CentOS7+FastDFS+Nginx 安装教程

1 安装 libfastcommon

1.1 获取 libfastcommon 安装包：

```
wget    https://github.com/happyfish100/libfastcommon/archive/V1.0.39.tar.gz
```

1.2 解压安装包：tar -zxvf V1.0.39.tar.gz

1.3 进入目录：cd libfastcommon-1.0.39

1.4 执行编译：./make.sh

1.5 安装：./make.sh install

> 可能遇到的问题：
>
> -bash: make: command not found
>
> -bash: gcc: command not found
>
> 解决方案：
>
> debian 通过 apt-get install gcc make 安装
>
> centos 通过 yum -y install gcc make 安装

2 安装 FastDFS

2.1 获取 fdfs 安装包：

```
wget https://github.com/happyfish100/fastdfs/archive/V5.11.tar.gz
```

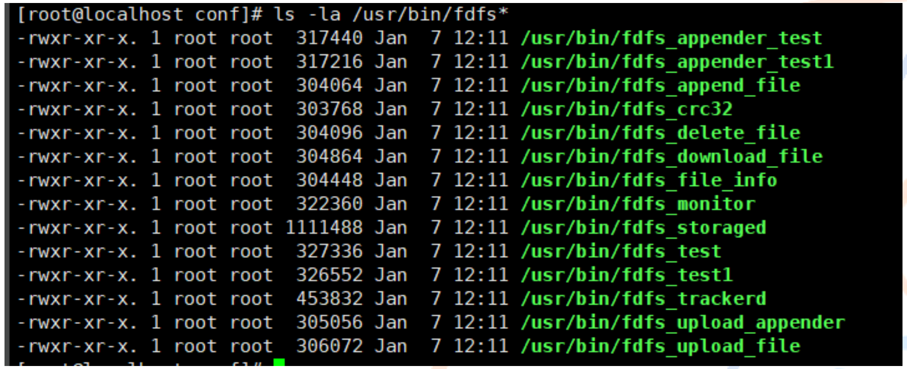
2.2 解压安装包：tar -zxvf V5.11.tar.gz

2.3 进入目录：cd fastdfs-5.11

2.4 执行编译：./make.sh

2.5 安装：./make.sh install

2.6 查看可执行命令：ls -la /usr/bin/fdfs*



```
[root@localhost conf]# ls -la /usr/bin/fdfs*
-rwxr-xr-x. 1 root root  317440 Jan  7 12:11 /usr/bin/fdfs_appender_test
-rwxr-xr-x. 1 root root  317216 Jan  7 12:11 /usr/bin/fdfs_appender_test1
-rwxr-xr-x. 1 root root  304064 Jan  7 12:11 /usr/bin/fdfs_append_file
-rwxr-xr-x. 1 root root  303768 Jan  7 12:11 /usr/bin/fdfs_crc32
-rwxr-xr-x. 1 root root  304096 Jan  7 12:11 /usr/bin/fdfs_delete_file
-rwxr-xr-x. 1 root root  304864 Jan  7 12:11 /usr/bin/fdfs_download_file
-rwxr-xr-x. 1 root root  304448 Jan  7 12:11 /usr/bin/fdfs_file_info
-rwxr-xr-x. 1 root root  322360 Jan  7 12:11 /usr/bin/fdfs_monitor
-rwxr-xr-x. 1 root root 1111488 Jan  7 12:11 /usr/bin/fdfs_storaged
-rwxr-xr-x. 1 root root  327336 Jan  7 12:11 /usr/bin/fdfs_test
-rwxr-xr-x. 1 root root  326552 Jan  7 12:11 /usr/bin/fdfs_test1
-rwxr-xr-x. 1 root root  453832 Jan  7 12:11 /usr/bin/fdfs_trackerd
-rwxr-xr-x. 1 root root  305056 Jan  7 12:11 /usr/bin/fdfs_upload_appender
-rwxr-xr-x. 1 root root  306072 Jan  7 12:11 /usr/bin/fdfs_upload_file
```

3 配置 Tracker 服务

3.1 进入/etc/fdfs 目录，有四个.sample 后缀的文件（自动生成的 fdfs 模板配置文件），通过 cp 命令拷贝 tracker.conf.sample，删除.sample 后缀作为正式文件：

3.1.1 cd /etc/fdfs/

3.1.2 cp tracker.conf.sample tracker.conf

3.1.3 编辑 tracker.conf: vim tracker.conf

修改相关参数：

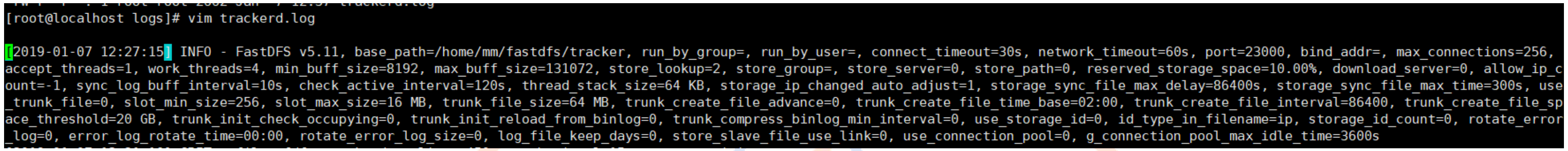
```
base_path=/home/fastdfs/tracker     #tracker 存储 data 和 log 的跟路径
port=22122                          #tracker 默认 22122
http.server_port=80                 #http 端口，需要和 nginx 相同
```

3.2 启动 tracker（支持 start|stop|restart）

/usr/bin/fdfs_trackerd   /etc/fdfs/tracker.conf   start

3.3 查看 tracker 启动日志：

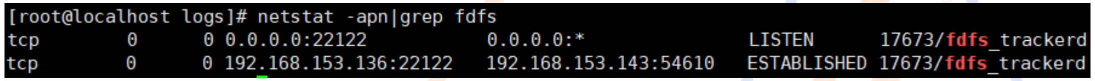
进入刚刚指定的 base_path(/home/fastdfs/tracker)中有个 logs 目录，查看 tracker.log 文件



```
[root@localhost logs]# vim trackerd.log

[2019-01-07 12:27:15] INFO - FastDFS v5.11, base_path=/home/mm/fastdfs/tracker, run_by_group=, run_by_user=, connect_timeout=30s, network_timeout=60s, port=23000, bind_addr=, max_connections=256,
accept_threads=1, work_threads=4, min_buff_size=8192, max_buff_size=131072, store_lookup=2, store_group=, store_server=0, store_path=0, reserved_storage_space=10.00%, download_server=0, allow_ip_c
ount=-1, sync_log_buff_interval=10s, check_active_interval=120s, thread_stack_size=64 KB, storage_ip_changed_auto_adjust=1, storage_sync_file_max_delay=86400s, storage_sync_file_max_time=300s, use
_trunk_file=0, slot_min_size=256, slot_max_size=16 MB, trunk_file_size=64 MB, trunk_create_file_advance=0, trunk_create_file_time_base=02:00, trunk_create_file_interval=86400, trunk_create_file_sp
ace_threshold=20 GB, trunk_init_check_occupying=0, trunk_init_reload_from_binlog=0, trunk_compress_binlog_min_interval=0, use_storage_id=0, id_type_in_filename=ip, storage_id_count=0, rotate_error
_log=0, error_log_rotate_time=00:00, rotate_error_log_size=0, log_file_keep_days=0, store_slave_file_use_link=0, use_connection_pool=0, g_connection_pool_max_idle_time=3600s
```

3.4 查看端口情况：netstat -apn|grep fdfs



```
[root@localhost logs]# netstat -apn|grep fdfs
tcp        0      0 0.0.0.0:22122           0.0.0.0:*               LISTEN      17673/fdfs_trackerd
tcp        0      0 192.168.153.136:22122   192.168.153.143:54610   ESTABLISHED 17673/fdfs_trackerd
```

可能遇到的报错：

/usr/bin/fdfs_trackerd: error while loading shared libraries: libfastcommon.so: cannot open shared object file: No such file or directory

解决方案：建立 libfastcommon.so 软链接

ln -s /usr/lib64/libfastcommon.so /usr/local/lib/libfastcommon.so

ln -s /usr/lib64/libfastcommon.so /usr/lib/libfastcommon.so

4 配置 Storage 服务

4.1 cd /etc/fdfs

进入/etc/fdfs 目录，有 cp 命令拷贝 storage.conf.sample，删除.sample 后缀作为正式文件;

4.2 编辑 storage.conf：vim storage.conf

修改相关参数：

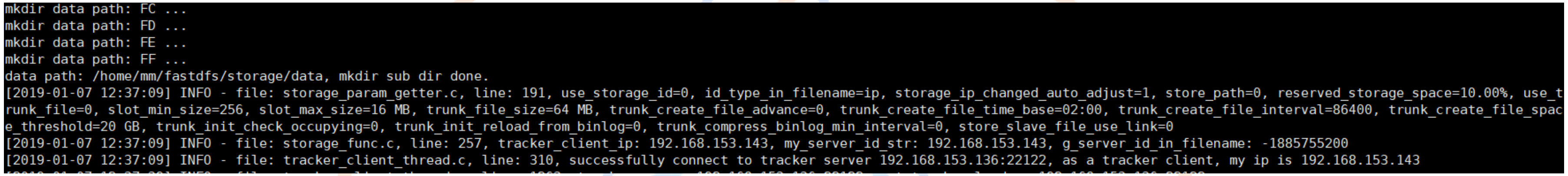
base_path=/home /fastdfs/storage	#storage 存储 data 和 log 的跟路径，必须提前创建好
port=23000	#storge 默认 23000，同一个组的 storage 端口号必须一致
group_name=group1	#默认组名，根据实际情况修改
store_path_count=1	#存储路径个数，需要和 store_path 个数匹配
store_path0=/home/mm/fastdfs/storage	#如果为空，则使用 base_path
tracker_server=10.122.149.211:22122	#配置该 storage 监听的 tracker 的 ip 和 port

4.3 启动 storage（支持 start|stop|restart）:

```
/usr/bin/fdfs_storaged  /etc/fdfs/storage.conf start
```

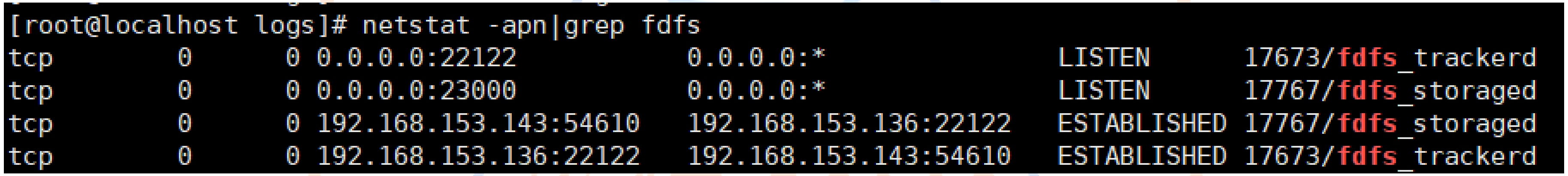
4.4 查看 storage 启动日志：进入刚刚指定的 base_path(/home/fastdfs/storage)中有个 logs 目录，查看 storage.log 文件

4.5 此时再查看 tracker 日志：发现已经开始选举，并且作为唯一的一个 tracker，被选举为 leader



```
mkdir data path: FC ...
mkdir data path: FD ...
mkdir data path: FE ...
mkdir data path: FF ...
data path: /home/mm/fastdfs/storage/data, mkdir sub dir done.
[2019-01-07 12:37:09] INFO - file: storage_param_getter.c, line: 191, use_storage_id=0, id_type_in_filename=ip, storage_ip_changed_auto_adjust=1, store_path=0, reserved_storage_space=10.00%, use_t
runk_file=0, slot_min_size=256, slot_max_size=16 MB, trunk_file_size=64 MB, trunk_create_file_advance=0, trunk_create_file_time_base=02:00, trunk_create_file_interval=86400, trunk_create_file_spac
e_threshold=20 GB, trunk_init_check_occupying=0, trunk_init_reload_from_binlog=0, trunk_compress_binlog_min_interval=0, store_slave_file_use_link=0
[2019-01-07 12:37:09] INFO - file: storage_func.c, line: 257, tracker_client_ip: 192.168.153.143, my_server_id_str: 192.168.153.143, g_server_id_in_filename: -1885755200
[2019-01-07 12:37:09] INFO - file: tracker_client_thread.c, line: 310, successfully connect to tracker server 192.168.153.136:22122, as a tracker client, my ip is 192.168.153.143
```

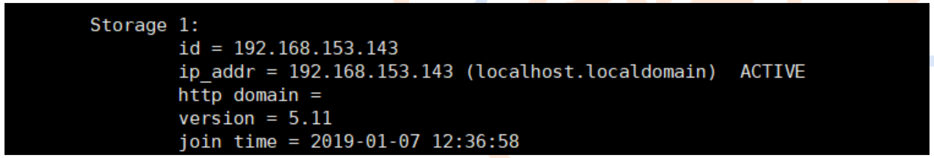
4.6 查看端口情况：netstat -apn|grep fdfs



```
[root@localhost logs]# netstat -apn|grep fdfs
tcp        0      0 0.0.0.0:22122           0.0.0.0:*               LISTEN      17673/fdfs_trackerd
tcp        0      0 0.0.0.0:23000           0.0.0.0:*               LISTEN      17767/fdfs_storaged
tcp        0      0 192.168.153.143:54610   192.168.153.136:22122   ESTABLISHED 17767/fdfs_storaged
tcp        0      0 192.168.153.136:22122   192.168.153.143:54610   ESTABLISHED 17673/fdfs_trackerd
```

4.8 通过 monitor 来查看 storage 是否成功绑定：

/usr/bin/fdfs_monitor /etc/fdfs/storage.conf



```
        Storage 1:
                id = 192.168.153.143
                ip_addr = 192.168.153.143 (localhost.localdomain)  ACTIVE
                http domain =
                version = 5.11
                join time = 2019-01-07 12:36:58
```

5 安装 Nginx 和 fastdfs-nginx-module 模块

5.1 获取 Nginx 安装包

```
wget http://nginx.org/download/nginx-1.15.8.tar.gz
```

5.2 下载 fastdfs-nginx-module 安装包

```
wget https://github.com/happyfish100/fastdfs-nginx-module/archive/V1.20.tar.gz
```

5.3 解压 nginx：tar -zxvf nginx-1.15.8.tar.gz

5.4 解压 fastdfs-nginx-module：tar -xvf V1.20.tar.gz -C /usr/local/src

5.5 进入 nginx 目录：cd nginx-1.15.8 安装依赖的库

```
yum -y  install libpcre3 libpcre3-dev openssl libssl-dev libperl-dev
```

5.6 配置，并加载 fastdfs-nginx-module 模块：

```
./configure --prefix=/usr/local/nginx  --add-module=/usr/local/src/fastdfs-nginx-module-1.20/src/
```

5.7 编译安装：

```
make
make install
```

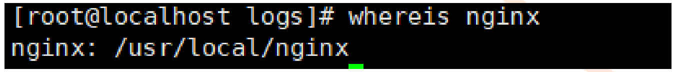
可能的报错：

```
/usr/include/fastdfs/fdfs_define.h:15:27: fatal error: common_define.h: No such file or directory
```

解决方案：修改 fastdfs-nginx-module-1.20/src/config 文件，然后重新第 5.6 步开始

```
ngx_module_incs="/usr/include/fastdfs /usr/include/fastcommon/"
CORE_INCS="$CORE_INCS /usr/include/fastdfs /usr/include/fastcommon/"
```

5.8 查看安装路径：whereis nginx



```
[root@localhost logs]# whereis nginx
nginx: /usr/local/nginx
```

5.9 启动、停止：

```
cd /usr/local/nginx/sbin/
```

```
./nginx                 #启动
./nginx -s stop         #此方式相当于先查出 nginx 进程 id 再使用 kill 命令强制杀掉进程
./nginx -s quit         #此方式停止步骤是待 nginx 进程处理任务完毕进行停止
./nginx -s reload
```

5.10 验证启动状态：wget http://本机 IP 地址

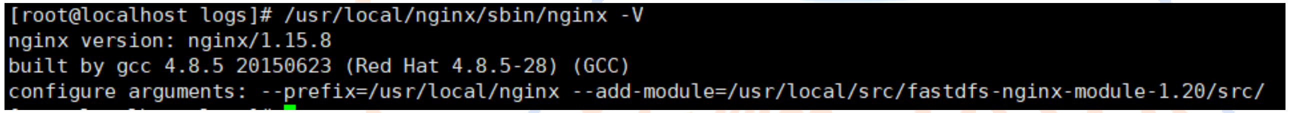
如果没有反应：

修改 nginx.conf 配置文件： server {

```
        listen          80;
        server_name     192.168.153.136;( 为当前机器 ip )
```

5.11 查看此时的 nginx 版本：发现 fastdfs 模块已经安装好了

/usr/local/nginx/sbin/nginx -V



```
[root@localhost logs]# /usr/local/nginx/sbin/nginx -V
nginx version: nginx/1.15.8
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-28) (GCC)
configure arguments: --prefix=/usr/local/nginx --add-module=/usr/local/src/fastdfs-nginx-module-1.20/src/
```

6 配置 Nginx 和 fastdfs-nginx-module 模块

6.1 配置 mod-fastdfs.conf，并拷贝到/etc/fdfs 文件目录下

```
cd fastdfs-nginx-module-1.20/src/
cp mod_fastdfs.conf /etc/fdfs
```

6.2 进入/etc/fdfs 修改 mod-fastdfs.conf：

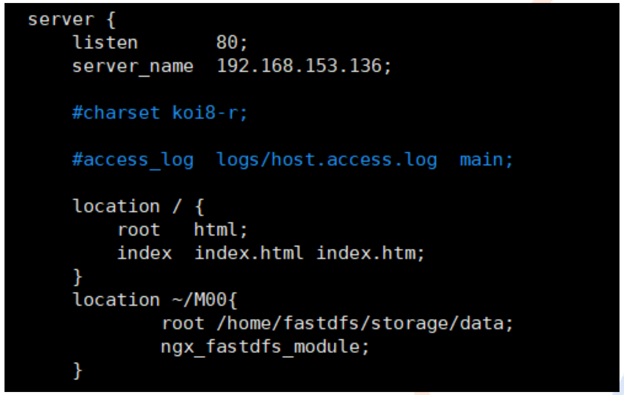
```
base_path=/home/fastdfs
tracker_server=192.168.153.136:22122          #tracker 的地址
url_have_group_name=true                      #url 是否包含 group 名称
storage_server_port=23000                     #需要和 storage 配置的相同
store_path_count=1                            #存储路径个数，需要和 store_path 个数匹配
```

store_path0=/home /fastdfs/storage　　#文件存储的位置

6.3 配置 nginx，80 端口 server 增加 location 如图：

cd /usr/local/nginx/conf/

vi nginx.conf



```
server {
    listen       80;
    server_name  192.168.153.136;

    #charset koi8-r;

    #access_log  logs/host.access.log  main;

    location / {
        root   html;
        index  index.html index.htm;
    }
    location ~/M00{
            root /home/fastdfs/storage/data;
            ngx_fastdfs_module;
    }
```

6.4 最后需要拷贝 fastdfs 解压目录中的 http.conf 和 mime.types：

cd /usr/local/src/fastdfs-5.11/conf

cp mime.types http.conf　/etc/fdfs/

7 FastDFS 常用命令测试

7.1 上传文件

7.1.1 进入/etc/fdfs 目录，用 cp 命令拷贝 client.conf.sample，删除.sample 后缀作为正式文件;

7.1.2 修改 client.conf 相关配置：

```
base_path=/home/fastdfs/tracker              #tracker 服务器文件路径
tracker_server=192.168.153.136:22122         #tracker 服务器 IP 地址和端口号
http.tracker_server_port=80                  # tracker 服务器的 http 端口号，必须和 tracker 的设置对应起来
```

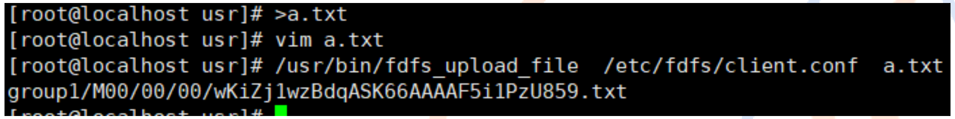
7.1.3 新建一个测试文档 1.txt，内容为 sssssssssss

命令：

/usr/bin/fdfs_upload_file　<config_file>　<local_filename>

示例：

/usr/bin/fdfs_upload_file　/etc/fdfs/client.conf　a.txt



```
[root@localhost usr]# >a.txt
[root@localhost usr]# vim a.txt
[root@localhost usr]# /usr/bin/fdfs_upload_file  /etc/fdfs/client.conf  a.txt
group1/M00/00/00/wKiZj1wzBdqASK66AAAAF5i1PzU859.txt
[root@localhost usr]# 
```

group1/M00/00/00/wKiZj1wzBdqASK66AAAAF5i1PzU859.txt

7.1.4 返回值分析

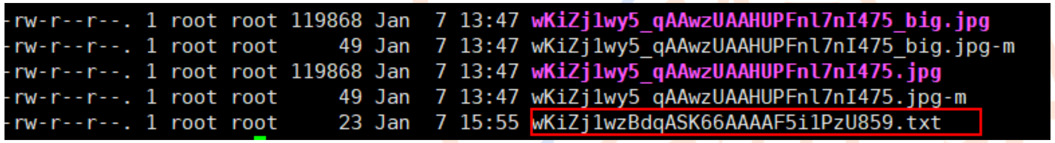
组名：group1

磁盘：M00

目录：00/00

文件名称：wKiZj1wzBdqASK66AAAAF5i1PzU859.txt

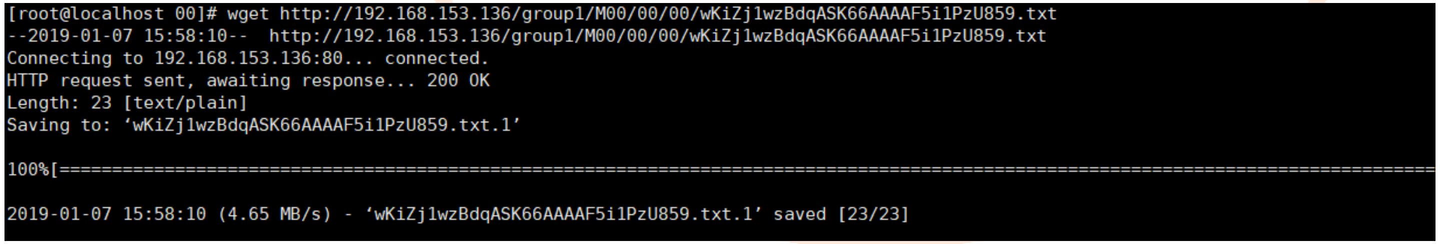
7.1.5 查看结果，进入 storage 的 data 目录：



```
-rw-r--r--. 1 root root 119868 Jan  7 13:47 wKiZj1wy5_qAAwzUAAHUPFnl7nI475_big.jpg
-rw-r--r--. 1 root root     49 Jan  7 13:47 wKiZj1wy5_qAAwzUAAHUPFnl7nI475_big.jpg-m
-rw-r--r--. 1 root root 119868 Jan  7 13:47 wKiZj1wy5_qAAwzUAAHUPFnl7nI475.jpg
-rw-r--r--. 1 root root     49 Jan  7 13:47 wKiZj1wy5_qAAwzUAAHUPFnl7nI475.jpg-m
-rw-r--r--. 1 root root     23 Jan  7 15:55 wKiZj1wzBdqASK66AAAAF5i1PzU859.txt
```

7.2 通过 wget 和浏览器方式访问成功：

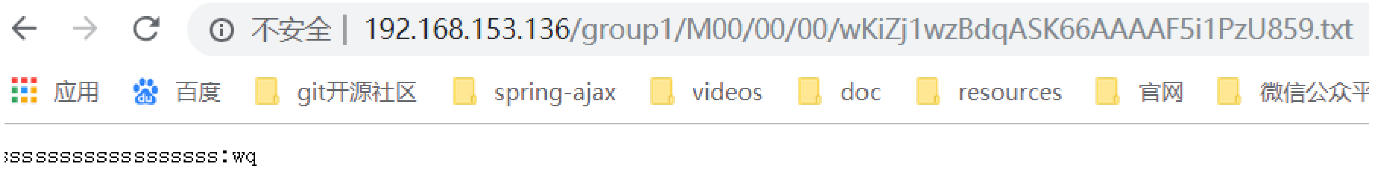
wget http://192.168.153.136/group1/M00/00/00/wKiZj1wzBdqASK66AAAAF5i1PzU859.txt



```
[root@localhost 00]# wget http://192.168.153.136/group1/M00/00/00/wKiZj1wzBdqASK66AAAAF5i1PzU859.txt
--2019-01-07 15:58:10--  http://192.168.153.136/group1/M00/00/00/wKiZj1wzBdqASK66AAAAF5i1PzU859.txt
Connecting to 192.168.153.136:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 23 [text/plain]
Saving to: ‘wKiZj1wzBdqASK66AAAAF5i1PzU859.txt.1’

100%[====================================================================================================================

2019-01-07 15:58:10 (4.65 MB/s) - ‘wKiZj1wzBdqASK66AAAAF5i1PzU859.txt.1’ saved [23/23]
```


不安全 | 192.168.153.136/group1/M00/00/00/wKiZj1wzBdqASK66AAAAF5i1PzU859.txt
应用
百度
git开源社区
spring-ajax
videos
doc
resources
官网
微信公众平

```
ssssssssssssssssss:wq
```

7.3 下载文件：

命令：

/usr/bin/fdfs_download_file   <config_file>   <file_id> [local_filename]

示例：

/usr/bin/fdfs_download_file /etc/fdfs/client.conf group1/M00/00/00/CnqV01trmeyAbAN0AAAABLh3frE677.txt a.txt

7.4 删除文件：

命令：

/usr/bin/fdfs_delete_file   <config_file>   <file_id>

示例：

/usr/bin/fdfs_delete_file /etc/fdfs/client.conf group1/M00/00/00/CnqV01trmeyAbAN0AAAABLh3frE677.txt

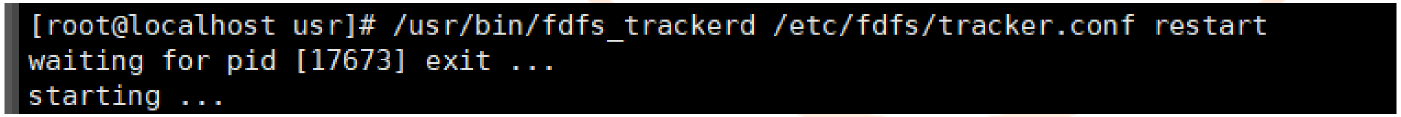
查看结果，进入 storage 的 data 目录文件不存在，通过 wget 再次获取 404：

8 确认服务是否可用

8.1 确认 tracker 是否启动

/usr/bin/fdfs_trackerd   /etc/fdfs/tracker.conf restart

如下图关闭线程再执行则表示成功

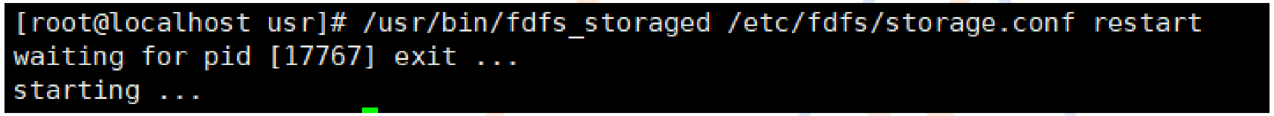


```
[root@localhost usr]# /usr/bin/fdfs_trackerd /etc/fdfs/tracker.conf restart
waiting for pid [17673] exit ...
starting ...
```

8.2 确认 storage 是否启动

/usr/bin/fdfs_storaged /etc/fdfs/storage.conf restart

如下图关闭线程再执行则表示成功



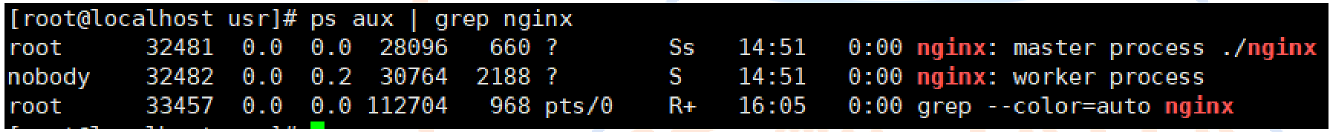
```
[root@localhost usr]# /usr/bin/fdfs_storaged /etc/fdfs/storage.conf restart
waiting for pid [17767] exit ...
starting ...
```

如果不成功，同样的命令再执行一次

8.3 确认 Nginx 是否可用

ps aux | grep nginx

查看线程是否存在



```
[root@localhost usr]# ps aux | grep nginx
root      32481  0.0  0.0  28096   660 ?        Ss   14:51   0:00 nginx: master process ./nginx
nobody    32482  0.0  0.2  30764  2188 ?        S    14:51   0:00 nginx: worker process
root      33457  0.0  0.0 112704   968 pts/0    R+   16:05   0:00 grep --color=auto nginx
```

9 java 代码客户端测试

9.1 添加依赖包

```
<dependency>
    <groupId>org.csource</groupId>
    <artifactId>fastdfs-client-java</artifactId>
    <version>1.27-SNAPSHOT</version>
</dependency>
```

获取地址：关注微信公众号：浅醉 JAVA

后台留言免费获取（公众号测试阶段，无法自动获取，留言联系）

附有简单 API 使用方法