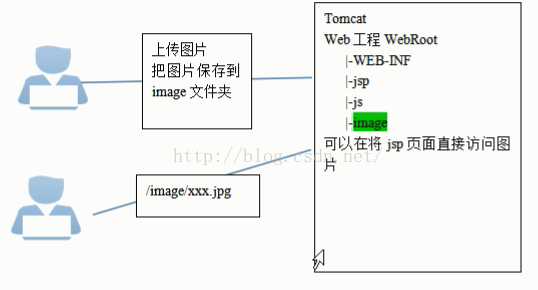
图片服务器的搭建

# 概述

## 传统图片服务器

传统的项目中，一般都是在web项目中添加一个文件夹，用来存放上传的图片。例如在工程的目录webroot下面创建一个images文件夹，把图片存放到此文件夹中就可以直接在项目中引用图片。

### 部署模型



### 优点

引用方便，便于管理。

### 缺点

多图片项目中，图片的不断的上传下载会增加服务器额外的压力。

## 分布式环境下的图片服务器

### 部署模型

请求图片

图片上传

Tomcat1

~~Image（存放图片）~~

~~|-image.jpg~~

Tomcat2

~~Image（存放图片）~~

负载均衡服务器

用户请求

图片服务器

Ip或者域名

Image.taotao.com  
nginx实现http服务

/images/2015/07/29/image.jpg

ftp服务接收上传

ftp上传到图片服务器

图片上传

### 优点

将图片的上传和请求分开，便于管理

当单个服务器压力较大时，可以增加服务器个数。

### 缺点

配置较为复杂

服务器个数较多，成本有所提升

# 搭建图片服务器

## 概述

使用VSFTP软件做上传工具、Nginx作为图片服务器。

## 准备

* centOS虚拟机
* nginx-1.8.1.tar.gz

## VSFTPd

|  |
| --- |
| //安装ftp  [root@localhost root]# yum install vsftpd  // 添加用户  [root@localhost root]# useradd ftpuser  // 为ftpuser用户添加密码：fptuser  [root@localhost root]# passwd ftpuser  // 开启21端口  [root@localhost root]# vi /etc/sysconfig/iptables  // 重启防火墙  [root@localhost root]# service iptables restart  // 修改selinux  // 使用命令查询状态  [root@localhost root]# getsebool -a | grep ftp  allow\_ftpd\_anon\_write --> off  allow\_ftpd\_full\_access --> off  allow\_ftpd\_use\_cifs --> off  allow\_ftpd\_use\_nfs --> off  ftp\_home\_dir --> off  ftpd\_connect\_db --> off  ftpd\_use\_passive\_mode --> off  httpd\_enable\_ftp\_server --> off  tftp\_anon\_write --> off  [root@localhost root]# setsebool -P allow\_ftpd\_full\_access on  [root@localhost root]# setsebool -P ftp\_home\_dir on  // 关闭匿名访问，将anonymoue\_enable=YES更改为NO  [root@localhost root]# vi /etc/vsftpd/vsftpd.conf  // 重启vsftpd服务  [root@localhost root]# service vsftpd restart  // 设置开机启动服务  [root@localhost root]# vi /etc/rc.local  在最后一行添加：/etc/rc.d/init.d/vsftpd start |

## NGINX

### nginx安装环境

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 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**

### 编译安装

将nginx-1.8.0.tar.gz拷贝至linux服务器。

解压：

tar -zxvf nginx-1.8.0.tar.gz

cd nginx-1.8.0

1. configure

./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：

./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

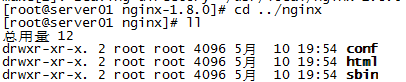
**注意：上边将临时文件目录指定为/var/temp/nginx，需要在/var下创建temp及nginx目录**

1. 编译安装

make

make install

安装成功查看安装目录 ：



### 启动nginx

cd /usr/local/nginx/sbin/

./nginx

查询nginx进程：



15098是nginx主进程的进程id，15099是nginx工作进程的进程id

**注意：执行./nginx启动nginx，这里可以-c指定加载的nginx配置文件，如下：**

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

**如果不指定-c，nginx在启动时默认加载conf/nginx.conf文件，此文件的地址也可以在编译安装nginx时指定./configure的参数（--conf-path= 指向配置文件（nginx.conf））**

### 停止nginx

方式1，快速停止：

cd /usr/local/nginx/sbin

./nginx -s stop

此方式相当于先查出nginx进程id再使用kill命令强制杀掉进程。

方式2，完整停止(建议使用)：

cd /usr/local/nginx/sbin

./nginx -s quit

此方式停止步骤是待nginx进程处理任务完毕进行停止。

### 重启nginx

方式1，先停止再启动（建议使用）：

对nginx进行重启相当于先停止nginx再启动nginx，即先执行停止命令再执行启动命令。

如下：

./nginx -s quit

./nginx

方式2，重新加载配置文件：

当nginx的配置文件nginx.conf修改后，要想让配置生效需要重启nginx，使用-s reload不用先停止nginx再启动nginx即可将配置信息在nginx中生效，如下：

./nginx -s reload

### 测试

nginx安装成功，启动nginx，即可访问虚拟机上的nginx：



到这说明nginx上安装成功。

#### 开机自启动nginx

#### 编写shell脚本

这里使用的是编写shell脚本的方式来处理

vi /etc/init.d/nginx (输入下面的代码)

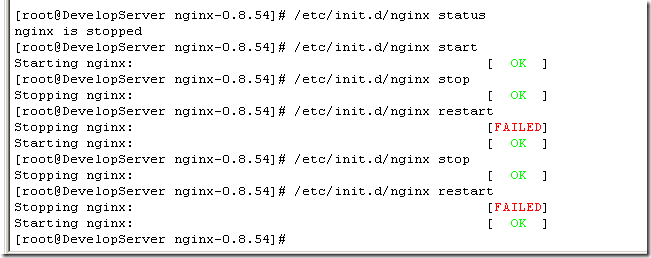
|  |
| --- |
| #!/bin/bash  # nginx Startup script for the Nginx HTTP Server  # it is v.0.0.2 version.  # chkconfig: - 85 15  # description: Nginx is a high-performance web and proxy server.  # It has a lot of features, but it's not for everyone.  # processname: nginx  # pidfile: /var/run/nginx.pid  # config: /usr/local/nginx/conf/nginx.conf  nginxd=/usr/local/nginx/sbin/nginx  nginx\_config=/usr/local/nginx/conf/nginx.conf  nginx\_pid=/var/run/nginx.pid  RETVAL=0  prog="nginx"  # Source function library.  . /etc/rc.d/init.d/functions  # Source networking configuration.  . /etc/sysconfig/network  # Check that networking is up.  [ ${NETWORKING} = "no" ] && exit 0  [ -x $nginxd ] || exit 0  # Start nginx daemons functions.  start() {  if [ -e $nginx\_pid ];then  echo "nginx already running...."  exit 1  fi  echo -n $"Starting $prog: "  daemon $nginxd -c ${nginx\_config}  RETVAL=$?  echo  [ $RETVAL = 0 ] && touch /var/lock/subsys/nginx  return $RETVAL  }  # Stop nginx daemons functions.  stop() {  echo -n $"Stopping $prog: "  killproc $nginxd  RETVAL=$?  echo  [ $RETVAL = 0 ] && rm -f /var/lock/subsys/nginx /var/run/nginx.pid  }  # reload nginx service functions.  reload() {  echo -n $"Reloading $prog: "  #kill -HUP `cat ${nginx\_pid}`  killproc $nginxd -HUP  RETVAL=$?  echo  }  # See how we were called.  case "$1" in  start)  start  ;;  stop)  stop  ;;  reload)  reload  ;;  restart)  stop  start  ;;  status)  status $prog  RETVAL=$?  ;;  \*)  echo $"Usage: $prog {start|stop|restart|reload|status|help}"  exit 1  esac  exit $RETVAL |

:wq 保存并退出

#### 设置文件的访问权限

chmod a+x /etc/init.d/nginx (a+x ==> all user can execute 所有用户可执行)

这样在控制台就很容易的操作nginx了：查看Nginx当前状态、启动Nginx、停止Nginx、重启Nginx…



如果修改了nginx的配置文件nginx.conf，也可以使用上面的命令重新加载新的配置文件并运行，可以将此命令加入到rc.local文件中，这样开机的时候nginx就默认启动了

#### 加入到rc.local文件中

vi /etc/rc.local

加入一行 /etc/init.d/nginx start 保存并退出，下次重启会生效。

## 配置

### 创建目录

|  |
| --- |
| // 进入ftpuser的目录  [root@study home]# cd /home/ftpuser/  // 创建www目录  [root@study home]# mkdir /home/ftpuser/www  // 创建images目录  [root@study home]# mkdir /home/ftpuser/www/images |

### 修改Nginx的配置文件

|  |
| --- |
| // 修改配置文件的目录  [root@study home]# vi /usr/local/nginx/conf/nginx.conf  location / {  root /home/ftpuser/www;  index index.html index.htm;  } |

### 修改/home/ftpuser的权限

|  |
| --- |
| // 设置图片文件夹的权限 【–R：递归设置】  [root@study home]# chmod -R 777 /home/ftpuser |

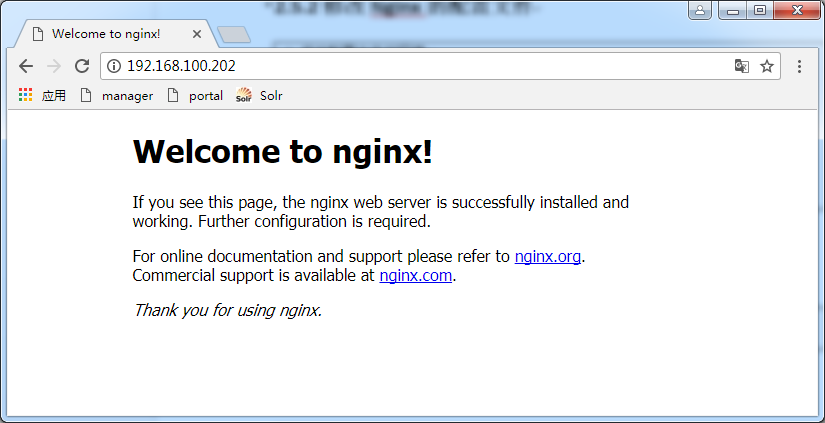
## 测试

### 简单测试

#### 实验

|  |
| --- |
| // 将Nginx安装目录下面的html目录下面的index.html复制到图片根目录  [root@study home]# cp /usr/local/nginx/html/index.html /home/ftpuser/www/ |

#### 结果



成功。

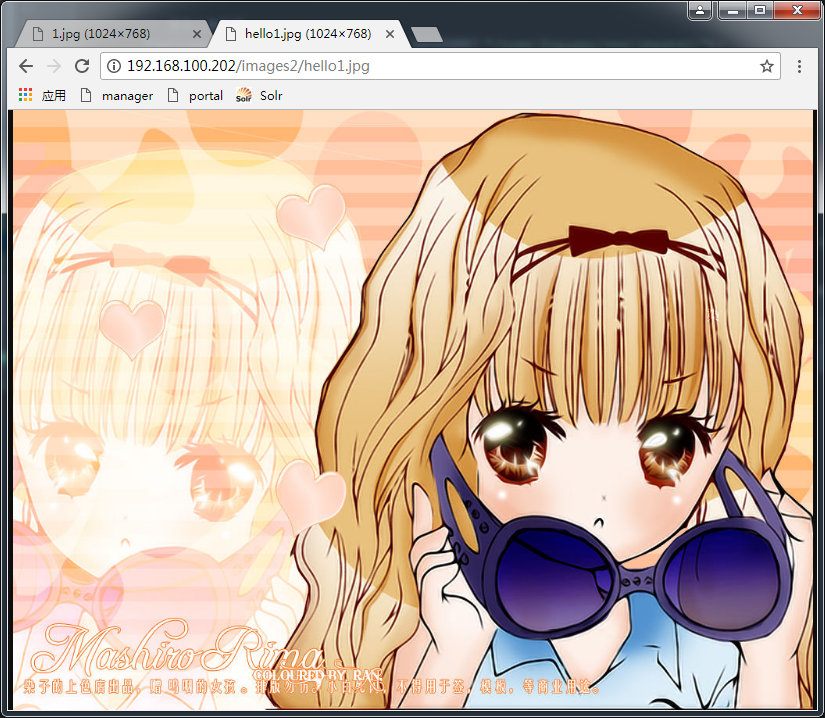
### 代码测试

#### 实验

|  |
| --- |
| @Test  public void testFtpClient() throws Exception {  */\*\*\**  *\* 1. 创建一个ftpClient*  *\* 2. 创建一个连接*  *\* 3. 使用用户名和密码登录到ftp*  *\* 4. 上传文件*  *\* 5. 关闭连接\*/*  FTPClient ftpClient = new FTPClient();  ftpClient.connect("192.168.100.202", 21);  ftpClient.login("ftpuser", "ftpuser");  FileInputStream fileInputStream = new FileInputStream(new File("C:\\Users\\Administrator\\Desktop\\img\\111.jpg"));  // 在服务器上面建立文件夹，然后进行上传  ftpClient.makeDirectory("/home/ftpuser/www/images2/");  // 更换上传目录  ftpClient.changeWorkingDirectory("/home/ftpuser/www/images2/");  ftpClient.setFileType(FTP.*BINARY\_FILE\_TYPE*);  ftpClient.storeFile("hello1.jpg", fileInputStream);  ftpClient.logout();  }  @Test  public void testFtpUtil() throws Exception{  // 测试时，注意关闭Windows的防火墙  FileInputStream fileInputStream = new FileInputStream(new File("C:\\Users\\Administrator\\Desktop\\img\\111.jpg"));  // 此方法不太好用，没有办法创建新的目录  FtpUtil.*uploadFile*("192.168.100.202", 21, "ftpuser", "ftpuser", "/home/ftpuser/www/images/", "2017/09/13", "1.jpg", fileInputStream);  } |

#### 结果

结果一：



结果二：

