



增量备份型文件迁移工具

用户使用指南

天翼云科技有限公司

目录

1. 简介	4
1.1 工具描述	4
1.2 功能特性	4
1.3 迁移动作描述	5
2. 安装与部署	7
2.1 部署需求	7
2.2 网络需求	7
2.3 安装	9
2.3.1 Agent	9
2.3.2 Server	10
2.4 部署	10
2.4.1 Agent	11
2.4.2 Server	13
3. 使用方法	14
3.1 启动服务	14
3.2 创建任务	15
3.3 开始任务	18
3.4 查询任务	18
3.5 停止任务	21
3.6 删除任务	21



4.注意事项.....	22
4.1 常见问题.....	22

1.简介

1.1 工具描述

增量备份型文件迁移工具是一款将数据迁移至 ZOS 的工具，主要提供将文件传输到云上 ZOS 的能力，工具本身作为服务部署时，支持长周期的迁移，并且在任务开启期间监控本地文件夹的变化，将本地文件夹的新增与修改（不包含删除）周期性的同步到云上。

也可以将云上 ZOS 中的对象文件下载到本地指定文件夹中（不支持周期型任务）。

1.2 功能特性

- 支持断点续传。迁移过程中，如果出现中断，重新启动工具后，可以继续执行迁移工作。已经迁移完成的部分数据，将会进行比对，如果一致则不进行迁移动作。

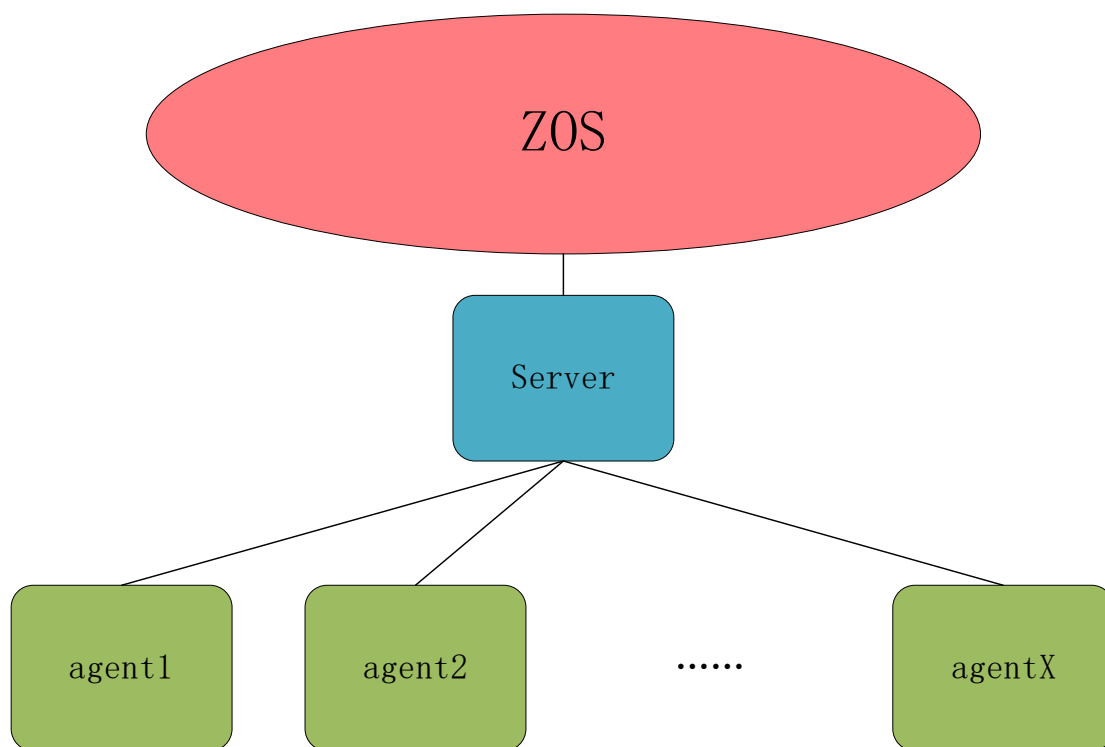
- 支持增量迁移，在迁移过程会根据配置的周期将本地文件的新增与修改同步到对象存储中（不包含删除）；

- 支持指定文件夹进行文件迁移；

- 文件上云后可以指定对应的前缀进行存储；

1.3 迁移动作介绍

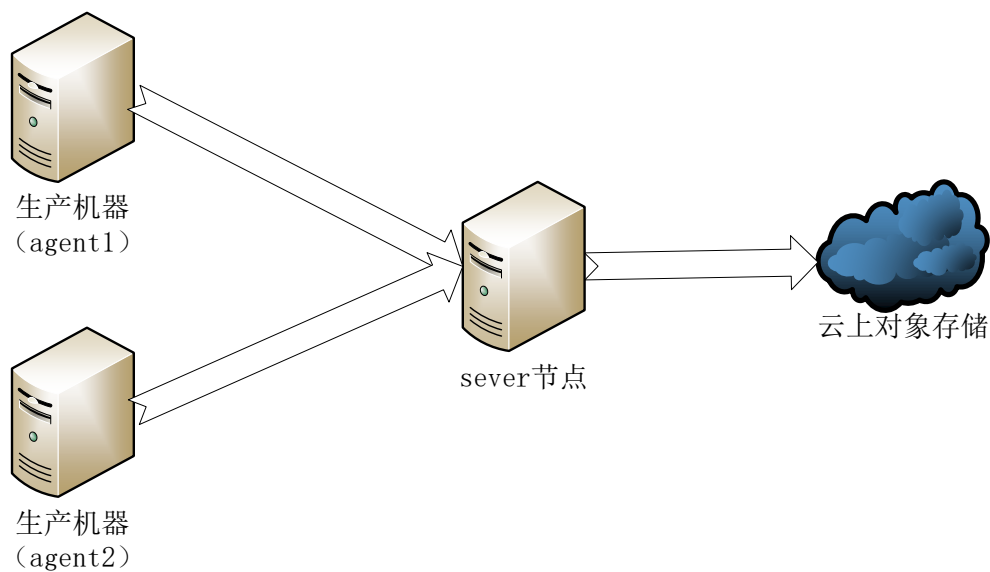
系统结构：



实践与数据路径：

本地机器 A 上有需要迁移的路径“/mnt/data”，已有存量数据 1T，且每天会生成新的数据文件，本地机器 B 上有日志“/var/log/”，每天会打包备份到“/var/log_bak”下以日期命名。

使用一台新的云主机作为 Server 节点，本地机器 A 与 B 作为 Agent 节点，部署完成后，创建任务分别指向“/mnt/data”和“/var/log_bak”，在存量数据迁移完成后，可以周期性的同步本地的修改与新增。



迁移任务类型：

增量备份型文件迁移工具的迁移任务主要分为两个类型：一次性任务与周期性任务。

其中一次性任务主要执行一个全量迁移动作，以开启任务的时间为节点，在此时间后产生变化（修改，新增等）的文件无法保证百分百的迁移成功；

周期性任务，分为全量迁移阶段与增量迁移阶段，开启周期性任务后，首先执行一个全量迁移动作，时间节点前的所有文件根据实际的配置同步到对象存储上；对于周期性任务，在开启整个任务时，agent 端就会开始记录本地的文件变化，并且上报到 Server 端进行记录，等待增量迁移阶段到达后执行迁移。

全量迁移完成后，进入增量迁移阶段，增量迁移模块会从数据库中获取当前存在的增量文件信息并且进行迁移。

2.安装与部署

2.1 部署需求

目前增量备份型迁移工具仅支持部署在 linux 系统上，具体运行环境：

- 系统要求是 Centos7/Ctyunos 2.0+
- 安装 python3.6~3.11

对于 agent 节点，额外需要以下需求：

- 安装 nfs-utils 组件，并且启动 nfsd 服务

对于 server 节点，需要以下需求：

- 安装 mysql5.6 以上版本

关于 mysql 的安装部署可以参考以下链接：

<https://www.ctyun.cn/developer/article/462390103912517>

如果节点所在的环境可以访问到相关的源也可以直接利用相关的管理工具进行安装（例如 yum，wget 等）

2.2 网络需求

由于通信需要，Agent 节点与 Server 节点都需要在入方向上保留一个自定义端口的通道。

入方向规则：

Agent 节点：



协议类型： TCP

端口范围： 8088（示例，实际值需要根据部署时设置的端口进行配置）

授权对象： 所有 IP 地址（或者根据你的实际需求，指定特定的 server 节点所在 IP 地址或 IP 地址段）

动作： 允许

协议类型： TCP

端口范围： 22（因为 server 需要通过 ssh 访问 agent），2049、30001:30003（nfsd 服务需要）

授权对象：（指定 server 节点所在 IP 地址或 IP 地址段）

动作： 允许

Server 节点：

协议类型： TCP

端口范围： 8000（示例，实际值需要根据部署时设置的端口进行配置）

授权对象： 所有 IP 地址（或者根据你的实际需求，指定特定的 agent 所在 IP 地址或 IP 地址段）

动作： 允许

出方向规则：

Agent 节点：

协议类型： TCP

端口范围： 不限



授权对象：Server 节点的 IP 地址或者是 IP 地址段

动作：允许

Server 节点：

协议类型：TCP

端口范围：不限

授权对象：对象存储 ZOS 的相关域名或者是指定的 IP 地址（通过云主机 vpc 内网的情况下），还需要对 agent 节点所在的 IP 开放。

动作：允许

2.3 安装

增量备份型文件迁移工具主要提供了以下几个 rpm 包：

- ctyun-mig-agent-0.9.0-1.0.ctl2.x86_64.rpm
- ctyun-mig-server-0.9.0-1.0.ctl2.x86_64.rpm
- rclone-v1.62.2-linux-amd64.rpm

2.3.1 Agent

工具的软件安装直接使用 rpm 进行即可，其中 agent 节点需要安装 ctyun-mig-agent 软件，如下图所示：

```
[root@localhost rpm]# ll
total 82M
-rw-r--r-- 1 root root 23M Mar 15 13:45 ctyun-mig-agent-0.9.0-1.0.ct12.x86_64.rpm
-rw-r--r-- 1 root root 42M Mar 15 13:45 ctyun-mig-server-0.9.0-1.0.ct12.x86_64.rpm
-rw-r--r-- 1 root root 19M Mar 13 16:38 rclone-v1.62.2-linux-amd64.rpm
[root@localhost rpm]#
[root@localhost rpm]#
[root@localhost rpm]# rpm -ivh ctyun-mig-agent-0.9.0-1.0.ct12.x86_64.rpm
Verifying... ##### [100%]
Preparing... ##### [100%]
Updating / installing...
 1:ctyun-mig-agent-0.9.0-1.0.ct12 ##### [100%]
[root@localhost rpm]#
```

Agent 节点在安装完整后主要查看“/usr/bin/ctyun-migration”目录下是否存在 agent 相关的可执行程序。

```
complete!
[root@localhost rpm]# ls /usr/bin/ctyun-migration/
agent  agent_tool
[root@localhost rpm]#
```

2.3.2 Server

Server 节点需要安装 ctyun-mig-server 与 rclone 两个软件

```
[root@localhost rpm]# ll
total 82M
-rw-r--r-- 1 root root 23M Mar 14 21:57 ctyun-mig-agent-0.9.0-1.0.ct12.x86_64.rpm
-rw-r--r-- 1 root root 42M Mar 14 21:57 ctyun-mig-server-0.9.0-1.0.ct12.x86_64.rpm
-rw-r--r-- 1 root root 19M Mar 15 15:47 rclone-v1.62.2-linux-amd64.rpm
[root@localhost rpm]#
[root@localhost rpm]# rpm -ivh ctyun-mig-server-0.9.0-1.0.ct12.x86_64.rpm
error: Failed dependencies:
  rclone is needed by ctyun-mig-server-0.9.0-1.0.ct12.x86_64
[root@localhost rpm]#
[root@localhost rpm]#
[root@localhost rpm]# rpm -ivh rclone-v1.62.2-linux-amd64.rpm ctyun-mig-server-0.9.0-1.0.ct12.x86_64.rpm
Verifying... ##### [100%]
Preparing... ##### [100%]
Updating / installing...
 1:rclone-0:1.62.2-1 ##### [ 50%]
 2:ctyun-mig-server-0.9.0-1.0.ct12 ##### [100%]
[root@localhost rpm]#
[root@localhost rpm]#
```

Server 节点在安装完整后主要查看“/usr/bin/ctyun-migration”目录下是否存在 webserver 相关的可执行程序。

2.4 部署

部署主要是介绍如何设置软件的必要配置以及各个节点上的一些环境需求

2.4.1 Agent

在 Agent 节点上,因为需要提供挂载点给到 Server 节点,因此需要安装 nfs—utils 组件提供 nfsd 服务。

```
[root@localhost ~]# yum info nfs-utils
Last metadata expiration check: 2:03:53 ago on Thu 14 Mar 2024 08:59:03 AM CST.
Available Packages
Name      : nfs-utils
Epoch    : 1
Version   : 2.5.1
Release   : 3.ct12
Architecture: x86_64
Size      : 319 k
Source    : nfs-utils-2.5.1-3.ct12.src.rpm
Repository: everything
Summary   : The Linux NFS userland utility package
URL       : http://sourceforge.net/projects/nfs/
License   : MIT and GPLv2 and GPLv2+ and BSD
Description: This is the nfs-utils tools package.
           : It contains the showmount, mount.nfs, umount.nfs and libnfsidmap
```

Agent 节点上主要需要配置 agent 的 ip 与 port, 用于监听来自 Server 节点的请求。执行 “/usr/bin/ctyun-migration/agent_tool config” 来进行交互, 根据提示按顺序填写以下信息:

- server ip: agent 注册用于监听的 ip 地址
- server port: agent 注册用于监听的端口号

```
[root@localhost rpm]# /usr/bin/ctyun-migration/agent_tool
Please confirm that you have entered the correct command.
Usage:
    config
        set agent ip and port
    Createtask
        create [uuid] [name] [point_type] [task_type] [path] [dest_url]
    Starttask
        start [uuid]
    Stoptask
        stop [uuid]
    Deletetask
        delete [uuid]
    Gettask
        get [uuid]
    Listtask
        list
    Help
        help

[root@localhost ~]# /usr/bin/ctyun-migration/agent_tool config
The content in '[]' is the current values. If you do not need to modify, press 'Enter'!
Set server ip(like http://x.x.x.x) []:192.168.122.277
Set server service port(like 8000) []:8888
[root@localhost ~]#
```

Agent 节点还需要修改部分环境变量, 首先使用 “sysctl -a|grep inotify” 命

令查询对应的值：

```
[root@localhost /]# sysctl -a|grep inotify
fs.inotify.max_queued_events = 16384
fs.inotify.max_user_instances = 128
fs.inotify.max_user_watches = 8192
user.max_inotify_instances = 128
user.max_inotify_watches = 8192
[root@localhost /]#
```

然后使用如下命令进行修改：

```
sysctl -w fs.inotify.max_user_watches=524288
```

```
sysctl -w fs.inotify.max_queued_events=10000000
```

```
[root@localhost /]# sysctl -w fs.inotify.max_user_watches=524288
fs.inotify.max_user_watches = 524288
[root@localhost /]# sysctl -w fs.inotify.max_queued_events=10000000
fs.inotify.max_queued_events = 10000000
[root@localhost /]#
[root@localhost /]#
[root@localhost /]#
[root@localhost /]# sysctl -a|grep inotify
fs.inotify.max_queued_events = 10000000
fs.inotify.max_user_instances = 128
fs.inotify.max_user_watches = 524288
user.max_inotify_instances = 128
user.max_inotify_watches = 524288
[root@localhost /]#
[root@localhost /]#
```

但是这种方法是临时的，一旦重启机器就需要重新设置，因此建议将该修改

固化到 sysctl.conf 中：

```
echo fs.inotify.max_user_watches=524288 >> /etc/sysctl.conf
```

```
echo fs.inotify.max_queued_events=10000000 >> /etc/sysctl.conf
```

```
sysctl -p
```

```
[root@localhost sysctl.d]#  
[root@localhost sysctl.d]# echo fs.inotify.max_user_watches=524288 >> /etc/sysctl.conf  
[root@localhost sysctl.d]# echo fs.inotify.max_queued_events=10000000 >> /etc/sysctl.conf  
[root@localhost sysctl.d]# sysctl -p  
kernel.sysrq = 0  
net.ipv4.ip_forward = 0  
net.ipv4.conf.all.send_redirects = 0  
net.ipv4.conf.default.send_redirects = 0  
net.ipv4.conf.all.accept_source_route = 0  
net.ipv4.conf.default.accept_source_route = 0  
net.ipv4.conf.all.accept_redirects = 0  
net.ipv4.conf.default.accept_redirects = 0  
net.ipv4.conf.all.secure_redirects = 0  
net.ipv4.conf.default.secure_redirects = 0  
net.ipv4.icmp_echo_ignore_broadcasts = 1  
net.ipv4.icmp_ignore_bogus_error_responses = 1  
net.ipv4.conf.all.rp_filter = 1  
net.ipv4.conf.default.rp_filter = 1  
net.ipv4.tcp_syncookies = 1  
kernel.dmesg_restrict = 1  
net.ipv6.conf.all.accept_redirects = 0  
net.ipv6.conf.default.accept_redirects = 0  
fs.inotify.max_user_watches = 524288  
fs.inotify.max_queued_events = 10000000  
[root@localhost sysctl.d]#
```

2.4.2 Server

Server 节点需要部署 mysql 数据库,本地的或者是容器化等其他方式也可以,需要将 mysql 服务的 host 和 port 以及认证信息提供给 Server 服务的软件 webserver。

Tips: 因为 webserver 服务会使用 “/mnt/hybrid/” 这个路径进行挂载,因此需要确认 Server 节点的 mnt 目录下不会存在互相干扰的软件需求。

在部署时,可以直接使用软件包中提的 webserver_tools 进行配置,这个命令行工具提供了 usage 和任务相关的各项操作。

```
[root@localhost rpm]# /usr/bin/ctyun-migration/webserver_tools  
error: list index out of range  
should use like this:  
python3 webserver_tools.py serverconfig  
python3 webserver_tools.py create  
python3 webserver_tools.py createinfo  
python3 webserver_tools.py start taskid  
python3 webserver_tools.py stop taskid  
python3 webserver_tools.py remove taskid  
python3 webserver_tools.py get taskid  
python3 webserver_tools.py getall  
python3 webserver_tools.py getstatus taskid  
python3 webserver_tools.py getallstatus  
python3 webserver_tools.py getfullerrs taskid  
python3 webserver_tools.py getincerrs taskid  
[root@localhost rpm]#
```

在命令行中执行 “/usr/bin/ctyun-migration/webserver_tools serverconfig” 即

可进入交互，根据提示按顺序填写以下信息：

- server ip: webservice 用来注册服务监听的地址
- server port: webservice 用来注册服务监听的端口
- Database host: 数据库服务提供的地址
- Database port: 数据库服务提供的端口
- Database user: 数据库服务的认证用户
- Database passwd: 数据库服务的认证密码

```
[root@localhost rpm]#  
[root@localhost rpm]# /usr/bin/ctyun-migration/webserver_tools serverconfig  
The content in '[' is the current values. If you do not need to modify, press 'Enter'!  
Set server ip(like http://x.x.x.x) [0.0.0.0]:http://192.168.122.96  
Set server service port(like 8000) [8000]:8000  
Set Database host port(like localhost) [127.0.0.1]:127.0.0.1  
Set Database port(like 3306) [3306]:3306  
Set Database user:root  
Set Database passwd:ctyun114!  
[root@localhost rpm]#
```

3.使用方法

3.1 启动服务

首先启动 server 节点的服务，在完成安装与部署之后，使用命令“nohup /usr/bin/ctyun-migration/webserver runserver --noreload &”启动服务，执行后，可以通过命令“/usr/bin/ctyun-migration/webserver_tools getall”查看当前服务的情况，如果可以正常反馈信息，则认为服务已经成功启动：

```
[root@localhost ~]#  
[root@localhost ~]# nohup /usr/bin/ctyun-migration/webserver runserver --noreload &  
[1] 330863  
[root@localhost ~]# nohup: ignoring input and appending output to 'nohup.out'  
  
[root@localhost ~]#  
[root@localhost ~]#
```



```
[root@localhost ctyun-migration]#  
[root@localhost ctyun-migration]#  
[root@localhost ctyun-migration]# /usr/bin/ctyun-migration/webserver_tools getall  
{'errcode': 302, 'errmsg': 'There is no task.', 'detail': ''}  
[root@localhost ctyun-migration]#
```

然后是启动 agent 节点的服务，使用命令 “nohup /usr/bin/ctyun-migration/agent &” 启动服务，使用命令 “/usr/bin/ctyun-migration/agent_tool list” 查询当前 agent 节点上的状态，如果正常反馈信息，任务 agent 节点服务已经成功启动。

```
[root@Agent ctyun-migration]#  
[root@Agent ctyun-migration]# nohup /usr/bin/ctyun-migration/agent &  
[1] 697207  
[root@Agent ctyun-migration]# nohup: ignoring input and appending output to 'nohup.out'  
  
[root@Agent ctyun-migration]#  
[root@Agent ctyun-migration]#  
[root@Agent ctyun-migration]#
```

```
[root@Agent ctyun-migration]#  
[root@Agent ctyun-migration]# /usr/bin/ctyun-migration/agent_tool list  
There is no task.  
  
[root@Agent ctyun-migration]#
```

3.2 创建任务

创建任务使用命令“/usr/bin/ctyun-migration/webserver_tools create”进行，create 命令是交互式操作，根据实际的提示，下面的参数表和图例参考填写具体的任务配置信息

```
[root@localhost ctyun-migration]#  
[root@localhost ctyun-migration]#  
[root@localhost ctyun-migration]# /usr/bin/ctyun-migration/webserver_tools create  
The content in '[' is the current values. If you do not need to modify, press 'Enter'!  
Set taskname(length: 4-32) [task1]:task1  
Set cycle(0:once, 1:period) [1]:1  
Set period(greater than 1800 seconds) [1800]:1800  
Set agent ip(like http://x.x.x.x) [http://192.168.122.227]:http://192.168.122.227  
Set agent service port(like 8888) [8888]:8888  
Set dir_path(absolute path, like '/data') [/mnt/test]:/mnt/test  
Set point_type(sshd or nfsd) [nfsd]:nfsd  
Set client user name [root]:root  
Set client password [Hys230905!]:Hys230905!  
Set type(0:upload, 1:download) [0]:0  
Set access_key [AQL01GGEIIFUUOSI4207]:AQL01GGEIIFUUOSI4207  
Set secret_key [IDXHLcj8792voam3JUii2PvQofPbZpBt01YkaJMC]:IDXHLcj8792voam3JUii2PvQofPbZpBt01YkaJMC  
Set endpoint [http://10.24.255.1]:http://10.24.255.1  
Set bucket [hybridSysTest]:  
Set prefix []:tooltest/  
{'errcode': 200, 'errmsg': 'Success.', 'detail': 'create task success.', 'task_id': 'f6c1bcf61607404996fc361430b413da'}  
[root@localhost ctyun-migration]#
```

任务创建成功会反馈具体的 task_id，这个 id 是 task 的唯一标识，对任务进行操



作都需要依赖这个 task_id，如果遗忘的话，可以通过“getall”命令查询进行核对。

```
[root@localhost ~]# /usr/bin/ctyun-migration/webserver_tools get f6c1bcf61607404996fc361430b413da
{"errcode": 200, "message": "Success.", "detail": "get task success.", "data": "show below....."}
task_id: f6c1bcf61607404996fc361430b413da
task_name: task1
cycle(0:once; 1:period): 1
mig_period(s): 1800
agent_ip: http://192.168.122.227
port: 8888
dir_path: /mnt/test
point_type: nfsd
task_type(0:upload; 1:download): 0
src_access_key(download): None
src_secret_key(download): None
src_endpoint(download): None
src_bucket(download): None
src_prefix(download): None
dest_access_key(upload): AQL01GGEI1FUUOSI4207
dest_secret_key(upload): IDXLcj8792vOam3JUii2PvQofPbZp8to1YkaJMC
dest_endpoint(upload): http://10.24.255.1
dest_bucket(upload): hybridSysTest
dest_prefix(upload): tooltest/
create_time: 2024-03-14 16:14:37
[root@localhost ~]#
```

参数列表：

参数项	释义	参数范围	示例
taskname	迁移任务的名称	4-32 位长度的字母和数字组成	task1 task2 testtask
cycle	迁移任务是否需要周期执行	0 表示非周期 1 表示周期	1
period	迁移任务的周期，单位是秒	大于等于 1800	1800 3600
agent ip	任务对应的 agent 节点的 ip	http://{ip} 这里需要填写附带 http 的 ip 信息	http://192.168.122.155 http://192.168.122.227
agent port	任务对应的 agent 服务的监听端口	1~65535	8088 8888
dir_path	任务在 agent 节点上的文件夹路径	linux 标准路径范围	/mnt/test /data/test
point_type	从 server 节点挂	nfsd 或者是 sshfs	nfsd

	载 agent 节点使用的协议		
user name	agent 节点可以访问到文件夹路径的用户	linux 的标准用户名	root secure
password	user name 账户对应的密码	linux 的用户密码标准	
migration type	迁移任务的类型（上云或者下载）	0 表示 upload 1 表示 download	0
access_key	对象存储的 AccessKey 字段，用于认证	无	无
secret_key	对象存储的 SecretKey 字段，用于认证	无	无
endpoint	对象存储的访问入口地址，通常可以通过桶的概览页面查询到公网入口与内网入口	对象存储的入口域名或者是内网的地址	https://huabei-2.zos.ctyun.cn http://100.86.2.1
bucket	对象存储的桶名	参考对象存储桶命名规范	bucket-ea2e bucket-test
prefix	迁移到对象存储的文件如果不希望直接放在桶的根目录，可以根据实际需求设置	linux 文件夹的路径拼写规则	详情请参考注意事项

3.3 开始任务

创建完任务后，可以通过 webserver_tools 的 start 命令开启任务，

```
[root@localhost ctyun-migration]#  
[root@localhost ctyun-migration]# /usr/bin/ctyun-migration/webserver_tools start f6c1bcf61607404996fc361430b413da  
{'errcode': 200, 'errmsg': 'Success.', 'detail': 'Task is starting'}  
[root@localhost ctyun-migration]#
```

开启任务后可以通过查询任务的指令查看任务的当前状态：

```
[root@localhost ctyun-migration]# /usr/bin/ctyun-migration/webserver_tools getstatus f6c1bcf61607404996fc361430b413da  
{'errcode': 200, 'message': 'Success.', 'detail': 'get task status success.', 'data': 'show below.....'}  
task_id:f6c1bcf61607404996fc361430b413da  
cycle(0:once; 1:period):1  
task_status(1:create; 2:executing; 3:stop; 4:success; 5:fail; 6:starting):2  
full_checked_files:100  
full_exist_files:0  
full_transferred_files:100  
full_transferred_size:9.766MiB/9.766MiB  
full_error_files:0  
inc_transferred_files:0  
inc_transferred_size:0  
inc_error_files:0  
inc_speed:0.08/s  
start_time:2024-03-14 16:43:56  
elapsed_time:None  
complete_time:2024-03-14 16:43:58  
full_mig(0:not finish; 1:finished):1  
[root@localhost ctyun-migration]#
```

3.4 查询任务

查询任务可以通过 webserver_tools 的命令：getall， get {task uuid}，
getallstatus， getstatus {task uuid}来完成。

getall:



```
[root@localhost ctyun-migration]# /usr/bin/ctyun-migration/webserver_tools getall
{'errcode': 200, 'errmsg': 'Success.', 'detail': 'get all task success.', 'data': 'show below.....'}
TASK INFO #1-----
task_id:f6c1bcf61607404996fc361430b413da
task_name:task1
cycle(0:once; 1:period):1
mig_period(s):1800
agent_ip:http://192.168.122.227
port:8888
dir_path:/mnt/test
point_type:nfsd
task_type(0:upload; 1:download):0
src_access_key(download):None
src_secret_key(download):None
src_endpoint(download):None
src_bucket(download):None
src_prefix(download):None
dest_access_key(upload):AQL01GGEIIFUUOSI4207
dest_secret_key(upload):IDXLHCj8792VOam3JUii2PvQofPbZpBt01YkaJMC
dest_endpoint(upload):http://10.24.255.1
dest_bucket(upload):hybridSysTest
dest_prefix(upload):tooltest/
create_time:2024-03-14 16:14:37
```

getallstatus:

```
[root@localhost ctyun-migration]# /usr/bin/ctyun-migration/webserver_tools getallstatus
{'errcode': 200, 'errmsg': 'Success.', 'detail': 'get all task status success.', 'data': 'show below.....'}
TASK STATUS #1-----
task_id:f6c1bcf61607404996fc361430b413da
cycle(0:once; 1:period):1
task_status(1:create; 2:executing; 3:stop; 4:succes; 5:fail; 6:starting):2
full_checked_files:100
full_exist_files:0
full_transferred_files:100
full_transferred_size:9.766MiB/9.766MiB
full_error_files:0
inc_transferred_files:11
inc_transferred_size:1000
inc_error_files:0
inc_speed:0.0B/s
start_time:2024-03-14 16:43:56
elapsed_time:None
complete_time:2024-03-14 16:43:58
full_mig(0:not finish; 1:finished):1
[root@localhost ctyun-migration]#
```

getstatus:

```
[root@localhost ctyun-migration]# /usr/bin/ctyun-migration/webserver_tools getstatus f6c1bcf61607404996fc361430b413da
{'errcode': 200, 'message': 'Success.', 'detail': 'get task status success.', 'data': 'show below.....'}
task_id:f6c1bcf61607404996fc361430b413da
cycle(0:once; 1:period):1
task_status(1:create; 2:executing; 3:stop; 4:succes; 5:fail; 6:starting):2
full_checked_files:100
full_exist_files:0
full_transferred_files:100
full_transferred_size:9.766MiB/9.766MiB
full_error_files:0
inc_transferred_files:11
inc_transferred_size:1000
inc_error_files:0
inc_speed:0.0B/s
start_time:2024-03-14 16:43:56
elapsed_time:None
complete_time:2024-03-14 16:43:58
full_mig(0:not finish; 1:finished):1
[root@localhost ctyun-migration]#
```

参数	释义	示例
task_status(1:create; 2:executing; 3:stop; 4:succes; 5:fail; 6:starting):3	任务当前处于的状态(1:已创建; 2:执行中; 3:已停止; 4:执行成功 (针对非周期任	2

	务);5:失败(出现异常);6:启动中(由于 agent 通信可能产生的中间态))	
full_checked_files:100	全量迁移阶段检查了多少文件	
full_exist_files:0	全量迁移过程对比对象存储中已经存在不需要迁移的文件数量	
full_transferred_files:100	全量迁移阶段总计迁移了多少文件	
full_transferred_size:9.766MiB/9.766MiB	全量迁移的速率	
full_error_files:0	全量迁移阶段产生的错误文件数量	
inc_transferred_files:11	当前增量迁移阶段中已经完成迁移的文件数量	1100
inc_transferred_size:1000	当前增量迁移阶段中已经完成迁移的文件总大小, 单位是 KB	10000
inc_error_files:0	增量迁移中产生的错误文件数量	0 100
inc_speed:0.0B/s	当前增量迁移的实时迁移速率	27.6MB/s

start_time:	整个迁移任务的 开启时间	2024-03-14 16:43:56
elapsed_time:	最近一次周期迁 移任务的执行时 间	2024-03-14 20:20:10
complete_time:	全量迁移任务的 完成时间	2024-03-14 16:43:58
full_mig(0:not finish; 1:finished)	是否已经完成了 全量迁移阶段	1

3.5 停止任务

停止任务通过命令“stop {task uuid}”完成：

```
[root@localhost ctyun-migration]#  
[root@localhost ctyun-migration]# /usr/bin/ctyun-migration/webserver_tools stop f6c1bcf61607404996fc361430b413da  
{'errcode': 200, 'errmsg': 'Success.', 'detail': 'Stop task success.'}  
[root@localhost ctyun-migration]#  
[root@localhost ctyun-migration]#
```

由于任务在迁移过程中以部分动作是异步完成的，因此在停止时需要稍等3~10s左右的时间来完成停止。

3.6 删除任务

删除任务通过命令“remove {task uuid}”完成，需要注意的是，处于“执行个”状态的任务是不能被删除的，需要先停止才能完成删除动作：

```
[root@localhost ctyun-migration]#  
[root@localhost ctyun-migration]# /usr/bin/ctyun-migration/webserver_tools remove f6c1bcf61607404996fc361430b413da  
{'errcode': 303, 'errmsg': 'Task is executing.', 'detail': 'Task is executing, please stop the task first.'}  
[root@localhost ctyun-migration]#
```

成功的示例：

```
[root@localhost ctyun-migration]#  
[root@localhost ctyun-migration]# /usr/bin/ctyun-migration/webserver_tools remove f6c1bcf61607404996fc361430b413da  
{'errcode': 200, 'errmsg': 'Success.', 'detail': 'Remove task success.'}  
[root@localhost ctyun-migration]#
```

4.注意事项

4.1 常见问题

安全组与防火墙

如果安全组或者是防火墙的设置存在问题，会可能导致 Server 节点与 Agent 节点的交互异常，导致创建任务，开启任务等指令无法正确响应。

配置错误

如果填写了错误的 ip 地址或者是端口号，可能会导致 webserver 在启动时报错 “Error: [Errno -2] Name or service not known”

```
[root@localhost ~]# /usr/bin/ctyun-migration/webserver runserver --noreload
Performing system checks...

System check identified no issues (0 silenced).
March 14, 2024 - 14:32:10
Django version 3.2.24, using settings 'server.settings'
Starting development server at http://http://192.168.122.96:8000/
Quit the server with CONTROL-C.
Error: [Errno -2] Name or service not known
[root@localhost ~]#
```

```
[service]
service_ip = http://192.168.122.96
service_port = 8000
checkers = 8
transfers = 8
full_threads = 16
inc_threads = 16
inc_nums = 100
log_size = 50
log_maxcnt = 5
min_period = 1800
```

Server 启动失败

数据库服务没有启动的情况下先去启动 Server，会因为获取不到数据库的操作连接导致服务启动失败。



4.2 支持