# Jenkins自动化部署CICD技术总结文档

Author: 刘力

Creation Date: 2019-07-03

Last Updated: 2019-07-03

目录

[1 项目技术架构图](#_Toc32048_WPSOffice_Level1) [4](#_Toc32048_WPSOffice_Level1)

[1.1 系统整体架构图](#_Toc22173_WPSOffice_Level2) [4](#_Toc22173_WPSOffice_Level2)

[图1-1 系统架构图](#_Toc22173_WPSOffice_Level3) [4](#_Toc22173_WPSOffice_Level3)

[1.2 系统部署架构图](#_Toc15656_WPSOffice_Level2) [4](#_Toc15656_WPSOffice_Level2)

[图1-2 部署架构图](#_Toc15656_WPSOffice_Level3) [5](#_Toc15656_WPSOffice_Level3)

[图1-3 部署分支架构图](#_Toc4561_WPSOffice_Level3) [5](#_Toc4561_WPSOffice_Level3)

[1.3 系统部署流程图](#_Toc4561_WPSOffice_Level2) [5](#_Toc4561_WPSOffice_Level2)

[图1-4详细部署流程图](#_Toc12325_WPSOffice_Level3) [6](#_Toc12325_WPSOffice_Level3)

[2 Jenkins集群搭建](#_Toc22173_WPSOffice_Level1) [6](#_Toc22173_WPSOffice_Level1)

[2.1 master节点的搭建](#_Toc12325_WPSOffice_Level2) [6](#_Toc12325_WPSOffice_Level2)

[2.1.1 rpm 安装方式](#_Toc8370_WPSOffice_Level3) [6](#_Toc8370_WPSOffice_Level3)

[2.1.2 Tomcat 安装方式](#_Toc19234_WPSOffice_Level3) [7](#_Toc19234_WPSOffice_Level3)

[2.1.3 命令启动安装方式](#_Toc31609_WPSOffice_Level3) [8](#_Toc31609_WPSOffice_Level3)

[2.1.4 将Jenkins设置为自启动](#_Toc12207_WPSOffice_Level3) [8](#_Toc12207_WPSOffice_Level3)

[2.2 slave节点的搭建](#_Toc8370_WPSOffice_Level2) [9](#_Toc8370_WPSOffice_Level2)

[2.2.1 Linux 作为slave节点](#_Toc1741_WPSOffice_Level3) [9](#_Toc1741_WPSOffice_Level3)

[2.2.2 Windows server作为slave节点](#_Toc7136_WPSOffice_Level3) [11](#_Toc7136_WPSOffice_Level3)

[3 项目实现方案](#_Toc15656_WPSOffice_Level1) [15](#_Toc15656_WPSOffice_Level1)

[3.1 maven项目的实现方案](#_Toc19234_WPSOffice_Level2) [15](#_Toc19234_WPSOffice_Level2)

[3.1.1 环境准备](#_Toc27733_WPSOffice_Level3) [15](#_Toc27733_WPSOffice_Level3)

[3.1.2 清空工作区](#_Toc28440_WPSOffice_Level3) [15](#_Toc28440_WPSOffice_Level3)

[3.1.3 拉取代码](#_Toc24793_WPSOffice_Level3) [16](#_Toc24793_WPSOffice_Level3)

[3.1.4 单元测试](#_Toc23087_WPSOffice_Level3) [16](#_Toc23087_WPSOffice_Level3)

[3.1.5 系统构建](#_Toc3150_WPSOffice_Level3) [16](#_Toc3150_WPSOffice_Level3)

[3.1.6 系统备份](#_Toc15801_WPSOffice_Level3) [16](#_Toc15801_WPSOffice_Level3)

[3.1.7 传送war包](#_Toc11152_WPSOffice_Level3) [16](#_Toc11152_WPSOffice_Level3)

[3.1.8 系统发布](#_Toc3953_WPSOffice_Level3) [16](#_Toc3953_WPSOffice_Level3)

[3.1.9 系统回退](#_Toc146_WPSOffice_Level3) [17](#_Toc146_WPSOffice_Level3)

[3.2 C#项目的实现方案](#_Toc31609_WPSOffice_Level2) [17](#_Toc31609_WPSOffice_Level2)

[3.2.1 清空工作区](#_Toc10262_WPSOffice_Level3) [17](#_Toc10262_WPSOffice_Level3)

[3.2.2 拉取代码](#_Toc3393_WPSOffice_Level3) [17](#_Toc3393_WPSOffice_Level3)

[3.2.3 单元测试](#_Toc9546_WPSOffice_Level3) [17](#_Toc9546_WPSOffice_Level3)

[3.2.4 系统构建](#_Toc13951_WPSOffice_Level3) [17](#_Toc13951_WPSOffice_Level3)

[3.2.5 系统备份](#_Toc10022_WPSOffice_Level3) [17](#_Toc10022_WPSOffice_Level3)

[3.2.6 系统发布](#_Toc6097_WPSOffice_Level3) [18](#_Toc6097_WPSOffice_Level3)

[3.2.7 系统回退](#_Toc12188_WPSOffice_Level3) [18](#_Toc12188_WPSOffice_Level3)

[3.3 nodeJs项目的实现方案](#_Toc12207_WPSOffice_Level2) [18](#_Toc12207_WPSOffice_Level2)

[3.3.1 清空工作区](#_Toc21954_WPSOffice_Level3) [18](#_Toc21954_WPSOffice_Level3)

[3.3.2 拉取代码](#_Toc23022_WPSOffice_Level3) [18](#_Toc23022_WPSOffice_Level3)

[3.3.3 单元测试](#_Toc27649_WPSOffice_Level3) [19](#_Toc27649_WPSOffice_Level3)

[3.3.4 系统构建](#_Toc4790_WPSOffice_Level3) [19](#_Toc4790_WPSOffice_Level3)

[3.3.5 系统备份](#_Toc7616_WPSOffice_Level3) [19](#_Toc7616_WPSOffice_Level3)

[3.3.6 系统发布](#_Toc32085_WPSOffice_Level3) [19](#_Toc32085_WPSOffice_Level3)

[3.3.7 系统回退](#_Toc8686_WPSOffice_Level3) [19](#_Toc8686_WPSOffice_Level3)

[3.4 原始servelet web 项目实现方案](#_Toc1741_WPSOffice_Level2) [20](#_Toc1741_WPSOffice_Level2)

[3.4.1 环境准备](#_Toc22984_WPSOffice_Level3) [20](#_Toc22984_WPSOffice_Level3)

[3.4.2 清空工作区](#_Toc12853_WPSOffice_Level3) [20](#_Toc12853_WPSOffice_Level3)

[3.4.3 拉取代码](#_Toc5683_WPSOffice_Level3) [20](#_Toc5683_WPSOffice_Level3)

[3.4.4 系统构建](#_Toc6878_WPSOffice_Level3) [20](#_Toc6878_WPSOffice_Level3)

[3.4.5 系统备份](#_Toc20890_WPSOffice_Level3) [20](#_Toc20890_WPSOffice_Level3)

[3.4.6 传送war包](#_Toc6938_WPSOffice_Level3) [21](#_Toc6938_WPSOffice_Level3)

[3.4.7 系统发布](#_Toc427_WPSOffice_Level3) [21](#_Toc427_WPSOffice_Level3)

[3.4.8 系统回退](#_Toc8249_WPSOffice_Level3) [21](#_Toc8249_WPSOffice_Level3)

[3.5 增量文件上传的实现方案](#_Toc7136_WPSOffice_Level2) [21](#_Toc7136_WPSOffice_Level2)

[3.5.1 环境准备](#_Toc15230_WPSOffice_Level3) [22](#_Toc15230_WPSOffice_Level3)

[3.5.2 清空工作区](#_Toc31693_WPSOffice_Level3) [22](#_Toc31693_WPSOffice_Level3)

[3.5.3 拉取代码](#_Toc24507_WPSOffice_Level3) [22](#_Toc24507_WPSOffice_Level3)

[3.5.4 系统构建](#_Toc29654_WPSOffice_Level3) [22](#_Toc29654_WPSOffice_Level3)

[3.5.5 系统备份](#_Toc11396_WPSOffice_Level3) [22](#_Toc11396_WPSOffice_Level3)

[3.5.6 系统发布](#_Toc17772_WPSOffice_Level3) [22](#_Toc17772_WPSOffice_Level3)

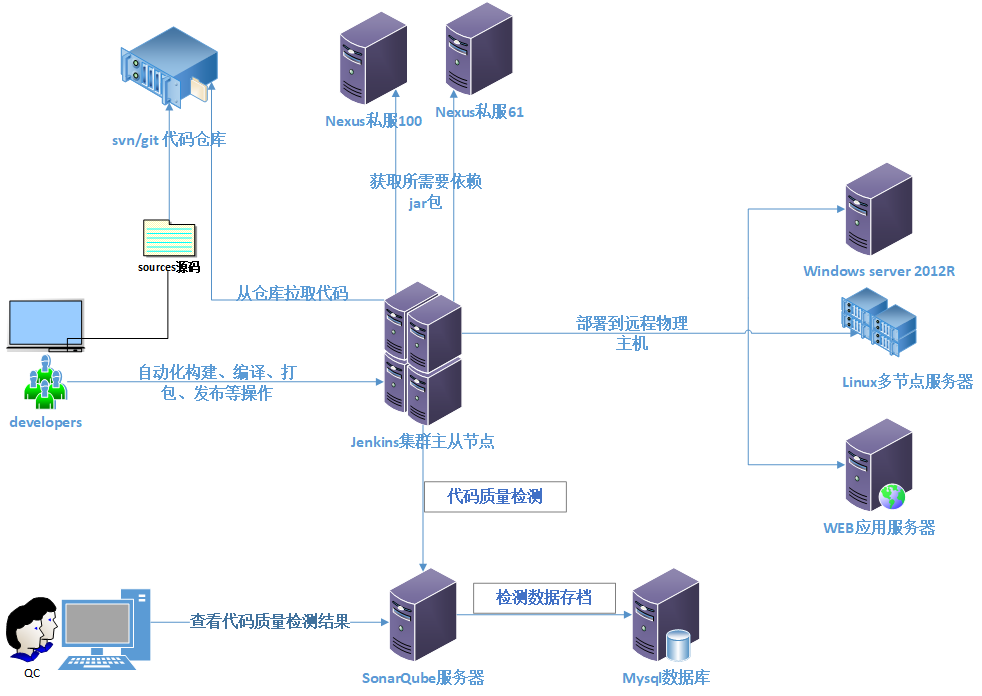
[3.5.7 系统回退](#_Toc3089_WPSOffice_Level3) [23](#_Toc3089_WPSOffice_Level3)

[3.6 php 项目的实现方案](#_Toc27733_WPSOffice_Level2) [23](#_Toc27733_WPSOffice_Level2)

**1 项目技术架构图**

* 1. **系统整体架构图**

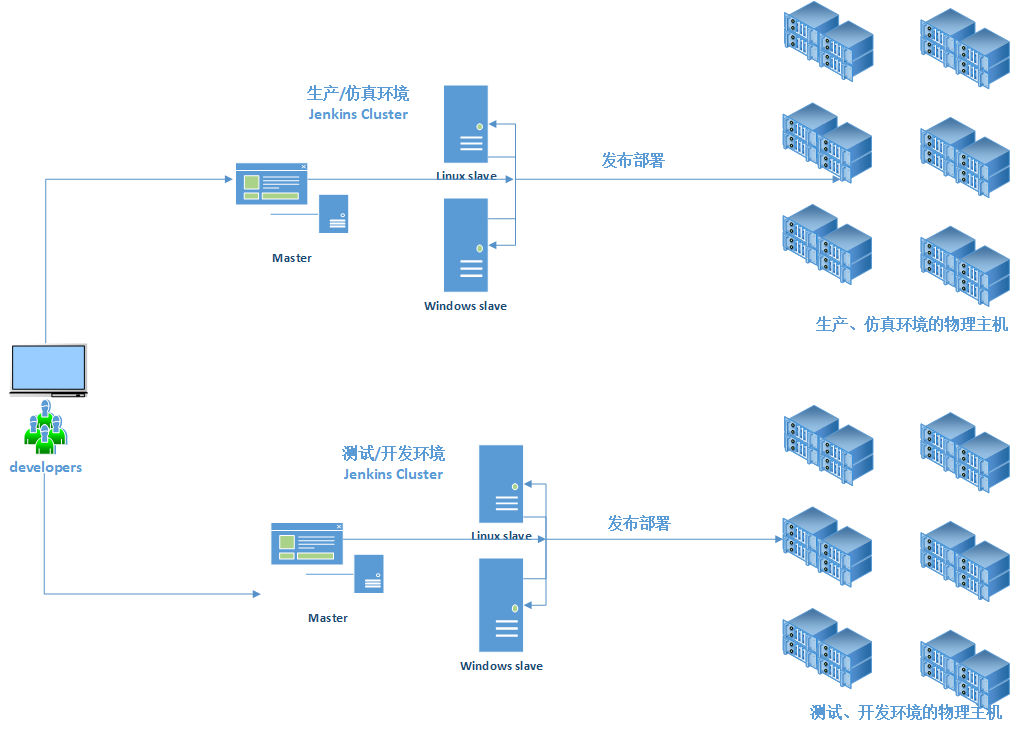
系统整体自动化持续部署架构图如图1-1所示:



**图1-1 系统架构图**

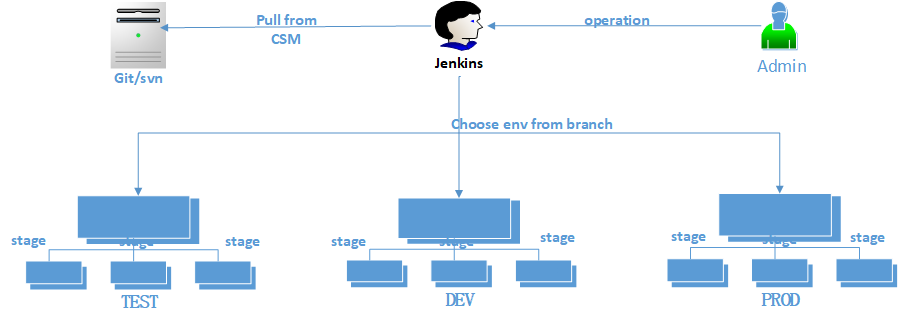
* 1. **系统部署架构图**

部署整体架构图如图1-2所示：



**图1-2 部署架构图**

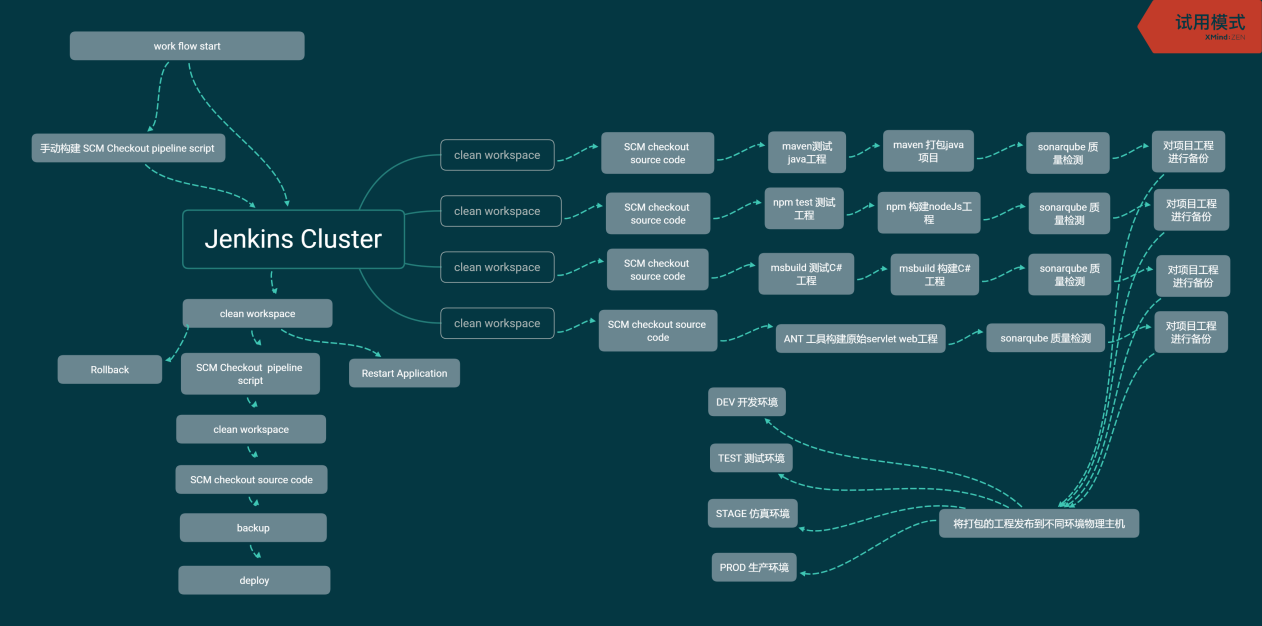
对自动化部署的系统搭建了有测试、开发、生产环境，因此会部署到不同的物理主机上，实现备份部署环境分离，如图1-3所示：



**图1-3 部署分支架构图**

* 1. **系统部署流程图**

系统流水线部署流程图如1-4 所示：



**图1-4详细部署流程图**

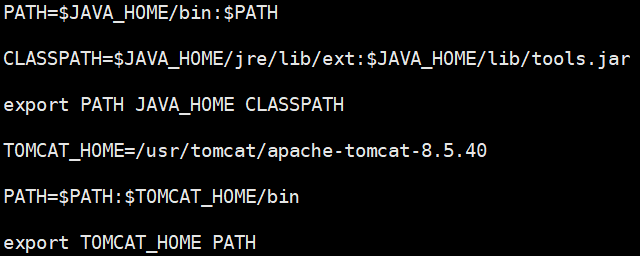
**2 Jenkins集群搭建**

**2.1 master节点的搭建**

Jenkins是由java开发的项目，因此部署起来需要有JDK 的环境；Jenkins官网提供了rpm包的安装方式，也阔以下载jenkins.war 当成一个java工程进行部署；安装一个Tomcat 把war包放入进去启动，也阔以直接java -jar 方式启动。

**2.1.1 rpm 安装方式**

1 安装JDK 用yum源与wget 都阔以，然后加入到用户的环境变量中去。



2 获取jenkins的rpm包，阔以通过wget 或者去官网下载。



自动安装完成之后：

/usr/lib/jenkins/jenkins.war    WAR包

/etc/sysconfig/jenkins       配置文件

/var/lib/jenkins/       默认的JENKINS\_HOME目录

/var/log/jenkins/jenkins.log    Jenkins日志文件

3 配置jenkins端口



4 启动jenkins



5 然后通过浏览器登录建立用户安装插件，初始化需要密码



6 设置为后台自启动应用程序

systemctl enable jenkins

7 可以更改jenkins的home目录，在启动之前通过导入环境变量：

export JENKINS\_HOME=/data/jenkins

**2.1.2 Tomcat 安装方式**

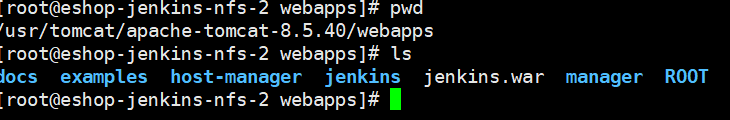
1 安装jdk ，下载Tomcat

2 下载jenkins的war包，在官网上<https://jenkins.io/download/>

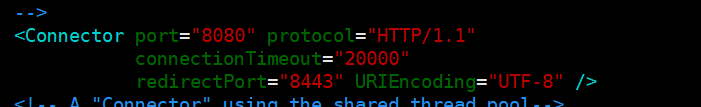
3 把下载的Tomcat 解压放到某个目录，并导入到环境变量中，开机自启动会用到



4 然后把下载下来的jenkins.war 放入webapps中



5 修改server.xml 内容为



6 启动Tomcat 就阔以查看密钥创建用户登录ip+port/jenkins 就阔以正常进入安装



**2.1.3 命令启动安装方式**

同理安装好jdk，导入环境变量中

下载jenkins.war 自定义JENKINS\_HOME

export JENKINS\_HOME=/data/jenkins

命令启动

nohup $JAVA\_HOME/bin/java -jar /usr/local/src/jenkins/jenkins.war --httpPort=8081 2>&1 &

**2.1.4 将Jenkins设置为自启动**

1 创建jenkins.sh 脚本

#!/bin/bash

###主要目的用于开机启动服务,不然 启动jenkins.war包没有java -jar的权限

JAVA\_HOME=/usr/local/src/jdk1.8.0\_192

pid=`ps -ef | grep jenkins.war | grep -v 'grep'| awk '{print $2}'| wc -l`

if [ "$1" = "start" ];then

if [ $pid -gt 0 ];then

echo 'jenkins is running...'

else

### java启动服务 配置java安装根路径,和启动war包存的根路径

nohup $JAVA\_HOME/bin/java -jar /usr/local/src/jenkins/jenkins.war --httpPort=8081 2>&1 &

fi

### java启动服务 配置java安装根路径,和启动Tomcat war包存的根路径

### $TOMCAT\_HOME/bin/startup.sh

elif [ "$1" = "stop" ];then

exec ps -ef | grep jenkins | grep -v grep | awk '{print $2}'| xargs kill -9

echo 'jenkins is stop..'

else

echo "Please input like this:"./jenkins.sh start" or "./jenkins stop""

Fi

2 **让jenkins.sh有可执行权限**



3 在**/lib/systemd/system下创建jenkins.service**

[Unit]

Description=Jenkins

After=network.target

[Service]

Type=forking

ExecStart=/usr/local/src/jenkins/jenkins.sh start

ExecReload=ExecStop=/usr/local/src/jenkins/jenkins.sh stop

PrivateTmp=true

[Install]

WantedBy=multi-user.target

4 **创建好服务后,执行一下命令刷新配置**



5 **设置开机启动**



**2.2 slave节点的搭建**

**2.2.1 Linux 作为slave节点**

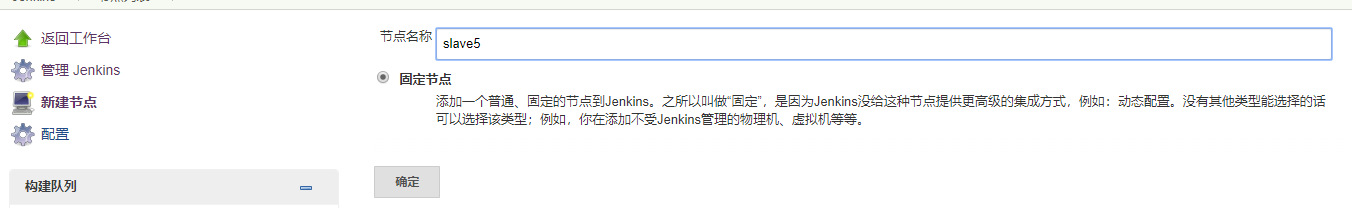
把Linux系统作为jenkins cluster 的从节点时，因为linux系统自带有SSH 所以只要用户指定的工作的目录有相应的权限即并且已经安装jdk

1登录主节点搭建好的jenkins 的web界面

2进入系统管理下的节点管理

3然后点击左上角新建节点

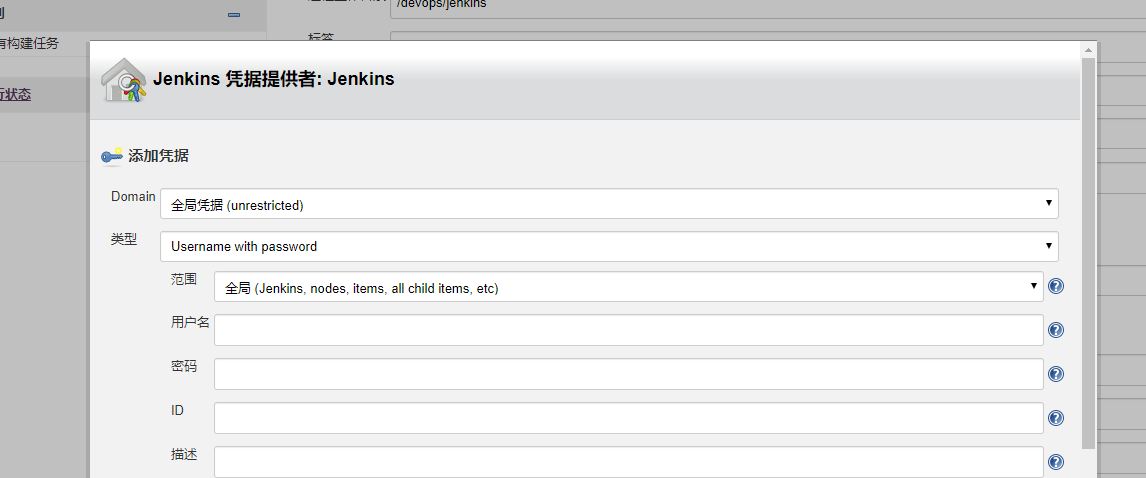
4输入节点名字



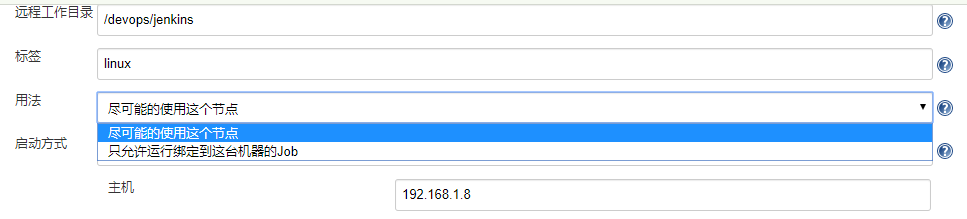
5 必须填写标签一栏，作为指定agent 时供主节点去调用的标识符



6 添加主机以及凭据



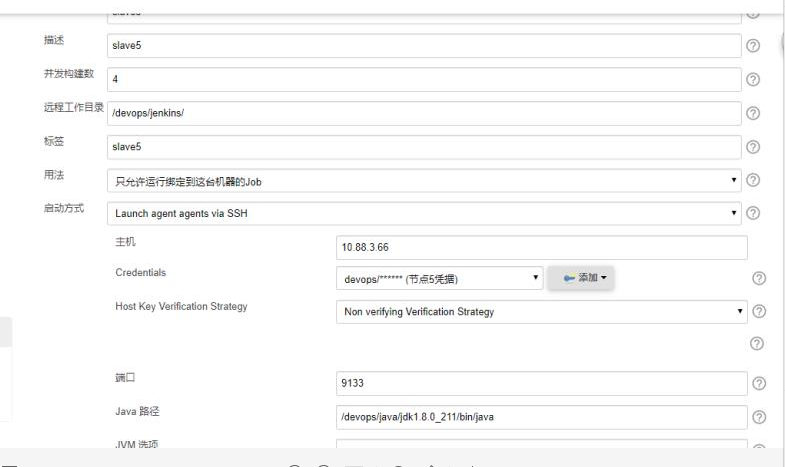
7 特别说明用法，有两个选项



第一个选项：只要这个节点可用就会使用它，这个节点空闲就会优先使用

第二个选项：只有在脚本里知名该节点的名字才能调度该节点进行使用

8 配置主机的端口以及jdk路径

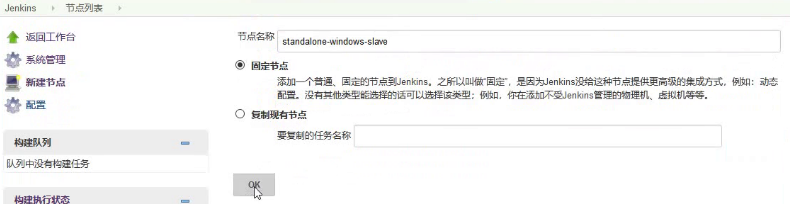


9点击保存，它就能自动连接从节点

**2.2.2 Windows server作为slave节点**

1 登录jenkins 服务器后，点击系统管理--》节点管理

2 点击新建节点，输入节点名，选择固定节点



3 进入系统管理--》全局安全配置开启代理可以选择指定也可以选择随机然后保存



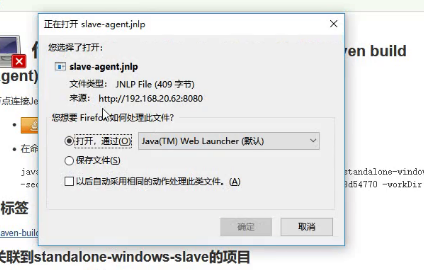
4 进入到节点的详细配置中去，启动方式选择通过java web启动代理然后保存



5 然后进入到启动界面，点击LANUCH去下载一个jar包传送到windows上，或者在windows主机上之执行下面的命令



6 现在通过LAUNCH按钮去下载jar

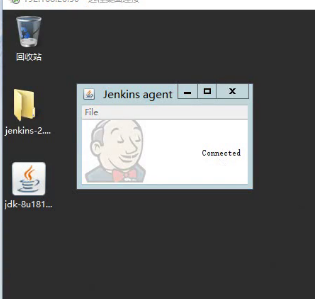


7 传送到windows 服务器上执行该jnlp，确保网络能通，防火墙关闭必须保障你jenkins 主节点上刚开始制定的端口号的防火墙是处于打开状态，否则无法连接到master节点

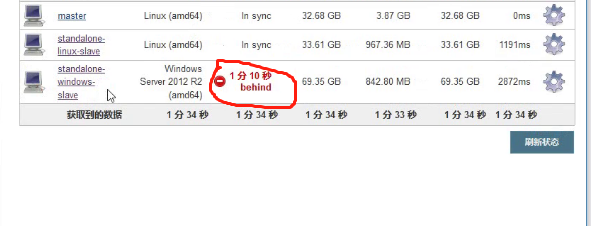
8 选择打开方式以java web launcher 的方式启动传送过的jnlp 文件



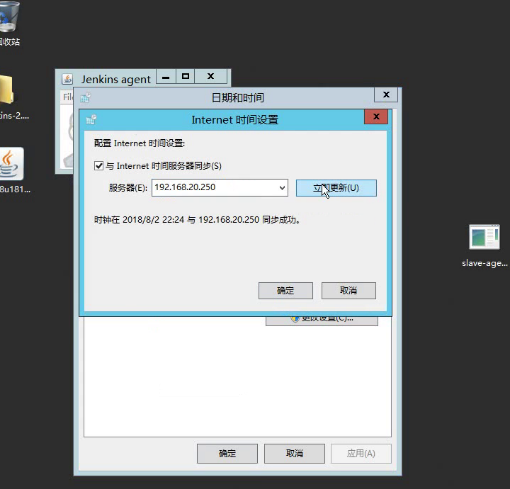
9 然后就能连接上，就能成为从节点；如果一直处于连接状态检查防火墙。检查jnlp协议指定的端口号是否能经过防火墙



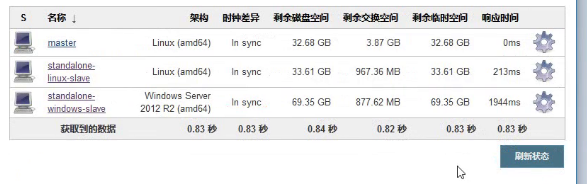
10 可能出现时间落后的情况。时间并没有同步



11 设置你windows server 上时钟的ip为你主节点的ip例如



12 然后刷新web页面，就会发现时间同步一致



# 3 项目实现方案

**3.1 maven项目的实现方案**

* 普通用户登录jenkins，点击所需要构建的项目进行构建
* 如果选择Deploy，并选择需要发布的节点（默认选择已配置的所有节点）则会执行出系统回退外的所有步骤
* 如果选择RollBack，则需要选择回退的部署包，并执环境准备以及系统回退步骤。
* 如果选择restartApplication ，则操作远程服务器，执行环境准备停止并重启应用
* 系统分为构建Build，Deploy，RollBack ，把构建和部署分离，分两步走
* 当选择Build 时，会拉下代码，构建打包，结束，然后再选择Deploy时会看到下面有个参数pre\_war\_time表示上次构建的打包的时间，执行deploy时直接发包并部署重新启动应用

### 3.1.1 环境准备

此步骤将检查目标服务器系统运行所需要的环境：

（1）检查是否需要创建运行用户，需要且无则创建

（2）检查是否需要JDK or OEPNJDK，需要且无则安装

（3）检查是否需要tomcat or jboss or weblogic，需要且无则安装

（4）检查是否需要apmgent，需要且无则安装

（5）检查项目目录，无则创建并传输必须文件

### 3.1.2 清空工作区

此步骤将清空上一次构建遗留在工作区的文件，避免缓存，避免磁盘占用浪费。

### 3.1.3 拉取代码

此步骤将从git仓库或者svn拉取代码。

### 3.1.4 单元测试

此步骤将执行maven单元测试，如果项目目录下存在测试用例。

### 3.1.5 系统构建

此步骤将执行maven构建。mvn deploy 根据需求加参数

### 3.1.6 系统备份

（1）此步骤将从目标服务器备份上一次发布的部署包：

（2）创建jenkins服务器备份目录

（3）检查备份目录下是否有备份文件

（4）如果没有创建本地备份临时目录

（5）拉取目标服务器上的部署代码到临时目录

（6）压缩发送到jenkins备份目录

（7）删除临时目录

### 3.1.7 传送war包

（1）建立临时目录

（2）jenkins将war传送到临时目录

### 3.1.8 系统发布

此步骤将把构建好的部署包，发布到目标服务器项目目录上。**此步骤在开始前，需要用户介入，确认并输入yes后继续。**

（1）停止目标服务器tomcat

（2）清空目标服务器项目启动

（3）从临时目录将war包移动到给目标目录

（4）解压war

（5）删除war包

（6）启动目标服务器tomcat

### 3.1.9 系统回退

此步骤将把用于选择的部署包，发到目标服务器项目目录上，**此步骤在开始前，需要用户介入，确认并输入yes后继续。**

（1）停止远程tomcat

（2）清空目标服务器项目启动目录

（3）jenkins发送war包给目标服务器目录

（4）解压war包

（5）删除war包

（6）启动tomcat

**3.2 C#项目的实现方案**

* 普通用户登录jenkins，点击所需要构建的项目进行构建
* 如果选择Deploy，并选择需要发布的节点（默认选择已配置的所有节点）则会执行出系统回退外的所有步骤
* 如果选择RollBack，则需要选择回退的部署包，并执环境准备以及系统回退步骤。
* 如果选择restartApplication ，则操作远程服务器，执行环境准备停止并重启应用

### 3.2.1 清空工作区

此步骤将清空上一次构建遗留在工作区的文件，避免缓存，避免磁盘占用浪费。

### 3.2.2 拉取代码

此步骤将从git仓库或者svn拉取代码。

### 3.2.3 单元测试

此步骤将执行msbuild单元测试，如果项目目录下存在测试用例。

### 3.2.4 系统构建

此步骤将执行msbuild构建。例如Msbuild.exe project.proj /t:rebuild

### 3.2.5 系统备份

此步骤将从目标服务器备份上一次发布的部署包：

（1）创建jenkins服务器备份目录

（2）检查备份目录下是否有备份文件

（3）如果没有创建本地备份临时目录

（4）拉取目标服务器上的部署代码到临时目录

（5）压缩发送到jenkins备份目录

（6）删除临时目录

### 3.2.6 系统发布

此步骤将把构建好的部署包，发布到目标服务器项目目录上。**此步骤在开始前，需要用户介入，确认并输入yes后继续。**

（1）jenkins将编译好的dll文件发送给目标服务器

（2）将目录下相同的dll文件覆盖

（3）传送配置文件，覆盖配置文件

### 3.2.7 系统回退

此步骤将把用于选择的部署包，发到目标服务器项目目录上，**此步骤在开始前，需要用户介入，确认并输入yes后继续。**

（1）清空目标服务器项目启动目录

（2）jenkins发送压缩包包给目标服务器目录

（3）解压缩

**3.3 nodeJs项目的实现方案**

* 普通用户登录jenkins，点击所需要构建的项目进行构建：完成仿真环境的搭建
* 如果选择Deploy，并选择需要发布的节点（默认选择已配置的所有节点）则会执行出系统回退外的所有步骤
* 如果选择RollBack，则需要选择回退的部署包，并执环境准备以及系统回退步骤。
* 如果选择restartApplication ，则操作远程服务器，执行环境准备停止并重启应用

### 3.3.1 清空工作区

此步骤将清空上一次构建遗留在工作区的文件，避免缓存，避免磁盘占用浪费。

### 3.3.2 拉取代码

此步骤将从git仓库或者svn拉取代码。

### 3.3.3 单元测试

此步骤将执行npm单元测试，如果项目目录下存在测试用例。

### 3.3.4 系统构建

（1）此步骤将执行npm构建。 npm build

（2）压缩dist目录为zip

### 3.3.5 系统备份

此步骤将从目标服务器备份上一次发布的部署包：

（1）创建jenkins服务器备份目录

（2）检查备份目录下是否有备份文件

（3）如果没有创建本地备份临时目录

（4）拉取目标服务器上的部署代码到临时目录

（5）压缩发送到jenkins备份目录

（6）删除临时目录

### 3.3.6 系统发布

此步骤将把构建好的压缩包，发布到目标服务器项目目录上。**此步骤在开始前，需要用户介入，确认并输入yes后继续。**

（1）清空目标服务器项目目录

（2）jenkins发送zip包给目标服务器

（3）解压zip包

（4）覆盖目录源文件

### 3.3.7 系统回退

此步骤将把用于选择的压缩包，发到目标服务器项目目录上，**此步骤在开始前，需要用户介入，确认并输入yes后继续。**

（1）清空目标服务器项目目录

（2）jenkins发压缩包给目标服务器目录

（3）解压压缩包

（4）把项目源文件全部覆盖

**3.4 原始servelet web 项目实现方案**

* 普通用户登录jenkins，点击所需要构建的项目进行构建：完成仿真环境的搭建
* 如果选择Deploy，并选择需要发布的节点（默认选择已配置的所有节点）则会执行出系统回退外的所有步骤
* 如果选择RollBack，则需要选择回退的部署包，并执环境准备以及系统回退步骤。
* 如果选择restartApplication ，则操作远程服务器，执行环境准备停止并重启应用

### 3.4.1 环境准备

此步骤将检查目标服务器系统运行所需要的环境：

（1）检查是否需要创建运行用户，需要且无则创建

（2）检查是否需要JDK or OEPNJDK，需要且无则安装

（3）检查是否需要tomcat or jboss or weblogic，需要且无则安装

（4）检查是否需要apmgent，需要且无则安装

（5）检查项目目录，无则创建并传输必须文件

### 3.4.2 清空工作区

此步骤将清空上一次构建遗留在工作区的文件，避免缓存，避免磁盘占用浪费。

### 3.4.3 拉取代码

此步骤将从git仓库或者svn拉取代码。

### 3.4.4 系统构建

此步骤将执行ant构建。直接执行ant命令

### 3.4.5 系统备份

（1）此步骤将从目标服务器备份上一次发布的部署包：

（2）创建jenkins服务器备份目录

（3）检查备份目录下是否有备份文件

（4）如果没有创建本地备份临时目录

（5）拉取目标服务器上的部署代码到临时目录

（6）压缩发送到jenkins备份目录

（7）删除临时目录

### 3.4.6 传送war包

（1）建立临时目录

（2）jenkins将war传送到临时目录

### 3.4.7 系统发布

此步骤将把构建好的部署包，发布到目标服务器项目目录上。**此步骤在开始前，需要用户介入，确认并输入yes后继续。**

（1）停止目标服务器

（2）清空目标服务器项目启动

（3）从临时目录将war包移动到给目标目录

（4）解压war

（5）删除war包

（6）启动目标服务器

### 3.4.8 系统回退

此步骤将把用于选择的部署包，发到目标服务器项目目录上，**此步骤在开始前，需要用户介入，确认并输入yes后继续。**

（1）停止远程

（2）清空目标服务器项目启动目录

（3）jenkins发送war包给目标服务器目录

（4）解压war包

（5）删除war包

（6）启动tomcat

**3.5 增量文件上传的实现方案**

* 普通用户登录jenkins，点击所需要构建的项目进行构建：完成仿真环境的搭建
* 如果选择Deploy，并选择需要发布的节点（默认选择已配置的所有节点）则会执行出系统回退外的所有步骤
* 如果选择RollBack，则需要选择回退的部署包，并执环境准备以及系统回退步骤。
* 如果选择restartApplication ，则操作远程服务器，执行环境准备停止并重启应用

### 3.5.1 环境准备

此步骤将检查目标服务器系统运行所需要的环境：

（1）检查是否需要创建运行用户，需要且无则创建

（2）检查是否需要JDK or OEPNJDK，需要且无则安装

（3）检查是否需要tomcat or jboss or weblogic，需要且无则安装

（4）检查是否需要apmgent，需要且无则安装

（5）检查项目目录，无则创建并传输必须文件

### 3.5.2 清空工作区

此步骤将清空上一次构建遗留在工作区的文件，避免缓存，避免磁盘占用浪费。

### 3.5.3 拉取代码

此步骤将从git仓库或者svn拉取代码。

### 3.5.4 系统构建

（1）执行svn diff oldversion:head -r 查看当前版本与之前部署的版本有那些文件发生更改

（2）形成文件列表，然后找到这些更改的文件列表压缩

### 3.5.5 系统备份

（1）此步骤将从目标服务器备份上一次发布的部署包：

（2）创建jenkins服务器备份目录

（3）检查备份目录下是否有备份文件

（4）如果没有创建本地备份临时目录

（5）拉取目标服务器上的部署代码到临时目录

（6）压缩发送到jenkins备份目录

（7）删除临时目录

### 3.5.6 系统发布

此步骤将把搭好的压缩包，发布到目标服务器项目目录上。**此步骤在开始前，需要用户介入，确认并输入yes后继续。**

（1）清空目标服务器项目目录

（2）jenkins发送压缩包给目标服务器

（3）解压压缩包

（4）覆盖项目目录源文件

### 3.5.7 系统回退

此步骤将把用于选择的压缩包，发到目标服务器项目目录上，**此步骤在开始前，需要用户介入，确认并输入yes后继续。**

（1）清空目标服务器项目目录

（2）jenkins发送压缩包给目标服务器目录

（3）解压压缩包

（4）覆盖这个歌源文件

**3.6 php 项目的实现方案**

此项目部署方式采用增量部署方案进行热部署。