

# Go 版本s3 linux 环境部署手册

## 1、说明

这里使用的是 centos7+，如果部署使用的是其他 linux 系统，请使用与之相应的命令

## 2、s3 下载安装

### 2.1 下载解压

将 s3 安装包先复制到指定的测试机（安装包可以到官网<https://www.yottachain.io/>，点击首页->下载（或开始使用）页面 ->下载Go S3 Client-linux)

然后进行解压放到指定目录，如下图：

```
[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          G0_YTS3_v2.0.0.17.tar.gz  iperf-3.9         temp
[root@nm-yaceji03 ~]# tar zxf G0_YTS3_v2.0.0.17.tar.gz
[root@nm-yaceji03 ~]# ls
anaconda-ks.cfg  docker-20200714.tar.gz  G0_YTS3_v2.0.0.17.tar.gz  iperf-3.9
config          ftp_download.sh        IdcNodeInit.tar.gz        iperf-3.9.tar.gz
docker          G0_YTS3_Linux         init                      swap
[root@nm-yaceji03 ~]#
```

解压成功后，将 GO\_YTS3\_Linux 目录复制到相应的位置，我这里复制到了/mnt 目录

下，然后进行服务安装和启动，目录结构如下图所示：

```
[root@VM_32_2_centos G0_YTS3_Linux]# ls
bin  conf  crt  YTS3
[root@VM_32_2_centos G0_YTS3_Linux]#
```

### 2.2 安装运行

进入s3 的 bin 目录，执行./YTS3 install 安装服务

```
[root@nm-yaceji03 G0_YTS3_Linux]# ./YTS3 install
I: 10:52:14 Path:./YTS3
I: 10:52:14 Install OK.
[root@nm-yaceji03 G0_YTS3_Linux]#
```

查看安装情况，执行 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]#
```

启动服务后查看状态，执行 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]#
```

查看日志可以看到服务已经在运行

```
[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、测试前准备

正式测试前要根据自己测试机的实际情况进行相关配置，并安装相应的测试工具

#### 3.1 配置修改

查看 conf 目录下ytf.properties 文件如下，然后根据情况进行配置修改

```
[root@nm-yaceji03 G0_YTS3_Linux]# cat conf/ytf.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n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=0
#从矿机取Token重试次数
uploadShardRetryTimes=3

s3cache="/mnt/YTS3/s3cache"
s3port="8080"
#同步模式:0上上传到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,不填默认在服务所属目录的cache子目录
cache=
#缓存空间上限(G)
```

如果机器的内存和带宽充足，可以相应的调整加载内存和块并发等参数，参数修改后要重新启动服务。

### 3.2 用户注册 (登陆)

每次重新启动服务后上传下载文件之前都需要注册用户，执行类似如下命令

```
curl -X POST -k https://localhost:8080/api/v1/insertuser -d
```

```
'userName=***&privateKey=***'
```

填写自己的用户名和私钥，用户名为 12 位长度字符，私钥为用户的存储私钥

注册成功如下图所示：

```
[root@VM_48_45_centos ~]# curl -X POST https://localhost:8080/api/v1/insertuser -d 'userName=ianmooney11&privateKey=57el-Du2hTB  
026x02-7199VfMDH=AP7as0jshs4q3t7N3W' -k  
{\"Msg\": \"Register success ianmooney11\", \"status\": 200}[root@VM_48_45_centos ~]#
```

注册成功后就可以使用工具进行文件的上传和下载。

### 3.3 安装 s3cmd 工具

执行yum install s3cmd -y

```
=====  
Install 1 Package  
=====  
Total download size: 194 k  
Installed size: 754 k  
Is this ok [y/d/M]: 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!
```

安装后配置 s3cmd，初次使用执行 s3cmd --configure 按照相应的提示填写地址和

公私钥等信息

```
[root@VM_32_2_centos ~]# s3cmd --configure  
Enter new values or accept defaults in brackets with Enter.  
Refer to user manual for detailed description of all options. 输入公钥  
Access key and Secret key are your identifiers for Amazon S3. Leave them empty for using the env variables.  
Access Key: [█]
```

```
[root@VM_32_2_centos ~]# s3cmd --configure  
Enter new values or accept defaults in brackets with Enter.  
Refer to user manual for detailed description of all options. 输入私钥  
Access key and Secret key are your identifiers for Amazon S3. Leave them empty for using the env variables.  
Access Key: [█]  
Secret Key: [█]
```

```
[root@VM_32_2_centos ~]# s3cmd --configure
Enter new values or accept defaults in brackets with Enter.
Refer to user manual for detailed description of all options.

Access key and Secret key are your identifiers for Amazon S3. Leave them empty for using the env variables.
Access Key: AKIAIOSFODNN7EXAMPLE
Secret Key: AJNANV777EXAMPLE
Default Region [US]:

Use "s3.amazonaws.com" for S3 Endpoint and not modify it to the target Amazon S3.
S3 Endpoint [s3.amazonaws.com]: localhost:8083

Use "%(bucket)s.s3.amazonaws.com" to the target Amazon S3. "%(bucket)s" and "%(location)s" vars can be used
if the target S3 system supports dns based buckets.
DNS-style bucket+hostname:port template for accessing a bucket [%s.s3.amazonaws.com]: localhost:8083
```

```
Encryption password:
Path to GPG program [/bin/gpg]:

When using secure HTTPS protocol all communication with Amazon S3
servers is protected from 3rd party eavesdropping. This method is
slower than plain HTTP, and can only be proxied with Python 2.7 or newer
Use HTTPS protocol [Yes]: No
```

最终的配置文件如下:

```
cloudfront_host = cloudfront.amazonaws.com
content_disposition =
content_type =
default_mime_type = binary/octet-stream
delay_updates = False
delete_after = False
delete_after_fetch = False
delete_removed = False
dry_run = False
enable_multipart = True
encrypt = False
expiry_date =
expiry_days =
expiry_prefix =
follow_symlinks = False
force = False
get_continue = False
gpg_command = /usr/bin/gpg
gpg_decrypt = %(gpg_command)s -d --verbose --no-use-agent --batch --yes --passphrase-fd %(passphrase_fd)s -o %(output_file)s %(input_file)s
gpg_encrypt = %(gpg_command)s -c --verbose --no-use-agent --batch --yes --passphrase-fd %(passphrase_fd)s -o %(output_file)s %(input_file)s
gpg_passphrase =
guess_mime_type = True
host_base = localhost:8083
host_bucket = localhost:8083
human_readable_sizes = False
invalidate_default_index_on_cf = False
invalidate_default_index_root_on_cf = True
invalidate_on_cf = False
```

支持 https 协议要修改部分配置, 编辑默认配置文件 vi /root/.s3cfg

修改 use\_https = True

```
use_http_expect = False
use_https = True
use_mime_magic = True
verbosity = WARNING
```

修改 check\_ssl\_certificate = False

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

配置后可以执行 s3cmd list 操作检查配置是否可用

```
[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 ~]#
```

到此测试环境安装成功，后续就可以进行功能和性能等相关测试。

#### 4、多用户注册

多个用户可以通过同一个s3 server 进行注册并上传和下载文件，我们可以按照如上文所说的用户注册，使用不同的用户名和私钥进行注册，然后使用不同的客户机配置s3cmd 连接到当前的 s3 server 实现多用户上传下载

使用如下命令注册：

```
[root@VM_48_45_centos ~]# curl -X POST https://localhost:8080/api/v1/insertuser -d 'userName=ianmooneyy11&privateKey=51a10h1kTR
{"Msg": "Register success ianmooneyy11", "status": 200}[root@VM_48_45_centos ~]#
```

如果在远端注册 localhost 要修改为 s3 server 的 ip 地址

如下为两个用户注册功能的日志：

```
[root@nm-yaceji03 ~]# tail -n 10 /log/log
[10:57:47.848][Infos][BooTer]Node Addr0: ip4/192.168.1.195/tcp/35660
[10:57:47.848][Infos][BooTer]Node Addr1: ip4/117.161.72.86/tcp/35660
[10:57:47.848][Infos][BooTer]Node Addr2: ip6/fe80::ff0a:74e9:dec9:b046/tcp/35660
[10:57:47.848][Infos][BooTer]Node Addr3: ip6/fe80::ebe7:1391:db33:c8c9/tcp/35660
[10:57:47.848][Infos][BooTer]Node Addr4: 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 'ianmooneyy11' 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: ianmooneyy11
[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
```

注册成功后可以分别在各自的客户端配置 s3cmd，填写自己的用户信息，如上文安装 s3cmd 的配置过程，配置完成后查看各自的 bucket 会显示自己的 bucket 列表

用户 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
```

用户 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
```

此时多个用户就可以同时上传和下载文件了