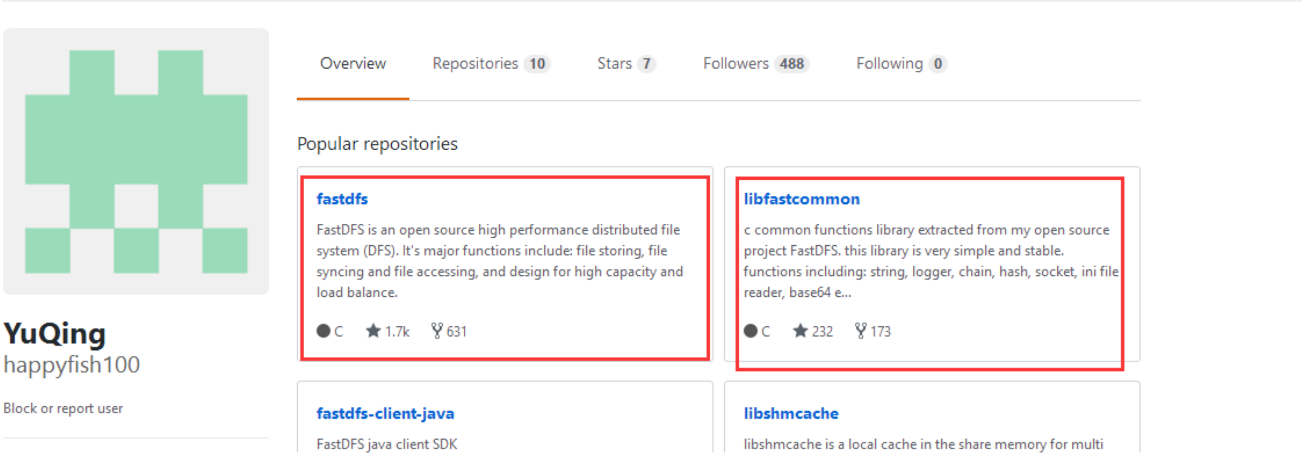
Centos7安装FastDFS

安装步骤：

1、下载FastDFS和libfastcommon

https://github.com/happyfish100/



fastdfs

FastDFS is an open source high performance distributed file system (DFS). It's major functions include: file storing, file syncing and file accessing, and design for high capacity and load balance.

libfastcommon

c common functions library extracted from my open source project FastDFS. this library is very simple and stable. functions including: string, logger, chain, hash, socket, ini file reader, base64 e...

fastdfs-client-java

FastDFS java client SDK

libshmcache

libshmcache is a local cache in the share memory for multi

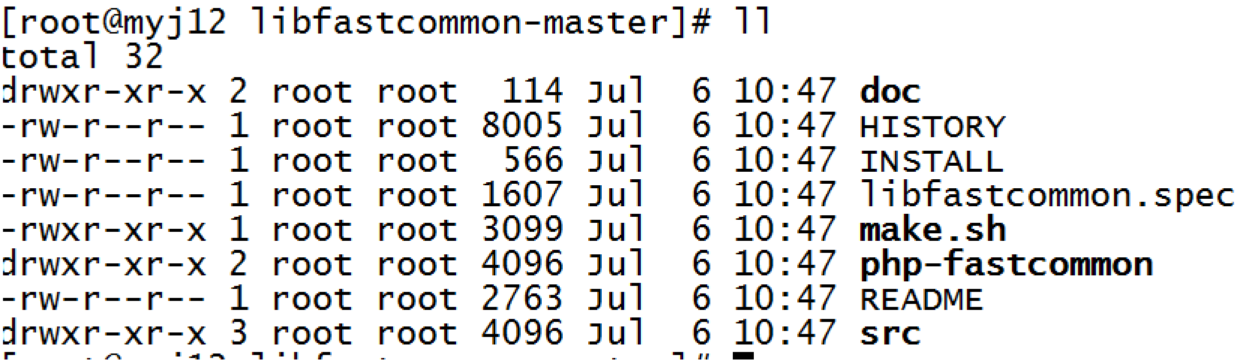
2、安装libfastcommon

将libfastcommon上传到的/home/tar/fastdfs目录下，直接解压：

cd /home/opt/fastdfs

unzip /home/tar/fastdfs/libfastcommon-master.zip

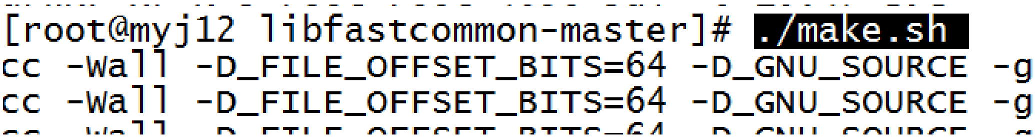
解压成功后进入libfastcommon-master目录后 ll 可以看一下压缩的文件：



```
[root@myj12 libfastcommon-master]# ll
total 32
drwxr-xr-x 2 root root  114 Jul  6 10:47 doc
-rw-r--r-- 1 root root 8005 Jul  6 10:47 HISTORY
-rw-r--r-- 1 root root  566 Jul  6 10:47 INSTALL
-rw-r--r-- 1 root root 1607 Jul  6 10:47 libfastcommon.spec
-rwxr-xr-x 1 root root 3099 Jul  6 10:47 make.sh
drwxr-xr-x 2 root root 4096 Jul  6 10:47 php-fastcommon
-rw-r--r-- 1 root root 2763 Jul  6 10:47 README
drwxr-xr-x 3 root root 4096 Jul  6 10:47 src
```

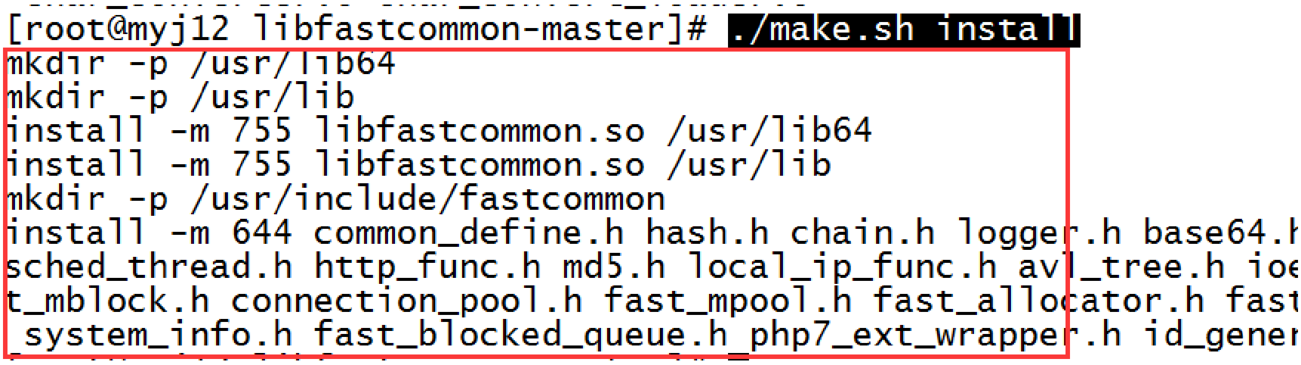
编译安装，分别执行./make.sh和./make.sh install

执行./make.sh进行编译



```
[root@myj12 libfastcommon-master]# ./make.sh
cc -Wall -D_FILE_OFFSET_BITS=64 -D_GNU_SOURCE -g
cc -Wall -D_FILE_OFFSET_BITS=64 -D_GNU_SOURCE -g
```

执行./make.sh install安装，看到类似如下提示信息就说明libfastcommon已安装成功



```
[root@myj12 libfastcommon-master]# ./make.sh install
mkdir -p /usr/lib64
mkdir -p /usr/lib
install -m 755 libfastcommon.so /usr/lib64
install -m 755 libfastcommon.so /usr/lib
mkdir -p /usr/include/fastcommon
install -m 644 common_define.h hash.h chain.h logger.h base64.h
sched_thread.h http_func.h md5.h local_ip_func.h avl_tree.h ioe
t_mblock.h connection_pool.h fast_mpool.h fast_allocator.h fast
 system_info.h fast_blocked_queue.h php7_ext_wrapper.h id_gener
```

至此libfastcommon就已经安装成功了，但注意一下上图中红色框标注的内容，libfastcommon.so 默认安装到了/usr/lib64/libfastcommon.so，但是FastDFS主程序设置的lib目录是/usr/local/lib，所以此处需要重新设置软链接（类似于Windows的快捷方式）：

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

设置完毕后就可以开始安装fastdfs了。

3、安装FastDFS

解压

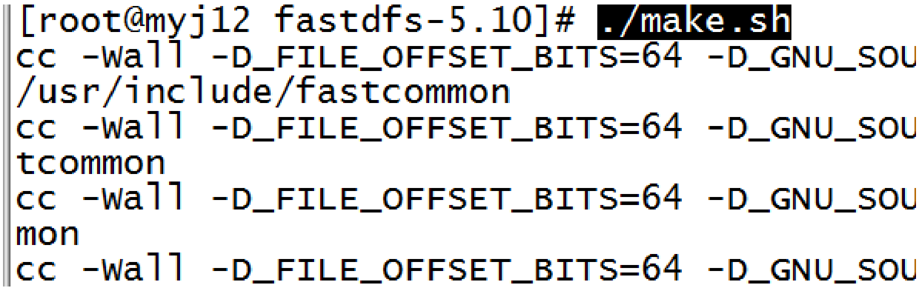
tar -zxf /home/tar/fastdfs/fastdfs-5.10.tar.gz -C /home/opt/fastdfs/

进入解压目录

cd /home/opt/fastdfs/fastdfs-5.10/

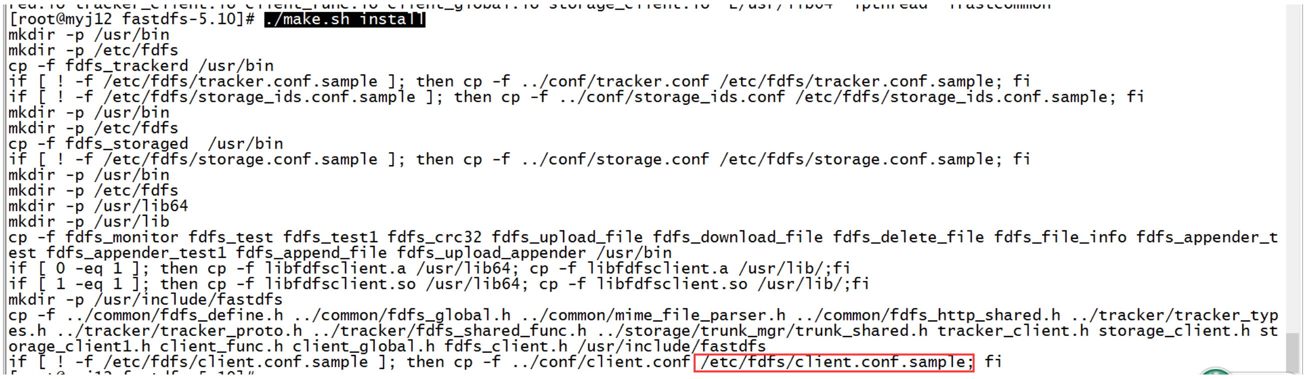
依次执行./make.sh和./make.sh install进行编译和安装：

执行./make.sh编译

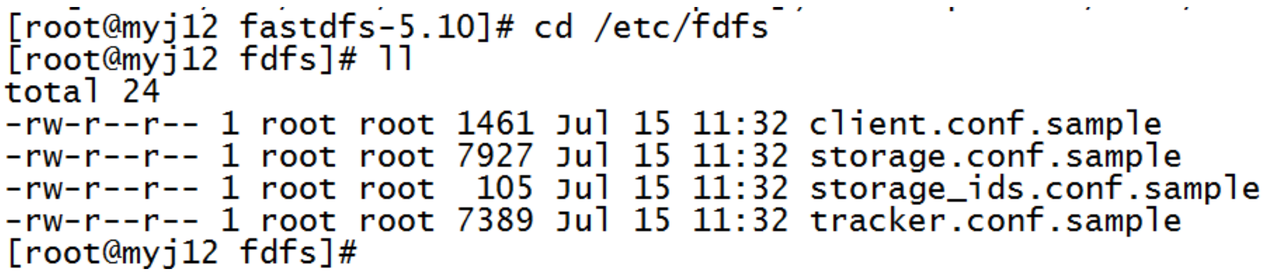


```
[root@myj12 fastdfs-5.10]# ./make.sh
cc -Wall -D_FILE_OFFSET_BITS=64 -D_GNU_SOU
/usr/include/fastcommon
cc -Wall -D_FILE_OFFSET_BITS=64 -D_GNU_SOU
tcommon
cc -Wall -D_FILE_OFFSET_BITS=64 -D_GNU_SOU
mon
cc -Wall -D_FILE_OFFSET_BITS=64 -D_GNU_SOU
```

执行./make.sh install安装，没有报错就说明安装成功了，在log中我们可以发现安装路径：



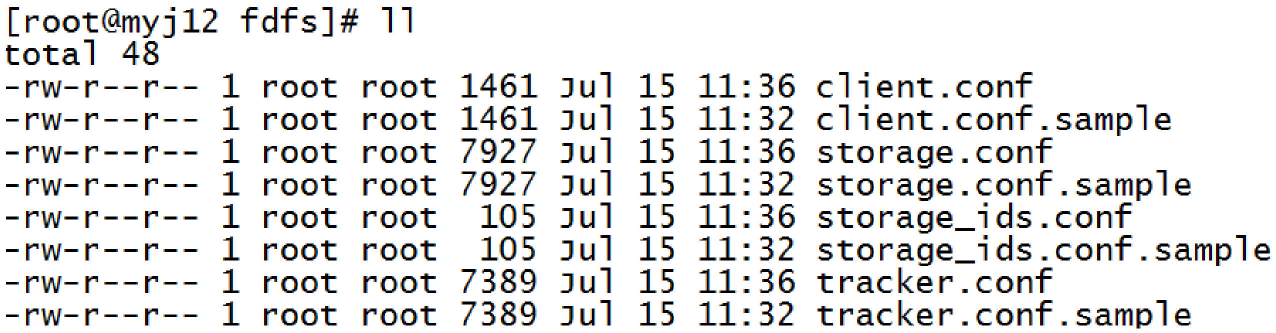
```
[root@myj12 fastdfs-5.10]# ./make.sh install
mkdir -p /usr/bin
mkdir -p /etc/fdfs
cp -f fdfs_trackerd /usr/bin
if [ ! -f /etc/fdfs/tracker.conf.sample ]; then cp -f ../conf/tracker.conf /etc/fdfs/tracker.conf.sample; fi
if [ ! -f /etc/fdfs/storage_ids.conf.sample ]; then cp -f ../conf/storage_ids.conf /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_t
est 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_typ
es.h ../tracker/tracker_proto.h ../tracker/fdfs_shared_func.h ../storage/trunk_mgr/trunk_shared.h tracker_client.h storage_client.h st
orage_client1.h client_func.h client_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@myj12 fastdfs-5.10]# cd /etc/fdfs
[root@myj12 fdfs]# ll
total 24
-rw-r--r-- 1 root root 1461 Jul 15 11:32 client.conf.sample
-rw-r--r-- 1 root root 7927 Jul 15 11:32 storage.conf.sample
-rw-r--r-- 1 root root  105 Jul 15 11:32 storage_ids.conf.sample
-rw-r--r-- 1 root root 7389 Jul 15 11:32 tracker.conf.sample
[root@myj12 fdfs]#
```

如上图，安装成功后就会生成如上的4个.sample文件（示例配置文件），我们再分别拷贝出4个后面用的正式的配置文件：

cp client.conf.sample client.conf
cp storage.conf.sample storage.conf
cp tracker.conf.sample tracker.conf
cp storage_ids.conf.sample storage_ids.conf



```
[root@myj12 fdfs]# ll
total 48
-rw-r--r-- 1 root root 1461 Jul 15 11:36 client.conf
-rw-r--r-- 1 root root 1461 Jul 15 11:32 client.conf.sample
-rw-r--r-- 1 root root 7927 Jul 15 11:36 storage.conf
-rw-r--r-- 1 root root 7927 Jul 15 11:32 storage.conf.sample
-rw-r--r-- 1 root root  105 Jul 15 11:36 storage_ids.conf
-rw-r--r-- 1 root root  105 Jul 15 11:32 storage_ids.conf.sample
-rw-r--r-- 1 root root 7389 Jul 15 11:36 tracker.conf
-rw-r--r-- 1 root root 7389 Jul 15 11:32 tracker.conf.sample
```

至此FastDFS已经安装完毕，接下来的工作就是依次配置Tracker和Storage了。

3、安装Tracker

在配置Tracker之前，首先需要创建Tracker服务器的文件路径，即用于存储Tracker的数据文件和日志文件等，我这里选择在/home/opt/fastdfs/fastdfs-5.10/data目录下创建一个fastdfs_tracker目录用于存放Tracker服务器的相关文件：

mkdir -p /home/opt/fastdfs/fastdfs-5.10/data/fastdfs_tracker

接下来就要重新编辑上一步准备好的/etc/fdfs目录下的tracker.conf配置文件，打开文件后依次做以下修改：

disabled=false #启用配置文件（默认启用）

port=22122 #设置tracker的端口号，通常采用22122这个默认端口

base_path=/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_tracker #设置tracker的数据文件和日志目录

http.server_port=6666 #设置http端口号，默认为8080

配置完成后就可以启动Tracker服务器了，但首先依然要为启动脚本创建软引用，因为fdfs_trackerd等命令在/usr/local/bin中并没有，而是在/usr/bin路径下：

ln -s /usr/bin/fdfs_trackerd /usr/local/bin

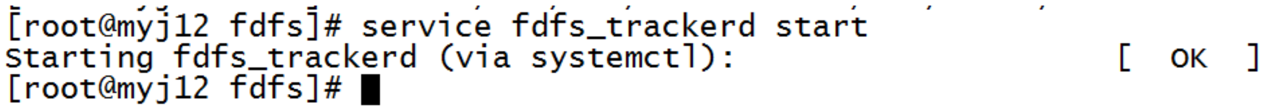
ln -s /usr/bin/stop.sh /usr/local/bin

ln -s /usr/bin/restart.sh /usr/local/bin

最后通过命令启动Tracker服务器：

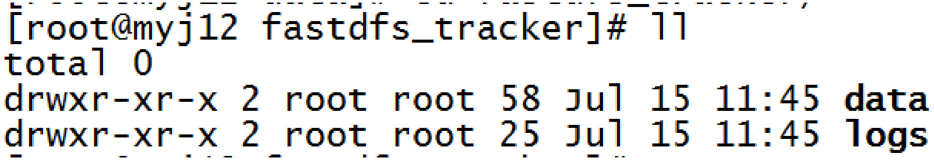
service fdfs_trackerd start

命令执行后可以看到以下提示：



```
[root@myj12 fdfs]# service fdfs_trackerd start
Starting fdfs_trackerd (via systemctl):                    [  OK  ]
[root@myj12 fdfs]# █
```

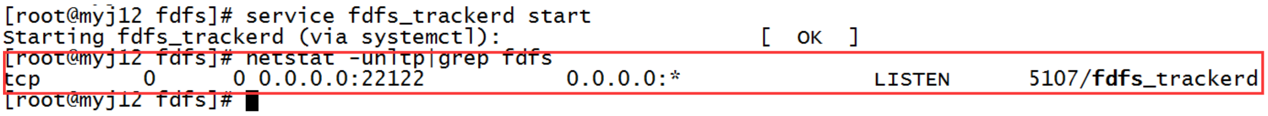
如果启动命令执行成功，那么同时在刚才创建的tracker文件目录/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_tracker中就可以看到启动后新生成的data和logs目录，tracker服务的端口也应当被正常监听，最后再通过netstat命令查看一下端口监听情况：



```
[root@myj12 fastdfs_tracker]# ll
total 0
drwxr-xr-x 2 root root 58 Jul 15 11:45 data
drwxr-xr-x 2 root root 25 Jul 15 11:45 logs
```

netstat -unltp|grep fdfs

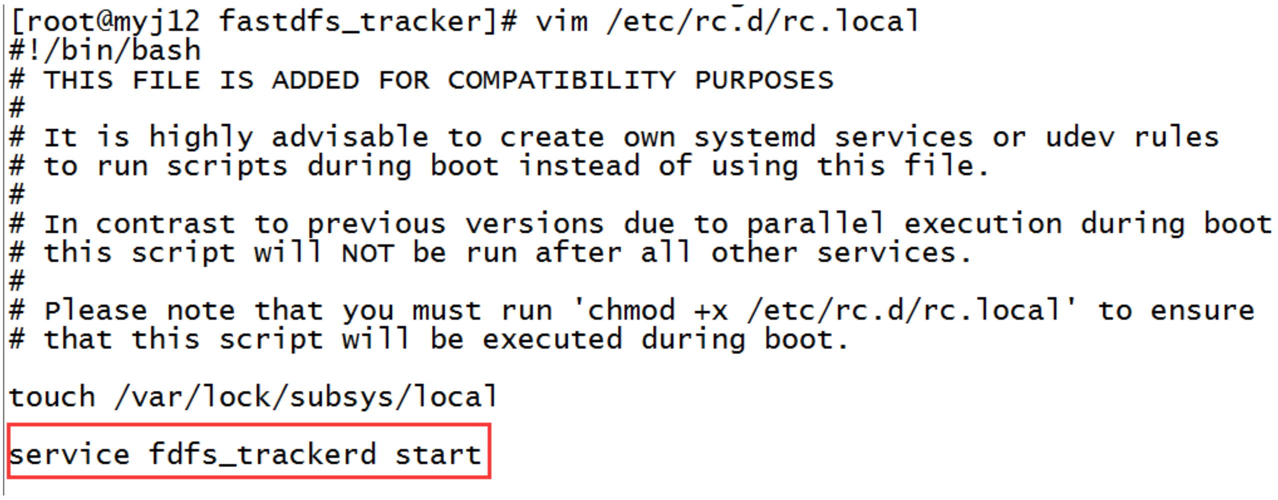
可以看到tracker服务运行的22122端口正常被监听：



```
[root@myj12 fdfs]# service fdfs_trackerd start
Starting fdfs_trackerd (via systemctl):                    [  OK  ]
[root@myj12 fdfs]# netstat -unltp|grep fdfs
tcp        0      0 0.0.0.0:22122           0.0.0.0:*               LISTEN      5107/fdfs_trackerd
[root@myj12 fdfs]# █
```

确认tracker正常启动后可以将tracker设置为开机启动，打开/etc/rc.d/rc.local并在其中加入以下配置：

service fdfs_trackerd start



```
[root@myj12 fastdfs_tracker]# vim /etc/rc.d/rc.local
#!/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

service fdfs_trackerd start
```

Tracker至此就配置好了，接下来就可以配置FastDFS的另一核心——Storage。

4、安装Storage

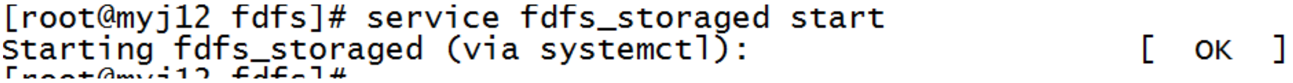
同理，步骤基本与配置Tracker一致，首先是创建Storage服务器的文件目录，需要注意的是同Tracker相比我多建了一个目录，因为Storage还需要一个文件存储路径，用于存放接收的文件：

mkdir /home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage
mkdir /home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage_data
接下来修改/etc/fdfs目录下的storage.conf配置文件，打开文件后依次做以下修改：

disabled=false #启用配置文件（默认启用）
group_name=group1 #组名，根据实际情况修改
port=23000 #设置storage的端口号，默认是23000，同一个组的storage端口号必须一致
base_path=/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage #设置storage数据文件和日志目录
store_path_count=1 #存储路径个数，需要和store_path个数匹配
store_path0=/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage_data #实际文件存储路径
tracker_server=myj12:22122 #tracker 服务器的 IP地址和端口号，如果是单机搭建，IP不要写127.0.0.1，否则启动不成功（此处的ip是我的CentOS虚拟机ip）
http.server_port=8888 #设置 http 端口号
配置完成后同样要为Storage服务器的启动脚本设置软引用：

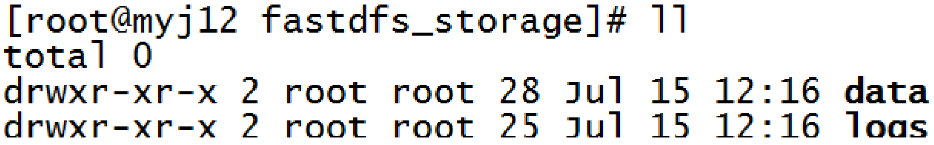
ln -s /usr/bin/fdfs_storaged /usr/local/bin
接下来就可以启动Storage服务了：

service fdfs_storaged start
命令执行后可以看到以下提示：



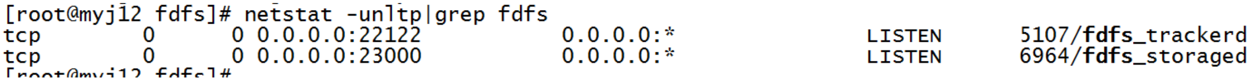
```
[root@myj12 fdfs]# service fdfs_storaged start
Starting fdfs_storaged (via systemctl):                    [  OK  ]
[root@myj12 fdfs]#
```

同理，如果启动成功，/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage中就可以看到启动后新生成的data和logs目录，端口23000也应被正常监听，还有一点就是文件存储路径下会生成多级存储目录，那么接下来看看是否启动成功了：



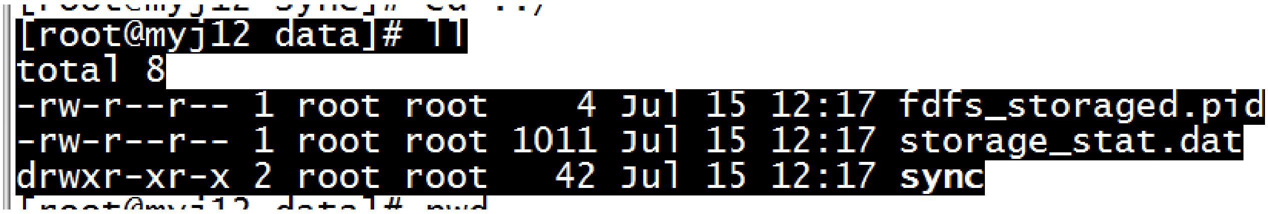
```
[root@myj12 fastdfs_storage]# ll
total 0
drwxr-xr-x 2 root root 28 Jul 15 12:16 data
drwxr-xr-x 2 root root 25 Jul 15 12:16 logs
```

为了确认我们看一下storage的端口情况：



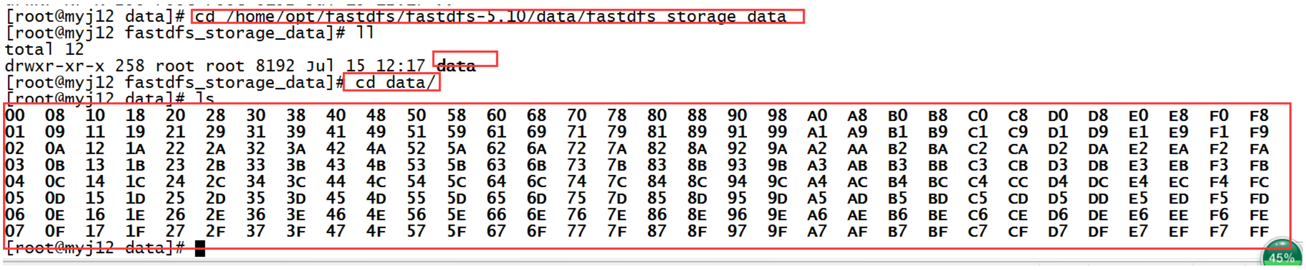
```
[root@myj12 fdfs]# netstat -unltp|grep fdfs
tcp        0      0 0.0.0.0:22122           0.0.0.0:*               LISTEN      5107/fdfs_trackerd
tcp        0      0 0.0.0.0:23000           0.0.0.0:*               LISTEN      6964/fdfs_storaged
[root@myj12 fdfs]#
```

可以看到/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage/data目录下生成好的pid文件和dat文件



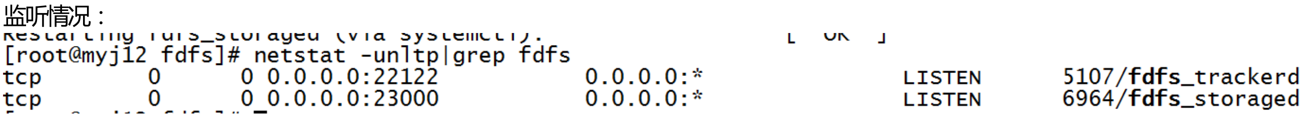
```
[root@myj12 data]# ll
total 8
-rw-r--r-- 1 root root    4 Jul 15 12:17 fdfs_storaged.pid
-rw-r--r-- 1 root root 1011 Jul 15 12:17 storage_stat.dat
drwxr-xr-x 2 root root   42 Jul 15 12:17 sync
```

再看一下实际文件存储路径下是否有创建好的多级目录呢：



```
[root@myj12 data]# cd /home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage_data
[root@myj12 fastdfs_storage_data]# ll
total 12
drwxr-xr-x 258 root root 8192 Jul 15 12:17 data
[root@myj12 fastdfs_storage_data]# cd data/
[root@myj12 data]# ls
00  08  10  18  20  28  30  38  40  48  50  58  60  68  70  78  80  88  90  98  A0  A8  B0  B8  C0  C8  D0  D8  E0  E8  F0  F8
01  09  11  19  21  29  31  39  41  49  51  59  61  69  71  79  81  89  91  99  A1  A9  B1  B9  C1  C9  D1  D9  E1  E9  F1  F9
02  0A  12  1A  22  2A  32  3A  42  4A  52  5A  62  6A  72  7A  82  8A  92  9A  A2  AA  B2  BA  C2  CA  D2  DA  E2  EA  F2  FA
03  0B  13  1B  23  2B  33  3B  43  4B  53  5B  63  6B  73  7B  83  8B  93  9B  A3  AB  B3  BB  C3  CB  D3  DB  E3  EB  F3  FB
04  0C  14  1C  24  2C  34  3C  44  4C  54  5C  64  6C  74  7C  84  8C  94  9C  A4  AC  B4  BC  C4  CC  D4  DC  E4  EC  F4  FC
05  0D  15  1D  25  2D  35  3D  45  4D  55  5D  65  6D  75  7D  85  8D  95  9D  A5  AD  B5  BD  C5  CD  D5  DD  E5  ED  F5  FD
06  0E  16  1E  26  2E  36  3E  46  4E  56  5E  66  6E  76  7E  86  8E  96  9E  A6  AE  B6  BE  C6  CE  D6  DE  E6  EE  F6  FE
07  0F  17  1F  27  2F  37  3F  47  4F  57  5F  67  6F  77  7F  87  8F  97  9F  A7  AF  B7  BF  C7  CF  D7  DF  E7  EF  F7  FF
[root@myj12 data]#
```

如上图，没有任何问题，data下有256个1级目录，每级目录下又有256个2级子目录，总共65536个文件，新写的文件会以hash的方式被路由到其中某个子目录下，然后将文件数据直接作为一个本地文件存储到该目录中。那么最后我们再看一下storage服务的端口监听情况：



```
[root@myj12 fdfs]# netstat -unltp|grep fdfs
tcp        0      0 0.0.0.0:22122           0.0.0.0:*               LISTEN      5107/fdfs_trackerd
tcp        0      0 0.0.0.0:23000           0.0.0.0:*               LISTEN      6964/fdfs_storaged
```

如上图，可以看到此时已经正常监听tracker的22122端口和storage的23000端口，至此storage服务器就已经配置完成，确定了storage服务器启动成功后，还有一项工作就是看看storage服务器是否已经登记到 tracker服务器（也可以理解为tracker与storage是否整合成功），运行以下命令：

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

[root@myj12 fdfs]# /usr/bin/fdfs_monitor /etc/fdfs/storage.conf

[2017-07-15 12:21:19] DEBUG - base_path=/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage, connect_timeout=30, network_timeout=60, tracker_server_count=1, anti_steal_token=0, anti_steal_secret_key length=0, use_connection_pool=0, g_connection_pool_max_idle_time=3600s, use_storage_id=0, storage server id count: 0

server_count=1, server_index=0

tracker server is 192.168.5.20:22122

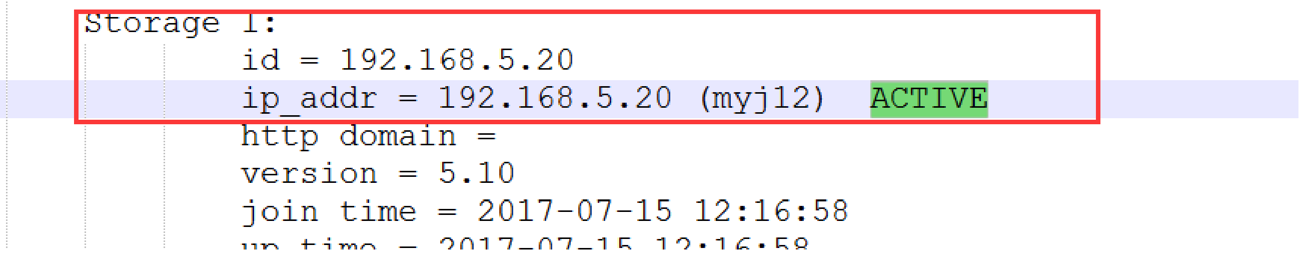
group count: 1

Group 1:
group name = group1
disk total space = 18121 MB
disk free space = 10812 MB
trunk free space = 0 MB
storage server count = 1
active server count = 1
storage server port = 23000
storage HTTP port = 8888
store path count = 1
subdir count per path = 256
current write server index = 0
current trunk file id = 0

Storage 1:
id = 192.168.5.20
ip_addr = 192.168.5.20 (myj12) ACTIVE
http domain =
version = 5.10
join time = 2017-07-15 12:16:58

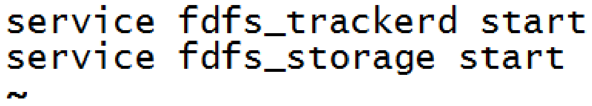
```
up time = 2017-07-15 12:16:58
total storage = 18121 MB
free storage = 10812 MB
upload priority = 10
store_path_count = 1
subdir_count_per_path = 256
storage_port = 23000
storage_http_port = 8888
current_write_path = 0
source storage id =
if_trunk_server = 0
connection.alloc_count = 256
connection.current_count = 0
connection.max_count = 0
total_upload_count = 0
success_upload_count = 0
total_append_count = 0
success_append_count = 0
total_modify_count = 0
success_modify_count = 0
total_truncate_count = 0
success_truncate_count = 0
total_set_meta_count = 0
success_set_meta_count = 0
total_delete_count = 0
success_delete_count = 0
total_download_count = 0
success_download_count = 0
total_get_meta_count = 0
success_get_meta_count = 0
total_create_link_count = 0
success_create_link_count = 0
total_delete_link_count = 0
success_delete_link_count = 0
total_upload_bytes = 0
success_upload_bytes = 0
total_append_bytes = 0
success_append_bytes = 0
total_modify_bytes = 0
success_modify_bytes = 0
stotal_download_bytes = 0
success_download_bytes = 0
total_sync_in_bytes = 0
success_sync_in_bytes = 0
total_sync_out_bytes = 0
success_sync_out_bytes = 0
total_file_open_count = 0
success_file_open_count = 0
total_file_read_count = 0
success_file_read_count = 0
total_file_write_count = 0
success_file_write_count = 0
last_heart_beat_time = 2017-07-15 12:21:04
last_source_update = 1970-01-01 08:00:00
last_sync_update = 1970-01-01 08:00:00
last_synced_timestamp = 1970-01-01 08:00:00
```

如上所示，看到192.168.5.20 ACTIVE 字样即可说明storage服务器已经成功登记到了tracker服务器，同理别忘了添加开机启动



```
Storage 1:
        id = 192.168.5.20
        ip_addr = 192.168.5.20 (myj12)  ACTIVE
        http domain =
        version = 5.10
        join time = 2017-07-15 12:16:58
        up time = 2017-07-15 12:16:58
```

打开/etc/rc.d/rc.local并将如下配置追加到文件中：
service fdfs_storage start



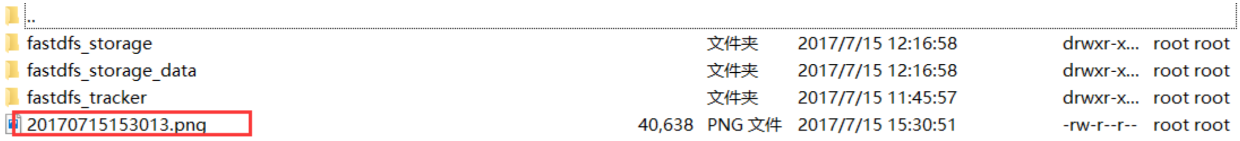
```
service fdfs_trackerd start
service fdfs_storage start
~
```

至此我们就已经完成了fastdfs的全部配置，此时也就可以用客户端工具进行文件上传下载的测试了。

5、初步测试

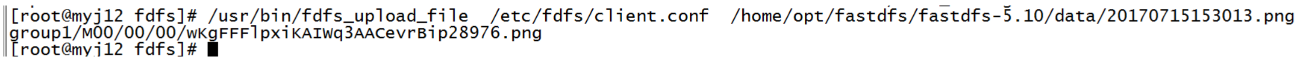
测试时需要设置客户端的配置文件，编辑/etc/fdfs目录下的client.conf 文件，打开文件后依次做以下修改：

base_path=/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_tracker #tracker服务器文件路径
tracker_server=myj12:22122 #tracker服务器IP地址和端口号
http.tracker_server_port=6666 # tracker 服务器的 http 端口号，必须和tracker的设置对应起来
配置完成后就可以模拟文件上传了，先给/home/opt/fastdfs/fastdfs-5.10/data目录下放一张图片：



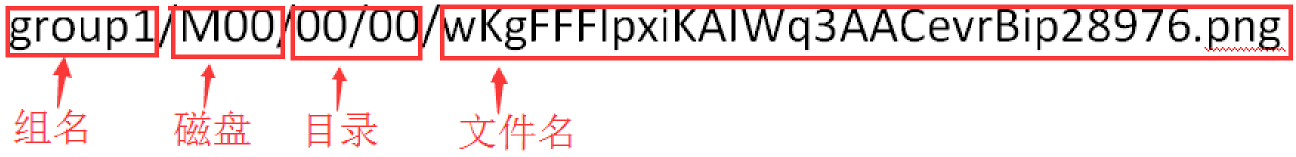
..					
fastdfs_storage		文件夹	2017/7/15 12:16:58	drwxr-x...	root root
fastdfs_storage_data		文件夹	2017/7/15 12:16:58	drwxr-x...	root root
fastdfs_tracker		文件夹	2017/7/15 11:45:57	drwxr-x...	root root
20170715153013.png	40,638	PNG 文件	2017/7/15 15:30:51	-rw-r--r--	root root

然后通过执行客户端上传命令尝试上传：
/usr/bin/fdfs_upload_file /etc/fdfs/client.conf /home/opt/fastdfs/fastdfs-5.10/data/20170715153013.png
运行后可以发现返回了一个路径：
group1/M00/00/00/wKgFFFlpxiKAIWq3AACevrBip28976.png

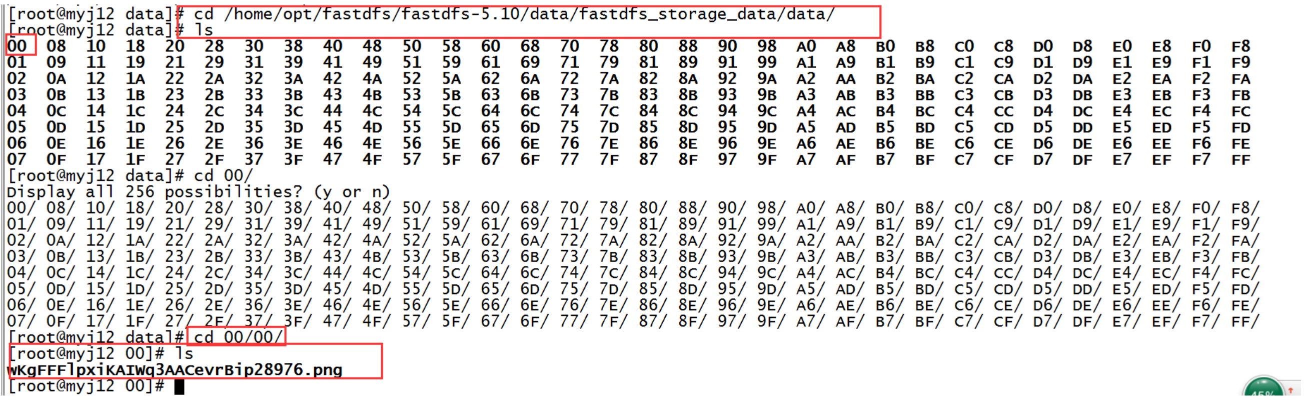


```
[root@myj12 fdfs]# /usr/bin/fdfs_upload_file  /etc/fdfs/client.conf  /home/opt/fastdfs/fastdfs-5.10/data/20170715153013.png
group1/M00/00/00/wKgFFFlpxiKAIWq3AACevrBip28976.png
[root@myj12 fdfs]# █
```

这就表示我们的文件已经上传成功了，当文件存储到某个子目录后，即认为该文件存储成功，接下来会为该文件生成一个文件名，文件名由group、存储目录、两级子目录、fileid、文件后缀名（由客户端指定，主要用于区分文件类型）拼接而成，如下图：



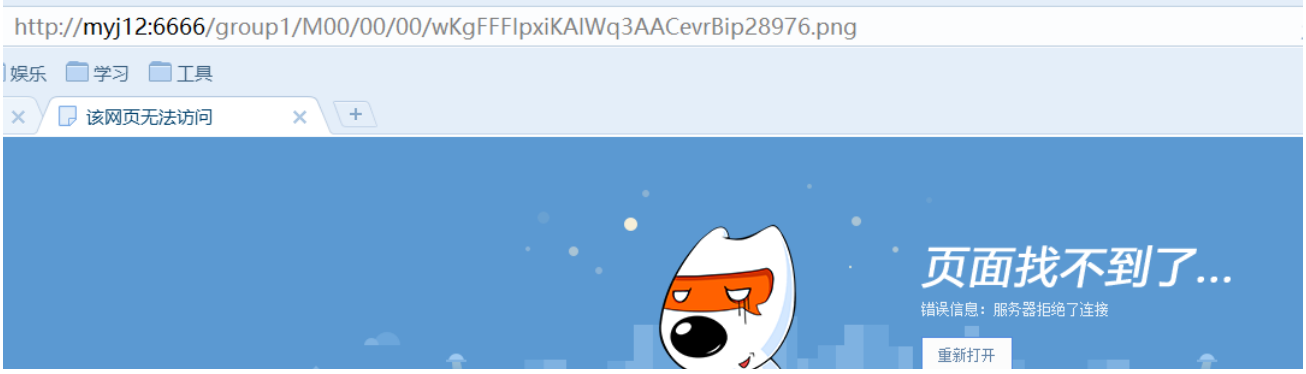
同时在之前配置的storage服务器的实际文件存储路径中也可以根据返回的路径找到实际文件：



```
[root@myj12 data]# cd /home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage_data/data/
[root@myj12 data]# ls
00  08  10  18  20  28  30  38  40  48  50  58  60  68  70  78  80  88  90  98  A0  A8  B0  B8  C0  C8  D0  D8  E0  E8  F0  F8
01  09  11  19  21  29  31  39  41  49  51  59  61  69  71  79  81  89  91  99  A1  A9  B1  B9  C1  C9  D1  D9  E1  E9  F1  F9
02  0A  12  1A  22  2A  32  3A  42  4A  52  5A  62  6A  72  7A  82  8A  92  9A  A2  AA  B2  BA  C2  CA  D2  DA  E2  EA  F2  FA
03  0B  13  1B  23  2B  33  3B  43  4B  53  5B  63  6B  73  7B  83  8B  93  9B  A3  AB  B3  BB  C3  CB  D3  DB  E3  EB  F3  FB
04  0C  14  1C  24  2C  34  3C  44  4C  54  5C  64  6C  74  7C  84  8C  94  9C  A4  AC  B4  BC  C4  CC  D4  DC  E4  EC  F4  FC
05  0D  15  1D  25  2D  35  3D  45  4D  55  5D  65  6D  75  7D  85  8D  95  9D  A5  AD  B5  BD  C5  CD  D5  DD  E5  ED  F5  FD
06  0E  16  1E  26  2E  36  3E  46  4E  56  5E  66  6E  76  7E  86  8E  96  9E  A6  AE  B6  BE  C6  CE  D6  DE  E6  EE  F6  FE
07  0F  17  1F  27  2F  37  3F  47  4F  57  5F  67  6F  77  7F  87  8F  97  9F  A7  AF  B7  BF  C7  CF  D7  DF  E7  EF  F7  FF
[root@myj12 data]# cd 00/
Display all 256 possibilities? (y or n)
00/ 08/ 10/ 18/ 20/ 28/ 30/ 38/ 40/ 48/ 50/ 58/ 60/ 68/ 70/ 78/ 80/ 88/ 90/ 98/ A0/ A8/ B0/ B8/ C0/ C8/ D0/ D8/ E0/ E8/ F0/ F8/
01/ 09/ 11/ 19/ 21/ 29/ 31/ 39/ 41/ 49/ 51/ 59/ 61/ 69/ 71/ 79/ 81/ 89/ 91/ 99/ A1/ A9/ B1/ B9/ C1/ C9/ D1/ D9/ E1/ E9/ F1/ F9/
02/ 0A/ 12/ 1A/ 22/ 2A/ 32/ 3A/ 42/ 4A/ 52/ 5A/ 62/ 6A/ 72/ 7A/ 82/ 8A/ 92/ 9A/ A2/ AA/ B2/ BA/ C2/ CA/ D2/ DA/ E2/ EA/ F2/ FA/
03/ 0B/ 13/ 1B/ 23/ 2B/ 33/ 3B/ 43/ 4B/ 53/ 5B/ 63/ 6B/ 73/ 7B/ 83/ 8B/ 93/ 9B/ A3/ AB/ B3/ BB/ C3/ CB/ D3/ DB/ E3/ EB/ F3/ FB/
04/ 0C/ 14/ 1C/ 24/ 2C/ 34/ 3C/ 44/ 4C/ 54/ 5C/ 64/ 6C/ 74/ 7C/ 84/ 8C/ 94/ 9C/ A4/ AC/ B4/ BC/ C4/ CC/ D4/ DC/ E4/ EC/ F4/ FC/
05/ 0D/ 15/ 1D/ 25/ 2D/ 35/ 3D/ 45/ 4D/ 55/ 5D/ 65/ 6D/ 75/ 7D/ 85/ 8D/ 95/ 9D/ A5/ AD/ B5/ BD/ C5/ CD/ D5/ DD/ E5/ ED/ F5/ FD/
06/ 0E/ 16/ 1E/ 26/ 2E/ 36/ 3E/ 46/ 4E/ 56/ 5E/ 66/ 6E/ 76/ 7E/ 86/ 8E/ 96/ 9E/ A6/ AE/ B6/ BE/ C6/ CE/ D6/ DE/ E6/ EE/ F6/ FE/
07/ 0F/ 17/ 1F/ 27/ 2F/ 37/ 3F/ 47/ 4F/ 57/ 5F/ 67/ 6F/ 77/ 7F/ 87/ 8F/ 97/ 9F/ A7/ AF/ B7/ BF/ C7/ CF/ D7/ DF/ E7/ EF/ F7/ FF/
[root@myj12 data]# cd 00/00/
[root@myj12 00]# ls
wKgFFFlpxiKAIWq3AACevrBip28976.png
[root@myj12 00]# █
```

接下来尝试用浏览器发送HTTP请求访问一下文件：

http://myj12:6666/group1/M00/00/00/wKgFFFlpxiKAIWq3AACevrBip28976.png



http://myj12:6666/group1/M00/00/00/wKgFFFlpxiKAIWq3AACevrBip28976.png

页面找不到了...

错误信息：服务器拒绝了连接

此时发现并不能访问，因为FastDFS目前已不支持http协议，所以提供了nginx上使用FastDFS的模块fastdfs-nginx-module，注意版本对应（否自编译ngnix会报错），FastDFS和astdfs-nginx-module版本对应如下：

Version 5.11对应的fastdfs-nginx-module的Version 1.20

Version 5.10对应的fastdfs-nginx-module的Version 1.19

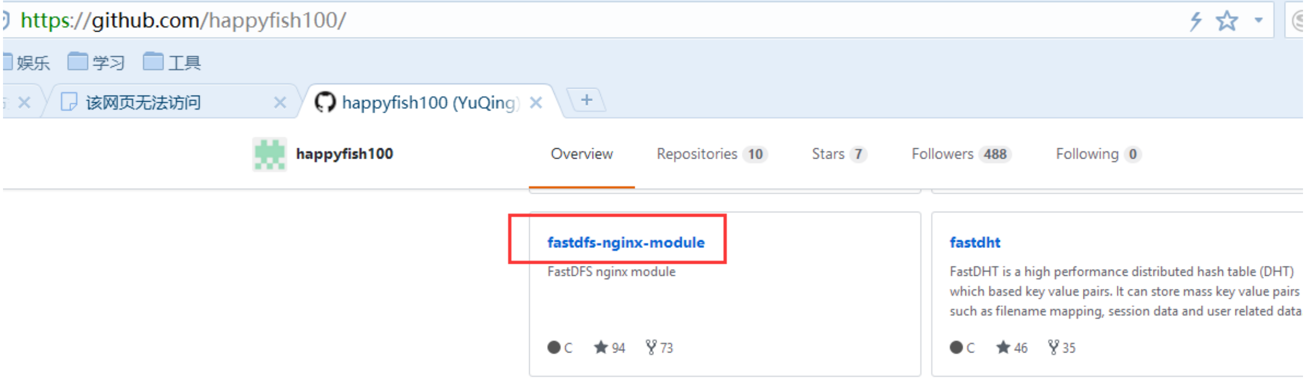
下载地址如下：

fastdfs-nginx-module Version 1.20 https://github.com/happyfish100/fastdfs-nginx-module

fastdfs-nginx-module Version 1.19 http://download.csdn.net/download/apporoad/9865242

（因为我安装的FastDFS是fastdfs-5.10所以下载的是fastdfs-nginx-module Version 1.19）

不支持http协议这样做最大的好处就是提供了HTTP服务并且解决了group中storage服务器的同步延迟问题，接下来就具体记录一下fastdfs-nginx-module的安装配置过程。



https://github.com/happyfish100/

fastdfs-nginx-module

FastDFS nginx module

fastdht

FastDHT is a high performance distributed hash table (DHT) which based key value pairs. It can store mass key value pairs such as filename mapping, session data and user related data

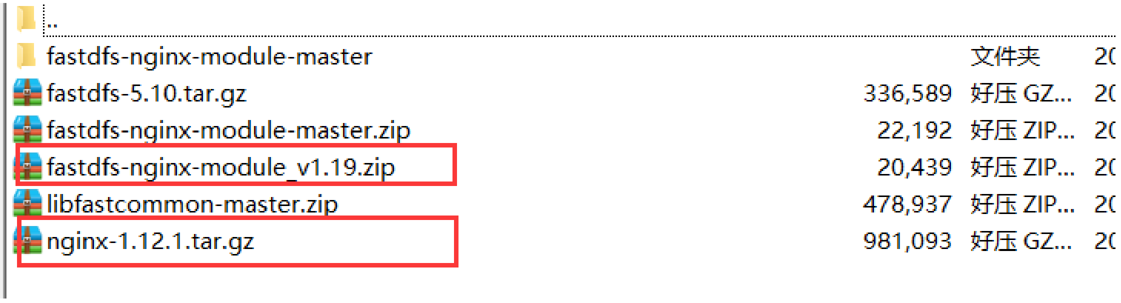
6、安装fastdfs-nginx-module

在GitHub上下载fastdfs-nginx-module后上传到CentOS中开始安装，在安装nginx之前需要先安装一些模块依赖的lib库：

yum -y install pcre pcre-devel
yum -y install zlib zlib-devel
yum -y install openssl openssl-devel
yum install -y gcc gcc-c++ autoconf automake bzip2-devel ncurses-devel

依次装好依赖之后就可以开始安装nginx。

首先是为storage服务器安装nginx，首先将nginx和fastdfs-nginx-module的安装包上传至CentOS：



..			
fastdfs-nginx-module-master		文件夹	20
fastdfs-5.10.tar.gz	336,589	好压 GZ...	20
fastdfs-nginx-module-master.zip	22,192	好压 ZIP...	20
fastdfs-nginx-module_v1.19.zip	20,439	好压 ZIP...	20
libfastcommon-master.zip	478,937	好压 ZIP...	20
nginx-1.12.1.tar.gz	981,093	好压 GZ...	20

首先分别进行解压：

mkdir /home/opt/nginx

cd /home/tar/fastdfs

tar -zxf nginx-1.12.1.tar.gz -C /home/opt/nginx

//unzip fastdfs-nginx-module-master.zip(注意版本，否则会报错： ‘FDFSHTTPParams’ has no member named ‘support_multi_range’ ，我第一次安装因为下载了最新的fastdfs-nginx-module-master.导致包改错误，后改成fastdfs-nginx-module_v1.19再次编译才通过)

unzip fastdfs-nginx-module_v1.19.zip

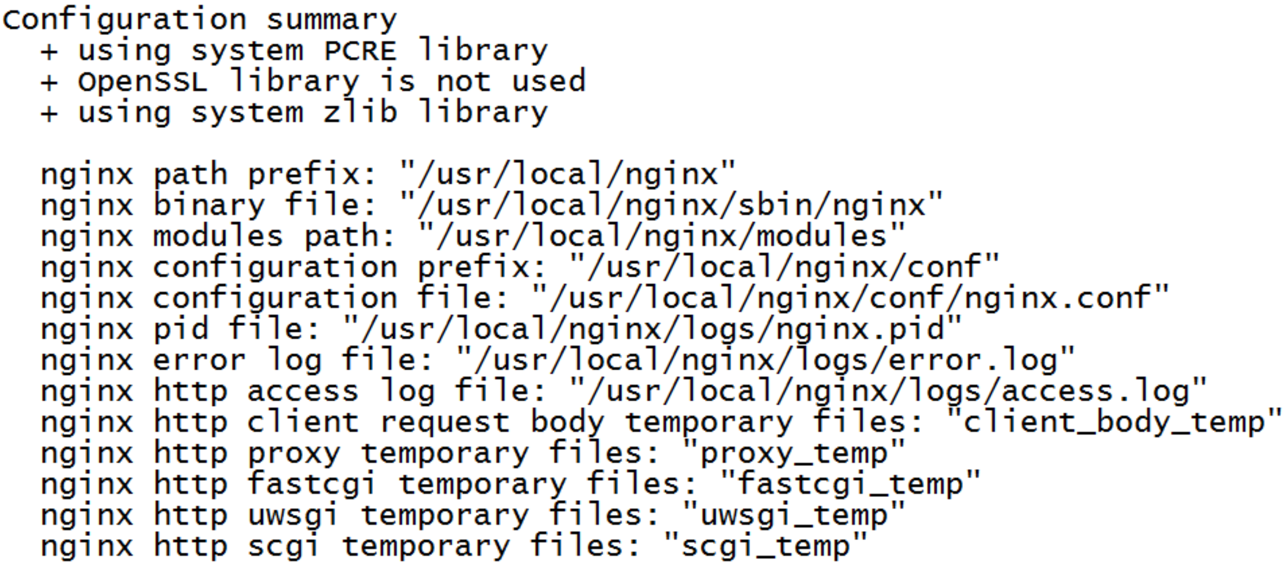
cp -r fastdfs-nginx-module-master /home/opt/fastdfs/

解压成功后就可以编译安装nginx了，进入nginx目录并输入以下命令进行配置：

cd /home/opt/nginx/nginx-1.12.1/

./configure --prefix=/usr/local/nginx --add-module=/home/opt/fastdfs/fastdfs-nginx-module-master/src

配置成功后会看到如下信息：

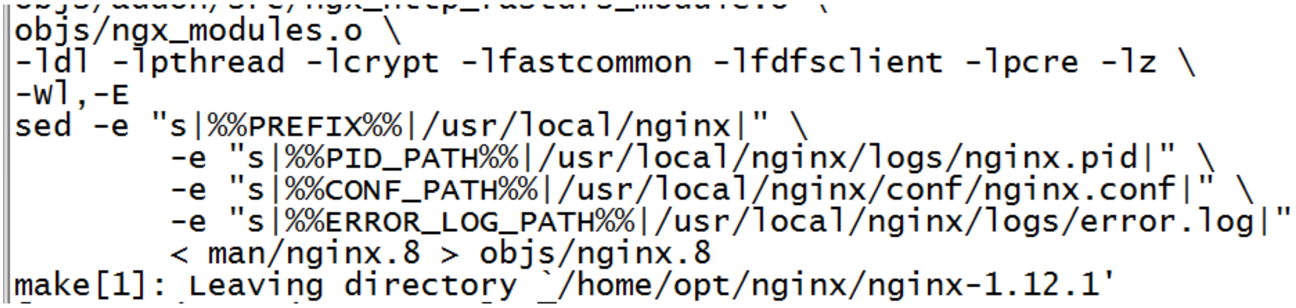


```
Configuration summary
  + using system PCRE library
  + OpenSSL library is not used
  + using system zlib library

  nginx path prefix: "/usr/local/nginx"
  nginx binary file: "/usr/local/nginx/sbin/nginx"
  nginx modules path: "/usr/local/nginx/modules"
  nginx configuration prefix: "/usr/local/nginx/conf"
  nginx configuration file: "/usr/local/nginx/conf/nginx.conf"
  nginx pid file: "/usr/local/nginx/logs/nginx.pid"
  nginx error log file: "/usr/local/nginx/logs/error.log"
  nginx http access log file: "/usr/local/nginx/logs/access.log"
  nginx http client request body temporary files: "client_body_temp"
  nginx http proxy temporary files: "proxy_temp"
  nginx http fastcgi temporary files: "fastcgi_temp"
  nginx http uwsgi temporary files: "uwsgi_temp"
  nginx http scgi temporary files: "scgi_temp"
```

紧接着就可以进行编译安装了，依次执行以下命令：

执行make进行编译

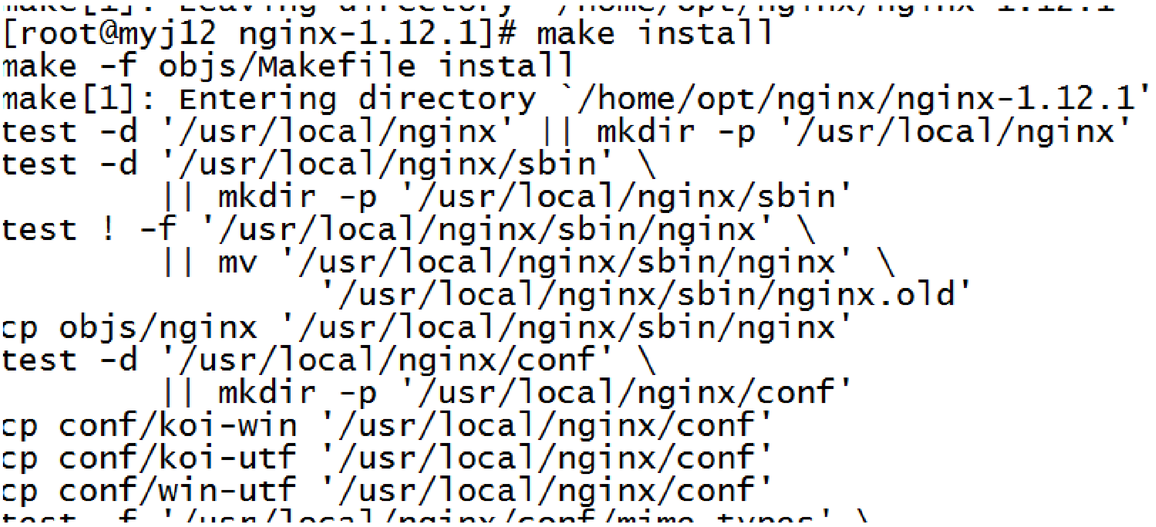
make



```
objs/addon/src/ngx_http_fastdfs_module.o \
objs/ngx_modules.o \
-ldl -lpthread -lcrypt -lfastcommon -lfdfsclient -lpcre -lz \
-Wl,-E
sed -e "s|%%PREFIX%%|/usr/local/nginx|" \
        -e "s|%%PID_PATH%%|/usr/local/nginx/logs/nginx.pid|" \
        -e "s|%%CONF_PATH%%|/usr/local/nginx/conf/nginx.conf|" \
        -e "s|%%ERROR_LOG_PATH%%|/usr/local/nginx/logs/error.log|" \
        < man/nginx.8 > objs/nginx.8
make[1]: Leaving directory `/home/opt/nginx/nginx-1.12.1'
```

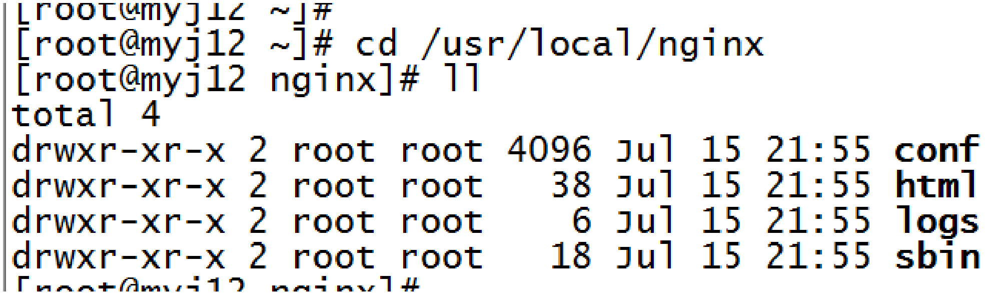
安装

make install



```
make[1]: Leaving directory `/home/opt/nginx/nginx-1.12.1'
[root@myj12 nginx-1.12.1]# make install
make -f objs/Makefile install
make[1]: Entering directory `/home/opt/nginx/nginx-1.12.1'
test -d '/usr/local/nginx' || mkdir -p '/usr/local/nginx'
test -d '/usr/local/nginx/sbin' \
        || mkdir -p '/usr/local/nginx/sbin'
test ! -f '/usr/local/nginx/sbin/nginx' \
        || mv '/usr/local/nginx/sbin/nginx' \
                '/usr/local/nginx/sbin/nginx.old'
cp objs/nginx '/usr/local/nginx/sbin/nginx'
test -d '/usr/local/nginx/conf' \
        || mkdir -p '/usr/local/nginx/conf'
cp conf/koi-win '/usr/local/nginx/conf'
cp conf/koi-utf '/usr/local/nginx/conf'
cp conf/win-utf '/usr/local/nginx/conf'
test -f '/usr/local/nginx/conf/mime.types' \
```

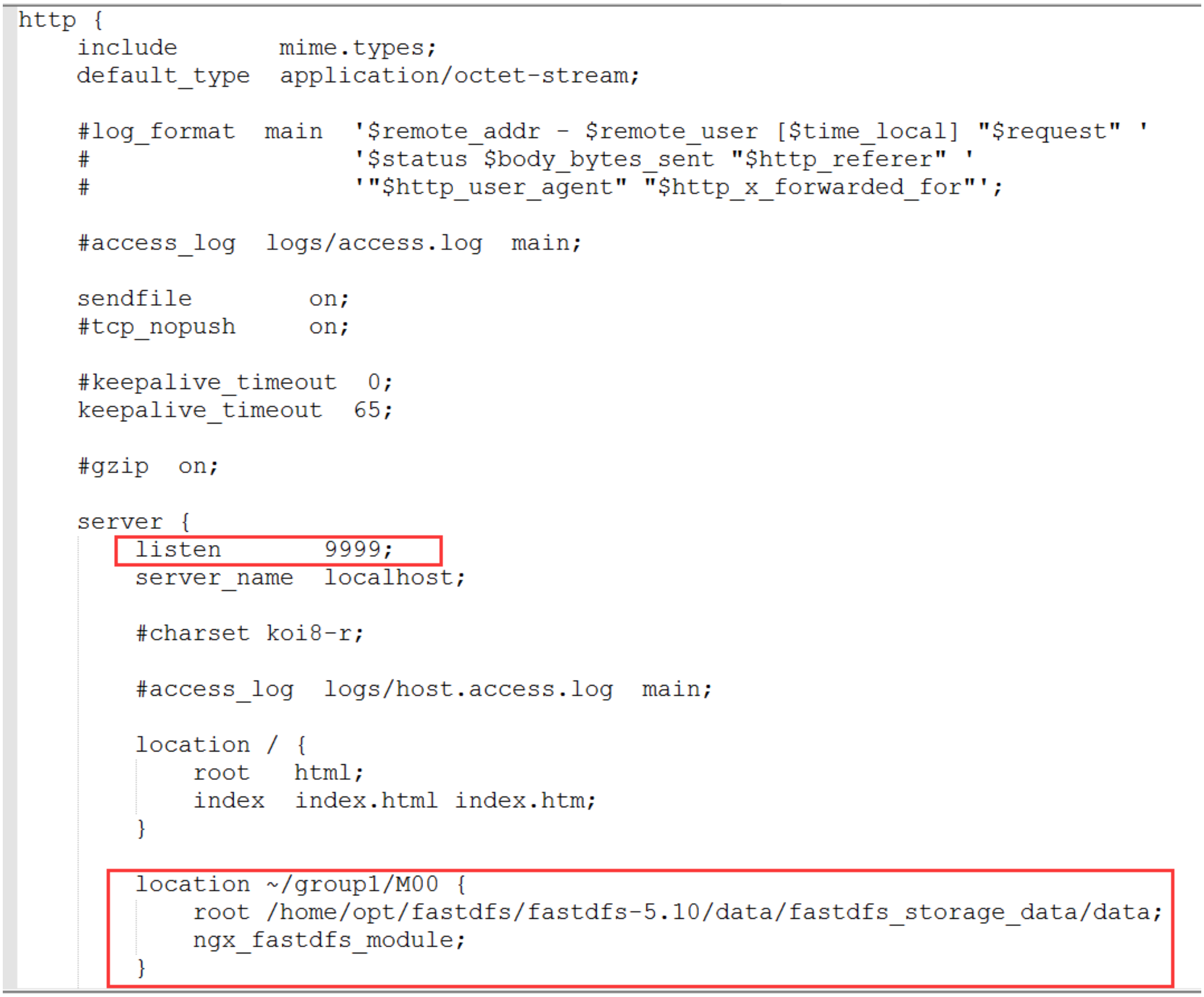
安装完成后，我们在我们指定的目录/usr/local/nginx中就可以看到nginx的安装目录了：



```
[root@myj12 ~]#
[root@myj12 ~]# cd /usr/local/nginx
[root@myj12 nginx]# ll
total 4
drwxr-xr-x 2 root root 4096 Jul 15 21:55 conf
drwxr-xr-x 2 root root   38 Jul 15 21:55 html
drwxr-xr-x 2 root root    6 Jul 15 21:55 logs
drwxr-xr-x 2 root root   18 Jul 15 21:55 sbin
[root@myj12 nginx]#
```

接下来要修改一下nginx的配置文件，进入conf目录并打开nginx.conf文件加入以下配置：

```
listen  9999;

location ~/group1/M00 {
 root /home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage_data/data;
 ngx_fastdfs_module;
}
```



```
http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;

    server {
        listen       9999;
        server_name  localhost;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

        location / {
            root   html;
            index  index.html index.htm;
        }

        location ~/group1/M00 {
            root /home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage_data/data;
            ngx_fastdfs_module;
        }
```

然后进入FastDFS的安装目录/home/opt/fastdfs/fastdfs-5.10目录下的conf目录，将http.conf和mime.types拷贝到/etc/fdfs目录下：

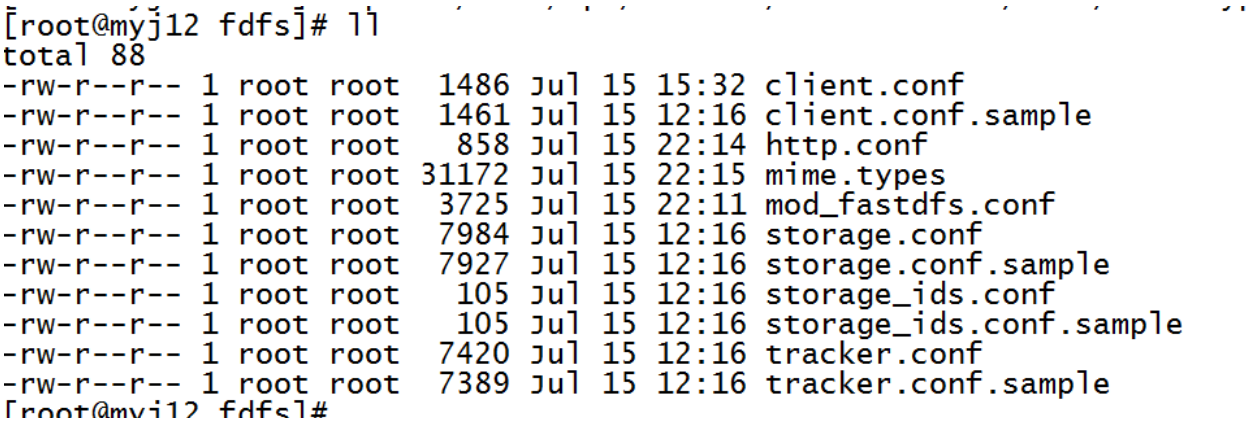
cp -r /home/opt/fastdfs/fastdfs-5.10/conf/http.conf /etc/fdfs/

cp -r /home/opt/fastdfs/fastdfs-5.10/conf/mime.types /etc/fdfs/

接下来还需要把fastdfs-nginx-module安装目录中src目录下的mod_fastdfs.conf也拷贝到/etc/fdfs目录下：

cp -r /home/opt/fastdfs/fastdfs-nginx-module-master/src/mod_fastdfs.conf /etc/fdfs/

看一下/etc/fdfs目录下当前所有的配置文件：



```
[root@myj12 fdfs]# ll
total 88
-rw-r--r-- 1 root root  1486 Jul 15 15:32 client.conf
-rw-r--r-- 1 root root  1461 Jul 15 12:16 client.conf.sample
-rw-r--r-- 1 root root   858 Jul 15 22:14 http.conf
-rw-r--r-- 1 root root 31172 Jul 15 22:15 mime.types
-rw-r--r-- 1 root root  3725 Jul 15 22:11 mod_fastdfs.conf
-rw-r--r-- 1 root root  7984 Jul 15 12:16 storage.conf
-rw-r--r-- 1 root root  7927 Jul 15 12:16 storage.conf.sample
-rw-r--r-- 1 root root   105 Jul 15 12:16 storage_ids.conf
-rw-r--r-- 1 root root   105 Jul 15 12:16 storage_ids.conf.sample
-rw-r--r-- 1 root root  7420 Jul 15 12:16 tracker.conf
-rw-r--r-- 1 root root  7389 Jul 15 12:16 tracker.conf.sample
[root@myj12 fdfs]#
```

接下来就需要编辑刚拷贝的mod_fastdfs.conf文件了，打开mod_fastdfs.conf并按顺序依次编译以下内容：

base_path=/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage #保存日志目录

tracker_server=myj12:22122 #tracker服务器的IP地址以及端口号

```
storage_server_port=23000 #storage服务器的端口号
url_have_group_name = true #文件 url 中是否有 group 名
store_path0=/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage_data # 存储路径
group_count = 3 #设置组的个数，事实上这次只使用了group1
```

设置了group_count = 3，接下来就需要在文件尾部追加这3个group setting：

设置了group_count = 3，接下来就需要在文件尾部追加这3个group setting：

```
[group1]
group_name=group1
storage_server_port=23000
store_path_count=1
store_path0=/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage_data

[group2]
group_name=group2
storage_server_port=23000
store_path_count=1
store_path0=/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage_data

[group3]
group_name=group3
storage_server_port=23000
store_path_count=1
store_path0=/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage_data
```

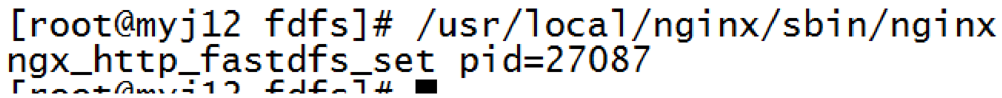
接下来还需要建立 M00 至存储目录的符号连接：

```
ln -s /home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage_data/data /home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage_data/data/M00
```

最后启动nginx：

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

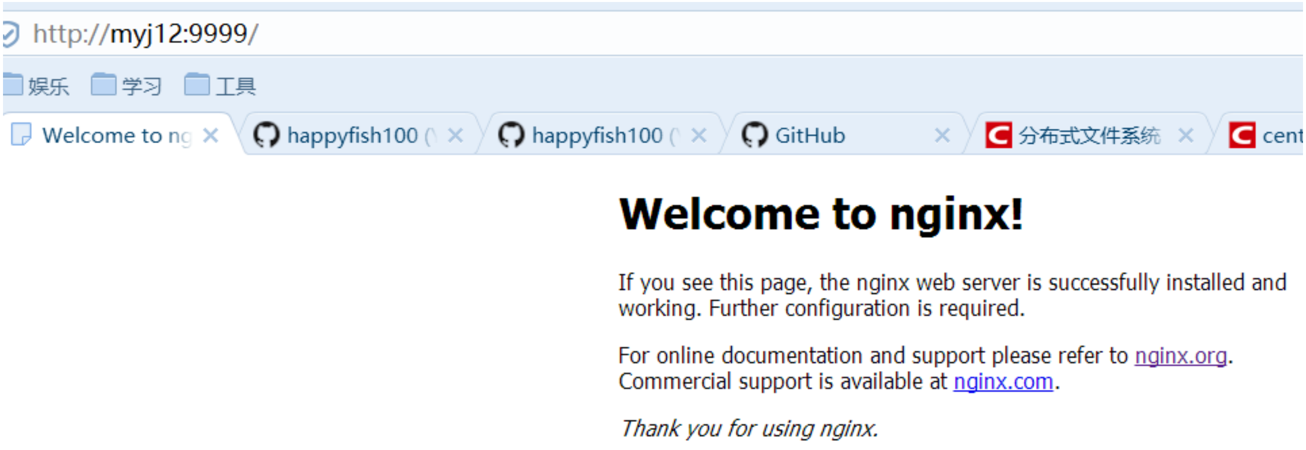
显示如下信息说明nginx已启动成功：



```
[root@myj12 fdfs]# /usr/local/nginx/sbin/nginx
ngx_http_fastdfs_set pid=27087
[root@myj12 fdfs]# █
```

通过浏览器也可以看到nginx的主页：

http://myj12:9999/



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

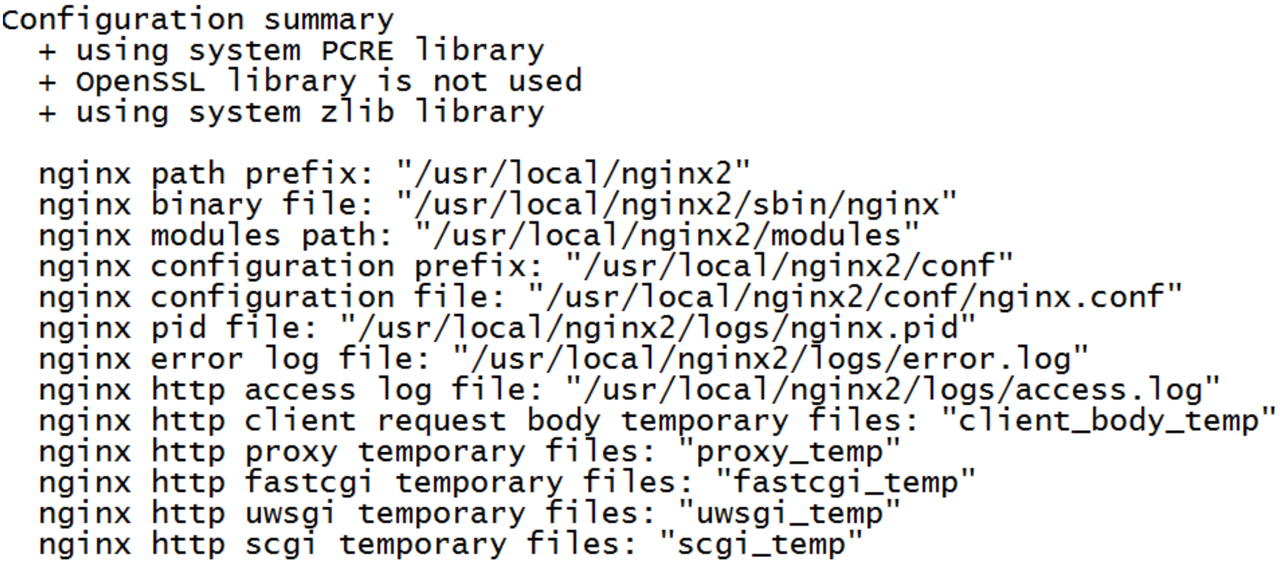
Thank you for using nginx.

storage服务器的nginx就已经安装完毕，接下来看一下tracker服务器的nginx安装。

7、**tracker nginx**

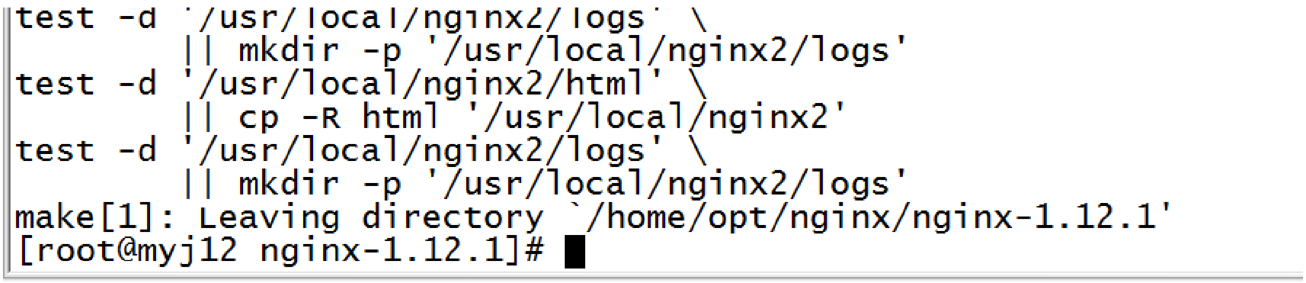
同理，再装一个nginx，目录命名为nginx2，安装路径依旧放在/usr/local下，由于和之前一样，此处就不再做详细解释：

mkdir nginx2
cd /home/opt/nginx/nginx-1.12.1/
./configure --prefix=/usr/local/nginx2 --add-module=/home/opt/fastdfs/fastdfs-nginx-module-master/src
编译
make



```
Configuration summary
  + using system PCRE library
  + OpenSSL library is not used
  + using system zlib library

  nginx path prefix: "/usr/local/nginx2"
  nginx binary file: "/usr/local/nginx2/sbin/nginx"
  nginx modules path: "/usr/local/nginx2/modules"
  nginx configuration prefix: "/usr/local/nginx2/conf"
  nginx configuration file: "/usr/local/nginx2/conf/nginx.conf"
  nginx pid file: "/usr/local/nginx2/logs/nginx.pid"
  nginx error log file: "/usr/local/nginx2/logs/error.log"
  nginx http access log file: "/usr/local/nginx2/logs/access.log"
  nginx http client request body temporary files: "client_body_temp"
  nginx http proxy temporary files: "proxy_temp"
  nginx http fastcgi temporary files: "fastcgi_temp"
  nginx http uwsgi temporary files: "uwsgi_temp"
  nginx http scgi temporary files: "scgi_temp"
```

安装
make install

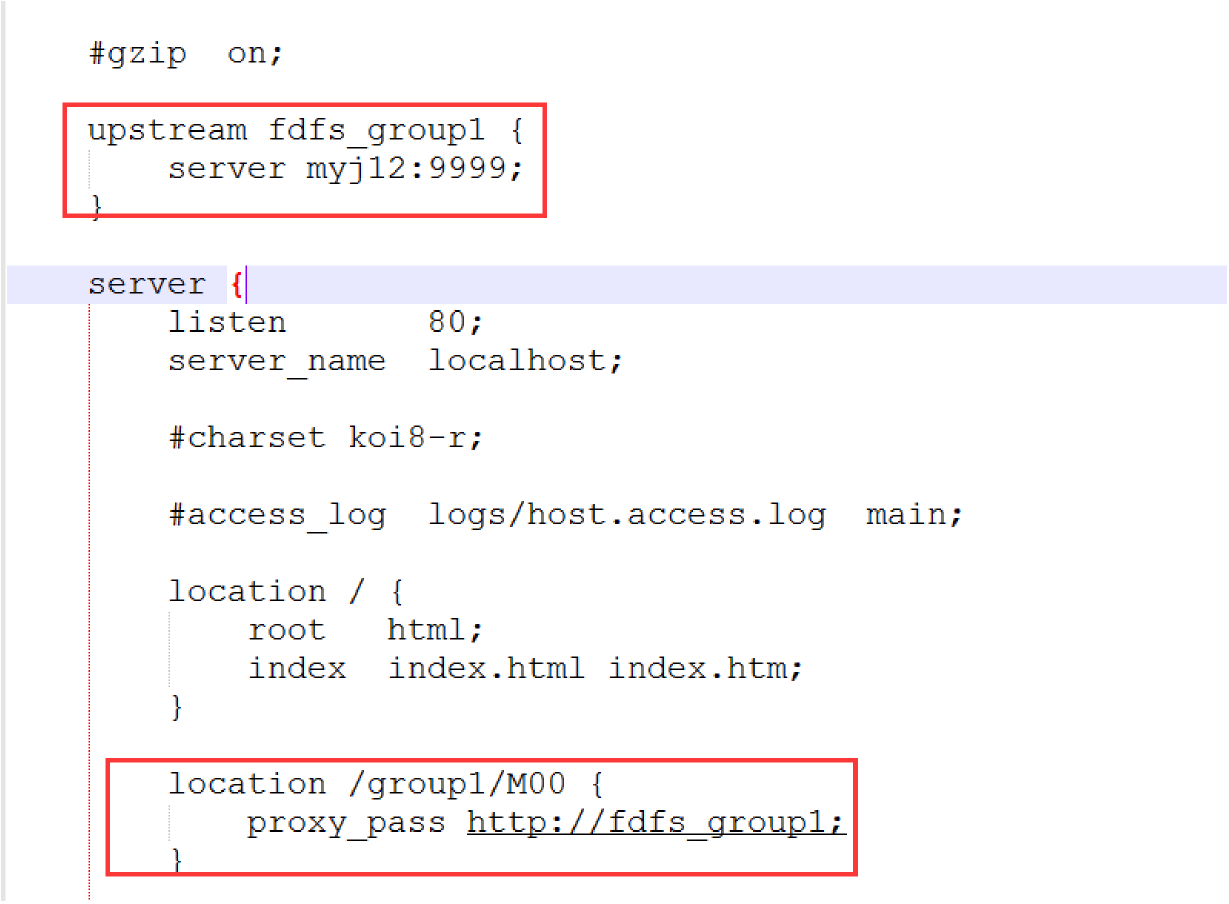


```
test -d '/usr/local/nginx2/logs' \
        || mkdir -p '/usr/local/nginx2/logs'
test -d '/usr/local/nginx2/html' \
        || cp -R html '/usr/local/nginx2'
test -d '/usr/local/nginx2/logs' \
        || mkdir -p '/usr/local/nginx2/logs'
make[1]: Leaving directory `/home/opt/nginx/nginx-1.12.1'
[root@myj12 nginx-1.12.1]# 
```

接下来依然是修改nginx2的配置文件，进入conf目录并打开nginx.conf文件加入以下配置，storage的nginx无需修改listen端口，即默认的80端口，并将upstream指向tracker的nginx地址：

upstream fdfs_group1 {

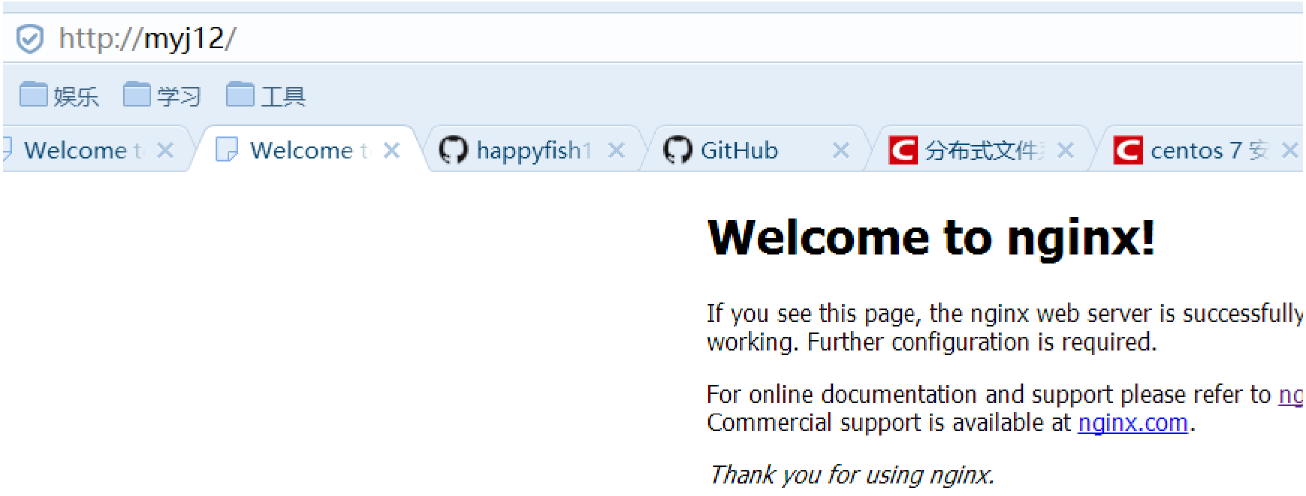
```
    server myj12:9999;
}

location /group1/M00 {
    proxy_pass http://fdfs_group1;
}
```



```
  #gzip  on;

  upstream fdfs_group1 {
      server myj12:9999;
  }

  server {
      listen       80;
      server_name  localhost;

      #charset koi8-r;

      #access_log  logs/host.access.log  main;

      location / {
          root   html;
          index  index.html index.htm;
      }

      location /group1/M00 {
          proxy_pass http://fdfs_group1;
      }
```

接下来启动nginx2：
/usr/local/nginx2/sbin/nginx

此时访问nginx2的主页，由于没有修改端口，直接访问ip地址即可：



Welcome to nginx!

If you see this page, the nginx web server is successfully working. Further configuration is required.

For online documentation and support please refer to ng
Commercial support is available at nginx.com.

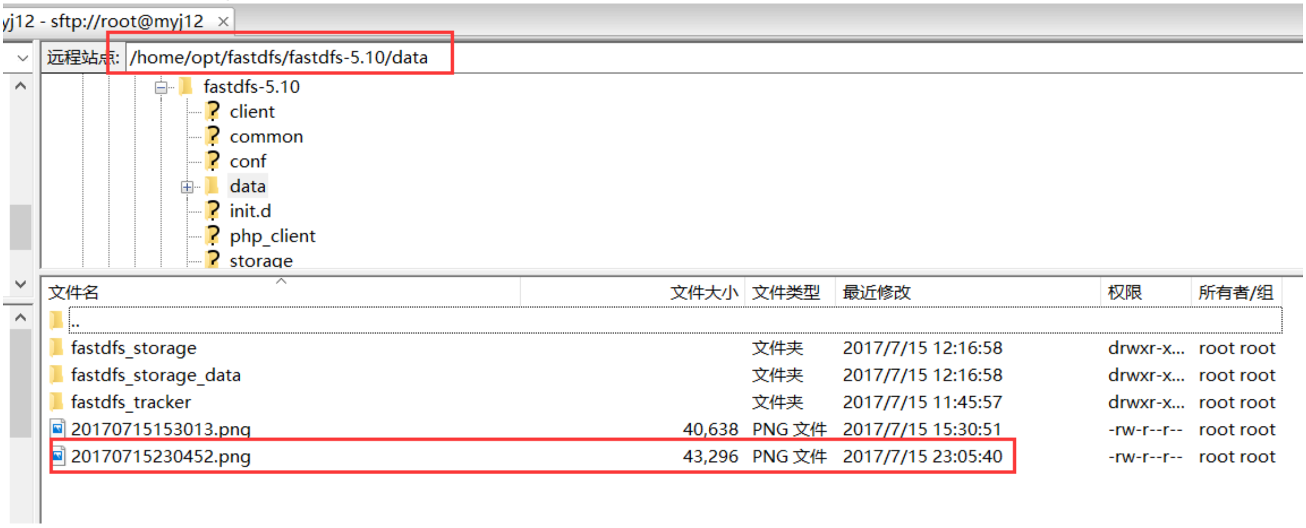
Thank you for using nginx.

最后一步就是需要修改/etc/fdfs目录下的client.conf文件，打开该文件并加入以下配置：

base_path=/home/opt/fastdfs/fastdfs-5.10/data/fastdfs_storage #日志存放路径
tracker_server=myj12:22122 #tracker 服务器 IP 地址和端口号

http.tracker_server_port=6666 # tracker 服务器的 http 端口号，必须和tracker的设置对应起来
至此关于fastdfs就已经全部配置完毕了，再一次进行测试看看是否能正常上传文件并通过http访问文件。

HTTP测试
再给/home/opt/fastdfs/fastdfs-5.10/data目录下上传一张图：



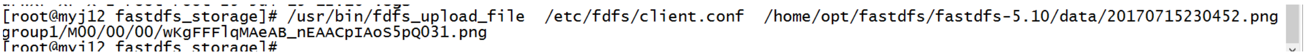
yj12 - sftp://root@myj12

远程站点: /home/opt/fastdfs/fastdfs-5.10/data

```
fastdfs-5.10
    client
    common
    conf
    data
    init.d
    php_client
    storage
```

文件名	文件大小	文件类型	最近修改	权限	所有者/组
..					
fastdfs_storage		文件夹	2017/7/15 12:16:58	drwxr-x...	root root
fastdfs_storage_data		文件夹	2017/7/15 12:16:58	drwxr-x...	root root
fastdfs_tracker		文件夹	2017/7/15 11:45:57	drwxr-x...	root root
20170715153013.png	40,638	PNG 文件	2017/7/15 15:30:51	-rw-r--r--	root root
20170715230452.png	43,296	PNG 文件	2017/7/15 23:05:40	-rw-r--r--	root root

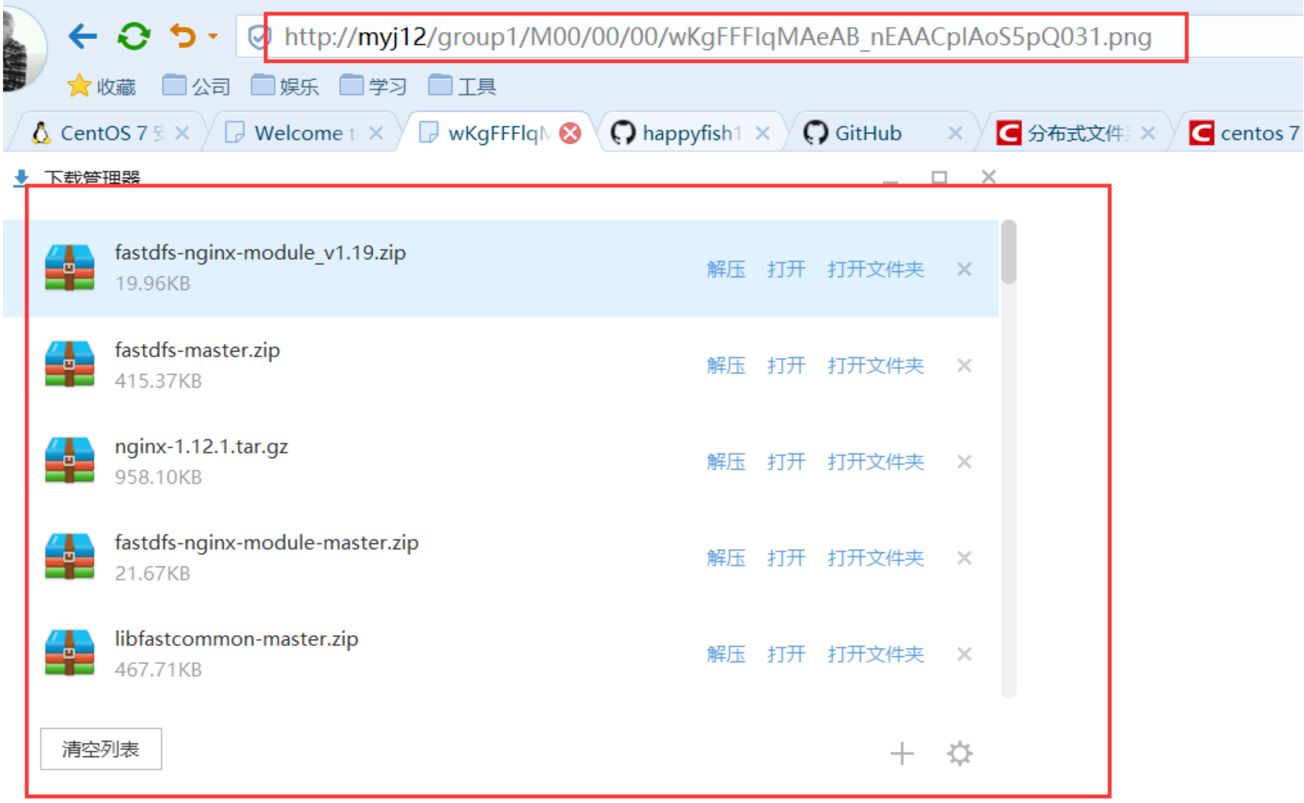
通过客户端命令测试上传：
/usr/bin/fdfs_upload_file /etc/fdfs/client.conf /home/opt/fastdfs/fastdfs-5.10/data/20170715230452.png



```
[root@myj12 fastdfs_storage]# /usr/bin/fdfs_upload_file  /etc/fdfs/client.conf  /home/opt/fastdfs/fastdfs-5.10/data/20170715230452.png
group1/M00/00/00/wKgFFFlqMAeAB_nEAACpIAoS5pQ031.png
[root@myj12 fastdfs_storage]#
```

返回路径
group1/M00/00/00/wKgFFFlqMAeAB_nEAACpIAoS5pQ031.png

如上图，依旧上传成功，接下来的关键就是通过HTTP测试文件访问，打开浏览器输入ip地址+文件名看看是否能正常访问该图片：
http://myj12/group1/M00/00/00/wKgFFFlqMAeAB_nEAACpIAoS5pQ031.png



http://myj12/group1/M00/00/00/wKgFFFlqMAeAB_nEAACpIAoS5pQ031.png

下载管理器

fastdfs-nginx-module_v1.19.zip 19.96KB 解压 打开 打开文件夹

fastdfs-master.zip 415.37KB 解压 打开 打开文件夹

nginx-1.12.1.tar.gz 958.10KB 解压 打开 打开文件夹

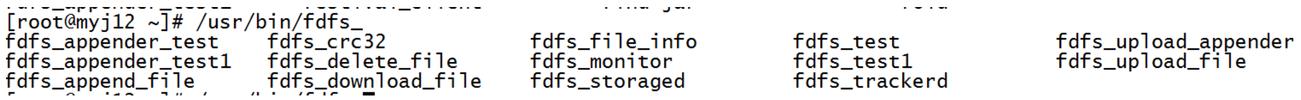
fastdfs-nginx-module-master.zip 21.67KB 解压 打开 打开文件夹

libfastcommon-master.zip 467.71KB 解压 打开 打开文件夹

清空列表

一切正常，至此关于FastDFS在CentOS 7下的部署测试就已经全部完成了。

8、fastdfs命令



```
[root@myj12 ~]# /usr/bin/fdfs_
fdfs_appender_test    fdfs_crc32            fdfs_file_info        fdfs_test             fdfs_upload_appender
fdfs_appender_test1   fdfs_delete_file      fdfs_monitor          fdfs_test1            fdfs_upload_file
fdfs_append_file      fdfs_download_file    fdfs_storaged         fdfs_trackerd
```

1：启动FastDFS

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

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

2：关闭FastDFS

tracker:/usr/bin/fdfs_trackerd /etc/fdfs/tracker.conf stop

storage:/usr/bin/fdfs_storaged /etc/fdfs/storage.conf stop

或者 killall fdfs_trackered(storaged) -------注意，千万不要使用-9强行杀死进程。

3：重启FastDFS

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

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

4：查看集群情况

在任意一台storage(tracker也可以) /usr/bin/fdfs_monitor /etc/fdfs/storage.conf

5：删除一个storage：

在任意一台storage(tracker也可以) /usr/bin/fdfs_monitor /etc/fdfs/storage.conf delete group2 myj12

总结

本篇文章记录了开源分布式文件系统FastDFS在Linux CentOS 7中安装部署以及测试的全过程，下一篇文章将会继续介绍通过Java客户端以及SpringMVC中结合FastDFS实现文件上传下载，The End。

注：安装参考地址

http://www.linuxidc.com/Linux/2016-09/135537.htm