前置阅读：

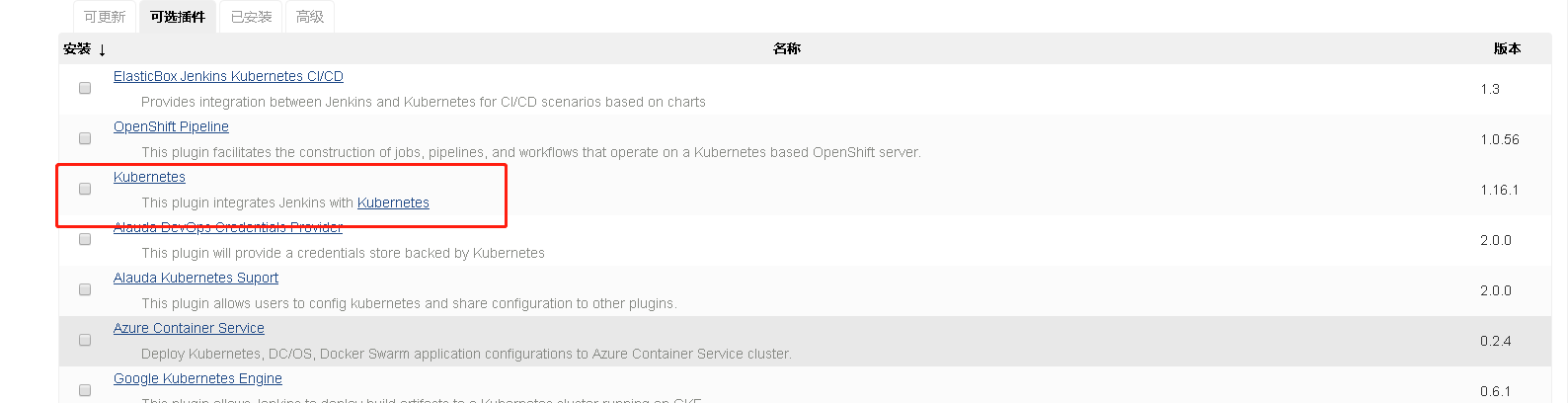
(1).jenkins slave pod配置

jenkins依赖插件清单

kubernetes

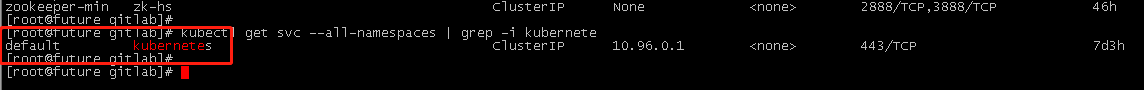
managed scripts

第1步. 我们需要安装kubernetes plugin， 点击 Manage Jenkins -> Manage Plugins -> Available -> Kubernetes plugin 勾选安装即可。 kubernetes plugin：



第2步. 安装完毕后，点击 Manage Jenkins —> Configure System —> (拖到最下方)Add a new cloud —> 选择 Kubernetes，然后填写 Kubernetes 和 Jenkins 配置信息。

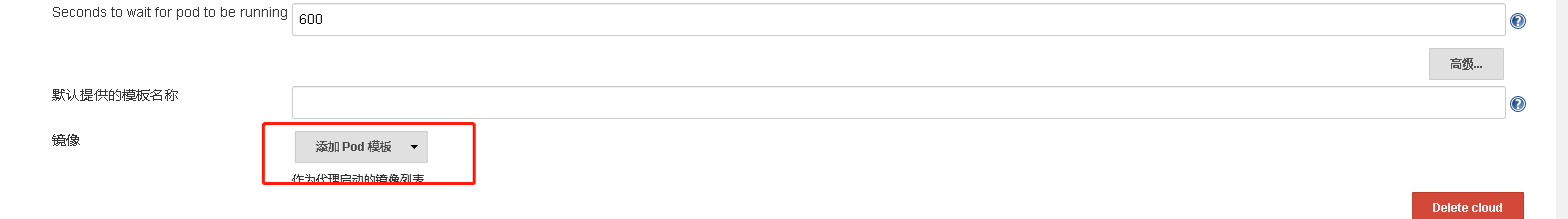
kubernetes地址写https://kubernetes.default，实际上就是kube-apiserver。kubernetes表示service域名,default表示命名空间，如：



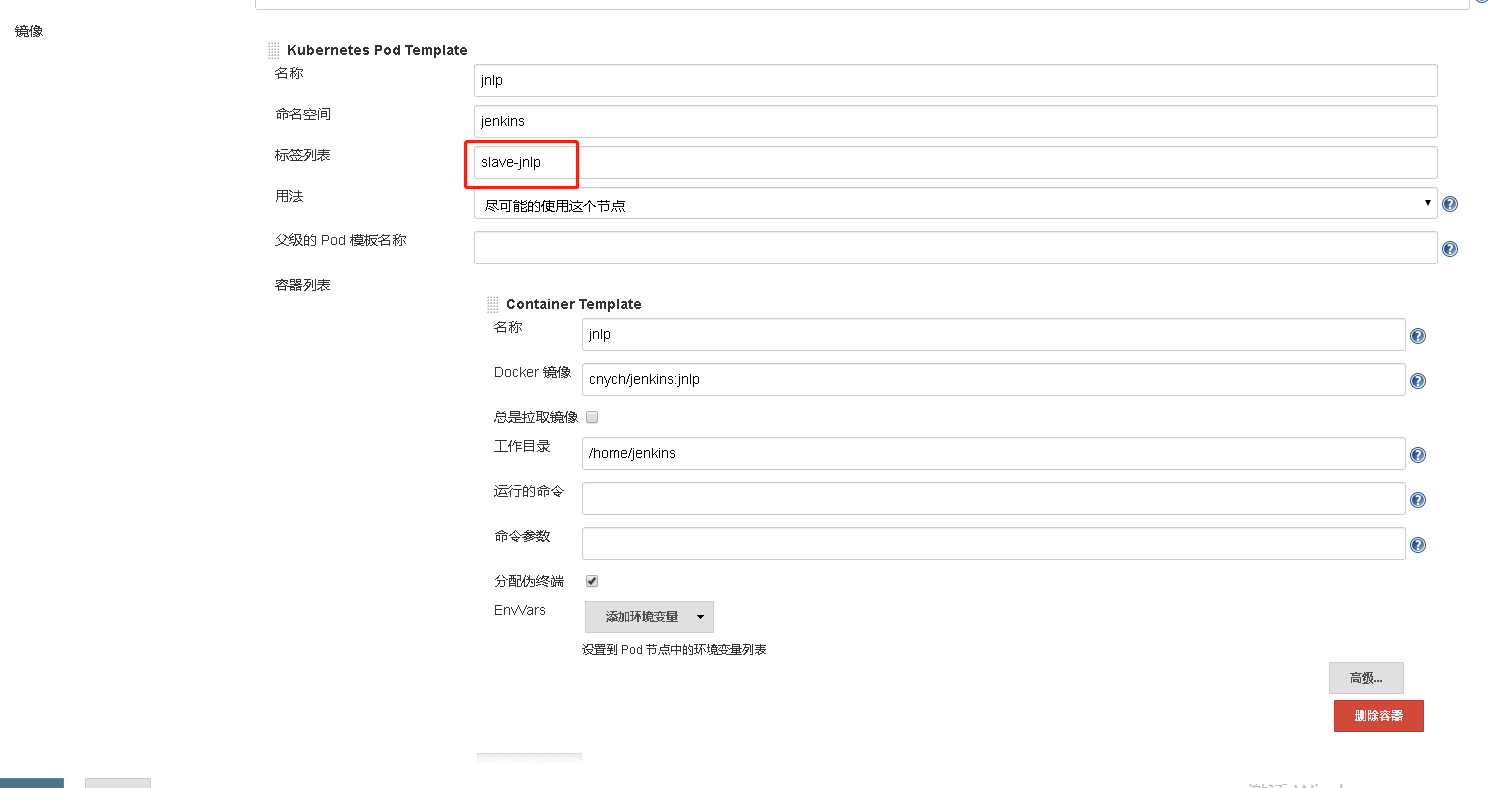


第3步. 配置 Pod Template，其实就是配置 Jenkins Slave 运行的 Pod 模板，命名空间我们同样是用 jenkins，Labels 这里也非常重要，对于后面执行 Job 的时候需要用到该值，然后我这里使用的是 cnych/jenkins:jnlp 这个官方镜像。这个镜像是在官方的 jnlp 镜像基础上定制的，加入了 kubectl 等一些实用的工具。

点击添加pod模板：

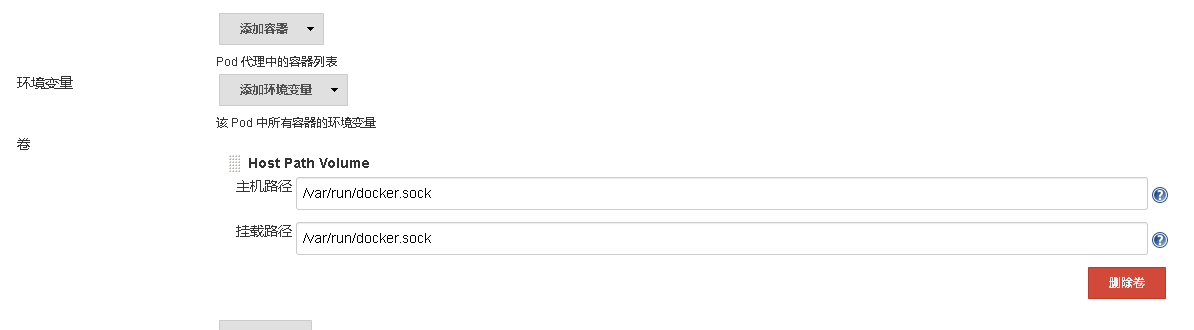


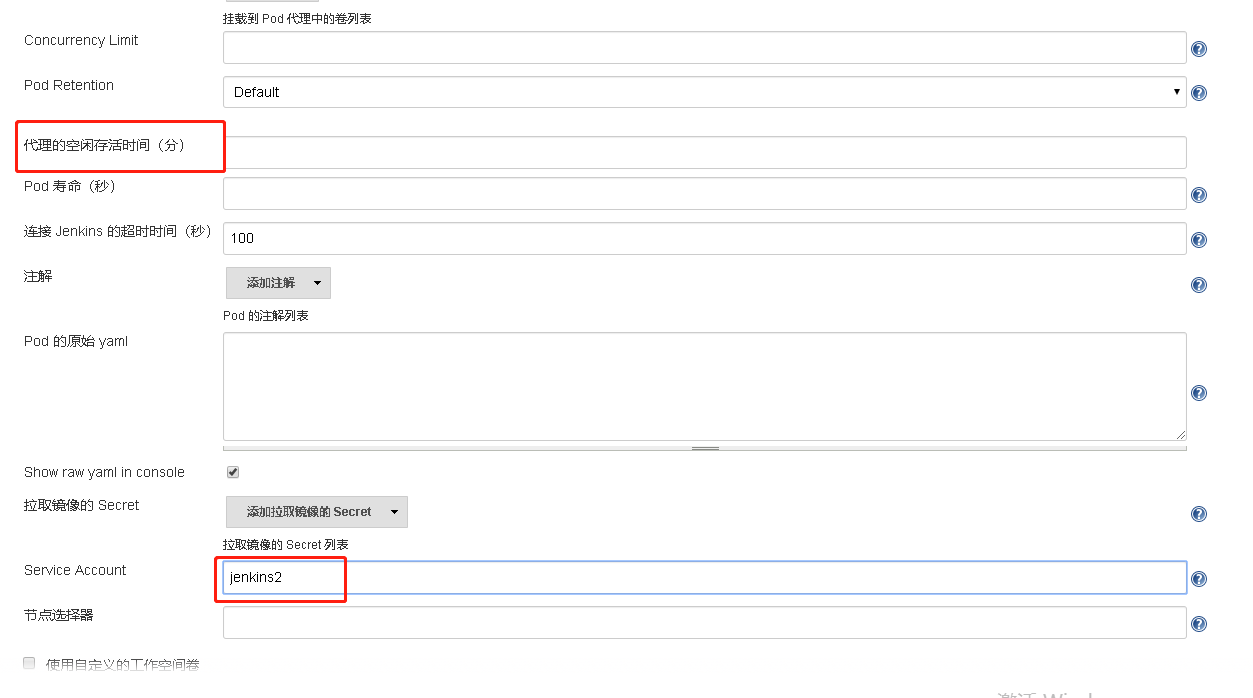
配置Pod Template：



另外需要注意我们这里需要在下面挂载一个主机目录，一个是 /var/run/docker.sock，该文件是用于 Pod 中的容器能够共享宿主机的 Docker，这就是大家说的 docker in docker 的方式，Docker 二进制文件我们已经打包到上面的镜像中了。如果在slave agent中想要访问kubernetes 集群中其他资源，我们还需要绑定之前创建jenkins-maseter是创建的Service Account 账号:jenkins2。

代理的空闲存活时间：表示这个slave pod空闲等待时间最大是多久，超过这个时间会被销毁。





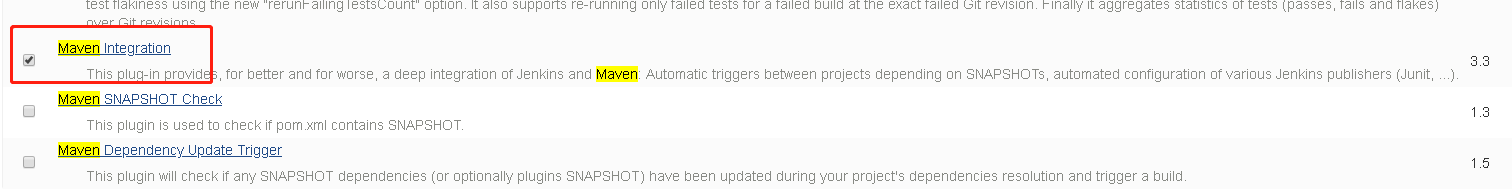
另外还有几个参数需要注意，如上图中的“代理的空闲存活时间”，这个参数表示的意思是当处于空闲状态的时候保留 Slave Pod 多长时间，这个参数最好我们保存默认就行了，如果你设置过大的话，Job 任务执行完成后，对应的 Slave Pod 就不会立即被销毁删除。

我没有把slave调成功，取了一个巧，把系统配置

(2).maven插件安装

点击安装Manage Jenkins -> Manage Plugins -> Available->Maven Integration

有可能plugin插件网站很慢，第一次可能失败，多执行几次。



配置maven到jenkins：

点击系统管理->系统设置



配置说明：

-Dmaven.wagon.http.ssl.insecure=true

-Dmaven.wagon.http.ssl.allowall=true

-Dmaven.wagon.http.ssl.ignore.validity.dates=true

http.ssl.insecure和http.ssl.allowall表示忽略SSL错误(并信任所有证书)。

http.ssl.ignore.validity.dates表示忽略证书日期的问题。

(3).其他系统配置

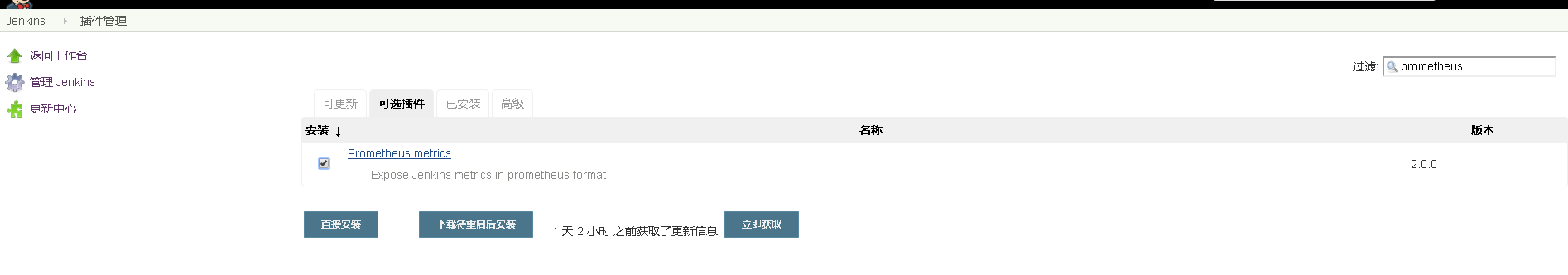
点击系统管理->系统设置：



(4).prometheus相关插件

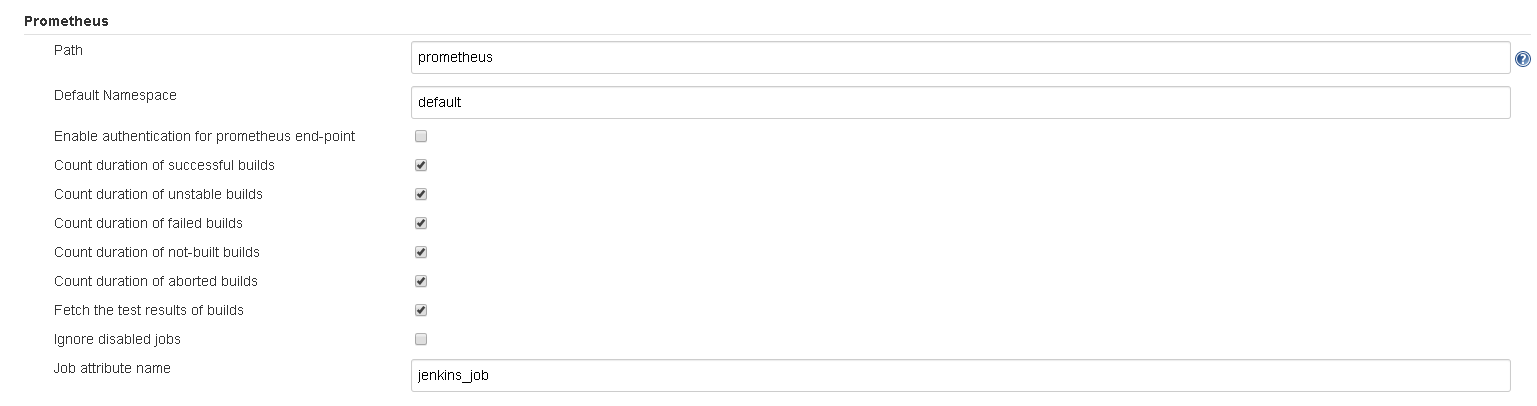
监控/统计jenkins的使用情况。

同样由于网络原因，可能要执行多次安装；由于我用的是阿里云ECS，还不行的话，打开8080端口。



安装完毕后进入：系统管理->系统设置

查看prometheus配置：



(5).Publish Over SSH插件

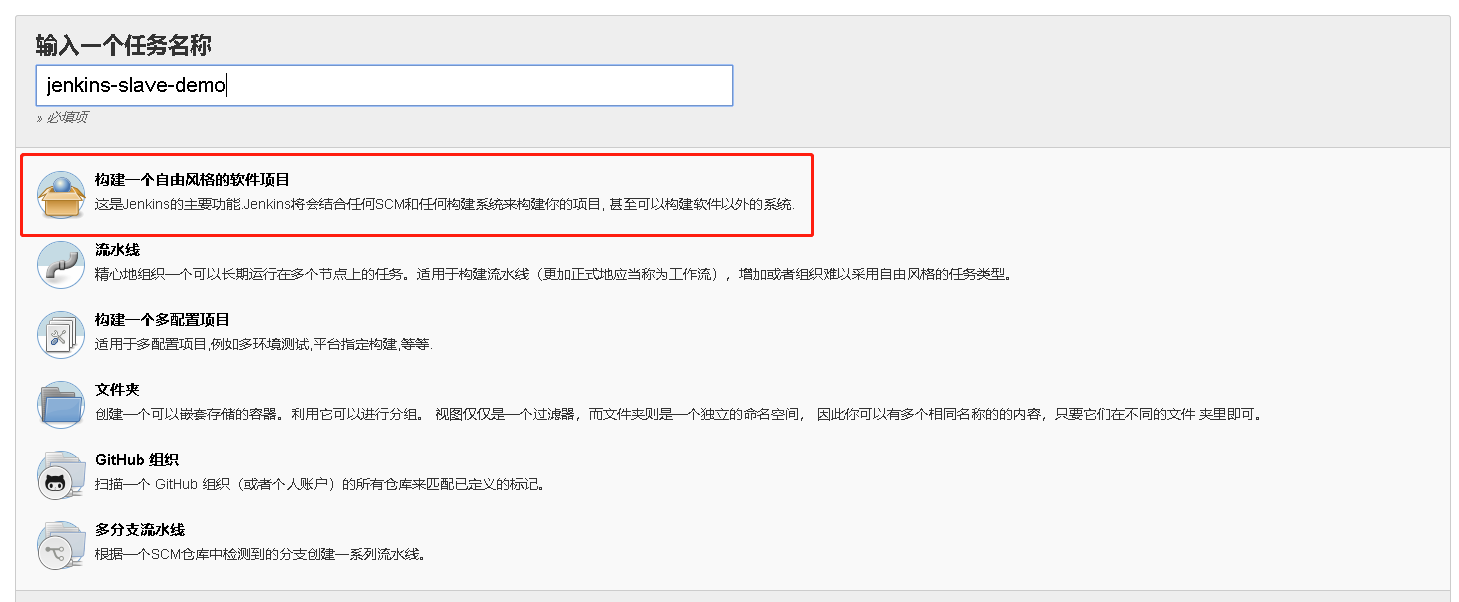
实现远程部署：



(6).测试jenkins slave pod(我没成功，给master节点打slave-jnlp取巧成功)

Kubernetes 插件的配置工作完成了，接下来我们就来添加一个 Job 任务，看是否能够在 Slave Pod 中执行，任务执行完成后看 Pod 是否会被销毁

在 Jenkins 首页点击create new jobs，创建一个测试的任务，输入任务名称，然后我们选择 Freestyle project 类型的任务： jenkins demo



注意在下面的 Label Expression 这里要填入haimaxy-jnlp，就是前面我们配置的 Slave Pod 中的 Label，这两个地方必须保持一致：



然后往下拉，在 Build 区域选择Execute shell：

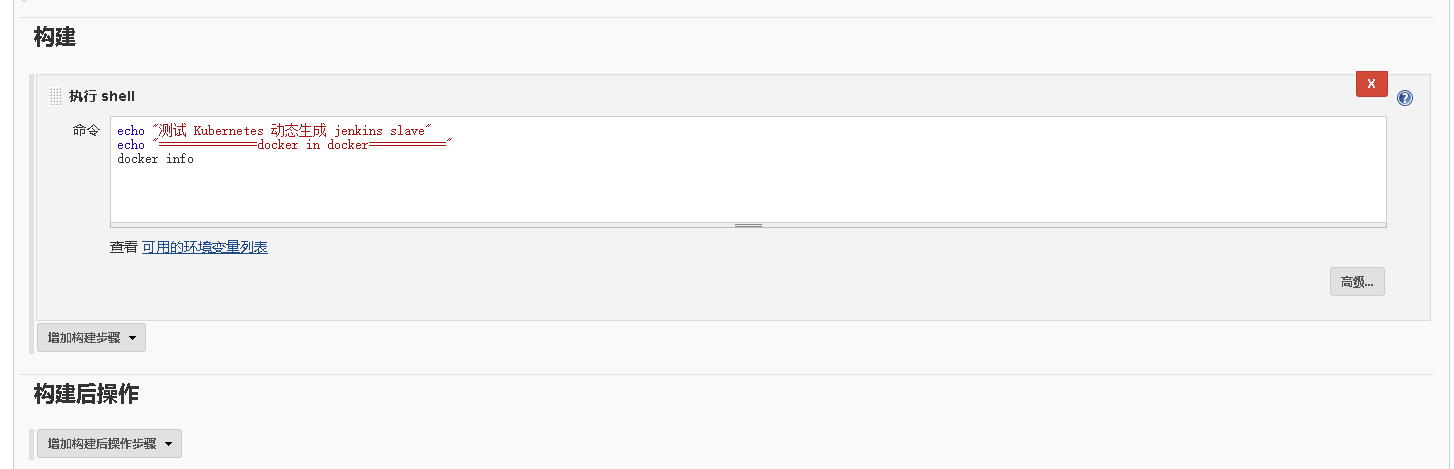
然后输入我们测试命令

echo "测试 Kubernetes 动态生成 jenkins slave"

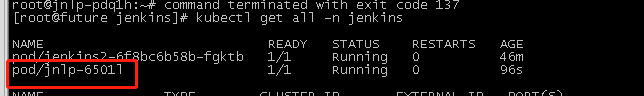
echo "==============docker in docker==========="

docker info

最后点击保存：



查看jenkins命名空间可以看到多了一个pod，过一会执行完毕后自动消失。



(7).创建一个pipeline

创建一个简单的 Pipeline

我们这里来给大家快速创建一个简单的 Pipeline，直接在 Jenkins 的 Web UI 界面中输入脚本运行。

新建 Job：在 Web UI 中点击 New Item -> 输入名称：pipeline-demo -> 选择下面的 Pipeline -> 点击 OK

配置：在最下方的 Pipeline 区域输入如下 Script 脚本，然后点击保存：

node('slave-jnlp') {

stage('Clone') {

echo "1.Clone Stage"

}

stage('Test') {

echo "2.Test Stage"

}

stage('Build') {

echo "3.Build Stage"

}

stage('Deploy') {

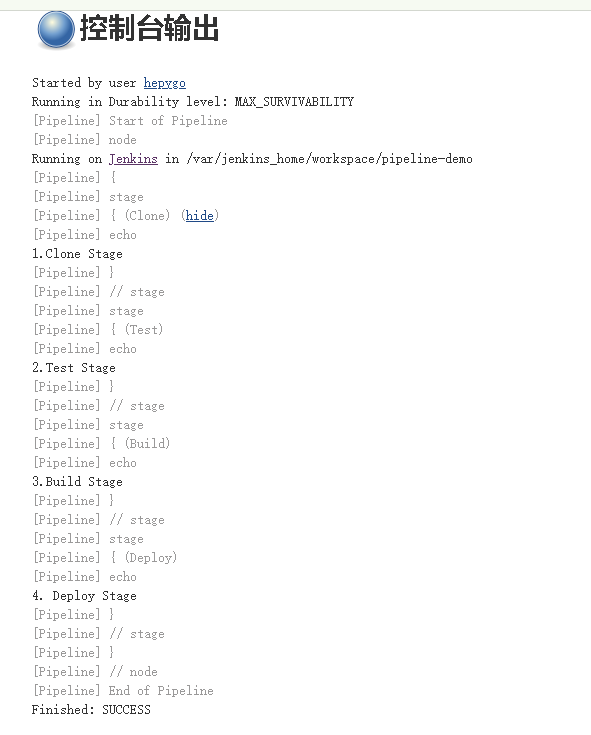
echo "4. Deploy Stage"

}

}

构建：点击左侧区域的 Build Now，可以看到 Job 开始构建了

隔一会儿，构建完成，可以点击左侧区域的 Console Output，我们就可以看到如下输出信息：



参考资料：

1.Jenkins 常用插件 -- slave

<http://blog.leanote.com/post/gua_l/jenkins-slave>

2.jenkins配置slave节点

<https://blog.csdn.net/testdeveloper/article/details/80144415>

3.Kubernetes与Jenkins的CI/CD

<https://blog.51cto.com/jacksoner/2340516>