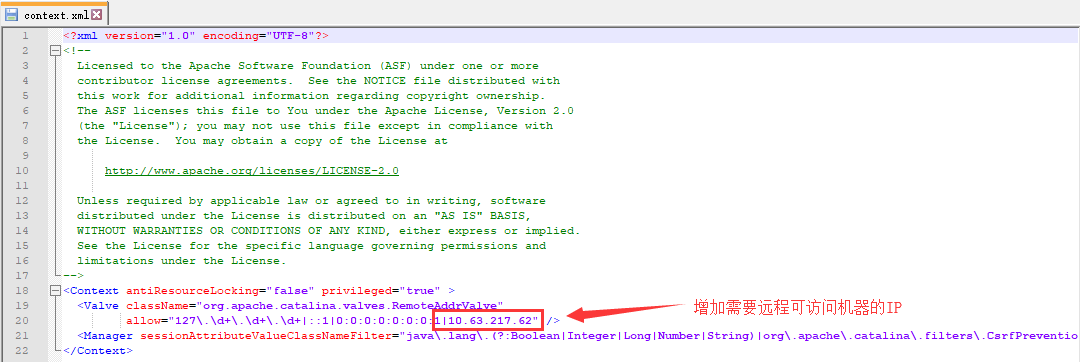
**自动部署系统部署操作文档**

# 使用前提条件

1. 必须是tomcat项目
2. 部署的war包所有节点必须相同，不能对节点做特殊配置
3. Tomcat必须启用远程控制能，且webapps下必须要有manager项目，tomcat配置如下：
   1. 项目中的$TOMCAT\_HOME/webapps/manager/META-INF/context.xml中需要修改：

在Value节点中增加需要远程访问的服务器IP或者将Value节点注释掉，如下图：



* 1. 在项目中的$TOMCAT\_HOME/ tomcat-users.xml配置tomcat账户密码，添加如下代码：

<!-- 注意 这里Jenkins自动部署的角色配置如下 -->

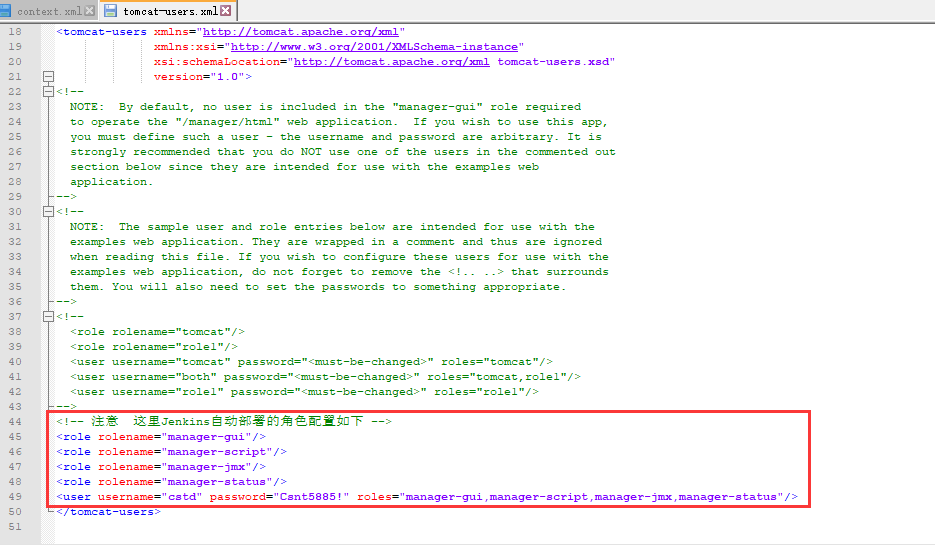
<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

<user username="cstd" password="Csnt5885!" roles="manager-gui,manager-script,manager-jmx,manager-status"/>



# Jenkins部署

## 下载Jenkins

直接在Jenkins的 war包版本即可，也可以直接使用提供的jenkins.war

## 安装jdk

安装jdk1.8接口，这里忽略

## 安装配置Jenkins

### 配置Jenkins的home目录

例如在linux上的配置如下：

打开/etc/profile文件，在最后增加如下代码：

JENKINS\_HOME=/cstc/application/jenkins/program

export JENKINS\_HOME

### 启动Jenkins

在linux和window上都通过命令行执行启动，如下：

java -jar $JENKINS\_ROOT/jenkins.war --httpPort=8888

### 登录Jenkins

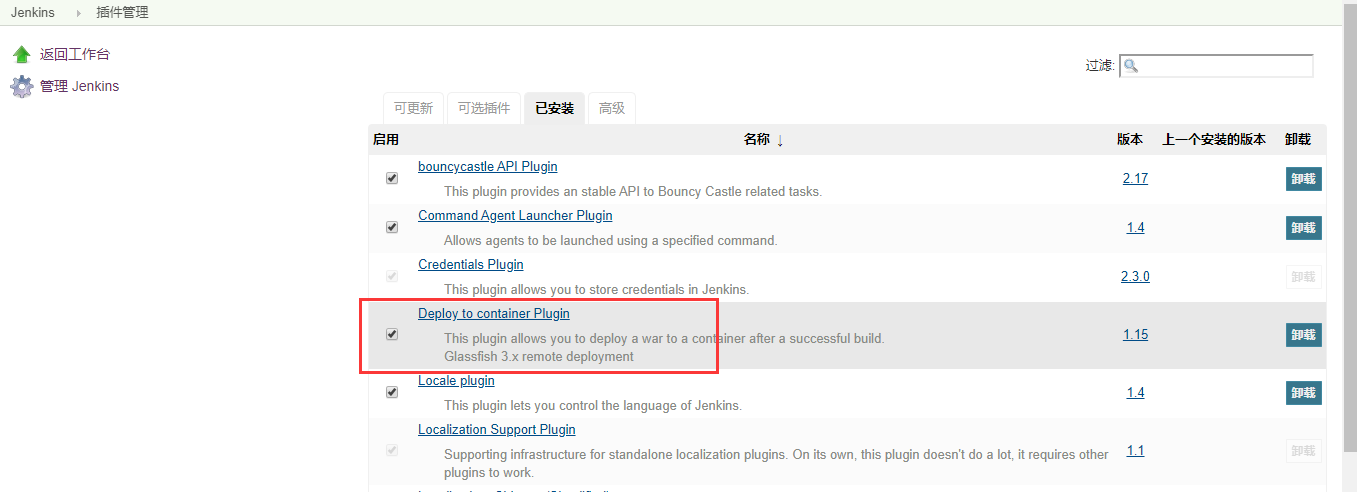
在浏览器中打开<http://ip:8888> ,打开如下页面：



然后再对应提示目录输入密码即可

### 安装Jenkins插件

这里我们本次用到的插件是deploy to container ，如下图：



如果是内网环境，可以直接将我打包的plugins目录替换$ JENKINS\_HOME下的目录，然后重启服务就可能生效了。

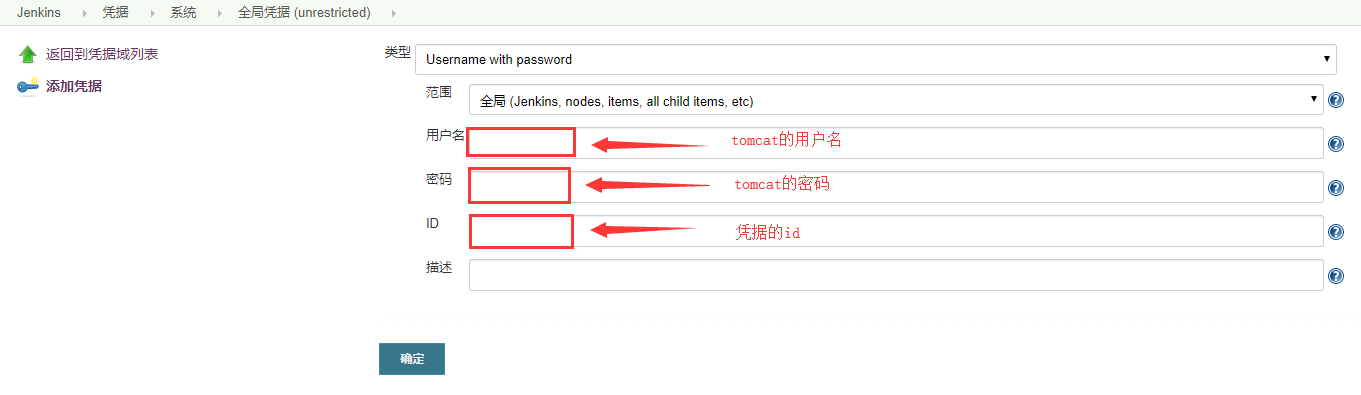
### 创建凭据

凭据是用来登录tomcat的密码文件，如下图创建：

1、登录Jenkins，点击“凭据”-“全局”-“添加凭据”



2、配置凭据，凭据的用户名和密码分别是1.3中配置的，该凭据会在后面创建job时使用，如下图：



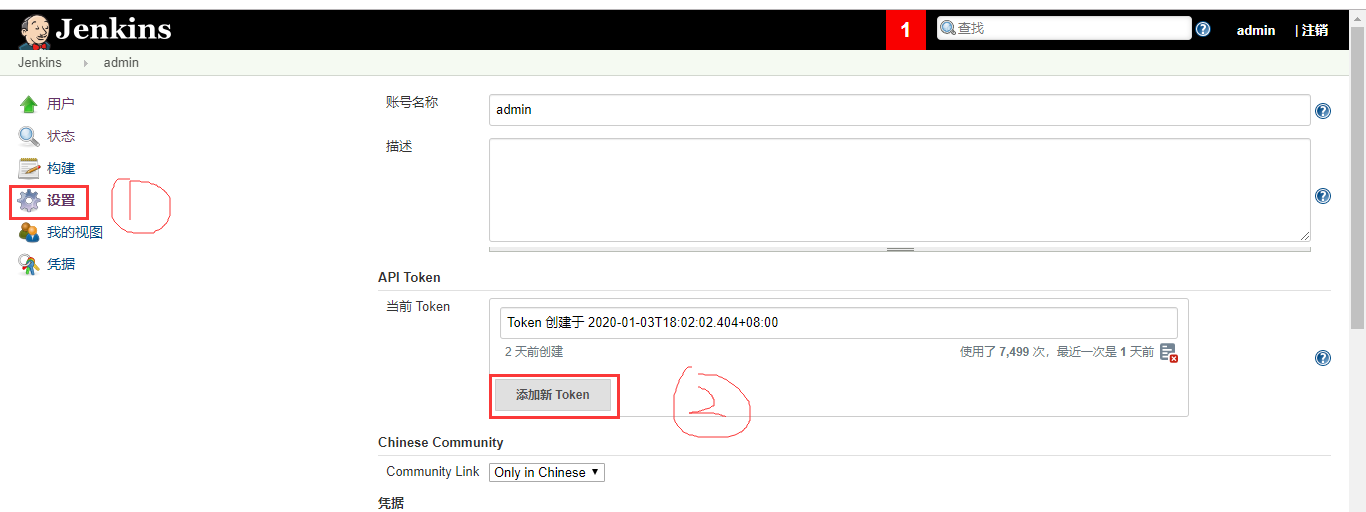
### 配置Jenkins的token

该token主要用于外部系统访问Jenkins系统时使用，生成后需要保留下来。

1. 登录Jenkins，然后点击“用户列表”-“admin”



1. 生成token，点击“设置”-“添加新token”，并复制保存好该token，用于配置在自动部署程序中使用。



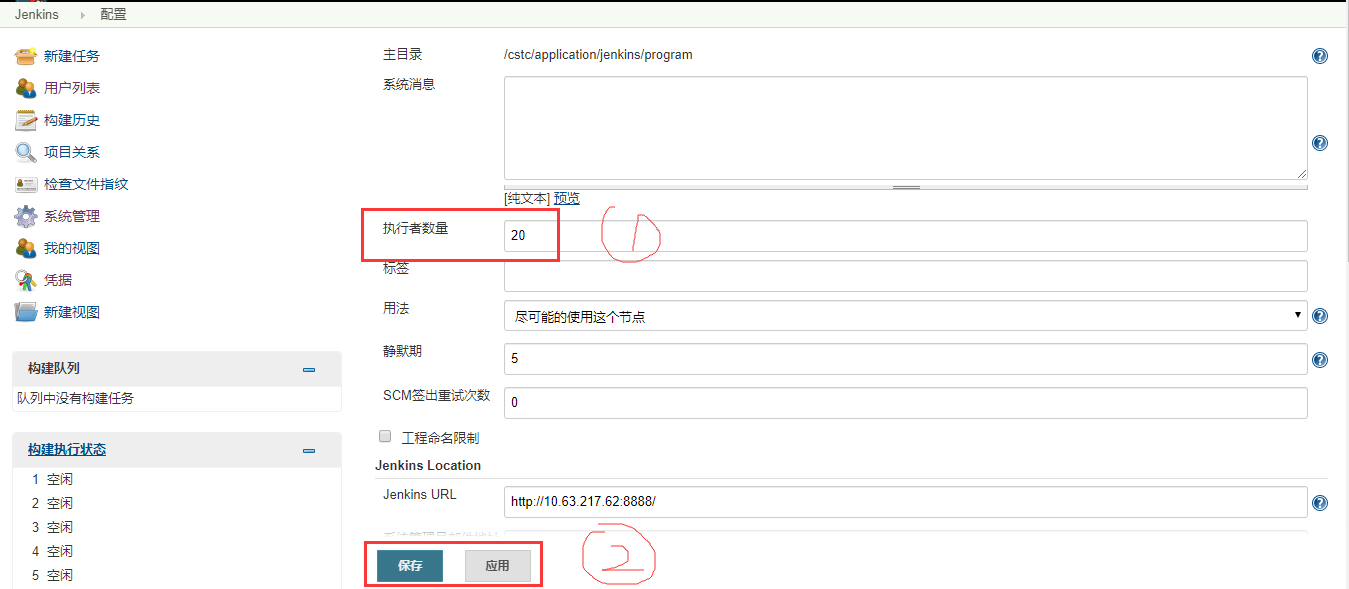
### 设置执行数

配置自动部署同时部署项目的线程数，这里一般是当前服务器的cpu的1.2倍左右，也可以根据实际情况来定。

1. 登录Jenkins，点击“系统管理”-“系统设置”



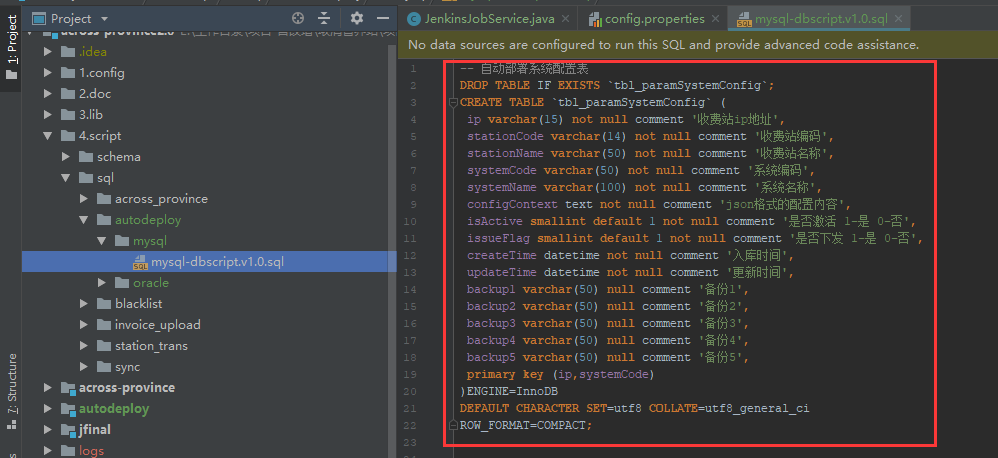
1. 设置执行者数量，最后保存即可。



# 自动部署程序配置

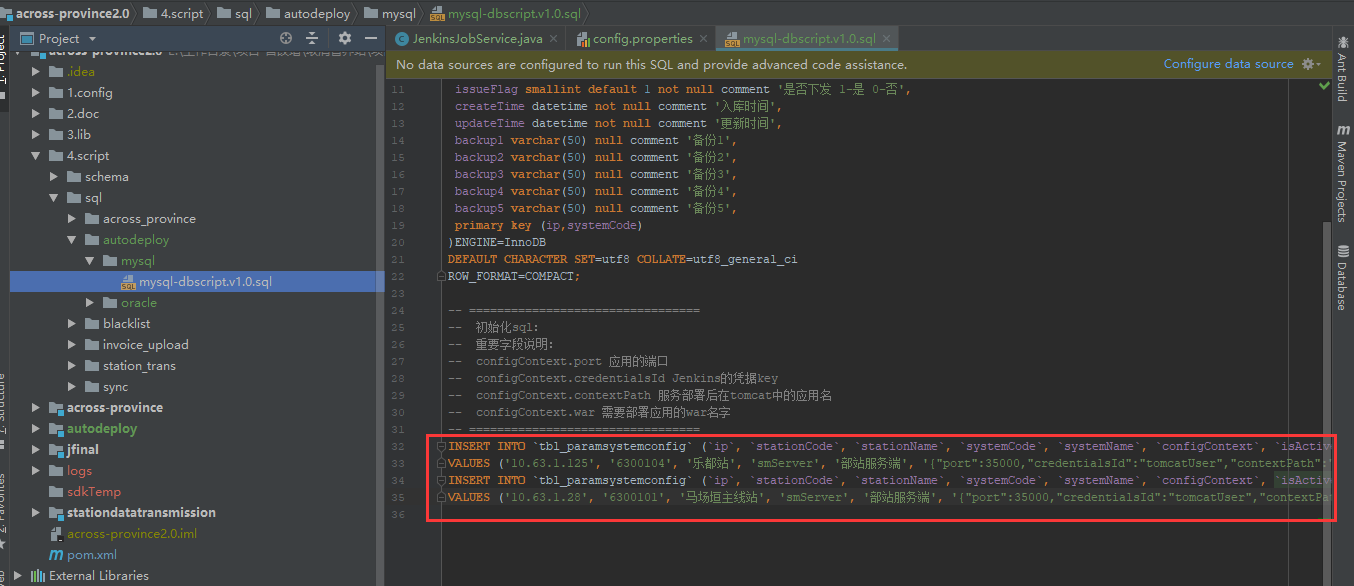
## 创建配置表

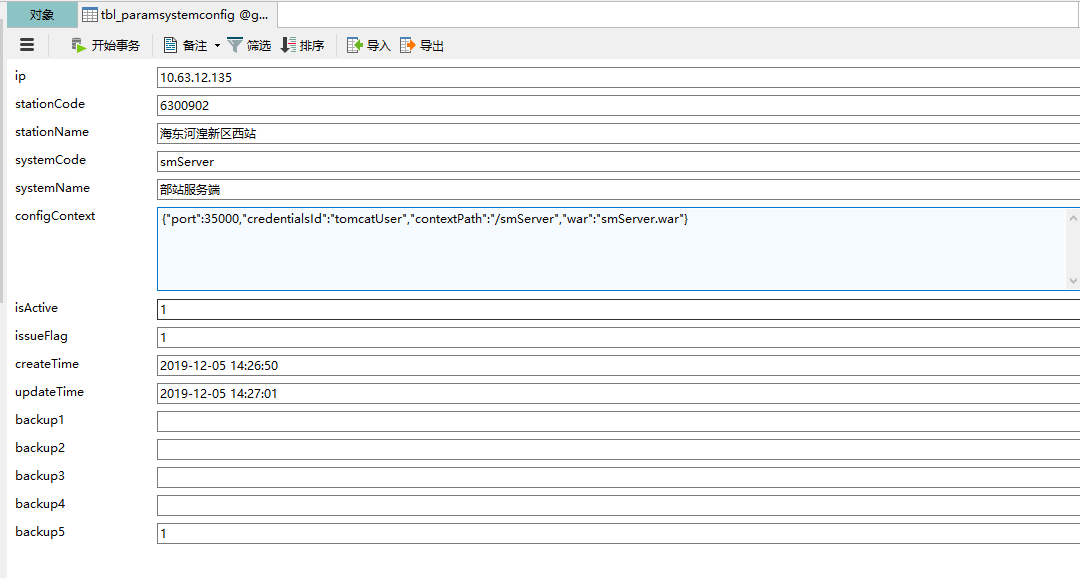
在across\_province2.0项目中的 across-province2.0\4.script\sql\autodeploy目录，执行创建表脚本，如下图：



## 初始化配置数据

需要参考我代码中提供的样例语句进行修改，如下图：





注意：

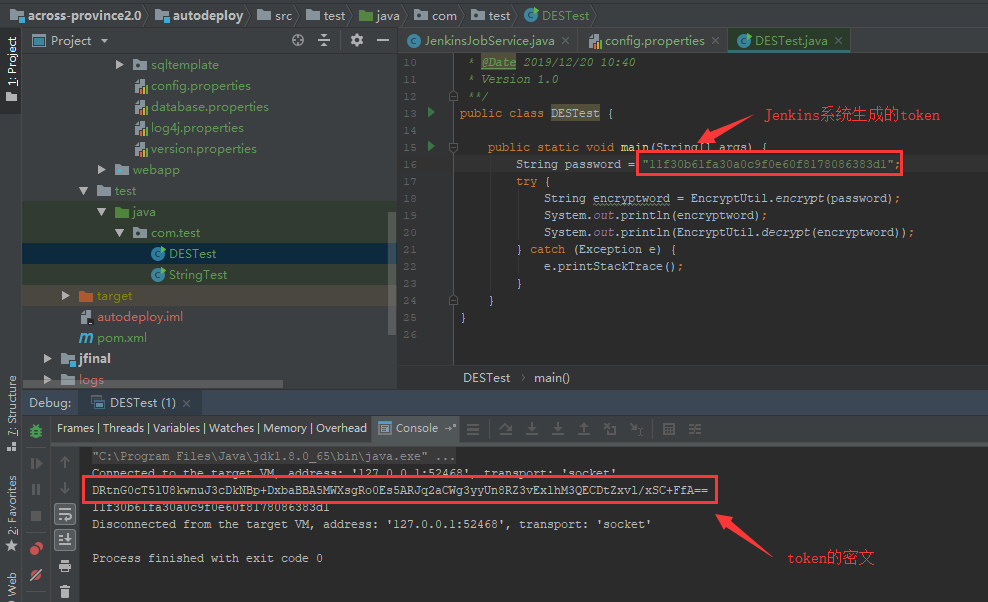
-- configContext.port 应用的端口  
-- configContext.credentialsId Jenkins的凭据key，即2.3.5步创建的  
-- configContext.contextPath 服务部署后在tomcat中的应用名  
-- configContext.war 需要部署应用的war名字

## 修改config.properties配置



## 密文生成方式

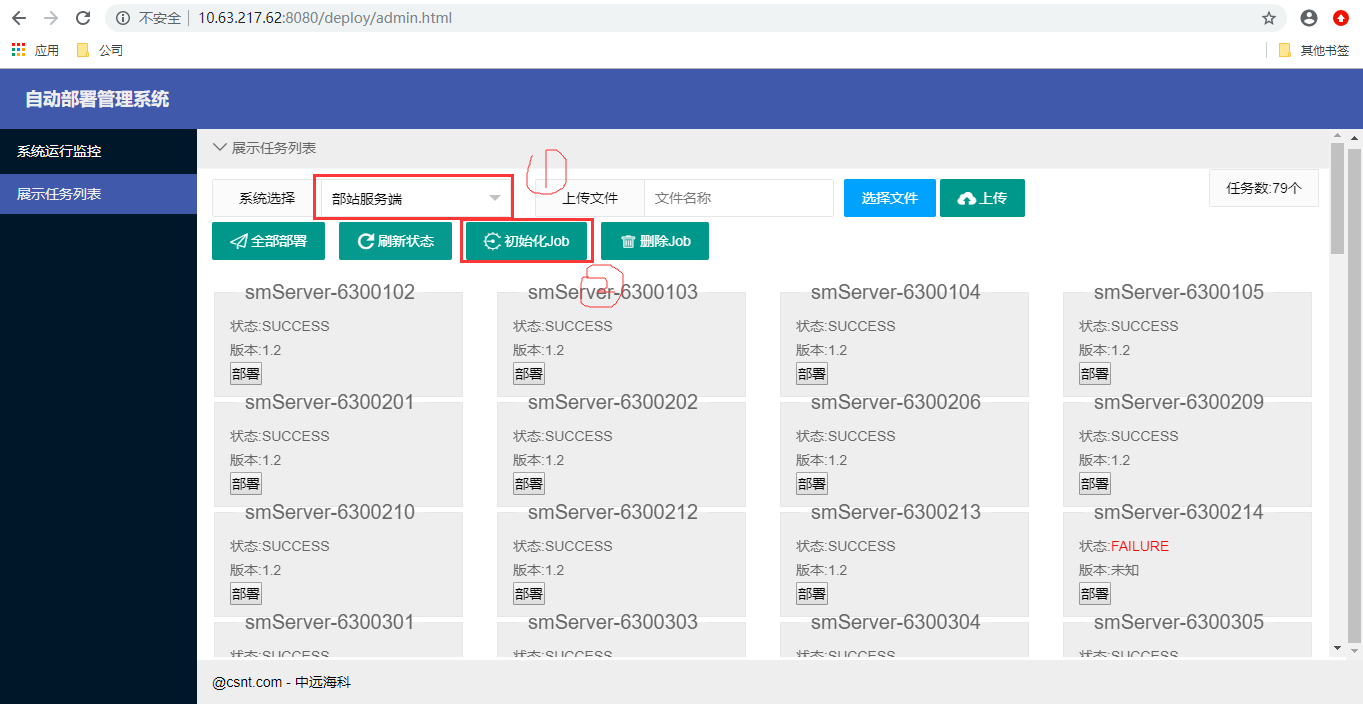
可以通过项目的test中的DESTest类生成，如下图：



# 自动部署系统使用

## 初始化项目job

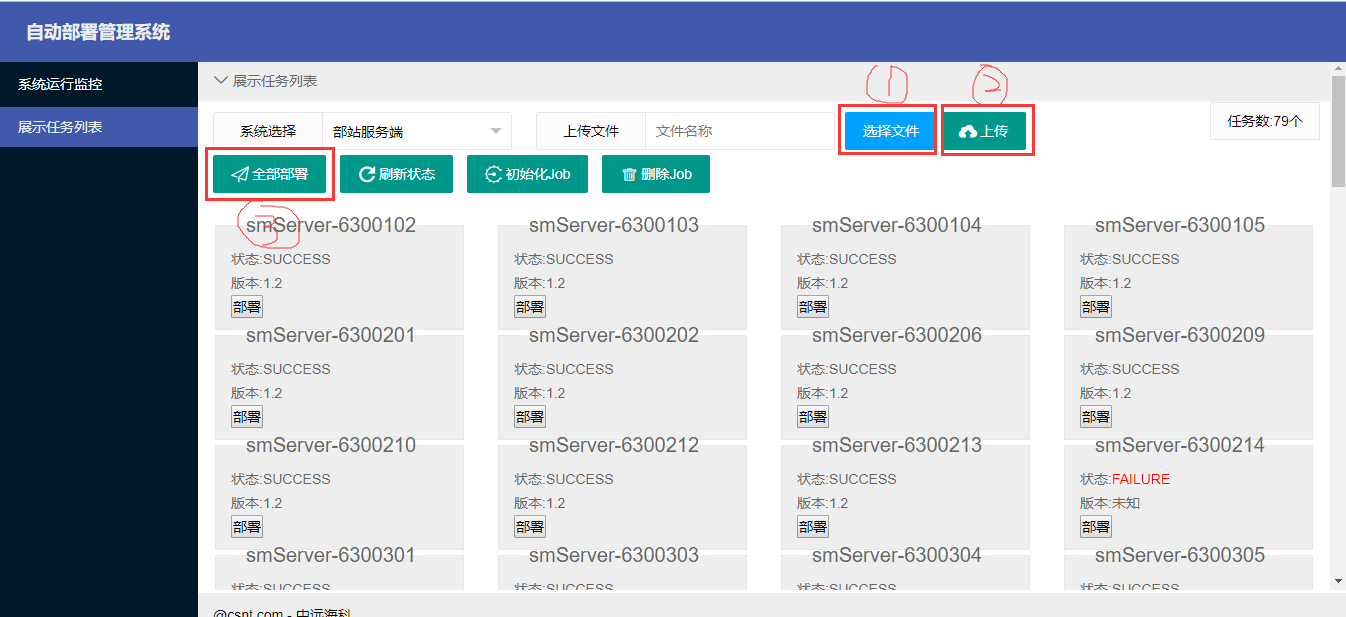
1. 在浏览器中输入<http://ip>:端口/deploy/admin.html，如下图：



1. 选择系统，然后点击“初始化job”即可

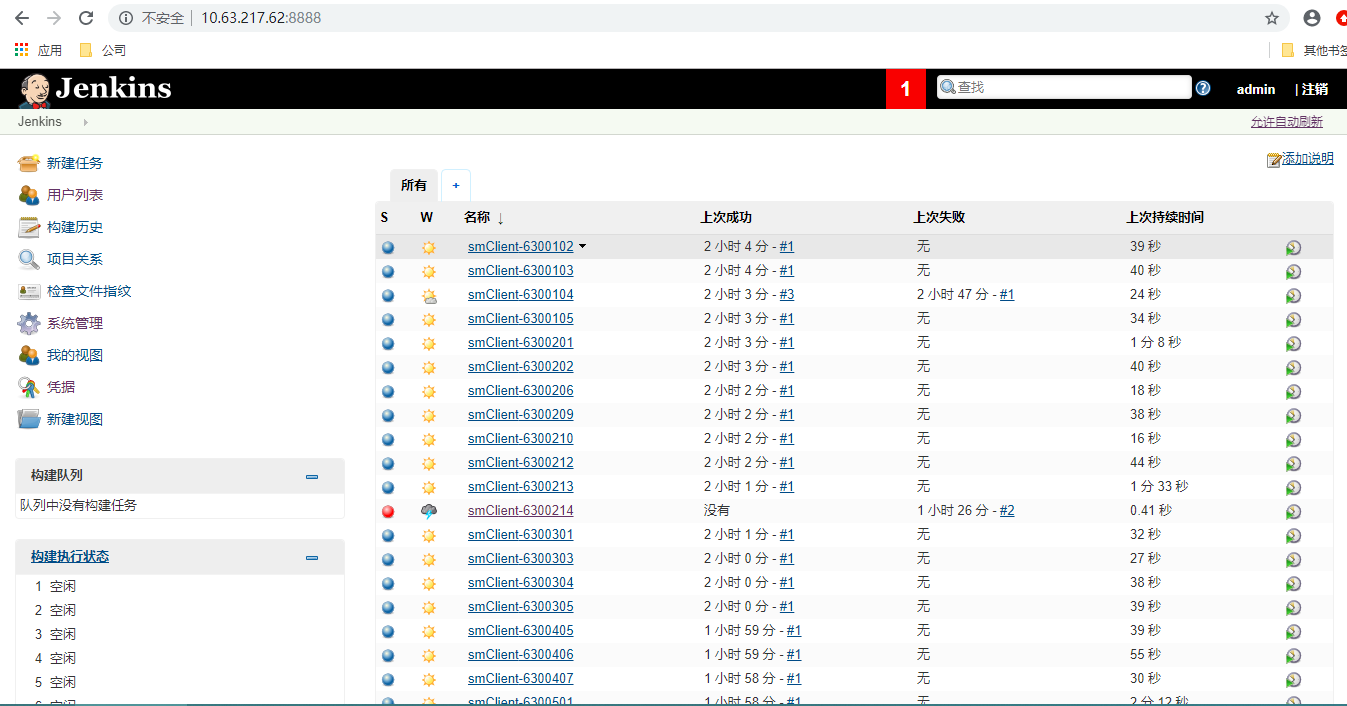
## 开始部署项目

1. 上传已经打好的项目war包，点击“全部部署”即可。



## 查看部署情况

这里部署情况需要在Jenkins的管理界面中查看，其中“蓝色”是部署成功，“红色”是部署失败，如下图：



## 查看部署失败原因

1. 点击失败job，并点击左侧build history



1. 查看“控制台输出”中的日志

