Linux(Ubuntu)基础使用

1. 常用命令

sudo(super user do):管理员权限执行命令

cd 进入磁盘目录（change directory）

mkdir 创建目录  (make directory)

ls 显示目录结构

ls -a 显示目录所有结构，包括隐藏文件

rm 移除文件 (remove)

rm -rf 递归移除文件及文件夹

pwd 显示文件当前路径

gedit 编辑指定文件【简单】

vi(m) 查看编辑指定文件【复杂】

1. 1.安装VMare Workstation

VMware\_workstation\_full\_12.5.0.11529.exe

1. 2.在VM中安装Ubuntu镜像

ubuntu-16.04.3-desktop-amd64.iso

Ubuntu一般做服务器（常用）  centOS与redHat为一个系列

1. 3.mysql数据库服务器搭建

**1.首先检查系统中是否已经安装了MySQL**

在终端里面输入 sudo netstat -tap | grep mysql

若没有反映，没有显示已安装结果，则没有安装。

graphic

若如下显示，则表示已经安装了。

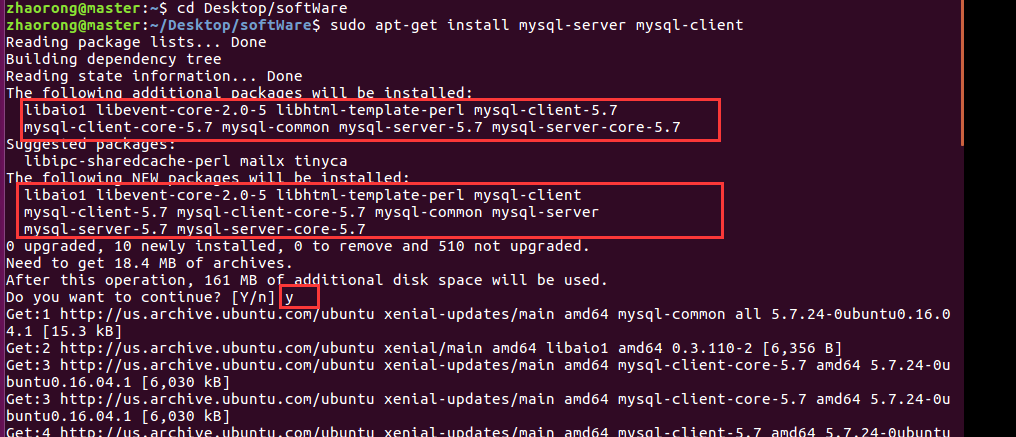
graphic

**2.如果没有安装，则安装MySQL。**

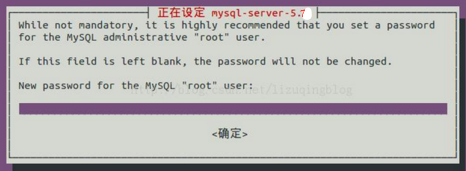
在终端输入sudo apt-get update

在终端输入 sudo apt-get install mysql-server mysql-client

运行结果如下所示:



**3.在此安装过程中会让你输入root用户(管理MySQL数据库用户，非Linux系统用户)密码，按照要求输入即可。输入完密码按tab键切换。如下所示：**



**4.测试安装是否成功：**

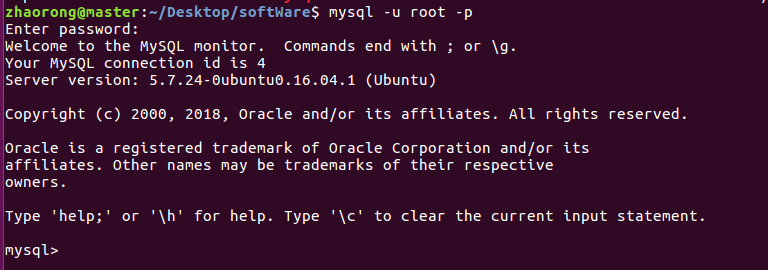
在终端里面输入 sudo netstat -tap | grep mysql

如下显示表示安装成功了。

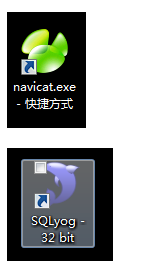
graphic

**5.也可以登录MySQL进行测试。**

在终端输入 mysql -uroot -p 接下来会提示你输入密码，输入正确密码(我的设为密码为root)，即可进入。如下所示：

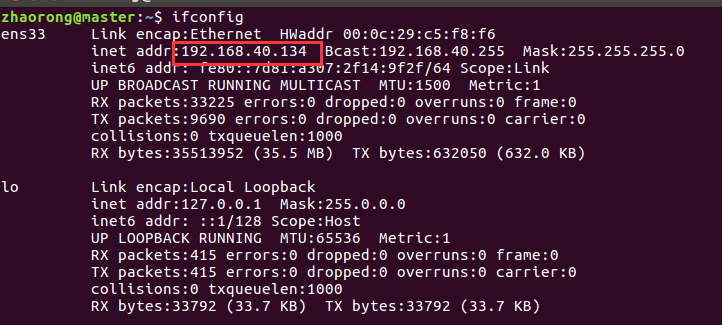


**6.给大家推荐两个远程挂载MySQL的软件，在Windows下远程操作Linux下的MySQL。navicat（主要是产看数据库），sqlyog（增删改查）。**

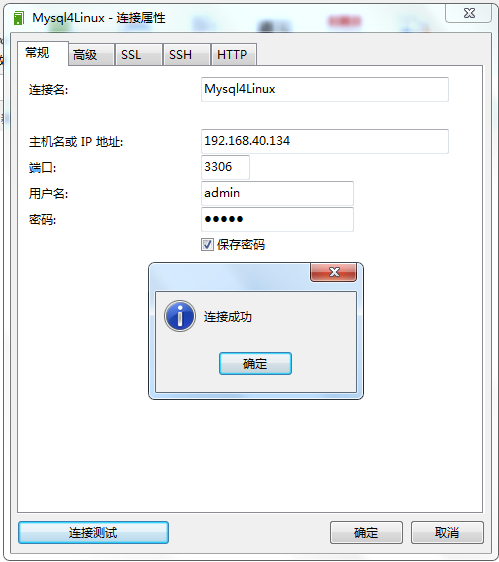


**7.在Windows下远程操作Linux下的MySQL测试**

 ①在终端输入 ifconfig 查看linux服务器的ip地址,效果如下所示：



② 使用navicat远程连接测试，如测试不成功，请授权用户，参考3.1



* 1. 3.1 navicat远程连接测试无法连接

1.登录mysql：

mysql -u 用户名 -p

2. 授权：

 mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'admin'@'%' IDENTIFIED BY 'admin' WITH GRANT OPTION;

3.修改生效：

mysql> FLUSH PRIVILEGES;

4.退出MySQL数据库：

mysql> quit;

-------------------------------------------------------------------------------------------------------

5.修改配置文件【备注：a修改，修改完成后 esc退出，输入 :wq保存退出】

sudo gedit /etc/mysql/mysql.conf.d/mysqld.cnf(命令无效：先移除sudo apt-get remove gedit，再安装sudo apt-get install gedit)

或

sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf

找到 "bind-address = 127.0.0.1" , 这一行要注释掉，只需在前面加个#，如下：

#bind-address = 127.0.0.1

6.关闭数据库

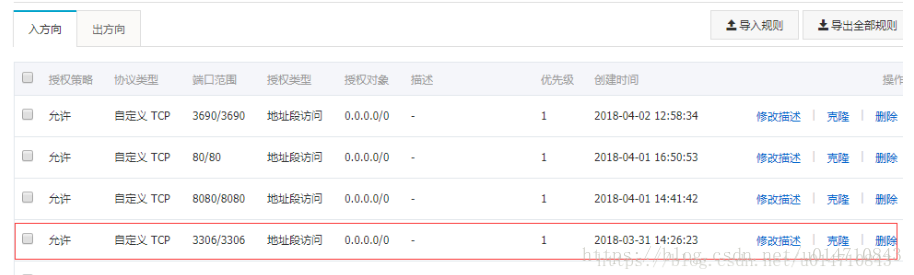
sudo /etc/init.d/mysql stop

7.启动数据库

sudo /etc/init.d/mysql start

-------------------------------------------------------------------------------------------------------------

8.登录阿里云=>控制台=>云服务器ECS=>网络和安全=>安全组，在入方向，点击配置规则可以看到下图，3306端口是访问服务器mysql的，没有的话就添加规则，端口范围选择 3306mysql,授权对象设置为0.0.0.0/0 允许所有ip访问。80端口是访问web的，22端口是远程连接服务器的。



1. 4.svn代码托管

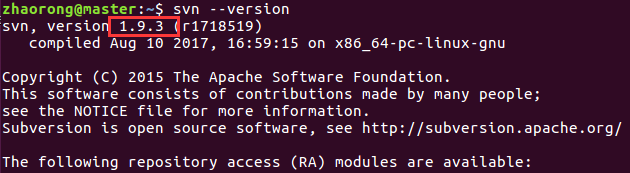
**1. 先查看是否已经安装了svn**

在终端输入 svn --version

如果没有安装svn则显示如下效果：

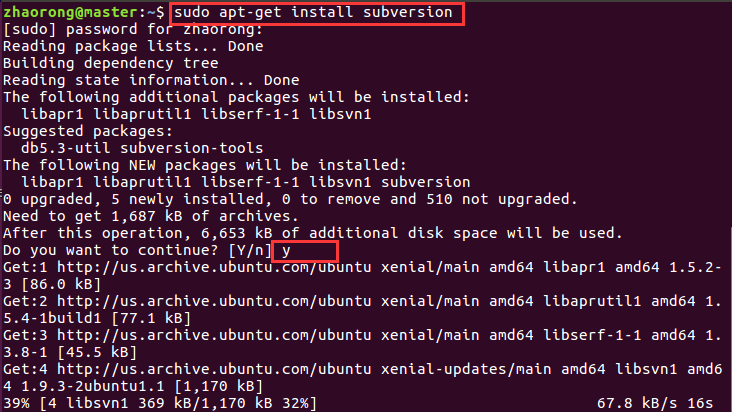
graphic

如果已安装svn则显示如下效果：



**2. 如未安装，则安装svn。**

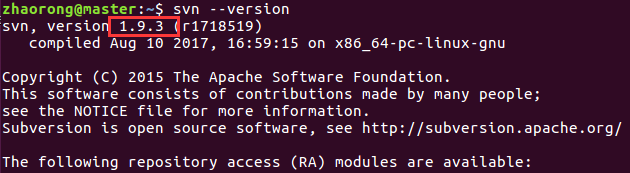
在终端输入 sudo apt-get install subversion,显示效果如下：



3.**查看是否安装成功**

在终端输入 svn --version

看到如下版本信息，则表示安装成功



* 1. 4.1创建版本库

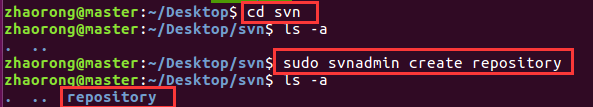
**1.创建版本库【完整路径/home/zhaorong/Desktop/svn/repository】**

在终端输入 mkdir svn 创建svn文件夹

graphic

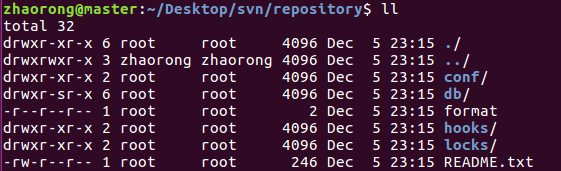
**2.创建目录repository利用svn命令指定为版本库**

用cd命令进入svn目录，在终端输入sudo svnadmin create repository



设置完成之后在repository下面会生产几个文件夹和文件conf、db、format等，主要的配置文件在conf文件夹下面：svnserve.conf，passwd，authz

graphic



**3.修改repository文件夹下面的内容，完善配置**

①修改con文件夹下面的svnserve.conf文件

在终端输入 sudo gedit /home/zhaorong/Desktop/svn/repository/conf/svnserve.conf

   #匿名用户可读

   anon-access = read（可改成none，即不可读）

   #权限用户可写

    auth-access = write

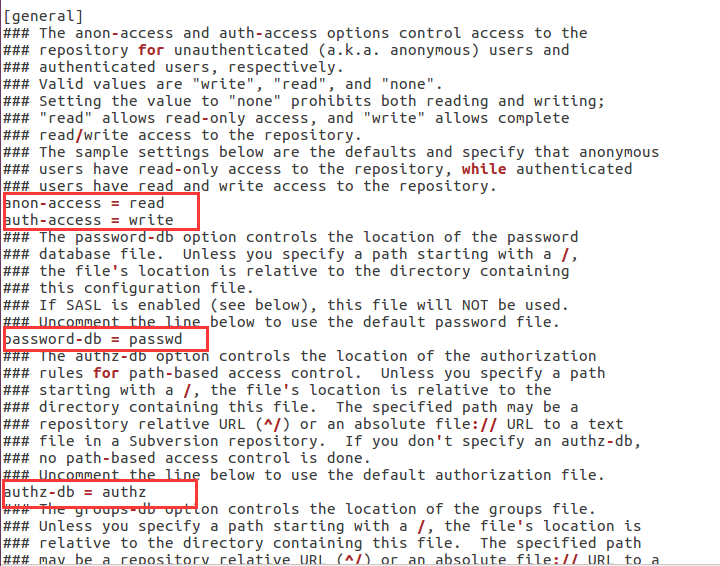
    #密码文件为passwd

    password-db = passwd

    #权限文件为authz

    authz-db = authz

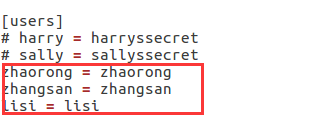
  （去掉前面的#，并且顶格，不然可能会报错）



 ②修改con文件夹下面的passwd文件，增加用户和用户密码

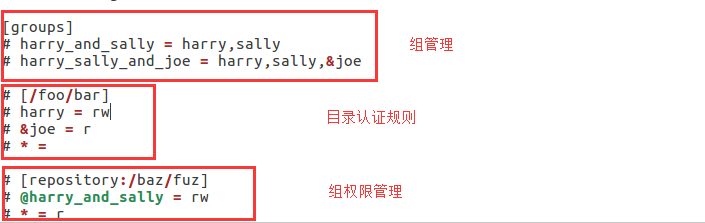
   格式：用户名=密码（明文）

在终端输入sudo gedit /home/zhaorong/Desktop/svn/repository/conf/passwd



 ③修改con文件夹下面的authz文件，增加用户组，目录认证，组权限

在终端输入sudo gedit /home/zhaorong/Desktop/svn/repository/conf/authz



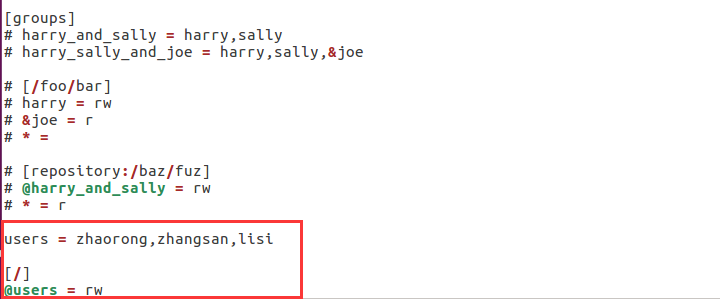
   如果组内有多个人可以逗号分隔如：组名owner，组员user1,user2则：owner=user1,user2

   对组的访问权限设置：rw，r，w。组名前面加@

   @owner=rw

   如果单用户设置则一个用户一行:

    user1=rw



    单库模式则设置为：[/] 本例子为单库模式-----------涉及访问路径

 ④修改db文件夹的权限，改为777。db文件夹是存储svn文件的地方

在终端输入  sudo chmod -R 777 /home/zhaorong/Desktop/svn/repository/db

**4. 启动svn版本库**

   在终端输入svnserve -d -r /home/zhaorong/Desktop/svn/repository

   -d：表示在后台运行 ，-r：指定服务器的根目录

**5.登录阿里云=>控制台=>云服务器ECS=>网络和安全=>安全组，在入方向，点击配置规则可以看到下图，3690端口是访问服务器SVN的，没有的话就添加规则，端口范围选择 3690 SVN,授权对象设置为0.0.0.0/0 允许所有ip访问。80端口是访问web的，22端口是远程连接服务器的。**



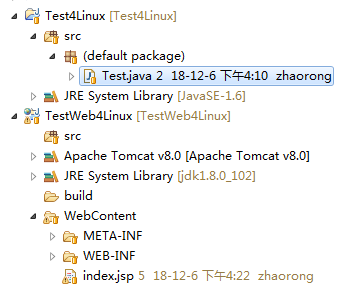
**6. 访问：通过svn://服务器ip/文件夹名字 从svn客户端访问服务器**

     默认svn协议。如果要配置http或者https则需要安装apache。

    本例因为是设置的[/]，则svn://ip地址/，例如 svn://192.168.40.134 就可以访问

**以eclipse共享代码为例：**

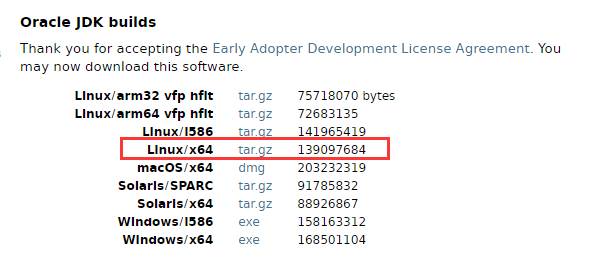
     项目右键-team-share project-svn - svn://192.168.40.134 -输入用户名与密码  -确认共享



1. 5.安装JDK

**1.下载**

    可以去官网下载：http://www.oracle.com/technetwork/java/javase/downloads/ea-jsp-142245.html

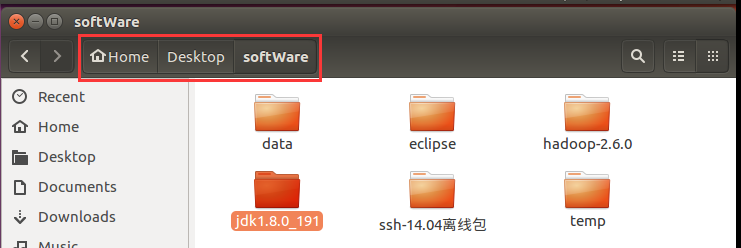


  下载完成后，如下：

graphic

**2. 解压（解压即后无需安装）**

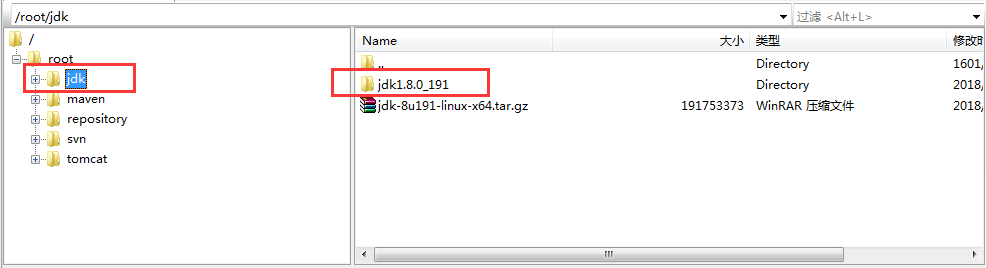
    【简单粗暴】直接拷贝拖拽到linux服务器某目录下，右键extract here解压到例如home/zhaorong/Desktop/software



    或

   【命令执行】

    通过windows操作系统上的SecureFX工具上传到某目录下，再通过命令解压



     打开xshell或SecureFX或终端，进入jdk的存放目录。cd /home/zhaorong/Desktop/software

     执行解压命令：sudo tar zxvf ./jdk-8u191-linux-x64.tar.gz

     执行权限命令：sudo chmod -R 777 /home/zhaorong/Desktop/softWare/jdk1.8.0\_191

**3. 配置环境变量**

   在/etc/profile.xml中配置环境变量，添加export JAVA\_HOME、path、classpath等，可命令vi或vim或gedit查看修改文件

  【打开编辑文件】sudo gedit /etc/profile

  【添加配置】打开/etc/profile文件在文件中添加如下配置，注意每一行后不要留空格

   export   JAVA\_HOME=/home/zhaorong/Desktop/softWare/jdk1.8.0\_191(pwd命令查看当前文件路径)

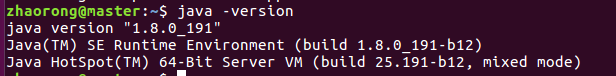
   export   CLASSPATH=$CLASSPATH:.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

   export   PATH=$JAVA\_HOME/bin:$PATH

   export   JRE\_HOME=$JAVA\_HOME/jre

  【重新加载，更新配置文件】source /etc/profile

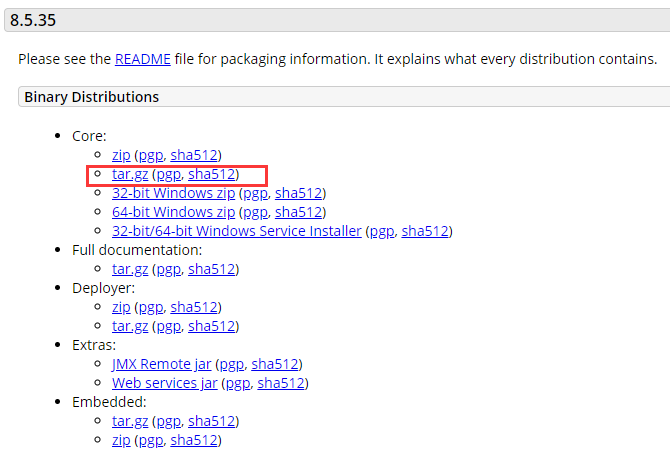
  【测试】java -version



1. 6.安装tomcat

**1.下载**

    可以去官网下载：https://tomcat.apache.org/download-80.cgi

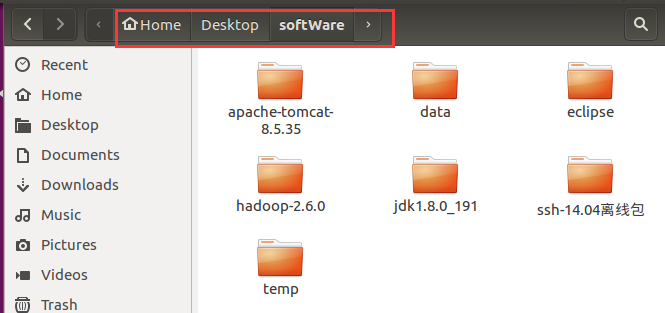


  下载完成后，如下：

graphic

**2. 解压（解压即后无需安装）**

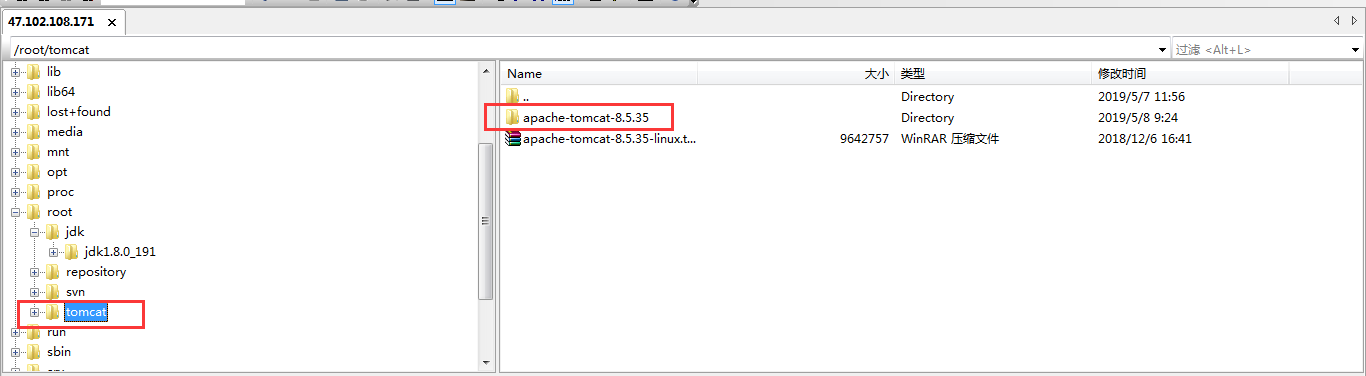
    【简单粗暴】直接拷贝拖拽到服务器某目录下，右键extract here解压到例如home/zhaorong/Desktop/software



    或

    【命令执行】

    通过windows操作系统上的SecureFX工具上传到某目录下，再通过命令解压



     打开xshell或SecureFXPortable或终端，进入tomcat的存放目录。cd /home/zhaorong/Desktop/software

     执行解压命令：sudo tar zxvf ./apache-tomcat-8.5.35-linux.tar.gz

     执行权限命令：sudo chmod -R 777 /home/zhaorong/Desktop/softWare/apache-tomcat-8.5.35

**3. 配置环境变量**

   在/bin/catalina.sh中配置环境变量，添加export JAVA\_HOME、JRE\_HOME，可命令vi或vim或gedit查看修改文件

  【打开编辑文件】sudo gedit /home/zhaorong/Desktop/softWare/apache-tomcat-8.5.35/bin/catalina.sh

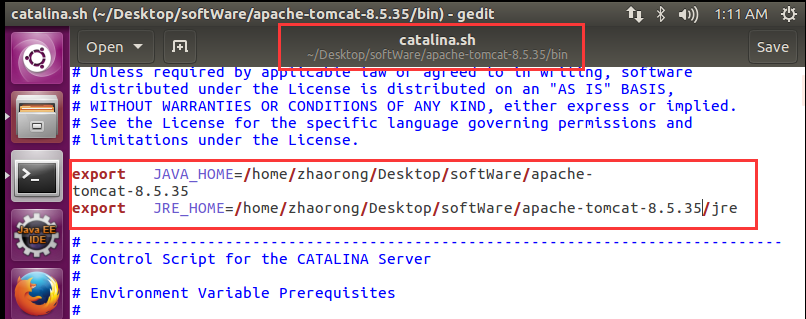
  【添加配置】打开/home/zhaorong/Desktop/softWare/apache-tomcat-8.5.35/bin/catalina.sh文件在文件中添加如下配置，注意每一行后不要留空格

    export   JAVA\_HOME=/home/zhaorong/Desktop/softWare/jdk1.8.0\_191

    export   CLASSPATH=$CLASSPATH:.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

    export   PATH=$JAVA\_HOME/bin:$PATH

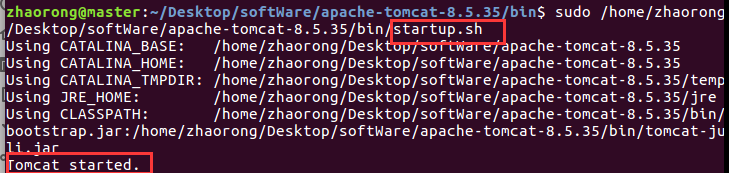
    export   JRE\_HOME=$JAVA\_HOME/jre



  【重新加载，更新配置文件】source /home/zhaorong/Desktop/softWare/apache-tomcat-8.5.35/bin/catalina.sh

**4. tomcat8安装后默认配置了服务。可以使用下面命令查看启动、停止**

  启动：sudo /home/zhaorong/Desktop/softWare/apache-tomcat-8.5.35/bin/startup.sh



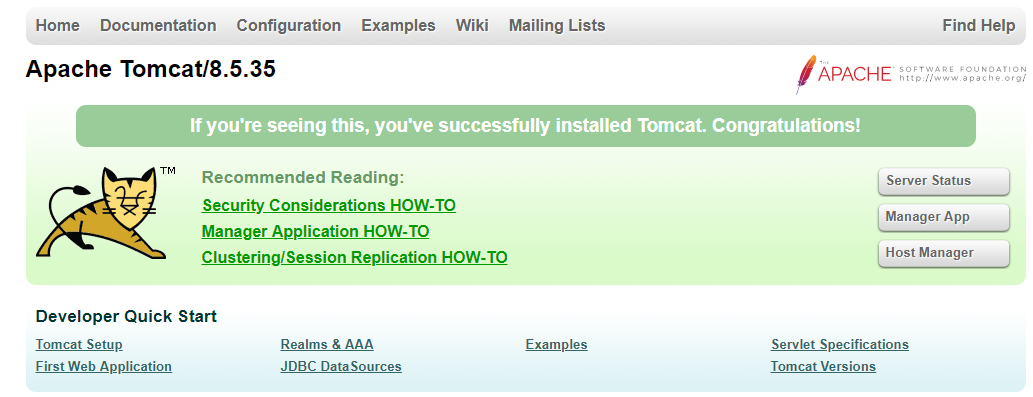
  停止：sudo /home/zhaorong/Desktop/softWare/apache-tomcat-8.5.35/bin/shutdown.sh

**5.登录阿里云=>控制台=>云服务器ECS=>网络和安全=>安全组，在入方向，点击配置规则可以看到下图，8080端口是访问服务器Tomcat的，没有的话就添加规则，端口范围选择8080 Tomcat,授权对象设置为0.0.0.0/0 允许所有ip访问。80端口是访问web的，22端口是远程连接服务器的。**



**6.路径测试**

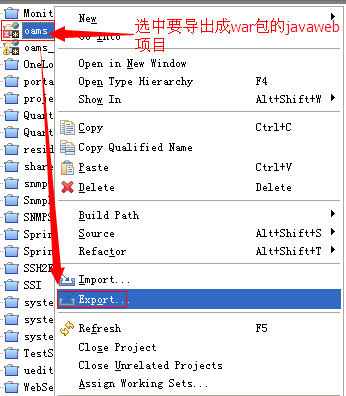
**http://192.168.40.134:8080/**

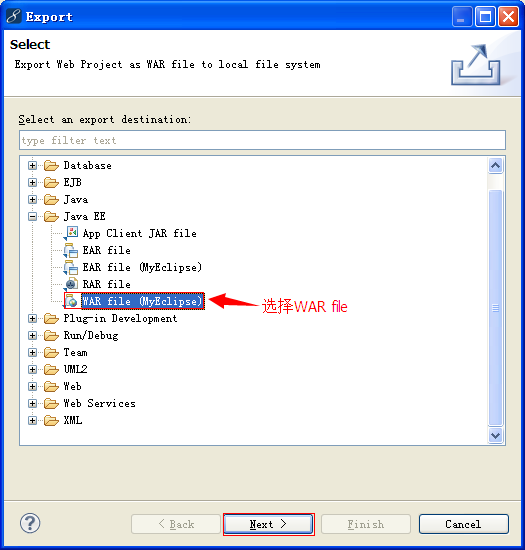


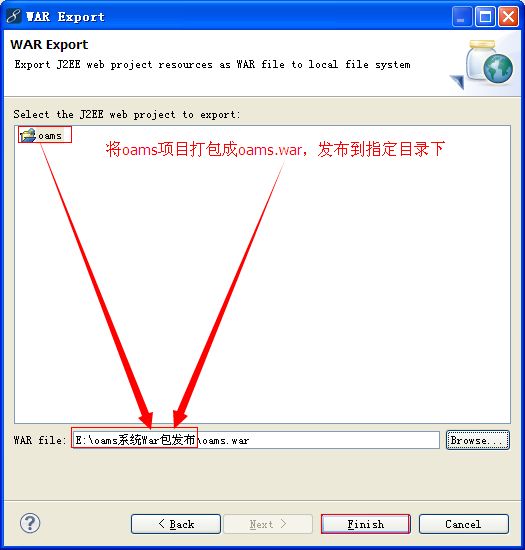
1. 7.将web应用部署到tomcat

**1.部署JavaWeb应用到Tomcat服务器**：就是将开放好的JavaWeb应用打包成war包，然后发布到tomcat服务器的webapps目录下

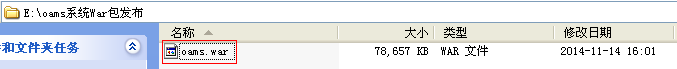
1.1、打包JavaWeb应用







　　项目打包成功，如下图所示：

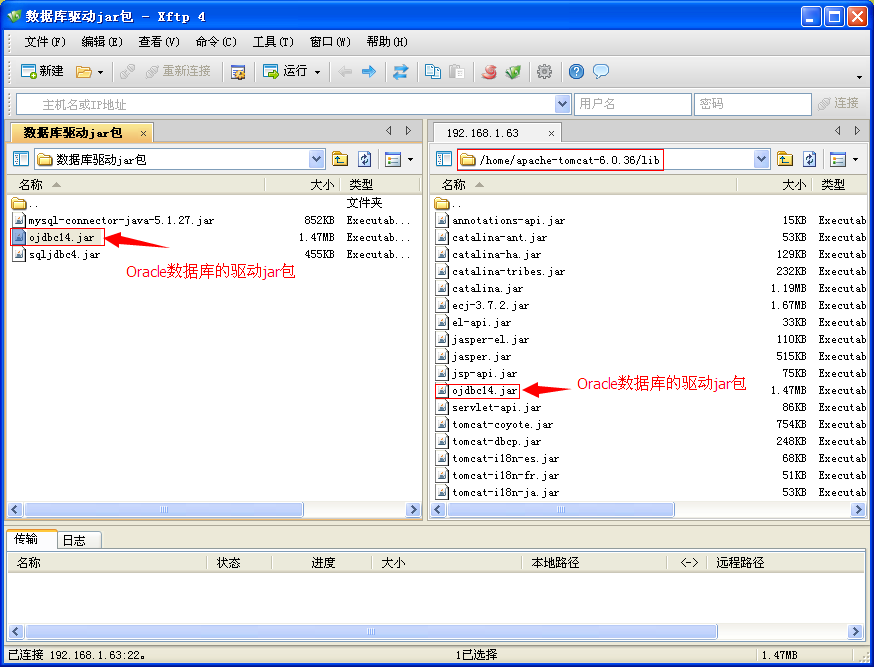


2.2、将war包发布到tomcat服务器的webapps目录下

　　将JavaWeb应用打包成war包之后，要将这个war包放到tomcat服务器的webapps目录下，由于要部署的javaWeb应用使用的是JNDI数据源，所以首先要把数据库对应的驱动jar包放到tomcat服务器的lib目录下。

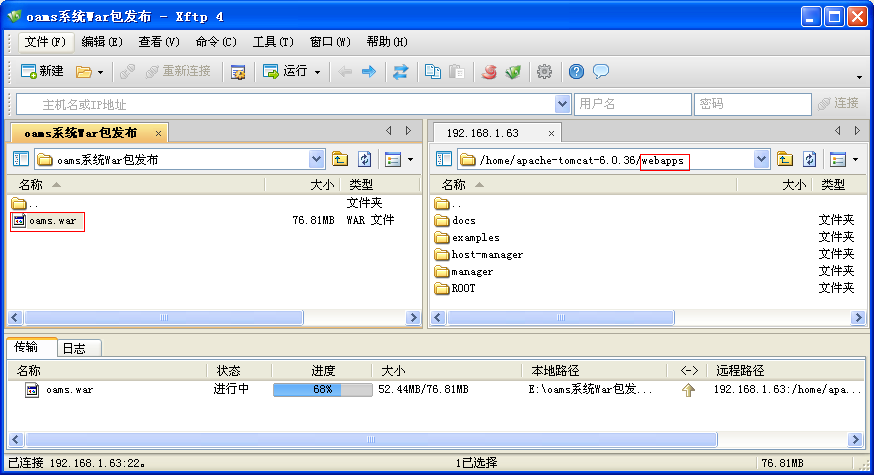
　　1、将Oracle数据库的驱动jar包上传到tomcat服务器的lib目录

　　如下图所示：

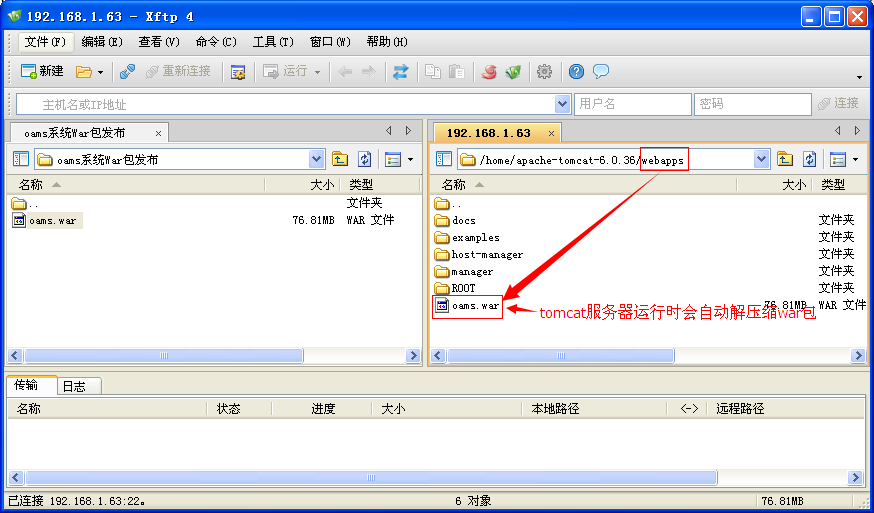


　　2、将打包好的war包上传到tomcat服务器的webapps目录

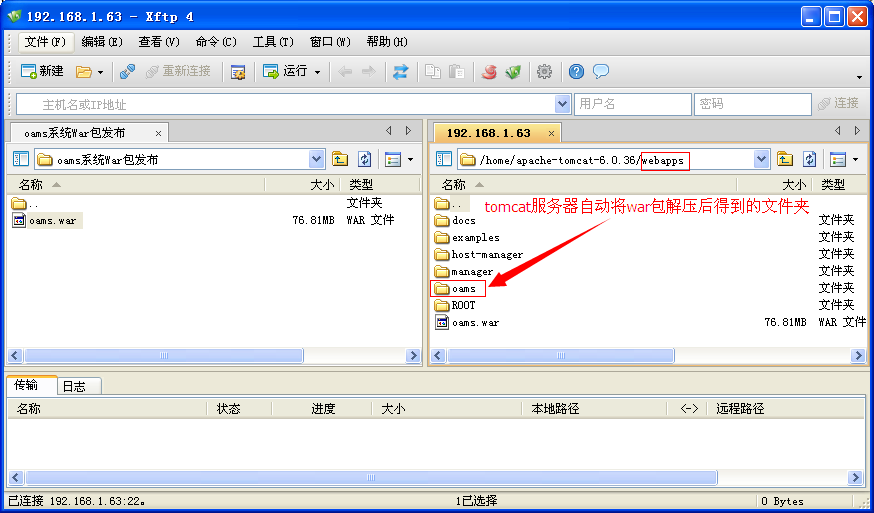
　　如下图所示：上传war包到webapps目录



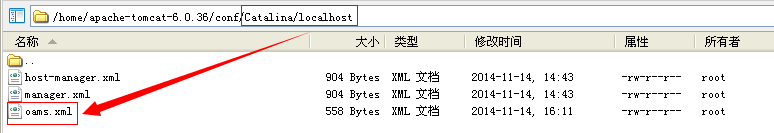
　　上传成功



　　Tomcat服务器自动解压缩war包



　　检查一下tomcat服务器是否自动在conf目录下创建了Catalina/localhost目录，在看看localhost目录下是否有tomcat服务器自动创建的oams.xml文件

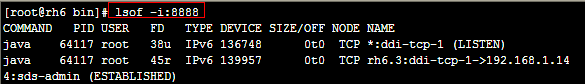


五、Linux下安装Tomcat服务器使用到的命令

5.1、linux查看端口被哪个进程占用

lsof -i:端口号

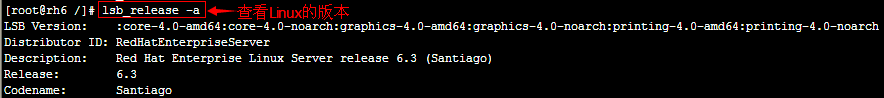
　　例如：



5.2、查看Linux服务器的版本

lsb\_release -a

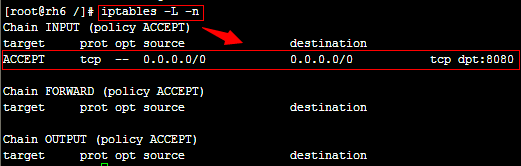
　　例如：



5.3、查看当前iptables(防火墙)规则

　　可使用"iptables –L –n"查看当前iptables规则

　　例如：



5.4、添加指定端口到防火墙中

　　iptables -I INPUT -p 协议 --dport 端口号 -j ACCEPT

　　例如：

1 iptables -I INPUT -p udp --dport 161 -j ACCEPT

2 iptables -I INPUT -p tcp --dport 8080 -j ACCEPT

1. 8.安装redis

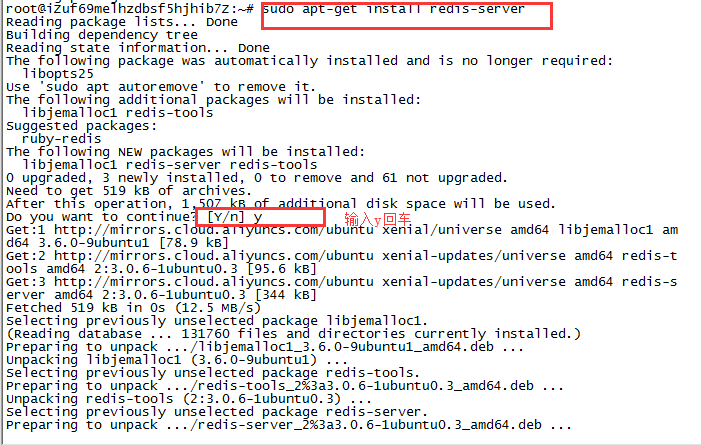
**0、更新安装源**

这一步并不是必须的，可以直接通过第二步的命令安装redis，如果报错说没有安装包的话，可以执行这个命令更新安装源。

sudo apt-get update

**1、安装redis**

sudo apt-get install redis-server



**2、修改配置文件**

sudo vim /etc/redis/redis.conf

**redis默认只能本机访问，需要修改一些配置**

注释掉这一句

 # bind 127.0.0.1

或者是改为

bind 0.0.0.0

**配置密码**：取消注释并改为自己需要的密码

requirepass 123456

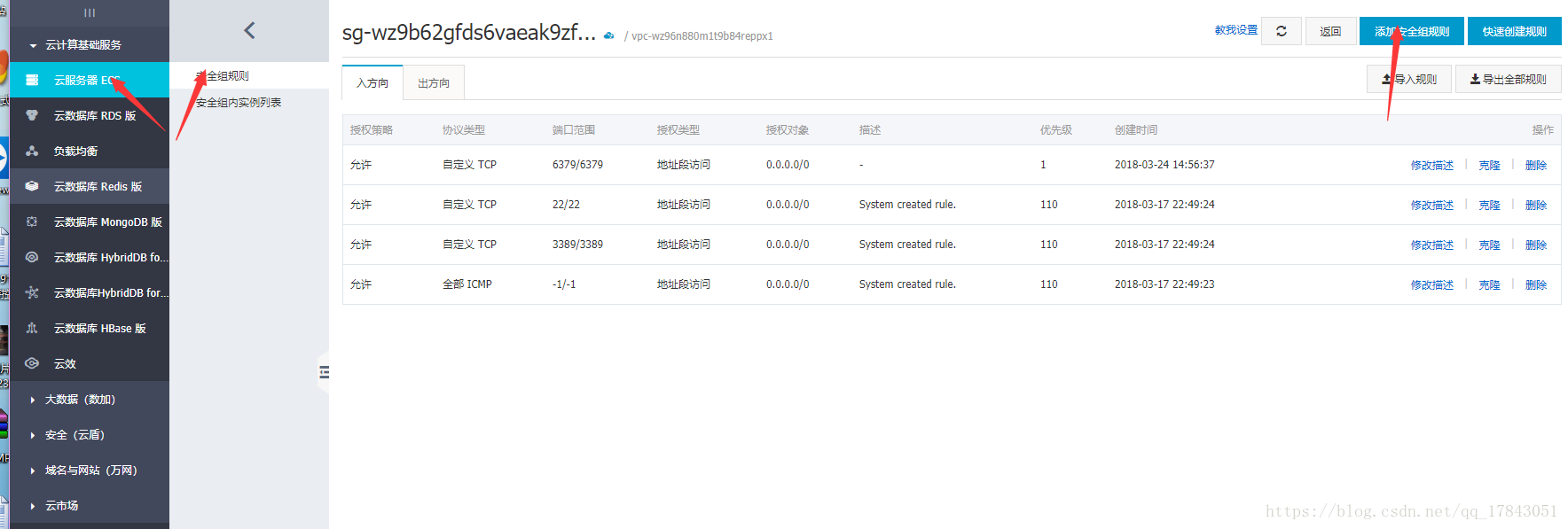
**登陆测试**

sudo service redis restart

redis-cli -a 123456 -h 47.102.108.171(地址)

**3、放开端口**

原以为大功告成，但是没有想到没有能够连接成功，猜测应该是防火墙的问题。后面经过测试，发现是阿里云ecs服务器本身有一个安全规则，没有把6379端口放开，所以打开6379端口：





重新测试连接，ok。

* 1. windows环境下测试

1.下载并解压D:\redis-64.2.8.2101

2.D:\redis-64.2.8.2101目录下，cmd运行服务器  redis-server.exe redis.windows.conf

3.【本机测试】双击打开客户端访问 redis-client.exe

    set username "redis" --存值

    get username --取值

    del username --移除值

4.【云服务器redis测试】D:\redis-64.2.8.2101目录下，cmd运行

   redis-cli -a 123456(密码) -h 47.102.108.171(地址)

5【java代码测试云服务器】

   1.在pom.xml导入redis与json的依赖 jedis.jar fastjson.jar

      <!-- redis -->

      <dependency>

        <groupId>redis.clients</groupId>

        <artifactId>jedis</artifactId>

        <version>2.10.0</version>

    </dependency>

    <!-- fastjson -->

    <dependency>

        <groupId>com.alibaba</groupId>

        <artifactId>fastjson</artifactId>

        <version>1.2.47</version>

    </dependency>

  2.在spring环境applicationContext.xml中管理jedis

     <!-- 6.redis分布式缓存框架 -->

     <bean id=*"jedis"* class=*"redis.clients.jedis.Jedis"*>

       <!-- <constructor-arg name="host" value="47.102.108.171"></constructor-arg>

       <constructor-arg name="port" value="6379"></constructor-arg> -->

       <constructor-arg name=*"host"* value=*"redisServer://password:123456@47.102.108.171:6379"*/>

     </bean>

  3.注入给service层，并运行测试

    @Resource(name="jedis")

**private** Jedis jedis;

   @Override

**public** Emp getEmpByEmpno(Integer empno) {

    String json = jedis.get(String.*valueOf*(empno));

    System.***out***.println("redis缓存中：" + json);

    Emp emp = **null**;

    //查询数据，当缓存中存在数据时，取缓存数据，效率高

**if**(json != **null**)

    {

      JSONObject jsonObj= JSONObject.*parseObject*(json);//[{"empno":60,"ename":"李文龙","job":"驱蚊达人1","salary":10.0}]

      System.***out***.println("redis缓存查询中...." + jsonObj.toJavaObject(Emp.**class**));

      emp = jsonObj.toJavaObject(Emp.**class**);

    }

    //查询数据，当缓存中不存在数据为null时，查询数据库取数据，效率低

**else**

    {

      emp = mapper.selectEmpByEmpno(empno);

      System.***out***.println("数据库查询中，并保存到redis缓存...." + JSONObject.*toJSONString*(emp));

      //将当次查询的信息保存到redis缓存

      jedis.set(String.*valueOf*(empno), JSONObject.*toJSONString*(emp));

    }

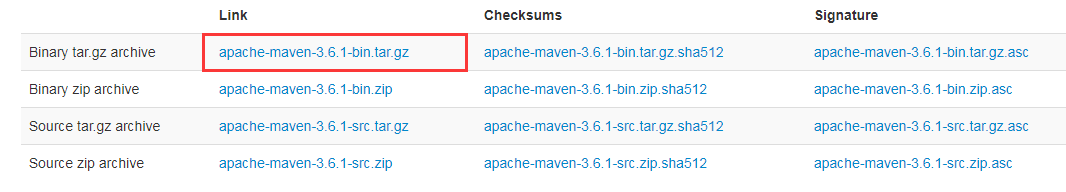
**return** emp;

  }

1. 9.安装maven

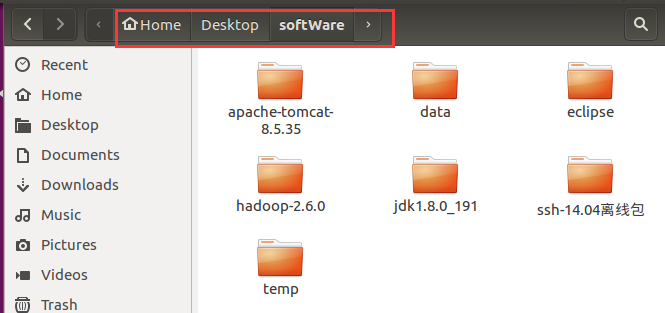
**1.下载maven**

   http://maven.apache.org/download.cgi



**2. 解压（解压即后无需安装）**

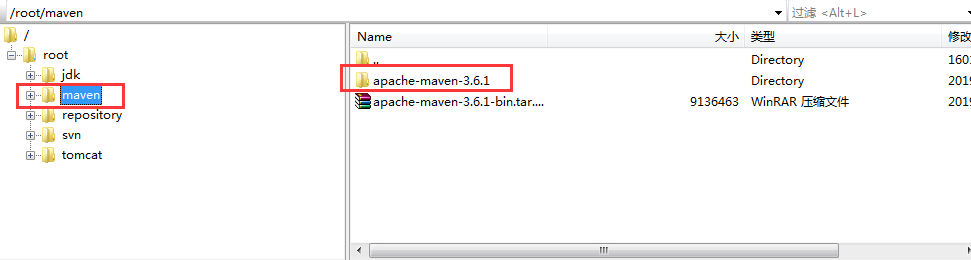
    【简单粗暴】直接拷贝拖拽到服务器某目录下，右键extract here解压到例如home/zhaorong/Desktop/software



    或

    【命令执行】

    通过windows操作系统上的SecureFX工具上传到某目录下，再通过命令解压



     打开xshell或SecureFXPortable或终端，进入tomcat的存放目录。cd /root/maven

     执行解压命令：sudo tar zxvf ./apache-maven-3.6.1-bin.tar.gz

     执行权限命令：sudo chmod -R 777 /root/maven/apache-maven-3.6.1

**3. 配置环境变量，编辑主目录下的 .bashrc 文件**

    【打开文件】sudo vim ~/.bashrc（a 编辑文件，esc退出编辑  :wq保存并退出）

    【编辑文件】在文件末尾添加，注意每一行后不要留空格

        export MAVEN\_HOME=$root/maven/apache-maven-3.6.1

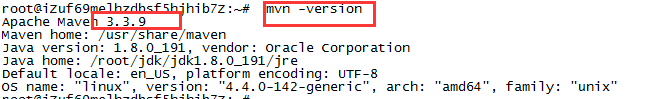
        export PATH=$PATH:${MAVEN\_HOME}/bin



    【更新配置，使配置立即生效】source ./.bashrc

**4.查看maven版本**

    在终端输入： mvn -version



**5.配置Maven镜像**

   编辑 MAVEN\_HOME，conf 文件夹下的 settings.xml，cd进入/root/maven/apache-maven-3.6.1/conf

       【打开文件】 sudo vim ./settings.xml 或直接 sudo vim /root/maven/apache-maven-3.6.1/conf/settings.xml

       【修改文件】找到mirrors镜像节点，增加mirror节点，如下所示：

       <mirror>

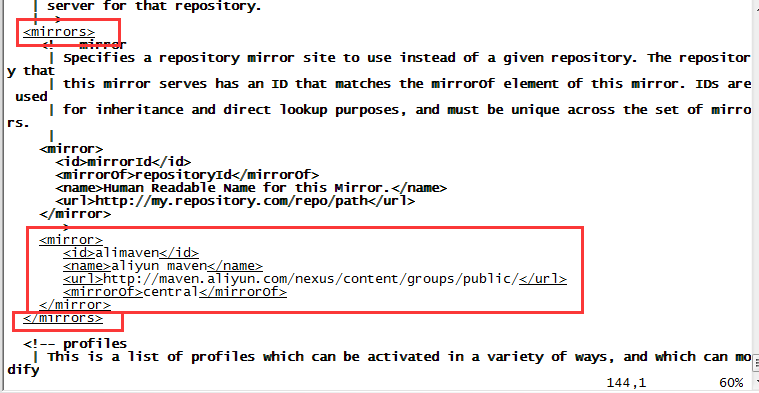
          <id>alimaven</id>

          <name>aliyun maven</name>

          <url>http://maven.aliyun.com/nexus/content/groups/public/</url>

          <mirrorOf>central</mirrorOf>

        </mirror>



       【更新配置】source /root/maven/apache-maven-3.6.1/conf/settings.xml