**02-Jenkins基础入门**

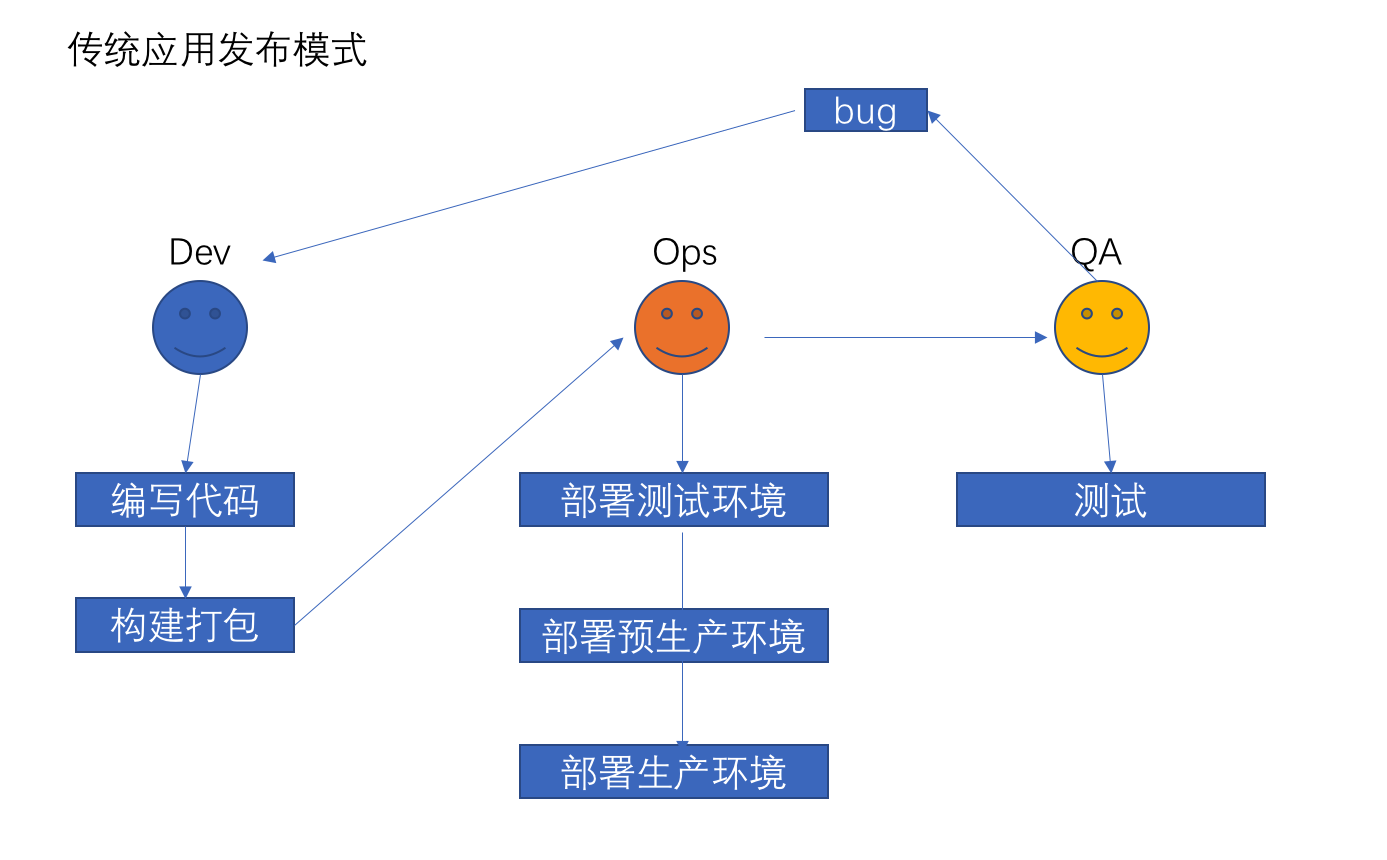
**课程安排**

|  |  |  |
| --- | --- | --- |
| **时间** | **标题** | **内容** |
| **下午** | **Jenkins基本使用** | 1. Jenkins简介与Jenkins的应用场景 2. Jenkins安装部署Server +Agent 3. Jenkins系统基本配置（数据目录、插件源） 4. Jenkins系统用户管理 5. Jenkins项目参数配置 |
| 1. Jenkins系统权限配置管理 2. Jenkins凭据管理与应用 3. Jenkins常用的项目管理 4. BlueOCean UI基本使用 |
|  |  |  |

**0. 传统应用发布模式**

一般一个开发团队中都会存在多个角色：**开发、测试、运维**。当时我们的应用发布模式可以能是这样的：

* **开发同学**进行项目代码开发，本地运行测试成功后将代码上传到**版本控制系统**。
* 开发同学通知**运维同学**项目可以发布了，然后运维同学**下载代码进行打包和构建，生成应用制品**。
* 运维同学使用部署脚本将生成的制品**部署到测试环境**，并提示测试同学可以进行产品的测试。
* 测试同学开始进行手动、自动化测试，测试完成后提醒运维同学可以进行**预生产环境部署**。
* 运维同学开始进行预生产环境部署，然后测试同学进行测试，测试完成后，开始**部署生产环境**。



手动操作很多、出现的问题很多。上面看似很流畅的过程，其实每次构建或发布都可能会出现问题。未对每次提交验证、构建环境不一致：开发人员本地测试成功后提交代码，运维同学下载代码进行编译却出现了错误。

**1.Jenkins简介与应用场景**

对于持续集成工具现在也有很多，目前**应用最多的是Jenkins**。Jenkins前身是Hudson，**使用java语言开发**的自动化发布工具。Jenkins是**跨平台**的可以在Win、Linux、MacOS部署。Jenkins是应用最广的**开源免费**的持续集成服务器，企业中普遍使用Jenkins来作为**项目集成发布工具**。Jenkins官方提供的**插件使Jenkins更为强大**。目前发展超过15年，比较成熟的CI工具（也可以CD）能够实现自动化集成发布。目前普遍使用Jenkins完成**持续集成CI/持续部署CD**等相关技术实践。 可以把Jenkins想象成一个脚本执行器，我们可以通过编写代码的方式灵活的控制Jenkins作业的运行与控制。

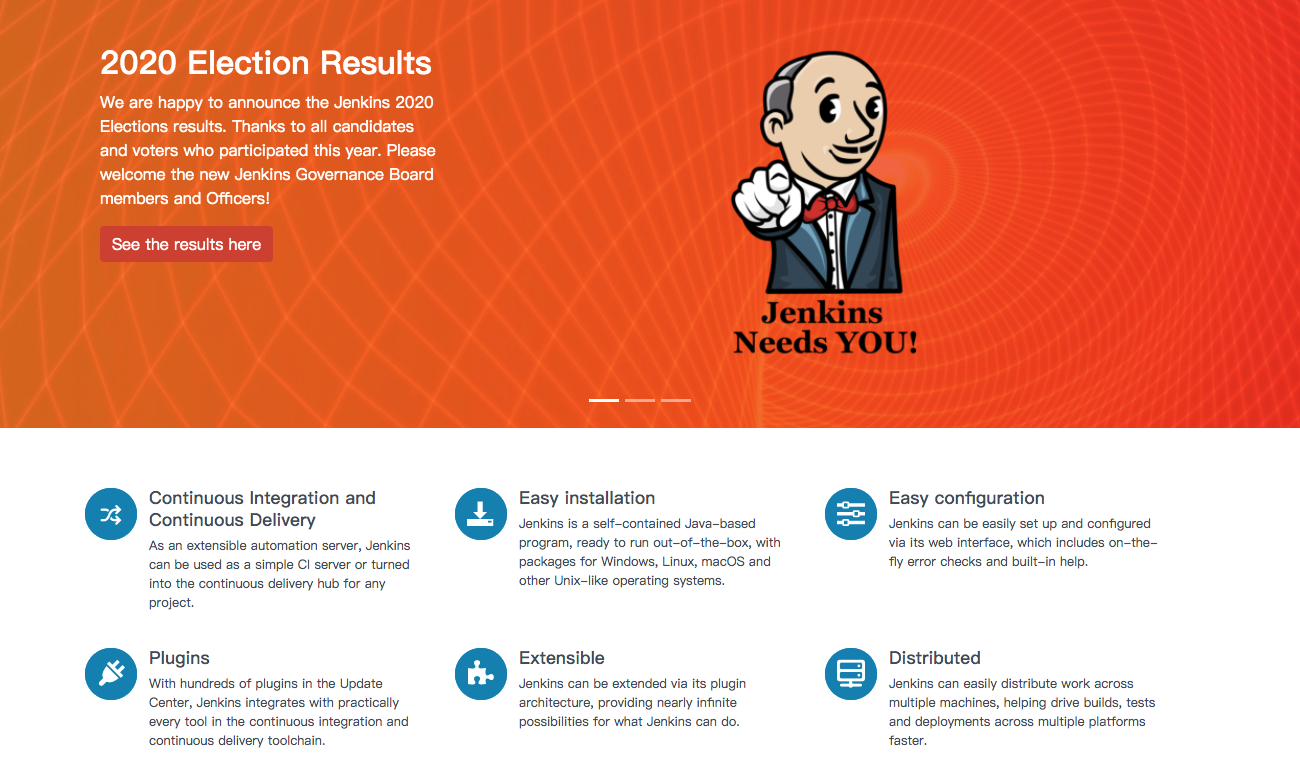
**1.1 应用场景**

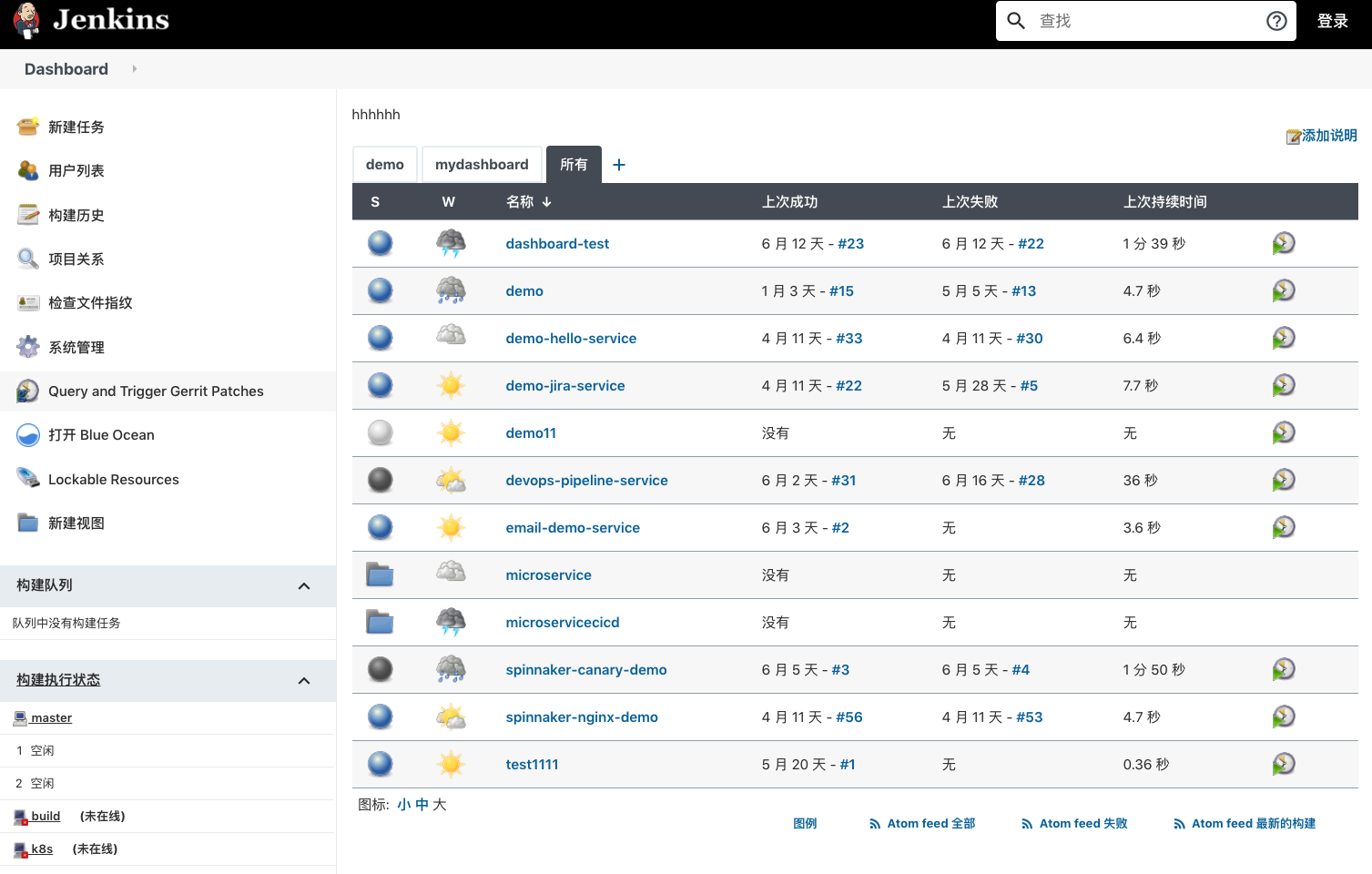
* 持续集成实践
  + 集成svn/git客户端实现源代码下载检出
  + 集成maven/ant/gradle/npm等构建工具实现源码编译打包单元测试
  + 集成sonarqube对源代码进行质量检查（坏味道、复杂度、新增bug等）
* 持续部署实践
  + 集成SaltStack/Ansible实现自动化部署发布
* 自动化测试实践
  + 集成Jmeter/Soar/Kubernetes/.....

**1.2 功能特点**

* 开源免费
* 多平台支持（windows/linux/macos）
* 主从分布式架构
* 提供web可视化配置管理页面
* 安装配置简单
* 插件资源丰富

官网： http://jenkins.io





**2.Jenkins安装部署**

关于Jenkins的安装可以使用rpm或者war包安装，需要提前安装好JDK8+环境。 Jenkins采用**分布式架构**，分为**server节点和agent节点**。server节点也是可以运行构建任务的，但我们一般使其主要来做**任务的调度**。（毕竟server节点挂了就都...）agent节点专门用于任务的执行。随着现在容器的盛行，我们可以将server节点和agent节点在**容器**或者基于**Kubernetes**中部署。关于agent节点借助容器可以实现**动态的资源分配**等等好处。agent节点可以分为**静态节点和动态节点。静态节点是固定的一台vm虚机或者容器。动态节点是随着任务的构建来自动创建agent节点。**

**2.1 安装JDK环境**

如果使用rpm的方式安装的Jenkins需要提前安装好JDK，Jenkins Slave运行也需要使用JDK。

jdk 下载链接：<https://mirrors.tuna.tsinghua.edu.cn/AdoptOpenJDK/8/jdk/x64/linux/OpenJDK8U-jdk_x64_linux_hotspot_8u282b08.tar.gz>

tar zxf OpenJDK8U-jdk\_x64\_linux\_hotspot\_8u282b08.tar.gz -C /usr/local

cd /usr/loca/jdk8u282-b08

pwd

vim /etc/profile

export JAVA\_HOME=/usr/local/jdk8u282-b08

export PATH=$JAVA\_HOME/bin:$PATH

source /etc/profile

java -version

openjdk version "1.8.0\_282"

OpenJDK Runtime Environment (AdoptOpenJDK)(build 1.8.0\_282-b08)

OpenJDK 64-Bit Server VM (AdoptOpenJDK)(build 25.282-b08, mixed mode)

**2.2 安装Jenkins Master**

redhat/centos: [https://mirrors.tuna.tsinghua.edu.cn/jenkins/redhat-stable/](https://mirrors.tuna.tsinghua.edu.cn/jenkins/redhat-stable/jenkins-2.277.1-1.1.noarch.rpm)

**2.2.1 rpm 安装**

wget https://mirrors.tuna.tsinghua.edu.cn/jenkins/redhat-stable/jenkins-2.277.1-1.1.noarch.rpm

rpm -ivh jenkins-2.277.1-1.1.noarch.rpm

/etc/sysconfig/jenkins

## service start

[root@zeyang-nuc-service jenkins]# netstat -anlpt | grep 8080

tcp6 0 0 :::8080 :::\* LISTEN 16016/java

[root@zeyang-nuc-service jenkins]# service jenkins status

systemctl stop firewalld

## 访问

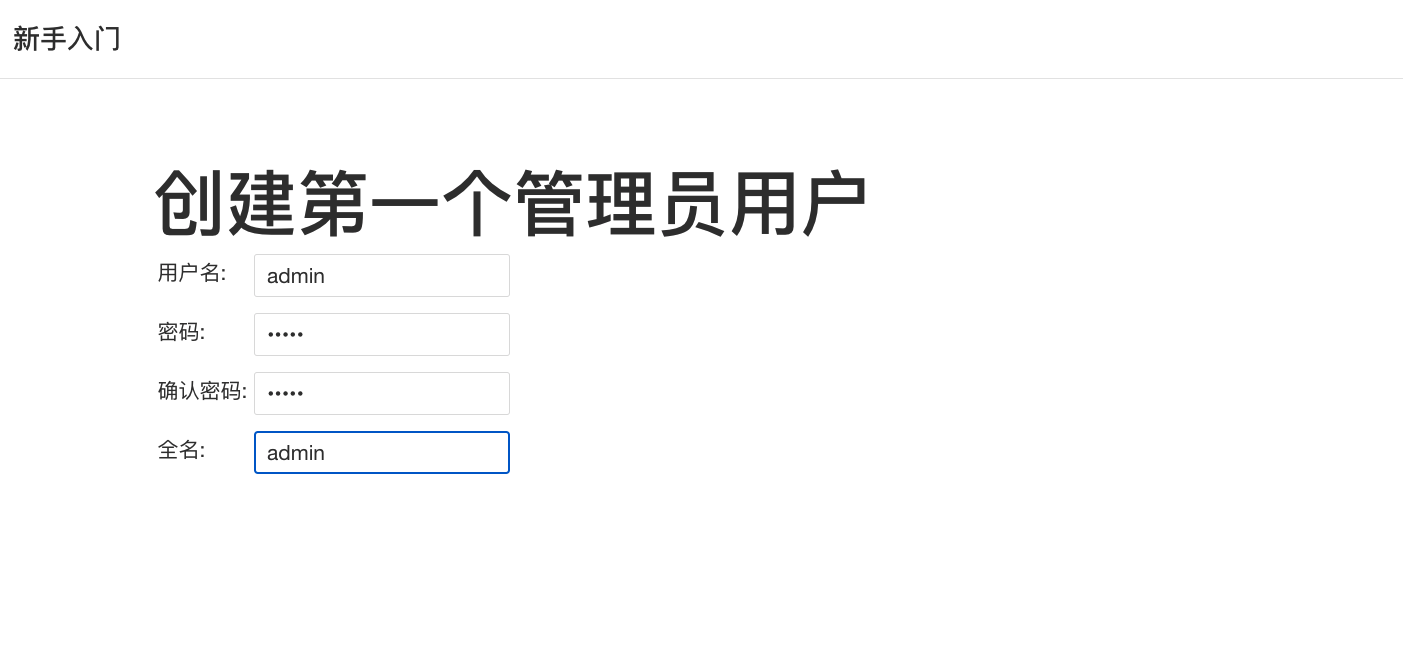
http://192.168.1.200:8080/



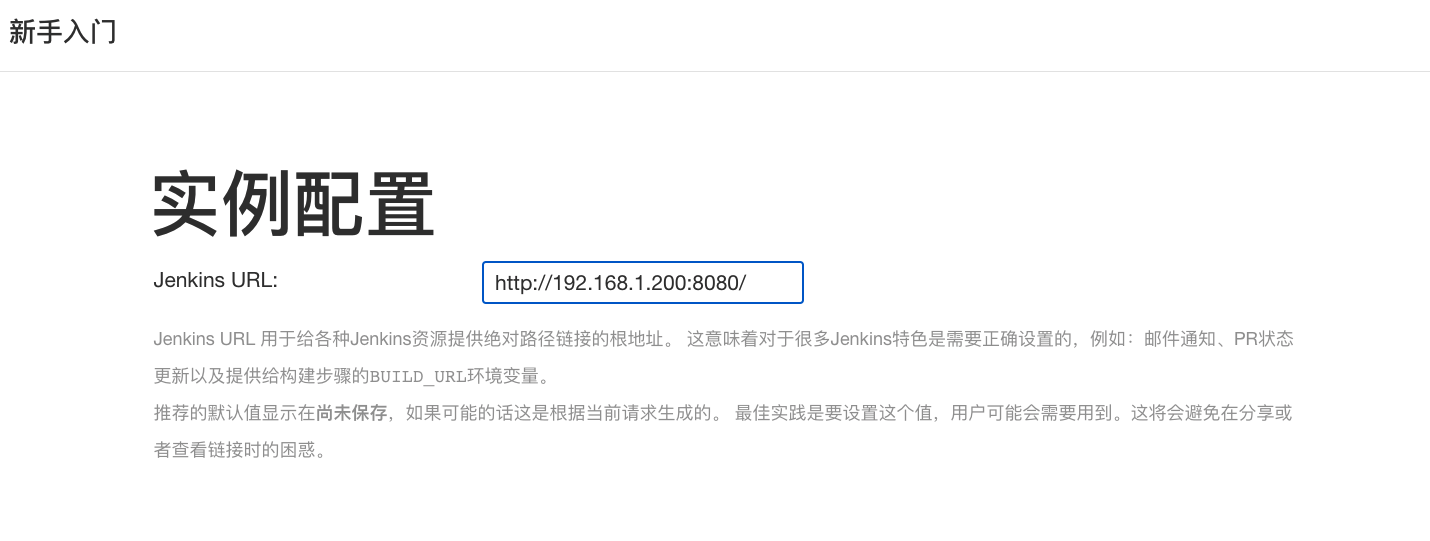
* 跳过默认的插件安装，仅选择中文社区插件。



* 创建admin用户



* 配置站点url



**2.2.2 docker安装**

docker pull jenkins/jenkins:2.289.1-lts-centos7

mkdir -p /data/cicd2/jenkins

chmod +x /data/cicd2/jenkins

## 指定时区

#System.setProperty('org.apache.commons.jelly.tags.fmt.timeZone', 'Asia/Shanghai')

docker run -itd --name jenkins \

-p 8080:8080 \

-p 50000:50000 \

-e JAVA\_OPTS="-Dorg.apache.commons.jelly.tags.fmt.timeZone='Asia/Shanghai'" \

--privileged=true \

-v /data/cicd2/jenkins:/var/jenkins\_home docker.io/jenkins/jenkins:2.289.1-lts-centos7

##FAQ

[root@zeyang-nuc-service ~]# docker logs -f jenkins

touch: cannot touch '/var/jenkins\_home/copy\_reference\_file.log': Permission denied

Can not write to /var/jenkins\_home/copy\_reference\_file.log. Wrong volume permissions?

chmod 777 -R jenkins/

* 默认端口8080
* 配置JENKINS\_HOME /var/lib/jenkins
* 配置启动用户Jenkins
* 配置插件更新源（Jenkins 中文社区源）

**2.2.3 配置插件源**

进入Jenkins首页，然后最下面（点击 Jenkins中文社区）



然后跳转到这个页面，按照图中的操作。 （2，3，4）



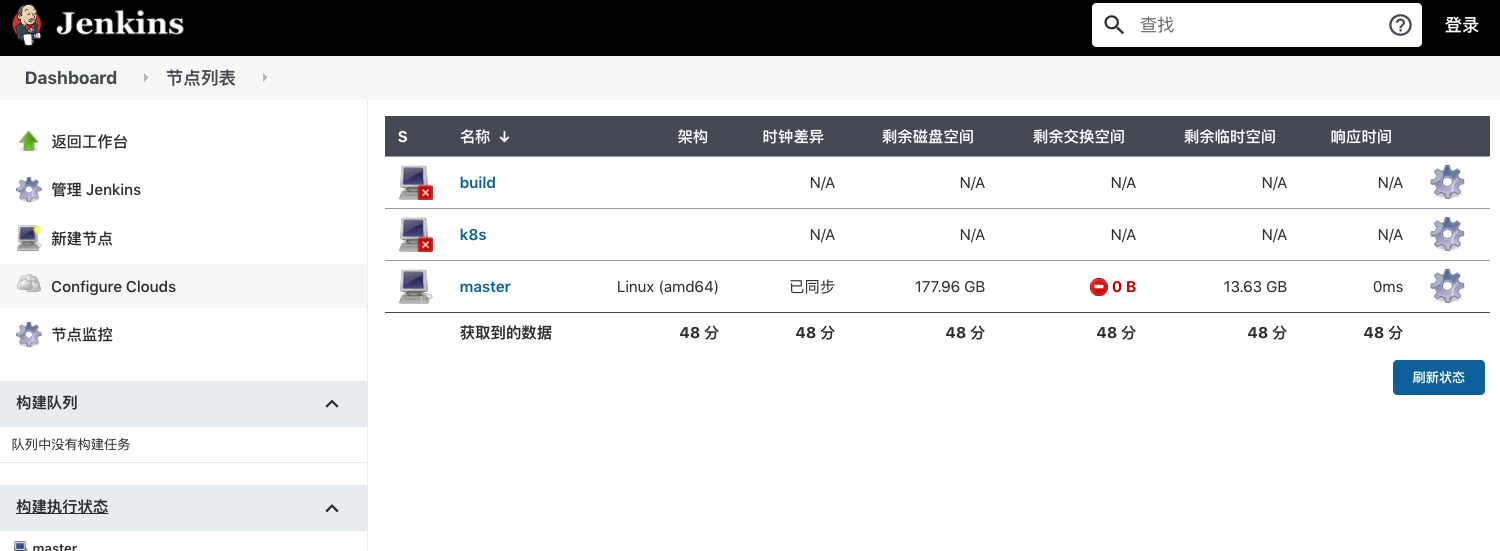
粘贴上个步骤2的复制内容。（<https://updates.jenkins-zh.cn/update-center.json>）

点击立即获取。



**2.3 安装Jenkins Slave**

系统设置 > 节点管理 > 新建节点



* 填写节点信息



* 下载agent.jar



## 创建agent目录

mkdir -p /data/cicd/jenkinsagent

cd /data/cicd/jenkinsagent

wget http://192.168.1.200:8080/jnlpJars/agent.jar

## 编写启动脚本

#!/bin/bash

echo e199ecad9b95bc93277d74469256a7b4b3aa7c9d2c67916e93cdd0f93c9f9a5e > secret-file

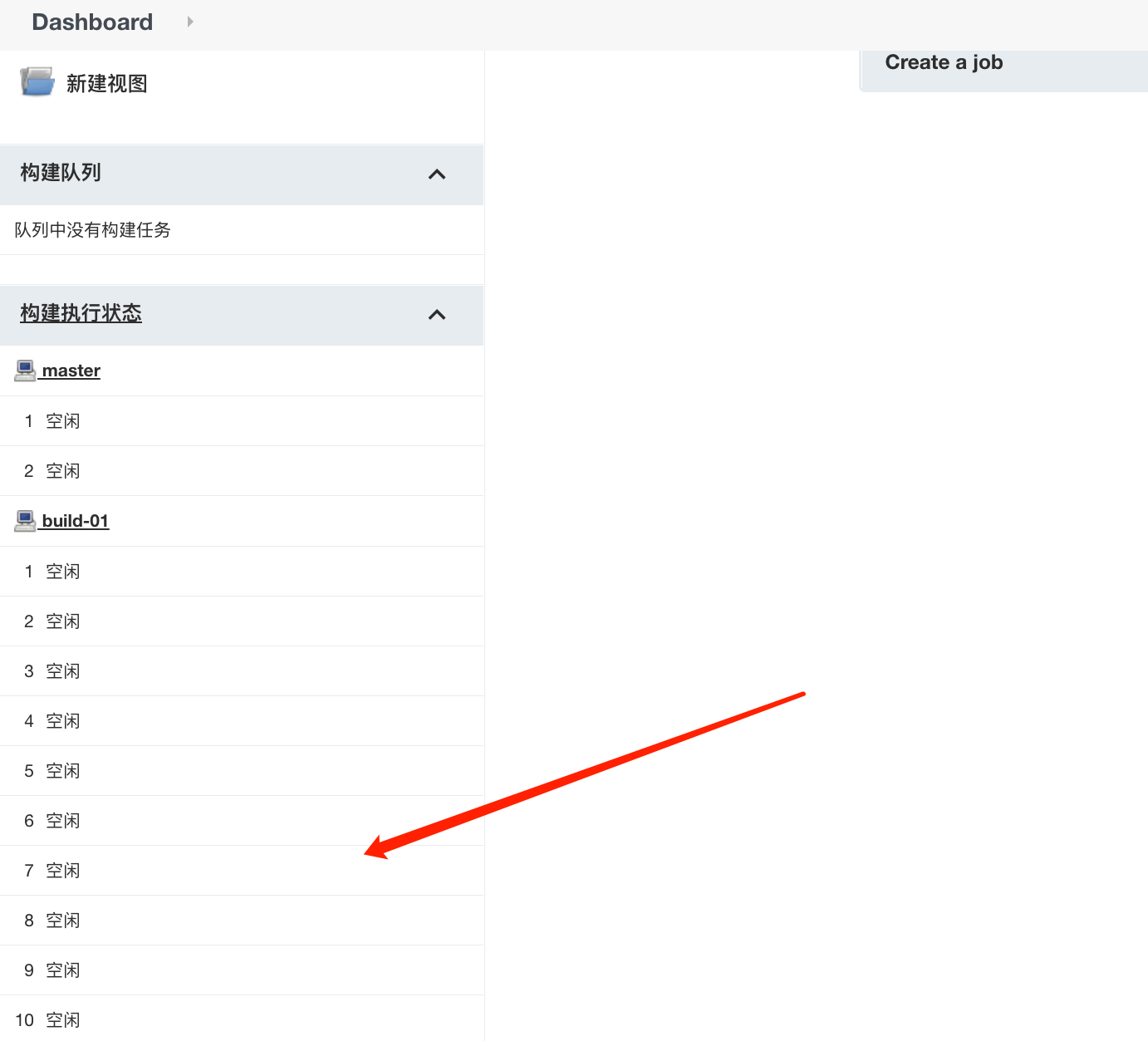
nohup java -jar agent.jar -jnlpUrl http://192.168.1.200:8080/computer/build-01/jenkins-agent.jnlp -secret @secret-file -workDir "/data/cicd/jenkinsagent" >/dev/null &

## 启动agent

sh -x start.sh

ps aux | grep agent

* 效果



自定义agent的通信端口，[Configure Global Security](http://192.168.1.200:8080/configureSecurity/)。



**3. Jenkins的数据目录**

Jenkins采用文件存储， 其配置相关信息都保存到了XML文件中。

[root@zeyang-nuc-service jenkins]# ls -d \*/

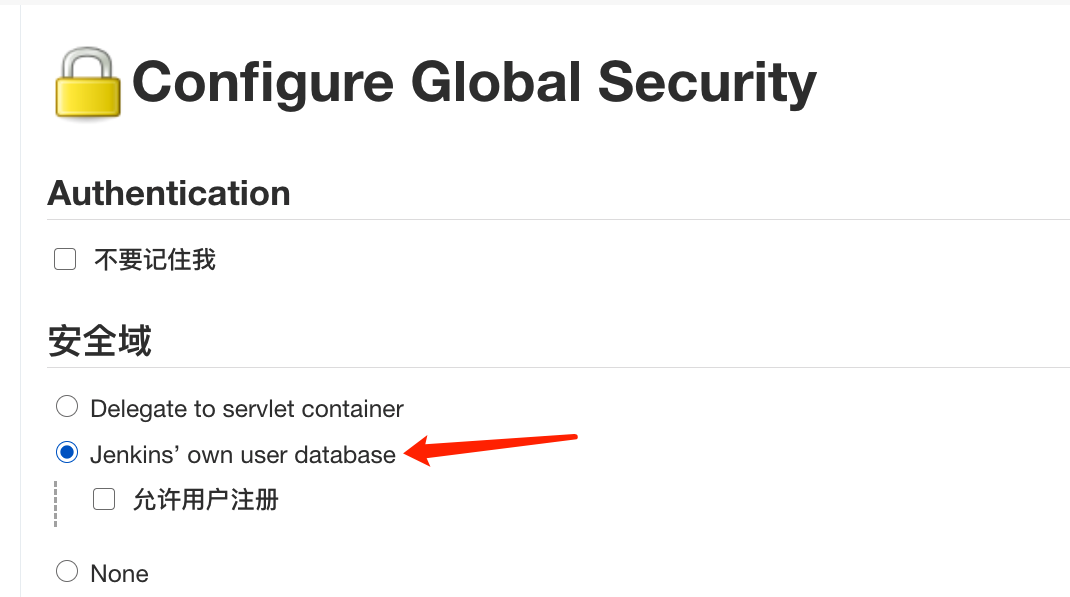
caches/ jobs/ nodes/ secrets/ userContent/ war/ workspace/

fingerprints/ logs/ plugins/ updates/ users/

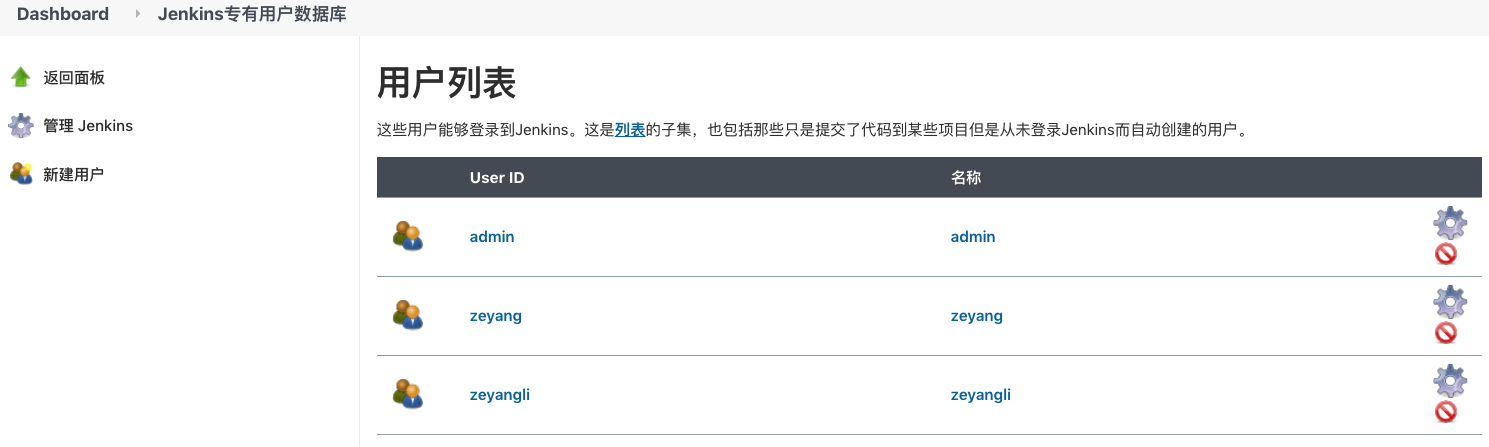
* caches: 系统缓存数据
* jobs： Jenkins项目作业
* nodes： Jenkins slave节点信息
* secrets： 秘钥信息
* userContent： 类似于web站点目录，可以上传一些文件
* workspace： 默认的工作目录
* fingerprints： 指纹验证信息
* logs ： 日志信息
* plugins： 插件相关配置
* updates： 插件更新目录
* users： jenkins系统用户目录

**4. Jenkins用户管理**

Jenkins默认使用的是**自带的数据库**管理用户，支持同步**LDAP、Github**等认证集成。



***系统设置 > 管理用户***

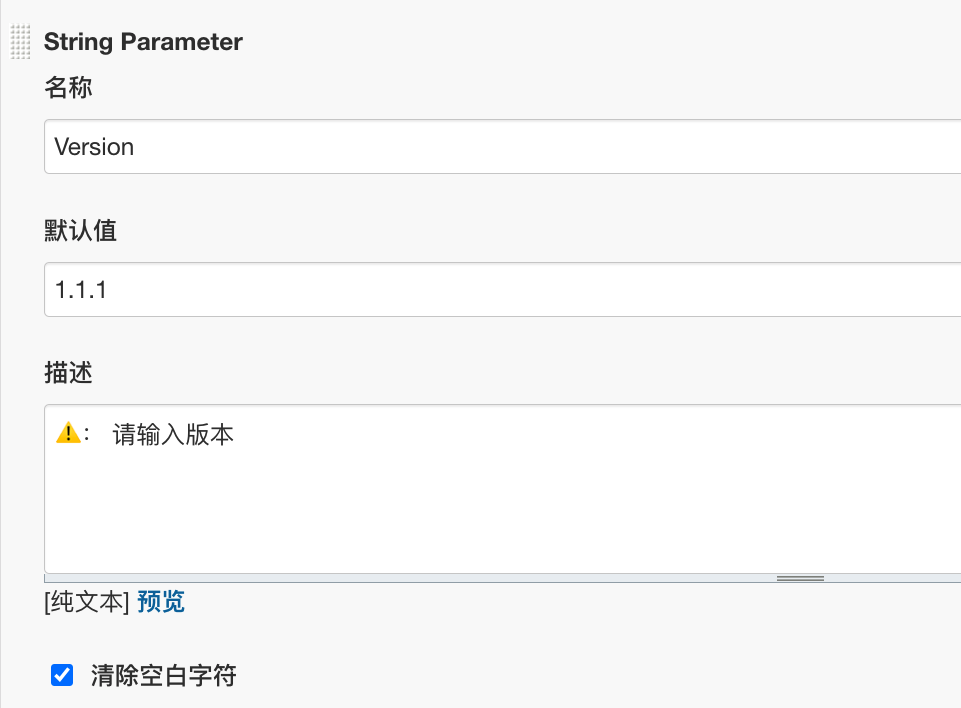


**5. 项目类型与参数**

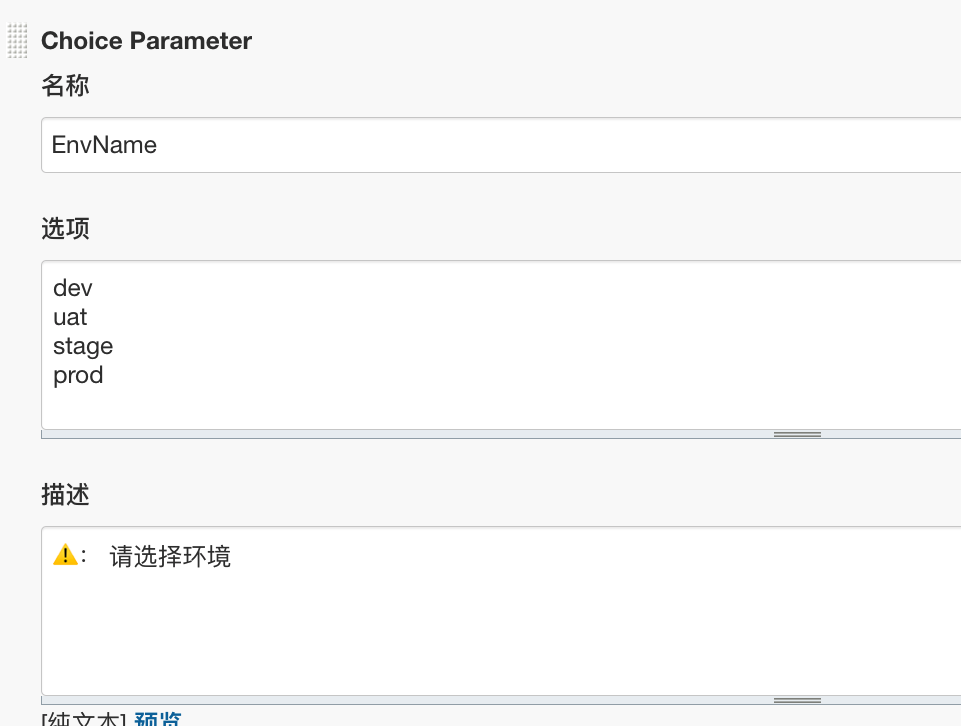


**参数化构建：**

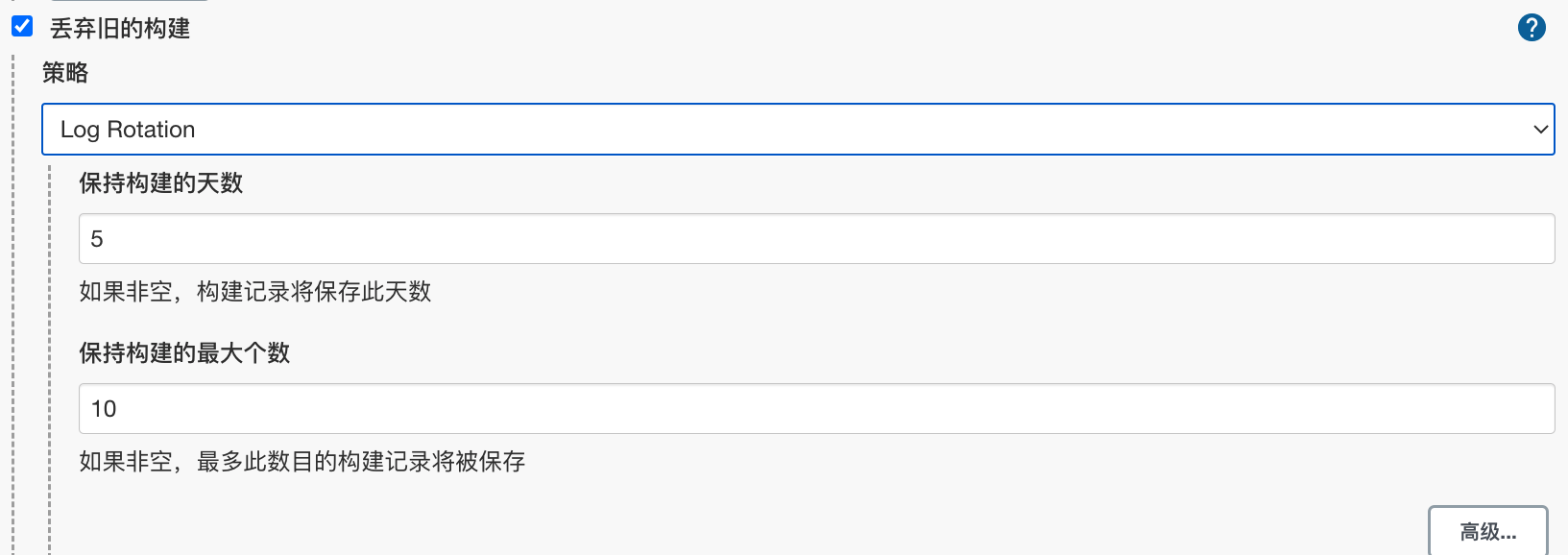
* 字符参数



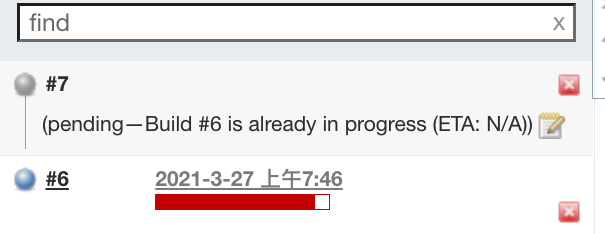
* 选项参数



**丢弃历史构建**



**并行构建**



microservicecicd@2

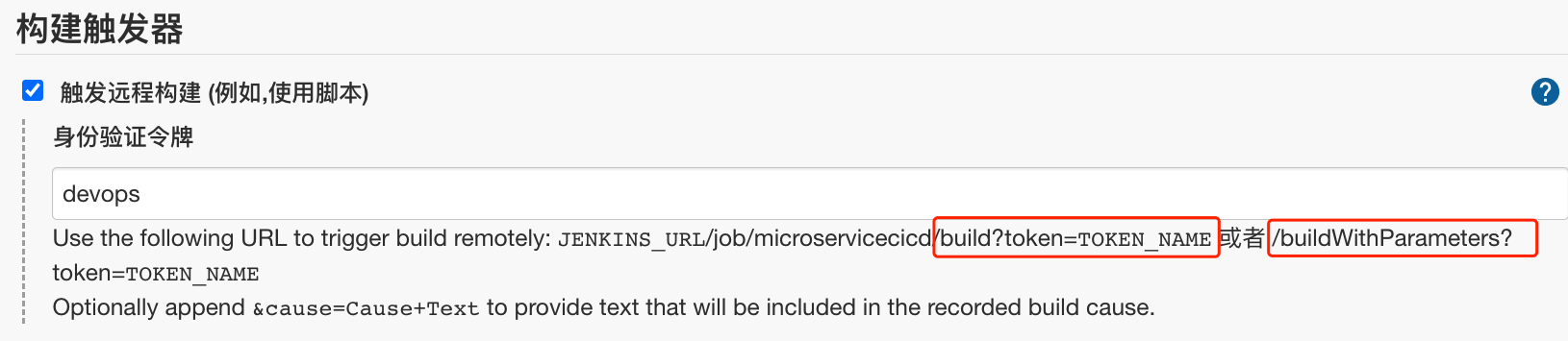
1.一个job 对应一个 microservicecicd

2.microservicecicd@2

3.microservicecicd@3

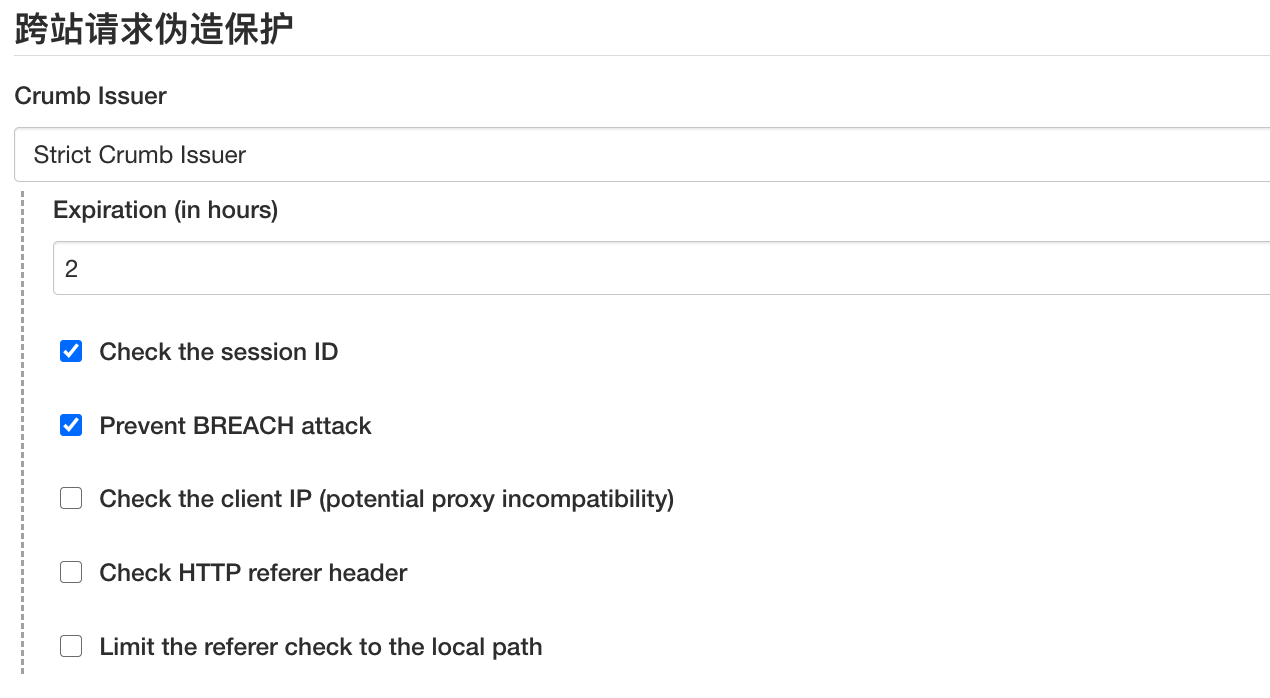
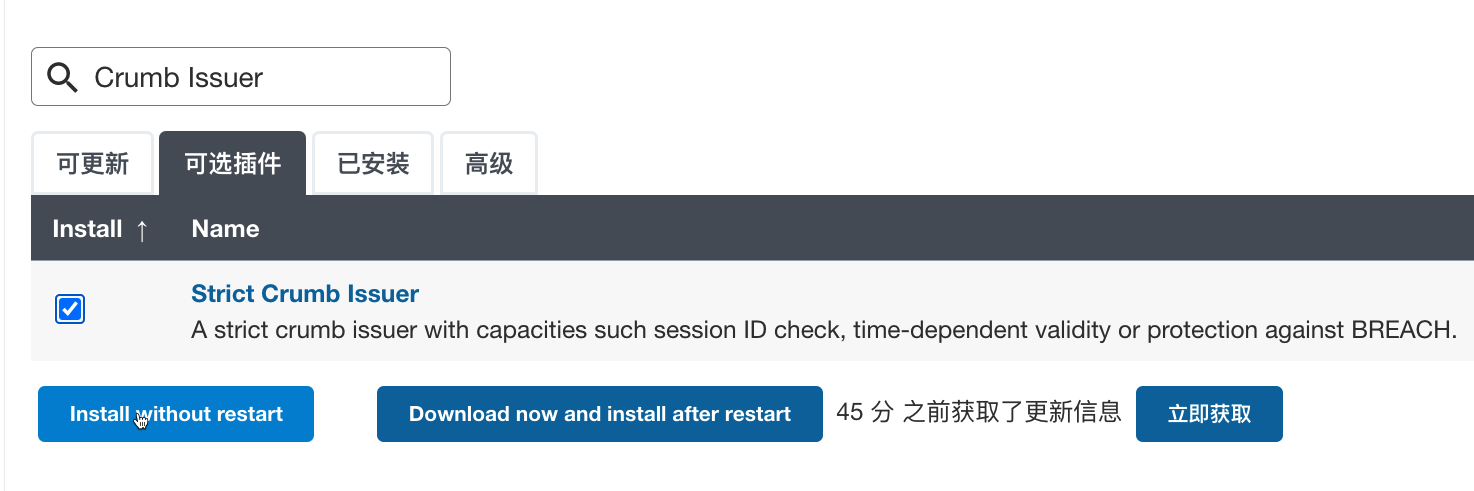
## 解决方法： 随机生成一个不同的workspace。

**构建触发器**



FAQ: 403 被拦截...

<https://plugins.jenkins.io/strict-crumb-issuer/>



## 未选择参数化构建

curl -u admin:admin http://192.168.1.200:8080/job/test1/build?token=devops

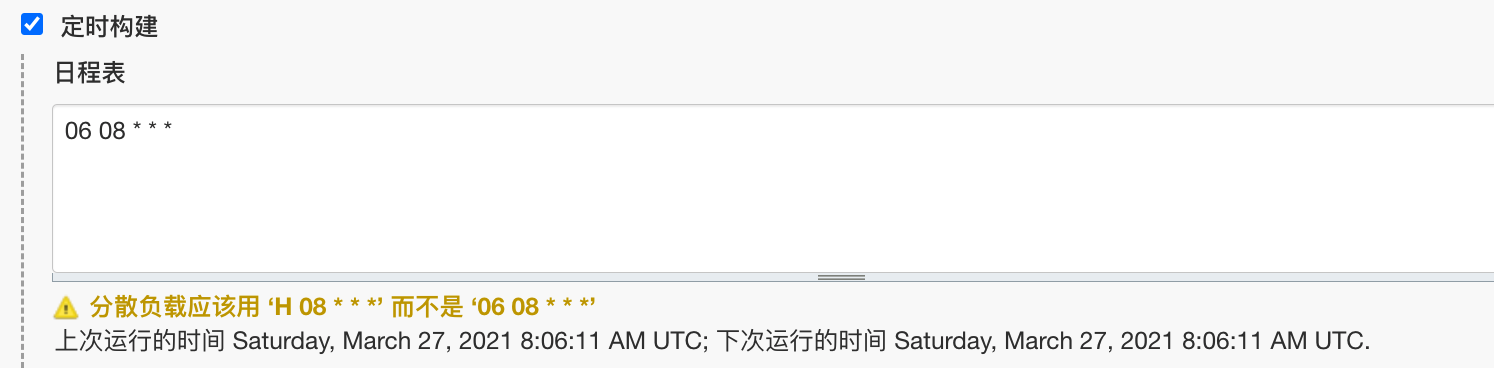
## 选择了参数化构建就不能使用上面的请求操作了。

curl -u admin:admin "http://192.168.1.200:8080/job/microservicecicd/buildWithParameters?token=devops&Version=1.1.54&EnvName=dev"

**上下游构建依赖**



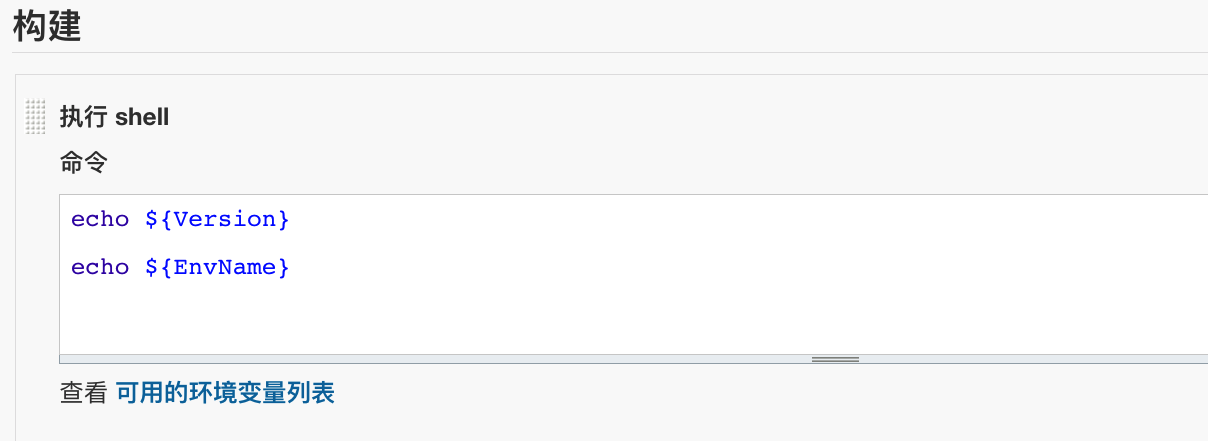
**定时构建**



**轮询版本控制系统**



**构建步骤**



**构建后配置**

当前运行的shell创建的目录，是在当前job所运行的节点的workspace下面。



* 效果



**视图配置：**



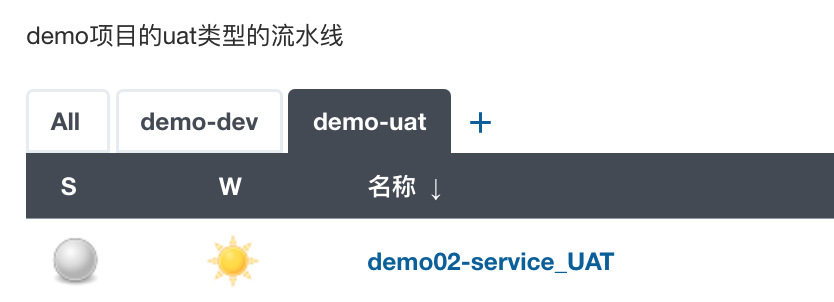
## 通过demo-uat视图访问URL

http://192.168.1.200:8080/view/demo-uat/job/demo02-service\_UAT/

## 通过all视图访问URL

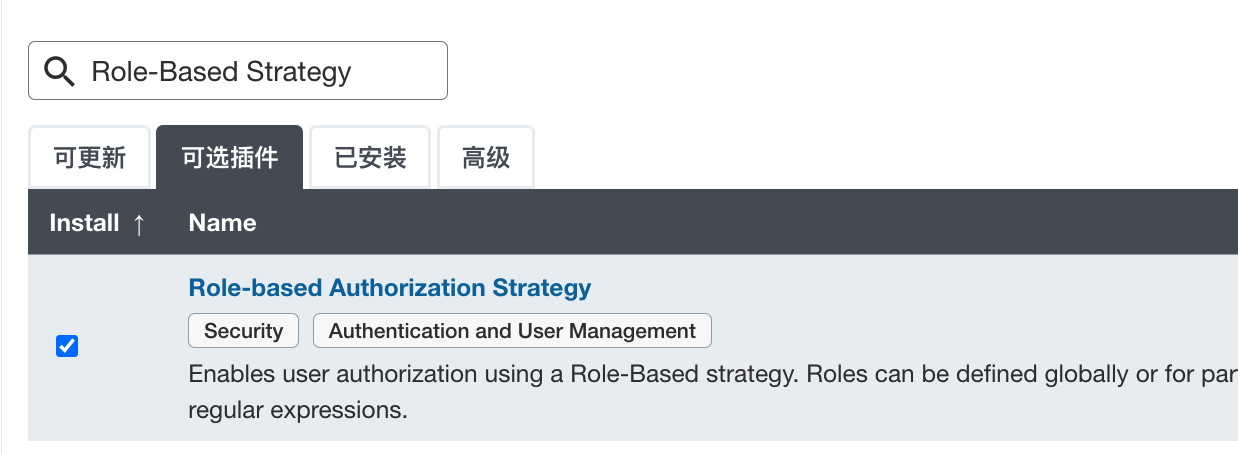
http://192.168.1.200:8080/job/demo02-service\_UAT/

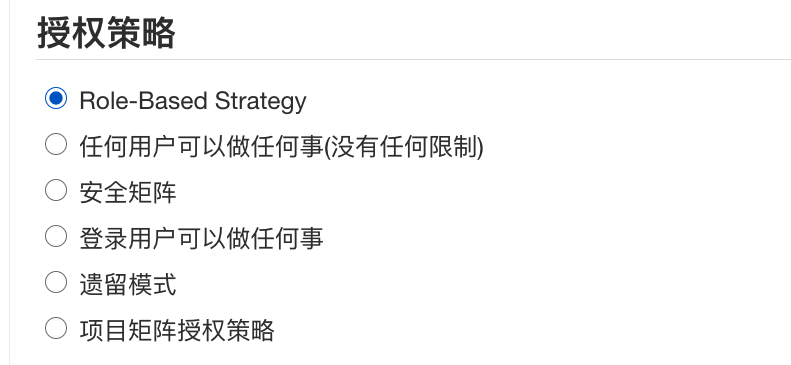
* 效果

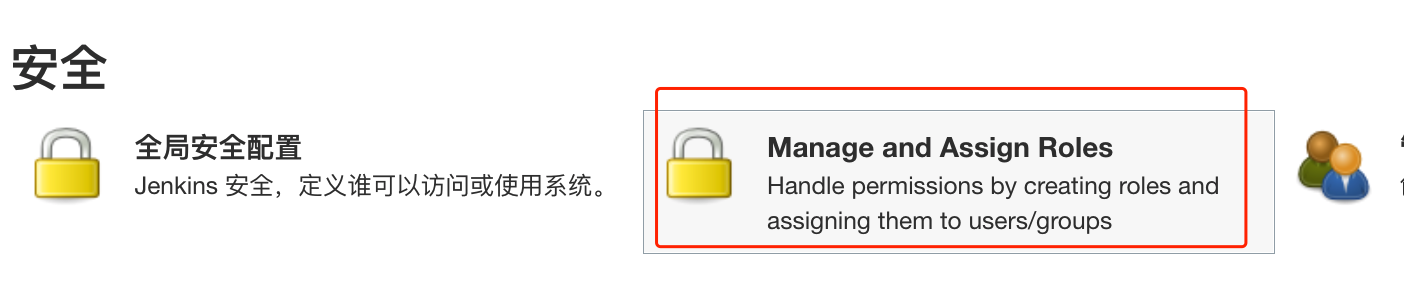


**6. Jenkins系统权限管理**

注意安装插件: 权限默认使用 Role-Based Strategy。

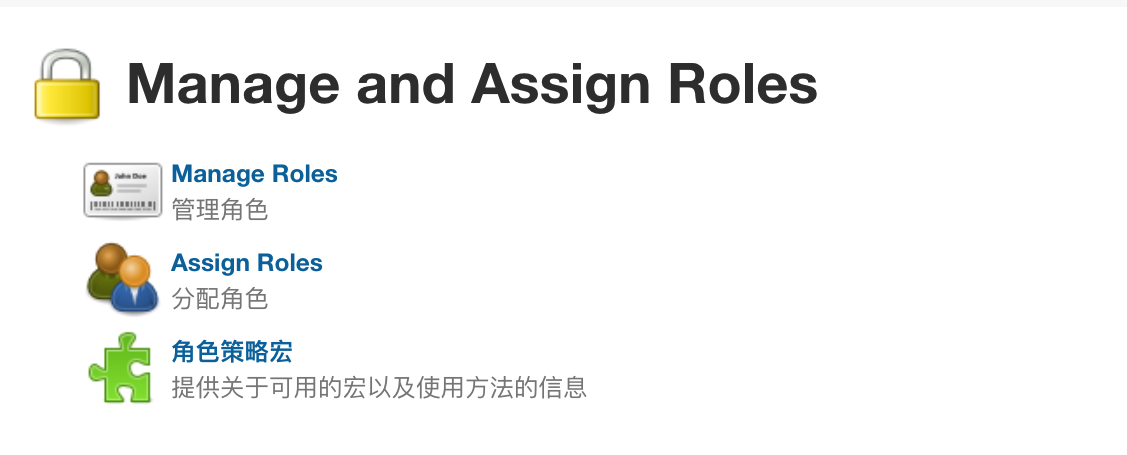




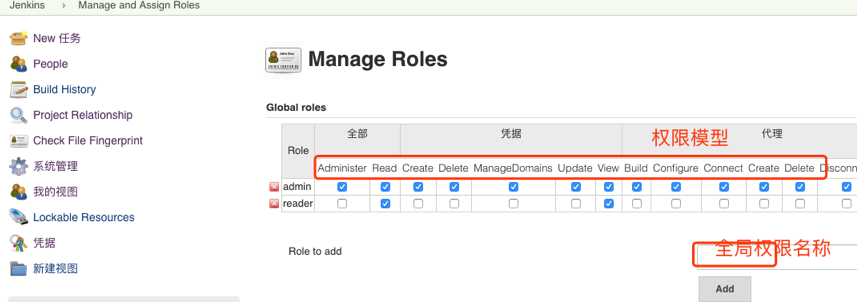


**6.1 管理权限**

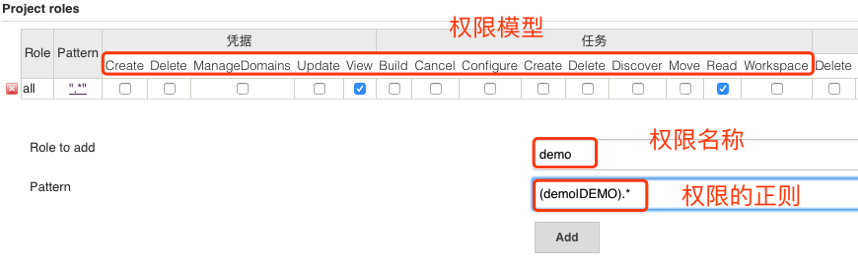
* 系统设置->Manage and Assign Roles->Manage Roles
* Global roles：添加用户和用户组，分配某个用户属于哪个用户组。
* Item roles：添加某个工程或某一组工程的用户或用户组
* Node roles：添加某个节点或某一组节点的用户或用户组



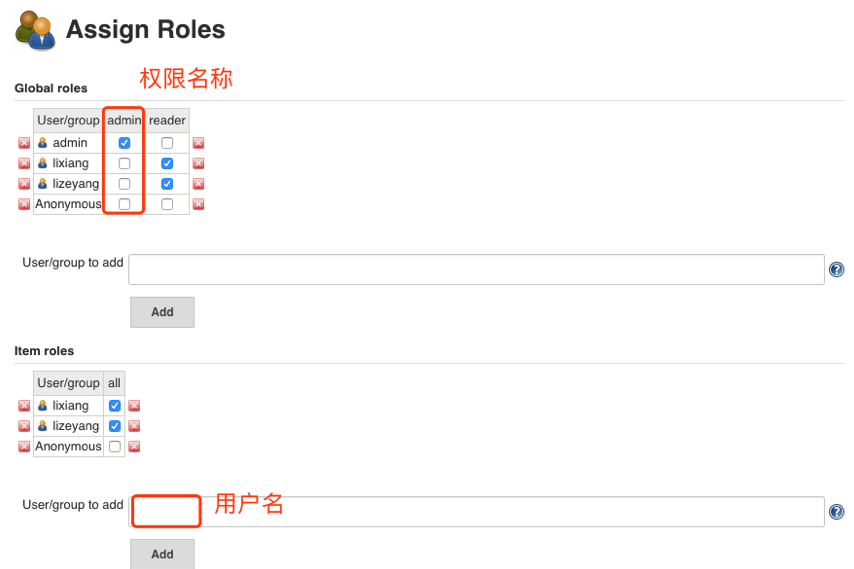
**6.2 全局权限**



**6.3 项目权限**

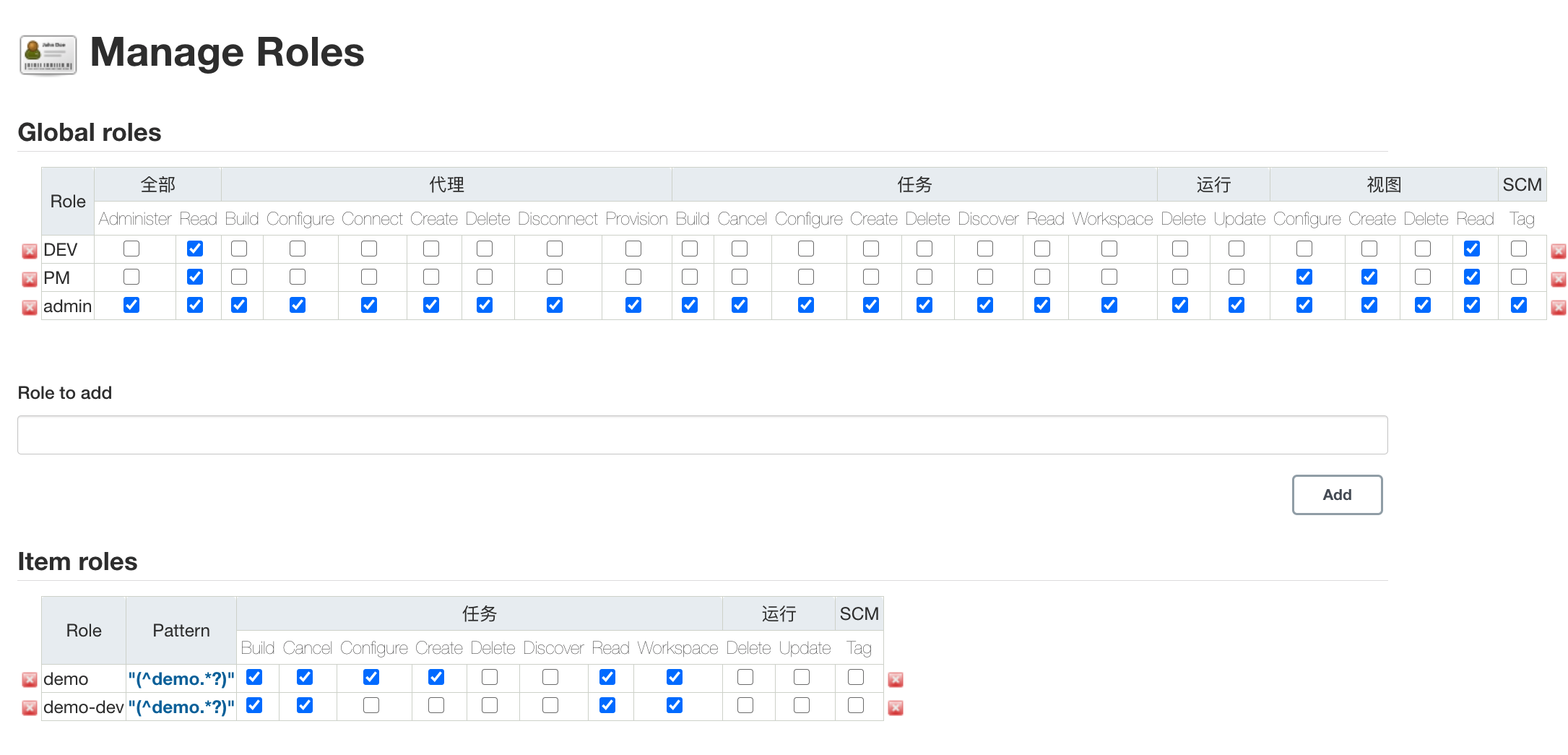


**6.4 项目授权**

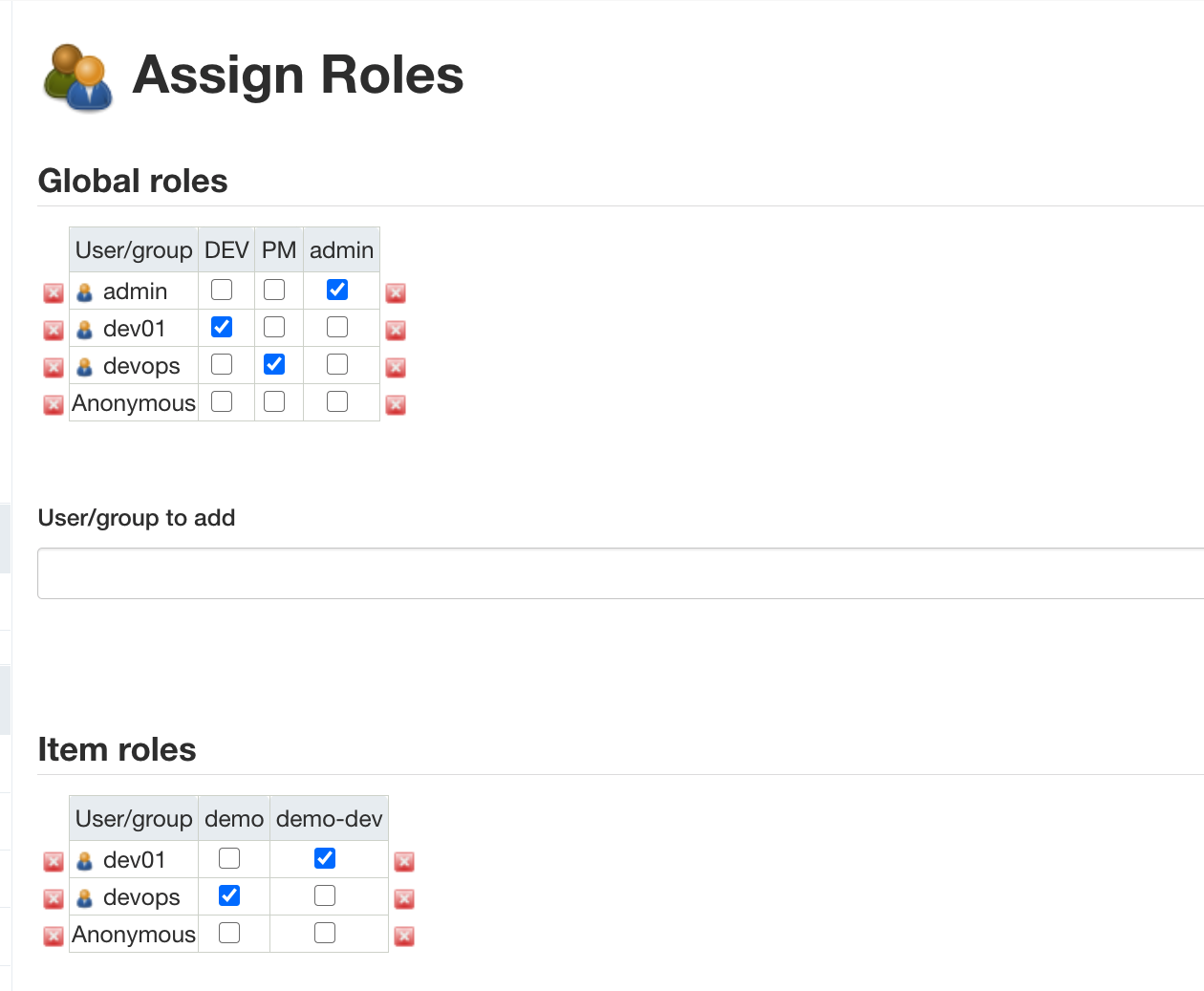


**6.5 实践**

**权限配置**



**权限授权**



**效果**

* **PM角色： 能够操作和管理所属的业务和项目**

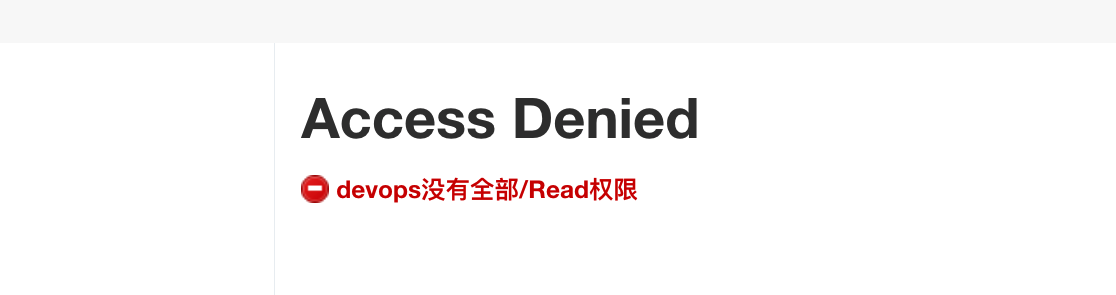


* **DEV角色：只允许对所属的项目进行构建和取消查看操作**



**FAQ:**

如果未赋予用户Global Role ”read“ 会报错。

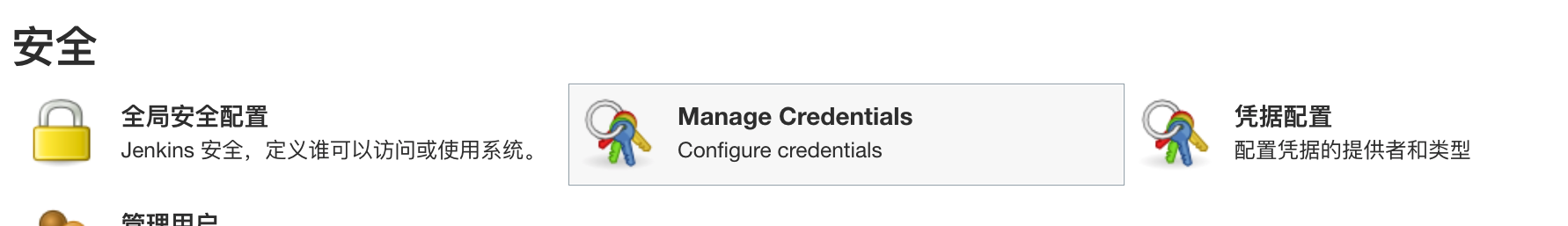


**先备份**

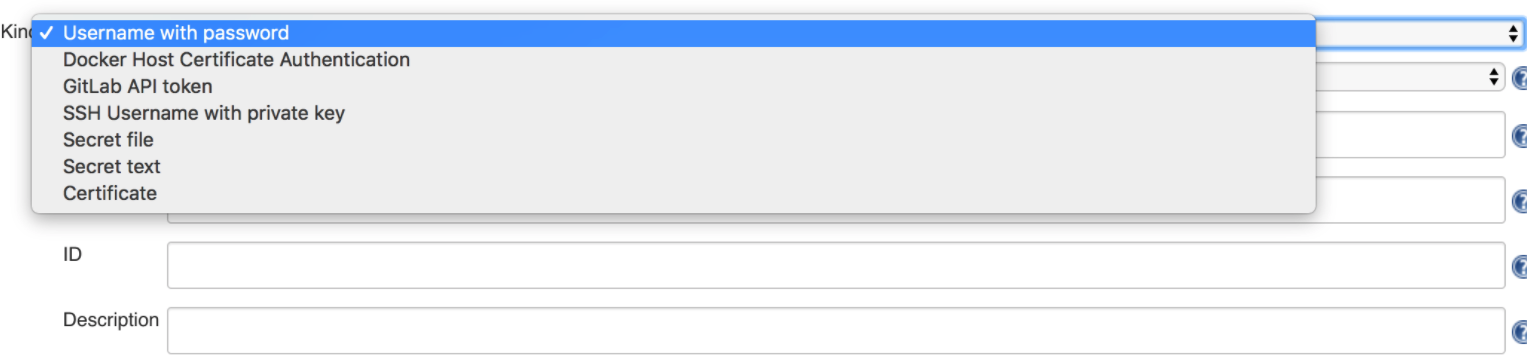
**7. Jenkins系统凭据管理**

Jenkins的凭据有很多种类型，可以将一些敏感信息存储为凭据，然后在项目中引用。 凭据-> 系统-> 全局凭据。

**安装插件 Credentials**







* Username with password类型存储Harbor或者其他系统的用户名和密码。
* GitLab API token类型存储Gitlab的用户API token。
* Secret text类型可以用来存储OpenShift等系统中的token。
* Certificate类型可以用户存储证书，例如k8s的用户证书。

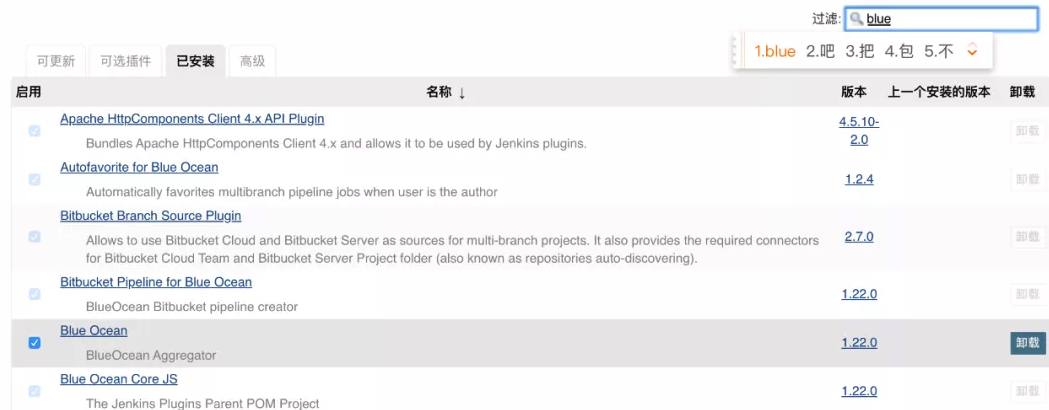
**8. BlueOCean UI基本使用**

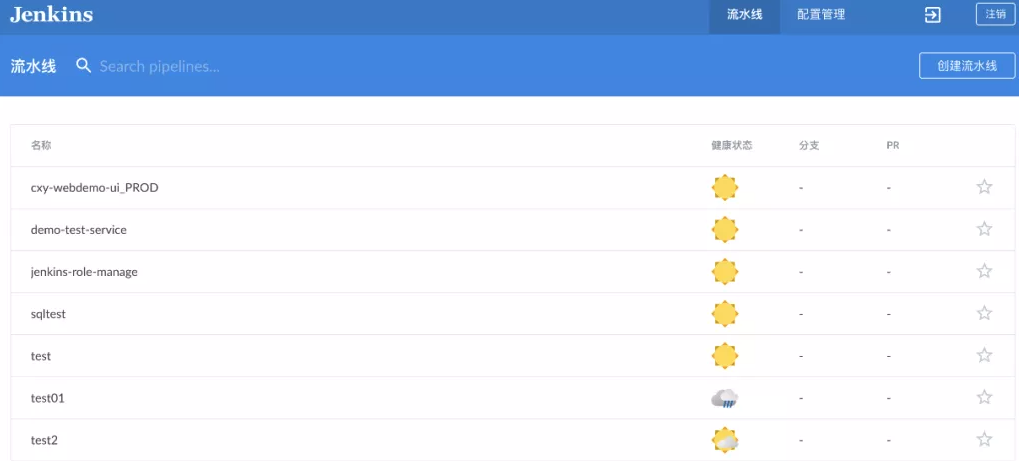
Blue Ocean减少了混乱而且进一步明确了团队中每个成员 Blue Ocean 的主要特性包括：

* 持续交付(CD)Pipeline的 **复杂可视化** ，可以让您快速直观地理解管道状态。
* **Pipeline 编辑器** - 引导用户通过直观的、可视化的过程来创建Pipeline，从而使Pipeline的创建变得平易近人。
* **个性化** 以适应团队中每个成员不同角色的需求。
* 在需要干预和/或出现问题时 **精确定位** 。Blue Ocean 展示 Pipeline中需要关注的地方， 简化异常处理，提高生产力
* **本地集成分支和合并请求**, 在与GitHub 和 Bitbucket中的其他人协作编码时实现最大程度的开发人员生产力。

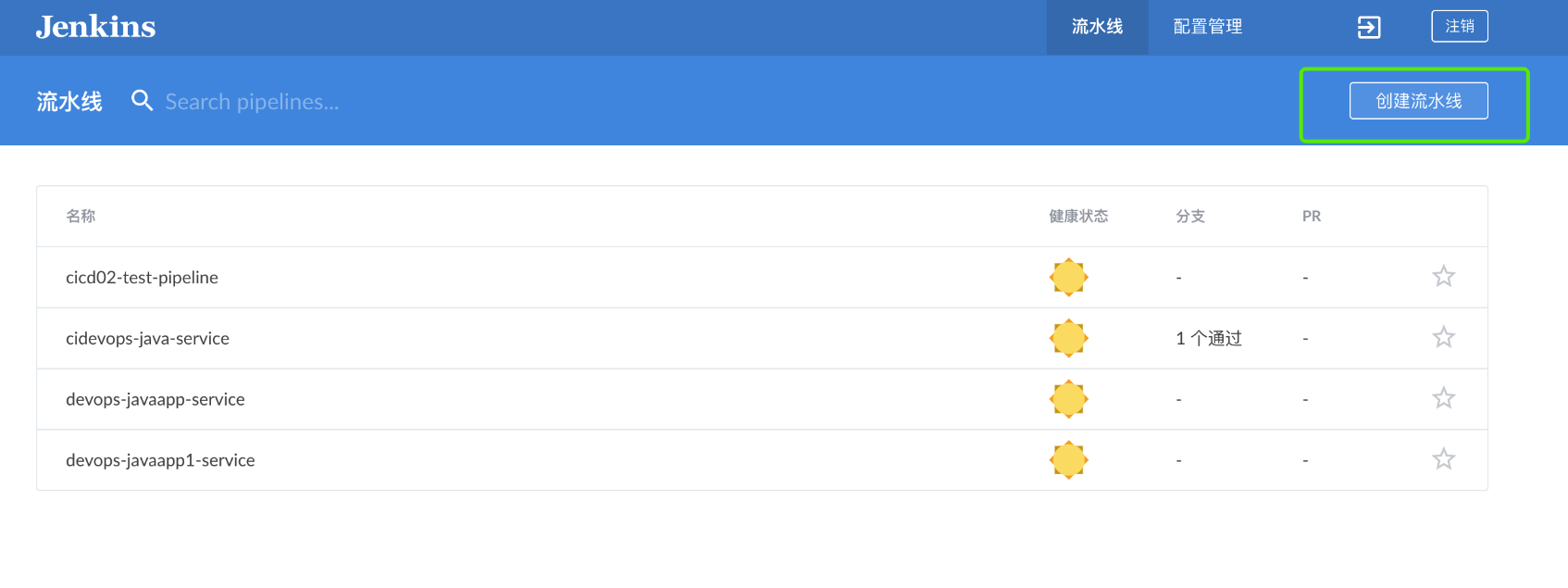
官方文档：<https://jenkins.io/zh/doc/book/blueocean/getting-started/>

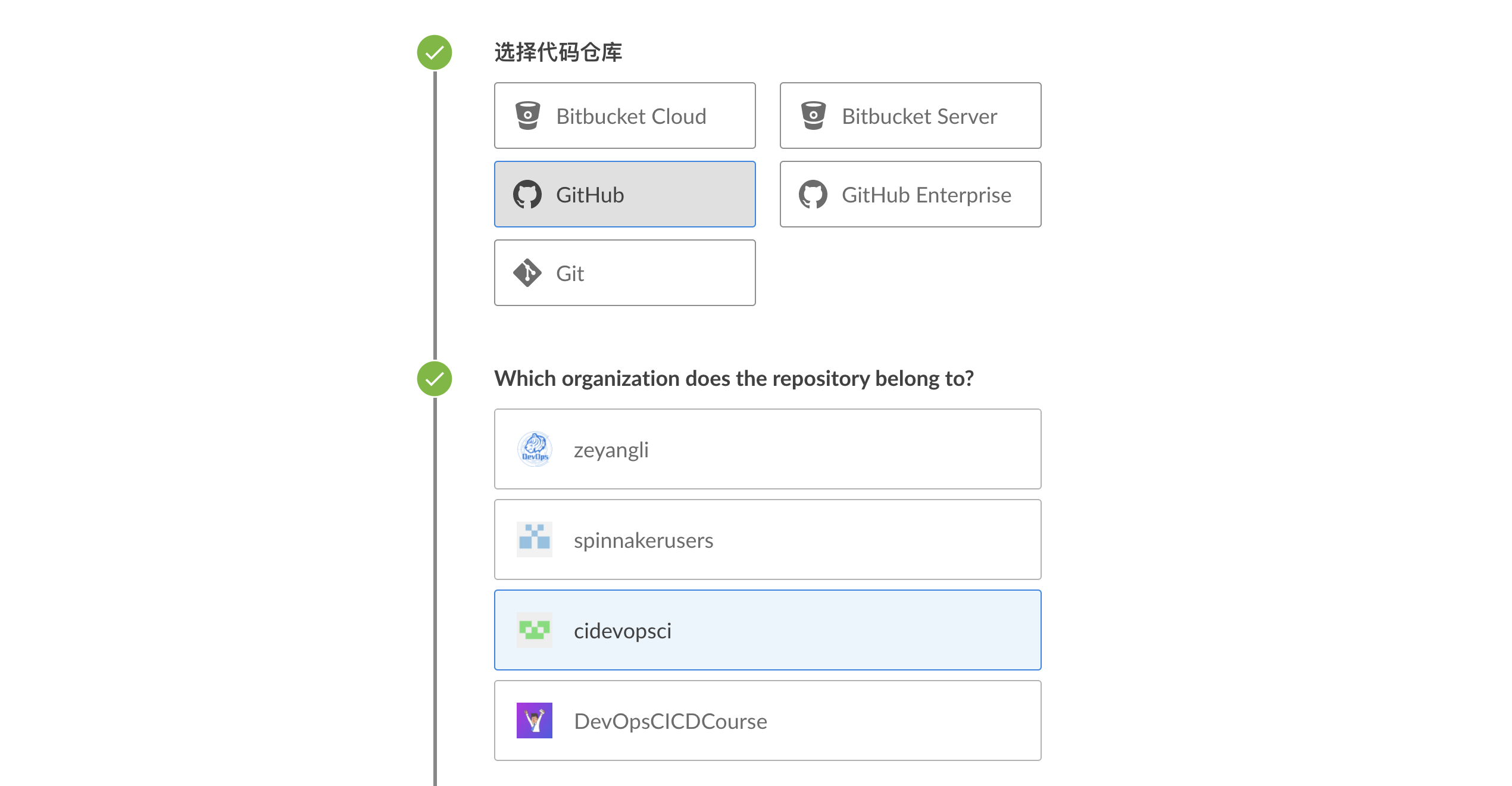
进入Jenkins插件管理然后安装blueocean插件。

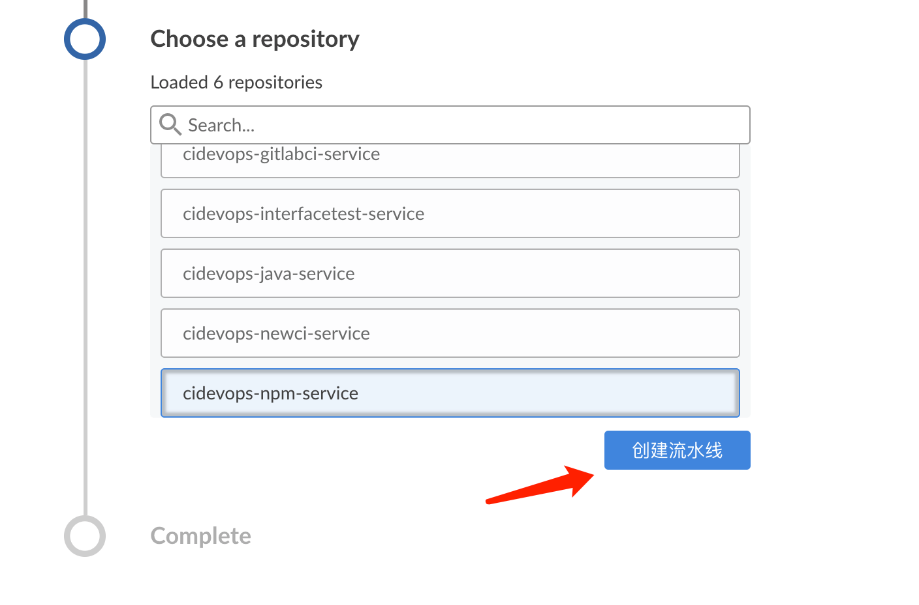




**使用BlueOcean可视化编辑Pipeline生成Jenkinsfile**







**9. Jenkins学习的一些建议**

只要熟练的掌握**Jenkinsfile**的编写，Jenkins什么事情都是可以做。限制Jenkins功能的不是Jenkins本身。你可以将Jenkins想象成一个可以配置的web前端页面，而后面的Jenkinsfile就是我们要写的后端代码。

谈谈我们学习Jenkins之前要具备哪些基础知识呢？

**基础要求**：

* 熟练Linux系统运维（常用的命令等等）
* 熟悉Shell脚本编程（Jenkins pipeline会调用很多shell语句）

**扩展知识**：

* 掌握一门编程语言例如Python（Jenkins会用到groovy语法）
* 掌握docker、kubernetes基础知识

是不是看上去很多呢？可以让你的流水线更加强大与灵活。其实这些都是我们前往**技术型**DevOps工程师所必备的知识点了。如果你是开发人员你需要补充运维相关的知识，如果你是运维人员需要补开发知识。**Jenkins是开发、运维同学到DevOps的转折点。**