[1、FastDFS介绍](#_Toc6062)

**FastDFS分布式文件系统**

# 1、FastDFS介绍

## 1.1、什么是FastDFS

FastDFS是用c语言编写的一款开源的分布式文件系统。FastDFS为互联网量身定制，充分考虑了冗余备份、负载均衡、线性扩容等机制，并注重高可用、高性能等指标，使用FastDFS很容易搭建一套高性能的文件服务器集群提供文件上传、下载等服务。由跟踪服务器（tracker server）、存储服务器（storage server）和客户端（client）组成。它主要解决的是海量数据存储的问题。

## 1.2、FastDFS的系统架构

FastDFS架构包括 Tracker server和Storage server。客户端请求Tracker server进行文件上传、下载，通过Tracker server调度最终由Storage server完成文件上传和下载。

Tracker server作用是负载均衡和调度，通过Tracker server在文件上传时可以根据一些策略找到Storage server提供文件上传服务。可以将tracker称为追踪服务器或调度服务器。

Storage server作用是文件存储，客户端上传的文件最终存储在Storage服务器上，Storage server没有实现自己的文件系统而是利用操作系统 的文件系统来管理文件。可以将storage称为存储服务器。



### 1.2.1、Tracker集群

FastDFS集群中的Tracker server可以有多台，Tracker server之间是相互平等关系同时提供服务，Tracker server不存在单点故障。客户端请求Tracker server采用轮询方式，如果请求的tracker无法提供服务则换另一个tracker。

### 1.2.2、Storage集群

Storage集群采用了分组存储方式。storage集群由一个或多个组构成，集群存储总容量为集群中所有组的存储容量之和。一个组由一台或多台存储服务器组成，组内的Storage server之间是平等关系，不同组的Storage server之间不会相互通信，同组内的Storage server之间会相互连接进行文件同步，从而保证同组内每个storage上的文件完全一致的。一个组的存储容量为该组内存储服务器容量最小的那个，由此可见组内存储服务器的软硬件配置最好是一致的。

采用分组存储方式的好处是灵活、可控性较强。比如上传文件时，可以由客户端直接指定上传到的组也可以由tracker进行调度选择。一个分组的存储服务器访问压力较大时，可以在该组增加存储服务器来扩充服务能力（纵向扩容）。当系统容量不足时，可以增加组来扩充存储容量（横向扩容）。

### 1.2.3、Storage的状态收集

Storage server会连接集群中所有的Tracker server，定时向他们报告自己的状态，包括磁盘剩余空间、文件同步状况、文件上传下载次数等统计信息。

### 1.2.4、文件上传流程

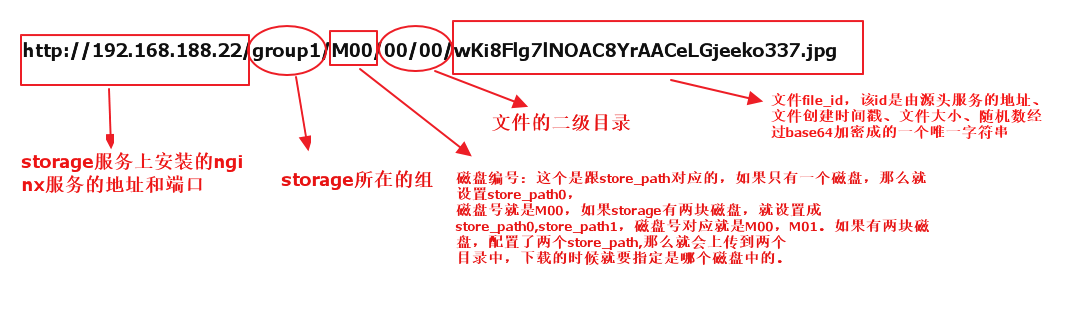


客户端上传文件后存储服务器将文件ID返回给客户端，此文件ID用于以后访问该文件的索引信息。文件索引信息包括：组名，磁盘编号，数据两级目录，文件名。这里的文件名是由文件上传的源头服务器、文件上传的时间戳、文件大小、随机数经过base64加密后的一个串。

* 组名：文件上传后所在的storage组名称，在文件上传成功后有storage服务器返回，需要客户端自行保存。
* 磁盘编号：storage配置的上传路store\_path\*与磁盘编号对应。如果配置了store\_path0则是M00，如果配置了store\_path1则是M01，以此类推。

因为FastDFS支持多次盘上传，也就是说如果你的服务器有多块儿挂在磁盘，那么可以配置多个Store\_path，如store\_path0、store\_path1..对应的磁盘编号就是M00、M01，这样在上传的时候会上传到多块磁盘上。

* 数据两级目录：storage服务器在每个虚拟磁盘路径下创建的两级目录，用于存储数据文件。
* 文件名：与文件上传时不同。是由存储服务器根据特定信息生成，文件名包含：源存储服务器IP地址、文件创建时间戳、文件大小、随机数和文件拓展名等信息。



### 1.2.5、文件下载流程



tracker根据请求的文件路径即文件ID 来快速定位文件。

比如请求下边的文件：



1.通过组名tracker能够很快的定位到客户端需要访问的存储服务器组是group1，并选择合适的存储服务器提供客户端访问。

2.选择完一台存储服务器后，存储服务器根据“磁盘编号M00”可以得知使用的是哪块磁盘是M00这块磁盘。由于M00对应的是store\_path0,所以根据store\_path0配置的路径就可以找到对应的目录。

3.根据两级目录/02/44/和文件名就可以获取到对应的文件了。

注意：由于FastDFS只提供了文件上传的服务并没有提供文件下载的接口，所以需要在storage上安装nginx来实现文件下载。

# 2、FastDFS+Nginx实现文件服务器

## 2.1、架构

### 2.1.1、架构图

Storage集群

Tracker Server 集群

tracker 1

192.168.101.3

部署程序：

FastDFS

nginx

tracker 2

192.168.101.4

部署程序：

FastDFS

nginx

Storage Server Group1

Storage Server Group2

storage 1

192.168.101.5

部署程序：

FastDFS

nginx

storage 2

192.168.101.6

部署程序：

FastDFS

nginx

storage 3

192.168.101.7

部署程序：

FastDFS

nginx

storage 4

192.168.101.8

部署程序：

FastDFS

nginx

Client

client API

http

tcp/http

### 2.1.2、架构描述

#### 2.1.2.1 Tracker Server

每个tracker server互相平等，tracker server上部署nginx不是为了对外提供http文件下载接口，tracker上nginx只是起到负载均衡、反向代理的作用。tracker的nginx会反向代理转发至storage上的nginx。

#### 2.1.2.2 Storage Server

每台storage上也部署nginx，storage上的nginx与tracker上的nginx有区别，storage上的nginx需要安装FastDSF-nginx模块，此模块的作用是使用FastDFS和nginx进行整合，nginx对外提供http文件下载接口，注意：nginx只提供文件下载接口不提供上传接口。文件上传仍然通过java client API进行。

## 2.2、Tracker Server安装

### 2.2.1、下载FastDFS

tracker和storage使用相同的安装包，下载地址：http://sourceforge.net/projects/FastDFS/ 或https://github.com/happyfish100/FastDFS（推荐）

### 2.2.2、Tracker安装环境准备

#### 2.2.2.1、安装C语言编译器gcc

FastDFS是C语言开发，建议在linux上运行，本教程使用Centos6.5作为安装环境。安装FastDFS需要先将官网下载的源码进行编译，编译依赖gcc环境，如果没有gcc环境，需要安装gcc：yum install gcc-c++

#### 2.2.2.2、安装libevent

FastDFS依赖libevent库，需要安装

#### 2.2.2.3、安装libfastcommon

libfastcommon是FastDFS官方提供的，libfastcommon包含了FastDFS运行所需要的一些基础库。

将libfastcommonV1.0.7.tar.gz拷贝至/usr/local/src/下

cd /usr/local/src

tar -zxvf libfastcommonV1.0.7.tar.gz –C /usr/local/

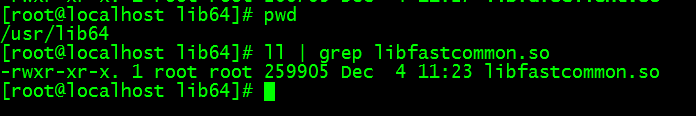
cd libfastcommon-1.0.7

./make.sh

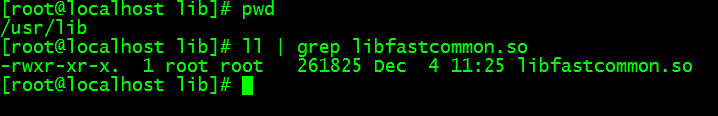
./make.sh install

注意：libfastcommon安装好后会自动将库文件拷贝至/usr/lib64下，由于FastDFS程序引用usr/lib目录所以需要将/usr/lib64下的库文件拷贝至/usr/lib下。

要拷贝的文件如下：

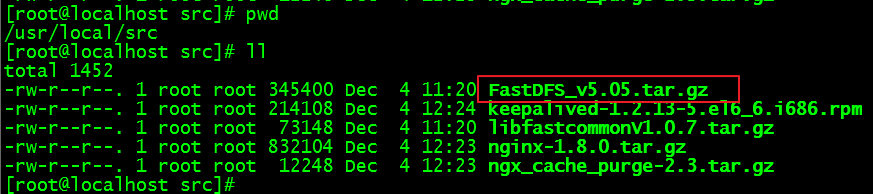


cp /usr/lib64/libfastcommon.so /usr/lib/



### 2.2.3 编译安装Tracker

将FastDFS\_v5.05.tar.gz拷贝至/usr/local/src/下



解压tar包

tar -zxvf FastDFS\_v5.05.tar.gz –C /usr/local/

解压后再/usr/local目录下会生成一个FastDFS文件夹，然后进入这个目录，编译，安装。

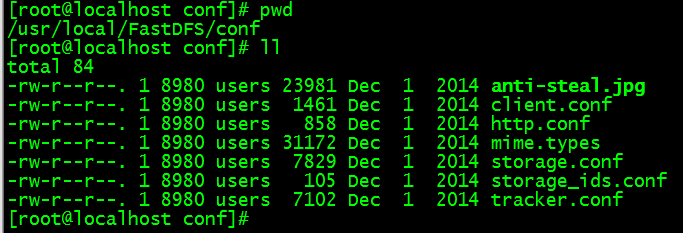
cd /usr/local/FastDFS

./make.sh

./make.sh install

这里采用的是默认的安装，配置文件都存储在/etc/fdfs/，命令文件都安装在/usr/bin/下了。

安装成功将安装目录下的conf下的文件拷贝到/etc/fdfs/下。/etc/fdfs/目录是FastDFS配置文件目录。



cp ./\* /etc/fdfs/

进入/etc/fdfs/目录下，修改复制

拷贝一份新的tracker配置文件：

cp tracker.conf.sample ./tracker.conf （文件拷贝，如果目标文件加了文件名，那么就是拷贝并重命名。）

修改内容：

Tracker默认的服务端口是22122，这个不用修改。

base\_path=/home/FastDFS

#/home/FastDFS 这个目录是需要先创建好的，该目录是tracker的数据文件目录，用来存储storage的信息。

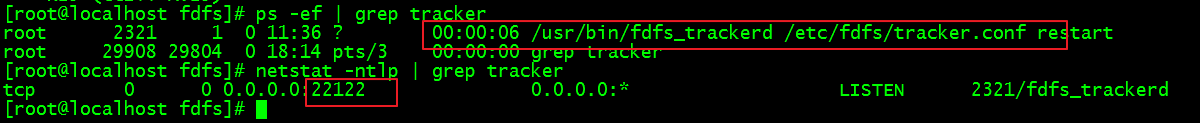
http.server\_port=8088

#Http服务端口修改成8088，这个可以任意修改。

### 2.2.4 启动tracker

/usr/bin/fdfs\_trackerd /etc/fdfs/tracker.conf restart

或者/usr/bin/fdfs\_trackerd /etc/fdfs/tracker.conf



## 2.3 Storage Server安装

### 2.3.1 Storage安装环境准备

跟Tracker Server一样。

### 2.3.2 编译安装Storage

在另外一台机器上安装Storage，这才是分布式。Storage需要安装两台，也就是一个group包含两个storage Server。

编译安装同Tracker一样，需要解压tar包，然后make，make install。

安装成功将安装目录下的conf下的文件拷贝到/etc/fdfs/下。

cp storage.conf.sample storage.conf

修改storage.conf

vi storage.conf

group\_name=group1

base\_path=/home/yuqing/FastDFS改为：base\_path=/home/FastDFS

store\_path0=/home/yuqing/FastDFS改为：store\_path0=/home/FastDFS/fdfs\_storage

#如果有多个挂载磁盘则定义多个store\_path，如下

#store\_path1=.....

#store\_path2=......  
tracker\_server=192.168.188.21:22122 #配置tracker服务器:IP

#如果有多个则配置多个tracker

#配置http端口

http.server\_port=8088

创建目录/home/FastDFS/fdfs\_storage

Storage中配置tracker的地址，storage是主动连接tracker的。

### 2.3.3 启动Storage

/usr/bin/fdfs\_storaged /etc/fdfs/storage.conf restart



## 2.4 测试上传

安装成功后可以通过/usr/bin/fdfs\_test测试上传

在tracker机器上或者是storage机器上修改/etc/fdfs/client.conf配置文件

base\_path=/home/fastdfs

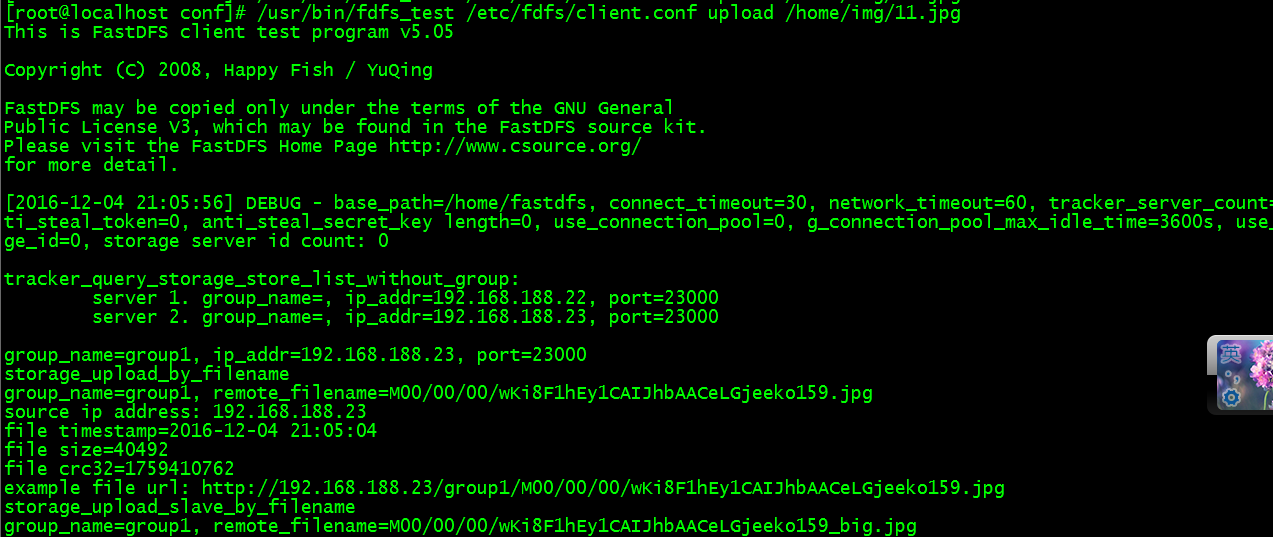
tracker\_server=192.168.188.21:22122

在home下创建fastdfs目录

使用格式：

/usr/bin/fdfs\_test /etc/fdfs/client.conf upload /home/img/11.png

出现下图则上传成功了：



目前还不能下载，因为fastDFS没有提供下载接口，需要使用nginx提供下载接口。Tracker和Storage都需要安装nginx。

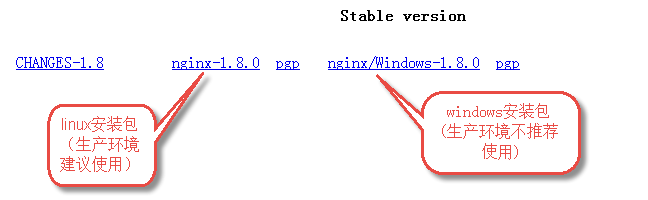
## 2.5 在Storage上安装nginx

### 2.5.1 Storage上安装nginx的目的

目的是提供http下载接口，通过访问nginx的服务，来下载文件。

### 2.5.2 nginx安装前准备

1、进入http://nginx.org/en/download.html 下载nginx1.8.0版本（当前最新稳定版本）。





2、安装nginx的依赖包

nginx是C语言开发，建议在linux上运行，本教程使用Centos6.5作为安装环境。

* gcc

安装nginx需要先将官网下载的源码进行编译，编译依赖gcc环境，如果没有gcc环境，需要安装gcc：yum install gcc-c++

Yum是通过网络yum源来下载rpm包，linux提供了一个官方的网络rpm服务器，供用户下载安装rpm包，yum –y 是自动安装依赖。

* PCRE

PCRE(Perl Compatible Regular Expressions)是一个Perl库，包括 perl 兼容的正则表达式库。nginx的http模块使用pcre来解析正则表达式，所以需要在linux上安装pcre库。

**yum install -y pcre pcre-devel**

注：pcre-devel是使用pcre开发的一个二次开发库。nginx也需要此库。

* zlib

zlib库提供了很多种压缩和解压缩的方式，nginx使用zlib对http包的内容进行gzip，所以需要在linux上安装zlib库。

**yum install -y zlib zlib-devel**

* openssl

OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及SSL协议，并提供丰富的应用程序供测试或其它目的使用。

nginx不仅支持http协议，还支持https（即在ssl协议上传输http），所以需要在linux安装openssl库。

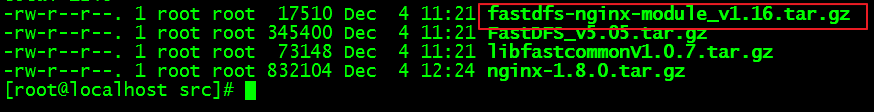
**yum install -y openssl openssl-devel**

### 2.5.3 安装FastDFS-nginx-module

FastDFS-nginx-module是FastDFS和Nginx整合的插件，因为nginx需要给Storage提供下载接口，所以在Storage上需要安装，Tracker上不需要安装，因为Tracker不需要提供下载接口。

FastDFS-nginx-module.tar.gz也是源码包，随着nginx一起编译，安装，配置在nginx的安装配置里就可以了。

将FastDFS-nginx-module\_v1.16.tar.gz传至/usr/local/src下



cd /usr/local/src/

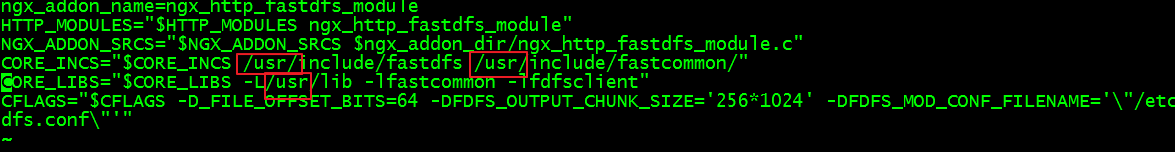
tar -zxvf FastDFS-nginx-module\_v1.16.tar.gz –C /usr/local/ 解压在当前目录也可以的。

cd /usr/local/fastdfs-nginx-module/src

修改config文件将/usr/local/路径改为/usr/

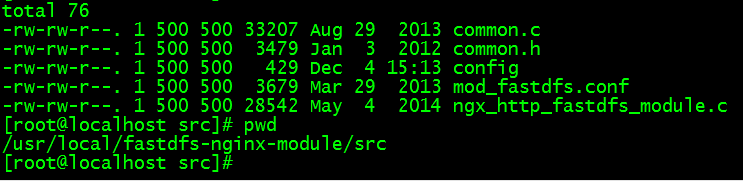
如/usr/include/fastdfs

默认是/usr/local/include/fastdfs



将FastDFS-nginx-module/src下的mod\_FastDFS.conf拷贝至/etc/fdfs/下

进入到fastdfs-nginx-module/src



cp mod\_FastDFS.conf /etc/fdfs/

并修改mod\_FastDFS.conf的内容：

vi /etc/fdfs/mod\_FastDFS.conf

base\_path=/home/FastDFS

tracker\_server=192.168.188.21:22122

url\_have\_group\_name=true #url中包含group名称

store\_path0=/home/FastDFS/fdfs\_storage #指定文件存储路径

#如果有多个

将libfdfsclient.so拷贝至/usr/lib下

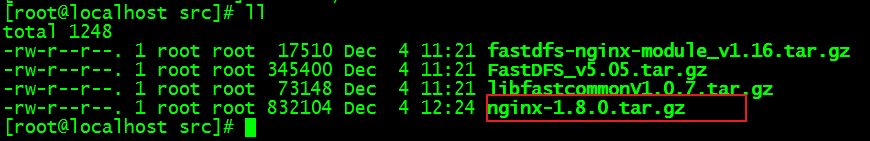
cp /usr/lib64/libfdfsclient.so /usr/lib/

创建nginx/client目录

mkdir -p /var/temp/nginx/client

### 2.5.4 安装nginx

1、将nginx-1.8.0.tar.gz（源码包）拷贝至/usr/local/src/下



2、解压nginx-1.8.0.tar.gz 到/usr/local/下(也可以解压到src目录下)

tar -zxvf nginx-1.8.0.tar.gz –C /usr/local/

cd /usr/local/nginx-1.8.0

* configure 安装前的参数配置，配置安装目录和命令文件和配置文件的位置。

./configure --help查询详细参数（参考本教程附录部分：nginx编译参数）

参数设置如下：设置安装前的设置如安装的位置，添加依赖包

./configure \

--prefix=/usr/local/nginx \

--pid-path=/var/run/nginx/nginx.pid \

--lock-path=/var/lock/nginx.lock \

--error-log-path=/var/log/nginx/error.log \

--http-log-path=/var/log/nginx/access.log \

--with-http\_gzip\_static\_module \

--http-client-body-temp-path=/var/temp/nginx/client \

--http-proxy-temp-path=/var/temp/nginx/proxy \

--http-fastcgi-temp-path=/var/temp/nginx/fastcgi \

--http-uwsgi-temp-path=/var/temp/nginx/uwsgi \

--http-scgi-temp-path=/var/temp/nginx/scgi \

--add-module=/usr/local/fastdfs-nginx-module/src

**注意：上边将临时文件目录指定为/var/temp/nginx，需要在/var下创建temp及nginx目录**

mkdir -p /var/temp/nginx

**解释：**

--prefix=/usr/local/nginx \ 是nginx的安装目录

--add-module=/usr/local/fastdfs-nginx-module/src 是添加依赖模块

编译安装

make

make install

### 2.5.5 修改nginx的配置文件

修改/usr/local/nginx安装目录下的conf下的nginx.conf配置文件

在server模块儿中修改：

listen 80;

server\_name localhost;

添加：

#nginx提供下载接口，用户请求nginx的服务，nginx通过配置的location

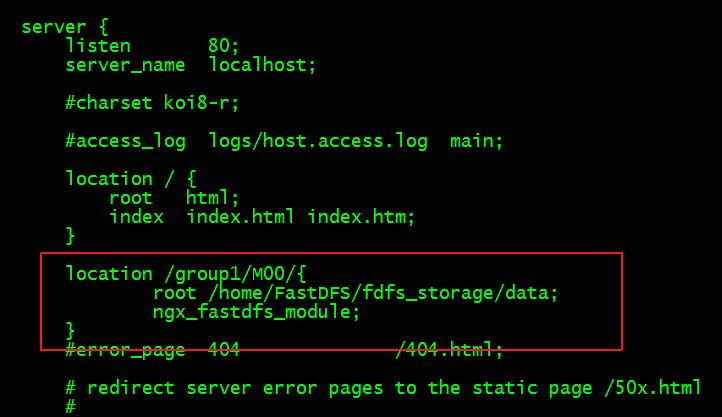
#找到对应的文件目录。

location /group1/M00/{

root /home/FastDFS/fdfs\_storage/data;

ngx\_fastdfs\_module;

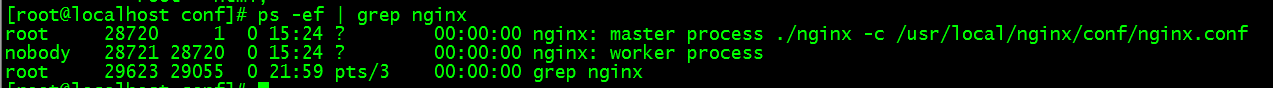
}



### 2.5.6 启动nginx

cd /usr/local/nginx/sbin/

./nginx -c /usr/local/nginx/conf/nginx.conf



出现上图nginx启动成功。

## 2.6 Tracker上安装nginx

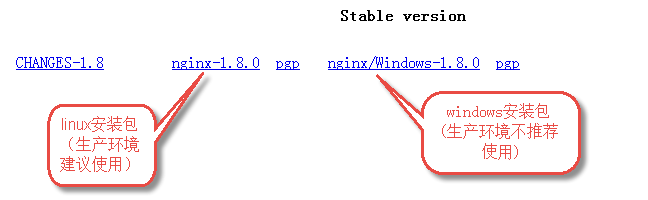
### 2.6.1 Tracker上安装nginx的目的

1、为了做反向代理，请求tracker上的nginx服务，反向代理选择出对应的group组，再根据tracker接收的storage的同步信息情况，最终决定一台storage服务器。然后给客户端返回storage的ip和端口。

2、做tracker的负载均衡。

### 2.6.2 nginx安装前准备

1、进入http://nginx.org/en/download.html 下载nginx1.8.0版本（当前最新稳定版本）。





2、安装nginx的依赖包

nginx是C语言开发，建议在linux上运行，本教程使用Centos6.5作为安装环境。

* gcc

安装nginx需要先将官网下载的源码进行编译，编译依赖gcc环境，如果没有gcc环境，需要安装gcc：yum install gcc-c++

Yum是通过网络yum源来下载rpm包，linux提供了一个官方的网络rpm服务器，供用户下载安装rpm包，yum –y 是自动安装依赖。

* PCRE

PCRE(Perl Compatible Regular Expressions)是一个Perl库，包括 perl 兼容的正则表达式库。nginx的http模块使用pcre来解析正则表达式，所以需要在linux上安装pcre库。

**yum install -y pcre pcre-devel**

注：pcre-devel是使用pcre开发的一个二次开发库。nginx也需要此库。

* zlib

zlib库提供了很多种压缩和解压缩的方式，nginx使用zlib对http包的内容进行gzip，所以需要在linux上安装zlib库。

**yum install -y zlib zlib-devel**

* openssl

OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及SSL协议，并提供丰富的应用程序供测试或其它目的使用。

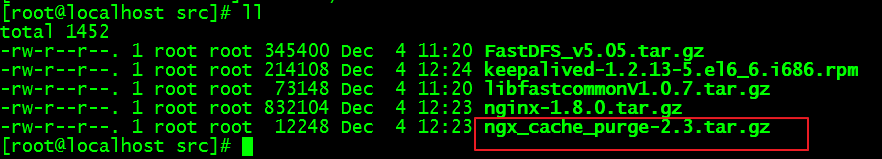
nginx不仅支持http协议，还支持https（即在ssl协议上传输http），所以需要在linux安装openssl库。

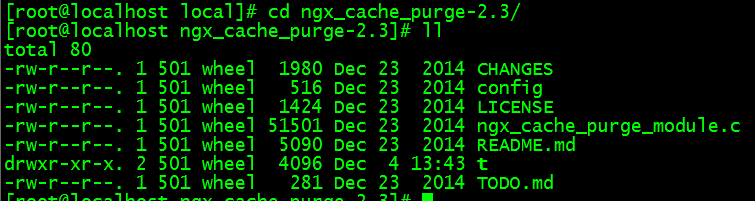
**yum install -y openssl openssl-devel**

### 2.6.3 安装nginx

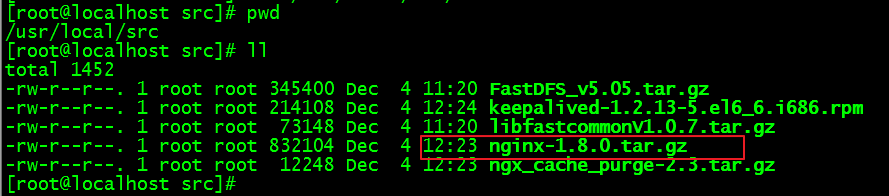
1、安装ngx\_cache\_purge-2.3.tar.gz插件，这个是nginx缓存插件。（可以不安装）

拷贝ngx\_cache\_purge-2.3.tar.gz到/usr/local/src然后解压到/usr/local/下





2、将nginx-1.8.0.tar.gz（源码包）拷贝至/usr/local/src/下



3、解压nginx-1.8.0.tar.gz 到/usr/local/下(也可以解压到src目录下)

tar -zxvf nginx-1.8.0.tar.gz –C /usr/local/

cd /usr/local/nginx-1.8.0

* configure 安装前的参数配置，配置安装目录和命令文件和配置文件的位置。

./configure --help查询详细参数（参考本教程附录部分：nginx编译参数）

参数设置如下：设置安装前的设置如安装的位置，添加依赖包

./configure \

--prefix=/usr/local/nginx \

--pid-path=/var/run/nginx/nginx.pid \

--lock-path=/var/lock/nginx.lock \

--error-log-path=/var/log/nginx/error.log \

--http-log-path=/var/log/nginx/access.log \

--with-http\_gzip\_static\_module \

--http-client-body-temp-path=/var/temp/nginx/client \

--http-proxy-temp-path=/var/temp/nginx/proxy \

--http-fastcgi-temp-path=/var/temp/nginx/fastcgi \

--http-uwsgi-temp-path=/var/temp/nginx/uwsgi \

--http-scgi-temp-path=/var/temp/nginx/scgi \

--add-module=/usr/local/ngx\_cache\_purge-2.3

**注意：上边将临时文件目录指定为/var/temp/nginx，需要在/var下创建temp及nginx目录**

mkdir -p /var/temp/nginx

**解释：**

--prefix=/usr/local/nginx \ 是nginx的安装目录

--add-module=/usr/local/ngx\_cache\_purge-2.3 是添加依赖

编译安装

make

make install

进入/usr/local/nginx/conf修改nginx.conf配置文件

在http模块儿添加

upstream storage\_server\_group1{

#192.168.188.22:80实际上是storage server上nginx的ip和端口，配置这

#个是为了做下载的反向代理，nginx提供了下载接口

#所以这个是nginx的ip和端口

server 192.168.188.22:80 weight=10;

#这个80端口不是storage的port也不是http.server\_port，而是storage上#nginx的服务

#端口，因为我们通过浏览器访问图片其实访问的是nginx提供的接口，其实访问的是nginx的服务

server 192.168.188.23:80 weight=10;

}

如果有group2的话还可以配置

upstream storage\_server\_group2{

server 192.168.101.7:80 weight=10;

server 192.168.101.8:80 weight=10;

}

在server模块儿中添加：

listen 80;

server\_name localhost;

location /group1{

proxy\_redirect off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_pass http://storage\_server\_group1;#做反向代理，访问tracker，将访问的资源反向代理到对应的storage对应的nginx上。

}

location /group2{

proxy\_redirect off;

proxy\_set\_header Host $host;

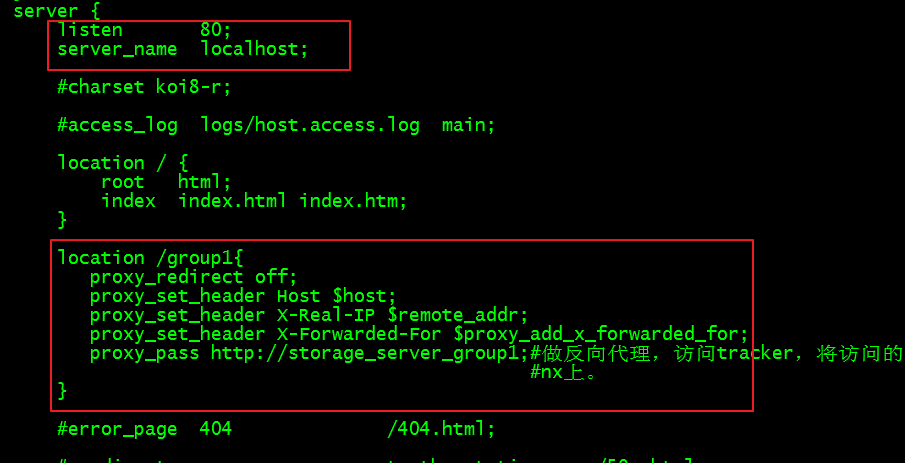
proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

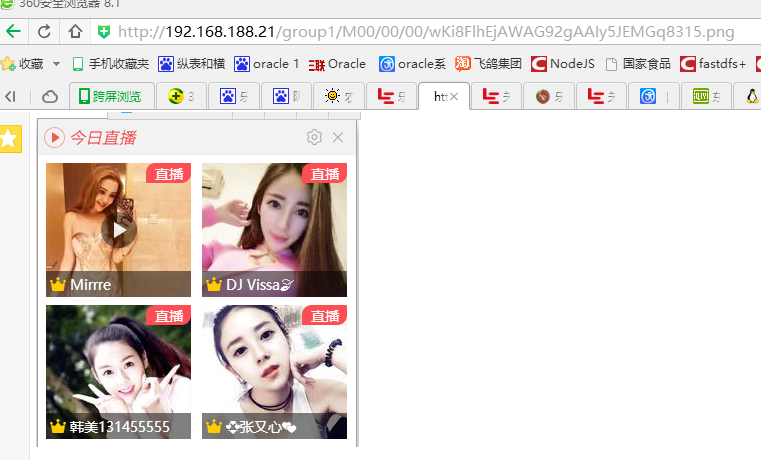
proxy\_pass http://storage\_server\_group2;

}





测试：



这里访问地址：

<http://192.168.188.21/group1/M00/00/00/wKi8FlhEjAWAG92gAAIy5JEMGq8315.png>

请求Tracker服务器下载图片，其实请求的是Tracker服务器上的nginx服务，因为nginx服务的端口是80端口，而tracker服务端口是22122。

tracker上的nginx起到http访问的反向代理作用，反向代理到group对应的所有storage上的nginx服务。客户端请求tracker上的nginx服务，根据配置的location /group1找到对应的group1下所有的storage服务器上的nginx服务的ip和端口（提供下载接口），然后由tracker决定使用哪台storage，最后将该storage的ngnix服务的ip和port返回给客户端。客户端拿到storage的nginx服务的ip和端口，去请求这个nginx服务，根据请求中的group1/M00与storage的nginx配置的location /group1/M00/{

root /home/FastDFS/fdfs\_storage/data;

ngx\_fastdfs\_module;

}

进行匹配，然后获取到文件的存储目录，然后在根据两级目录和文件名来获取到文件。

为什么非得要用nginx呢，请求中不是已经指定group了吗，tracker就可以定位到对应的组了？

虽然tracker理论上是可以根据请求的group定位到对应的storage Server，但是却不能获取到storage上的nginx服务，因为nginx提供了下载接口。所以只能通过tracker上的nginx做http访问的反向代理，根据group 代理到group对应的storage上的nginx服务，然后根据tracker中同步的信息决定选择哪台，最后将storage上的nginx服务返给客户端。