fastDFS系统搭建

1、 下载安装 libfastcommon

libfastcommon是从 FastDFS 和 FastDHT 中提取出来的公共 C 函数库，基础环境，安装即可

(1) 下载libfastcommon

```
wget https://github.com/happyfish100/libfastcommon/archive/V1.0.7.tar.gz
```

(2) 解压

```
 tar -zxvf V1.0.7.tar.gz
 cd libfastcommon-1.0.7
```

(3) 安装基础环境gcc g++

```
gcc:
yum -y install gcc automake autoconf libtool make
g++:
yum install gcc gcc-c++
pcre:
wget ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.42.tar.gz
解压pcre：tar -zxvf pcre-8.42.tar.gz
进入prce-8.42文件夹执行命令:./configure
                            make
                            make install
```

(4) 编译、安装

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

(5) libfastcommon.so 安装到了/usr/lib64/libfastcommon.so，但是FastDFS主程序设置的lib目录是/usr/local/lib，所以需要创建软链接

```
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
```

2、下载安装FastDFS

（1）下载FastDFS

```
wget https://github.com/happyfish100/fastdfs/archive/V5.05.tar.gz
```

（2）解压

```
tar -zxvf V5.05.tar.gz
cd fastdfs-5.05
```

（3）编译、安装

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

（4）默认安装方式安装后的相应文件与目录

A、服务脚本：

```
/etc/init.d/fdfs_storaged
/etc/init.d/fdfs_trackerd
```

B、配置文件（这三个是作者给的样例配置文件）：

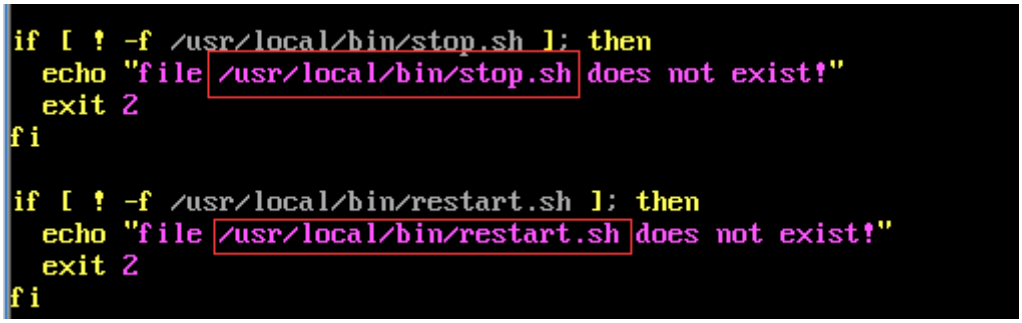
```
/etc/fdfs/client.conf.sample
/etc/fdfs/storage.conf.sample
/etc/fdfs/tracker.conf.sample
```

C、命令工具在 /usr/bin/ 目录下：

```
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
stop.sh
restart.sh
```

(5) FastDFS 服务脚本设置的 bin 目录是 /usr/local/bin， 但实际命令安装在 /usr/bin/ 下。
两种方式:
①修改FastDFS 服务脚本中相应的命令路径，也就是把 /etc/init.d/fdfs_storaged 和 /etc/init.d/fdfs_tracker 两个脚本中的 /usr/local/bin 修改成 /usr/bin。

```
vim fdfs_trackerd
使用查找替换命令进统一修改:%s+/usr/local/bin+/usr/bin
 vim fdfs_storaged
 使用查找替换命令进统一修改:%s+/usr/local/bin+/usr/bin
```



②是建立 /usr/bin 到 /usr/local/bin 的软链接，我是用这种方式。

```
ln -s /usr/bin/fdfs_trackerd   /usr/local/bin
ln -s /usr/bin/fdfs_storaged   /usr/local/bin
ln -s /usr/bin/stop.sh         /usr/local/bin
ln -s /usr/bin/restart.sh      /usr/local/bin
```

3、配置FastDFS跟踪器(Tracker)

（1） 进入 /etc/fdfs，复制 FastDFS 跟踪器样例配置文件 tracker.conf.sample，并重命名为 tracker.conf。

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

（2）编辑tracker.conf ，后2项的需要修改下，其它的默认即可

```
#配置文件是否不生效，false 为生效
disabled=false
#提供服务的端口
port=22122
# Tracker 数据和日志目录地址(根目录必须存在,子目录会自动创建)
base_path=/ljzsg/fastdfs/tracker
# HTTP 服务端口
http.server_port=80
```

（3）创建tracker基础数据目录，即base_path对应的目录

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

（4）防火墙中打开跟踪端口（默认的22122）

```
vim /etc/sysconfig/iptables
添加如下端口行:
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22122 -j ACCEPT
重启防火墙:
service iptables restart
```

（5）启动Tracker
初次成功启动，会在 /ljzsg/fdfsdfs/tracker/ (配置的base_path)下创建 data、logs 两个目录。

```
可以用这种方式启动
/etc/init.d/fdfs_trackerd start
也可以用这种方式启动，前提是上面创建了软链接，后面都用这种方式
service fdfs_trackerd start
```

查看 FastDFS Tracker 是否已成功启动，22122端口正在被监听，则算是Tracker服务安装成功。

```
netstat -unltp|grep fdfs
```

（6）关闭Tracker命令：

```
service fdfs_trackerd stop
```

（7）设置Tracker开机启动

```
chkconfig fdfs_trackerd on
或者：
 vim /etc/rc.d/rc.local
加入配置：
/etc/init.d/fdfs_trackerd start
```

（8）tracker server 目录及文件结构
Tracker服务启动成功后，会在base_path下创建data、logs两个目录。目录结构如下

```
${base_path}
|__data
|   |__storage_groups.dat：存储分组信息
|   |__storage_servers.dat：存储服务器列表
|__logs
|   |__trackerd.log： tracker server 日志文件
```

4、 配置 FastDFS 存储 (Storage)

（1）进入 /etc/fdfs 目录，复制 FastDFS 存储器样例配置文件 storage.conf.sample，并重命名为 storage.conf

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

（2）编辑storage.conf
标#号的需要修改，其它的默认即可。

```
配置文件是否不生效，false 为生效
disabled=false

指定此 storage server 所在 组(卷)
group_name=group1
storage server 服务端口
port=23000

心跳间隔时间，单位为秒 (这里是指主动向 tracker server 发送心跳)
heart_beat_interval=30

# Storage 数据和日志目录地址(根目录必须存在，子目录会自动生成)
base_path=/ljzsg/fastdfs/storage

存放文件时 storage server 支持多个路径。这里配置存放文件的基路径数目，通常只配一个目录。
store_path_count=1

逐一配置 store_path_count 个路径，索引号基于 0。

# 如果不配置 store_path0，那它就和 base_path 对应的路径一样。
store_path0=/ljzsg/fastdfs/file

FastDFS 存储文件时，采用了两级目录。这里配置存放文件的目录个数。
如果本参数只为 N（如： 256），那么 storage server 在初次运行时，会在 store_path 下自动创建 N * N
subdir_count_per_path=256

tracker_server 的列表 ，会主动连接 tracker_server

# 有多个 tracker server 时，每个 tracker server 写一行
tracker_server=192.168.241.147:22122

允许系统同步的时间段 (默认是全天) 。一般用于避免高峰同步产生一些问题而设定。
sync_start_time=00:00
sync_end_time=23:59
# 访问端口
http.server_port=80
```

(3) 创建Storage基础数据目录，对应base_path目录

```
mkdir -p /ljzsg/fastdfs/storage
这是配置的store_path0路径
mkdir -p /ljzsg/fastdfs/file
```

(4) 防火墙中打开存储器端口（默认的 23000）

```
vim /etc/sysconfig/iptables
添加如下端口行:
-A INPUT -m state --state NEW -m tcp -p tcp --dport 23000 -j ACCEPT
重启防火墙:
service iptables restart
```

(5) 启动 Storage
启动Storage前确保Tracker是启动的。初次启动成功，会在 /ljzsg/fastdfs/storage 目录下创建 data、 logs 两个目录。

查看Tracker是否启动，若没启动 则启动

```
netstat -unltp|grep fdfs
```

可以用这种方式启动

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

也可以用这种方式，后面都用这种

```
service fdfs_storaged start
```

查看 Storage 是否成功启动，23000 端口正在被监听，就算 Storage 启动成功

```
netstat -unltp|grep fdfs
```

关闭Storage命令:

```
service fdfs_storaged stop
```

查看Storage和Tracker是否在通信:

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

```
^[[B    Storage 1:
            id = 192.168.241.147
            ip_addr = 192.168.241.147 (localhost.localdomain)  ACTIVE
            http domain =
            version = 5.05
            join time = 2019-01-03 17:34:25
            up time = 2019-01-03 17:35:44
            total storage = 51059 MB
            free storage = 49674 MB
            upload priority = 10
```

Storage和Tracker是否在通信

（6）设置 Storage 开机启动

```
chkconfig fdfs_storaged on
或者：
vim /etc/rc.d/rc.local
加入配置：
/etc/init.d/fdfs_storaged start
```

（7）Storage 目录

同 Tracker，Storage 启动成功后，在base_path 下创建了data、logs目录，记录着 Storage Server 的信息。

在 store_path0 目录下，创建了N*N个子目录：

```
[root@localhost /]# cd ljzsg/fastdfs/file/
[root@localhost file]# cd data/
[root@localhost data]# ls
00  08  10  18  20  28  30  38  40  48  50  58  60  68  70  78  80  8
01  09  11  19  21  29  31  39  41  49  51  59  61  69  71  79  81  8
02  0A  12  1A  22  2A  32  3A  42  4A  52  5A  62  6A  72  7A  82  8
03  0B  13  1B  23  2B  33  3B  43  4B  53  5B  63  6B  73  7B  83  8
04  0C  14  1C  24  2C  34  3C  44  4C  54  5C  64  6C  74  7C  84  8
05  0D  15  1D  25  2D  35  3D  45  4D  55  5D  65  6D  75  7D  85  8
06  0E  16  1E  26  2E  36  3E  46  4E  56  5E  66  6E  76  7E  86  8
07  0F  17  1F  27  2F  37  3F  47  4F  57  5F  67  6F  77  7F  87  8
```

*N*N个子目录*

5、文件上传测试

（1）修改 Tracker 服务器中的客户端配置文件

```
cd /etc/fdfs
cp client.conf.sample client.conf
vim client.conf
```

修改如下配置即可，其它默认。

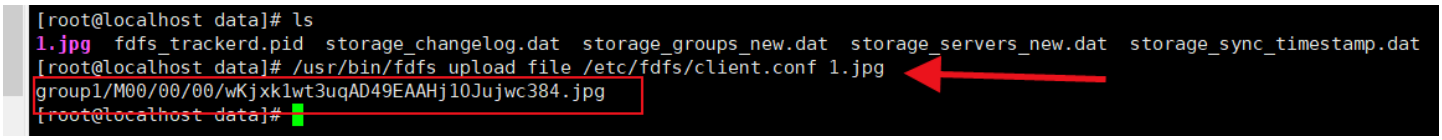
```
# Client 的数据和日志目录
base_path=/ljzsg/fastdfs/client
Tracker端口
tracker_server=192.168.241.147:22122
```

（2）上传测试

在linux内部执行如下命令上传 namei.jpeg 图片

```
/usr/bin/fdfs_upload_file /etc/fdfs/client.conf namei.jpeg
```

上传成功后返回文件ID
group1/M00/00/00/wKjxk1wt3uqAD49EAAHj1OJujwc384.jpg



ID

6、安装Nginx(此处省略)

修改nginx配置文件

```
vim /usr/local/nginx/conf/nginx.conf
```

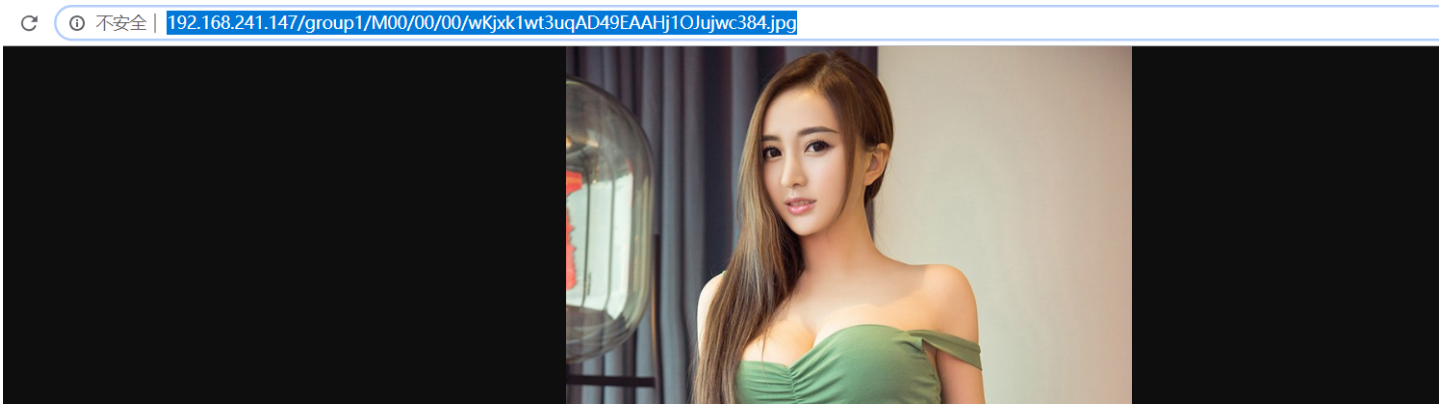
添加如下行，将 /group1/M00 映射到 /ljzsg/fastdfs/file/data

```
location /group1/M00 {
alias /ljzsg/fastdfs/file/data;
}
```

重启nginx

```
/usr/local/nginx/sbin/nginx -s reload
```

在浏览器访问之前上传的图片、成功。
http://192.168.241.147/group1/M00/00/00/wKjxk1wt3uqAD49EAAHj1OJujwc384.jpg



测试上传图片

7、 FastDFS 配置 Nginx 模块

1、安装配置Nginx模块
① fastdfs-nginx-module 模块说明

FastDFS 通过 Tracker 服务器，将文件放在 Storage 服务器存储， 但是同组存储服务器之间需要进行文件复制， 有同步延迟的问题。

假设 Tracker 服务器将文件上传到了 192.168.51.128，上传成功后文件 ID已经返回给客户端。

此时 FastDFS 存储集群机制会将这个文件同步到同组存储 192.168.51.129，在文件还没有复制完成的情况下，客户端如果用这个文件 ID 在 192.168.51.129 上取文件,就会出现文件无法访问的错误。

而 fastdfs-nginx-module 可以重定向文件链接到源服务器取文件，避免客户端由于复制延迟导致的文件无法访问错误。

② 下载 fastdfs-nginx-module、解压
这里为啥这么长一串呢，因为最新版的master与当前nginx有些版本问题

```
wget https://github.com/happyfish100/fastdfs-nginx-module/archive/5e5f3566bbfa57418b5506aae
```

解压

```
unzip 5e5f3566bbfa57418b5506aaefbe107a42c9fcb1.zip
```

重命名

```
mv fastdfs-nginx-module-5e5f3566bbfa57418b5506aaefbe107a42c9fcb1  fastdfs-nginx-module-mast
```

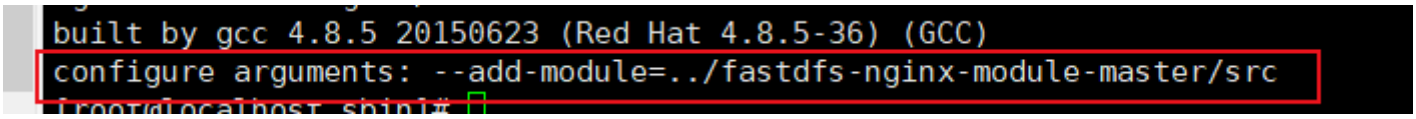
③ 配置Nginx
在nginx中添加模块

```
先停掉nginx服务
    /usr/local/nginx/sbin/nginx -s stop

进入解压包目录
    cd /usr/loacl/nginx-1.14.1/

添加模块
    ./configure --add-module=../fastdfs-nginx-module-master/src

重新编译、安装
     make &amp;amp;&amp;amp; make install
```

④ 查看Nginx的模块

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

有下面这个就说明添加模块成功



nginx添加模块

⑤复制 fastdfs-nginx-module 源码中的配置文件到/etc/fdfs 目录， 并修改

```
cd /usr/local/fastdfs-nginx-module-master/src
cp mod_fastdfs.conf /etc/fdfs/
```

修改如下配置，其它默认

```
# 连接超时时间
connect_timeout=10

# Tracker Server
tracker_server=192.168.241.147:22122

# StorageServer 默认端口
```

```
storage_server_port=23000

# 如果文件ID的uri中包含/group**，则要设置为true
url_have_group_name = true

# Storage 配置的store_path0路径，必须和storage.conf中的一致
store_path0=/ljzsg/fastdfs/file
```

⑥ 复制 FastDFS 的部分配置文件到/etc/fdfs 目录

```
cd /usr/local/fastdfs-5.05/conf/
cp anti-steal.jpg http.conf mime.types /etc/fdfs/
```

⑦ 配置nginx，修改nginx.conf

```
vim /usr/local/nginx/conf/nginx.conf
```

修改配置，其它的默认

```
将原来添加的配置注释掉
#location /group1/M00 {
            #alias /ljzsg/fastdfs/file/data;
      # }

在80端口下添加fastdfs-nginx模块
    location ~/group([0-9])/M00 {
    ngx_fastdfs_module;
    }
```

```
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   html;
    }

  # location /group1/M00 {
    #   alias /ljzsg/fastdfs/file/data;
  # }

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

nginx配置

注意：

listen 80 端口值是要与 /etc/fdfs/storage.conf 中的 http.server_port=80 (前面改成80了)相对应。如果改成其它端口，则需要统一，同时在防火墙中打开该端口。

location 的配置，如果有多个group则配置location ~/group([0-9])/M00 ，没有则不用配group。

⑧ 在/ljzsg/fastdfs/file 文件存储目录下创建软连接，将其链接到实际存放数据的目录，这一步可以省略。

```
ln -s /ljzsg/fastdfs/file/data/ /ljzsg/fastdfs/file/data/M00
```

⑨ 启动nginx

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

打印处如下就算配置成功

```
[root@localhost conf]# /usr/local/nginx/sbin/nginx
ngx_http_fastdfs_set pid=7403
[root@localhost conf]#
```

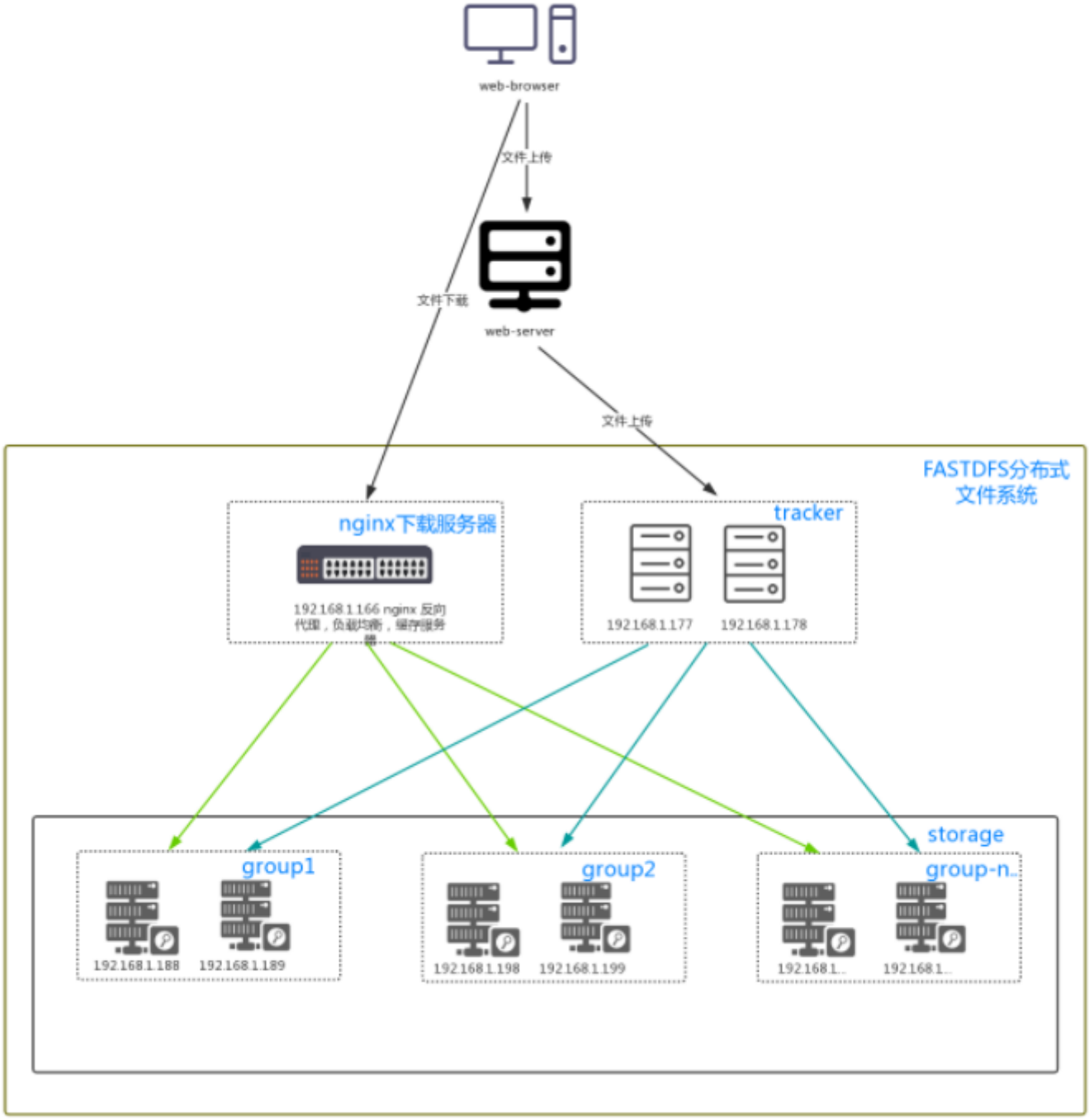
nginx配置成功

⑩ 在地址栏访问。

能下载文件就算安装成功。注意和第三点中直接使用nginx路由访问不同的是，这里配置 fastdfs-nginx-module 模块，可以重定向文件链接到源服务器取文件。

http://192.168.241.147/group1/M00/00/00/wKjxk1wt3uqAD49EAAHj1OJujwc384.jpg

最终部署结构图(盗的图)：可以按照下面的结构搭建环境。



fastDFS部署结构图

8、 Java客户端

前面文件系统平台搭建好了，现在就要写客户端代码在系统中实现上传下载，这里只是简单的测试代码。

1、首先需要搭建 FastDFS 客户端Java开发环境
① 项目中使用maven进行依赖管理，可以在pom.xml中引入如下依赖即可：

```
net.oschina.zcx7878
fastdfs-client-java
1.27.0.0
```

其它的方式，参考官方文档：https://github.com/happyfish100/fastdfs-client-java

```
192.168.241.147:80/group1/M00/00/00/wKjxk1x9FxiABljDAABXBJT7CUI108.jpg
```