

FastDFS服务搭建

FastDFS只有两个角色：**Tracker server**和**Storage server**。

Tracker server作为中心结点，其主要作用是负载均衡和调度。Tracker server在内存中记录分组和Storage server的状态等信息，不记录文件索引信息，占用的内存量很少。另外，客户端（应用）和Storage server访问Tracker server时，Tracker server扫描内存中的分组和Storage server信息，然后给出应答。由此可以看出Tracker server非常轻量化，不会成为系统瓶颈。

FastDFS中的Storage server在其他文件系统中通常称作Trunk server或Data server。Storage server直接利用OS的文件系统存储文件。FastDFS不会对文件进行分块存储，客户端上传的文件和Storage server上的文件一一对应。

一、单机搭建

环境说明

服务器：（centos7）

- Tracker server安装在 192.168.3.50 ；
- Storage server安装在192.168.3.51 ；

安装包：

- libfastcommon-1.0.39.tar.gz ：公共库安装包，github下载地址：<https://github.com/happyfish100/libfastcommon/releases>
- fastdfs-5.11.tar.gz ：fastdfs安装包，github下载地址：<https://github.com/happyfish100/fastdfs/releases>
- fastdfs-nginx-module-1.20.tar.gz ：fastdfs的nginx模块，github下载地址：<https://github.com/happyfish100/fastdfs-nginx-module/releases>
- nginx-1.16.0.tar.gz ：nginx 安装包，下载地址：<http://nginx.org/en/download.html>

1、安装前的环境准备（两台机器）

注意：这台机器（3.50、3.51）都需要安装

（1）安装gcc、pcre、zlib等

FastDFS是C写的，安装环境必须支持C编译。

```
yum -y install zlib zlib-devel pcre pcre-devel gcc gcc-c++ openssl openssl-devel libevent libevent-devel perl unzip net-tools wget
```

（2）新建目录，并上传安装包到该目录

```
mkdir -p /usr/local/fast
```

新建 fast 目录后，xshell上传安装包到目录下。（安装完成后可以删除安装包，避免占用空间）

2、安装libfastcommon（两台机器）

（1）进入 fast 目录

```
cd /usr/local/fast
```

（2）解压 libfastcommon-1.0.39.tar.gz

```
tar -zxvf libfastcommon-1.0.39.tar.gz
```

（3）进入解压的目录

```
cd libfastcommon-1.0.39
```

（4）编译和安装

编译：

```
./make.sh
```

如果编译没问题，则安装。有问题可能是C编译环境没准备好（现在没发现，如有度量）

```
./make.sh install
```



```
red_queue.o multi_socket_client.o skiplist_set.o
[root@zxh libfastcommon-1.0.39]# ./make.sh install
mkdir -p /usr/lib64
mkdir -p /usr/lib
mkdir -p /usr/include/fastcommon
install -m 755 libfastcommon.so /usr/lib64
install -m 644 common_define.h hash.h chain.h logger.h base64.h shared_func.h pthread_func.h ini_file_reader.h _os_define.h sockopt.h sched_thread.h ht
tp_func.h md5.h local_ip_func.h avl_tree.h ioevent.h ioevent_loop.h fast_task_queue.h fast_timer.h process_ctrl.h fast_mblock.h connection_pool.h fast_
mpool.h fast_allocator.h fast_buffer.h skiplist.h multi_skiplist.h flat_skiplist.h skiplist_common.h system_info.h fast_blocked_queue.h php7_ext_wrappe
r.h id_generator.h char_converter.h char_convert_loader.h common_blocked_queue.h multi_socket_client.h skiplist_set.h fc_list.h /usr/include/fastcommon
if [ ! -e /usr/lib/libfastcommon.so ]; then ln -s /usr/lib64/libfastcommon.so /usr/lib/libfastcommon.so; fi
[root@zxh libfastcommon-1.0.39]#
```

可以看出，libfastcommon默认安装到 /usr/lib64/ 下

（5）创建软连接。

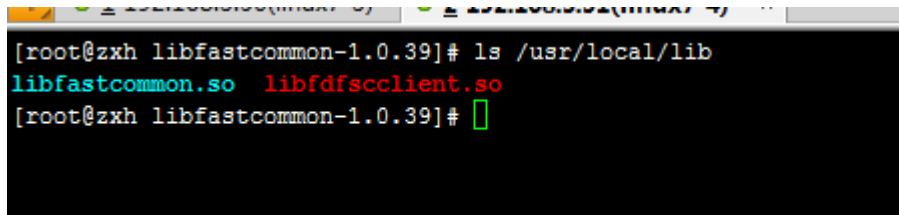
将FastDFS主程序设置的目录为 /usr/local/lib/，所有需要创建 /usr/lib64/ 下的一些核心执行程序的软连接文件。

如果没有则创建 /usr/local/lib/ 目录

```
mkdir /usr/local/lib/
```

创建软连接

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



```
[root@zxx libfastcommon-1.0.39]# ls /usr/local/lib
libfastcommon.so  libbdfscclient.so
[root@zxx libfastcommon-1.0.39]#
```

3、安装FastDFS（两台机器）

（1）解压fastdfs-5.11.tar.gz

```
cd /usr/local/fast
tar -zxvf /usr/local/fast/fastdfs-5.11.tar.gz
```

（2）进入解压目录

```
cd /usr/local/fast/fastdfs-5.11
```

（3）编译安装

编译：

```
./make.sh
```

安装

```
./make.sh install
```

```
[root@zxh fastdfs-5.11]# ./make.sh install
mkdir -p /usr/bin
mkdir -p /etc/fdfs
cp -f fdfs_trackerd /usr/bin
if [ ! -f /etc/fdfs/tracker.conf.sample ]; then cp -f ../conf/tracker.conf /etc/fdfs/tracker.conf.sample; fi
if [ ! -f /etc/fdfs/storage_ids.conf.sample ]; then cp -f ../conf/storage_ids.conf /etc/fdfs/storage_ids.conf.sample; fi
mkdir -p /usr/bin
mkdir -p /etc/fdfs
cp -f fdfs_storaged /usr/bin
if [ ! -f /etc/fdfs/storage.conf.sample ]; then cp -f ../conf/storage.conf /etc/fdfs/storage.conf.sample; fi
mkdir -p /usr/bin
mkdir -p /etc/fdfs
mkdir -p /usr/lib64
mkdir -p /usr/lib
cp -f fdfs_monitor fdfs_test fdfs_test1 fdfs_crc32 fdfs_upload_file fdfs_download_file fdfs_delete_file fdfs_file_info fdfs_appender_test fdfs_appender
_test1 fdfs_append_file fdfs_upload_appender /usr/bin
if [ 0 -eq 1 ]; then cp -f libfdfsclient.a /usr/lib64; cp -f libfdfsclient.a /usr/lib; fi
if [ 1 -eq 1 ]; then cp -f libfdfsclient.so /usr/lib64; cp -f libfdfsclient.so /usr/lib; fi
mkdir -p /usr/include/fastdfs
cp -f ../common/fdfs_define.h ../common/fdfs_global.h ../common/mime_file_parser.h ../common/fdfs_http_shared.h ../tracker/tracker_types.h ../tracker/t
racker_proto.h ../tracker/fdfs_shared_func.h ../storage/trunk_mgr/trunk_shared.h tracker_client.h storage_client.h storage_client1.h client_func.h clie
nt_global.h fdfs_client.h /usr/include/fastdfs
if [ ! -f /etc/fdfs/client.conf.sample ]; then cp -f ../conf/client.conf /etc/fdfs/client.conf.sample; fi
[root@zxh fastdfs-5.11]#
```

安装完成后，查看配置文件和脚本。

1、服务脚本（在 /etc/init.d 下）：

```
[root@zxh fastdfs-5.11]# ls /etc/init.d
fdfs_storaged fdfs_trackerd functions netconsole network README
[root@zxh fastdfs-5.11]#
```

- fdfs_storaged：Storage server脚本
- fdfs_trackerd：Trackerd server脚本

2、配置文件（在 /etc/fdfs/ 下）

```
[root@zxh fastdfs-5.11]# ls /etc/fdfs
client.conf.sample storage.conf.sample storage_ids.conf.sample tracker.conf.sample
[root@zxh fastdfs-5.11]#
```

- client.conf.sample：客户端配置文件样板
- storage.conf.sample：storage服务配置样板
- tracker.conf.sample：tracker服务配置样板
- storage_ids.conf.sample：storage id配置

3、命令行执行脚本（在 /usr/bin/ 下）

```

client.conf.sample  storage.conf.sample  storage_ids.conf.sample  tracker.conf.sample
[root@zxxh fastdfs-5.11]# cd /usr/bin/ && ls | grep fdfs
fdfs_appender_test
fdfs_appender_test1
fdfs_append_file
fdfs_crc32
fdfs_delete_file
fdfs_download_file
fdfs_file_info
fdfs_monitor
fdfs_storaged
fdfs_test
fdfs_test1
fdfs_trackerd
fdfs_upload_appender
fdfs_upload_file
[root@zxxh bin]#

```

(4) 修改FastDFS服务脚本配置 (旧版本需要 , 当前安装版本不需要)

如果正确则无需修改。

因为FastDFS服务脚本设置的bin目录为 `/usr/local/bin` 下, 但是实际上我们按照在 `/usr/bin/` 下。所以我们需要修改FastDFS配置文件中的路径。

- 修改文件 `/etc/init.d/fdfs_storaged` :将 `/usr/local/bin` 替换 `/usr/bin/`
- 修改文件 `/etc/init.d/fdfs_trackerd` :将 `/usr/local/bin` 替换 `/usr/bin/`

4、配置跟踪器Tracker server (192.168.3.50节点)

(1) 进入/etc/fdfs目录 , 并复制一份tracker.conf配置文件

```

cd /etc/fdfs
cp tracker.conf.sample tracker.conf

```

```

[root@zxxh fastdfs-5.11]# cd /etc/fdfs
[root@zxxh fdfs]# ls
client.conf.sample  storage.conf.sample  storage_ids.conf.sample  tracker.conf.sample
[root@zxxh fdfs]# cp tracker.conf.sample tracker.conf
[root@zxxh fdfs]#

```

(2) 修改tracker.conf文件

修改base_path为自己定义的路径 : `base_path=/fastdfs/tracker`

```
18 # default value is 30s
19 network_timeout=60
20
21 # the base path to store data and log files
22 #base_path=/home/yuging/fastdfs
23 base_path=/fastdfs/tracker
24
25 # max concurrent connections this server supported
26 max_connections=256
27
```

更多配置可查看：<http://bbs.chinaunix.net/thread-1941456-1-1.html>

(3) 创建指定的tracker保存路径

上一步我们指定tracker保存的目录为/fastdfs/tracker。所有要创建该目录

```
mkdir -p /fastdfs/tracker
```

(4) 防火墙开放端口

tracker.conf配置文件默认指定端口为22122，所有要开放此端口

开放端口：

```
firewall-cmd --zone=public --add-port=22122/tcp --permanent
```

重启防火墙：

```
systemctl restart firewalld.service
```

(5) 启动跟踪器

启动命令：

```
/etc/init.d/fdfs_trackerd start
```

查看进程：

```
ps -ef|grep fdfs
```

查看trackerd的数据目录

```
cd /fastdfs/tracker/ && ll
```

```
[root@zxxh ~]# /etc/init.d/fdfs_trackerd start
Starting fdfs_trackerd (via systemctl): [ 确定 ]
[root@zxxh ~]# ps -ef|grep fdfs
root      3435      1  0 12:28 ?        00:00:00 /usr/bin/fdfs_trackerd /etc/fdfs/tracker.conf
root      3443  3399  0 12:28 pts/0    00:00:00 grep --color=auto fdfs
[root@zxxh ~]# cd /fastdfs/tracker/ && ll
总用量 0
drwxr-xr-x. 2 root root 60 6月  6 12:25 data
drwxr-xr-x. 2 root root 26 6月  6 12:25 logs
[root@zxxh tracker]#
```

如果要关闭跟踪器,命令如下：

```
/etc/init.d/fdfs_trackerd stop
```

注意：千万不要用kill命令关闭，因为这样可能会丢失数据。

(6) 设置开机启动

生成环境一般需求设置开机启动一些服务，如keepalived、tomcat等

修改 /etc/rc.d/rc.local，加入配置： /etc/init.d/fdfs_trackerd start

```
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local
/etc/init.d/fdfs_trackerd start
```

5、配置FastDFS服务（192.168.3.51节点）

(1) 进入/etc/fdfs目录，并复制一份storage.conf

```
cd /etc/fdfs/
cp storage.conf.sample storage.conf
```

(2) 修改storage.conf文件

修改内容：

```
base_path=/fastdfs/storage #存储数据路径
store_path0=/fastdfs/storage
tracker_server=192.168.3.50:22122 # tracker服务
http.server_port=8888
```

(3) 创建存储目录

```
mkdir -p /fastdfs/storage
```

(4) 防火墙开放端口

Storage server配置文件/etc/fdfs/storage.conf默认端口是23000。所有防火墙需要开发端口。

开放端口：

```
firewall-cmd --zone=public --add-port=23000/tcp --permanent
```

重启防火墙：

```
systemctl restart firewalld.service
```

(5) 启动storage

注意：启动Storage server之前必须先启动Traker server

启动命令：

```
/etc/init.d/fdfs_storaged start
```

查看进程：

```
ps -ef|grep fdfs
```

查看trackerd的数据目录


```
cd /fastdfs/storage/ && ll
```

```
success
[root@zxxh fdfs]# /etc/init.d/fdfs_storaged start 启动
Reloading systemd: [ 确定 ]
Starting fdfs_storaged (via systemctl): [ 确定 ]
[root@zxxh fdfs]# ps -ef|grep fdfs 查看服务
root      16221      1  32  13:48 ?        00:00:01 /usr/bin/fdfs_storaged /etc/fdfs/storage.conf
root      16232   3405    0  13:48 pts/0    00:00:00 grep --color=auto fdfs
[root@zxxh fdfs]# cd /fastdfs/storage/ && ll 查看存储文件
总用量 12
drwxr-xr-x. 259 root root 8192 6月  6 13:48 data
drwxr-xr-x.  2 root root  26 6月  6 13:48 logs
[root@zxxh storage]#
```

如果要关闭跟踪器,命令如下：

```
/etc/init.d/fdfs_storaged stop
```

注意：千万不要用kill命令关闭，因为这样可能会丢失数据。

（6）设置开机启动

生成环境一般需求设置开机启动一些服务，如keepalived、tomcat等

修改 `/etc/rc.d/rc.local`，加入配置：`/etc/init.d/fdfs_storaged start`

到此FastDFS环境初步已经搭建完成

6、测试搭建的环境

（1）使用命令上传文件（192.168.3.50）

主要：这里是使用跟踪器上传，所以在192.168.3.50操作

1.1、进入/etc/fdfs目录

```
cd /etc/fdfs/
```

1.2、copy一份client.conf配置文件

```
cp client.conf.sample client.conf
```

1.3、修改client.conf配置文件，修改内容为：

```
base_path=/fastdfs/tracker #tracker存储路径
tracker_server=192.168.3.50:22122 #tracker 服务
```

1.4、找到命令脚本，并使用命令进行文件上传

进入脚本目录：

```
cd /usr/bin/  
ls | grep fdfs #查看脚本
```

```
[root@zxxh fdfs]# cp client.conf.sample client.conf  
[root@zxxh fdfs]# cd /usr/bin/  
[root@zxxh bin]# ls | grep fdfs  
fdfs_appender_test  
fdfs_appender_test1  
fdfs_append_file  
fdfs_crc32  
fdfs_delete_file  
fdfs_download_file  
fdfs_file_info  
fdfs_monitor  
fdfs_storaged  
fdfs_test  
fdfs_test1  
fdfs_trackerd  
fdfs_upload_appender  
fdfs_upload_file  
[root@zxxh bin]#
```

上传附件：

```
/usr/bin/fdfs_upload_file /etc/fdfs/client.conf /opt/1.jpg
```

- /usr/bin/fdfs_upload_file：上传命令脚本
- /etc/fdfs/client.conf：客户端配置文件
- /opt/1.jpg：上传的文件

```
[root@zxxh fdfs]# cd /usr/bin/  
[root@zxxh bin]# ls | grep fdfs  
fdfs_appender_test  
fdfs_appender_test1  
fdfs_append_file  
fdfs_crc32  
fdfs_delete_file  
fdfs_download_file  
fdfs_file_info  
fdfs_monitor  
fdfs_storaged  
fdfs_test  
fdfs_test1  
fdfs_trackerd  
fdfs_upload_appender  
fdfs_upload_file  
[root@zxxh bin]# /usr/bin/fdfs_upload_file /etc/fdfs/client.conf /opt/1.jpg  
[2019-06-06 14:12:32] ERROR - file: connection_pool.c, line: 130, connect to 192.168.3.51:23000 fail, errno: 113, error info: No route to host  
upload file fail, error no: 113, error info: No route to host  
[root@zxxh bin]# /usr/bin/fdfs_upload_file /etc/fdfs/client.conf /opt/1.jpg  
group1/M00/00/00/wKgDM1z4r5KAFunGAAcWmbcRMT8341.jpg  
[root@zxxh bin]#
```

上传

返回了文件的全路径

51开放端口后再次执行上传

报错：连接不上，可能是storage server服务防火墙端口23000没有开放。开放端口即可

开发端

ps:上传之前可以到storage服务 (192.168.3.51) 存储路径下看是否有数据，再对比上传后存储目录

```
root 16232 3405 0 13:48 pts/0 00:00:00 grep --color=auto fdfs
[root@zxxh fdfs]# cd /fastdfs/storage/ && ll
总用量 12
drwxr-xr-x. 259 root root 8192 6月 6 13:48 data
drwxr-xr-x. 2 root root 26 6月 6 13:48 logs
[root@zxxh storage]# ls
data logs
[root@zxxh storage]# cd data
[root@zxxh data]# ls
00 09 12 1B 24 2D 36 3F 48 51 5A 63 6C 75 7E 87 90 99 A2 AB B4 BD C6 CF D8 E1 EA F3 FC
01 0A 13 1C 25 2E 37 40 49 52 5B 64 6D 76 7F 88 91 9A A3 AC B5 BE C7 D0 D9 E2 EB F4 FD
02 0B 14 1D 26 2F 38 41 4A 53 5C 65 6E 77 80 89 92 9B A4 AD B6 BF C8 D1 DA E3 EC F5 FE fdfs_storaged.pid
03 0C 15 1E 27 30 39 42 4B 54 5D 66 6F 78 81 8A 93 9C A5 AE B7 C0 C9 D2 DB E4 ED F6 FF
04 0D 16 1F 28 31 3A 43 4C 55 5E 67 70 79 82 8B 94 9D A6 AF B8 C1 CA D3 DC E5 EE F7 FF
05 0E 17 20 29 32 3B 44 4D 56 5F 68 71 7A 83 8C 95 9E A7 B0 B9 C2 CB D4 DD E6 EF F8 storage_stat.dat
06 0F 18 21 2A 33 3C 45 4E 57 60 69 72 7B 84 8D 96 9F A8 B1 BA C3 CC D5 DE E7 F0 F9 sync
07 10 19 22 2B 34 3D 46 4F 58 61 6A 73 7C 85 8E 97 A0 A9 B2 BB C4 CD D6 DF E8 F1 FA
08 11 1A 23 2C 35 3E 47 50 59 62 6B 74 7D 86 8F 98 A1 AA B3 BC C5 CE D7 E0 E9 F2 FB
[root@zxxh data]# cd 00
[root@zxxh 00]# ls
00 07 0E 15 1C 23 2A 31 38 3F 46 4D 54 5B 62 69 70 77 7E 85 8C 93 9A A1 A8 AF B6 BD C4 CB D2 D9 E0 E7 EE F5 FC
01 08 0F 16 1D 24 2B 32 39 40 47 4E 55 5C 63 6A 71 78 7F 86 8D 94 9B A2 A9 B0 B7 BE C5 CC D3 DA E1 E8 EF F6 FD
02 09 10 17 1E 25 2C 33 3A 41 48 4F 56 5D 64 6B 72 79 80 87 8E 95 9C A3 AA B1 B8 BF C6 CD D4 DB E2 E9 F0 F7 FE
03 0A 11 18 1F 26 2D 34 3B 42 49 50 57 5E 65 6C 73 7A 81 88 8F 96 9D A4 AB B2 B9 C0 C7 CE D5 DC E3 EA F1 F8 FF
04 0B 12 19 20 27 2E 35 3C 43 4A 51 58 5F 66 6D 74 7B 82 89 90 97 9E A5 AC B3 BA C1 C8 CF D6 DD E4 EB F2 F9
05 0C 13 1A 21 28 2F 36 3D 44 4B 52 59 60 67 6E 75 7C 83 8A 91 98 9F A6 AD B4 BB C2 C9 D0 D7 DE E5 EC F3 FA
06 0D 14 1B 22 29 30 37 3E 45 4C 53 5A 61 68 6F 76 7D 84 8B 92 99 A0 A7 AE B5 BC C3 CA D1 D8 DF E6 ED F4 FB
[root@zxxh 00]# cd 00
[root@zxxh 00]# ls
[root@zxxh 00]# pwd
/fastdfs/storage/data/00/00 分区存储，00/00下没有数据
[root@zxxh 00]# ls
[root@zxxh 00]#
```

上传之后再，发现已经有文件。

```
[root@zxxh 00]# systemctl restart firewalld.service
[root@zxxh 00]# ls
wKgDM1z4r5KAFuuGAACwmbcRMT8341.jpg
[root@zxxh 00]#
```

7、FastDFS与Nginx整合 (192.168.3.51)

想要http访问文件服务器，需要整合nginx。

在storage server机器上 (192.168.3.51)

(1) 进入目录，解压fastdfs-nginx-module-1.20.tar.gz

进入安装包目录并解压，解压后进入fastdfs-nginx-module-1.20/src/

```
cd /usr/local/fast
tar -zxvf fastdfs-nginx-module-1.20.tar.gz
cd fastdfs-nginx-module-1.20/src/
```

```
[root@zxxh local]# cd fast/
[root@zxxh fast]# ls
fastdfs-5.11 fastdfs-5.11.tar.gz fastdfs-nginx-module-1.20.tar.gz libfastcommon-1.0.39 libfastcommon-1.0.39.tar.gz nginx-1.1
[root@zxxh fast]# pwd
/usr/local/fast
[root@zxxh fast]# tar -zxvf fastdfs-nginx-module-1.20.tar.gz
fastdfs-nginx-module-1.20/
fastdfs-nginx-module-1.20/HISTORY
fastdfs-nginx-module-1.20/INSTALL
fastdfs-nginx-module-1.20/src/
fastdfs-nginx-module-1.20/src/common.c
fastdfs-nginx-module-1.20/src/common.h
fastdfs-nginx-module-1.20/src/config
fastdfs-nginx-module-1.20/src/mod_fastdfs.conf
fastdfs-nginx-module-1.20/src/nginx_http_fastdfs_module.c
[root@zxxh fast]# ls
fastdfs-5.11 fastdfs-nginx-module-1.20 libfastcommon-1.0.39 nginx-1.16.0.tar.gz
fastdfs-5.11.tar.gz fastdfs-nginx-module-1.20.tar.gz libfastcommon-1.0.39.tar.gz
[root@zxxh fast]# cd fastdfs-nginx-module-1.20/src/
[root@zxxh src]# ls
common.c common.h config mod_fastdfs.conf ngx_http_fastdfs_module.c
[root@zxxh src]#
```

(2) 修改fastdfs-nginx-module-1.20/src/config文件

config文件修改：

/usr/local/include 修改为 /usr/include/fastdfs /usr/include/fastcommon/

```
1 ngx_addon_name=ngx_http_fastdfs_module
2
3 if test -n "${ngx_module_link}"; then
4     ngx_module_type=HTTP
5     ngx_module_name=$ngx_addon_name
6     ngx_module_incs="/usr/include/fastdfs /usr/include/fastcommon/"
7     ngx_module_libs="-lfastcommon -lifdfsclient"
8     ngx_module_srcs="$ngx_addon_dir/nginx_http_fastdfs_module.c"
9     ngx_module_deps=
10    CFLAGS="$CFLAGS -D_FILE_OFFSET_BITS=64 -DFDFS_OUTPUT_CHUNK_SIZE='256*1024' -DFDFS_MOD_CONF_FILENAME='\"/etc/fdfs/mod_fastdfs.conf\"'"
11    . auto/module
12 else
13     HTTP_MODULES="$HTTP_MODULES ngx_http_fastdfs_module"
14     NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/nginx_http_fastdfs_module.c"
15     CORE_INCS="$CORE_INCS /usr/include/fastdfs /usr/include/fastcommon/"
16     CORE_LIBS="$CORE_LIBS -lfastcommon -lifdfsclient"
17     CFLAGS="$CFLAGS -D_FILE_OFFSET_BITS=64 -DFDFS_OUTPUT_CHUNK_SIZE='256*1024' -DFDFS_MOD_CONF_FILENAME='\"/etc/fdfs/mod_fastdfs.conf\"'"
18 fi
```

(3) FastDFS与nginx进行集成

1、进入nginx的安装包存放目录并解压。

```
cd /usr/local/fast/
tar -zxvf nginx-1.16.0.tar.gz
```

2、进入解压目录，并配置加入模块

```
cd nginx-1.16.0
./configure --add-module=/usr/local/fast/fastdfs-nginx-module-1.20/src
```

注意：这里没有设置nginx的安装目录，默认会安装到/usr/local下

3、nginx编译和安装

```
make && make install
```

(4) 复制mod_fastdfs.conf到/etc/fdfs/目录

```
cp /usr/local/fast/fastdfs-nginx-module-1.20/src/mod_fastdfs.conf /etc/fdfs/
```

```
make[1]: 离开目录“/usr/local/fast/nginx-1.16.0”
[root@zxxh nginx-1.16.0]# cd /usr/local/fast/fastdfs-nginx-module-1.20/src
[root@zxxh src]# ls
common.c common.h config mod_fastdfs.conf ngx_http_fastdfs_module.c
[root@zxxh src]# ls -l
总用量 84
-rw-rw-r--. 1 root root 43501 7月 5 2018 common.c
-rw-rw-r--. 1 root root 3995 7月 5 2018 common.h
-rw-rw-r--. 1 root root 902 6月 6 14:43 config
-rw-rw-r--. 1 root root 3725 7月 5 2018 mod_fastdfs.conf
-rw-rw-r--. 1 root root 28668 7月 5 2018 ngx_http_fastdfs_module.c
[root@zxxh src]# cp /usr/local/fast/fastdfs-nginx-module-1.20/src/mod_fastdfs.conf /etc/fdfs/
[root@zxxh src]# ls /etc/fdfs/
client.conf.sample mod_fastdfs.conf storage.conf storage.conf.sample storage_ids.conf.sample tracker.conf.sample
[root@zxxh src]#
```

复制文件

(5) 修改/etc/fdfs/mod_fastdfs.conf

修改内容：

```
connect_timeout=10
tracker_server=192.168.3.50:22122
url_have_group_name = true
store_path0=/fastdfs/storage
```

(6) 复制文件http.conf、mime.types到/etc/fdfs/

```
cp /usr/local/fast/fastdfs-5.11/conf/http.conf /etc/fdfs/
cp /usr/local/fast/fastdfs-5.11/conf/mime.types /etc/fdfs/
```

```
[root@zxxh conf]# pwd
/usr/local/fast/fastdfs-5.11/conf
[root@zxxh conf]# ls -l
总用量 84
-rw-rw-r--. 1 root root 23981 6月 3 2017 anti-steal.jpg
-rw-rw-r--. 1 root root 1461 6月 3 2017 client.conf
-rw-rw-r--. 1 root root 955 6月 3 2017 http.conf
-rw-rw-r--. 1 root root 31172 6月 3 2017 mime.types
-rw-rw-r--. 1 root root 7927 6月 3 2017 storage.conf
-rw-rw-r--. 1 root root 105 6月 3 2017 storage_ids.conf
-rw-rw-r--. 1 root root 7389 6月 3 2017 tracker.conf
[root@zxxh conf]# cp http.conf /etc/fdfs/
[root@zxxh conf]# cp mime.types /etc/fdfs/
[root@zxxh conf]# ls /etc/fdfs/
client.conf.sample http.conf mime.types mod_fastdfs.conf storage.conf storage.conf.sample storage_ids.conf.sample tracker.conf.sample
[root@zxxh conf]#
```

(7) 创建软连接

在/fastdfs/storage文件存储目录下创建软连接，将其连接到实际存放数据的目录

```
ln -s /fastdfs/storage/data/ /fastdfs/storage/data/M00
```

(8) 修改nginx配置

nginx安装在/usr/local下载，找到/usr/local/nginx/conf/nginx.conf文件修改

```
#gzip on;

server {
    listen 8888;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location ~/group([0-9])/M00 {
        ngx_fastdfs_module;
    }

    #error_page 404 /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
```

修改端口

修改匹配规则

注意：nginx里的端口要和FastDFS存储中的storage.conf文件配置一致。http.server_port=8888

启动nginx

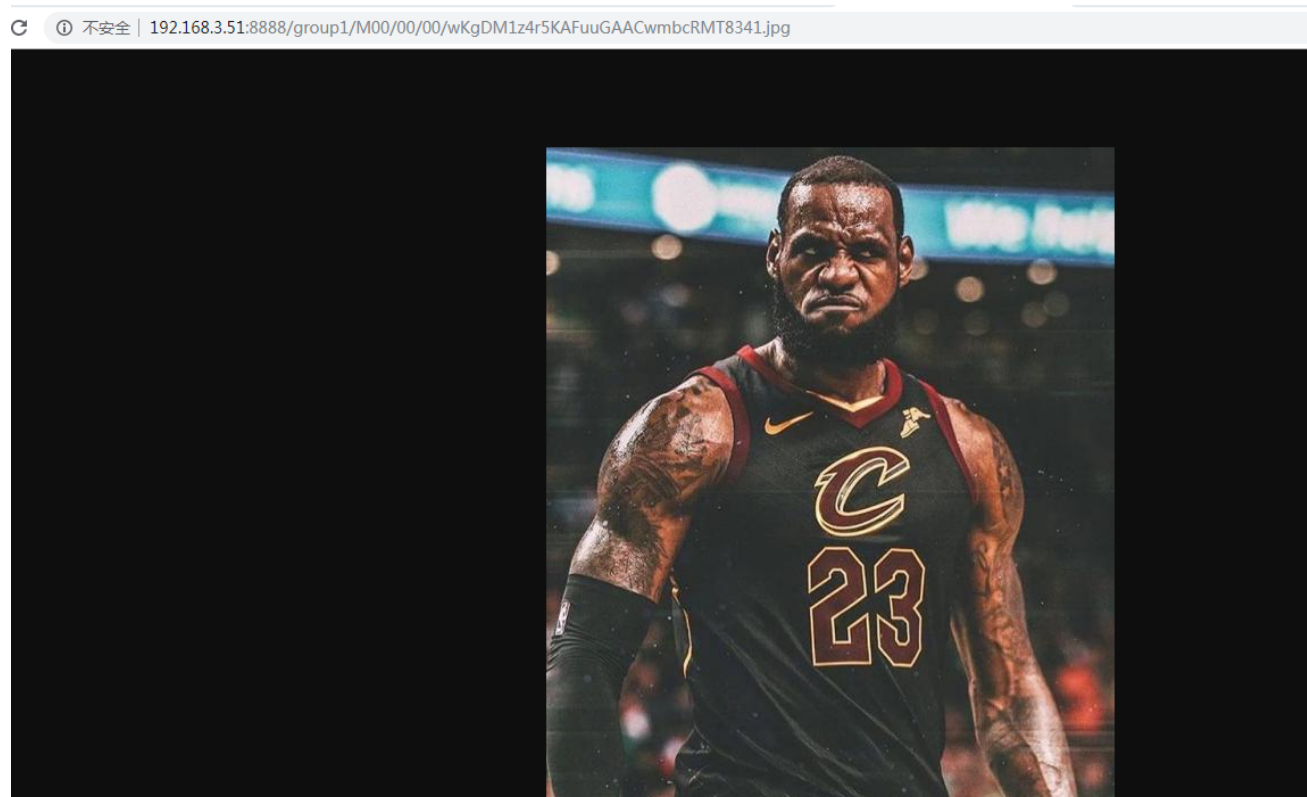
```
./nginx
```

如果<http://192.168.3.51:8888> 返回不到，则需要防火墙开放8888端口

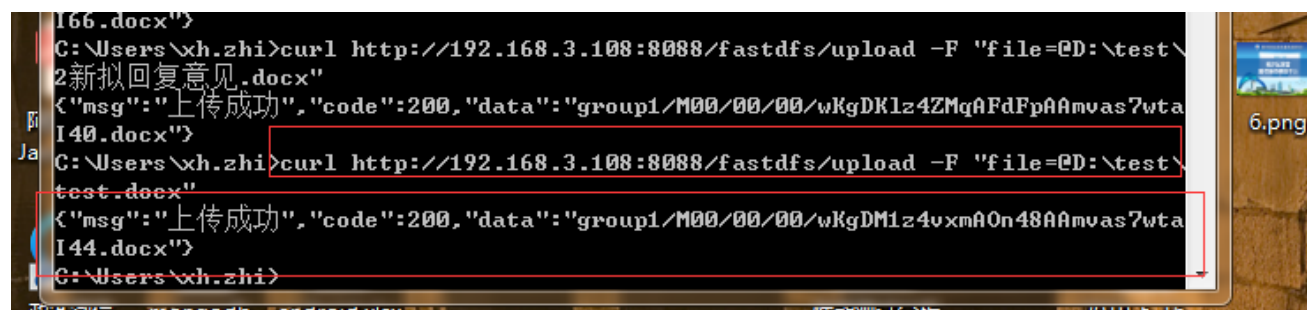
```
firewall-cmd --zone=public --add-port=8888/tcp --permanent
systemctl restart firewalld.service
```

(9) 测试

之前使用命令上传过图片，返回地址 `group1/M00/00/00/wKgDM1z4r5KAFuuGAACwmbcRMT8341.jpg`。我们使用 http 访问文件服务器。可以访问即可



程序上传：



启动和关闭命令

启动命令：

启动tracker命令： `/etc/init.d/fdfs_trackerd start`

查看进程命令： `ps -el | grep fdfs`

启动storage命令： `/etc/init.d/fdfs_storaged start`

查看进程命令： `ps -el | grep fdfs`

启动nginx命令： `/usr/local/nginx/sbin/nginx`

停止命令：

停止tracker命令： `/etc/init.d/fdfs_trackerd stop`

关闭storage命令： `/etc/init.d/fdfs_storaged stop`

关闭nginx命令： `/usr/local/nginx/sbin/nginx -s stop`