# 安装 jdk1.8

本文主要介绍的是如何是Linux环境下安装JDK的，因为Linux环境下，很多时候也离不开Java的，下面笔者就和大家一起分享如何jdk1.8的过程吧。

## 1、安装环境

操作系统：CentOS7 64 位

JDK版本：1.8

工具：Xshell5、Xftp5

说明：本文是通过Xshell5工具远程连接Linux操作，如果是直接在Linux可视化界面操作那就更方便了，原理一样。

也可使用工具：SecureCRT 8.3、FileZilla\_3.42.1

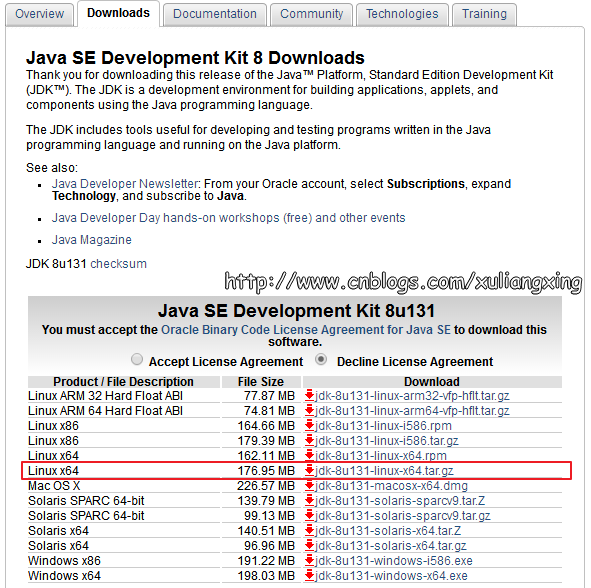
## 2、安装步骤

**第一步：下载安装包**

下载Linux环境下的jdk1.8，请去（[官网](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html" \t "_blank)）中下载jdk的安装文件；

由于我的Linux是64位的，因此我下载[jdk-8u131-linux-x64.tar.gz](http://download.oracle.com/otn-pub/java/jdk/8u131-b11/d54c1d3a095b4ff2b6607d096fa80163/jdk-8u131-linux-x64.tar.gz)。

如下图所示：



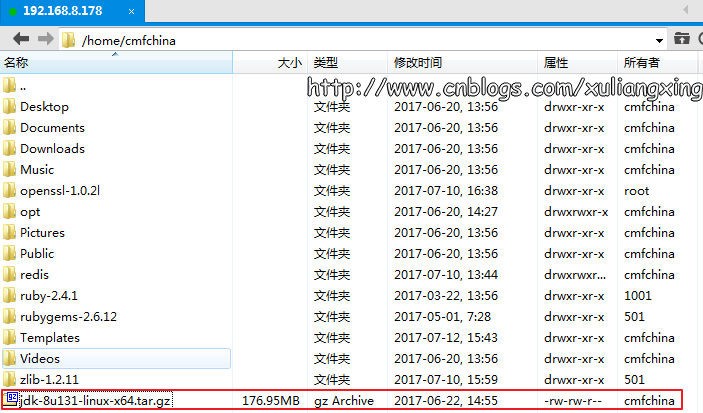
如果Linux本身连接到互联网，我们可以直接通过wget命令直接把JDK安装包下载下来，如图所示：



要是没有外网的环境，还是安装上面的方法下载安装包，然后上传到服务器当中

**第二步、解压安装包**

将我们下载好的JDK安装包上传到服务器，进行解压



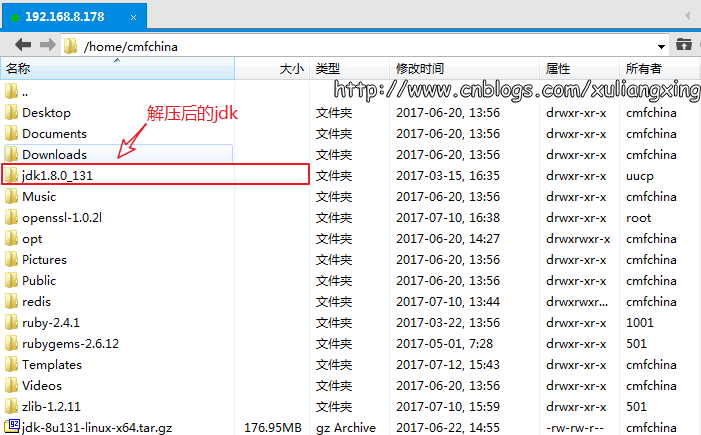
解压命令进行解压

1 $ cd /home/cmfchina

2 $ tar -zxvf jdk-8u131-linux-x64.tar.gz

https://images2015.cnblogs.com/blog/506829/201707/506829-20170712154949462-1567604342.png

解压完成之后，可以在当前目录下看到一个名字为【jdk1.8.0\_131】的目录，里面存放的是相关文件



我们要将解压后的【jdk1.8.0\_131】里面的所有数据移动到我们需要安装的文件夹当中，我们打算将jdk安装在usr/java当中，我们在usr目录下新建一个java文件夹

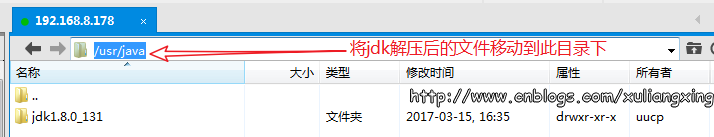
mkdir /usr/java

https://images2015.cnblogs.com/blog/506829/201707/506829-20170712155622509-501402302.png

将【jdk1.8.0\_131】里的数据拷贝至java目录下

mv /home/cmfchina/jdk1.8.0\_131 /usr/java

https://images2015.cnblogs.com/blog/506829/201707/506829-20170712160243556-345559176.png



**第三步、修改环境变量**

至此，我们最后需要修改环境变量，通过命令

vim /etc/profile

https://images2015.cnblogs.com/blog/506829/201707/506829-20170712161307134-1500571117.png

用vim编辑器来编辑profile文件，在文件末尾添加一下内容（按“i”进入编辑）：

1 export JAVA\_HOME=/usr/java/jdk1.8.0\_131

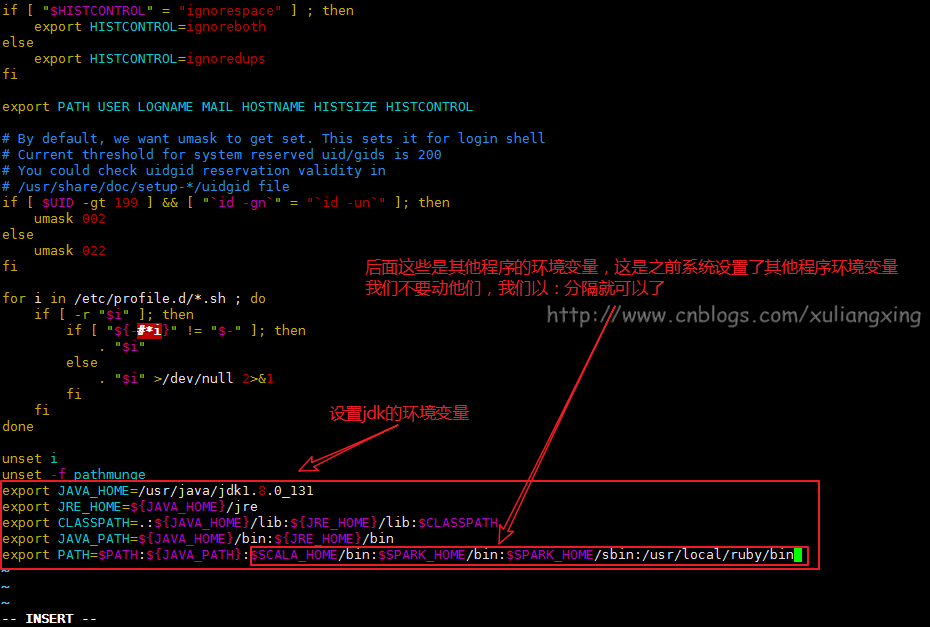
2 export JRE\_HOME=${JAVA\_HOME}/jre

3 export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib:$CLASSPATH

4 export JAVA\_PATH=${JAVA\_HOME}/bin:${JRE\_HOME}/bin

5 export PATH=$PATH:${JAVA\_PATH}

如图所示：



然后，保存并退出(按：wq!)

保存完之后，我们还需要让这个环境变量配置信息里面生效，要不然只能重启电脑生效了。

通过命令source /etc/profile让profile文件立即生效，如图所示

https://images2015.cnblogs.com/blog/506829/201707/506829-20170712162556978-2141450378.png

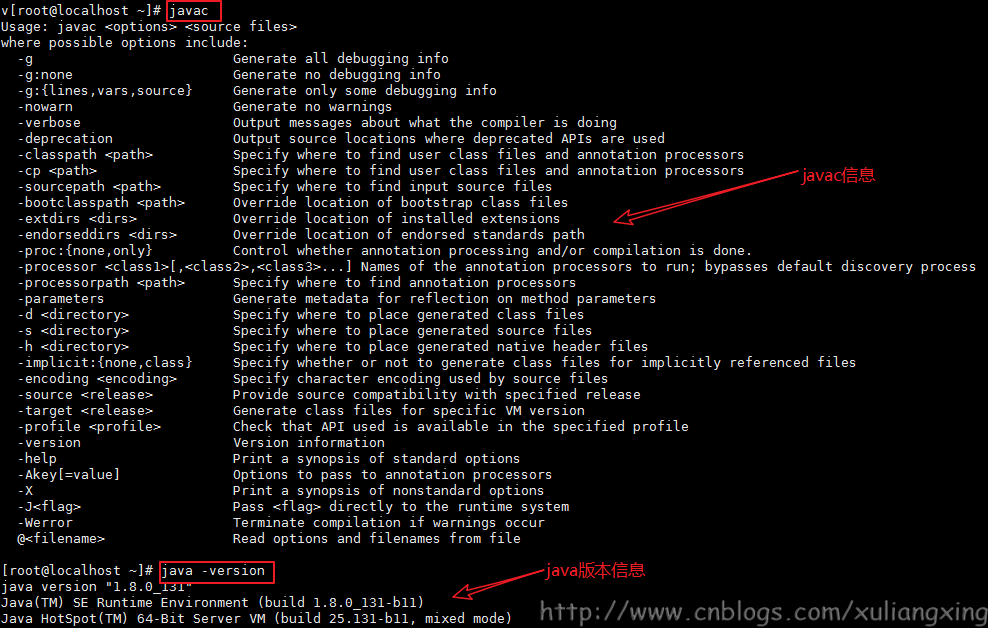
**第四步、测试是否安装成功**

①、使用javac命令，不会出现command not found错误

②、使用java -version，出现版本为java version "1.8.0\_131"

③、echo $PATH，看看自己刚刚设置的的环境变量配置是否都正确

如图所示：



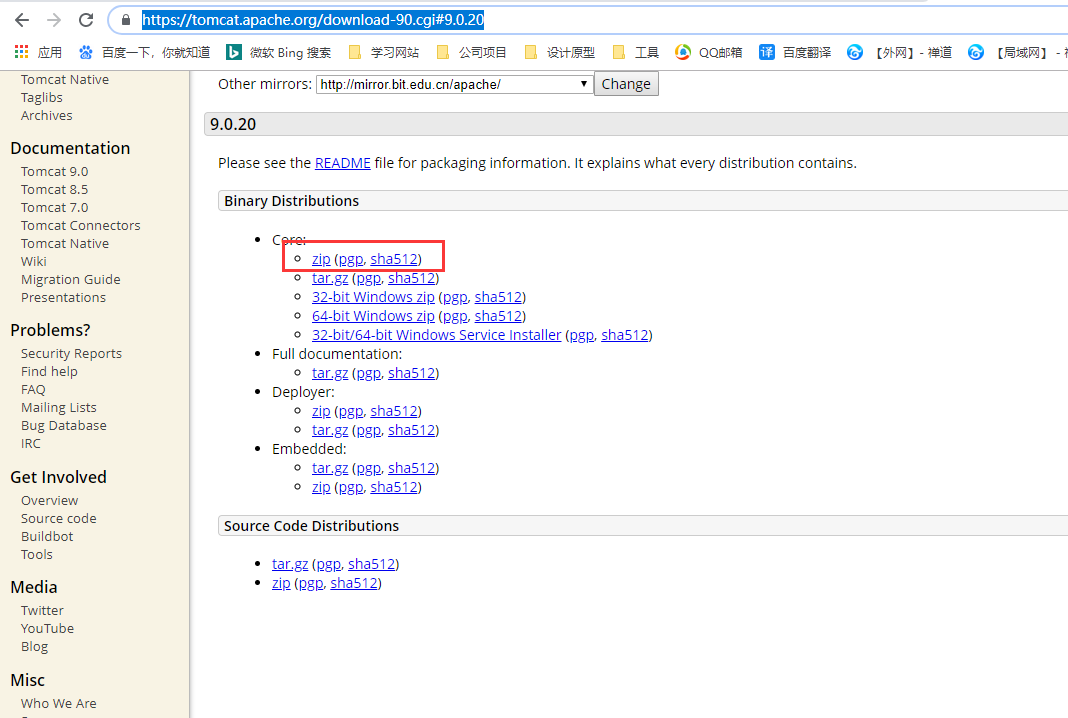
至此，安装结束

# 安装 tomcat及问应用

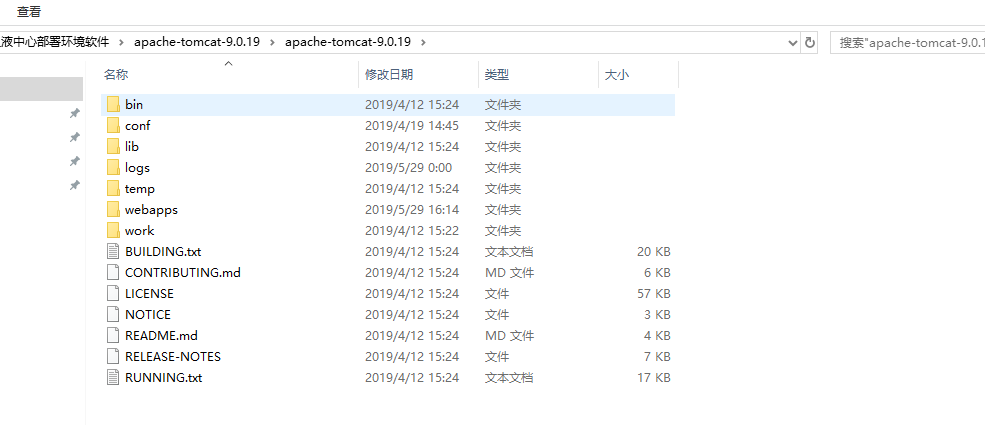
## Tomcat安装和启动

### **第一步：下载apache-tomcat-9.0.19**

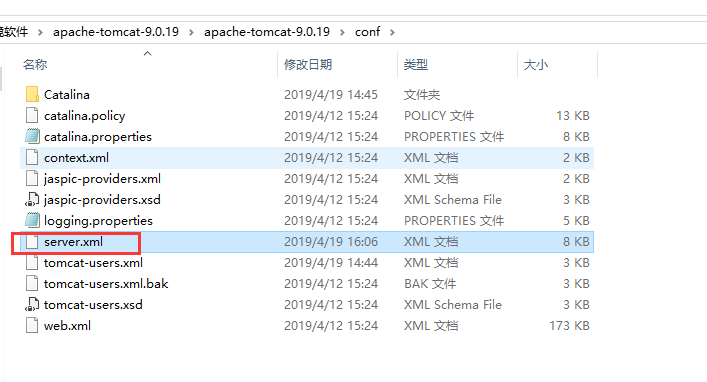
官网下载地址：<https://tomcat.apache.org/download-90.cgi#9.0.20>

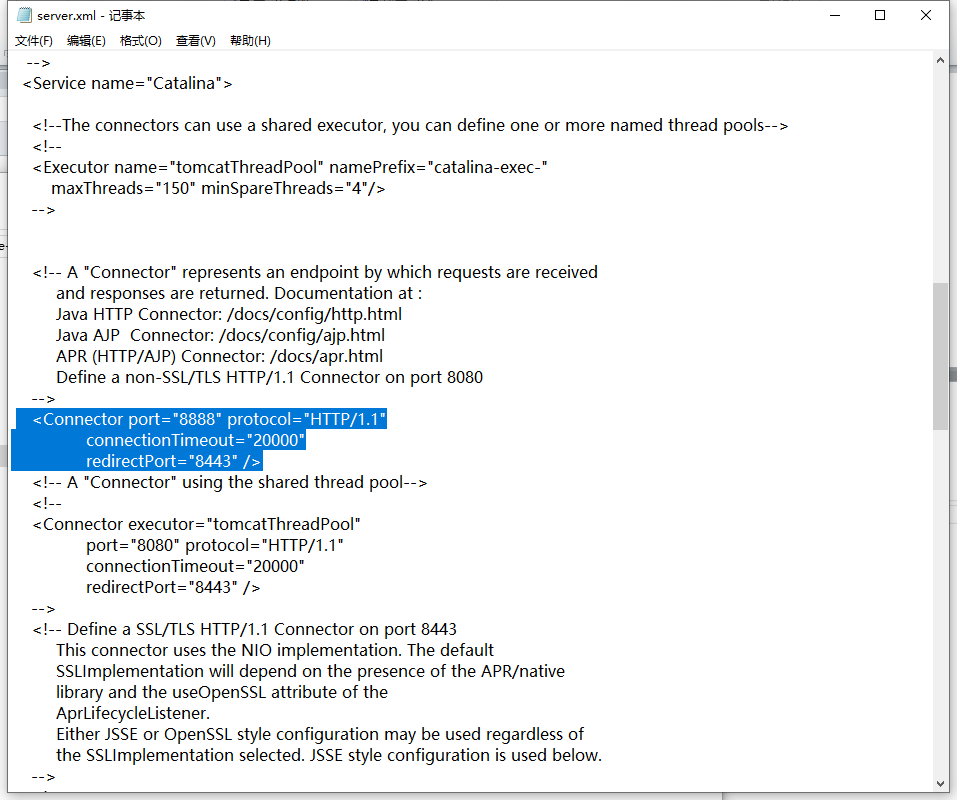


### **第二步：解压tomcat**



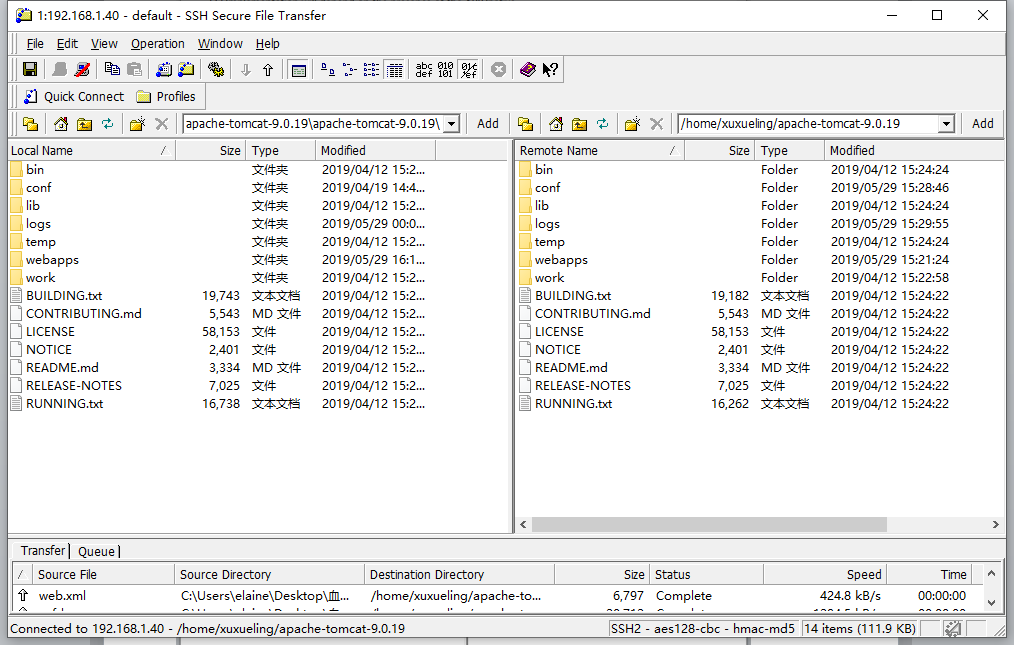
### **第三步：更改tomcat端口配置server.xml，例如更改为8888**





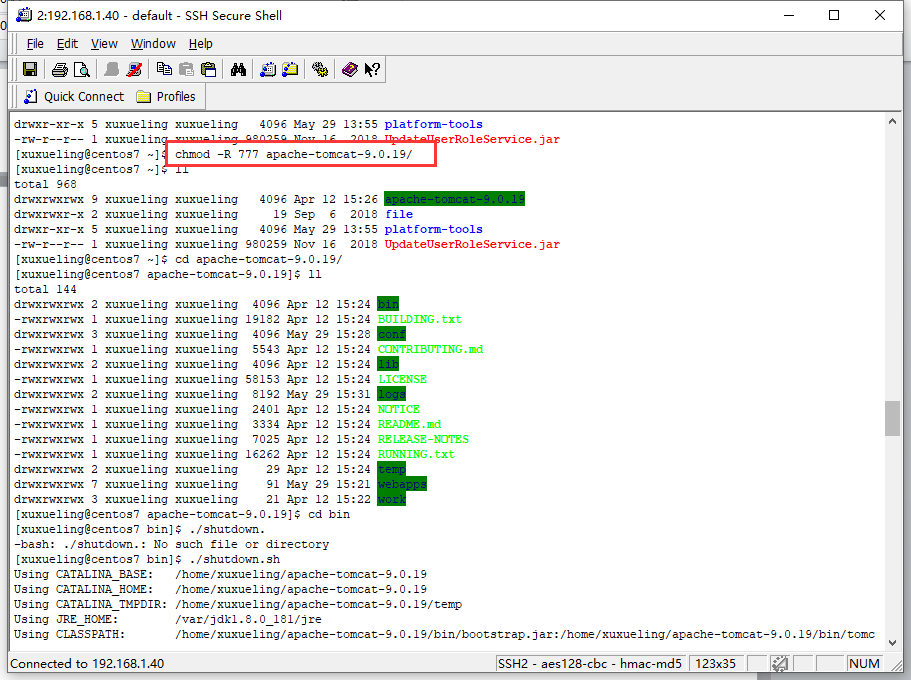
### **第四步：将更改好的tomcat拷贝到linux中**

可使用工具SSHSecureShellClient进行文件传输和远程访问测试



### **第五步：使用linux监视器窗口进行命令执行**

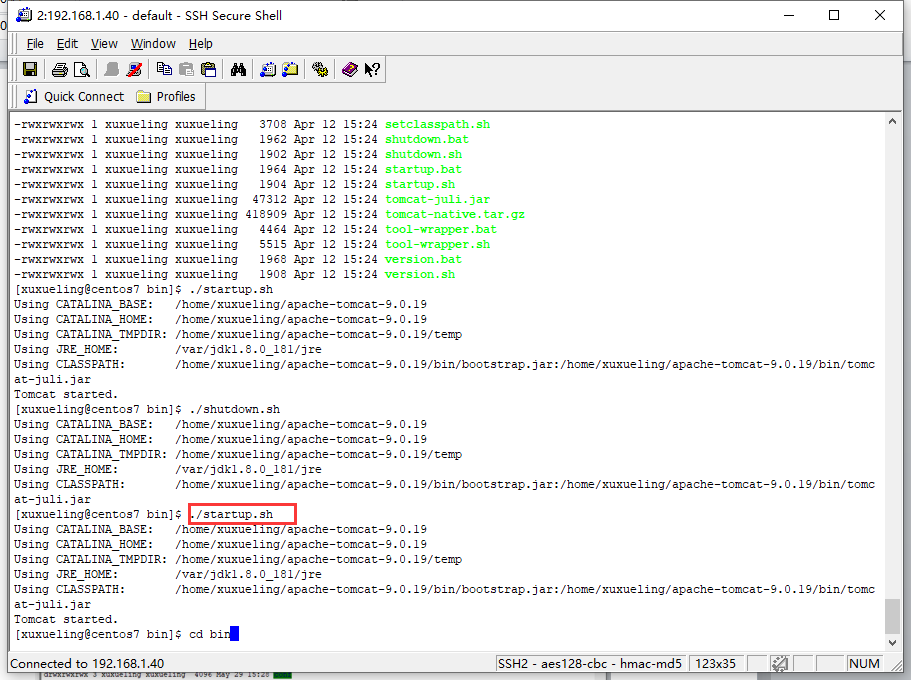
权限设置



使用cd命令转到bin目录下，启动tomcat

./startup.sh 启动

./shutdown.sh 关闭



**查看tomcat运行日志（和IDEA控制台输出是一样的内容）**

先切换到：cd Tomcat安装目录/logs (需根据个人tomcat目录进入)

tail -f catalina.out   (实时查看运行日志)

Ctrl+c 是退出tail命令

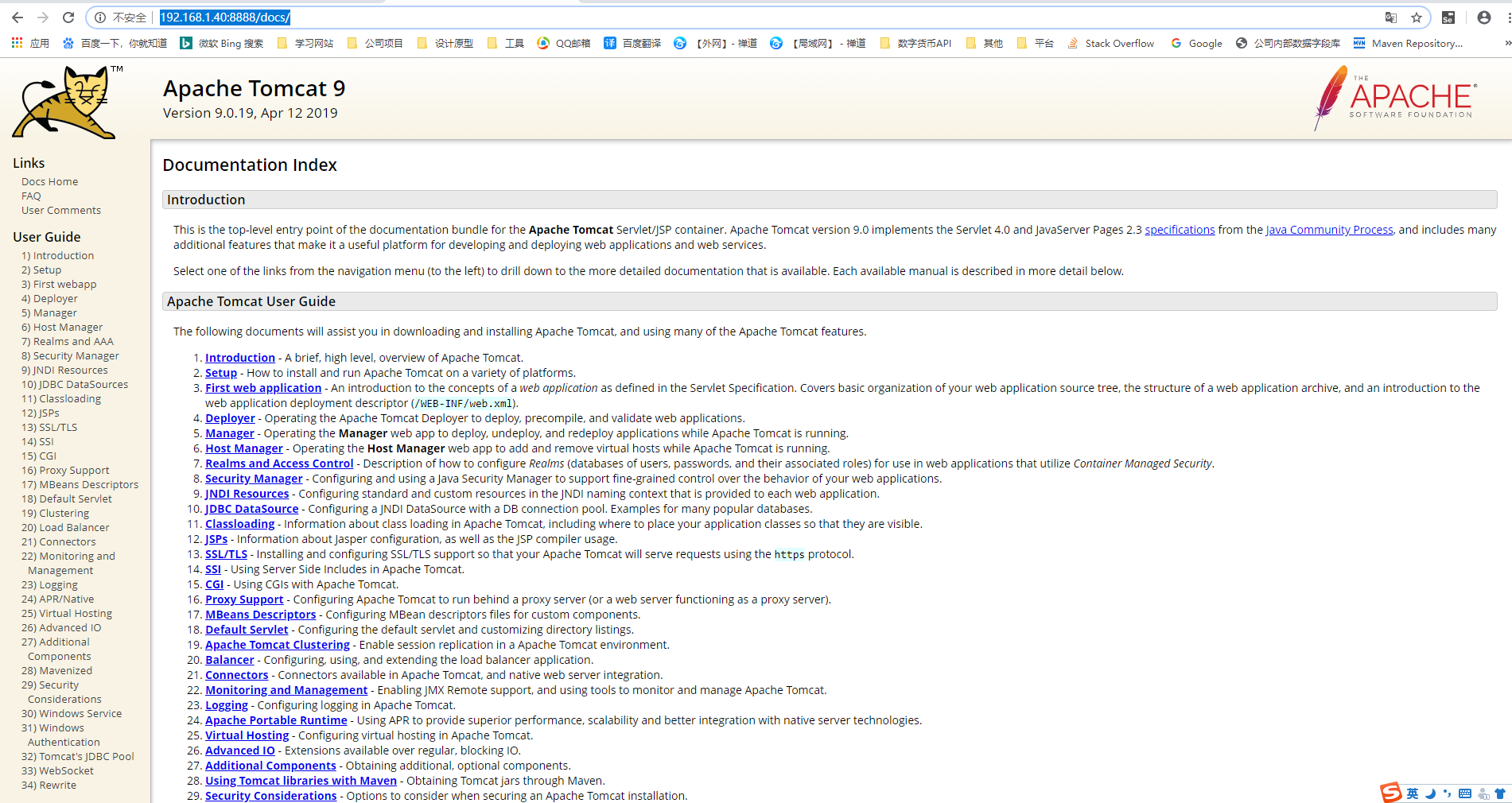
**更多查看tomcat使用命令可查看网页资料：**

<https://www.cnblogs.com/duodmi/articles/9809530.html>

### **第六步：检查访问是否正常**

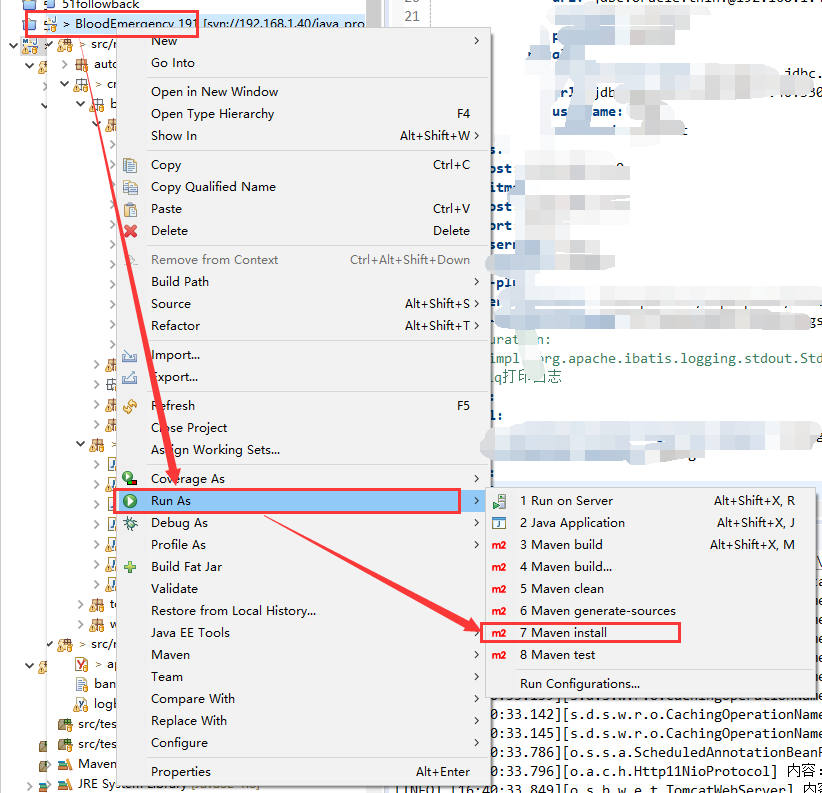
在浏览器中访问<http://ip地址:8888/docs/>

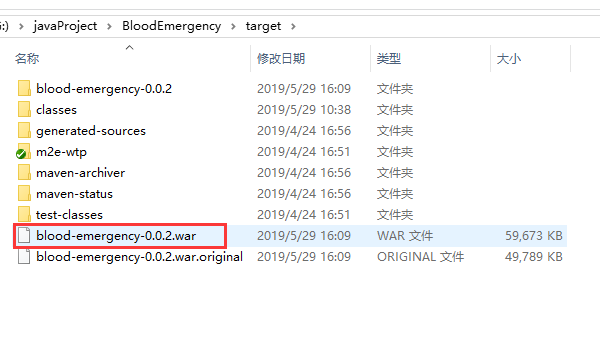
正常显示界面，则tomcat启动成功



## 部署web应用

### **第一步：使用编辑器将web应用源码打包成war包**





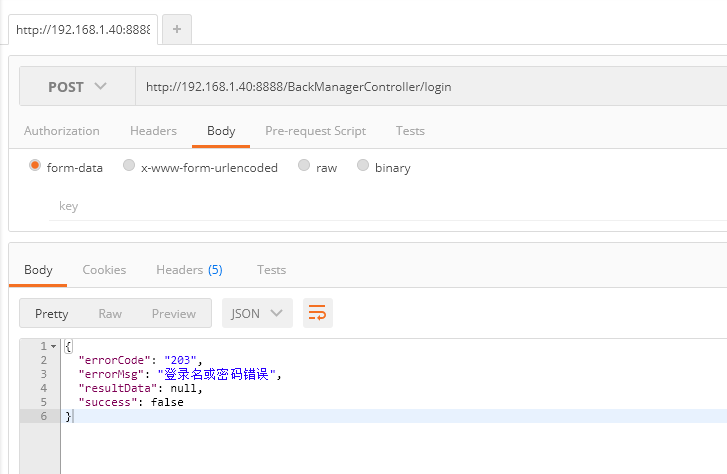
### **第二步：重命名war包名称为ROOT.war，并拷贝到linux下的apache-tomcat-9.0.19\webapps下**

注：最好提前将程序依赖的配置文件改为部署对象服务器所需的信息

### **第三步：启动tomcat，会自动解压war包**

### **第四步：进行访问检查**

找一个简单的请求接口进行访问请求

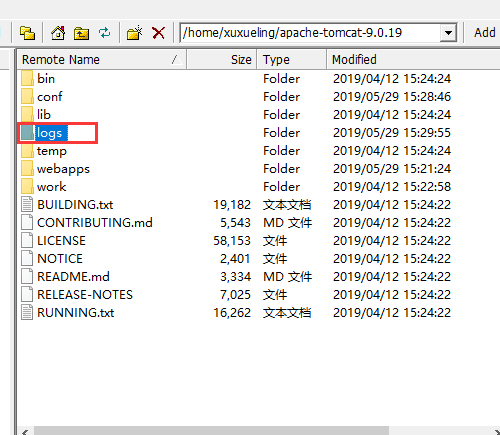


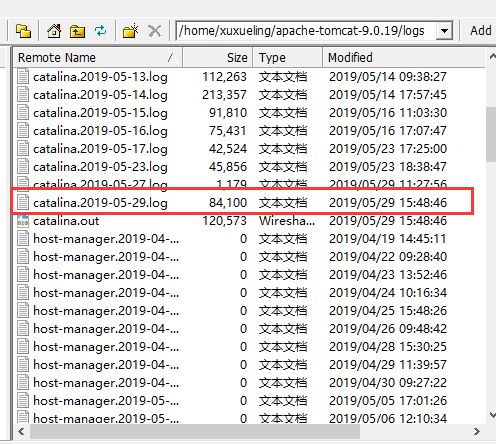
# 部署应用过程中遇到的问题

## 1、Tomcat页面能正常访问，但是无法访问到web应用

找到tomcat下的logs文件夹，进去查看最新的catalina日志（catalina.2019-05-29.log），查看报错详细信息。

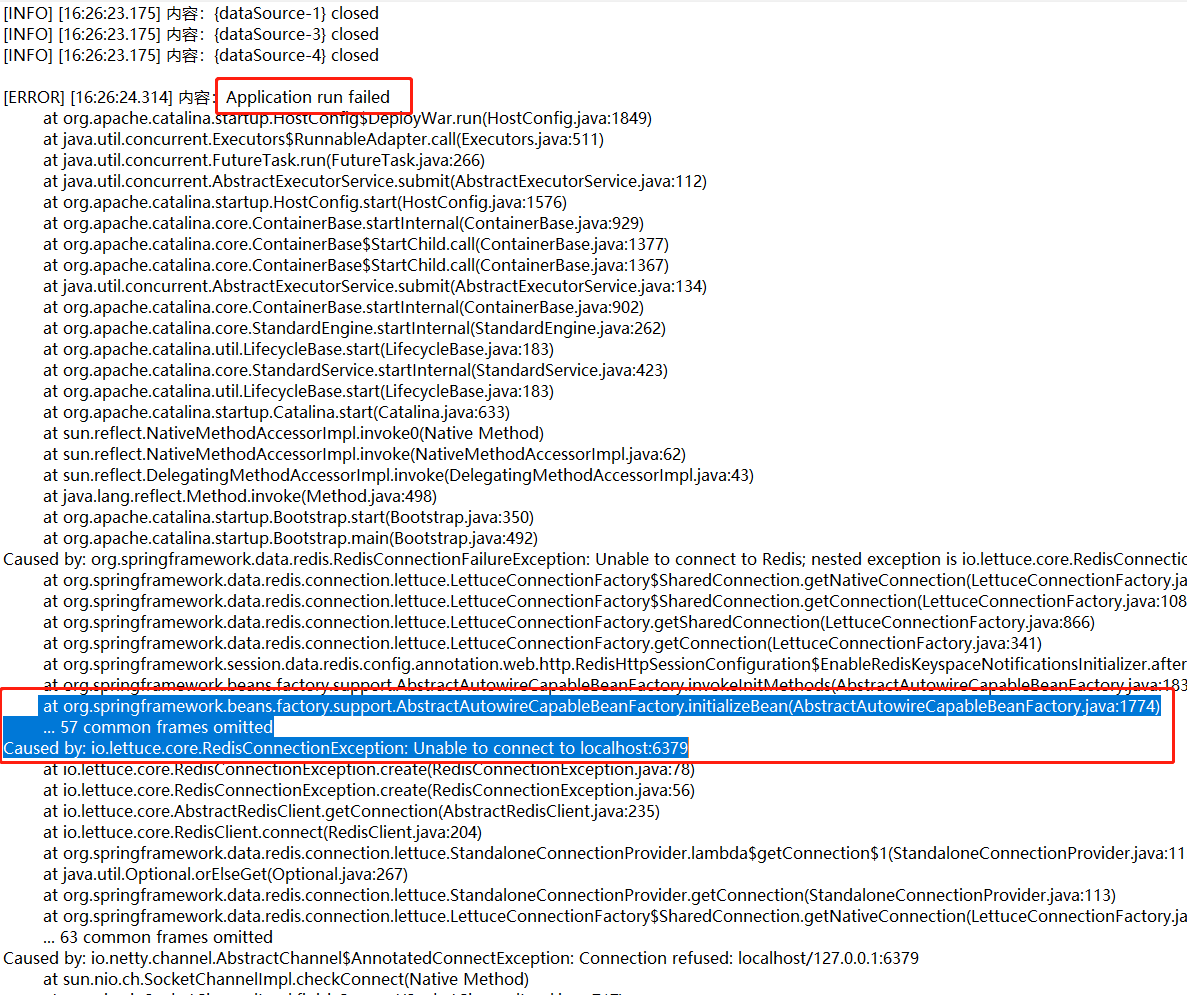
目前遇到过的情况为子项目启动报错，配置文件找不到，更改正确后重新启动就可以啦。如果遇到其他原因自己在网上找答案啦





## 2、Application run failed RedisConnectionException

common frames omitted Caused by: io.lettuce.core.RedisConnectionException: Unable to connect to localhost:6379

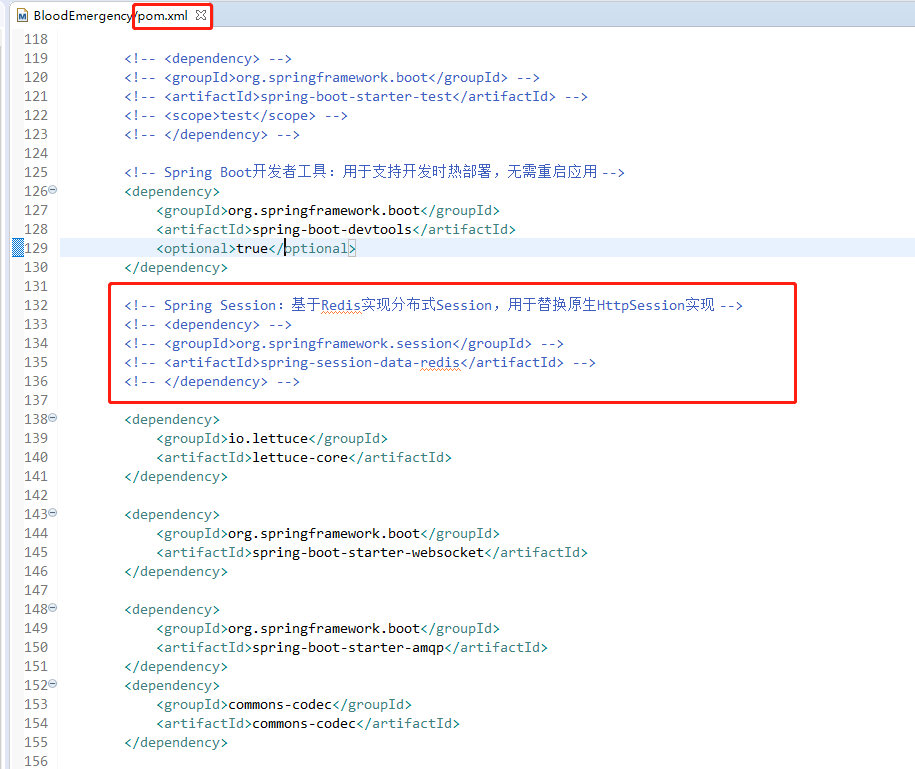
aused by: io.netty.channel.AbstractChannel$AnnotatedConnectException: Connection refused: localhost/127.0.0.1:6379 

问题原因：因为配置了redis连接，但是无法找到redis服务器，故报错

**解决方案：**

第一步：在application.yml中删除redis配置

第二步：注释掉redis绑定的依赖包（spring-session-data-redis），如果不删除，即使把配置注释掉，该依赖包仍然会使用默认配置（localhost）进行连接，以至于tomcat无法正常启动



## 4、No matching authentication protocol

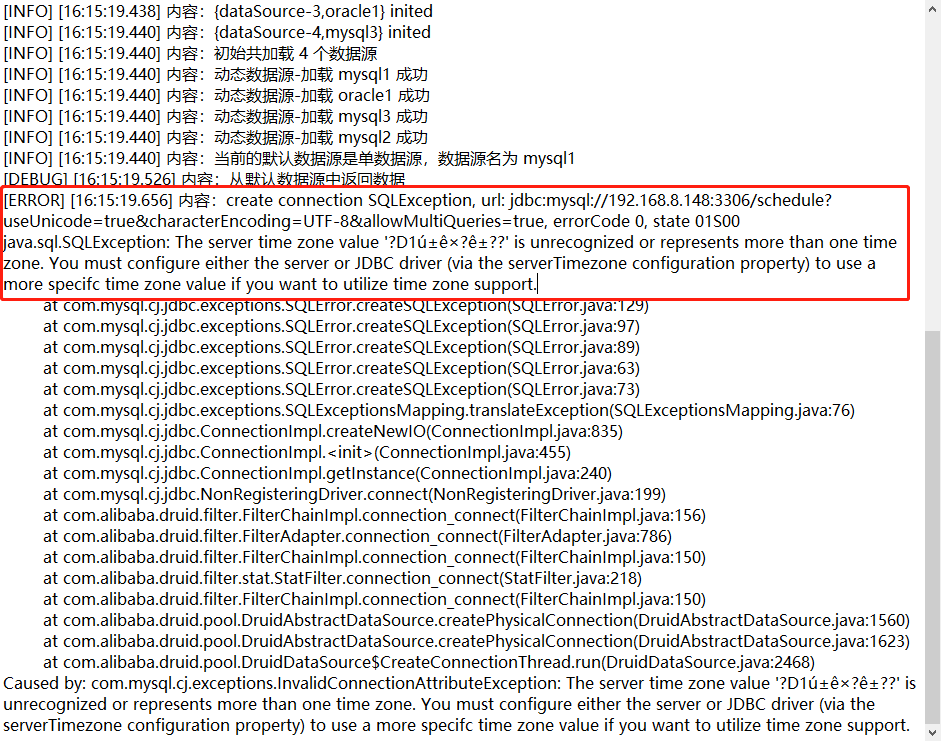


问题原因：由于应用导入的oracle Druid版本和使用的oracle版本不一致或不兼容导致

**解决方案：**更换对应的druid版本，之前使用的是oracle6，更换为oracle7就OK了

## 3、java.sql.SQLException: The server time zone value '?D1ú±ê×?ê±??' is unrecognized or represents more than one time zone.

You must configure either the server or JDBC driver (via the serverTimezone configuration property) to use a more specifc time zone value if you want to utilize time zone support.



**问题原因：**mysql数据库连接字符串问题

**解决方案：**在application.yml中mysql连接后面加上&serverTimezone=UTC就好了

例如：

jdbc:mysql://192.168.8.148:3306/schedule?useUnicode=true&characterEncoding=UTF-8&allowMultiQueries=true&serverTimezone=UTC