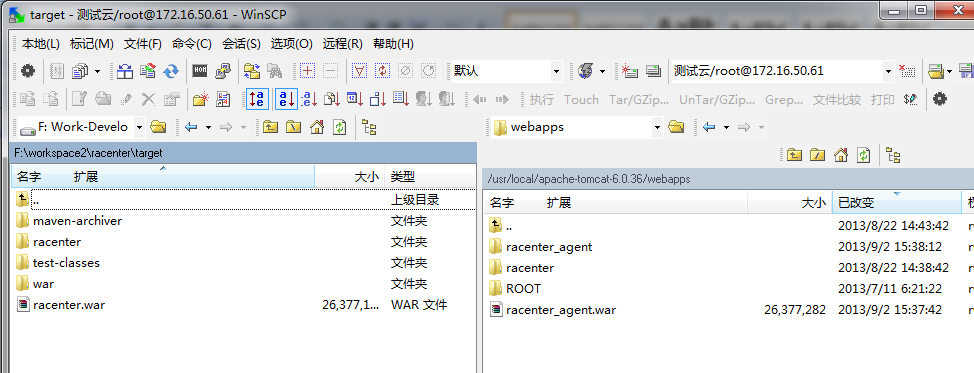
**云平台自服务系统部署流程**

(注：以测试云的服务器为例。)

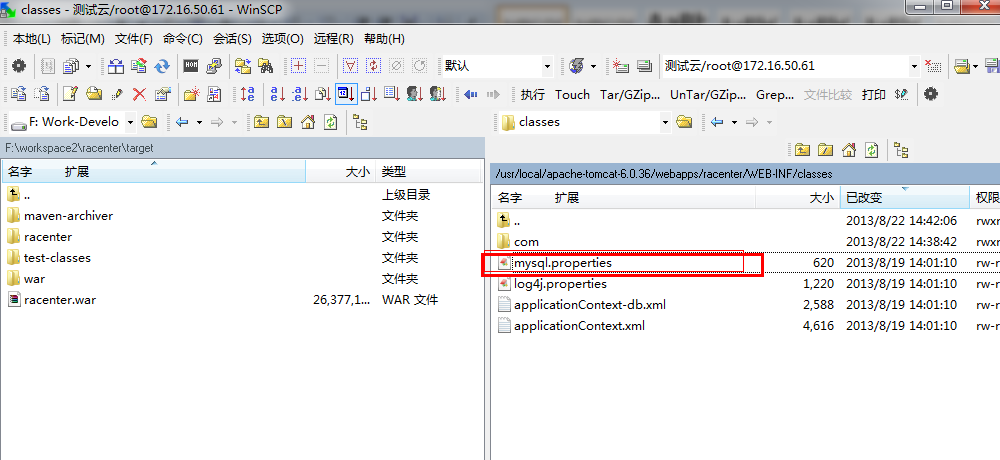
1. 172.16.50.61：Racenter源文件：racenter.tar.gz或者是racenter.war

把源文件拷贝到61机器上,然后解压拷贝到（如果是racenter.war就直接拷贝到/usr/local/apache-tomcat-6.0.36/webapps/目录下就可以了）

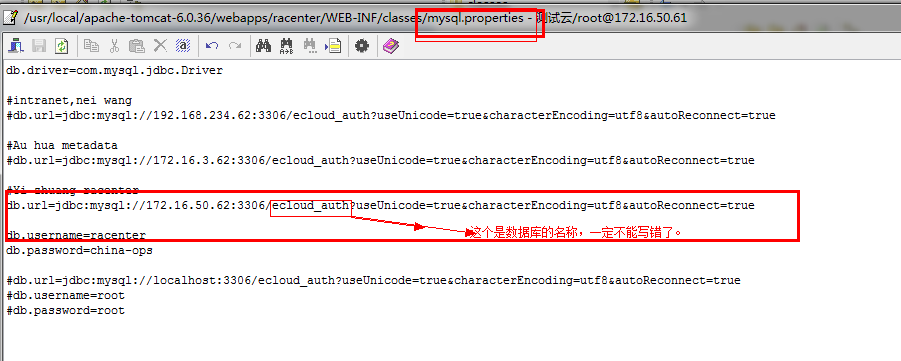
/usr/local/apache-tomcat-6.0.36/webapps/目录下：如图：



然后编辑此文件：（mysql.properties）



然后修改这个文件里面的配置：



这个文件是连接mysql数据库的。需要修改db.url 后面的那个ip地址，ip地址就是数据库的那台机器的地址，后面是数据库的名称。

最后启动tomcat：

/usr/local/apache-tomcat-6.0.36/bin/startup.sh

这台机器就完事了。

**2**．172.16.50.62 ：ecloud\_auth是172.16.50.61(racenter)对应的数据服务器

首先需要一份现有的racenter DB 的备份，我们是从这个备份中恢复一个新的数据库出来。

登录到想要做为模板的那台MySQL数据库Server上，执行如下命令：

mysql dump –default -character-set=utf8 -u root -p ecloud\_auth>ecloud\_auth\_xxxxxxx.sql

这样这个MySQL库就备份到文件ecloud\_auth\_xxxxxxx.sql中了。把这个文件上传到需要部署的那台ecloud\_authdb服务器上。

以下以测试云的ecloud\_auth\_db 172.16.50.62为例。

1). 首先要确认已经安装了MySQL-Server包

# yum list installed mysql-server

如果提示Error: No matching Packages to list，则说明mysql-server包没有安装。

2)..安装mysql-server包

# yum install mysql-server

3)..启动mysql

# servicemysqld start

4).修改mysql的root密码。

# /usr/bin/mysqladmin -u root password 'new-password'

5)．让mysql在开机时自动运行。

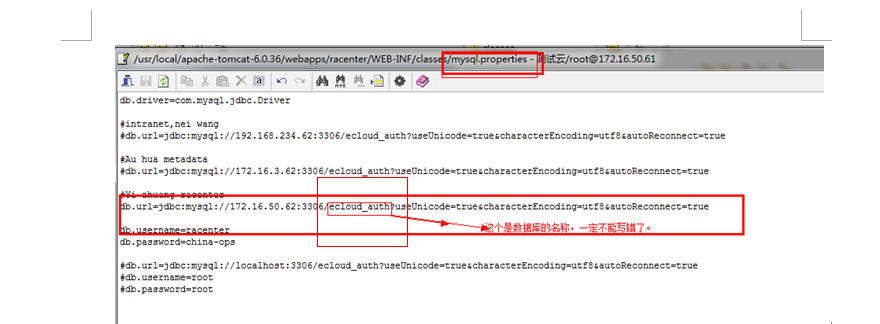
# chkconfigmysqld on

6). 登录mysql

# mysql –uroot –p

7). 恢复racenter Database.

新建一个数据库。

mysql>create database **ecloud\_auth**;(这个数据库名称就是：)

选中此数据库

mysql>use ecloud\_auth;

把上传的那个备份文件恢复到本库中。

mysql>source/你存放的目录/ecloud\_auth\_xxxxxx.sql

为这个库授权，以便我们的应用能访问它

mysql>grant all privileges on ecloud\_auth.\* to **racenter**@'%' identified by '**china-ops**' with grant option;

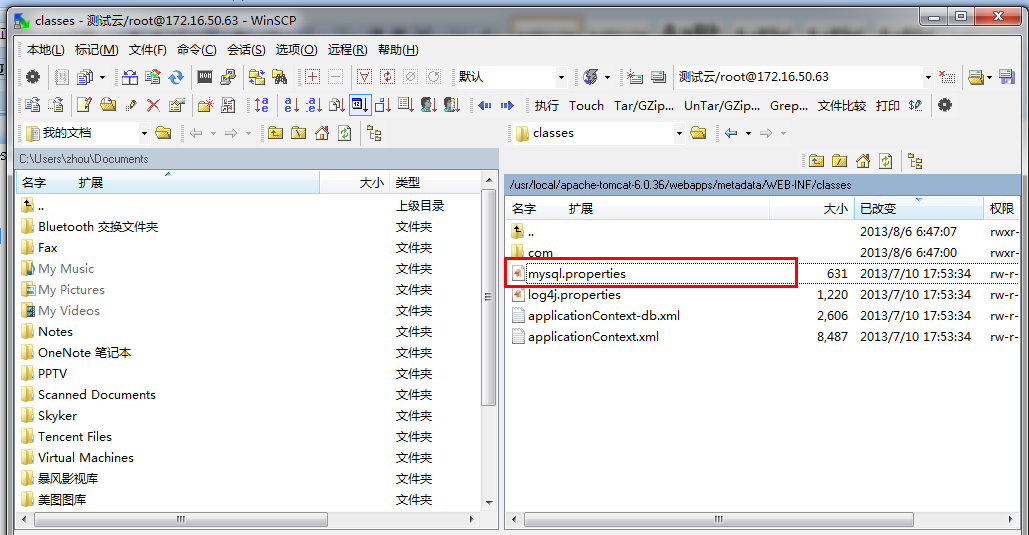
mysql>flush privileges;

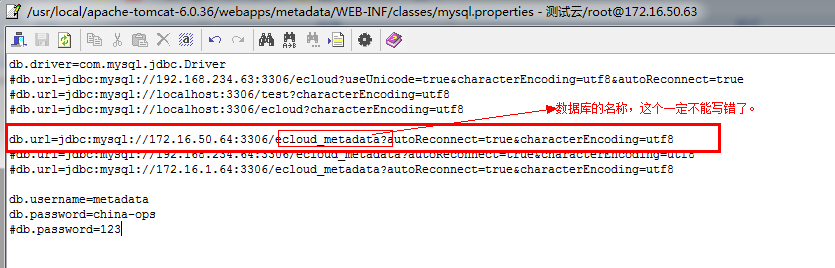
至此racenter数据库服务器就部署好了。172.16.50.61上的程序就可以访问这个数据库了。

**3**．172.16.50.63 ：medatada源文件：metadata.tar.gz或者是metadata.war

这台机器的部署步骤和172.16.50.61机器是一模一样的：

需要注意的是也是要修改mysql.properties文件：如下：





**4**．172.16.50.64 ：medatada是172.16.50.63(medatada)对应的数据服务器

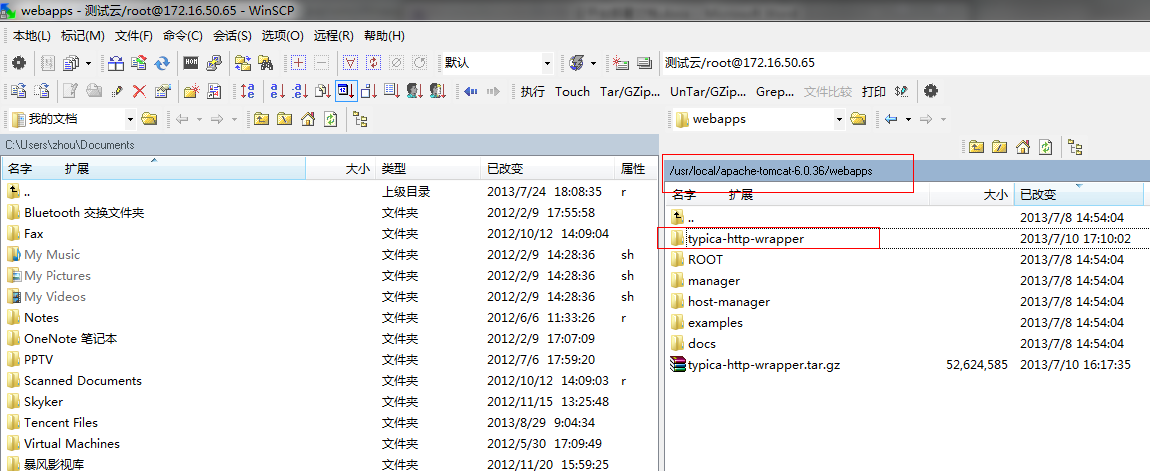
这台机器的部署步骤和172.16.50.62是一样的。

也是由一个备份的数据库源文件恢复一个新的数据库。

**（注意数据库的名称和赋权的时候的用户名，密码）**

**5**．172.16.50.65 ：是typica 源文件：typica-http-wrapper.tgz

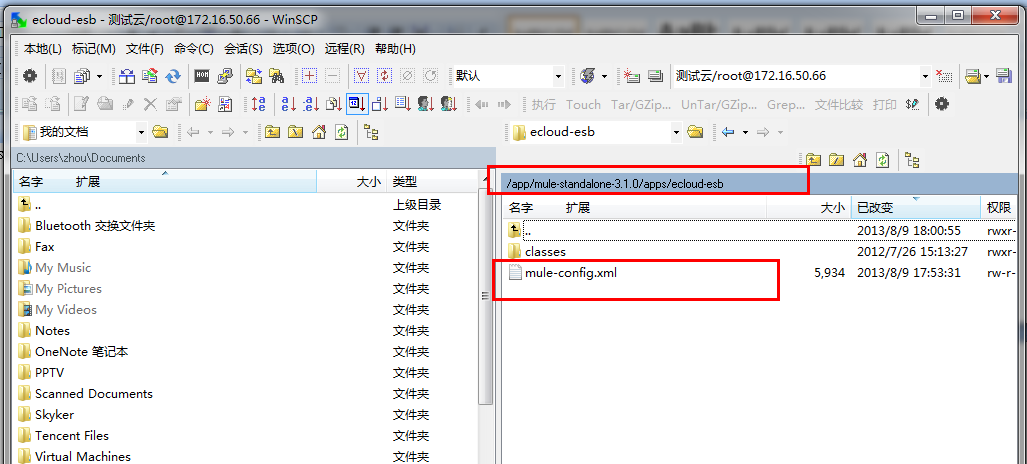
这台机器和172.16.50.61一样，把文件拷贝到：



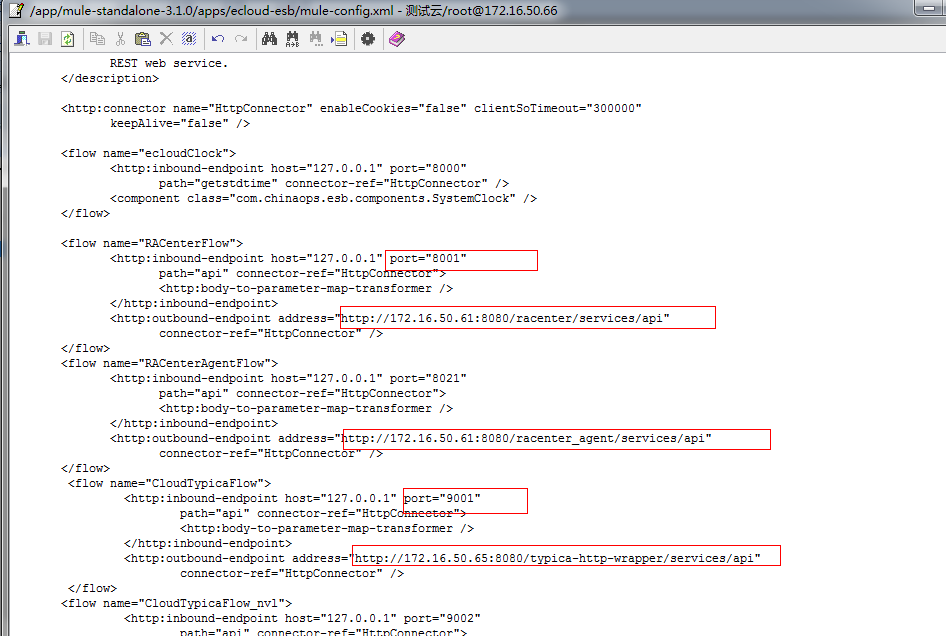
不用修改配置文件，直接启动tomcat就可以了。

**6**．172.16.50.66 ：是mule\_server 源文件：mule-standalone-3.1.0.tgz

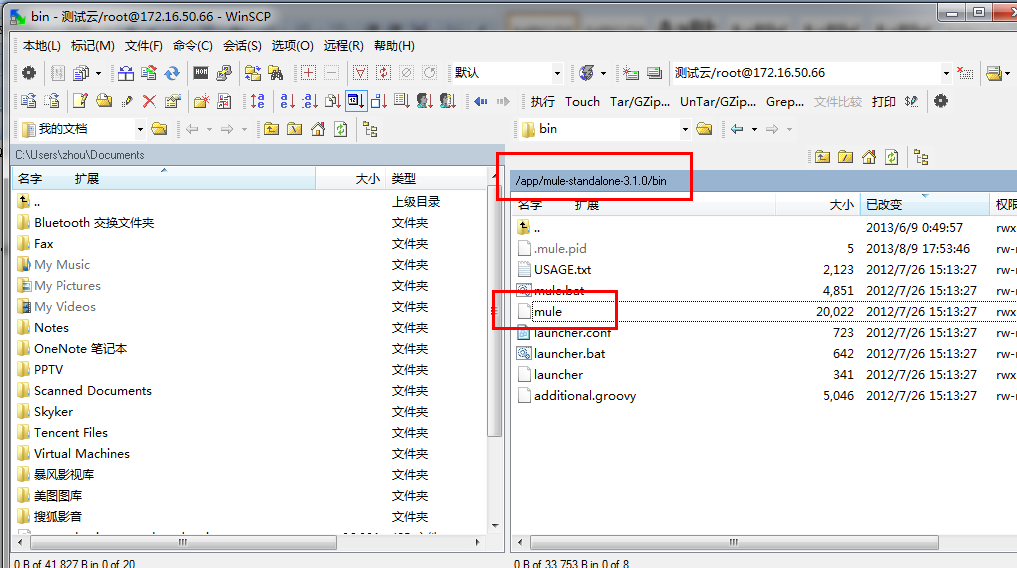
把源文件拷贝到机器上，解压后：开始修改配置文件（**mule-config.xml**）：



这个文件里面配置的是供前台访问的程序都是通过端口来映射的：



配置文件改好之后启动mule:



**./mule start**

**7**．172.16.50.67：ecloud\_console: 包含运营管理员的前台界面和公司管理员

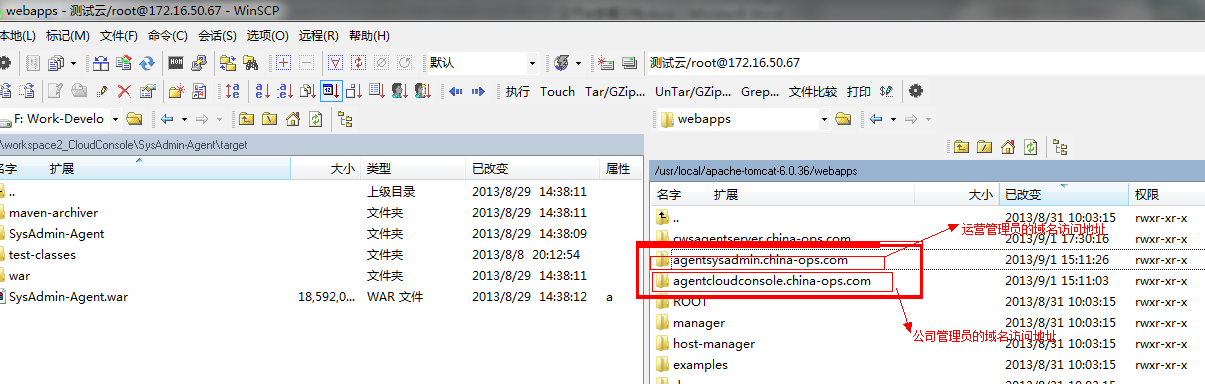
的前台界面。源文件：包名都是ROOT.war。例如：admin/ROOT.war(这个就是公

司管理员的前台界面)。Sysadmin/ROOT.war(这个就是运营管理员的前台界面)

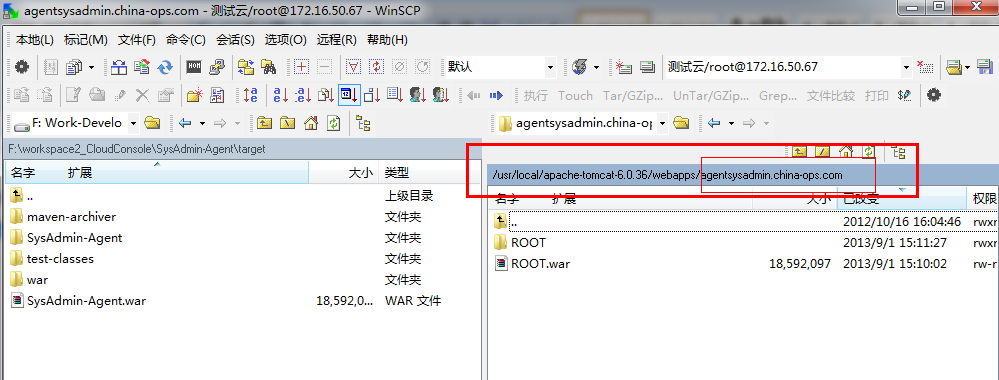
(因为都是用域名访问的，当2个平台的不同war包文件上传到服务器之后，在

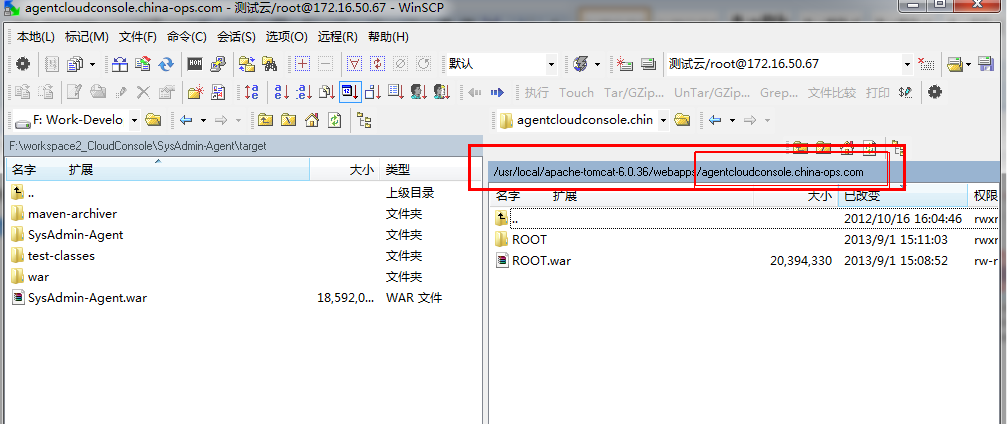
部署到tomcat下面的时候都重命名为ROOT.war)

首先要在/usr/local/apache-tomcat-6.0.36/webapps/下新建2个目录，**(这2给目录的名称也将是平台的域名访问地址):**

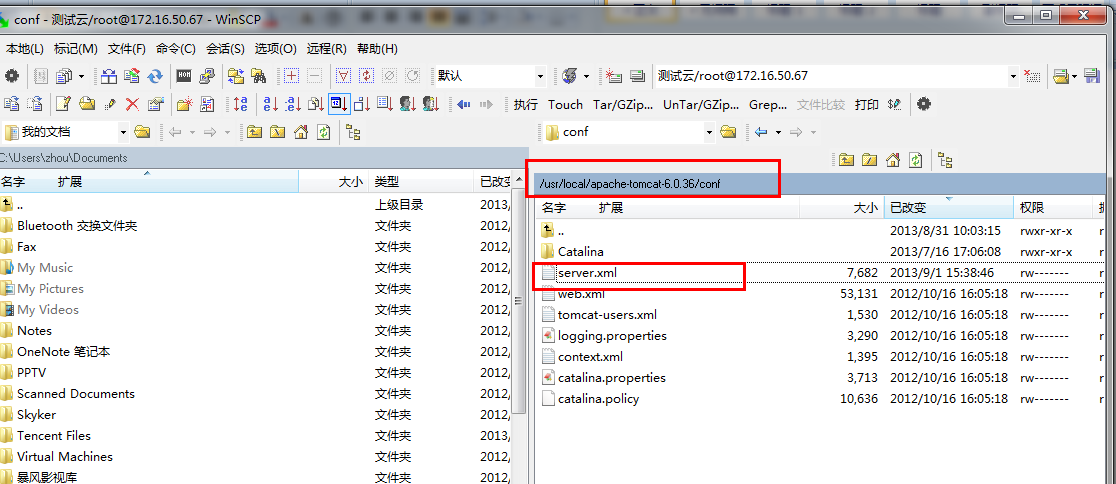
****

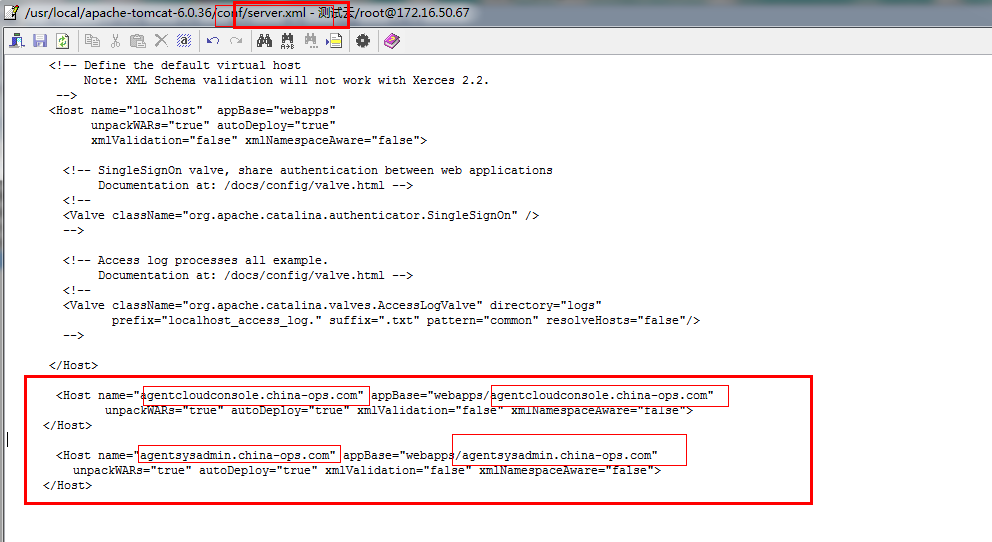
然后把2个不同的ROOT.war包分别拷贝到这2个目录下：(上传源文件到机器上的时候都各自建好相应的文件夹，由于最后改名之后都一样都是ROOT.war，不能把个平台的弄混了。)





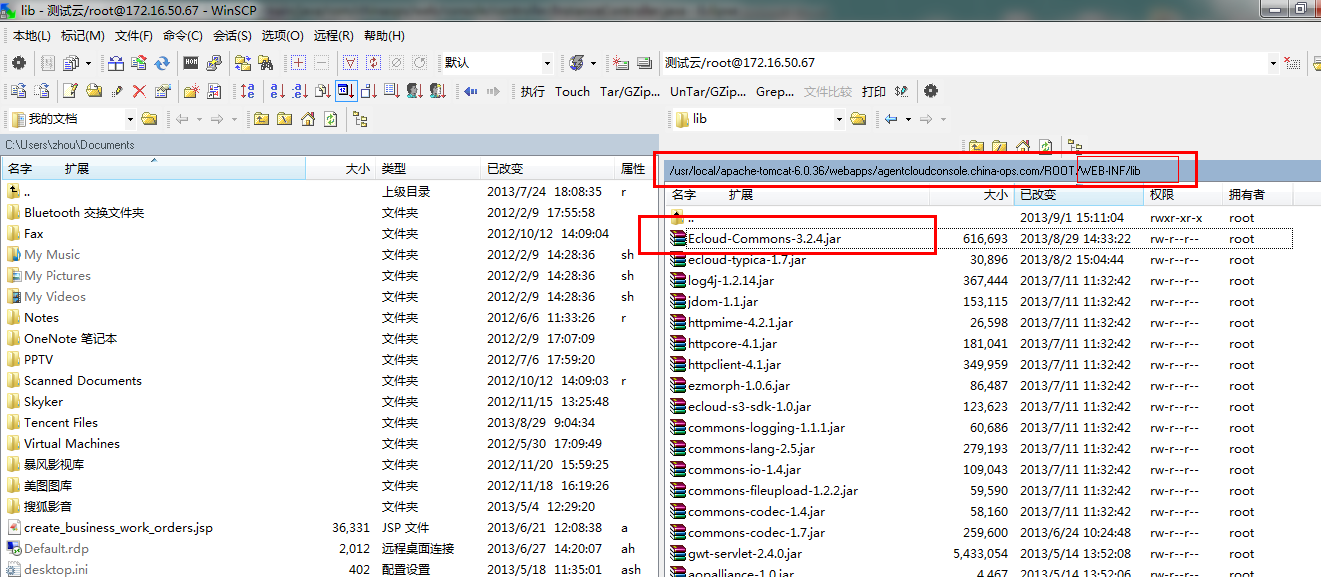
然后需要编辑tomcat的server.xml文件：

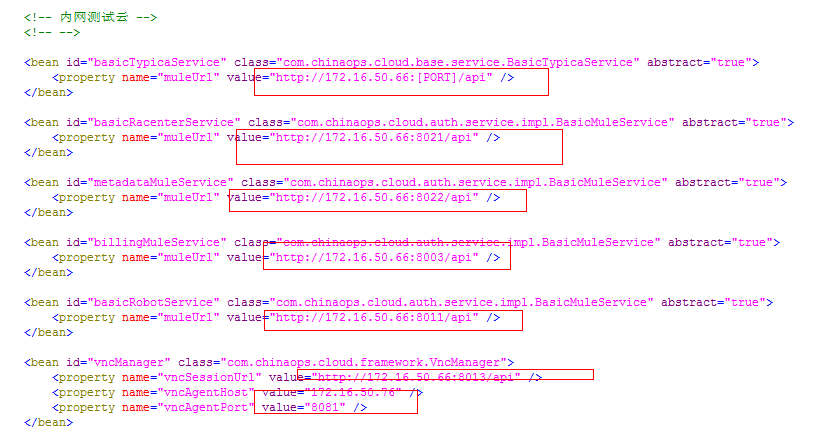
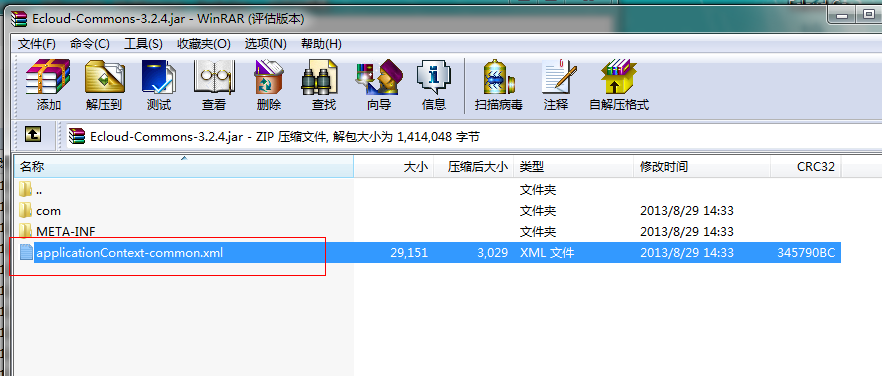


主要是添加那2个自己定义的域名访问：

然后还需要修改下图路径中的：Ecloud-Commons-3.2.4.jar中的

applicationContext-common.xml：





Vnc配置中的IP为50段网关的外网IP，端口为76下startProxy.sh中配置的端口：8081端口。

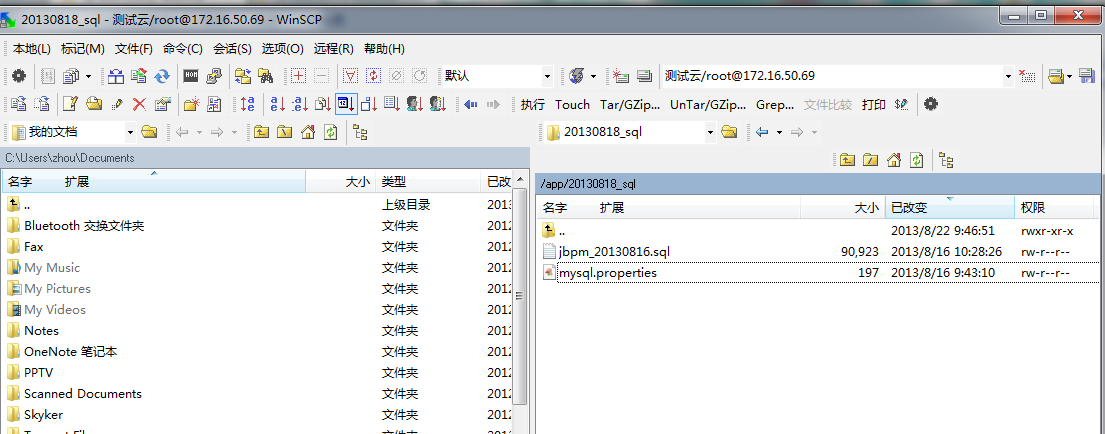
最后启动tomcat就可以了。

**8**．172.16.50.68 ：func\_cloud

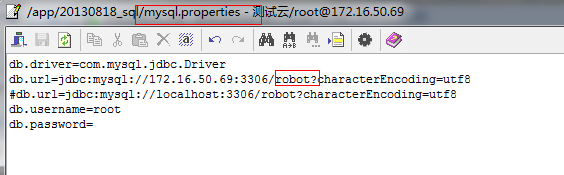
**9．**172.16.50.69：jbpm\_server：设置HA的数据库和程序

这台机器上既有数据库也有对应的程序：

1).首先需要在这台机器上安全mysql,然后像172.16.50.62/64那样把从另外机器上备份出来的mysql数据库恢复到这台机器上，这样设置HA的数据库就部署好了。



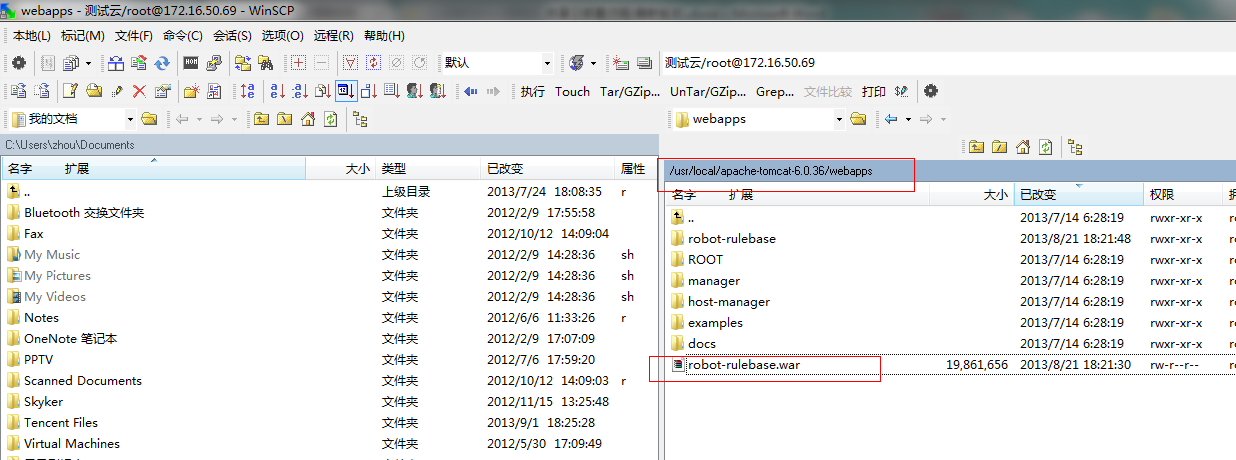
也是需要修改mysql.properties文件的：



2).然后就是程序的部署：源文件：robot-rulebase.war

程序的部署也是和172.16.5062/74一样的。源文件拷贝都这台机器上：

然后放到/usr/local/apache-tomcat-6.0.36/webapps下面，启动tomcat。



**10**．172.16.50.70：robot\_agent

11．172.16.50.71：robot\_controller

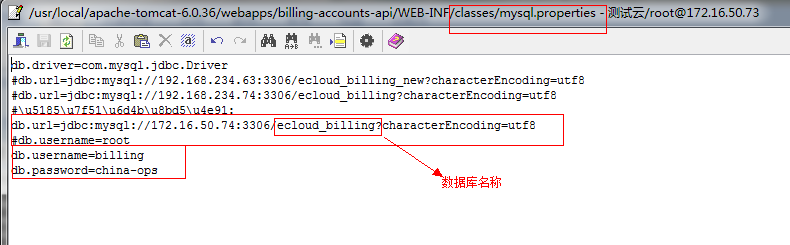
**12．**172.16.50.72：rrdgraph

**13．**172.16.50.73：billing\_api

（计费的机器的程序:源文件：billing-accounts-api.war）

这台机器的部署和172.16.50.61/63是一样的。

需要注意的是一些需要修改mysql.poperties文件：



**14．**172.16.50.74：ecloud\_billing\_db (这个是billing\_api对应的数据库。)

这台机器的部署步骤和172.16.50.62/64是一样的。

也是由一个备份的数据库源文件恢复一个新的数据库。

**（注意数据库的名称和赋权的时候的用户名，密码）**

**15.**172.16.50.75：

这台机器部署的也是数据服务器。部署步骤和172.16.50.62/64是一样的。

也是由一个备份的数据库源文件(xentop\_\*\*\*\*.sql)恢复一个新的数据库。

**（注意数据库的名称和赋权的时候的用户名，密码）**

VNC部署说明：

1.系统架构图

Dispatch server

|

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

| |

Proxy server 1 Proxy server 2

2. Dispatch serve

2.1 Dispatch server. 是一个分发器，修改过的tinyvnc.jar在连接时会先发一个请求过来,dispatch server

读取serverList.txt中的Proxy server的ip和port，随机返回其中的一个proxy server给tinyvnc.jar，tinyvnc.jar便开始连接

返回的这个proxy server。

2.2 启动dispatch server，启动参数是它监听的端口。这个参数需要与ecloud-commons里applicationContext-common.xml中的

vncManager的vncAgentPort的值一致。

下面是启动脚本：

count2=`ps -ef | grep "java" | grep "vncDispatcher" | grep -v "grep"`

if [ "$?" != "0" ]; then

cd /app;

nohup java -jar vncDispatcher.jar 8081 >> dispatcher.log 2>&1 &

fi

其中的端口8081要与云管理平台项目里的ecloud-commons中applicationContext-common.xml的vncManager的vncAgentPort的值一致。

2.3 serverList.txt格式。

serverList.txt内容如下（示范）：

123.126.91.132:80

123.126.91.133:80

123.126.91.132:8056

serverList.txt中保存着所有的VNC Proxy server的列表。这些proxy server必须是外网能访问的，如果使用squid或nginx做了转发，

比如上面的serverList.txt中的第三条，123.126.91.132:8056映射到内网的172.16.50.77:8080，那么在serverList.txt中要保存123.126.91.132:8056,而不是172.16.50.77:8080

3. Proxy server

3.1 Proxy server 根据tinyvnc.jar发过来的请求，在mule上查询相关的物理机IP及端口，连接上去，并把结果返回tinyvnc.jar。通常我们会要求Proxy server有外IP， 能便tinyvnc.jar能访问它。在实际的部署中， proxy server可能只有内网IP，需要用squid或nginx做反向代理，tinyvnc.jar才能访问。我们要把这个proxy server保存在serverList.txt供VNC Dispatcher读取。

3.2 启动 Proxy server，它的启动参数有两个：1. 监听的port，2. Mule server IP。

启动脚本：

count=`ps -ef | grep "java" | grep "vncProxy" | grep -v "grep"`

if [ "$?" != "0" ]; then

cd /app; #这是vnc proxy所在的目录

nohup java -jar vncProxy.jar 8083 172.16.50.66 >> vnc.log 2>&1 &

fi

8083是这个proxy server监听的端口，172.16.50.66（这是个例子）就是mule server的IP，你需要把这个IP修改为部署mule server的那台服务器的IP。

proxy根据tinyvnc.jar里发过来的session id，调用mule里的服务（VNC Detector）来决定要连接后台物理机的哪个端口。这个服务就部署在77上，默认绑定在mule的8013端口上，

proxy启动不能指定VNC Detector在mule上的端口，这个端口是在程序里写死的，就是8013的。所以部署mule时要注意。

3.3 部署proxy server的主机必须从tinyvnc.jar能访问，也就是前台页面能访问到，通常需配置外网IP。

**16**.172.16.50.76：vnc1

**VNC Dispatcher和Proxy部署在76上，解压安装包，copy为/app,修改里面的start.sh.**

**count=`ps -ef | grep "java" | grep "vncProxy" | grep -v "grep"`**

**if [ "$?" != "0" ]; then**

**cd /app;**

**#把下面的端口改为要监听的端口，第二个参数是mule server的IP。**

**nohup java -jar vncProxy.jar 8083 172.16.50.66 >> vnc.log 2>&1 &**

**fi**

**count2=`ps -ef | grep "java" | grep "vncDispatcher" | grep -v "grep"`**

**if [ "$?" != "0" ]; then**

**cd /app;**

**# 端口（这里是8081），需要与ecloud-commons里applicationContext-common.xml中的vncManager的vncAgentPort的值一致。**

**nohup java -jar vncDispatcher.jar 8081 >> dispatcher.log 2>&1 &**

**fi**

**17.**172.16.50.77：vnc2

NC Detector通常部署在77上，这是个web程序，需要安装java和tomcat。下面我们用$TOMCAT\_HOME来表示tomcat的安装目录。

把vnc-detector-api.war复制到$TOMCAT\_HOME/webapps/下

启动tomcat,tomcat会解压这个war，在 $TOMCAT\_HOME/webapps 目录下生成一个 vnc-detector-api，

编辑vnc-detector-api/WEB-INF/classes/mysql.properties,把里面的参数确保用能够连接到xentop数据库。

db.driver=com.mysql.jdbc.Driver

db.url=jdbc:mysql://172.16.50.75:3306/xentop?characterEncoding=utf8

db.username=xentop

db.password=china-ops

保存后重启tomcat。