## Nginx

### 什么是nginx

是一个使用c语言开发的高性能的http服务器及反向代理服务器。

Nginx是一款高性能的http 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器。由俄罗斯的程序设计师Igor Sysoev所开发，官方测试nginx能够支支撑5万并发链接，并且cpu、内存等资源消耗却非常低，运行非常稳定。

### Nginx的应用场景

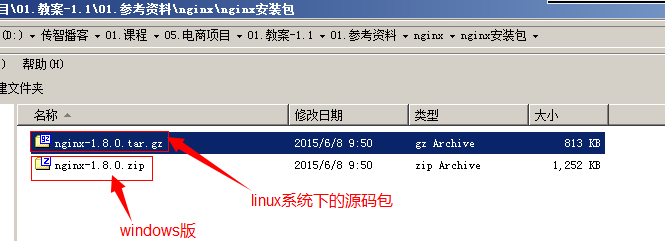
1. http服务器。Nginx是一个http服务可以独立提供http服务。可以做网页静态服务器。
2. 虚拟主机。可以实现在一台服务器虚拟出多个网站。例如个人网站使用的虚拟主机。
3. 反向代理，负载均衡。当网站的访问量达到一定程度后，单台服务器不能满足用户的请求时，需要用多台服务器集群可以使用nginx做反向代理。并且多台服务器可以平均分担负载，不会因为某台服务器负载高宕机而某台服务器闲置的情况。

### Nginx的安装

Nginx一般推荐安装到linux系统，而且要安装c语言的编译环境gcc。

#### 下载：

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



#### 先安装nginx依赖的包：

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++

* 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 pcrepcre-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 zlibzlib-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 opensslopenssl-devel**

#### 安装步骤

第一步：把nginx的源码上传到linux系统

第二步：把压缩包解压缩。

第三步：进行configure。

./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

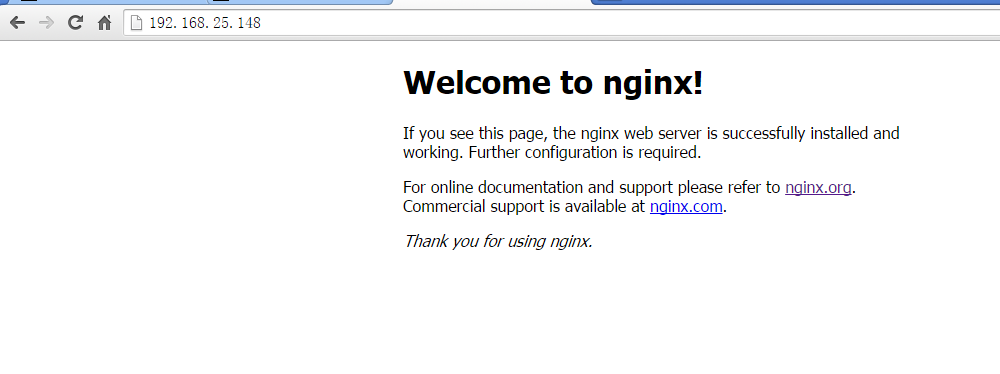
**注意：上边将临时文件目录指定为/var/temp/nginx，需要在/var下创建temp及nginx目录**

第四步：make

第五步：make install

### Nginx的启动、停止

1、启动：进入nginx的sbin目录，./nginx就可以启动。



如果访问不到，首先查看防火墙是否关闭。

2、关闭nginx：

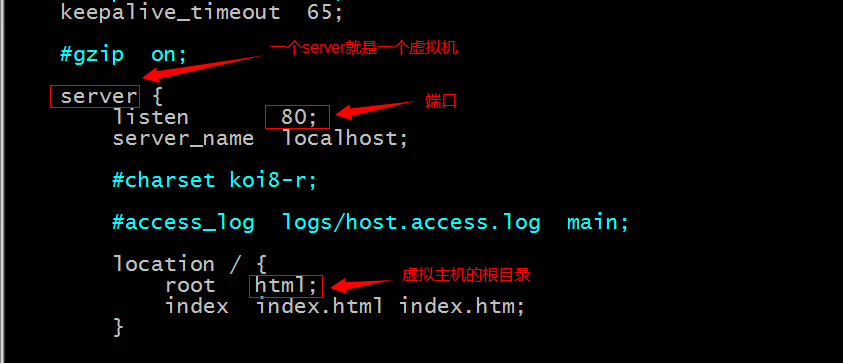
可以使用kill命令，但是不推荐使用。

推荐使用：./nginx -s stop

1. 刷新配置：./nginx -s reload

### Nginx的配置

在/usr/local/nginx/conf目录下nginx.conf文件是nginx的配置文件。



### 使用nginx配置虚拟机

#### 通过端口区分虚拟机

在nginx.conf文件中添加一个Service节点，修改端口号就可以

|  |
| --- |
| server {  listen 81;  server\_name localhost;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  root html81;  index index.html index.htm;  }  } |

#### 通过域名区分虚拟机

##### 域名介绍

DNS服务器

通过ip地址访问服务器

返回ip地址

Pc机

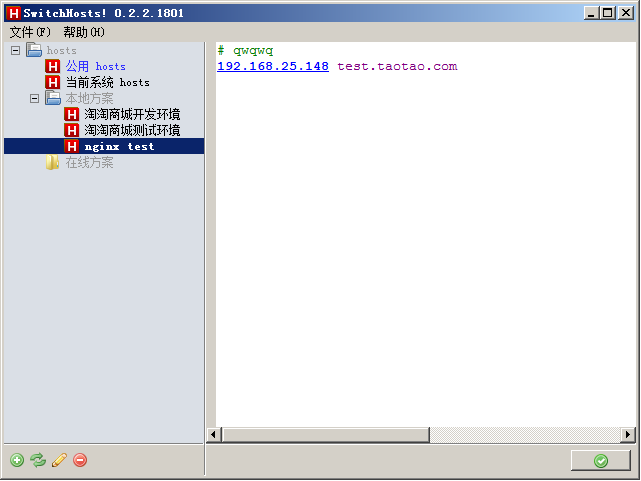
Web服务器

根据域名换ip地址

可以通过修改host文件指定域名的ip地址。

Host文件的位置：C:\Windows\System32\drivers\etc

可以使用工具：



##### 配置基于域名的虚拟主机

需要修改nginx.conf配置文件。

|  |
| --- |
| server {  listen 80;  server\_name test3.taotao.com;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  root html-test3;  index index.html index.htm;  }  } |

修改配置后需要重新加载配置文件。

# Nginx的反向代理

## 什么是反向代理

正向代理

Pc机

Pc机

Pc机

代理服务器

反向代理：

Pc机

反向代理服务器

[www.taobao.com](http://www.taobao.com)

功能就是请求的转发

Nginx实现

淘宝首页

服务器

淘宝首页

服务器

淘宝首页

服务器

反向代理服务器是引用在服务端。决定哪台服务器提供服务。

## 反向代理的模拟

### 反向代理

应该有一个nginx服务器有多个应用服务器（可以是tomcat）

可以使用一台虚拟机，安装一个nginx，多个tomcat，来模拟。

Nginx的配置文件：

|  |
| --- |
| upstream tomcats{  server 192.168.25.148:8080;  server 192.168.25.148:8081;  }  server {  listen 80;  server\_name tomcat.taotao.com;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  proxy\_pass http://tomcats;  index index.html index.htm;  }  } |

### 负载均衡

只需要在upstream的server后面添加一个weight即可代表权重。权重越高，分配请求的数量就越多。默认权重是1



# 图片服务器的搭建

## 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。



## 文件上传流程



## 文件下载流程



## 上传文件的文件名

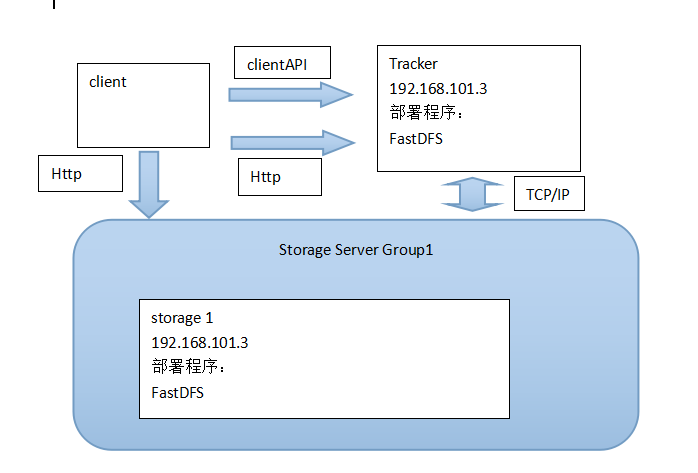
客户端上传文件后存储服务器将文件ID返回给客户端，此文件ID用于以后访问该文件的索引信息。文件索引信息包括：组名，虚拟磁盘路径，数据两级目录，文件名。



* 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。
* 虚拟磁盘路径：storage配置的虚拟路径，与磁盘选项store\_path\*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。
* 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。

## FastDFS搭建



可以使用一台虚拟机来模拟，只有一个Tracker、一个Storage服务。

配置nginx访问图片。

### 搭建步骤

第一步：把fastDFS都上传到linux系统。

第二步：安装FastDFS之前，先安装libevent工具包。

yum -y install libevent

第三步：安装libfastcommonV1.0.7工具包。

1. 解压缩
2. ./make.sh
3. ./make.sh install
4. 把/usr/lib64/libfastcommon.so文件向/usr/lib/下复制一份

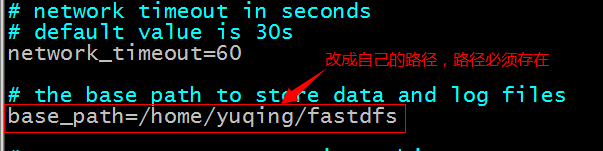
第四步：安装Tracker服务。

1. 解压缩
2. ./make.sh
3. ./make.sh install

安装后在/usr/bin/目录下有以fdfs开头的文件都是编译出来的。

配置文件都放到/etc/fdfs文件夹

1. 把/root/FastDFS/conf目录下的所有的配置文件都复制到/etc/fdfs下。
2. 配置tracker服务。修改/root/FastDFS/conf/tracker.conf文件。

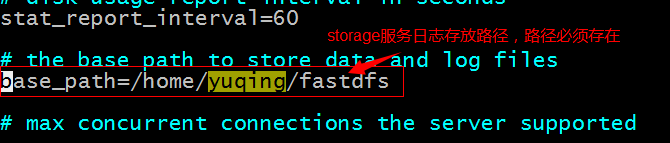


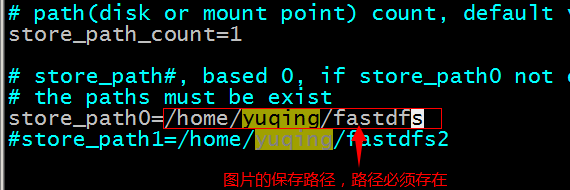
1. 启动tracker。/usr/bin/fdfs\_trackerd /etc/fdfs/tracker.conf

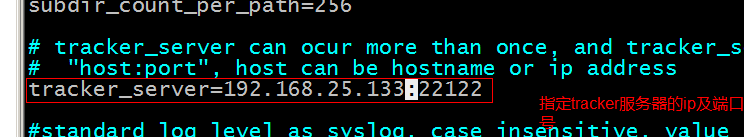
重启使用命令：/usr/bin/fdfs\_trackerd /etc/fdfs/tracker.confrestart

第五步：安装storage服务。

1. 如果是在不同的服务器安装，第四步的1~4需要重新执行。
2. 配置storage服务。修改/root/FastDFS/conf/storage.conf文件





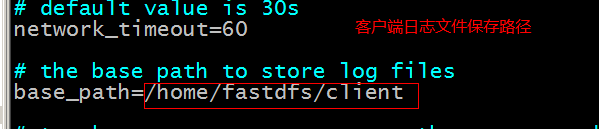


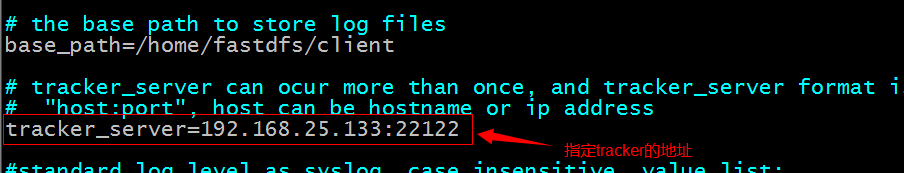
1. 启动storage服务。

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

第六步：测试服务。

1. 修改配置文件/etc/fdfs/client.conf





1. 测试

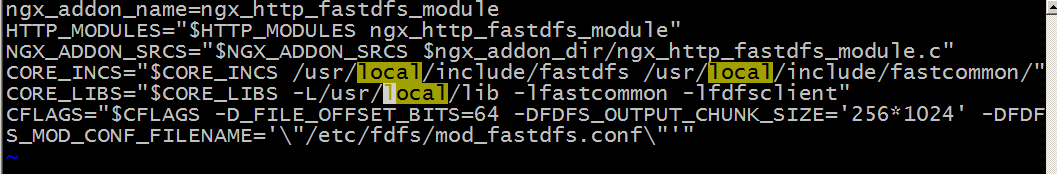
/usr/bin/fdfs\_test /etc/fdfs/client.conf upload anti-steal.jpg

第七步：搭建nginx提供http服务。

可以使用官方提供的nginx插件。要使用nginx插件需要重新编译。

fastdfs-nginx-module\_v1.16.tar.gz

1. 解压插件压缩包
2. 修改/root/fastdfs-nginx-module/src/config文件，把其中的local去掉。



1. 对nginx重新config

./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 \

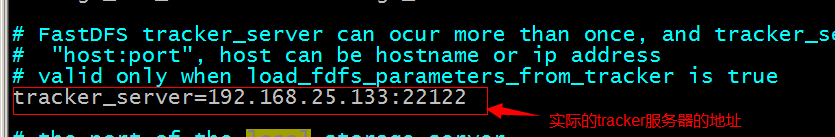
--http-uwsgi-temp-path=/var/temp/nginx/uwsgi \

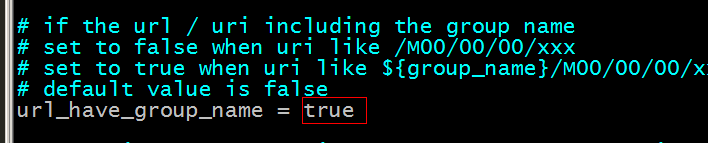
--http-scgi-temp-path=/var/temp/nginx/scgi \

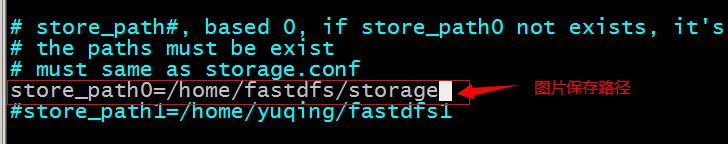
--add-module=/root/fastdfs-nginx-module/src

1. make
2. make install
3. 把/root/fastdfs-nginx-module/src/mod\_fastdfs.conf文件复制到/etc/fdfs目录下。编辑：









1. nginx的配置

在nginx的配置文件中添加一个Server：

server {

listen 80;

server\_name192.168.101.3;

location /group1/M00/{

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

ngx\_fastdfs\_module;

}

}

1. 将libfdfsclient.so拷贝至/usr/lib下

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

1. 启动nginx

## 图片服务的使用

官方提供一个jar包

使用方法：

1. 把FastDFS提供的jar包添加到工程中
2. 初始化全局配置。加载一个配置文件。
3. 创建一个TrackerClient对象。
4. 创建一个TrackerServer对象。
5. 声明一个StorageServer对象，null。
6. 获得StorageClient对象。
7. 直接调用StorageClient对象方法上传文件即可。

|  |
| --- |
| @Test  **publicvoid**testUpload() **throws** Exception {  // 1、把FastDFS提供的jar包添加到工程中  // 2、初始化全局配置。加载一个配置文件。  ClientGlobal.*init*("D:\\workspaces-itcast\\JaveEE18\\taotao-manager\\taotao-manager-web\\src\\main\\resources\\properties\\client.conf");  // 3、创建一个TrackerClient对象。  TrackerClienttrackerClient = **new**TrackerClient();  // 4、创建一个TrackerServer对象。  TrackerServertrackerServer = trackerClient.getConnection();  // 5、声明一个StorageServer对象，null。  StorageServerstorageServer = **null**;  // 6、获得StorageClient对象。  StorageClientstorageClient = **new**StorageClient(trackerServer, storageServer);  // 7、直接调用StorageClient对象方法上传文件即可。  String[] strings = storageClient.upload\_file("D:\\Documents\\Pictures\\images\\2f2eb938943d.jpg", "jpg", **null**);  **for** (String string :strings) {  System.***out***.println(string);  }  } |

Client.conf

|  |
| --- |
| tracker\_server=192.168.25.133:22122 |

## 使用工具类上传图片

|  |
| --- |
| @Test  **publicvoid**testFastDfsClient() **throws** Exception {  FastDFSClientclient = **new**FastDFSClient("D:\\workspaces-itcast\\JaveEE18\\taotao-manager\\taotao-manager-web\\src\\main\\resources\\properties\\client.conf");  String uploadFile = client.uploadFile("D:\\Documents\\Pictures\\images\\200811281555127886.jpg", "jpg");  System.***out***.println(uploadFile);  } |