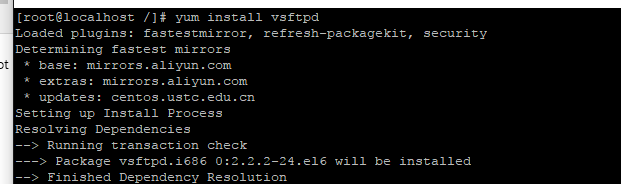
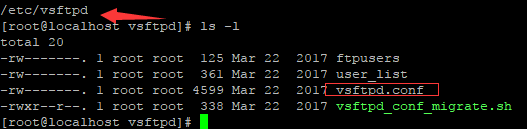
一搭建ftp服务器:

1. 切换到root用户:su root centos开桥接模式。

输入密码:xxxxx



(二):修改/etc/vsftpd/vsftpd.conf文件下的属性



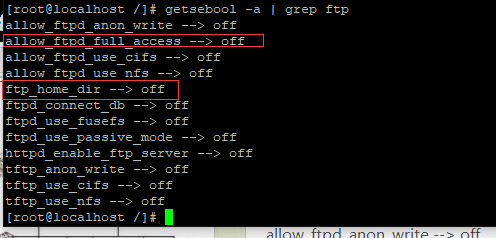
(三):YES/NO 分别表示允许或是不允许匿名访问



(四) : 添加用户与设置密码，登陆后默认的路径是/home/ftpuser.

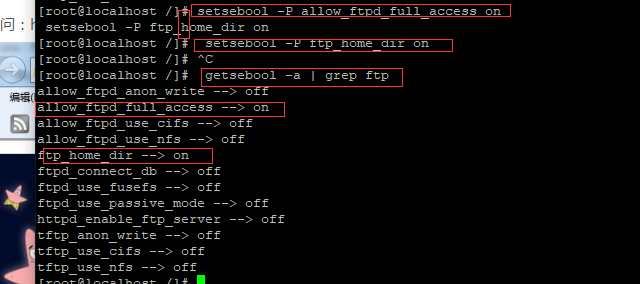


(五):查看外网访问权限

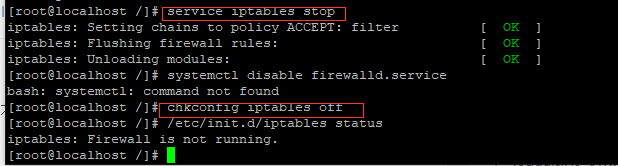


(六)修改让外网可以访问

执行如下图所示的指令，开启外网访问权限。



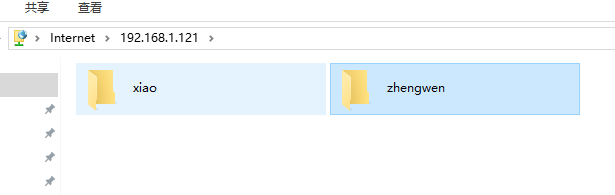
（七）关闭防火墙



访问测试：ftpuser就是linux的系统用户



在windows服务器下新建文件夹



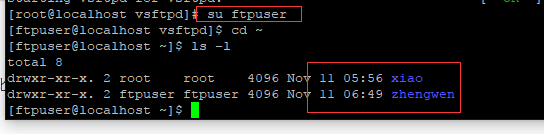
此时切换到ftpuser用户在。

启动ftp命令　　#service vsftpd start

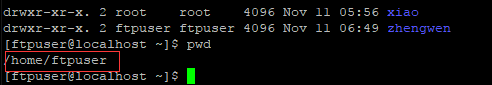
停止ftp命令　　#service vsftpd stop

重启ftp命令　　#service vsftpd restart

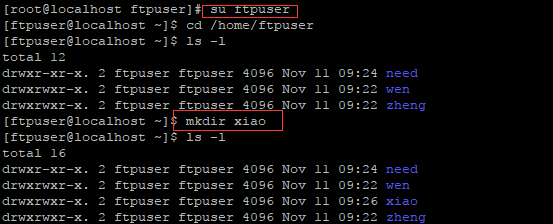
(二)java客户端ftp文件上传:



在home/ftpuser下就能看到文件夹



创建文件夹时要注意是在root用户创建的文件的话，普通用户是无权访问的如上图ftpuser登录是无权访问的，只有自己创建的文件夹，才有文件上传的权限。



二:java客户端上传与下载文件

package com.yonyou.common.util;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import org.apache.commons.net.ftp.FTP;

import org.apache.commons.net.ftp.FTPClient;

import org.apache.commons.net.ftp.FTPFile;

import org.apache.commons.net.ftp.FTPReply;

/\*\*

\*

\* @author Administrator

\*

\*/

public class FtpUtil {

/\*\*

\* Description: 向FTP服务器上传文件

\* @param host FTP服务器主机地址

\* @param port FTP服务器端号

\* @param username FTP登录账号/linux用户 名

\* @param password FTP登录密码 /linux密码

\* @param basePath linux用户下的文件路径

\* @param filePath 文件夹路径

\* @param filename 上传到FTP服务器上的文件名

\* @param input 文件输入流

\* @return 成功返回true，否则返回false

\*/

public static boolean uploadFile(String host, int port, String username, String password, String basePath,

String filePath, String filename, InputStream input) throws Exception {

boolean result = false;

FTPClient ftp = new FTPClient();

int reply;

ftp.connect(host, port);// 连接FTP服务器

// 如果采用默认端口，可以使用ftp.connect(host)的方式直接连接FTP服务端口

ftp.login(username, password);// 登录

reply = ftp.getReplyCode();

if (!FTPReply.isPositiveCompletion(reply)) {

ftp.disconnect();

return result;

}

//切换到上传路径下

boolean change = ftp.changeWorkingDirectory(basePath+filePath);

//切换到上传目目录

if (false==change) {

//如果目录不存在创建目目录

String[] dirs = filePath.split("/");

String tempPath = basePath;

for (String dir : dirs) {

if (null == dir || "".equals(dir)) continue;

tempPath += "/" + dir;

if (!ftp.changeWorkingDirectory(tempPath)) {

if (!ftp.makeDirectory(tempPath)) {

return result;

} else {

ftp.changeWorkingDirectory(tempPath);

}

}

}

}

//设置上传文件的类型为二进制类型

ftp.setFileType(FTP.BINARY\_FILE\_TYPE);

boolean storeFile=true;

try{

storeFile = ftp.storeFile(new String(filename), input);

}catch(Exception e){

e.printStackTrace();

}

if(false==storeFile){

return false;

}

input.close();

ftp.logout();

result = true;

if (ftp.isConnected()) {

try {

ftp.disconnect();

} catch (IOException ioe) {

}

}

return result;

}

/\*\*

\* Description: 从FTP服务器下载文件

\* @param host FTP服务器hostname

\* @param port FTP服务器端口

\* @param username FTP登录账号

\* @param password FTP登录密码

\* @param remotePath FTP服务器上的相对路径

\* @param fileName 要下载的文件名

\* @param localPath 下载后保存到本地的路径

\* @return

\*/

public static boolean downloadFile(String host, int port, String username, String password, String remotePath,

String fileName, String localPath) {

boolean result = false;

FTPClient ftp = new FTPClient();

try {

int reply;

ftp.connect(host, port);

// 如果采用默认端口，可以使用ftp.connect(host)的方式直接连接FTP服务�?

ftp.login(username, password);// 登录

reply = ftp.getReplyCode();

if (!FTPReply.isPositiveCompletion(reply)) {

ftp.disconnect();

return result;

}

ftp.changeWorkingDirectory(remotePath);// 转移到FTP服务器目录下

FTPFile[] fs = ftp.listFiles();

for (FTPFile ff : fs) {

if (ff.getName().equals(fileName)) {

File localFile = new File(localPath + "/" + ff.getName());

OutputStream is = new FileOutputStream(localFile);

ftp.retrieveFile(ff.getName(), is);

is.close();

}

}

ftp.logout();

result = true;

} catch (IOException e) {

e.printStackTrace();

} finally {

if (ftp.isConnected()) {

try {

ftp.disconnect();

} catch (IOException ioe) {

}

}

}

return result;

}

public static void main(String[] args) {

try {

FileInputStream in=new FileInputStream(new File("C:/Users/worn/Desktop/0P91Z1S01.jpg"));

try{

boolean flag = uploadFile("192.168.1.121", 21, "ftpuser", "ftpuser", "/home/ftpuser/","xiao", "0P91Z1S01.jpg", in);

System.out.println(flag);

}catch(Exception e){

e.printStackTrace();

}

} catch (FileNotFoundException e) {

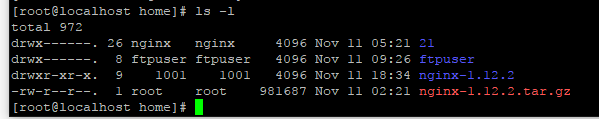
e.printStackTrace();

}

}

}

三：搭建nginx服务器，访问文件。



上传nginx服务器。由于nginx依赖于pcre和zlib，所以在安装nginx时，需要先安装pcre和zlib,OpenSSL。

  PCRE(PerlCompatible 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

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。

 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。

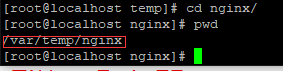
         nginx不仅支持http协议，还支持https（即在ssl协议上传输http），所以需要在linux安装openssl库



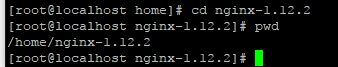




一：首先创建ngixn的临时目录:



进入nginx



执行下面的脚本，创建makefile文件

./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

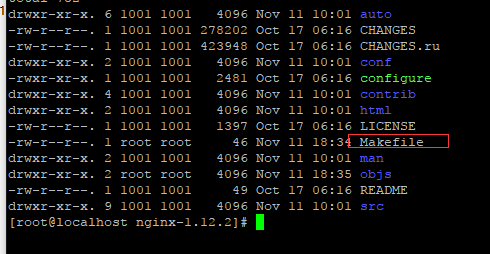
--http-client-body-temp-path=/var/temp/nginx/client

--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



接下来

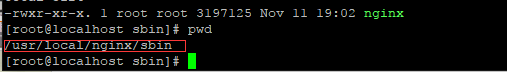
编译



安装



进入nginx目录



可以执行如下指令来控制nginx的启动与停止:

./nginx  //启动

./nginx -s stop //停止

./nginx -s reload //重启

接下来要检测防火强

 查看防火墙状态

1) Linux操作系统中永久性生效，重启后不会复原

开启： chkconfig iptables on

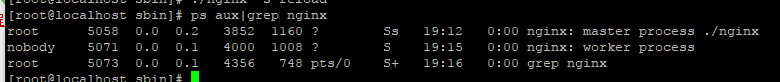
关闭： chkconfig iptables off

2) 即时生效，重启后复原

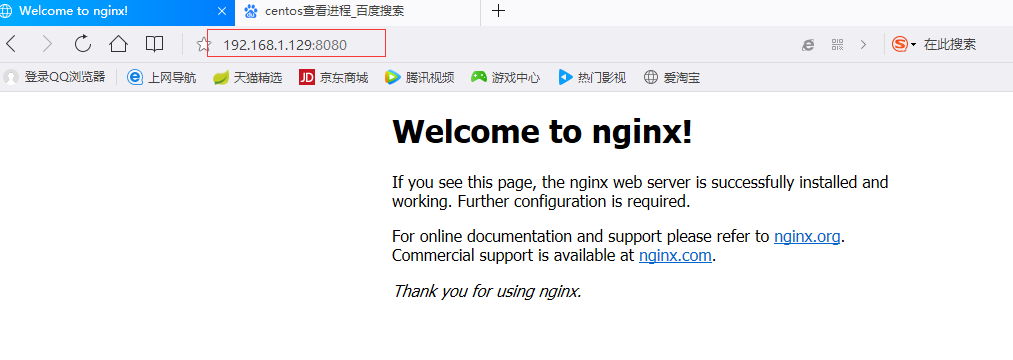
开启： service iptables start

关闭： service iptables stop

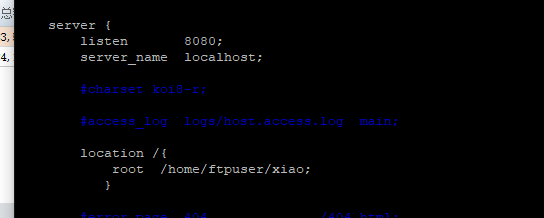
关闭防火墙,接下修改监听端口，并启动nginx， 查看进程



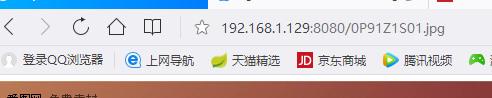
如图所示访问到如下页面nginx启动成功



打开nginx.conf文件配置root地址路径指向我们的ftp文件路径。

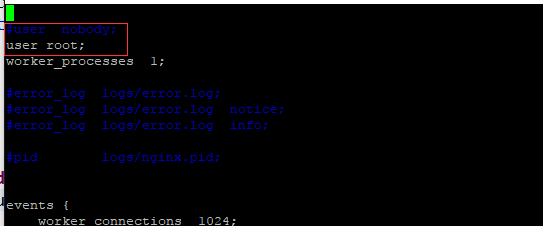


地址映射到指定的目录地址，此时访问

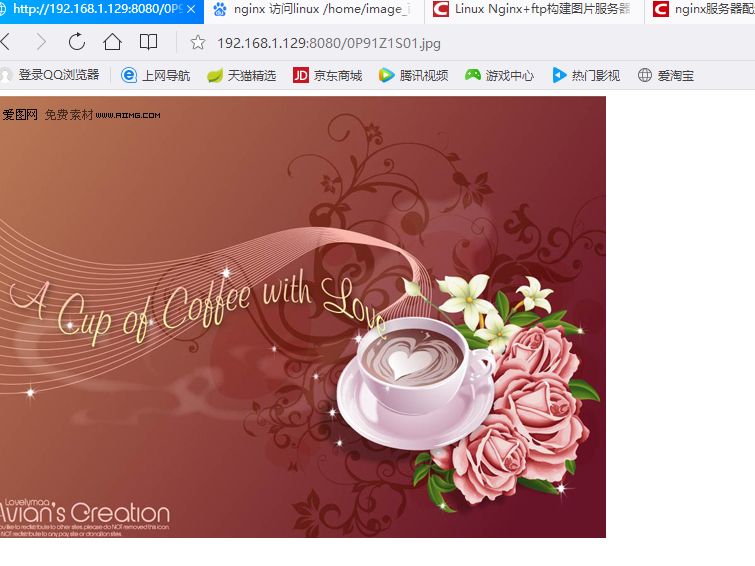


报403错误。

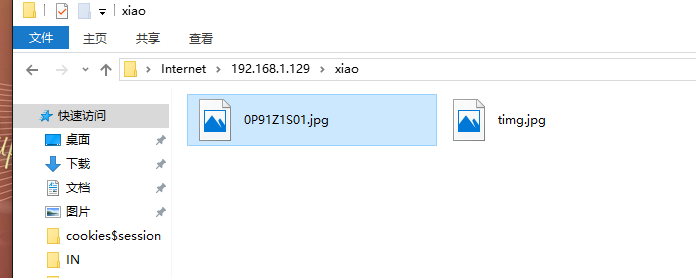
这里原因是没有访问权限，所以要在nginx.conf里面配置root用户的权限



接下来我们再来访问图片



发现图片已经可以通过nginx访问图片



再上传一张图片。



至此整个访问与上传过程已经结束。