/io_control/in_file_test_io_0
/benchmarks/TestDFSIO/io_control/in_file_test_io_1
/benchmarks/TestDFSIO/io_data
/benchmarks/TestDFSIO/io_data/test_io_0
/benchmarks/TestDFSIO/io_data/test_io_1
/benchmarks/TestDFSIO/io_write
/benchmarks/TestDFSIO/io_write/_SUCCESS
/benchmarks/TestDFSIO/io_write/part-00000
```

源目录：指源文件列表所在的目录，这个目录通常也是 `distcp` 命令进行数据迁移时的源路径。如下所示，`hdfs://host:port/source_dir` 为源目录。



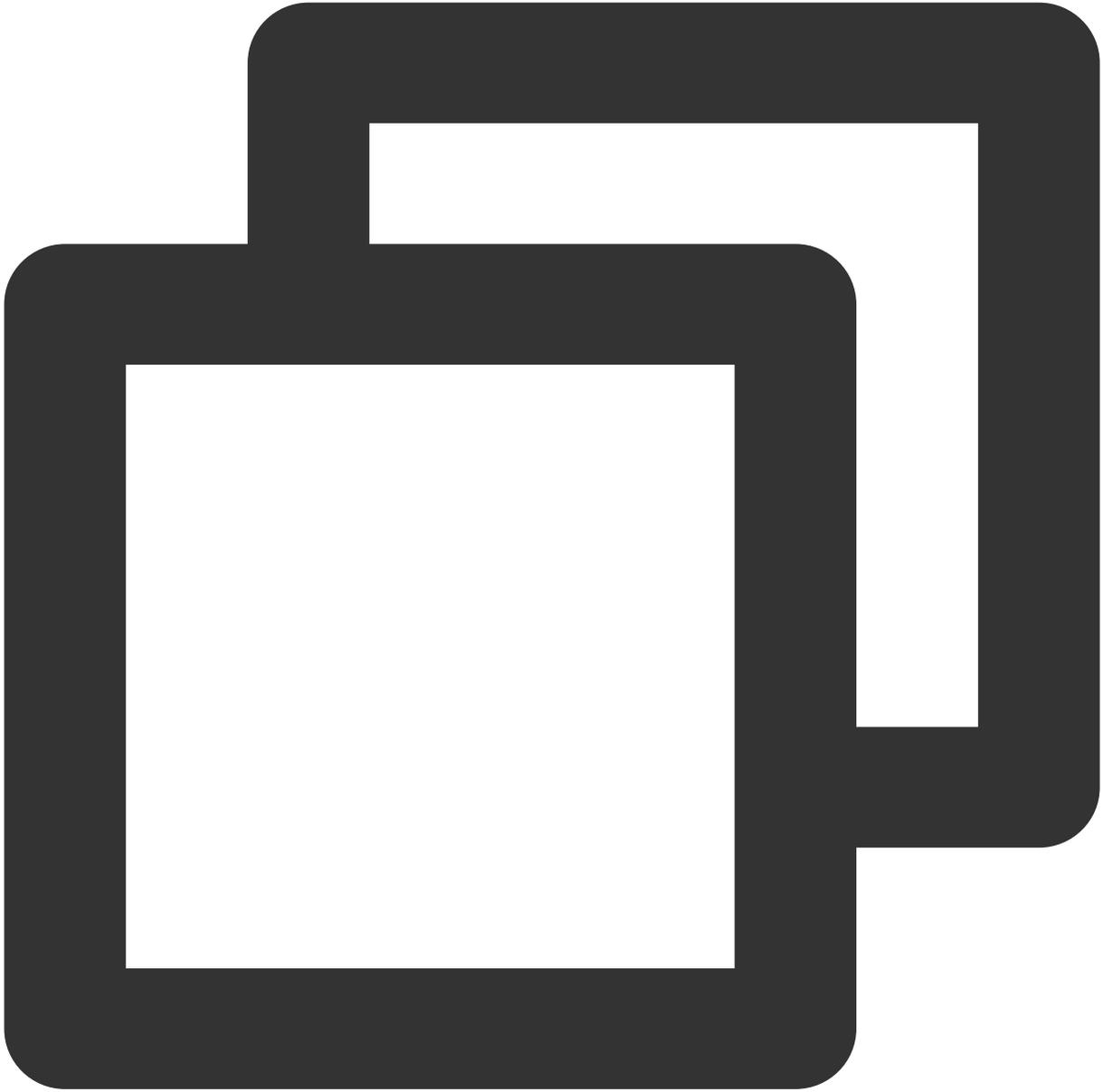
```
hadoop distcp hdfs://host:port/source_dir cosn://examplebucket-appid/dest_dir
```

此外，这个路径也是**源文件路径列表**中公共父目录，例如上述的源文件列表的公共父目录就是：`/benchmarks`。

目标目录：待比较的目标目录。

命令行格式

Hadoop-cos-DistChecker 是一个 MapReduce 作业程序，按照 MapReduce 作业的提交流程即可：



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

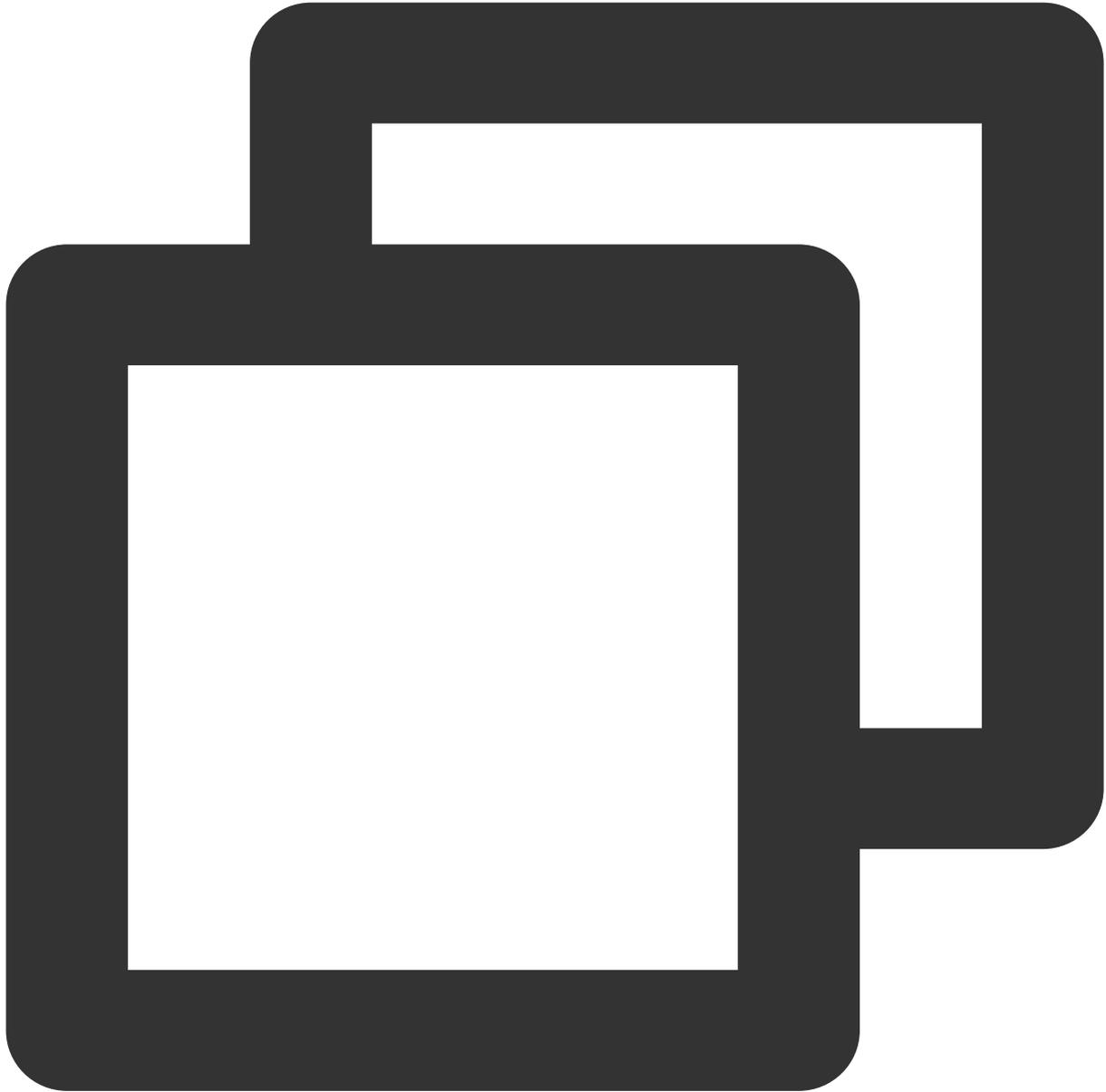
说明

Optional parameters 表示 Hadoop 可选参数。

使用步骤

下面以校验 `hdfs://10.0.0.3:9000/benchmarks` 和 `cosn://hdfs-test-1250000000/benchmarks` 为例，介绍工具的使用步骤。

首先，执行以下命令：



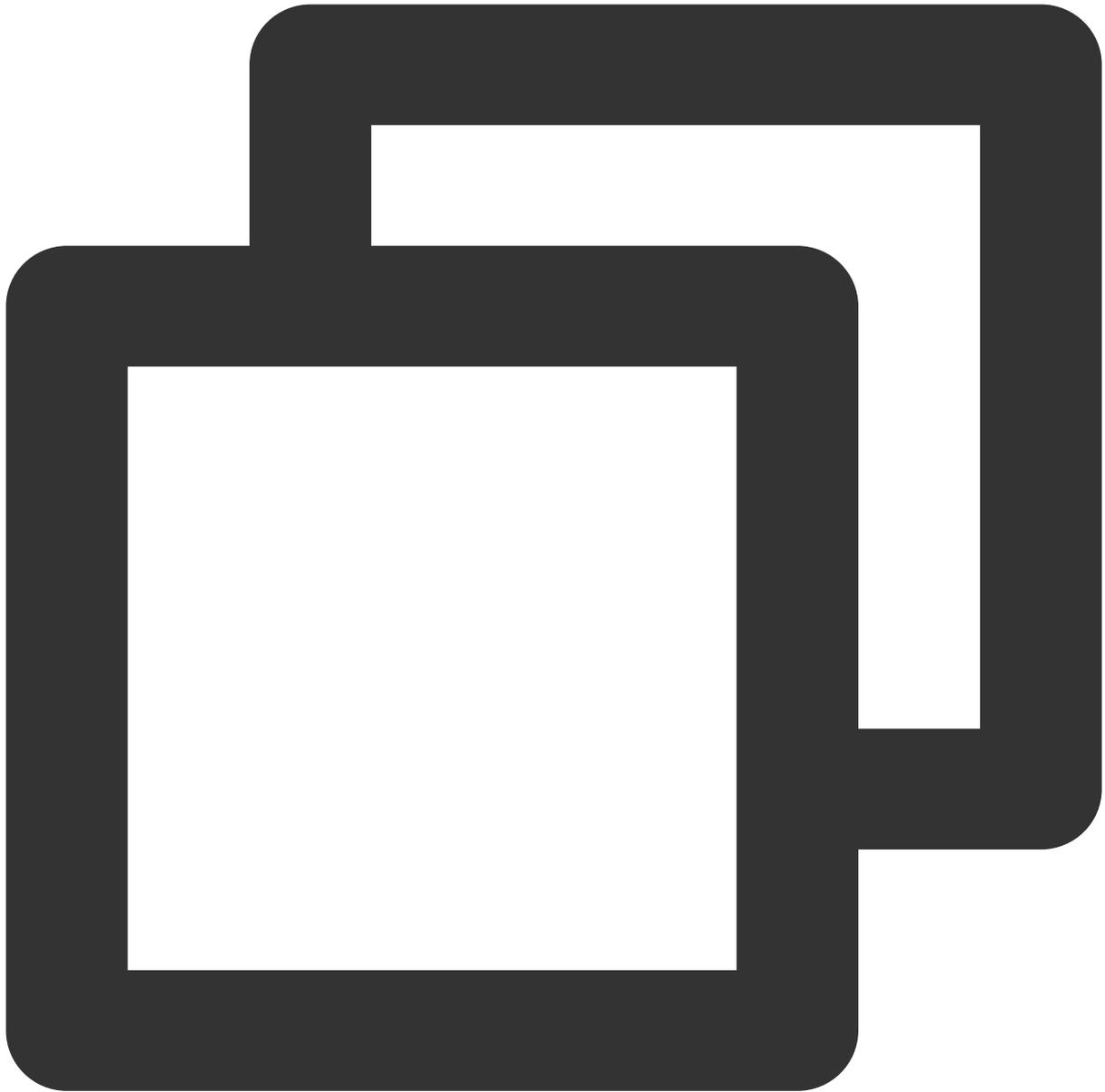
```
hadoop fs -ls -R hdfs://10.0.0.3:9000/benchmarks | awk '{print $8}' > check_list.txt
```

```
[root@VM_38_97_centos ~]# hadoop fs -ls -R hdfs://10.0.0.3:9000/benchmarks | awk '{print $8}' > check_list.txt
```

将待检查源路径导出到一个 `check_list.txt` 的文件中，这个文件里面保存的就是源文件路径列表，如下所示：

```
[root@VM_38_97_centos ~]$ cat check_list.txt
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_control
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_control/in_file_test_io_0
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_control/in_file_test_io_1
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_data
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_data/test_io_0
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_data/test_io_1
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_write
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_write/_SUCCESS
hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_write/part-00000
```

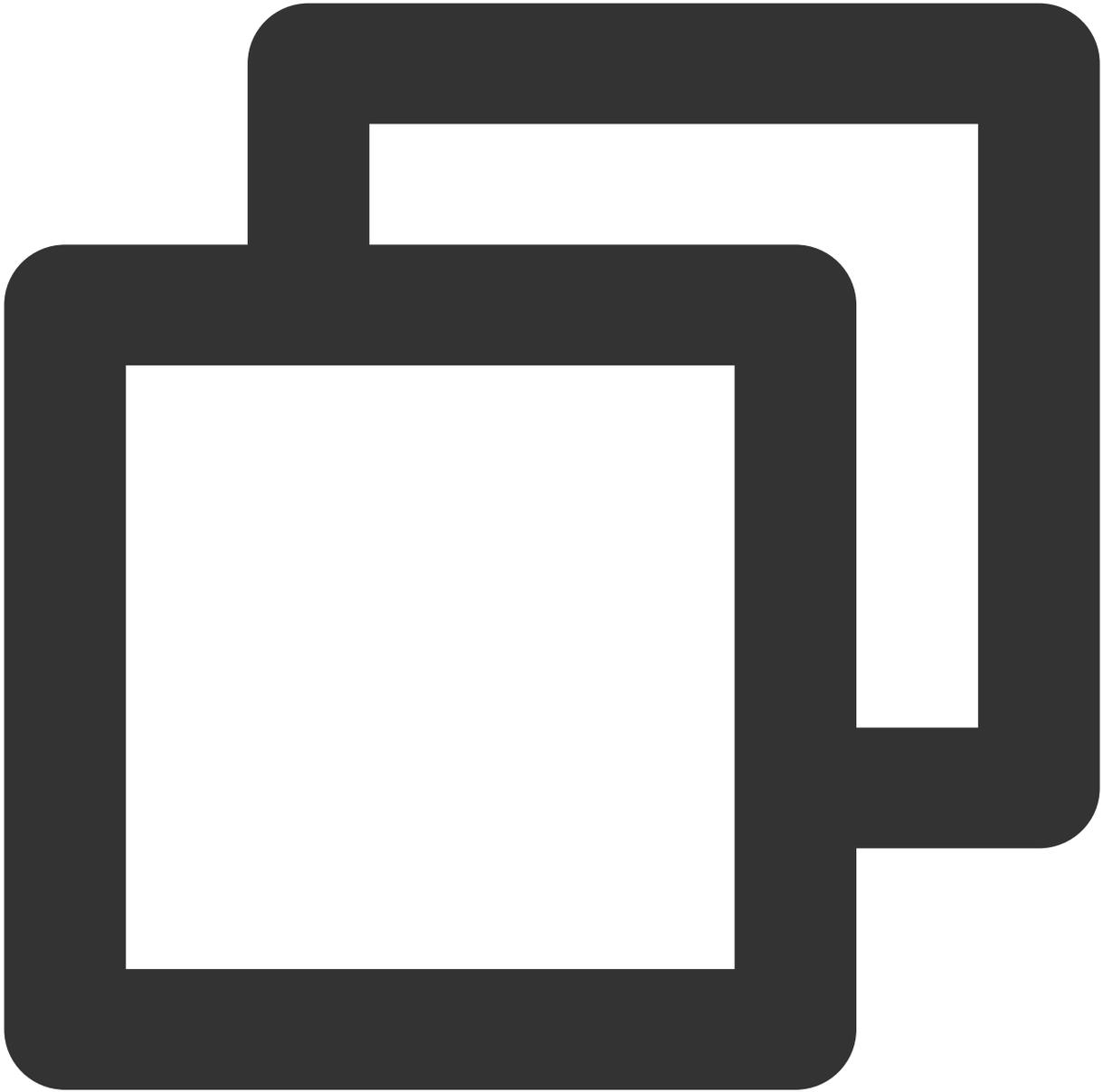
然后，将 check_list.txt 放入到 HDFS 中，执行如下：



```
hadoop fs -put check_list.txt hdfs://10.0.0.3:9000/
```

```
[root@VM_38_97_centos ~]$ hadoop fs -put check_list.txt hdfs://10.0.0.3:9000/
[root@VM_38_97_centos ~]$ hadoop fs -ls hdfs://10.0.0.3:9000/
Found 2 items
drwxr-xr-x  - ia supergroup          0 2020-02-20 22:21 hdfs://10.0.0.3:9000/benchmarks
-rw-r--r--  2 ia supergroup        580 2020-02-21 17:59 hdfs://10.0.0.3:9000/check_list.txt
```

最后，执行 Hadoop-cos-DistChecker，将 `hdfs://10.0.0.3:9000/benchmarks` 和 `cosn://hdfs-test-1250000000/benchmarks` 进行对比，然后输出结果保存到 `cosn://hdfs-test-1250000000/check_result` 路径下，命令格式如下：



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

```

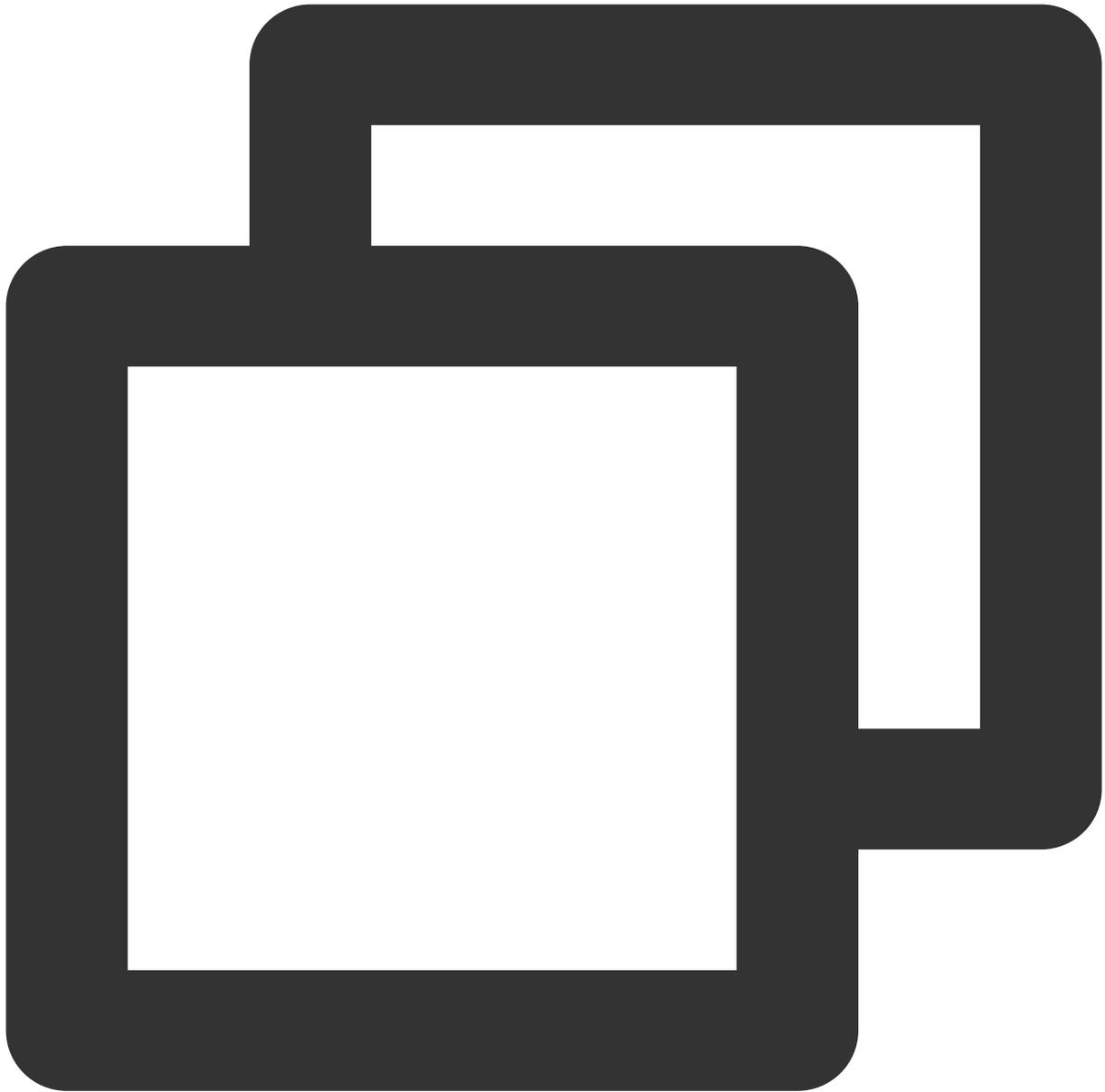
[~]# @VM_38_97_centos ~]$ hadoop jar hadoop-cos-distchecker-2.8.5-1.0-SNAPSHOT.jar com.qcloud.cos.hadoop.distchecker.App hdfs://10.0.0.3:9000/check_list.txt hdfs://10.0.0.3:9000/benchmarks cosn://hdfs-test-125a.../benchmarks cosn://h
dfs-test-125a.../check_result
20/02/21 18:08:55 INFO fs.BufferPool: The type of the upload buffer pool is [MAPPED_DISK]. Buffer size:[134217728]
20/02/21 18:08:55 INFO fs.BufferPool: Initialize the MAPPED_DISK buffer pool. size: 16
20/02/21 18:08:55 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
20/02/21 18:08:56 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
20/02/21 18:08:56 INFO input.FileInputFormat: Total input files to process : 1
20/02/21 18:08:56 INFO mapreduce.JobSubmitter: number of splits:1
20/02/21 18:08:57 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local2054621574_0001
20/02/21 18:08:57 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
20/02/21 18:08:57 INFO mapreduce.Job: Running job: job_local2054621574_0001
20/02/21 18:08:57 INFO mapred.LocalJobRunner: OutputCommitter set in config null
20/02/21 18:08:57 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
20/02/21 18:08:57 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup_temporary folders under output directory:false, ignore cleanup failures: false
20/02/21 18:08:57 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
20/02/21 18:08:58 INFO mapred.LocalJobRunner: Waiting for map tasks
20/02/21 18:08:58 INFO mapred.LocalJobRunner: Starting task: attempt_local2054621574_0001_m_000000_0
20/02/21 18:08:58 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
20/02/21 18:08:58 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]
20/02/21 18:08:58 INFO mapred.MapTask: Processing split: hdfs://10.0.0.3:9000/check_list.txt:0:4580
20/02/21 18:08:58 INFO mapreduce.Job: Job job_local2054621574_0001 running in uber mode : false
20/02/21 18:08:58 INFO mapreduce.Job: map 0% reduce 0%
20/02/21 18:08:58 INFO fs.CosFsDataOutputStream: The consistency checker is enabled.
20/02/21 18:08:58 INFO distchecker.CheckMapper: The target file path: cosn://hdfs-test-125a.../benchmarks/TestDFSIO/
20/02/21 18:08:59 INFO distchecker.CheckMapper: The target file path: cosn://hdfs-test-125a.../benchmarks/TestDFSIO/io_control/
20/02/21 18:08:59 INFO distchecker.CheckMapper: The target file path: cosn://hdfs-test-125a.../benchmarks/TestDFSIO/io_control/in_file_test_io_0
20/02/21 18:08:59 INFO distchecker.CheckMapper: Comparing the MD5 Hash between the source file [hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_control/in_file_test_io_0] and dest file [cosn://hdfs-test-125a.../benchmarks/TestDFSIO/io_control/in_file_test_io_0]
20/02/21 18:08:59 INFO distchecker.CheckMapper: finish compute the md5 hash.
20/02/21 18:08:59 INFO distchecker.CheckMapper: The target file path: cosn://hdfs-test-125a.../benchmarks/TestDFSIO/io_control/in_file_test_io_1
20/02/21 18:08:59 INFO distchecker.CheckMapper: Comparing the MD5 Hash between the source file [hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_control/in_file_test_io_1] and dest file [cosn://hdfs-test-125a.../benchmarks/TestDFSIO/io_control/in_file_test_io_1]
20/02/21 18:08:59 INFO distchecker.CheckMapper: finish compute the md5 hash.
20/02/21 18:08:59 INFO distchecker.CheckMapper: The target file path: cosn://hdfs-test-125a.../benchmarks/TestDFSIO/io_data/
20/02/21 18:08:59 INFO distchecker.CheckMapper: The target file path: cosn://hdfs-test-125a.../benchmarks/TestDFSIO/io_data/test_io_0
20/02/21 18:08:59 INFO distchecker.CheckMapper: Comparing the CRC64 between the source file [hdfs://10.0.0.3:9000/benchmarks/TestDFSIO/io_data/test_io_0] and the target file [cosn://hdfs-test-125a.../benchmarks/TestDFSIO/io_data/test_io_0]
20/02/21 18:09:10 INFO mapred.LocalJobRunner: map > map
    
```

Hadoop-cos-DistChecker 会读取源文件列表和源目录执行 MapReduce 作业，进行分布式检查，最后的检查报告会输出到 `cosn://examplebucket-appid/check_result` 路径下。

```

[~]# @VM_38_97_centos ~]$ hadoop fs -ls -R cosn://hdfs-test-125a.../check_result
20/02/21 17:37:54 INFO fs.BufferPool: The type of the upload buffer pool is [MAPPED_DISK]. Buffer size:[134217728]
20/02/21 17:37:54 INFO fs.BufferPool: Initialize the MAPPED_DISK buffer pool. size: 16
-rw-rw-rw- 1 ia... ia... 0 2020-02-21 17:36 cosn://hdfs-test-125a.../check_result/SUCCESS
-rw-rw-rw- 1 ia... ia... 1690 2020-02-21 17:36 cosn://hdfs-test-125a.../check_result/part-m-00000
20/02/21 17:37:55 INFO fs.BufferPool: Close a buffer pool instance.
20/02/21 17:37:55 INFO fs.BufferPool: Begin to release the buffers.
    
```

检查报告如下：

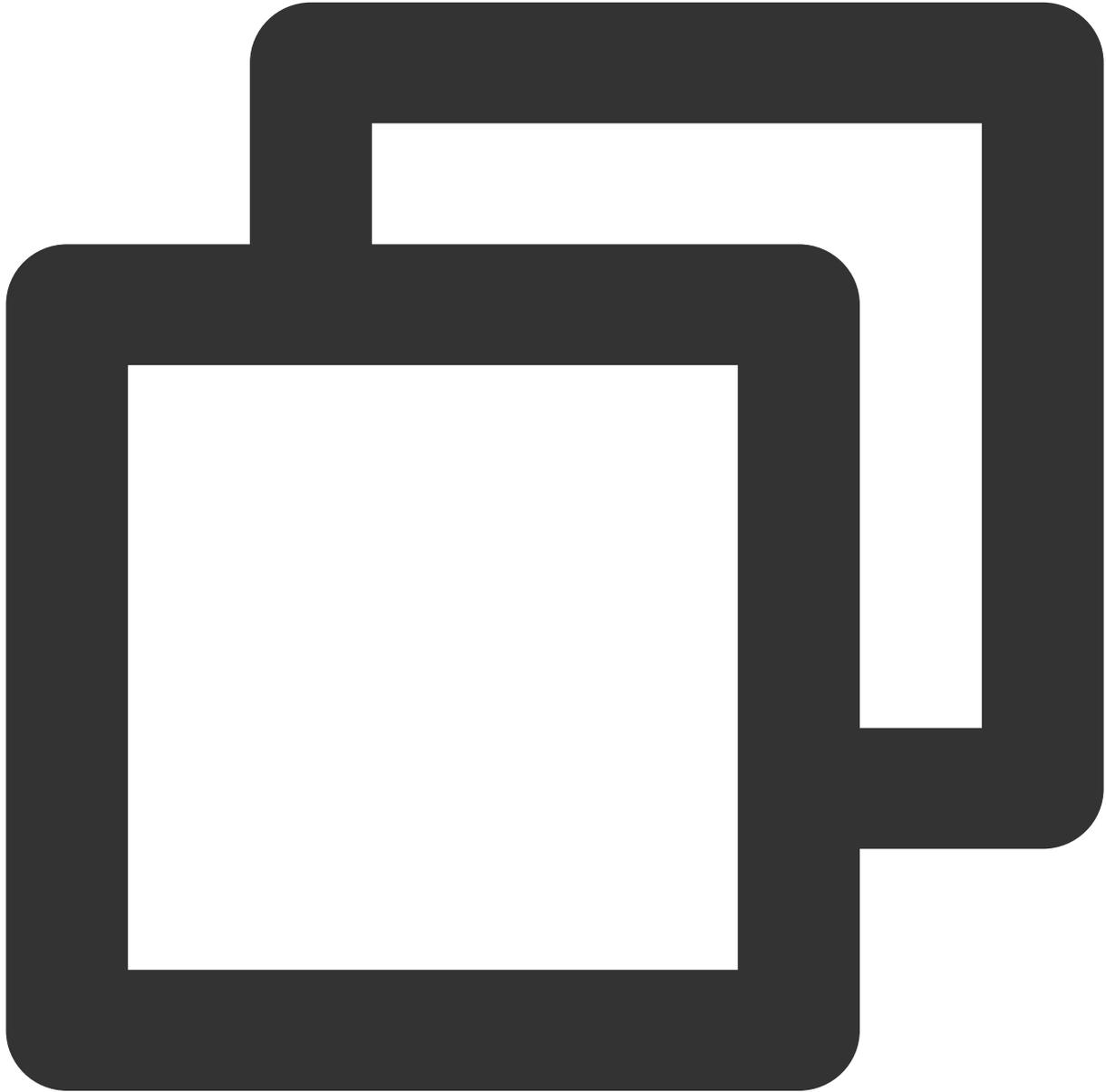


```

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

检查报告格式

检查报告是以如下格式展示：



`check_list.txt`中的源文件路径 源文件绝对路径,目标文件绝对路径,Checksum算法,源文件的checksum值,

其中检查结果分为以下7种：

SUCCESS：表示源文件和目标文件都存在，且一致。

MISMATCH：表示源文件和目标文件都存在，但不一致。

UNCONFIRM：无法确认源文件和目标文件是否一致，这种状态主要是由于 COS 上的文件可能是 CRC64 校验码特性上线前就存在的文件，无法获取到其 CRC64 的校验值。

UNCHECKED：未检查。这种状态主要是由于源文件无法读取或无法获取源文件的 checksum 值。

SOURCE_FILE_MISSING：源文件不存在。

TARGET_FILE_MISSING：目标文件不存在。

TARGET_FILESYSTEM_ERROR：目标文件系统不是 CosN 文件系统。

常见问题

为什么检查报告的 CRC64 值出现负数？

因为 CRC64 值，有可能是20位的值，此时已超过 Java long 型的表示范围，但是其底层字节是一致的，而打印 long 型时，会出现负数表示。

HDFS TO COS 工具

最近更新时间：2024-01-06 10:43:44

功能说明

HDFS TO COS 工具用于将 HDFS 上的数据拷贝到腾讯云对象存储（Cloud Object Storage，COS）上。

使用环境

系统环境

Linux 或 Windows 系统。

软件依赖

JDK 1.7或1.8。

安装与配置

具体环境安装与配置请参见 [Java 安装与配置](#)。

配置方法

1. 安装 Hadoop-2.7.2 及以上版本，具体安装步骤请参见 [Hadoop 安装与测试](#)。
2. 在 [GitHub](#) 下载 HDFS TO COS 工具并解压缩。
3. 将要同步的 HDFS 集群的 core-site.xml 拷贝到 conf 文件夹中，其中 core-site.xml 中包含 NameNode 的配置信息。
4. 编辑配置文件 cos_info.conf，存储桶（Bucket）、地域（Region）以及 API 密钥信息，其中存储桶名称，由用户自定义字符串和系统生成 APPID 数字串以及中划线连接而成，例如 examplebucket-1250000000。
5. 在命令行参数中指定配置文件位置，默认位置 conf/cos_info.conf。

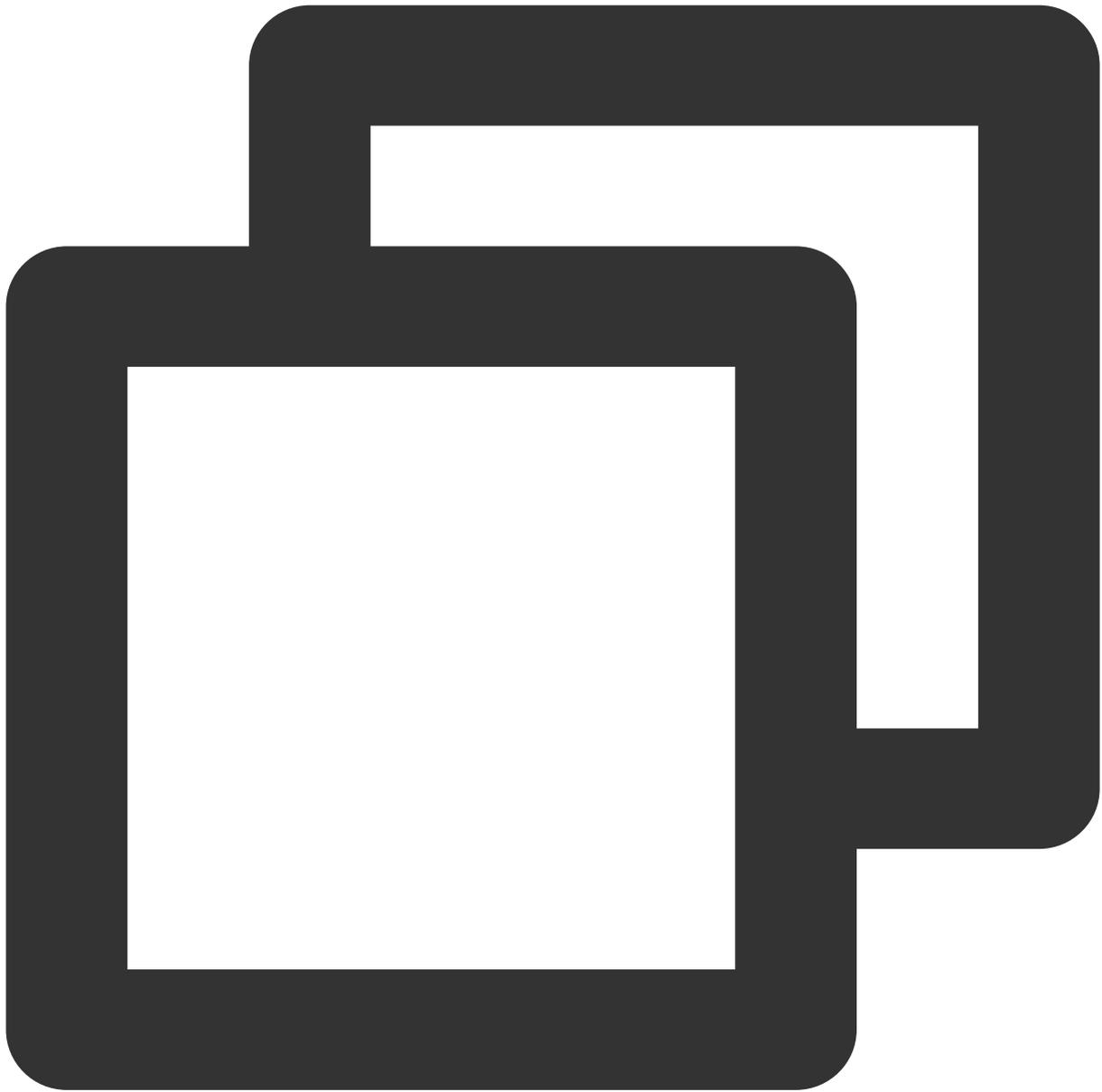
注意

当命令行参数中的参数与配置文件重合时，以命令行为准。

使用方法

说明

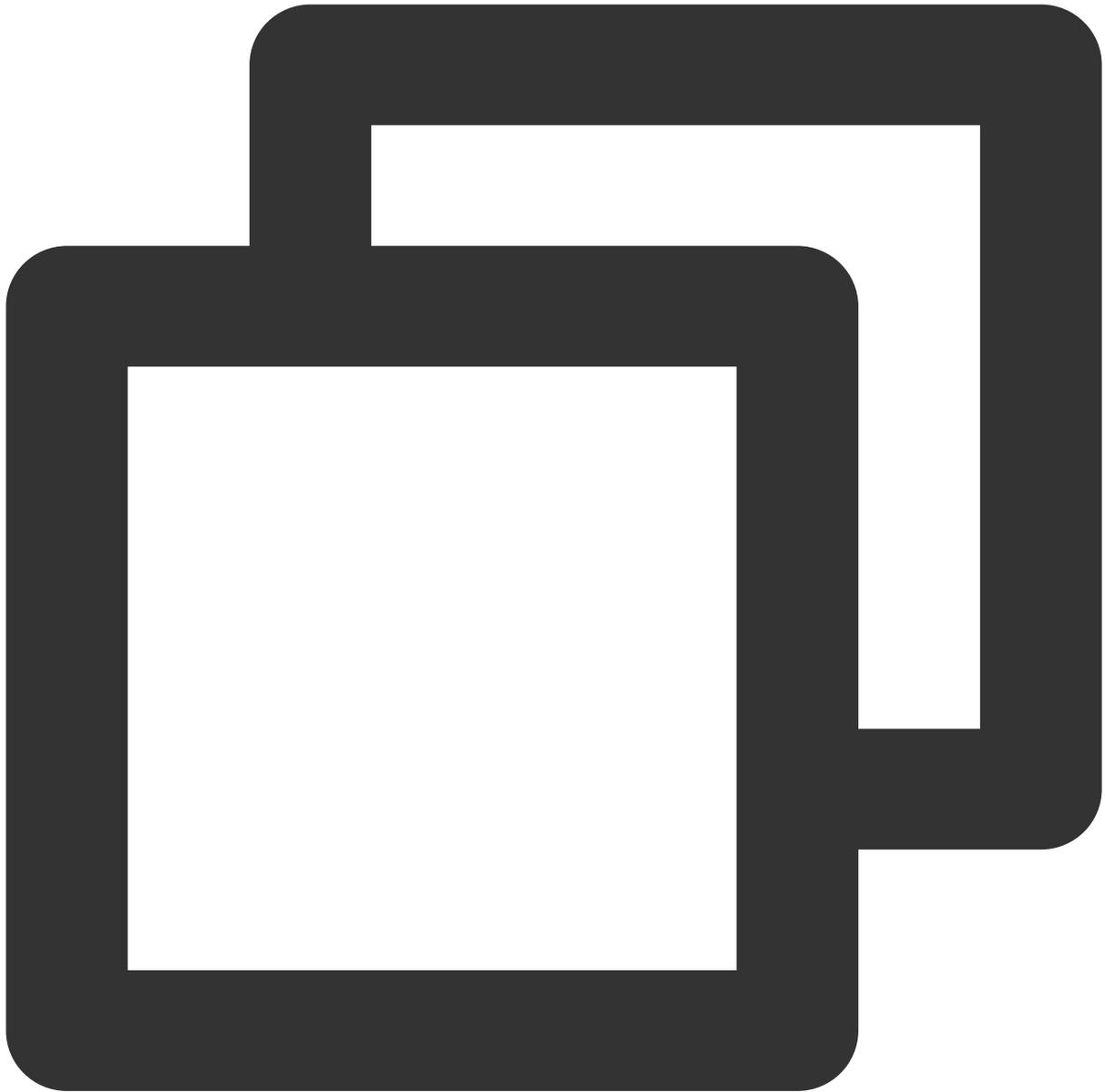
下面以 Linux 为例介绍如何使用。

[查看帮助](#)

```
./hdfs_to_cos_cmd -h
```

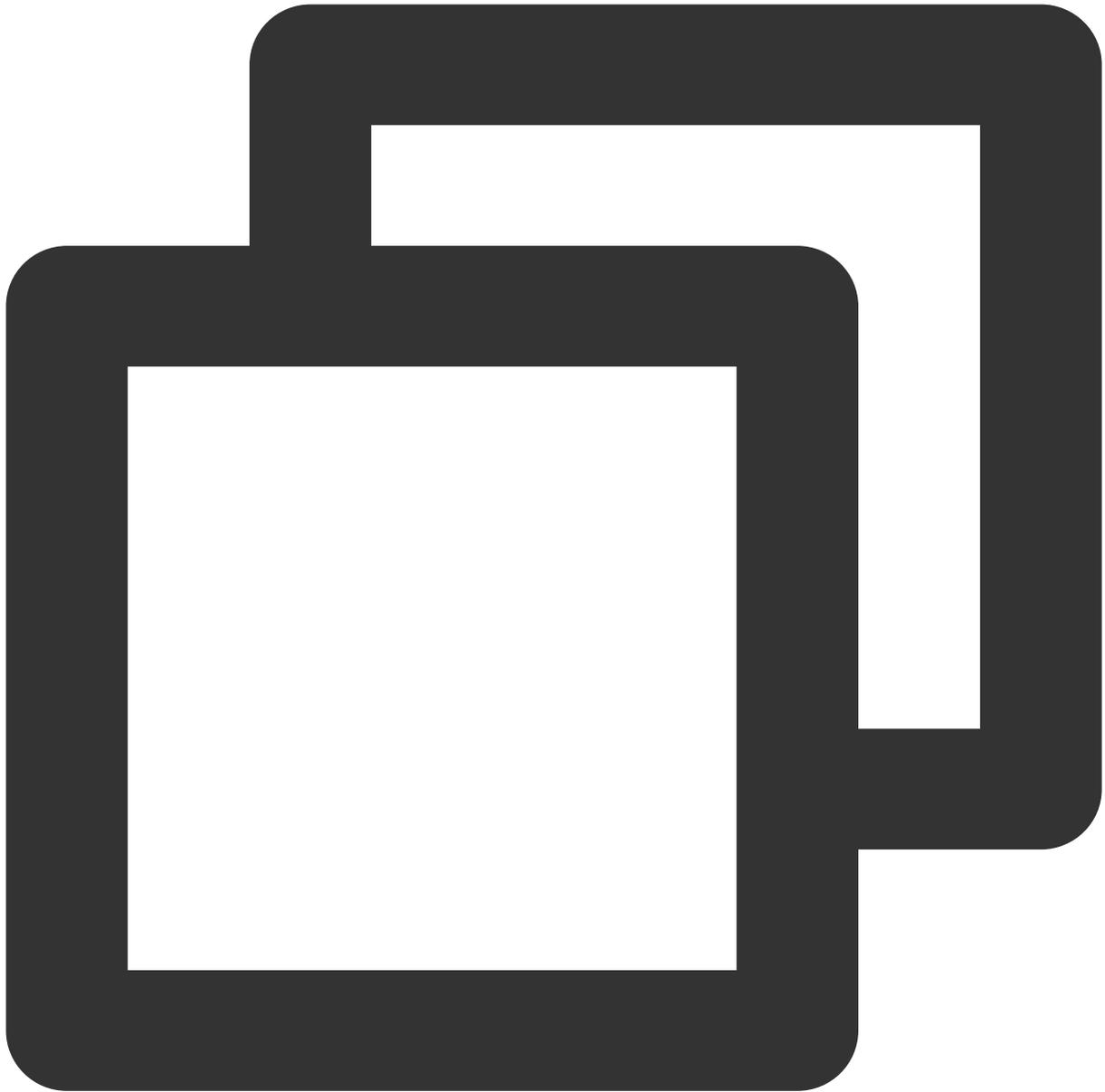
文件拷贝

从 HDFS 拷贝到 COS，若 COS 上已存在同名文件，则会覆盖原文件。



```
./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/
```

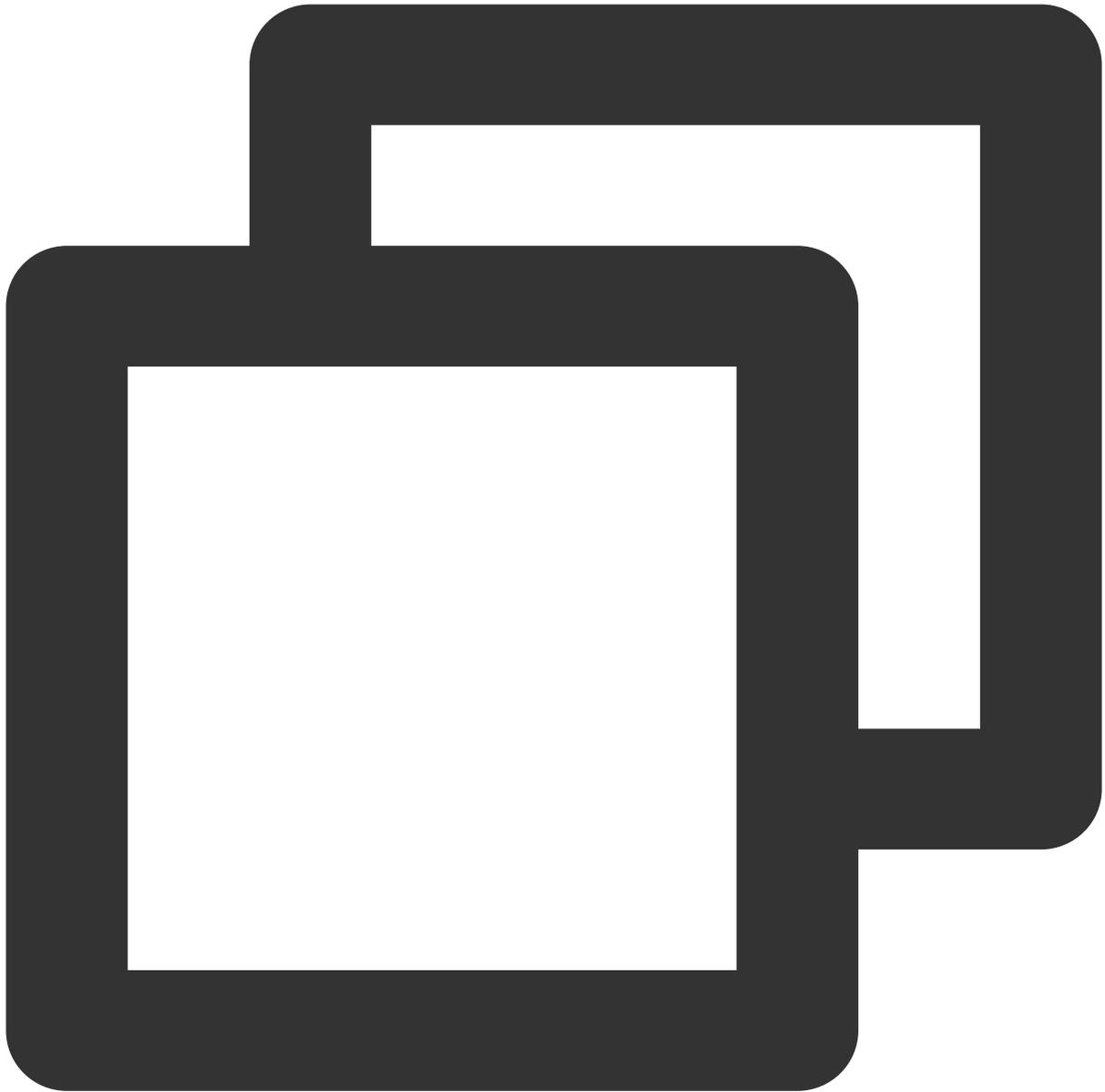
从 HDFS 拷贝到 COS，若 COS 上已存在同名且长度一致的文件时，则忽略上传（适用于拷贝一次后，重新拷贝）。



```
./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/ -skip_if_len_mat
```

这里只对长度进行判断，因为如果将 Hadoop 上的文件摘要进行计算，开销较大。

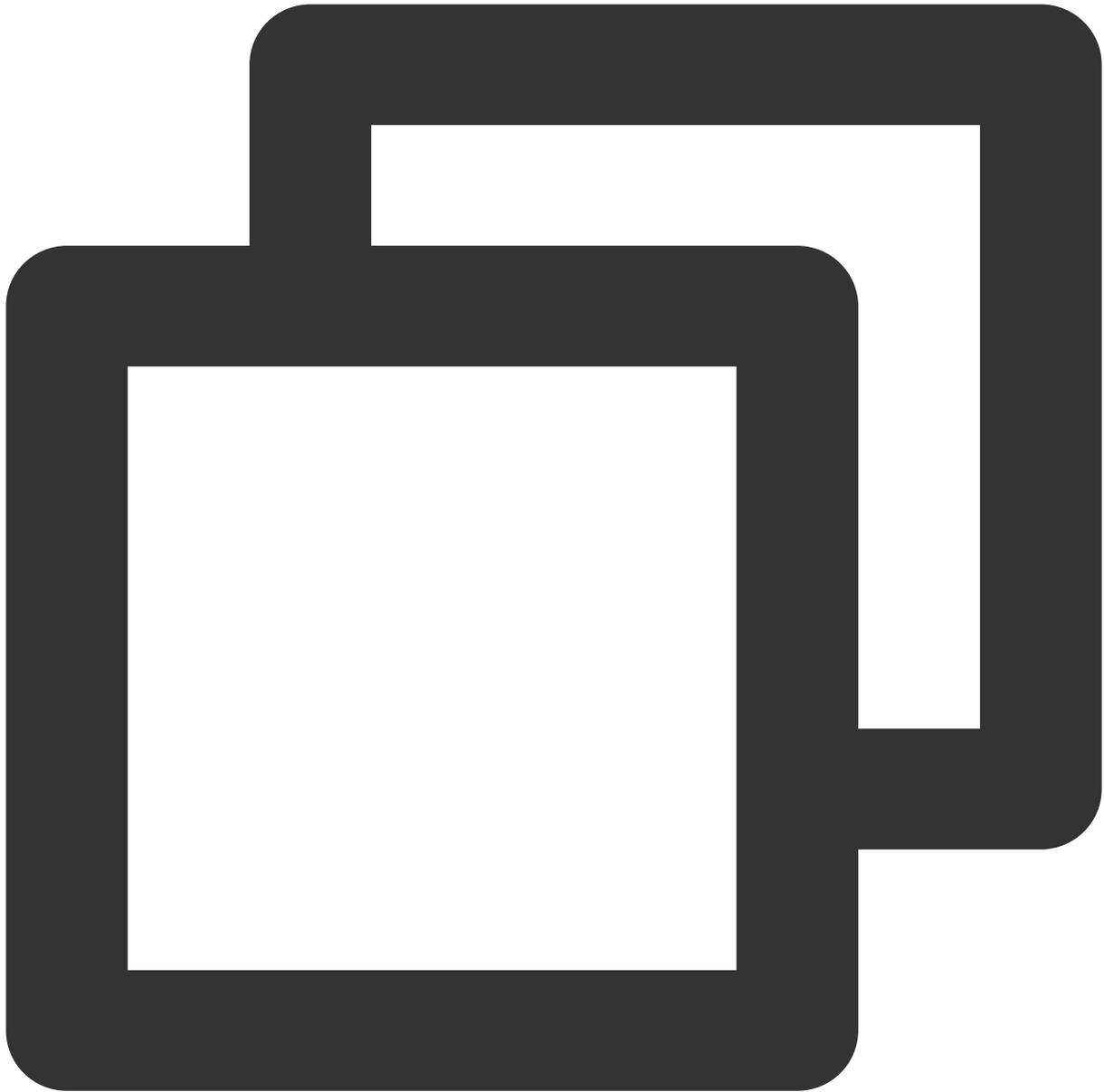
从 HDFS 拷贝到 COS，若 HDFS 中存在 Har 目录（Hadoop Archive 归档文件），通过指定 `--decompress_har` 参数可以自动解压 har 文件：



```
./hdfs_to_cos_cmd --decompress_har --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/
```

若未指定 `--decompress_har` 参数，默认按照普通的 HDFS 目录进行拷贝，即 `.har` 目录下的 `index` 和 `masterindex` 等文件原样拷贝。

目录信息



```
conf : 配置文件, 用于存放 core-site.xml 和 cos_info.conf  
log  : 日志目录  
src  : Java 源程序  
dep  : 编译生成的可运行的 JAR 包
```

问题与帮助

[关于配置信息](#)

请确保填写的配置信息正确，包括存储桶（Bucket）、地域（Region）以及 API 密钥信息，其中，存储桶的名字，由用户自定义字符串和系统生成 APPID 数字串由中划线连接而成，例如 examplebucket-1250000000。并保证机器的时间和本地时间一致（相差1分钟左右是正常的），如果相差较大，请重新设置机器时间。

关于 DataNode

请保证对于 DataNode，拷贝程序所在的机器也可以连接。NameNode 有外网 IP 可以连接，但获取的 block 所在的 DataNode 机器是内网 IP，是无法直接连接上的。因此建议同步程序放在 Hadoop 的某个节点上执行，保证对 NameNode 和 DataNode 皆可访问。

关于权限

请使用 Hadoop 命令下载文件，检查是否正常，再使用同步工具同步 Hadoop 上的数据支持。

关于文件覆盖

对于 COS 上已存在的文件，默认进行重传覆盖。除非用户明确的指定 `-skip_if_len_match`，当文件长度一致时，则跳过上传。

关于 cos path

cos path 默认为是目录，最终从 HDFS 上拷贝的文件都会存放在该目录下。

关于从腾讯云 EMR HDFS 拷贝数据

从腾讯云 EMR HDFS 拷贝数据到 COS，建议使用高性能 Distcp 工具，请参见 [Hadoop 文件系统与 COS 之间的数据迁移](#)。

在线辅助工具

COS 请求工具

最近更新时间：2024-01-06 10:43:44

功能概述

COS 请求工具是腾讯云对象存储（Cloud Object Storage，COS）为您提供的 Web 端调试工具，集成在云 API 3.0 Explorer 平台上，可用于 API 接口的调试工作。

注意

COS 请求工具发送的请求将被真实发送到 COS 的业务服务器，**所有操作均等同于真实操作，当您选择 DELETE 类操作时请谨慎操作。**

目前 COS 请求工具所支持的 API 版本为 XML API，不支持 JSON API 版本。

JSON API 是腾讯云 COS 服务在推出 XML API 前为用户提供接入使用 COS 的 API 接口，接口与标准 XML 的 API 底层架构相同，数据互通，可以交叉使用，但是接口不兼容。

XML API 具有更加丰富的功能和特性，腾讯云对象存储强烈建议您升级使用 XML API 版本。

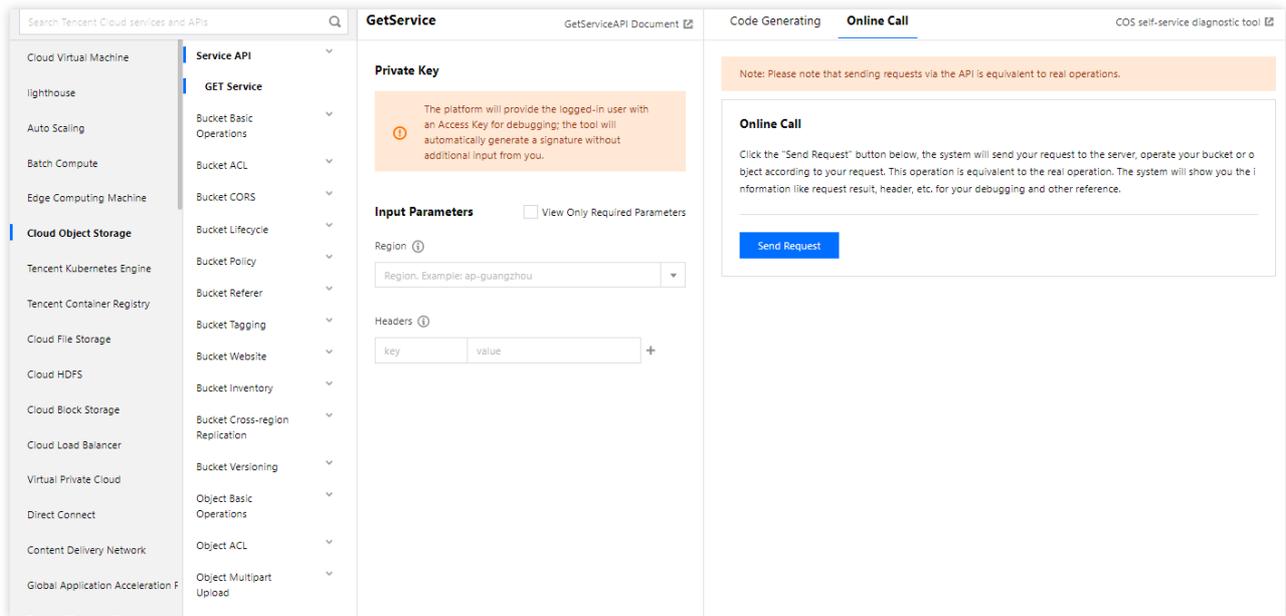
工具地址

单击进入 [COS 请求工具](#)。

使用方法

选择**对象存储**产品，选择所需的 API 接口，填写该接口下相应的参数，单击发送请求后获取相应的请求响应结果。

COS 请求工具的整体页面，从左至右依次是产品栏，接口栏，参数栏和结果栏。您可以在不同的栏目执行相应的操作，最终在结果栏发送请求并获取响应结果和相关的过程参数信息。如下图所示：



有关 COS 请求工具的详细操作请查看如下步骤。

1. 选择对象存储产品

在最左侧的产品栏中单击**对象存储**，将可以看到在 API 接口栏中与 COS 相关的 API 接口。

说明

COS 请求工具集成在云 API 3.0 平台，该平台上搭载着其他腾讯云产品的 API 调试工具，您同样可以根据您的需要在该平台上选择其他产品并调试相应的接口。

2. 选择需要调试的 API 接口

您可以根据您的需求选择相应的 API 接口进行调试。API 接口栏中展示了三大类与 COS 相关的 API 接口：Service 接口、Bucket 接口和 Object 接口。

在 Service 类接口中，如 GET Service。此 API 可列出您账号下的所有存储桶信息，您需要输入您的 API 密钥信息。如果您需要获取您账户在指定区域内的存储桶信息，可在参数栏中选择对应的地域。如需了解更多该 API 的详细信息，请参见 [GET Service](#) 文档。

在 Bucket 类接口中，包含对存储桶进行操作的 API 接口，如 PUT Bucket lifecycle。如需了解更多的 Bucket 类接口，请参见 [Bucket 接口](#)。

在 Object 类接口中，包含各个可对对象进行操作的 API 接口，如 PUT Object。如需了解更多的 Object 类接口，请参见 [Object 接口](#)。

3. 输入 API 所需的参数信息

参数栏将展示您所选的 API 接口的必填参数。有关 COS 的各个 API 接口的参数信息，请参见 [API 文档](#) 了解。

API 密钥信息是在调用 API 接口这一环节中必填的参数。当您使用 API 接口对您的存储桶或者对象等资源进行操作时，您需要输入 API 密钥信息，用于授权此次 API 请求操作。您可以在访问管理控制台的 [API 密钥管理](#) 页面上获取您的 API 密钥信息。

说明

对于每一个 API 接口，COS 请求工具在每一个必填的参数项后均加上了红色星号标识以提醒您该参数项为必填项。您也可以勾选**只看必填参数**，以减少参数栏上非必填的参数显示。

4. 发送请求及查看响应结果

当您选中 API，填写好相应的参数后，在**在线调用**选项卡中单击**发送请求**，此时系统会将您的请求发送到服务器，根据您的请求操作您的存储桶或者对象资源。

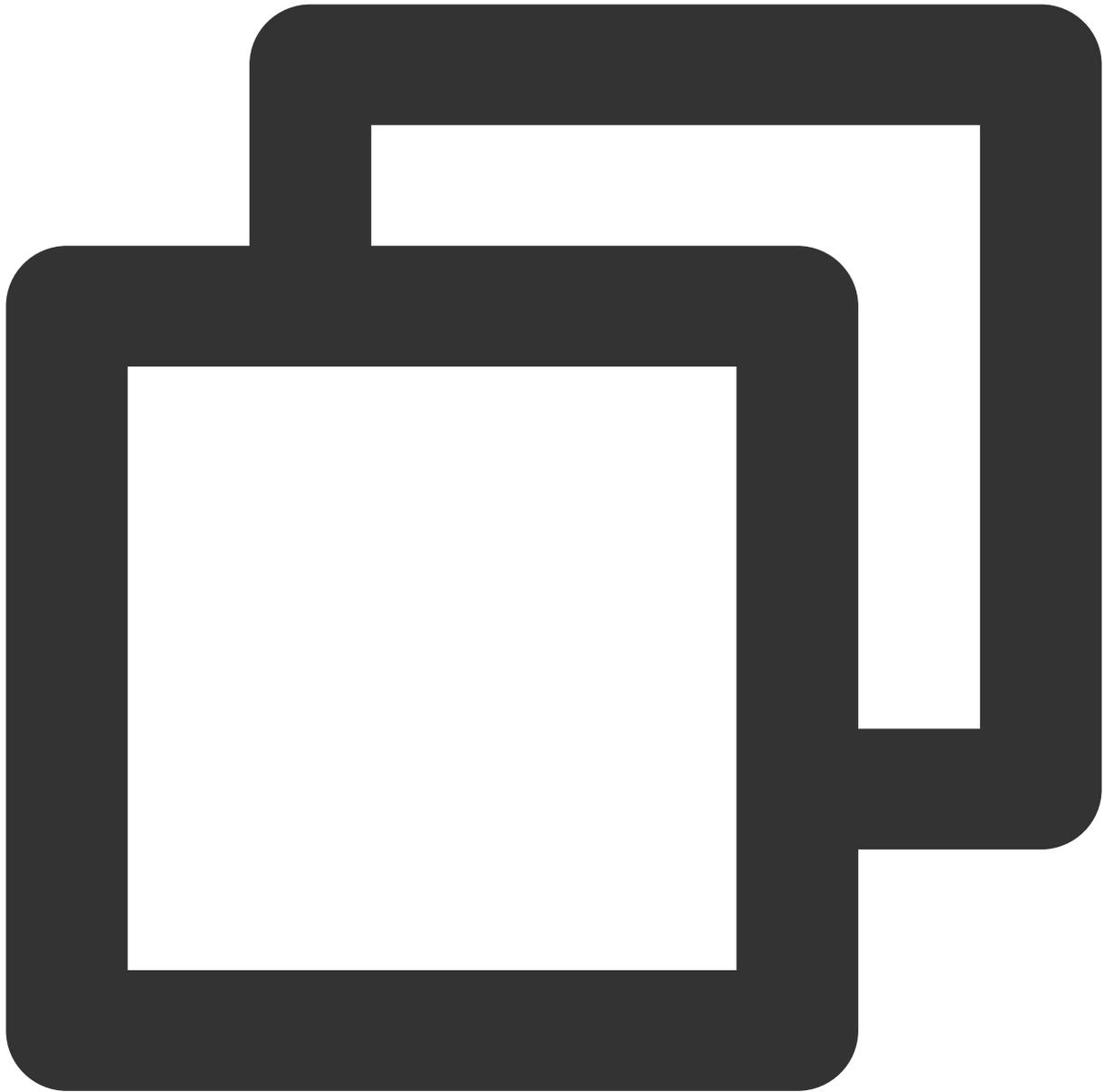
注意

COS 请求工具发送的请求将真实发送到 COS 的业务服务器，**所有操作均等同于真实操作**，当您选择 **DELETE** 类操作时请谨慎操作。

发送请求后，返回的结果和请求参数将会显示在结果栏下方。在结果栏下方，您可以在**请求参数**中查看您的 HTTP 请求体信息，在**响应结果**中查看本次请求的响应体信息，在**签名过程**中查看本次请求中涉及的签名及其生成流程，同时系统会在**Curl**中为您提供 Curl 调用的语句。

示例

以 GET Object 为例，当发送请求以获取一份名为0001.txt的文件时，**请求参数**中将展示您本次请求的请求参数信息，本次请求的请求示例如下：

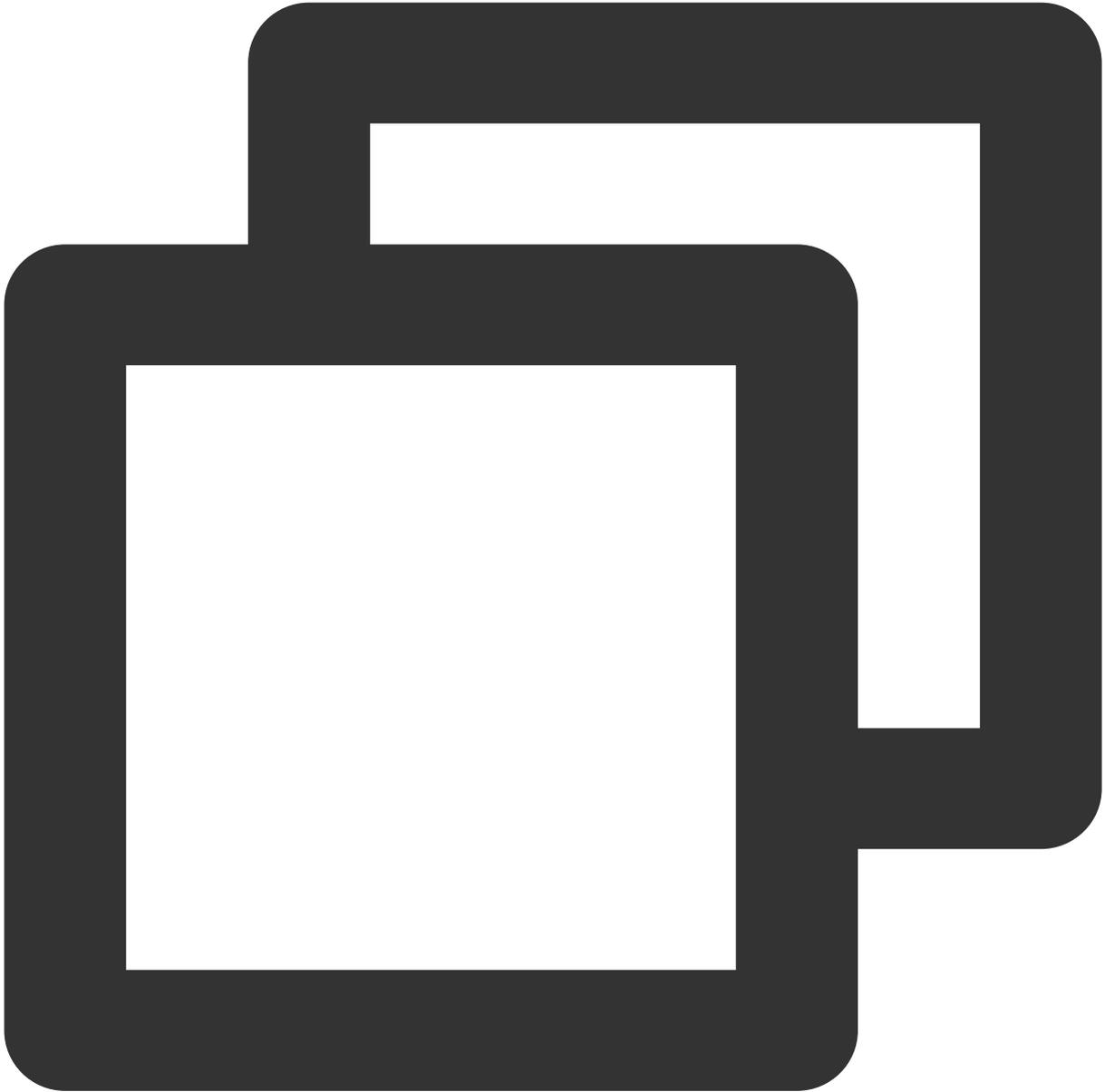


```
GET https://bucketname-appid.cos.ap-region.myqcloud.com/0001.txt
Host: bucketname-appid.cos.ap-region.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDwqaGoCIWIG4hDWdJUTL5e3hn04xi****&q-si
```

首行展示的是您的 HTTP Verb 及访问的链接，次行展示的是访问的域名，最后一行展示的是本次请求的签名信息。对于 PUT 类的请求，其请求头部信息较为复杂，但同样存在一些公共请求头部。有关公共请求头部的信息，请参见[公共请求头部](#)。

签名过程中可以查看本次请求中涉及的签名及其生成流程。有关签名算法的详细介绍，请参见[请求签名](#)详细了解，如果您需要对请求签名进行生成和调试，建议您使用COS 签名工具。

COS 返回的响应结果如下：



```
200 OK
content-type: text/plain
content-length: 6
connection: close
accept-ranges: bytes
date: Wed, 28 Nov 2018 09:42:49 GMT
etag: "5a8dd3ad0756a93ded72b823b19dd877"
last-modified: Tue, 27 Nov 2018 20:05:26 GMT
server: tencent-cos
```

```
x-cos-request-id: NWJmZTYzMTlflowUxYzBiMDlfOTA4NF8yMWI2****  
x-cos-version-id: MTg0NDY3NDI1MzAzODkyMjU****  
hello!
```

首行200 OK代表该请求返回的状态码信息，如请求失败，则会返回相应的错误码，有关错误码详细信息，请参见 [错误码](#) 文档。此后为响应头部信息，不同 API 的响应体均有所差异，但存在一些公共响应头部信息。有关公共响应头部的详细信息，请参见 [公共响应头部](#)。

注意事项

当您单击**发送请求**，确认将已填写必填参数的请求发送至 COS 服务端时，COS 将对您的存储桶和对象进行相应的操作，该操作无法撤回，无法回滚，请您在操作前慎重考虑。

自助诊断工具

最近更新时间：2024-01-06 16:18:07

功能概述

对象存储（Cloud Object Storage, COS）自助诊断工具是腾讯云对象存储为用户提供的 Web 工具，可用于错误请求的自助诊断排查。您可以在工具页面上填入请求的 RequestId（获取方法可参见 [获取 RequestId 操作指引](#)），单击诊断后，工具会对请求进行智能诊断，并展示该请求的基本信息以及帮助指引和诊断提示，帮助您快速定位使用 COS API 的错误问题。

工具地址

点此进入 [COS 自助诊断工具](#)。

使用方法

1. 单击 [COS 自助诊断工具](#)，进入“COS 自助诊断工具”页面。
2. 在顶部的 RequestId 输入框中，输入待诊断的 RequestId，并单击**开始诊断**。
3. 稍侯片刻，便能看到相应的智能诊断结果。

其中，诊断结果分为两部分：诊断结果和请求信息。

诊断结果为请求诊断后的建议与帮助，您可以根据建议与帮助快速定位到 COS API 的错误问题。

请求信息是该 RequestId 对应的基本信息。

4. 对诊断结果进行反馈

您可以单击诊断结果下方的**有帮助**或**无帮助**，将您对本次诊断结果的意见反馈给我们，以便我们继续优化工具。

常见问题

除了利用 RequestId 进行诊断排查外，该工具还提供丰富的常见问题样例以及对应的解决方案，方便您更好地使用 COS API 以及排查相关问题。此外，若您有疑问，您也可以 [联系我们](#)。

注意事项

标准的 COS 请求 RequestId 具有以下几个特征：

-
1. 以大写字母 N 开头。
 2. 长度不得小于30个字符。
 3. 遵循 base64字符标准。