**简介**:

**FastDFS**是一款开源的轻量级分布式文件系统纯C实现，支持Linux、FreeBSD等UNIX系统类google FS，不是通用的文件系统，只能通过专有API访问，目前提供了C、Java和PHP API为互联网应用量身定做，解决大容量文件存储问题，追求高性能和高扩展性FastDFS可以看做是基于文件的key value pair存储系统，称作分布式文件存储服务更为合适。  
 tracker-server: 跟踪服务器， 主要做调度工作， 起负载均衡的作用。 在内存中记录集群中所有存储组和存储服务器的状态信息， 是客户端和数据服务器交互的枢纽。 相比GFS中的master更为精简， 不记录文件索引信息， 占用的内存量很少。

Storage-server: 存储服务器（ 又称：存储节点或数据服务器） ， 文件和文件属性（ metadata） 都保存到存储服务器上。 Storage server直接利用OS的文件系统调用管理文件。

Group: 组， 也可称为卷。 同组内服务器上的文件是完全相同的 ，同一组内的storage server之间是对等的， 文件上传、 删除等操作可以在任意一台storage server上进行 。

meta data：文件相关属性，键值对（ Key Value Pair） 方式，如：width=1024,heigth=768 。

**安装**

Centos 6.4

Su password root权限

**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 。

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

解压：

# tar -zxvf V1.0.7.tar.gz

# cd libfastcommon-1.0.7

编译，安装：

# ./make.sh

# ./make.sh install

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**

下载

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

解压：

# tar -zxvf V5.05.tar.gz

# cd fastdfs-5.05

编译，安装：

# ./make.sh

# ./make.sh install

默认安装方式安装后的相应文件和目录：

A．服务脚本：

/etc/init.d/fdfs\_storaged

/etc/init.d/fdfs\_tracker

B．配置文件：

/etc/fdfs/client.conf.sample

/etc/fdfs/storage.conf.sample

/etc/fdfs/tracker.conf.sample

C．命令工具（/usr/bin）

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

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。

创建软连接

# 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tracker**

进入复制 FastDFS 跟踪器样例配置文件 tracker.conf.sample，重命名为 tracker.conf。

# cd /etc/fdfs

# cp tracker.conf.sample tracker.conf

# vim tracker.conf

编辑tracker.conf

# 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

创建tracker基础数据目录，即base\_path对应的目录

# mkdir -p /ljzsg/fastdfs/tracker

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

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。

启动方式:

# /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



关闭Tracker命令：

# service fdfs\_trackerd stop

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

 tracker server 目录及文件结构

${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storage

进入/etc/fdfs 目录，复制 storage.conf.sample，并重命名为 storage.conf

# cd /etc/fdfs

# cp storage.conf.sample storage.conf  
# vim storage.conf

编辑storage.conf

# 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 个存放文件的子目录。

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=file.ljzsg.com: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

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

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

启动 Storage

# /etc/init.d/fdfs\_storaged start

也可以用这种方式，后面都用这种

# service fdfs\_storaged start

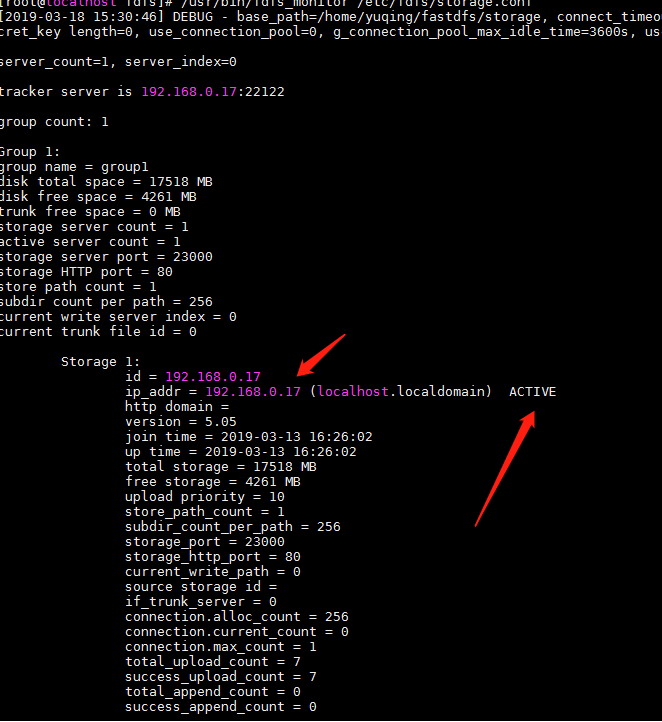


关闭Storage命令：

# service fdfs\_storaged stop

查看Storage和Tracker是否在通信：

/usr/bin/fdfs\_monitor /etc/fdfs/storage.conf



即为通信

 设置 Storage 开机启动

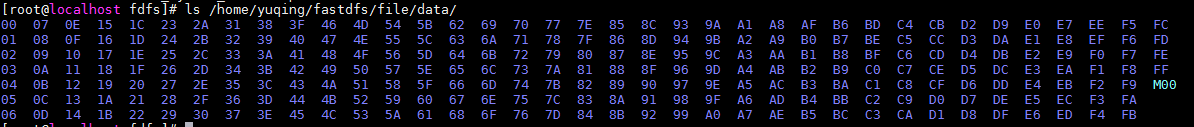
# chkconfig fdfs\_storaged on

或者：  
# vim /etc/rc.d/rc.local  
加入配置：  
/etc/init.d/fdfs\_storaged start

 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：



**文件上传测试**

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=/home/yuqing/fastdfs/client

# Tracker端口

tracker\_server=192.168.0.17:22122

上传测试

/usr/bin/fdfs\_upload\_file /etc/fdfs/client.conf 文件

上传成功后返回文件ID号：group1/M00/00/00/wKgz6lnduTeAMdrcAAEoRmXZPp870.jpeg

5.安装nginx

以上步骤只能上传，还无法下载，nginx作为服务器支持http方式访问文件。

### 

### 安装nginx所需环境

① gcc 安装

# yum install gcc-c++

② PCRE pcre-devel 安装

# yum install -y pcre pcre-devel

③ zlib 安装

# yum install -y zlib zlib-devel

④ OpenSSL 安装

# yum install -y openssl openssl-devel

### 安装Nginx

① 下载nginx

# wget -c https://nginx.org/download/nginx-1.12.1.tar.gz

② 解压

# tar -zxvf nginx-1.12.1.tar.gz

# cd nginx-1.12.1

③ 使用默认配置

# ./configure

④ 编译、安装

# make

# make install

启动nginx

[复制代码](javascript:void(0);)

# cd /usr/local/nginx/sbin

/

# ./nginx

其它命令

# ./nginx -s stop

# ./nginx -s quit

# ./nginx -s reload

设置开机启动

[复制代码](javascript:void(0);)

# vim /etc/rc.local

添加一行：

/usr/local/nix/sin/nix  
  
# 设置执行权限  
# chimed 755 rc.local

查看nix的版本及模块

/usr/local/nix/sin/nix -V



防火墙中打开Nginx端口（默认的 80）

添加后就能在本机使用80端口访问了。

[复制代码](javascript:void(0);)

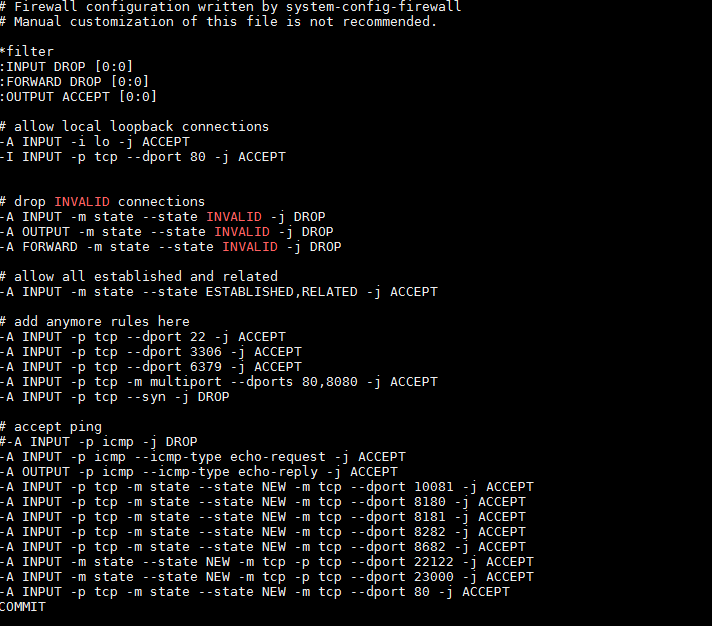
# vim /etc/sysconfig/iptables

添加如下端口行：

-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT

重启防火墙：

# service iptables restart



**访问文件**

简单的测试访问文件

① 修改nginx.conf

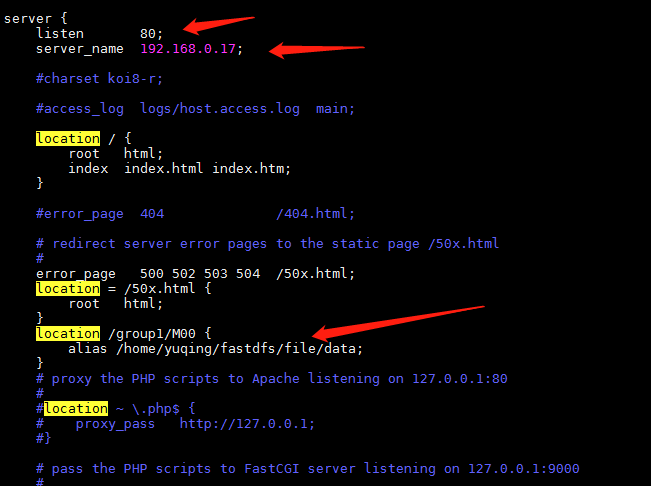
# vim /usr/local/nix/conf/nginx.conf

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/sin/nginx -s reload



在浏览器访问之前上传的图片、成功。

<http://192.168.0.17/group1/M00/00/00/>文件名

**FastDFS 配置 Nginx 模块**

### 1、安装配置Nginx模块

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。

下载 fastdfs-nginx-module、解压

# wget https://github.com/happyfish100/fastdfs-nginx-module/archive/5e5f3566bbfa57418b5506aaefbe107a42c9fcb1.zip

# 解压

# unzip 5e5f3566bbfa57418b5506aaefbe107a42c9fcb1.zip

# 重命名

# mv fastdfs-nginx-module-5e5f3566bbfa57418b5506aaefbe107a42c9fcb1 fastdfs-nginx-module-master

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 /softpackages/nginx-1.12.1/

# 添加模块

# ./configure --add-module=../fastdfs-nginx-module-master/src

重新编译、安装

# make && make install

查看Nginx的模块

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



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

# cd /softpackages/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=192.168.0.17:22122

# 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=/home/yuqing/fastdfs/file

复制 FastDFS 的部分配置文件到/etc/fdfs 目录

# cd /softpackages/fastdfs-5.05/conf/

# cp anti-steal.jpg http.conf mime.types /etc/fdfs/

配置nginx，修改nginx.conf

# vim /usr/local/nginx/conf/nginx.conf

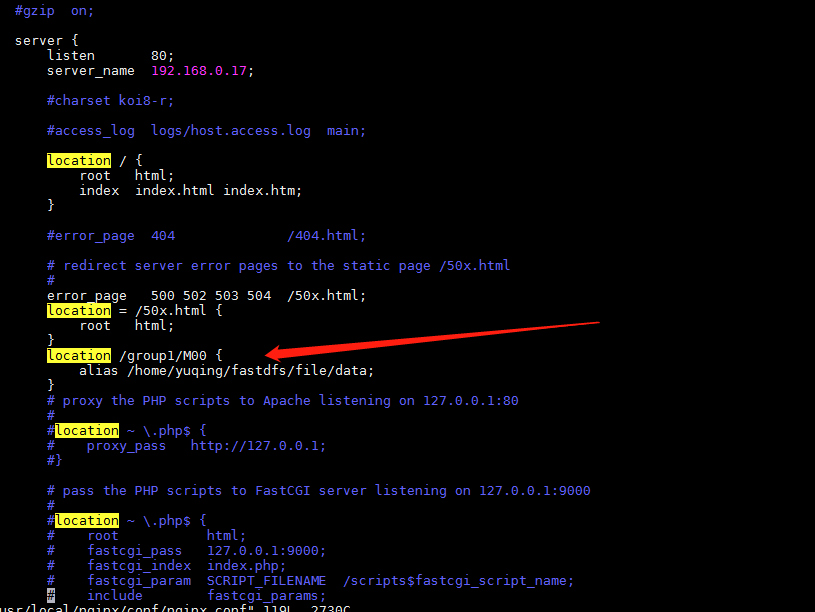
修改配置，其它的默认

在80端口下添加fastdfs-nginx模块

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

ngx\_fastdfs\_module;

}



在/home/yuqing/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 /home/yuqing/fastdfs/file/data/ /home/yuqing/fastdfs/file/data/M00

⑨ 启动nginx

# /usr/local/nginx/sbin/nginx

在地址栏访问。

**可能出现的问题**

如果不是root 用户 你必须在除了cd的命令之外 全部加sudo

如果不是root 用户 编译和安装分开进行 先编译再安装

如果上传成功 但是nginx报错404 先检查mod\_fastdfs.conf文件中的store\_path0是否一致

如果nginx无法访问 先检查防火墙 和 mod\_fastdfs.conf文件tracker\_server是否一致

参考文档：https://github.com/happyfish100/fastdfs/wiki