# AAMS全员考核管理系统部署手册

1. **说明**

本文档提供给系统维护人员使用。按照此文档可以完成部署AAMS应用，并使其达到设计要求。

1. **服务器要求**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 用途 | 数量 | 系统 | CPU | 内存 | 硬盘 | 带宽 | 安装软件包 |
| 应用服务器 | 1 | Linux | 2.0Ghz  1CPU  2核 | 4G | 100G | 10Mbps | Nginx、tomcat、redis、java |
| 数据库服务器 | 1 | Linux | 2.0Ghz  1CPU  2核 | 4G | 200G | 10Mbps | mysql |

1. **服务器配置**
2. 服务器安装系统：安装linux发行版本：ubuntu server lts长期支持版
3. 服务器安装软件

ufw防火墙

Nginx反向代理和负载均衡服务器

Redis缓存服务器，用来加快AAMS系统的访问速度

Java jre运行时

Tomcat Web容器，用来运行AAMS打包后的war包

Mysql 数据库，用来存储日常运行过程中产生的数据

1. **ufw安装**

1. 安装：apt-get ufw
2. 启用：ufw enable
3. 开启：80端口、22端口等
4. **Redis安装配置**
   * + - 1. **在**[**Ubuntu**](http://www.linuxidc.com/topicnews.aspx?tid=2) **Linux终端中安装Redis服务器端**

sudo apt-get install redis-server

安装完成后，Redis服务器会自动启动

* + - * 1. **//在终端中检查Redis服务器系统进程**

ps -aux|grep redis

* + - * 1. **在终端中通过启动命令检查Redis服务器状态**

netstat -nlt|grep 6379

显示: tcp 0 0 127.0.0.1:6379 0.0.0.0:\* LISTEN

//通过启动命令检查Redis服务器状态

sudo /etc/init.d/redis-server status

显示: redis-server is running

* + - * 1. **使用Redis的访问账号**

默认情况下，访问Redis服务器是不需要密码的，为了增加安全性我们需要设置Redis服务器的访问密码。设置访问密码为redis。

用vi打开Redis服务器的配置文件redis.conf

~ sudo vi /etc/redis/redis.conf

#取消注释requirepass

requirepass redis

* + - * 1. **让Redis服务器被远程访问**

默认情况下，Redis服务器不允许远程访问，只允许本机访问，所以我们需要设置打开远程访问的功能。

用vi打开Redis服务器的配置文件redis.conf

~ sudo vi /etc/redis/redis.conf

#注释bind

#bind 127.0.0.1

修改后，重启Redis服务器。

~ sudo /etc/init.d/redis-server restart

Stopping redis-server: redis-server.

Starting redis-server: redis-server.

* + - * 1. /**/检查Redis服务器占用端口**

~ netstat -nlt|grep 6379

tcp 0 0 0.0.0.0:6379 0.0.0.0:\* LISTEN

我们看到从之间的网络监听从 127.0.0.1:6379 变成 0 0.0.0.0:6379，表示Redis已经允许远程登陆访问。

1. **java 运行时jre安装**
2. **登录Linux，切换到root用户**

su root 获取root用户权限，当前工作目录不变(需要root密码)或sudo -i 不需要root密码直接切换成root（需要当前用户密码）

1. **在usr目录下建立java安装目录**

cd /usr

mkdir java

1. **将jdk-8u60-linux-x64.tar.gz拷贝到java目录下**

cp /mnt/hgfs/linux/jdk-8u60-linux-x64.tar.gz /usr/java/

1. **解压jdk到当前目录**

tar -zxvf jdk-8u60-linux-x64.tar.gz

得到文件夹 jdk1.8.0\_60

1. **编辑配置文件，配置环境变量**

vim /etc/profile

添加如下内容：JAVA\_HOME根据实际目录来

JAVA\_HOME=/usr/java/jdk1.8.0\_60

CLASSPATH=$JAVA\_HOME/lib/

PATH=$PATH:$JAVA\_HOME/bin

export PATH JAVA\_HOME CLASSPATH

1. **重启机器或执行命令 ：source /etc/profile**

sudo shutdown -r now

1. **1.8查看安装情况**

java -version

java version "1.8.0\_60"

Java(TM) SE Runtime Environment (build 1.8.0\_60-b27)

Java HotSpot(TM) Client VM (build 25.60-b23, mixed mode)

可能出现的错误信息：

bash: ./java: cannot execute binary file

出现这个错误的原因可能是在32位的操作系统上安装了64位的jdk，

查看jdk版本和Linux版本位数是否一致。

查看你安装的Ubuntu是32位还是64位系统：

sudo uname --m

i686 //表示是32位

x86\_64 // 表示是64位

1. **Nginx安装配置**
2. **下载pcre**

官网下载：http://www.pcre.org/

# wget http://sourceforge.net/projects/pcre/files/pcre/8.35/pcre-8.35.tar.gz/download

# cd /pcre-8.35

1. **下载purge模块（用于删除Nginx缓存）**

# wget http://labs.frickle.com/files/ngx\_cache\_purge-2.1.tar.gz  
# tar zxvf ngx\_cache\_purge-2.1.tar.gz

1. **安装zlib**

# wget http://prdownloads.sourceforge[**.NET**](http://lib.csdn.net/base/dotnet)/libpng/zlib-1.2.8.tar.gz?download

# tar zxvf zlib-1.2.8.tar.gz

# cd zlib-1.2.8

# ./configure --prefix=/usr/local/zlib

# make && make install

1. **安装openssl**

# wget http://www.openssl.org/source/openssl-1.0.1h.tar.gz

# cd openssl-1.0.1h.tar.gz

# tar zxfv openssl-1.0.1h.tar.gz

# cd openssl-1.0.1h

# ./config   
# ./config --prefix=/usr/local/openssl  
# make && make install

1. **安装gcc-c++**  
   # yum install -y gcc-c++
2. **下载最新的nginx稳定版进行安装**  
   # wget http://nginx.org/download/nginx-1.7.2.tar.gz  
   # tar zxvf nginx-1.7.2.tar.gz  
   # ./configure --with-pcre=/data/software/pcre-8.35 --add-module=../ngx\_cache\_purge-2.1 --prefix=/usr/local/nginx --with-http\_stub\_status\_module --with-http\_ssl\_module --with-zlib=/usr/local/zlib --with-openssl=/data/software/openssl-1.0.1h

安装：make && make install

1. **Nginx反向代理和负载均衡配置**

1. **通过Nginx做反向代理**

首先，需要安装nginx

sudo apt-get install nginx

然后，需要对Nginx进行配置   
cd /etc/nginx/sites-available/   
这个目录下面默认会有一个default的配置文件，内容如下   
image   
它是在80端口进行监听。你可以直接修改这个文件，例如我上图中其实已经设置了将所有请求转发（proxy\_pass）到之前我们启动的那个网站（8888端口）。   
我们也可以自己创建一个独立的配置文件，另外指定一个端口   
cat > nancydemo

然后输入如下的内容

server {

listen 81;

server\_name yourdomainname.com;

root /var/www/nancydemo;

location /Content/ {

alias /var/www/nancydemo/Content/;

location ~\* \.(jpg|jpeg|png|gif|ico|css|js|ttf)$ {

expires 365d;

}

}

location / {

proxy\_pass http://127.0.0.1:8888;

}

}

按下CTRL +D保存这个文件，然后重新加载一下配置信息，就可以在81端口访问到原先8888端口的内容了

sudo /etc/init.d/nginx reload

1. **启用两台Web服务器，通过nginx做负载均衡**

首先通过tomcat启动两个AAMS系统，端口分别为8888和9999。

编辑nancydemo这个配置文件

upstream xizhang{

server 127.0.0.1:8888;

server 127.0.0.1:9999;

}

server {

listen 81;

server\_name yourdomainname.com;

root /var/www/nancydemo;

location /Content/ {

alias /var/www/nancydemo/Content/;

location ~\* \.(jpg|jpeg|png|gif|ico|css|js|ttf)$ {

expires 365d;

}

}

location / {

proxy\_pass http://xizhang;

}

}

image

配置完成之后，我们需要重新加载一下配置信息

sudo /etc/init.d/nginx reload

测试结果表明每秒请求数明显上升

image

我们可以在服务器中看到有两个nginx进程，以及mono的进程。

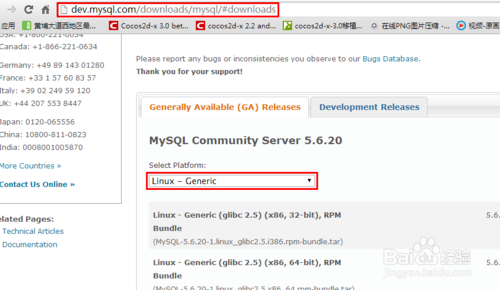
image

关于使用nginx实现负载均衡，还有一些细节，有兴趣可以参考  <http://nginx.org/en/docs/http/load_balancing.html>

nginx还可以作为静态资源服务器，实现动静分离（并且可以缓存起来），进一步提高Web服务器的性能和吞吐量。这个话题就不过多展开了，有兴趣的朋友们可以参考官方的文档

1. **Mysql数据库安装**
2. **到mysql官网下载mysql编译好的二进制安装包**

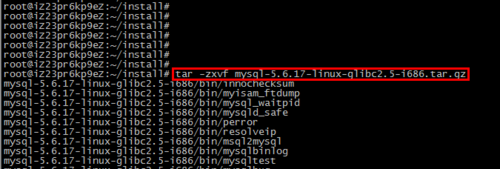
在下载页面Select Platform:选项选择linux-generic，然后把页面拉到底部，64位系统下载Linux - Generic (glibc 2.5) (x86, 64-bit)，32位系统下载Linux - Generic (glibc 2.5) (x86, 32-bit)





1. **解压32位安装包**

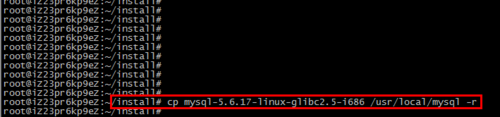
进入安装包所在目录，执行命令：tar mysql-5.6.17-linux-glibc2.5-i686.tar.gz



1. **复制解压后的mysql目录到系统的本地软件目录**

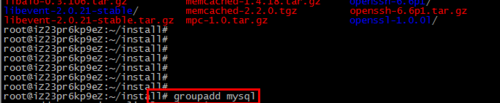
执行命令：cp mysql-5.6.17-linux-glibc2.5-i686 /usr/local/mysql -r

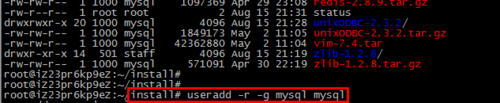
注意：目录结尾不要加/



1. **添加系统mysql组和mysql用户**

执行命令：groupadd mysql和useradd -r -g mysql mysql





1. **安装数据库**

进入安装mysql软件目录：执行命令 cd /usr/local/mysql

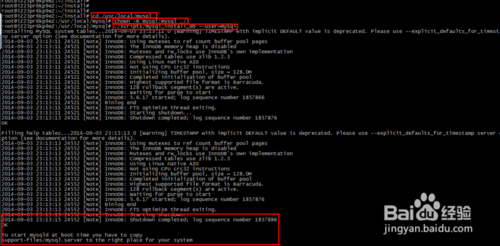
修改当前目录拥有者为mysql用户：执行命令 chown -R mysql:mysql ./

安装数据库：执行命令 ./scripts/mysql\_install\_db --user=mysql

修改当前目录拥有者为root用户：执行命令 chown -R root:root ./

修改当前data目录拥有者为mysql用户：执行命令 chown -R mysql:mysql data

到此数据库安装完毕



1. **启动mysql服务和添加开机启动mysql服务**

添加开机启动：执行命令cp support-files/mysql.server /etc/init.d/mysql，把启动脚本放到开机初始化目录

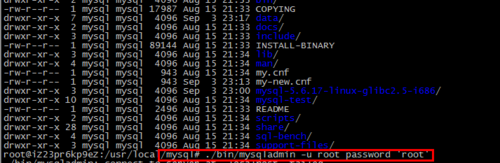
启动mysql服务：执行命令service mysql start

执行命令：ps -ef|grep mysql 看到mysql服务说明启动成功，如图



1. **修改mysql的root用户密码，root初始密码为空的**

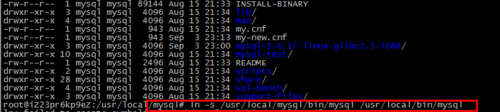
执行命令：./bin/mysqladmin -u root password '密码'



1. **把mysql客户端放到默认路径**

ln -s /usr/local/mysql/bin/mysql /usr/local/bin/mysql

注意：建议使用软链过去，不要直接包文件复制，便于系统安装多个版本的mysql



1. **Mysql主从数据库架构配置**
2. **master配置（主库）**

1.修改[**MySQL**](http://lib.csdn.net/base/mysql)配置文件

sudo vi /etc/[**mysql**](http://lib.csdn.net/base/mysql)/my.cnf

在[mysqld]模块中添加以下配置（默认是被注释掉的）

server-id               = 1

log\_bin                 = /var/log/mysql/mysql-bin.log

binlog\_do\_db            = yy\_test

binlog\_ignore\_db        = mysql

2.重启mysql

3.登录Mysql,执行  如下sql

mysql>show master status;  
   +------------------+----------+--------------+------------------+  
   | File             | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |  
   +------------------+----------+--------------+------------------+  
   | mysql-bin.000002 |      445 |              |                  |  
   +------------------+----------+--------------+------------------+  
   1 row in set (0.00 sec)

注：执行完此步骤后不要再操作主服务器MYSQL，防止主服务器状态值变化

记录    File  和   Position的值;

1. **配置slave（从库）**

1.修改mysql配置文件

sudo vi /etc/mysql/my.cnf

在[mysqld]模块中添加以下配置（默认是被注释掉的）

server-id      =  2

log\_bin                 = /var/log/mysql/mysql-bin.log

replicate\_do\_db         = yy\_test

replicate\_ignore\_db     = mysql

2.重启mysql

3.slave进入mysql,执行以下sql

stop salve;

4.设置从master复制日志配置：

change master to master\_host='192.168.1.108',master\_user='root',master\_password='123',master\_log\_file='mysql-bin.000002' ,master\_log\_pos=445; （mast

er\_log\_file和master\_log\_pos  是以上查询的值）

5.在salve上执行以下sql

  start slave;

  SHOW SLAVE STATUS;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

              Slave\_IO\_State: Waiting for master to send event

              Master\_Host: 192.168.1.108  //主服务器地址

              Master\_User: root   //授权帐户名，尽量避免使用root

              Master\_Port: 3306    //[**数据库**](http://lib.csdn.net/base/mysql)端口，部分版本没有此行

              Connect\_Retry: 60

              Master\_Log\_File: mysql-bin.000002

              Read\_Master\_Log\_Pos: 600     //#同步读取二进制日志的位置，大于等于Exec\_Master\_Log\_Pos

              Relay\_Log\_File: ddte-relay-bin.000003

              Relay\_Log\_Pos: 251

              Relay\_Master\_Log\_File: mysql-bin.000002

             Slave\_IO\_Running:

Yes    //此状态必须YES

              Slave\_SQL\_Running: Yes     //此状态必须YES

                    ......

出现以上信息表示配置成功

1. **Tomcat容器安装**

#### 下载tomcat安装包

从tomcat官方网站[http://tomcat.apache.org](http://tomcat.apache.org/)下载安装包，然后解压到某个目录，比如: ~/opt/apache-tomcat-7.0.63  
官方文档中建议不要使用各个Linux发行版中已经打好的安装包，因为各个发行版中的安装包将tomcat安装到不同的位置，所以建议不要使用源里的tomcat。  
$tar xzvf apache-tomcat-7.0.63.tar.gz -C ~/opt

#### 监听80端口

修改tomcat安装目录下conf目录下的server.xml   
$cd ~/opt/apache-tomcat-7.0.63/conf  
$vim server.xml  
...  
<Connector port="80"...>  
...

#### 增加用户

修改tomcat7安装目录下conf目录下的tomcat-users.xml文件  
$cd ~/opt/apache-tomcat-7.0.63/conf  
$vim tomcat-user.xml  
...  
<role rolename="admin-gui"/>  
<role rolename="manager-gui"/>  
<user username="admin" password="1234" roles="admin-gui"/>  
<user username="manager" password="1234" roles="manager-gui"/>  
...

#### 设置工作目录

Tomcat下的web程序默认目录为$TOMCAT\_INSTALL\_DIR/webapps，只要将自己的web程序目录放置到该目录下就可，但用户通常需要设置自己的工作目录。两种方法：

1）通过Context的docBase变量设置  
$vim $TOMCAT\_INSTALL\_DIR/conf/Catalina/localhost/mywebapp.xml  
<Context path="/mywebapp" docBase="~/working/project/program/webapps/mywebapp" reloadable="true"/>  
注意：tomcat 5.5以后的版本，path变量可以不用设置，tomcat根据文件名来确定路径，例如：文件名为mywebapp.xml, 则路径为/mywebapp

2）软链接  
$cd $TOMCAT\_INSTALL\_DIR/webapps  
$ln -s ~/working/project/program/webapps/mywebapp mywebapp

#### 安装Tomcat为随系统启动的服务

1) 在$TOMCAT\_INSTALL\_DIR/bin/setenv.sh脚本中设置JAVA\_HOME环境变量  
$vim $TOMCAT\_INSTALL\_DIR/bin/setenv.sh

#!/bin/sh

JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-i386

2）拷贝$TOMCAT\_INSTALL\_DIR/bin/catalina.sh到/etc/init.d目录下，并在脚本开头加入LSB的设置信息、tomcat安装位置环境变量设置：  
$sudo cp catalina.sh /etc/init.d/tomcat  
$sudo vim /etc/init.d/tomcat

### BEGIN INIT INFO  
# Provides: apache-tomcat  
# Required-Start: $all  
# Required-Stop:  
# Default-Start: 2 3 4 5  
# Default-Stop: 0 1 6   
# Short-Description: Run /etc/rc.local if it exist  
### END INIT INFO

CATALINA\_HOME="/home/cb/opt/apache-tomcat-7.0.63"  
cd $CATALINA\_HOME/bin

3）安装服务  
$sudo update-rc.d -f tomcat defaults

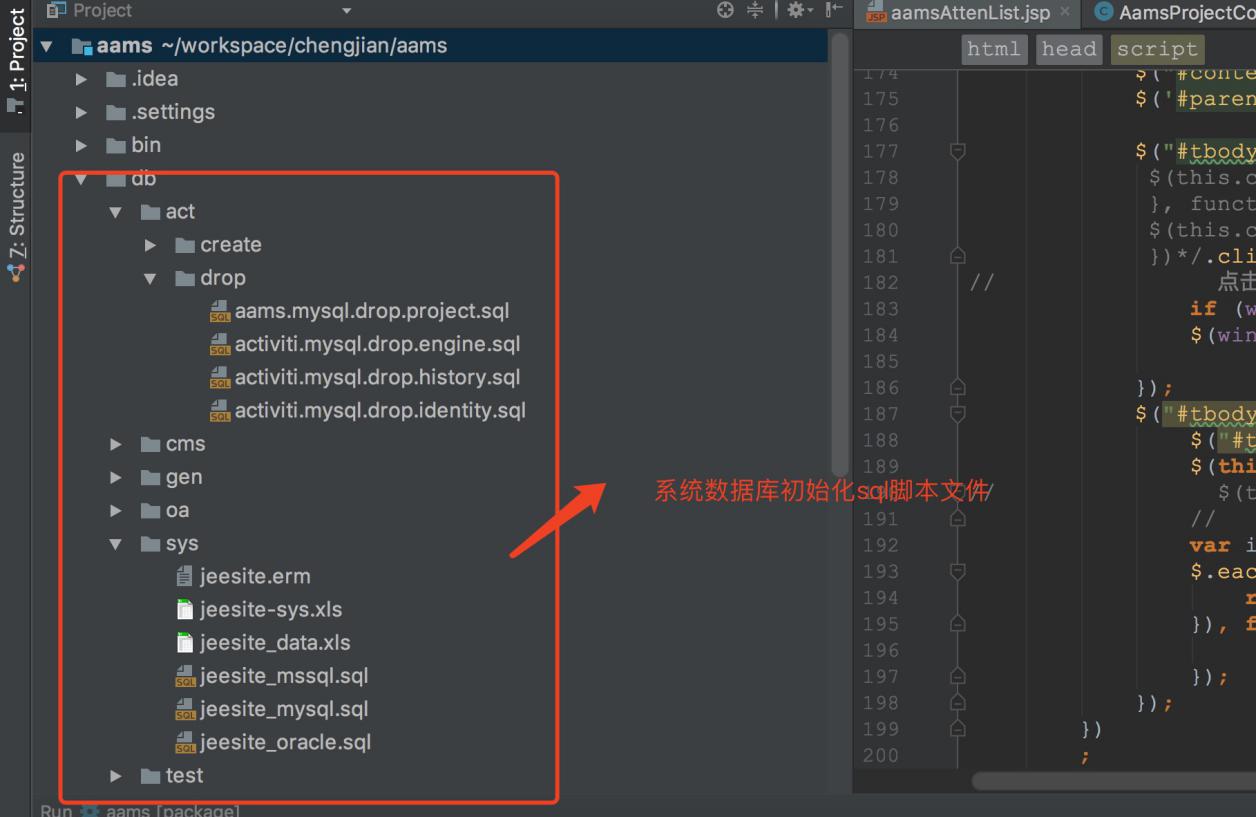
4）启动/停止服务  
$sudo service tomcat start  
$sudo service tomcat stop

#### 测试

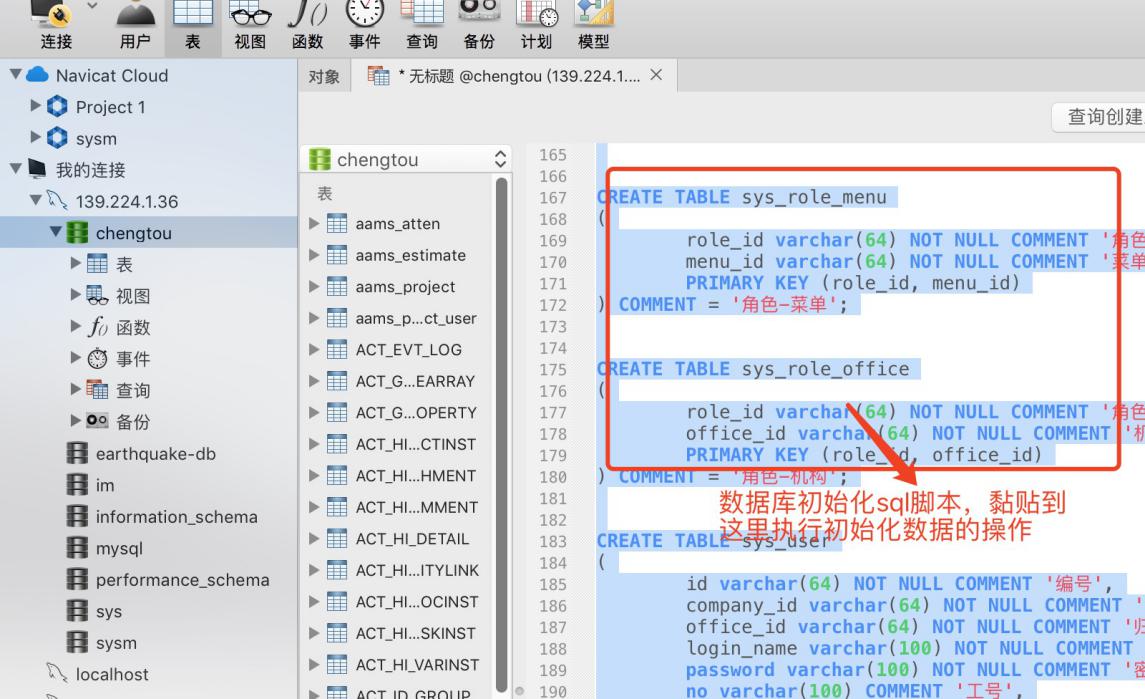
$sudo service tomcat start

1. **数据初始化**
2. 管理员分配部署人员数据库操作权限

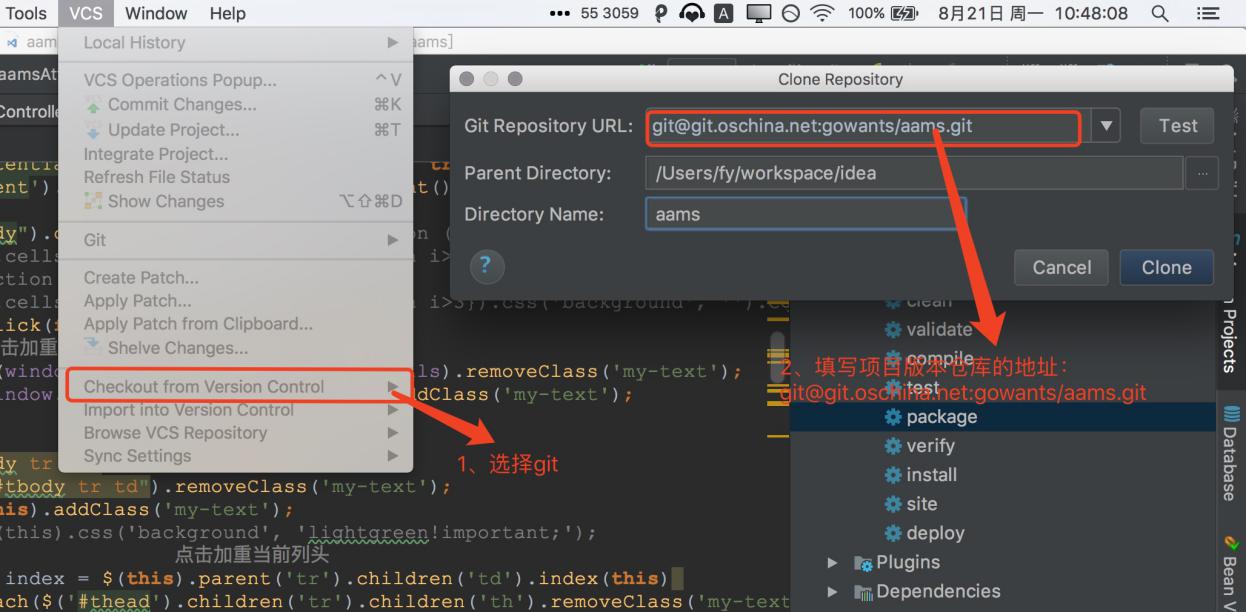
1. 管理员分配部署人员版本控制服务器操作权限
2. 部署人员从版本控制服务器下载源码中的db目录



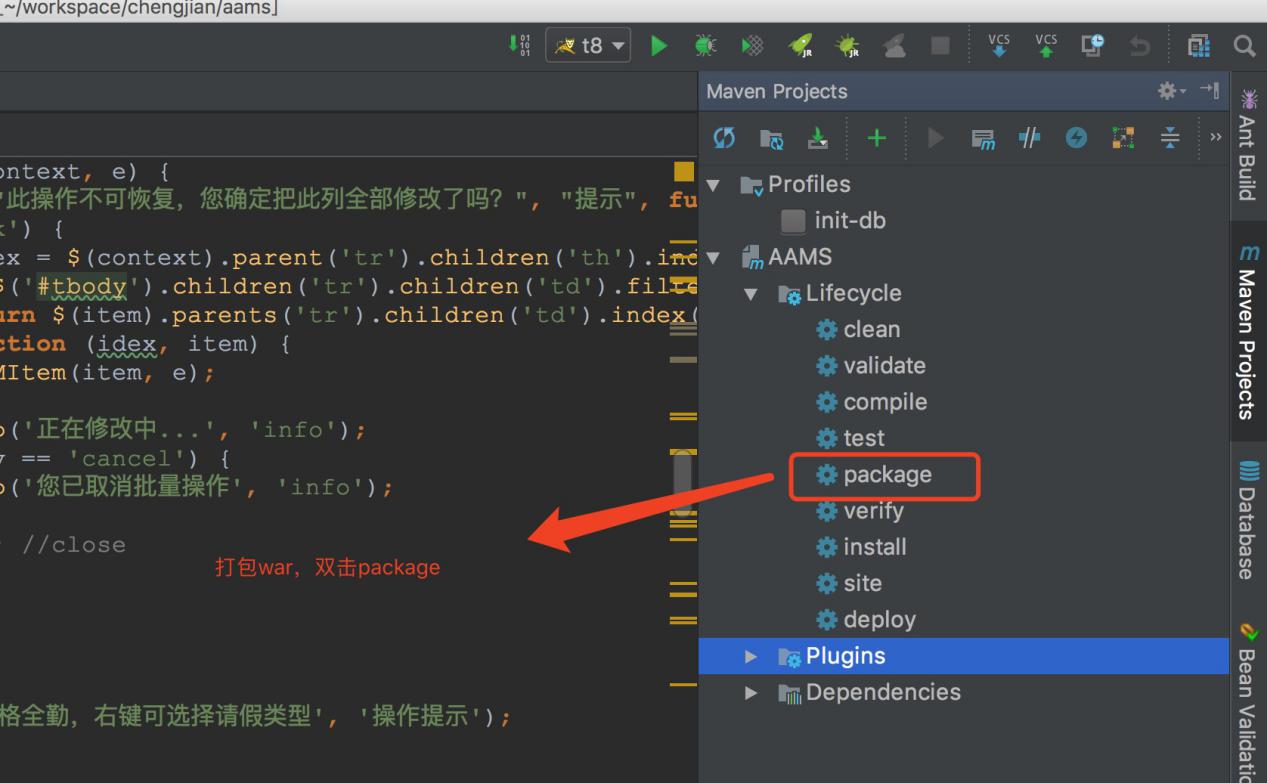
1. 部署人员用Navcat配置数据库账号，登陆数据库
2. 在navcat中执行db目录中的所有sql脚本初始化好数据库

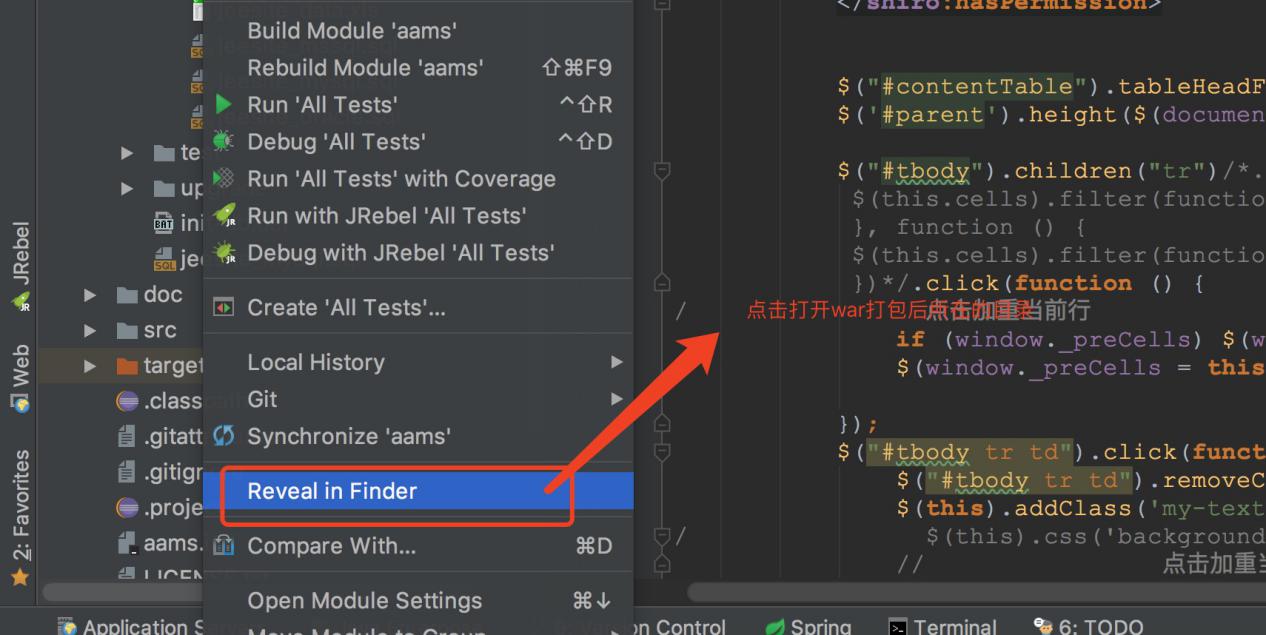


1. **服务器部署**
2. 公司管理员分配部署人员代码仓库权限,仓库地址：git@git.oschina.net:gowants/aams.git

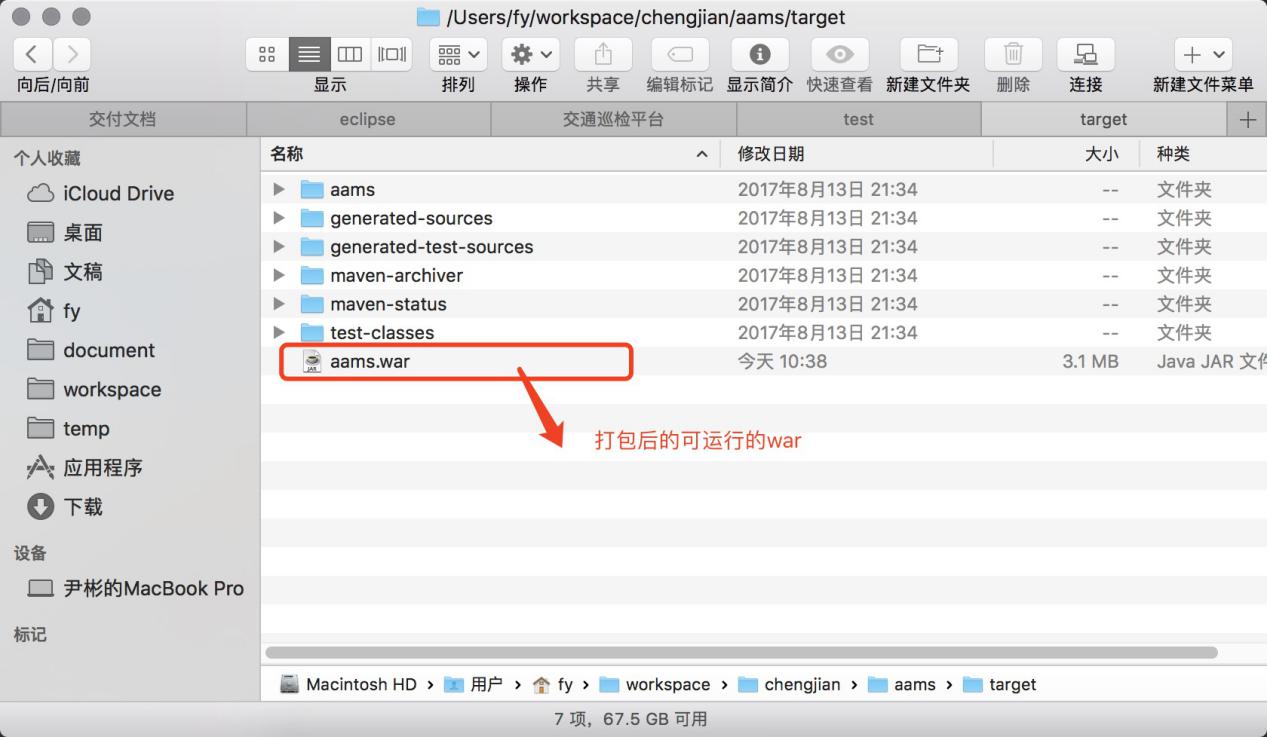


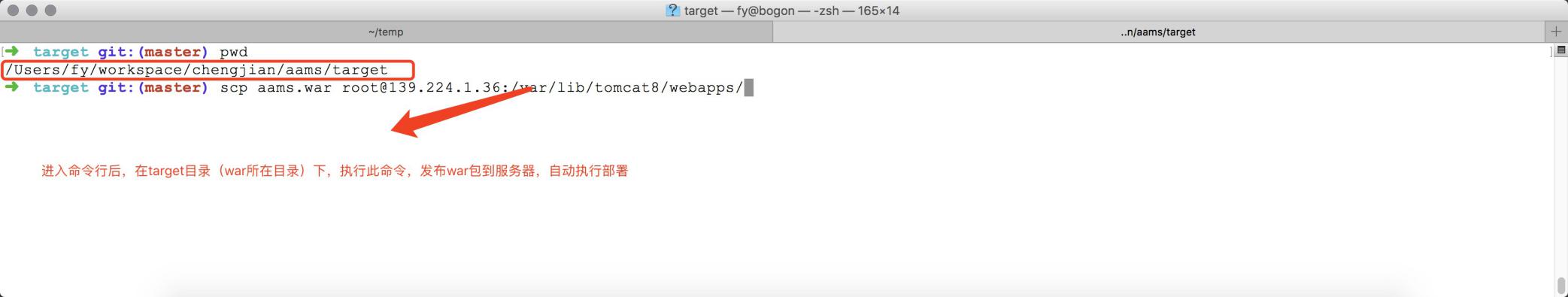
2) 打包成可执行程序





1. 与客户协商暂停服务器运行的时间
2. 将文件为aams.war的包上传到tomcat部署目录：var/lib/tomcat8/webapps/目录下，然后重启tomcat，如下命令：./catalina.sh run





1. 在客户端输入http://ip/aams，显示登录页面表明项目部署成功。

