目录

[1、文件的上传下载本地 1](#_Toc523730974)

[1.1、上传图片 2](#_Toc523730975)

[1.2、上传文件 2](#_Toc523730976)

[1.3、文件下载（仅用于windows,且只能用浏览器运行） 4](#_Toc523730977)

[1.4、文件下载（通用） 6](#_Toc523730978)

[1.5、多文件的上传 7](#_Toc523730979)

[2、文件的上传下载到远程服务器（推荐） 8](#_Toc523730980)

[2.1、安装ftp组件,配置ftp服务器（centos7） 8](#_Toc523730981)

[2.1.1、安装yum 8](#_Toc523730982)

[2.1.2、通过yum安装ftp 8](#_Toc523730983)

[2.1.3、修改配置文件（匿名用户模式） 8](#_Toc523730984)

[2.1.4、修改配置文件（系统用户模式） 11](#_Toc523730985)

[2.2搭建nginx文件服务器（centos7） 14](#_Toc523730986)

[2.2.1、下载源码（linux环境的只有通过源码安装。windows环境的直接解压就可以使用） 14](#_Toc523730987)

[2.2.2、安装编译环境 15](#_Toc523730988)

[2.2.3、安装nginx 15](#_Toc523730989)

[2.2.4、启动nginx 17](#_Toc523730990)

[2.2.5、测试 18](#_Toc523730991)

[2.2.6、修改/usr/local/nginx/conf/nginx.conf 18](#_Toc523730992)

[2.2.7、每次打开虚拟机，启动nginx的命令 21](#_Toc523730993)

[2.3、java代码（springboot） 22](#_Toc523730994)

[2.3.1、添加依赖 22](#_Toc523730995)

[2.3.1、application. properties 22](#_Toc523730996)

[2.3.2、实体类 FtpConfig 23](#_Toc523730997)

[2.3.3、上传的工具类FtpUtil 25](#_Toc523730998)

[2.3.4、工具类UploadUtils.java 30](#_Toc523730999)

[2.3.5、controller层 31](#_Toc523731000)

[2.3.6、测试 34](#_Toc523731001)

[3、fastdfs分布式文件系统 34](#_Toc523731002)

[3.1、分布式文件系统FastDFS详解 34](#_Toc523731003)

[3.2、FastDFS分布式文件系统集群安装与配置 34](#_Toc523731004)

# 1、文件的上传下载本地

具体解释一下“本地”即前提：项目和文件服务器（nginx）需一起部署到本地（win10）或者一起部署到服务器上（linux）

## 1.1、上传图片

|  |
| --- |
| //上传头像 @PostMapping("/headImg") public Object uploadHeadimg(@RequestParam("file") MultipartFile file) {  String username = SecurityContextHolder.*getContext*().getAuthentication().getName();//获取当前登录用户  System.*out*.println( username );  User user = userService.getUserByUsername(username);  if (!file.isEmpty()) { //文件不是空文件  try {  BufferedOutputStream out = new BufferedOutputStream(  //C:\IDEA\_mode\_project\agriculture\src\main  new FileOutputStream(new File(filepath + username + ".jpg")));//保存图片到目录下,建立保存文件的输入流  out.write(file.getBytes());  out.flush();  out.close();  String filename = filepath + username + ".jpg";  user.setAvater(filename); //设置头像路径  userService.saveOrUpdate(user);//修改用户信息  } catch (FileNotFoundException e) {  e.printStackTrace();  return new Reponse(false,"上传失败," + e.getMessage());  //return "上传失败," + e.getMessage(); //文件路径错误  } catch (IOException e) {  e.printStackTrace();  return new Reponse(false,"上传失败," + e.getMessage());  //return "上传失败," + e.getMessage(); //文件IO错误  }  return new Reponse(true,"上传头像成功",user);//返回用户信息  } else {  return new Reponse(false,"上传失败，因为文件是空的");  } } |

## 1.2、上传文件

可以参考：<https://juejin.im/post/5a5dddde5188257328216ef4>

|  |
| --- |
| @PostMapping("/docUpload") public Object docUpload(@RequestParam("title") String title,  @RequestParam("description") String description,  @RequestParam("file") MultipartFile file ) {  String author = SecurityContextHolder.*getContext*().getAuthentication().getName();//获取当前登录用户  User user = userService.getUserByUsername( author );  //System.out.println( user );  String fileName = file.getOriginalFilename().toString();//获取文件名  //System.out.println( fileName );  if(fileName.indexOf('?')!=fileName.length()-1)  fileName=title+fileName.substring(fileName.lastIndexOf("."));   final SimpleDateFormat sDateFormate = new SimpleDateFormat("yyyymmddHHmmss"); //设置时间格式  String nowTimeStr = sDateFormate.format(new Date()); // 当前时间  fileName=fileName.substring(0,fileName.indexOf("."))+nowTimeStr+fileName.substring(fileName.lastIndexOf("."));   Doc doc = new Doc();  if (!file.isEmpty()) {  try {  BufferedOutputStream out = new BufferedOutputStream(  new FileOutputStream(new File(filepath + fileName)));//保存图片到目录下,建立保存文件的输入流  out.write(file.getBytes());  out.flush();  out.close();  String filename = filepath+fileName;  Long fileSize = file.getSize();  System.*out*.println( file.getSize());   doc.setTitle( title );  doc.setAvatar( filename );  doc.setAuthor( author );  doc.setAuthor\_picture(user.getAvater());  doc.setUptime( new Date() );  doc.setDescription( description );  doc.setFileSize( fileSize );  docService.saveOrUpdateDoc( doc );   } catch (FileNotFoundException e) {  e.printStackTrace();  return new Reponse(false,"上传文件失败," + e.getMessage());  } catch (IOException e) {  e.printStackTrace();  return new Reponse(false,"上传文件失败," + e.getMessage());  }  return new Reponse(true,"上传文件成功",doc);//返回文件信息   }  else {  return new Reponse(false,"上传失败，因为文件是空的");  } } |

## 1.3、文件下载（仅用于windows,且只能用浏览器运行）

|  |
| --- |
| */\*\*  \* Author: zj  \*  \* 文件下载（仅用于windows,且只能用浏览器运行）  \*  \* 如果不设置，则默认下载到C:\\users\\downloads，即本机的默认下载的目录  \** ***@param*** *id  \** ***@param*** *request  \** ***@param*** *response  \** ***@return*** *\*/* @RequestMapping(value = "/download",method = RequestMethod.*GET*)  public Object downloadFile(@RequestParam("id") Long id, org.apache.catalina.servlet4preview.http.HttpServletRequest request, HttpServletResponse response){  Doc doc = docService.getDocById( id );  String fileName = doc.getAvatar();//获取文件路径  System.*out*.println( fileName );  String filename = fileName.split( "/")[4];//获取文件名，该数字要具体情况修改  System.*out*.println( filename );   //String downPath = "C:\\Users\\zj\\Desktop\\"+filename;  if (fileName != null) {  File file = new File(fileName);   System.*out*.println( file );  if (file.exists()) {  response.setContentType("application/force-download");// 设置强制下载不打开  response.addHeader("Content-Disposition",  "attachment;fileName=" + filename);// 设置文件名 // response.addHeader("Content-Disposition", // "attachment;filePath=" + downPath);// 设置文件路径     byte[] buffer = new byte[1024];  FileInputStream fis = null;  BufferedInputStream bis = null;  try {  fis = new FileInputStream(file);//下载的源文件流  bis = new BufferedInputStream(fis);  OutputStream os = response.getOutputStream();  int i = bis.read(buffer);  while (i != -1) {  os.write(buffer, 0, i);  i = bis.read(buffer);  }  //System.out.println("success");  Integer downloads = doc.getDownloads();  doc.setDownloads( downloads+1 );  docService.saveOrUpdateDoc( doc );  dataService.editDocData();  return new Reponse( true,"文件下载成功");  } catch (Exception e) {  e.printStackTrace();  } finally {  if (bis != null) {  try {  bis.close();  } catch (IOException e) {  e.printStackTrace();  }  }  if (fis != null) {  try {  fis.close();  } catch (IOException e) {  e.printStackTrace();  }  }  }  }  }  return new Reponse( false,"未知错误，请联系管理员" );  } |

参考：参考：<https://blog.csdn.net/coding13/article/details/54577076/>

## 1.4、文件下载（通用）

|  |
| --- |
| */\*\*  \*下载通用接口（本项目，安卓可以忽略）  \** ***@param*** *id  \** ***@return*** *\** ***@throws*** *IOException  \*/* @RequestMapping(value = "/auth/download\_android\_disable",method = RequestMethod.*GET*) public Object downLoadFromUrl(@RequestParam("id") Long id) throws IOException{  Doc doc = docService.getDocById( id );  String fileName1 = doc.getAvatar();//获取文件路径  System.*out*.println( fileName1 );  String filename = fileName1.split( "/")[4];//获取文件名，该数字要具体情况修改  String savePath = "C:\\Users\\zj\\Desktop\\"+filename;//保存地址,请自己修改    String urlStr = "http://127.0.0.1:8088/"+filename;  URL url = new URL(urlStr);  HttpURLConnection conn = (HttpURLConnection)url.openConnection();  //设置超时间为3秒  conn.setConnectTimeout(3\*1000);  //防止屏蔽程序抓取而返回403错误  conn.setRequestProperty("User-Agent", "Mozilla/4.0 (compatible; MSIE 5.0; Windows NT; DigExt)");   //得到输入流  InputStream inputStream = conn.getInputStream();  //获取自己数组  byte[] getData = *readInputStream*(inputStream);   File file = new File(savePath);  FileOutputStream fos = new FileOutputStream(file);  fos.write(getData);  if(fos!=null){  fos.close();  }  if(inputStream!=null){  inputStream.close();  }   System.*out*.println("info:"+url+" download success");   Integer downloads = doc.getDownloads();  doc.setDownloads( downloads+1 );  docService.saveOrUpdateDoc( doc );  return null;  } |

## 1.5、多文件的上传

参考：

<https://blog.csdn.net/qq_39774931/article/details/81773706>

<https://blog.csdn.net/qq_34083066/article/details/80652183>

<https://blog.csdn.net/coding13/article/details/54577076/>

前端和配置文件，依赖略，百度一搜就是一大堆

# 2、文件的上传下载到远程服务器（推荐）

功能：文件服务器(nginx)是部署到服务器上（linux），但项目可以部署在本地（win10）

前提：

在linux服务器上安装ftp组件

在linux服务器上部署nginx文件服务器，

## 2.1、安装ftp组件,配置ftp服务器（centos7）

### 2.1.1、安装yum

apt install yum

如果报错，就apt-get update一下

### 2.1.2、通过yum安装ftp

yum -y install vsftpd

### 2.1.3、修改配置文件（匿名用户模式）

**用户访问模式配置**

**vsftpd服务访问模式有三种：匿名用户模式，系统用户模式和虚拟用户模式！**

我这里采用的是匿名用户模式

其他的模式请参考：<https://blog.csdn.net/aiynmimi/article/details/77012507>

(1)/etc/vsftpd/vsftpd.conf这个文件是vsftpd服务的核心配置文件！

我们在修改配置文件的时候，最好先备份一份！

cp /etc/vsftpd/vsftpd.conf /etc/vsftpd/vsftpd.conf.older

(2)了解配置的大致内容：<https://blog.csdn.net/aiynmimi/article/details/77012507>

（3）匿名用户模式的配置

Vsftpd默认以匿名用户访问，匿名用户默认访问的FTP服务器端路径为：/var/ftp/pub，**匿名用户只有查看权限，无法创建、删除、修改**。   
这种模式下，**不需改动配置文件**，直接启动服务即可访问！

|  |
| --- |
| service vsftpd start  //或者  systemctl start vsftpd.service  重启服务使用restart |

测试1：

在cmd 中输入 ftp+ip地址

默认用户ftp,密码为空



测试2：在浏览器输入<ftp://ip>地址



ps:如果有蓝灯，要关掉蓝灯，否则会报错

（4）如果想要允许匿名用户能够上传、下载、删除文件，需修改/etc/vsftpd/vsftpd.conf配置文件中：

添加：

anon\_upload\_enable=YES #允许匿名用户上传文件；

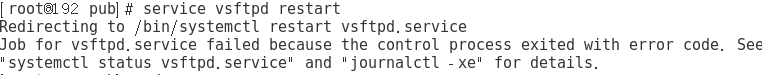
anon\_mkdir\_write\_enable=YES #允许匿名用户创建目录；

anon\_other\_write\_enable=YES #允许匿名用户其他写入权限。

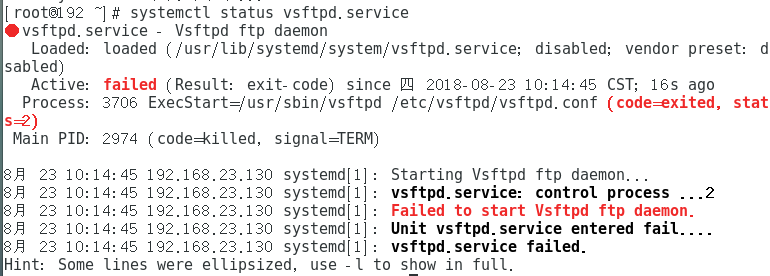
修改好重启vsftpd

service vsftpd restart

发现报错



查看一下：systemctl status vsftpd.service



我最终还是没有解决这个问题，而且匿名模式开启传输权限，太危险了。

我觉得改用系统用户模式

另外默认Vsftpd匿名用户有两个：anonymous、ftp，所以匿名用户如果需要上传文件、删除及修改等权限，需要ftp用户对/var/ftp/pub目录有写入权限，使用如下chown和chmod任意一种即可，设置命令如下：

chown -R ftp pub/

ps:如果要关闭匿名用户登录，只需设置：（我们这里不要设置）

anonymous\_enable=NO

### 2.1.4、修改配置文件（系统用户模式）

从上面测试2重新开始：

匿名模式可以让任何人使用ftp服务，比较公开！多适用于共享文件！如果我们想要特定用户使用，就需要使用系统用户登录访问！这种模式，需要我们新建不同用户，linux创建用户：

（1）创建系统用户

useradd 新的用户名

passwd 新的用户名

**ftpuser 123456**

（2）修改配置文件

修改：

|  |
| --- |
| anonymous\_enable=NO #禁止匿名用户登录，把默认YES改成NO  #开启被动模式  #这样远程连接才可以进行传输数据  #默认是开启的，但是要指定一个端口范围，打开vsftpd.conf文件，在后面加上  pasv\_min\_port=30000  pasv\_max\_port=30999 |

重新启动：

systemctl restart vsftpd

（3）关闭防火墙

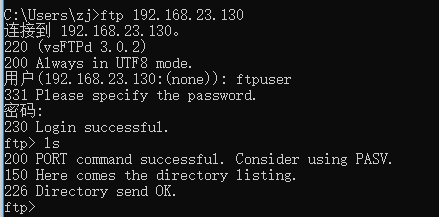
service firewalld stop

（4）测试

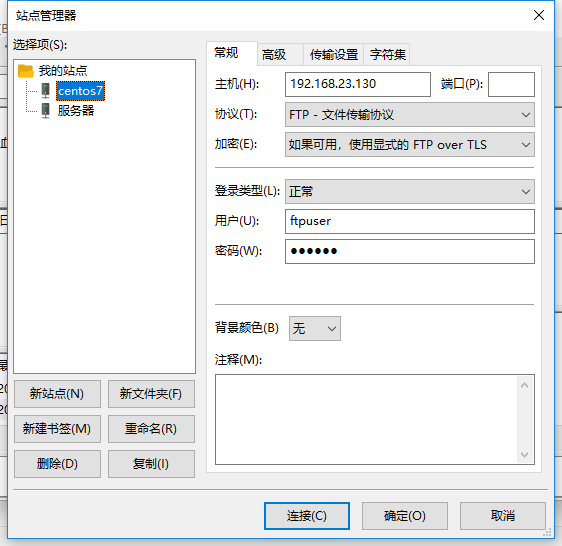
测试1



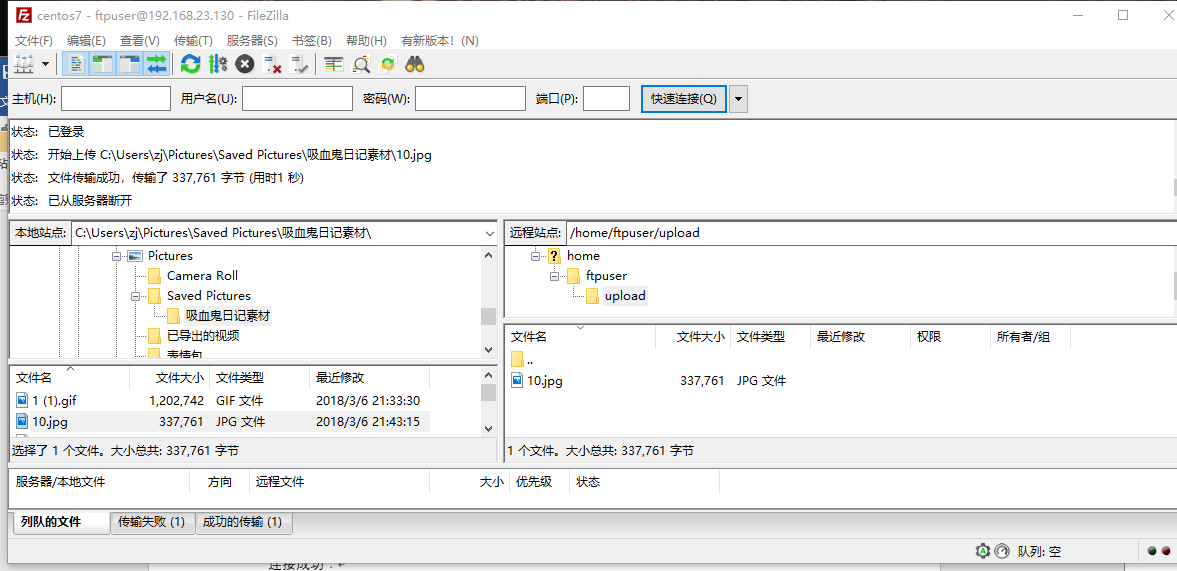
测试2



测试3：



连接成功：



发现传文件过去是报错

修改文件权限

cd /home/ftpuser

chmod 777 upload

ps: ftpuser不能为777，但他的子目录可以为777

成功

这种模式下，登录访问的目录就是/home/新建用户/

自此成功！！！

参考：

centos7:

<https://blog.csdn.net/maoyuanming0806/article/details/78067180>

<https://blog.csdn.net/aiynmimi/article/details/77012507>

<https://blog.csdn.net/csdn_lqr/article/details/53333946>

ubuntu:

<https://blog.csdn.net/xlgen157387/article/details/40115509>

<https://blog.csdn.net/qq_33279781/article/details/73607466>

<https://blog.csdn.net/yancey_blog/article/details/52790451>

## 2.2搭建nginx文件服务器（centos7）

nginx我之前有一篇文章，不过是ubuntu,但两者差不多

### 2.2.1、下载源码（linux环境的只有通过源码安装。windows环境的直接解压就可以使用）

官方网站：<http://nginx.org/en/download.html>

下载好了，移到虚拟机中

### 2.2.2、安装编译环境

1、需要安装gcc的环境。yum install gcc-c++

2、第三方的开发包。

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

### 2.2.3、安装nginx

1、把nginx的源码包上传到linux系统

2、解压缩

tar zxf nginx-1.14.0.tar.gz

3、使用configure命令创建一makeFile文件

进入解压文件中可以看到有configure



然后进行配置生成makefile

|  |
| --- |
| ./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 |

复制上面的，粘贴在命令行，然后回车

|  |
| --- |
| C:\Users\zj\Desktop\2345截图20180823234935.png |

表示成功

生成了makefile

|  |
| --- |
| C:\Users\zj\Desktop\2345截图20180823235017.png |

4、执行安装命令：

make &&make install

安装成功后：



### 2.2.4、启动nginx

进入 /usr/local/nginx/sbin

有个nginx可执行文件

./nginx 及开启了服务器。

一个错误，启动是缺少文件夹：所以nginx的需要的文件夹必须存在，即使没有文件在其中



创建：

**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**

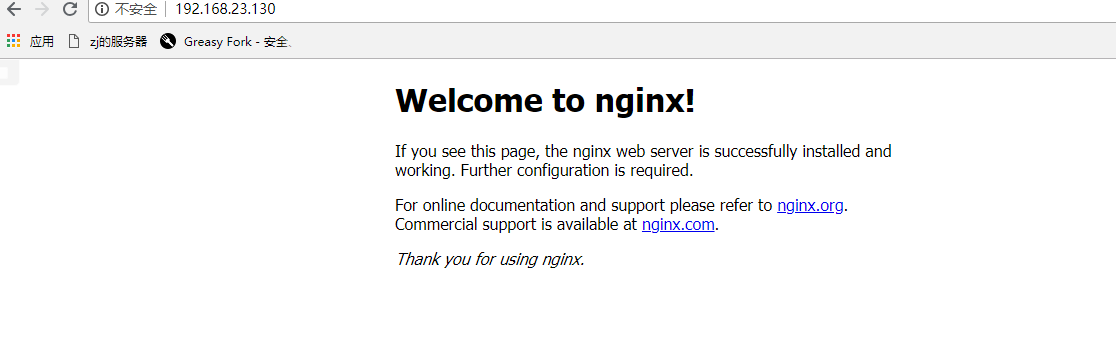
[root@localhost sbin]# mkdir /var/temp/nginx/client -p

注意：-p   
表示级联创建文件夹





### 2.2.5、测试



自此nginx安装成功

### 2.2.6、修改/usr/local/nginx/conf/nginx.conf

最后的想要的结果是：

图片通过ftp服务上传到/home/ftpuser/upload/目录下，我想通过访问Nginx服务器来访问ftp目录下的图片文件，该url为http:// 192.168.23.130 /picture/x.jpg。即使用http请求访问原本需要使用ftp请求才能访问到的资源文件。

|  |
| --- |
| user root;//解决403 forbidden  server {  listen 80;  server\_name 192.168.23.130;//虚拟机或服务器的ip地址  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  root html;  index index.html index.htm;  }  #zj add  location /picture/ { #location指定了我们的图片服务器存放图片的地>址  root /home/ftpuser/upload; #指定图片/视频存放的地址  access\_log on;  autoindex on;//打开浏览功能  } |

注意：

图片是放在root /home/ftpuser/upload/picture

ftpuser 目录权限不能为777

upload,picture目录权限为777

参考：

Nginx 出现 403 Forbidden 最终解决方法\_nginx\_脚本之家

<https://www.jb51.net/article/121064.htm>

配置完后需要重启nginx服务器：

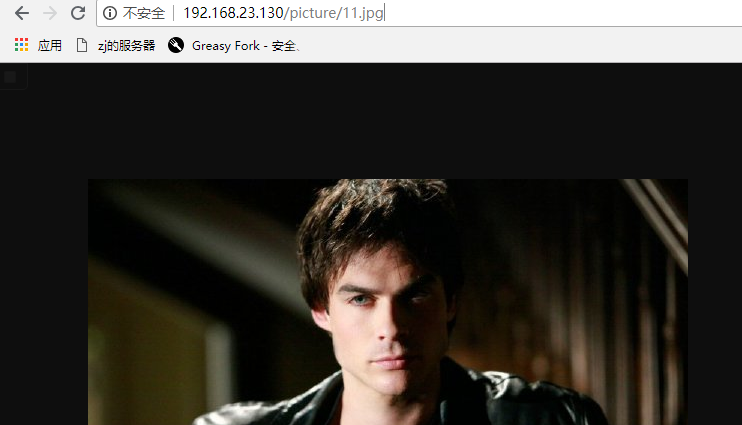
./nginx -s reload

测试1：



注意最后一个“/”

测试2：



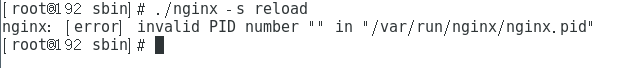
第二天打开报错：



CentOS启动nginx出现nginx: [emerg] open() "/var/run/nginx/nginx.pid" failed (2: No such file or director） - CSDN博客

<https://blog.csdn.net/z_cc_csdn/article/details/78122282>

但又出现新的问题：



linux操作系统重启后 解决nginx的pid消失问题 - 陈钦 - 博客园

<https://www.cnblogs.com/cyq632694540/p/6483535.html>

最终解决方案：

1.首先你要先确保关闭nginx进程了（可以通过pkill -9 nginx关闭进程）

2.进入你nginx的安装目录里面 cd `路径/nginx/sbin`里面 然后输入

>nginx/sbin/nginx -c nginx/conf/nginx.conf 就可以了（sbin目录下面有个nginx为启动程序，-c为以什么配置启动，后面接着nginx配置文件的路径）

pkill -9 nginx

cd /usr/local/nginx/sbin

./nginx -c conf/nginx.conf

### 2.2.7、每次打开虚拟机，启动nginx的命令

cd /usr/local/nginx/sbin

mkdir /var/run/nginx

cd /var/run/nginx

touch nginx.conf

cd –

./nginx -c conf/nginx.conf

ps:

为什么要这样，因为每次虚拟机重启后，var/run/nginx，nginx这个文件夹都会被删除，搞得每一次都要去建立nginx这个文件夹。也可以写个sh脚本

如果你不想像上面每次启动那么麻烦。请参考：

<https://blog.csdn.net/zyhlearnjava/article/details/71932719>

如果修改配置文件后，需要重启nginx服务器：

./nginx -s reload

## 2.3、java代码（springboot）

### 2.3.1、添加依赖

|  |
| --- |
| <!-- 加入上传文件组件 --> <dependency>  <groupId>commons-fileupload</groupId>  <artifactId>commons-fileupload</artifactId>  <version>1.3.1</version> </dependency> <dependency>  <groupId>commons-io</groupId>  <artifactId>commons-io</artifactId>  <version>2.1</version> </dependency> <dependency>  <groupId>commons-net</groupId>  <artifactId>commons-net</artifactId>  <version>3.3</version> </dependency> |

### 2.3.1、application. properties

|  |
| --- |
| #ftp相关配置 FTP\_ADDRESS=192.168.23.130 FTP\_PORT=21 FTP\_USERNAME=ftpuser FTP\_PASSWORD=123456 FTP\_BASEPATH=/home/ftpuser/upload/picture #图片服务器相关配置 IMAGE\_BASE\_URL=http://192.168.23.1303/picture |

后来发现该配置没有起作用，故删去

### 2.3.2、实体类 FtpConfig

|  |
| --- |
| */\*\*  \** ***@Auther:*** *zj  \** ***@Date:*** *2018/8/24 17:28  \** ***@Description:*** *配置ftp服务器的相关参数  \*/* @Component public class FtpConfig {   */\*\*  \* 获取ip地址  \*/* @Value("${FTP\_ADDRESS}")  private String FTP\_ADDRESS="192.168.23.130";   */\*\*  \* 端口号  \*/* @Value("${FTP\_PORT}")  private String FTP\_PORT="21";   */\*\*  \* 用户名  \*/* @Value("${FTP\_USERNAME}")  private String FTP\_USERNAME="ftpuser";   */\*\*  \* 密码  \*/* @Value("${FTP\_PASSWORD}")  private String FTP\_PASSWORD="123456";   */\*\*基本路径  \*/* @Value("${FTP\_BASEPATH}")  private String FTP\_BASEPATH="/home/ftpuser/upload/picture";   */\*\*  \* 下载地址地基础url  \*/* @Value("${IMAGE\_BASE\_URL}")  private String IMAGE\_BASE\_URL="http://192.168.23.130/picture";   public FtpConfig() {   }   public String getFTP\_ADDRESS() {  return FTP\_ADDRESS;  }   public void setFTP\_ADDRESS(String FTP\_ADDRESS) {  this.FTP\_ADDRESS = FTP\_ADDRESS;  }   public String getFTP\_PORT() {  return FTP\_PORT;  }   public void setFTP\_PORT(String FTP\_PORT) {  this.FTP\_PORT = FTP\_PORT;  }   public String getFTP\_USERNAME() {  return FTP\_USERNAME;  }   public void setFTP\_USERNAME(String FTP\_USERNAME) {  this.FTP\_USERNAME = FTP\_USERNAME;  }   public String getFTP\_PASSWORD() {  return FTP\_PASSWORD;  }   public void setFTP\_PASSWORD(String FTP\_PASSWORD) {  this.FTP\_PASSWORD = FTP\_PASSWORD;  }   public String getFTP\_BASEPATH() {  return FTP\_BASEPATH;  }   public void setFTP\_BASEPATH(String FTP\_BASEPATH) {  this.FTP\_BASEPATH = FTP\_BASEPATH;  }   public String getIMAGE\_BASE\_URL() {  return IMAGE\_BASE\_URL;  }   public void setIMAGE\_BASE\_URL(String IMAGE\_BASE\_URL) {  this.IMAGE\_BASE\_URL = IMAGE\_BASE\_URL;  } } |

### 2.3.3、上传的工具类FtpUtil

|  |
| --- |
| */\*\*  \** ***@Auther:*** *zj  \** ***@Date:*** *2018/8/24 17:33  \** ***@Description:*** *上传的工具类FtpUtil  \*/* public class FtpUtil {   */\*\*  \* ftp上传文件方法  \*title:pictureUpload  \*****@param*** *ftpConfig 由spring管理的FtpConfig配置，在调用本方法时，可以在使用此方法的类中通过@AutoWared注入该属性。由于本方法是静态方法，所以不能在此注入该属性  \*****@param*** *picNewName 图片新名称--防止重名 例如："1.jpg"  \*****@param*** *picSavePath 图片保存路径。注：最后访问路径是 ftpConfig.getFTP\_ADDRESS()+"/images"+picSavePath  \*****@param*** *inputStream 要上传的文件（图片）  \*****@return*** *若上传成功，返回图片的访问路径，若上传失败，返回null  \** ***@throws*** *IOException  \*/* public static String pictureUploadByConfig(FtpConfig ftpConfig, String picNewName, String picSavePath, InputStream inputStream) throws IOException {   String picHttpPath = null;  //String picHttpPath = "";   boolean flag = *uploadFile*(ftpConfig.getFTP\_ADDRESS(), ftpConfig.getFTP\_PORT(), ftpConfig.getFTP\_USERNAME(),  ftpConfig.getFTP\_PASSWORD(), ftpConfig.getFTP\_BASEPATH(), picSavePath, picNewName, inputStream);   if(!flag){  System.*out*.println( "上传失败" );  return picHttpPath;  }   //picHttpPath = ftpConfig.getFTP\_ADDRESS()+"/images"+picSavePath+"/"+picNewName;  picHttpPath = ftpConfig.getIMAGE\_BASE\_URL()+picSavePath+"/"+picNewName;  System.*out*.println("上传成功==="+picHttpPath);  return picHttpPath;  }   */\*\*  \* ftp下载文件的方法  \* \*@param ftpConfig 由spring管理的FtpConfig配置，在调用本方法时，可以在使用此方法的类中通过@AutoWared注入该属性。由于本方法是静态方法，所以不能在此注入该属性  \*****@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*** *若上传成功，返回  \** ***@throws*** *IOException  \*/* public static String pictureDowmloadByConfig(FtpConfig ftpConfig, String remotePath, String fileName, String localPath) throws IOException {   boolean flag = *downloadFile*(ftpConfig.getFTP\_ADDRESS(),Integer.*parseInt*(ftpConfig.getFTP\_PORT()),ftpConfig.getFTP\_USERNAME(),  ftpConfig.getFTP\_PASSWORD(),remotePath,fileName,localPath);   if(!flag){  System.*out*.println( "下载失败" );  return null;  }   System.*out*.println( "下载成功" );  return null;  }    */\*\*  \* Description: 向FTP服务器上传文件  \** ***@param*** *host FTP服务器hostname  \** ***@param*** *ftpPort FTP服务器端口  \** ***@param*** *username FTP登录账号  \** ***@param*** *password FTP登录密码  \** ***@param*** *basePath FTP服务器基础目录  \** ***@param*** *filePath FTP服务器文件存放路径。例如分日期存放：/2015/01/01。文件的路径为basePath+filePath  \** ***@param*** *filename 上传到FTP服务器上的文件名  \** ***@param*** *input 输入流  \** ***@return*** *成功返回true，否则返回false  \*/* public static boolean uploadFile(String host, String ftpPort, String username, String password, String basePath,  String filePath, String filename, InputStream input) {  int port = Integer.*parseInt*(ftpPort);  boolean result = false;  FTPClient ftp = new FTPClient();  try {  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服务器  ftp.login(username, password);// 登录  reply = ftp.getReplyCode();  if (!FTPReply.*isPositiveCompletion*(reply)) {  ftp.disconnect();  return result;  }  //切换到上传目录  if (!ftp.changeWorkingDirectory(basePath+filePath)) {  //如果目录不存在创建目录  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*);  ftp.enterLocalPassiveMode();//这个设置允许被动连接--访问远程ftp时需要  //上传文件  if (!ftp.storeFile(filename, input)) {  return result;  }  input.close();  ftp.logout();  result = true;  } catch (IOException e) {  e.printStackTrace();  } finally {  if (ftp.isConnected()) {  try {  ftp.disconnect();  } catch (IOException ioe) {  }  }  }  return result;  }     //下载文件方法不用看，可能日后有用，先留在这里==========================================    */\*\*  \* 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服务器  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  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;  } } |

### 2.3.4、工具类UploadUtils.java

|  |
| --- |
| */\*\*  \** ***@Auther:*** *zj  \** ***@Date:*** *2018/8/24 17:40  \** ***@Description:*** *上传工具类 UploadUtils  \*/* public class UploadUtils {   */\*\*  \* 得到真实文件名  \** ***@param*** *fileName  \** ***@return*** *\*/* public static String subFileName(String fileName){  //查找最后一个 \ (文件分隔符)位置  int index = fileName.lastIndexOf( File.*separator*);  if(index == -1){  //没有分隔符，说明是真实名称  return fileName;  }else {  return fileName.substring(index+1);  }  }   */\*\*  \* 获得随机UUID文件名  \** ***@param*** *fileName  \** ***@return*** *\*/* public static String generateRandonFileName(String fileName){  //首相获得扩展名，然后生成一个UUID码作为名称，然后加上扩展名  String ext = fileName.substring(fileName.lastIndexOf("."));  return UUID.*randomUUID*().toString()+ext;  }   */\*\*  \* 获得hashcode 生成二级目录  \** ***@param*** *uuidFileName  \** ***@return*** *\*/* public static String generateRandomDir(String uuidFileName){  int hashCode = uuidFileName.hashCode();//得到它的hashcode编码  //一级目录  int d1 = hashCode & 0xf;  //二级目录  int d2 = (hashCode >> 4) & 0xf;  return "/"+d1+"/"+d2;  } } |

### 2.3.5、controller层

|  |
| --- |
| //上传头像  @PostMapping("/headImg")  public Object uploadHeadimg(@RequestParam("file") MultipartFile file) {  String username = SecurityContextHolder.*getContext*().getAuthentication().getName();//获取当前登录用户  System.*out*.println( username );  User user = userService.getUserByUsername(username);  if (!file.isEmpty()) { //文件不是空文件  try {  BufferedOutputStream out = new BufferedOutputStream(  //C:\IDEA\_mode\_project\agriculture\src\main  new FileOutputStream(new File(filepath + username + ".jpg")));//保存图片到目录下,建立保存文件的输入流  out.write(file.getBytes());  out.flush();  out.close();  String filename = filepath + username + ".jpg";  user.setAvater(filename); //设置头像路径  userService.saveOrUpdate(user);//修改用户信息  } catch (FileNotFoundException e) {  e.printStackTrace();  return new Reponse(false,"上传失败," + e.getMessage());  //return "上传失败," + e.getMessage(); //文件路径错误  } catch (IOException e) {  e.printStackTrace();  return new Reponse(false,"上传失败," + e.getMessage());  //return "上传失败," + e.getMessage(); //文件IO错误  }  return new Reponse(true,"上传头像成功",user);//返回用户信息  } else {  return new Reponse(false,"上传失败，因为文件是空的");  }  }  //上传文件（ftp + nginx)（多文件上传或单文件上传）  @PostMapping("/ftp/upload")  public Object uploadHeadImgNew(@RequestParam("file") MultipartFile[] files) throws IOException {  String username = SecurityContextHolder.*getContext*().getAuthentication().getName();//获取当前登录用户  System.*out*.println( username );  //User user = userService.getUserByUsername(username);   FtpConfig ftpConfig = new FtpConfig();  //System.out.println( ftpConfig.getFTP\_ADDRESS() );  for (MultipartFile file : files) {  //Photo photo = new Photo();  String oldName = file.getOriginalFilename();// 获取图片原来的名字  System.*out*.println( oldName );  String picNewName = UploadUtils.*generateRandonFileName*(oldName);// 通过工具类产生新图片名称，防止重名  System.*out*.println( picNewName );  //String picSavePath = UploadUtils.generateRandomDir(picNewName);// 通过工具类把图片目录分级  String picSavePath = "/"+username;//以用户名为目录  System.*out*.println( picSavePath ); /\*  \* photo.setPhotoUrl(picSavePath + "/");//  \* 设置图片的url--》就是存储到数据库的字符串url photo.setAlbumId(albumId);//  \* 设置图片所属相册id photo.setUser\_id("wk");  \* photo.setPhoteName(picNewName);  \*/  //photoList.add(photo);  FtpUtil.*pictureUploadByConfig*(ftpConfig, picNewName, picSavePath, file.getInputStream());// 上传到图片服务器的操作  // 添加到数据库  }   return new Reponse( true,"上传头像成功" );  }     */\*\*  \* 下载文件（ftp + nginx)（单个文件下载）  \** ***@param*** *remotePath 绝对路径或相对路径  \** ***@param*** *fileName 文件名  \** ***@param*** *localPath 下载后保存到本地的路径  \** ***@throws*** *IOException  \* for example:  remotePath="/home/ftpuser/upload/picture/18083764688"+  fileName="33691e23-b864-4069-a83c-2427c27cdd96.jpg"+  localPath="C:\Users\zj\Desktop"  \*/* @PostMapping("/ftp/download")  public void downloadHeadImgNew(@RequestParam("remotePath") String remotePath,  @RequestParam("fileName") String fileName,  @RequestParam("localPath") String localPath) throws IOException {  String username = SecurityContextHolder.*getContext*().getAuthentication().getName();//获取当前登录用户  System.*out*.println( username );  //User user = userService.getUserByUsername(username);   FtpConfig ftpConfig = new FtpConfig();     FtpUtil.*pictureDowmloadByConfig*( ftpConfig,remotePath,fileName,localPath );  System.*out*.println( "文件下载成功" );  } |

### 2.3.6、测试

用postman亲测可用

参考：

<https://blog.csdn.net/qq_36762765/article/details/79539251>

<https://blog.csdn.net/u012149181/article/details/80608752>

<https://blog.csdn.net/maoyuanming0806/article/details/78068091>

# 3、fastdfs分布式文件系统

## 3.1、分布式文件系统FastDFS详解

分布式文件系统FastDFS详解 - 纯洁的微笑 - 博客园

<https://www.cnblogs.com/ityouknow/p/8240976.html>

## 3.2、FastDFS分布式文件系统集群安装与配置

1、SpringBoot文件上传下载和多文件上传（图文详解） - CSDN博客

<https://blog.csdn.net/coding13/article/details/54577076>

java 从网络Url中下载文件 - CSDN博客

<https://blog.csdn.net/xb12369/article/details/40543649/>

Springboot文件下载 - CSDN博客

<https://blog.csdn.net/stubbornness1219/article/details/72356632>