

GO Version S3 Linux Environment Deployment Manual

1. Description

The instructions shown below are applicable for Centos7+, if the deployment is being done through other Linux systems, please use the corresponding commands

2. s3 download and install

2.1 Downloading and Unzipping

First copy the s3 installation folder to the designated test machine (the installation folder can be downloaded from the official website, <https://www.yottachain.io/> -> go to the top menu, in Download(or Get Started) -> download Go S3 Client-linux), then unzip it to the specified directory, as shown below:

```
[root@nm-yaceji03 ~]# ls
anaconda-ks.cfg  docker-20200714.tar.gz  IdcNodeInit.tar.gz  iperf-3.9.tar.gz  yts
config          ftp_download.sh        init                swap
docker         GO_YTS3_v2.0.0.17.tar.gz  iperf-3.9         temp
[root@nm-yaceji03 ~]# tar zxf GO_YTS3_v2.0.0.17.tar.gz
[root@nm-yaceji03 ~]# ls
anaconda-ks.cfg  docker-20200714.tar.gz  GO_YTS3_v2.0.0.17.tar.gz  iperf-3.9
config          ftp_download.sh        IdcNodeInit.tar.gz        iperf-3.9.tar.gz
docker         GO_YTS3_Linux         init                        swap
[root@nm-yaceji03 ~]#
```

After successful unzipping, copy the "GO_YTS3_Linux" directory to the corresponding location, here we copied it to the "/mnt" directory, then install and start the service. The directory structure is shown in the following figure:

```
[root@VM_32_2_centos GO_YTS3_Linux]# ls
bin  conf  crt  YTS3
[root@VM_32_2_centos GO_YTS3_Linux]#
```

2.2 Installation and operation

Enter the bin directory of s3 and execute "./YTS3 install" to install the service

```
[root@nm-yaceji03 GO_YTS3_Linux]# ./YTS3 install
I: 10:52:14 Path:./YTS3
I: 10:52:14 Install OK.
[root@nm-yaceji03 GO_YTS3_Linux]#
```

Check the installation statuses, execute "systemctl status yts3.service"

```
[root@nm-yaceji03 G0_YTS3_Linux]# systemctl status yts3.service
● yts3.service - go yts3 daemons service
   Loaded: loaded (/etc/systemd/system/yts3.service; enabled; vendor preset: disabled)
   Active: inactive (dead)
[root@nm-yaceji03 G0_YTS3_Linux]#
```

Check the installation statuses, execute "systemctl start yts3.service"

```
[root@nm-yaceji03 G0_YTS3_Linux]# systemctl status yts3.service
● yts3.service - go yts3 daemons service
   Loaded: loaded (/etc/systemd/system/yts3.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2020-12-14 10:53:54 CST; 4s ago
   Main PID: 29750 (YTS3)
   CGroup: /system.slice/yts3.service
           └─29750 /mnt/G0_YTS3_Linux/YTS3

Dec 14 10:53:54 nm-yaceji03 YTS3[29750]: [GIN-debug] GET    /api/v1/createBucket    -> github.com/yottachain/YTS3/controll...ndlers
Dec 14 10:53:54 nm-yaceji03 YTS3[29750]: [GIN-debug] POST   /api/v1/upload          -> github.com/yottachain/YTS3/controll...ndlers
Dec 14 10:53:54 nm-yaceji03 YTS3[29750]: [GIN-debug] GET    /api/v1/getObject       -> github.com/yottachain/YTS3/controll...ndlers
Dec 14 10:53:54 nm-yaceji03 YTS3[29750]: [GIN-debug] GET    /api/v1/getObjectProgress -> github.com/yottachain/YTS3/controll...ndlers
Dec 14 10:53:54 nm-yaceji03 YTS3[29750]: [GIN-debug] GET    /api/v1/listBucket      -> github.com/yottachain/YTS3/controll...ndlers
Dec 14 10:53:54 nm-yaceji03 YTS3[29750]: [GIN-debug] GET    /api/v1/listAllBucket   -> github.com/yottachain/YTS3/controll...ndlers
Dec 14 10:53:54 nm-yaceji03 YTS3[29750]: [GIN-debug] GET    /api/v1/getProgress      -> github.com/yottachain/YTS3/controll...ndlers
Dec 14 10:53:54 nm-yaceji03 YTS3[29750]: [GIN-debug] GET    /api/v1/getYts3Version  -> github.com/yottachain/YTS3/controll...ndlers
Dec 14 10:53:54 nm-yaceji03 YTS3[29750]: [10:53:54.587][Infos]HTTPServer start Success 8080
Dec 14 10:53:54 nm-yaceji03 YTS3[29750]: [GIN-debug] Listening and serving HTTP on listener what's bind with address@0.0.0.0:8080
Hint: Some lines were ellipsized, use -l to show in full.
[root@nm-yaceji03 G0_YTS3_Linux]#
```

Check the log to see that the service is already running

```
[root@nm-yaceji03 G0_YTS3_Linux]# pwd
/mnt/G0_YTS3_Linux
[root@nm-yaceji03 G0_YTS3_Linux]# ls
bin  cache  conf  log  p2phostinfo.log  YTS3
[root@nm-yaceji03 G0_YTS3_Linux]# tailf log/log
[10:53:54.265][Infos][BooTer]Node Addr 0:/ip4/192.168.1.195/tcp/38866
[10:53:54.265][Infos][BooTer]Node Addr 1:/ip4/117.161.72.86/tcp/38866
[10:53:54.265][Infos][BooTer]Node Addr 2:/ip6/fe80::ff0a:74e9:dec9:b046/tcp/38866
[10:53:54.265][Infos][BooTer]Node Addr 3:/ip6/fe80::ebe7:1391:db33:c8c9/tcp/38866
[10:53:54.265][Infos][BooTer]Node Addr 4:/ip6/fe80::838e:e545:ffba:baef/tcp/38866
[10:53:54.385][Infos][NodeMgr]Snlist init ok,Size:21
[10:53:54.586][Infos][Cache]LocalDB init...Path:/mnt/G0_YTS3_Linux/cache/dbcache/cache.db
[10:53:54.587][Infos][Cache]Sum cache size 0
[10:53:54.587][Infos][Cache]Clearing expired files...
[10:53:54.587][Infos]HTTPServer start Success 8080
```

3. Pre-Test Preparation

Before the final test, you should configure the relevant test machine according to the actual statuses of your test machine, and install the corresponding test tool

3.1 Configuration modification

Check the "yts.properties" file in the "conf" directory as follows, then modify the configuration according to the situation

```
[root@nm-yaceji03 G0_YTS3_Linux]# cat conf/yts.properties
#日志输出级别, debug, info... 不填填错不写日志
logLevel=debug

#矿机列表长度
PMN = 1000
#每分钟更新一次矿机列表
PTR = 2

#矿机优先级排序
ALLOC_MODE=0

#上传时加载到内存中最大数据大小(M)
uploadFileMaxMemory=300
#最大块上传并发数
uploadBlockThreadNum=200
#最大分片上传并发数
uploadShardThreadNum=3000

#最大分片下载并发数
downloadThread =200

#连接超时
P2PHOST_CONNECTTIMEOUT=15000
#请求超时
P2PHOST_WRITEOUT=10000
#下载重试次数
downloadRetryTimes=8
#从矿机取Token重试次数
uploadShardRetryTimes=3

s3cache="/mnt/YTS3/s3cache"
s3port="8080"
#同步模式:0上传到yotta返回 1写入cache目录返回,异步写入yotta 2写入cache目录返回,执行异步离线编码
syncmode=1
#本地缓存目录,不默认在服务所属目录的cache子目录
cache=
#缓存空间上限(G)
```

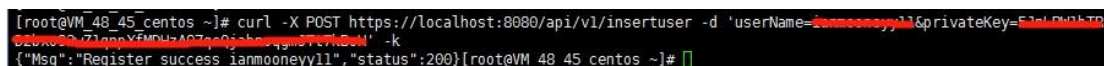
If the memory and bandwidth of the machine are sufficient, parameters such as load memory and block concurrency can be adjusted accordingly. After parameter modification, the service should be restarted.

3.2 User Registration (Login)

Each time after restarting the service, you need to register the user before uploading and downloading the files. Execute commands like the following:

```
"curl -X POST -k https://localhost:8080/api/v1/insertuser -d 'userName=***&privateKey=***' -k"
```

Fill in your user name and private key. The user name is 12-bit character length. The private key is the user's storage private key and it is registered successfully when "status" :200 , as shown in the figure below:



```
[root@VM_48_45_centos ~]# curl -X POST https://localhost:8080/api/v1/insertuser -d 'userName=ianmooney11&privateKey=51e1P0H1TB  
02w020_7100vYEMDU=407ac0gsho4q0P7V10B0N!' -k  
{"Msg": "Register success ianmooney11", "status": 200}[root@VM_48_45_centos ~]#
```

After successful registration, you can upload and download files.

3.3 Install s3cmd tool

Execute "yum install s3cmd -y"



```
-----  
Install 1 Package  
-----  
Total download size: 194 k  
Installed size: 754 k  
Is this ok [y/d/N]: y  
Downloading packages:  
s3cmd-2.1.0-1.el7.noarch.rpm | 194 kB 00:00:05  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
Installing : s3cmd-2.1.0-1.el7.noarch 1/1  
Verifying : s3cmd-2.1.0-1.el7.noarch 1/1  
Installed:  
s3cmd.noarch 0:2.1.0-1.el7  
Complete!
```

Configure s3cmd after installation, and execute "s3cmd --if configure" for the first time. Fill in the address, public and private keys and other information according to the corresponding highlighted parameters

Modify "check_ssl_certificate = False"

```
cache_file =
check_ssl_certificate = False
check_ssl_hostname = True
cloudfront_host = cloudfront.amazonaws.com
```

After configuration, you can execute "s3cmd ls" list operation to check whether the configuration is available

```
[root@nm-yaceji03 ~]# s3cmd ls
2020-12-14 11:18 s3://2020-12-04
2020-12-14 11:18 s3://2020-12-07
2020-12-14 11:18 s3://2020-12-08
2020-12-14 11:18 s3://2020-12-09
2020-12-14 11:18 s3://2020-12-10
2020-12-14 11:18 s3://2020-12-11
2020-12-14 11:18 s3://2020-12-14
2020-12-14 11:18 s3://bucket0
2020-12-14 11:18 s3://bucket1
2020-12-14 11:18 s3://forup
2020-12-14 11:18 s3://newbucket
2020-12-14 11:18 s3://polly
2020-12-14 11:18 s3://polly.2020-11
2020-12-14 11:18 s3://polly.2020-12
2020-12-14 11:18 s3://test
2020-12-14 11:18 s3://tmpupload-yunpan-1
2020-12-14 11:18 s3://ybscan
2020-12-14 11:18 s3://yunpan-1
2020-12-14 11:18 s3://zhangzhengyan
[root@nm-yaceji03 ~]#
```

At this point, the test environment is successfully installed, and functions and performance related tests can be performed later.

4. Multi-user registration

Multiple users can register, upload and download files through the same S3 Server. We can register according to the user guidelines mentioned above using different user names and private keys for registration, and then use different clients to configure S3CMD to connect to the current S3 Server to achieve multi-user upload and download.

Use the following command to register:

```
[root@VM_48_45_centos ~]# curl -X POST https://localhost:8080/api/v1/insertuser -d 'userName=ianmooney11&privateKey=51olDh4kTQ
02Ux0c0z1gppYfMDUzAG2g0jshoqgmsrFkD0M' -k
{"Msg": "Register success ianmooney11", "status": 200}[root@VM_48_45_centos ~]#
```

If you register localhost on a remote side, change to

S3 Server's IP address as follows:

```
[root@nm-yaceji03 ~]# tail -n 10 /var/log/...
[10:57:47.848][Infos][Booter]Node Addr 0:/ip4/192.168.1.195/tcp/35660
[10:57:47.848][Infos][Booter]Node Addr 1:/ip4/117.161.72.86/tcp/35660
[10:57:47.848][Infos][Booter]Node Addr 2:/ip6/fe80::ff0a:74e9:dec9:b046/tcp/35660
[10:57:47.848][Infos][Booter]Node Addr 3:/ip6/fe80::ebe7:1391:db33:c8c9/tcp/35660
[10:57:47.848][Infos][Booter]Node Addr 4:/ip6/fe80::838e:e545:ffba:baef/tcp/35660
[10:57:47.959][Infos][NodeMgr]Snlist init ok,Size:21
[10:57:48.015][Infos][Cache]LocalDB init...Path:/mnt/YTS3/cache/dbcache/cache.db
[10:57:48.015][Infos][Cache]Sum cache size 0
[10:57:48.015][Infos][Cache]Clearing expired files...
[10:57:48.017][Infos]HTTPServer start Success 8080
[10:57:59.746][Infos][Regist]User 'ianmooney11' registration successful,ID-KeyNumber:39/0,at sn 18
[10:58:00.042][Infos][PreAllocNode]Return 351 nodes,Excludes 0 nodes.
[10:58:00.768][Infos]User Register Success,UserName: ianmooney11
[10:58:31.045][Infos][Regist]User 'pollyzhang11' registration successful,ID-KeyNumber:151/0,at sn 4
[10:58:31.125][Infos][PreAllocNode]Return 351 nodes,Excludes 0 nodes.
[10:58:31.158][Infos]User Register Success,UserName: pollyzhang11
```

After a successful registration, you can configure S3CMD in your own client and fill in your own user information, as shown in the above configuration process of installing S3CMD. After the configuration is complete, opening the respective buckets will display your own bucket list

User 1:

```
[root@nm-yaceji03 ~]# s3cmd ls
2020-12-29 11:40 s3://2020-12-04
2020-12-29 11:40 s3://2020-12-07
2020-12-29 11:40 s3://2020-12-08
2020-12-29 11:40 s3://2020-12-09
2020-12-29 11:40 s3://2020-12-10
2020-12-29 11:40 s3://2020-12-11
2020-12-29 11:40 s3://2020-12-14
2020-12-29 11:40 s3://2020-12-15
2020-12-29 11:40 s3://2020-12-16
2020-12-29 11:40 s3://2020-12-17
2020-12-29 11:40 s3://2020-12-21
2020-12-29 11:40 s3://2020-12-22
2020-12-29 11:40 s3://2020-12-24
2020-12-29 11:40 s3://2020-12-28
2020-12-29 11:40 s3://20201228
2020-12-29 11:40 s3://bucket0
2020-12-29 11:40 s3://bucket1
2020-12-29 11:40 s3://forup
2020-12-29 11:40 s3://new-bucket-d35e042f
2020-12-29 11:40 s3://newbucket
2020-12-29 11:40 s3://nnnn
2020-12-29 11:40 s3://polly
2020-12-29 11:40 s3://polly.2020-11
2020-12-29 11:40 s3://polly.2020-12
2020-12-29 11:40 s3://ssss
2020-12-29 11:40 s3://test
2020-12-29 11:40 s3://tmpupload-yunpan-1
2020-12-29 11:40 s3://ybscan
2020-12-29 11:40 s3://yunpan-1
2020-12-29 11:40 s3://zhangzhengyan
```

User 2:

```
[root@nm-zhengyan-ceshiji ~]# s3cmd ls
2020-12-29 11:45 s3://2zzzznbd1
2020-12-29 11:45 s3://33wvs
2020-12-29 11:45 s3://bucket0
2020-12-29 11:45 s3://bucket1
2020-12-29 11:45 s3://bucket2
2020-12-29 11:45 s3://bucket3
2020-12-29 11:45 s3://bucket4
2020-12-29 11:45 s3://bucket5
2020-12-29 11:45 s3://bucket6
2020-12-29 11:45 s3://bucket7
2020-12-29 11:45 s3://bucket8
2020-12-29 11:45 s3://bucket9
2020-12-29 11:45 s3://gotest
2020-12-29 11:45 s3://new-bucket-0099adc0
2020-12-29 11:45 s3://new-bucket-29380614
2020-12-29 11:45 s3://new-bucket-7a6ccda6
2020-12-29 11:45 s3://owner
2020-12-29 11:45 s3://phs-test
2020-12-29 11:45 s3://polly
2020-12-29 11:45 s3://test
2020-12-29 11:45 s3://test03
2020-12-29 11:45 s3://test04
2020-12-29 11:45 s3://test05
2020-12-29 11:45 s3://test07
2020-12-29 11:45 s3://test08
2020-12-29 11:45 s3://test09
2020-12-29 11:45 s3://test1
2020-12-29 11:45 s3://test10
```

At this point, multiple users can upload and download files at the same time