代码自动发布-docker

一、分析

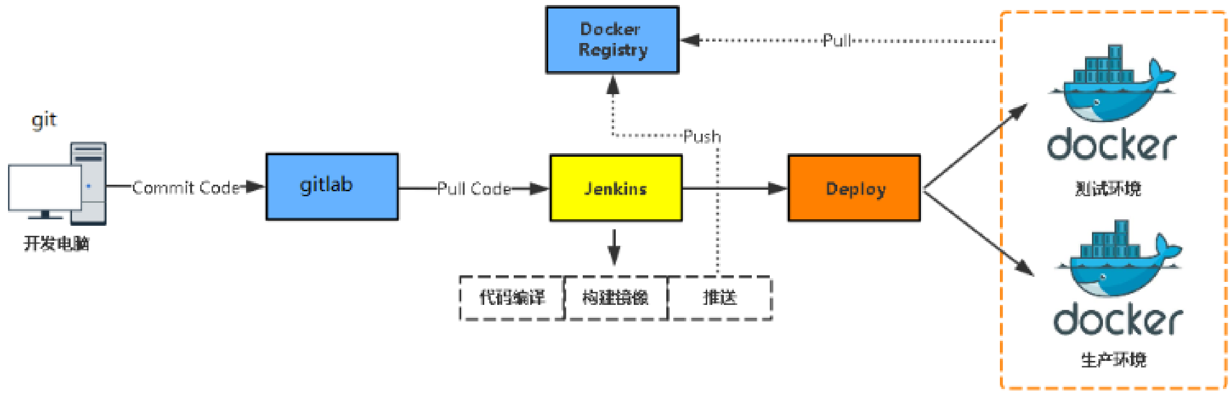
旧：

- 代码发布环境提前准备，以主机为颗粒度
- 静态

新：

- 代码发布环境 多套，以容器为颗粒度
- 编译

二、业务发布逻辑设计图

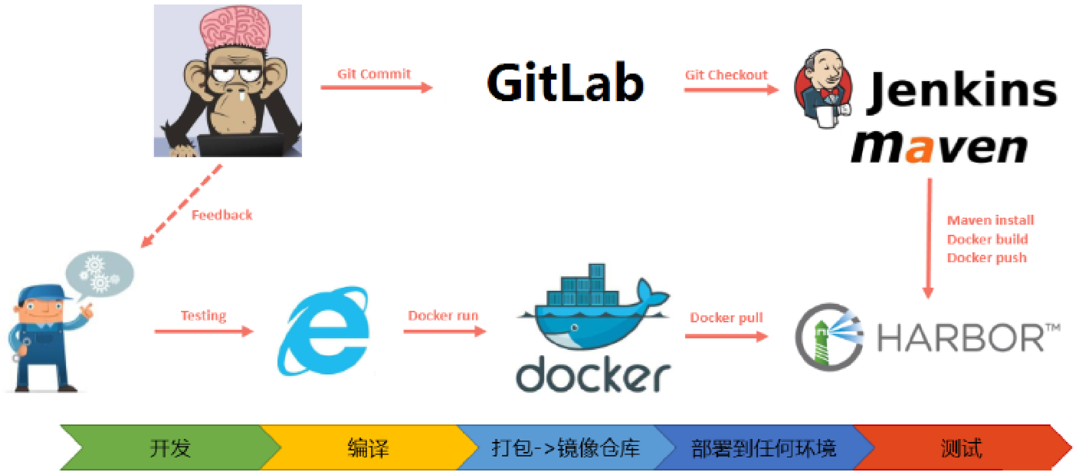


三、工具使用流程图

- 工具

  - git
  - gitlab
  - jenkins
  - tomcat
  - maven
  - harbor
  - docker

- 流程图



四、主机规划

序号	主机名	主机IP	主机功能	软件
1	dev	192.168.122.5	开发者 项目代码 solo	git
2	gitlab-server	192.168.122.6	代码仓库	gitlab
3	jenkins-server	192.168.122.7	编译代码、打包镜像、项目发布	jenkins、docker、git
4	harbor-server	192.168.122.8	存储容器镜像	harbor、docker
5	web-server	192.168.122.9	运行容器，项目上线	docker

五、主机准备

5.1 主机名

```
[root@localhost ~]# hostnamectl set-hostname XXX
```

5.2 IP

```
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
TYPE="Ethernet"
BOOTPROTO="dhcp"
NAME="eth0"
DEVICE="eth0"
ONBOOT="yes"
IPADDR="192.168.122.X"
PREFIX="24"
GATEWAY="192.168.122.1"
DNS1="119.29.29.29"

```

5.3 主机名解析

```
[root@localhost ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.122.5 dev
192.168.122.6 gitlab-server
192.168.122.7 jenkins-server
192.168.122.8 harbor-server
192.168.122.9 web-server
```

5.4 安全

```
[root@localhost ~]# firewall-cmd --state
not running
[root@localhost ~]# getenforce
Disabled

```

5.5 时间同步

```
[root@dev ~]# crontab -l
0 */1 * * * ntpdate time1.aliyun.com
```

六、软件安装

6.1 安装git

- 开发人员主机安装git
- 下载项目及上传代码至代码仓库

```
[root@dev ~]# yum -y install git
```

6.2 安装gitlab

YUM

```
[root@gitlab-server ~]# cat /etc/yum.repos.d/gitlab.repo
[gitlab]
name=gitlab-ce
baseurl=https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el7
enabled=1
gpgcheck=0
```

安装gitlab-ce

```
[root@gitlab-server ~]# yum -y install gitlab-ce
```

修改gitlab配置文件

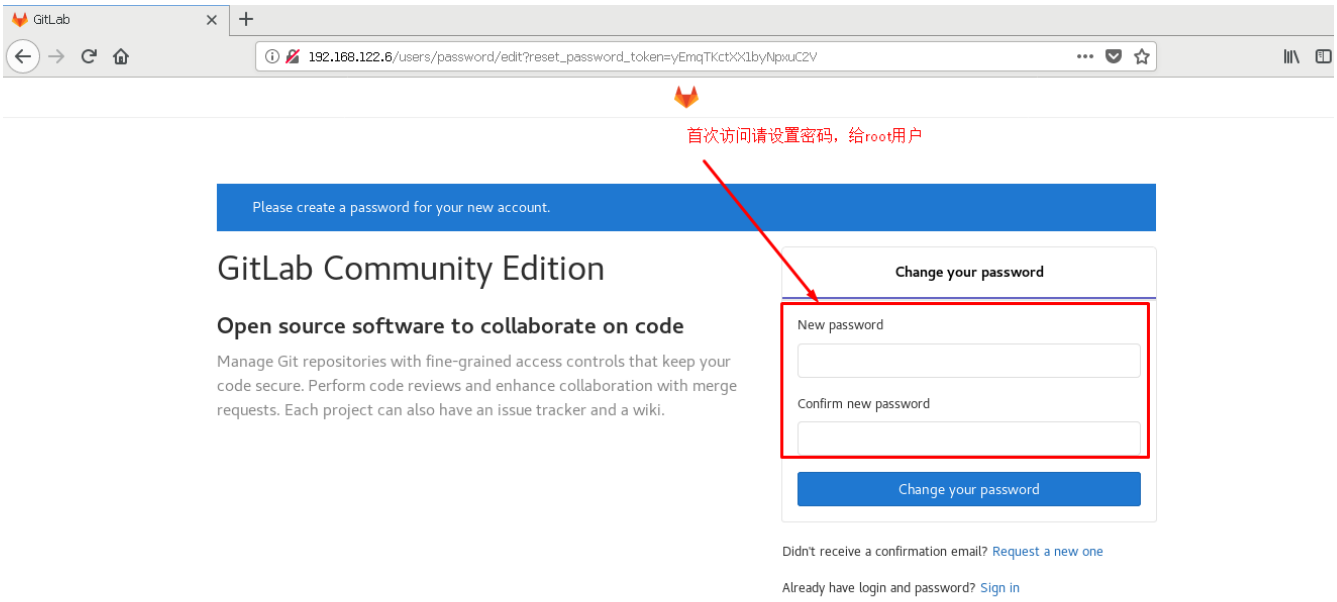
```
[root@gitlab-server ~]# vim /etc/gitlab/gitlab.rb

external_url 'http://192.168.122.6'
#在13行位置
```

启动gitlab

```
[root@gitlab-server ~]# gitlab-ctl reconfigure

[root@gitlab-server ~]# gitlab-ctl status
```

通过web页面访问

http://192.168.122.6



192.168.122.6/users/password/edit?reset_password_token=yEmqTKctXX1byNpxuC2V

首次访问请设置密码，给root用户

Please create a password for your new account.

GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Change your password

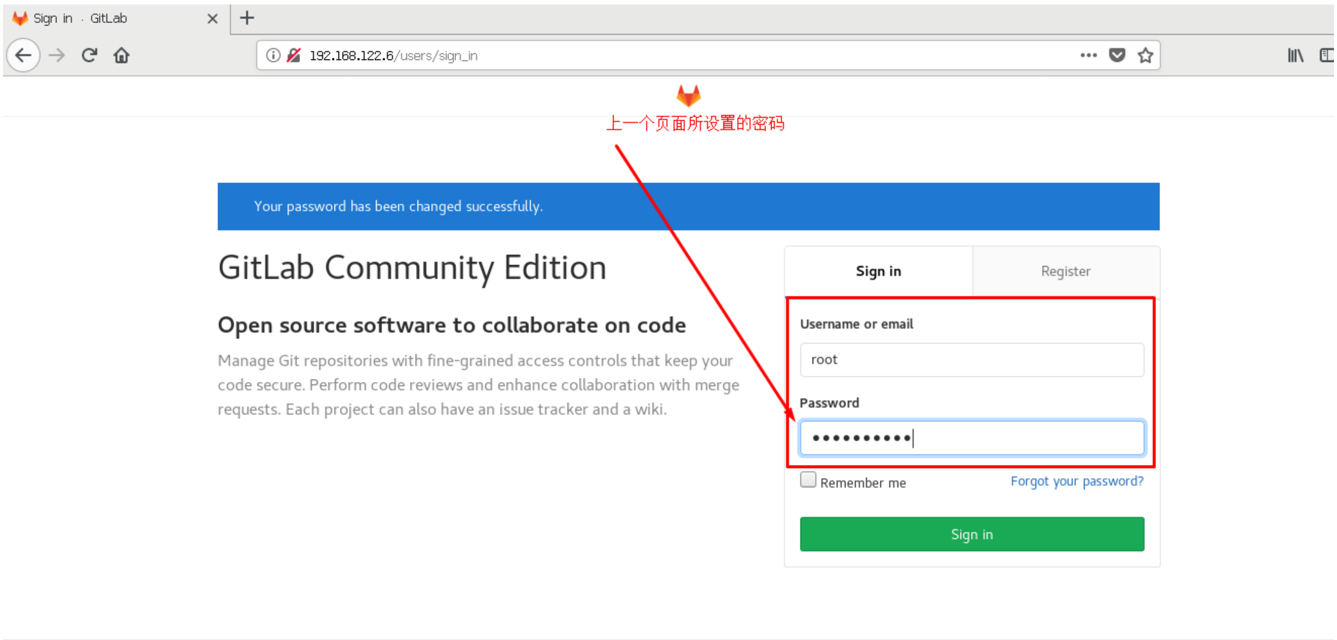
New password

Confirm new password

Change your password

Didn't receive a confirmation email? Request a new one

Already have login and password? Sign in



上一个页面所设置的密码

Your password has been changed successfully.

GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in | Register

Username or email: root

Password: ••••••••••

□ Remember me Forgot your password?

Sign in

6.3 安装jenkins、docker、maven

在jenkins-server上安装

jdk

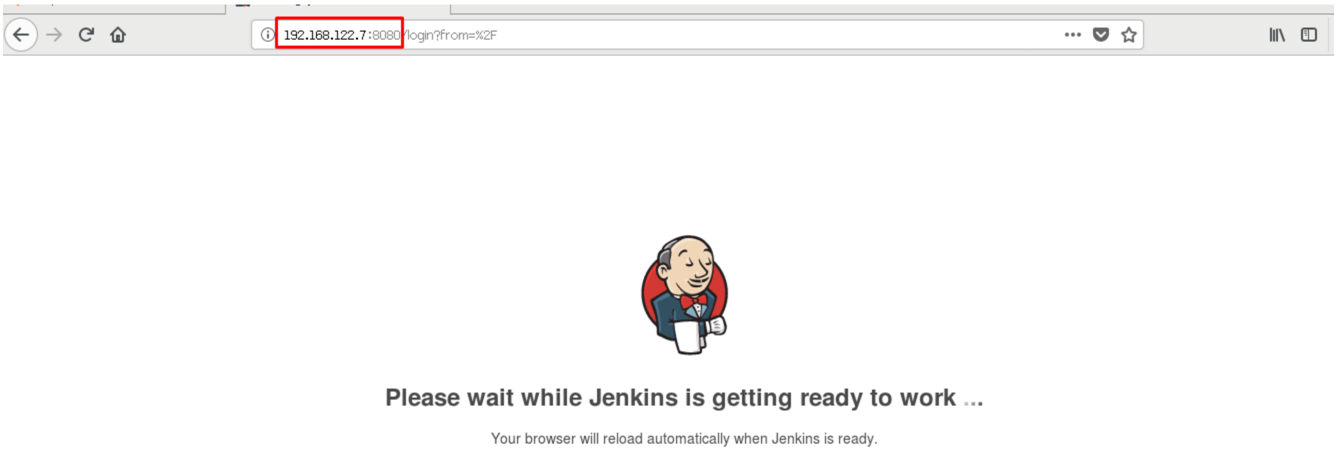
```
[root@jenkins-server ~]# tar xf jdk-8u191-linux-x64.tar.gz
[root@jenkins-server ~]# ls
anaconda-ks.cfg                jdk1.8.0_191
apache-maven-3.6.1-bin.tar.gz  jdk-8u191-linux-x64.tar.gz
apache-tomcat-8.5.40.tar.gz
[root@jenkins-server ~]# mv jdk1.8.0_191 /usr/local/jdk
[root@jenkins-server ~]# ls /usr/local/jdk
bin             lib           src.zip
COPYRIGHT       LICENSE       THIRDPARTYLICENSEREADME-JAVAFX.txt
include         man           THIRDPARTYLICENSEREADME.txt
javafx-src.zip  README.html
jre             release


[root@jenkins-server ~]# tail -2 /etc/profile
export JAVA_HOME=/usr/local/jdk
export PATH=${JAVA_HOME}/bin:$PATH

[root@jenkins-server ~]# source /etc/profile
[root@jenkins-server ~]# java -version
```

```
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)

```

jenkins

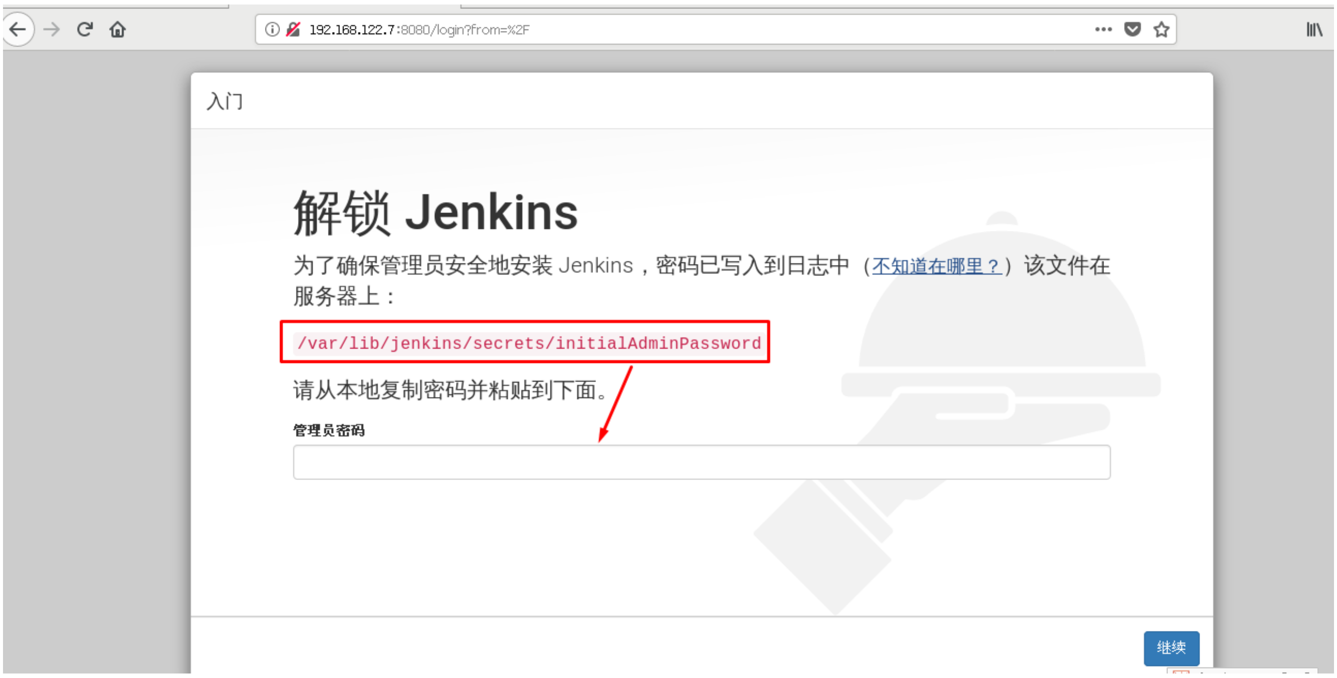
```
[root@jenkins-server ~]# wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat/jenkins.repo

[root@jenkins-server ~]# rpm --import https://pkg.jenkins.io/redhat/jenkins.io.key


[root@jenkins-server ~]# yum -y install jenkins


#修改/etc/rc.d/init.d/jenkins,添加java

84 /usr/local/jdk/bin/java

#修改/etc/sysconfig/jenkins,添加java
19 JENKINS_JAVA_CMD="/usr/local/jdk/bin/java"


#检查是否开机自启动
[root@jenkins-server ~]# chkconfig --list

注：该输出结果只显示 SysV 服务，并不包含
原生 systemd 服务。SysV 配置数据
可能被原生 systemd 配置覆盖。

      要列出 systemd 服务，请执行 'systemctl list-unit-files'。
      查看在具体 target 启用的服务请执行
      'systemctl list-dependencies [target]'。

jenkins         0:关    1:关    2:关    3:开    4:关    5:开    6:关

#如果没有开机自启动
[root@jenkins-server ~]# chkconfig jenkins on

#启动jenkins
[root@jenkins-server ~]# systemctl start jenkins
```

jenkins访问



Please wait while Jenkins is getting ready to work ...

Your browser will reload automatically when Jenkins is ready.



解锁 Jenkins

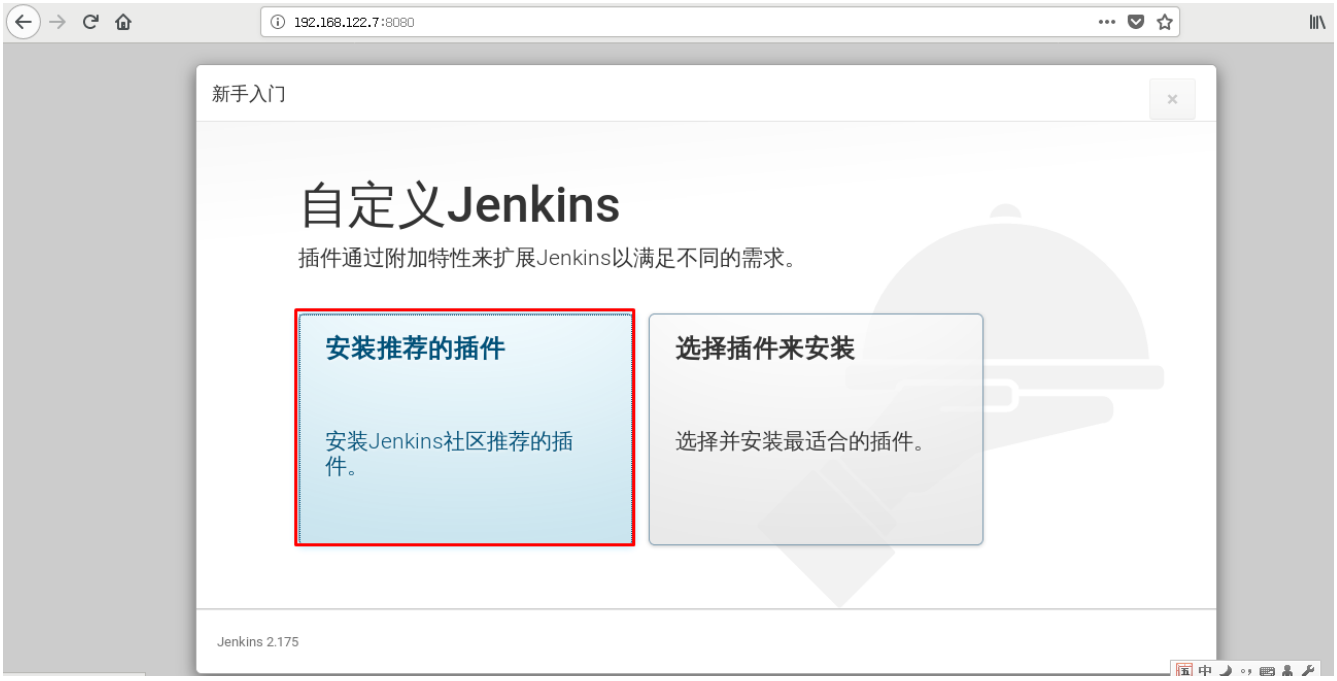
为了确保管理员安全地安装 Jenkins，密码已写入到日志中（不知道在哪里？）该文件在服务器上：

```
/var/lib/jenkins/secrets/initialAdminPassword
```

请从本地复制密码并粘贴到下面。

管理员密码

```
[root@jenkins-server ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
1143fd06d41f4673803e94533c7c71be
```



192.168.122.7:8080

新手入门

自定义Jenkins

插件通过附加特性来扩展Jenkins以满足不同的需求。

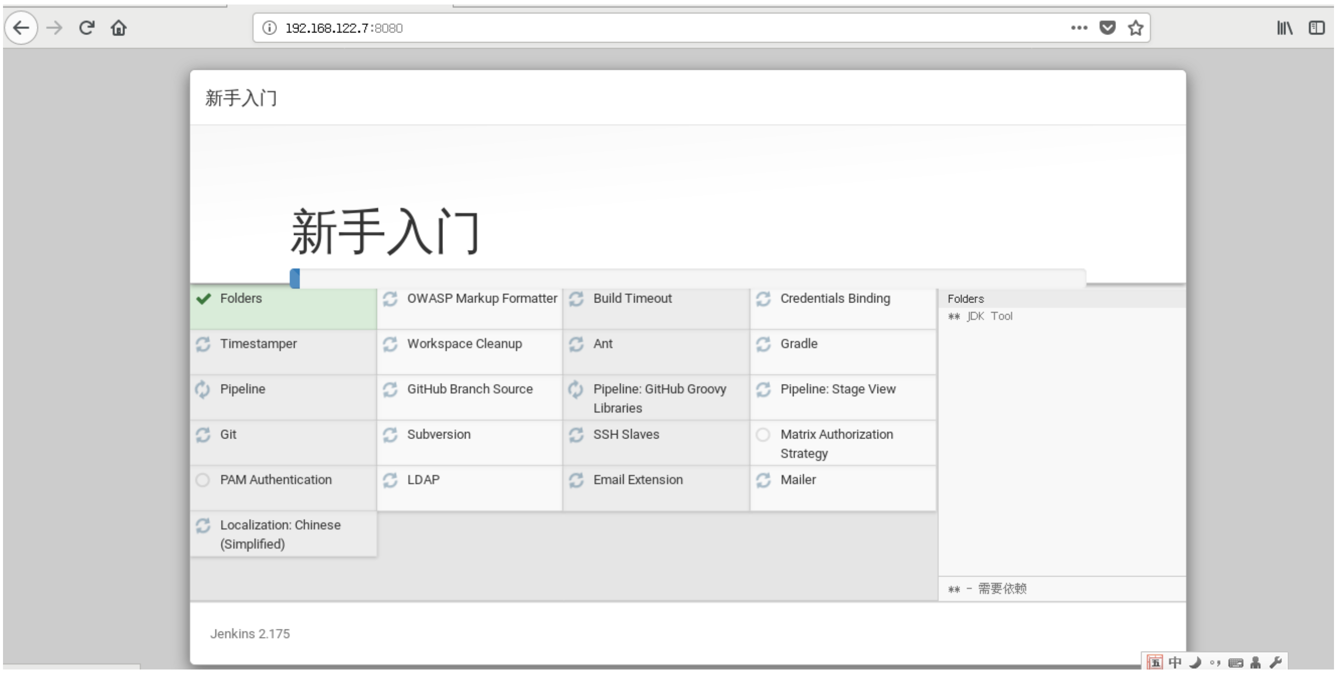
安装推荐的插件

安装Jenkins社区推荐的插件。

选择插件来安装

选择并安装最适合的插件。

Jenkins 2.175



192.168.122.7:8080

新手入门

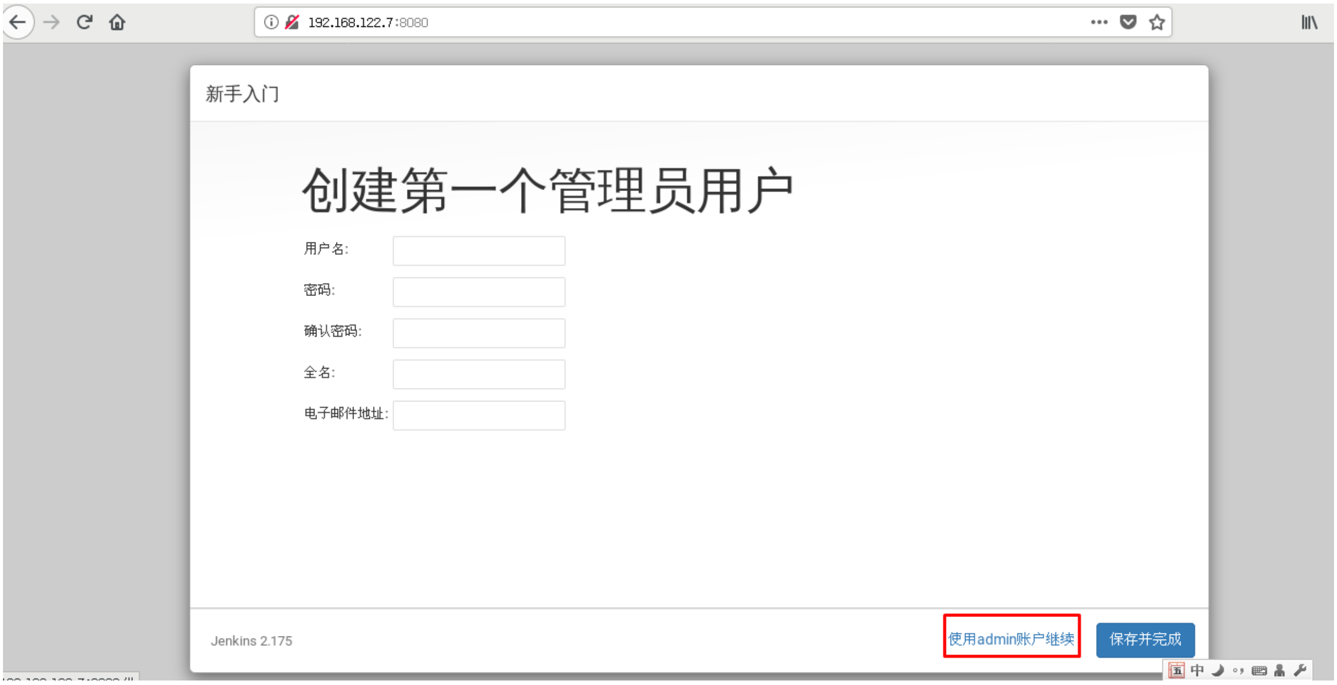
新手入门

✓ Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	Subversion	SSH Slaves	○ Matrix Authorization Strategy
○ PAM Authentication	LDAP	Email Extension	Mailer
Localization: Chinese (Simplified)			

Folders
** JDK Tool

** - 需要依赖

Jenkins 2.175



新手入门

创建第一个管理员用户

用户名：

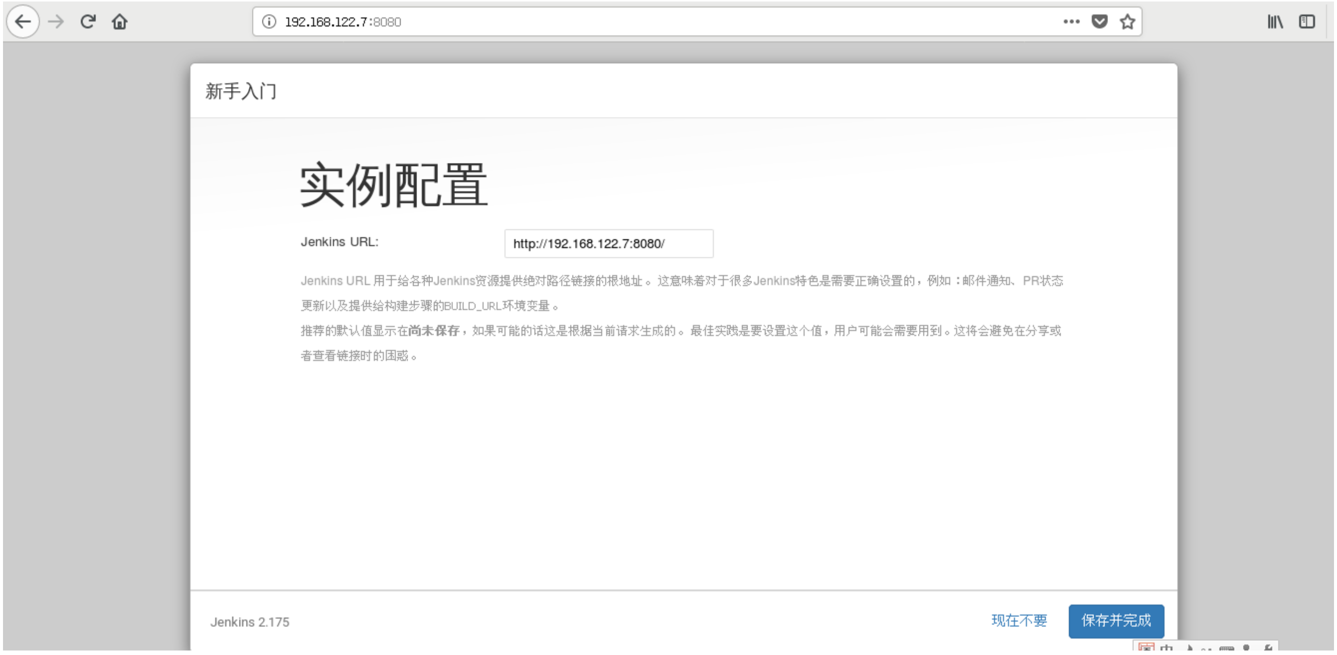
密码：

确认密码：

全名：

电子邮件地址：

Jenkins 2.175



新手入门

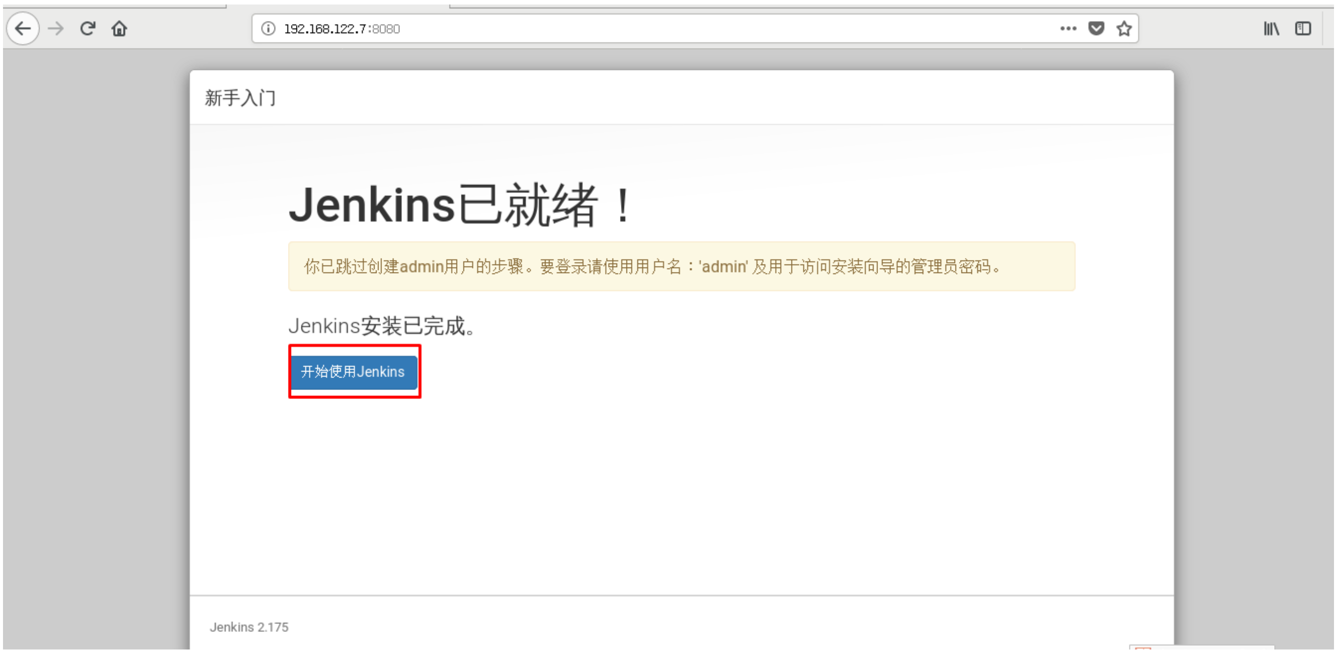
实例配置

Jenkins URL: http://192.168.122.7:8080/

Jenkins URL 用于给各种Jenkins资源提供绝对路径链接的根地址。这意味着对于很多Jenkins特色是需要正确设置的，例如：邮件通知、PR状态更新以及提供给构建步骤的BUILD_URL环境变量。

推荐的默认值显示在**尚未保存**，如果可能的话这是根据当前请求生成的。最佳实践是要设置这个值，用户可能会需要用到。这将会避免在分享或者查看链接时的困惑。

Jenkins 2.175



192.168.122.7:8080

新手入门

Jenkins已就绪！

你已跳过创建admin用户的步骤。要登录请使用用户名：'admin' 及用于访问安装向导的管理员密码。

Jenkins安装已完成。

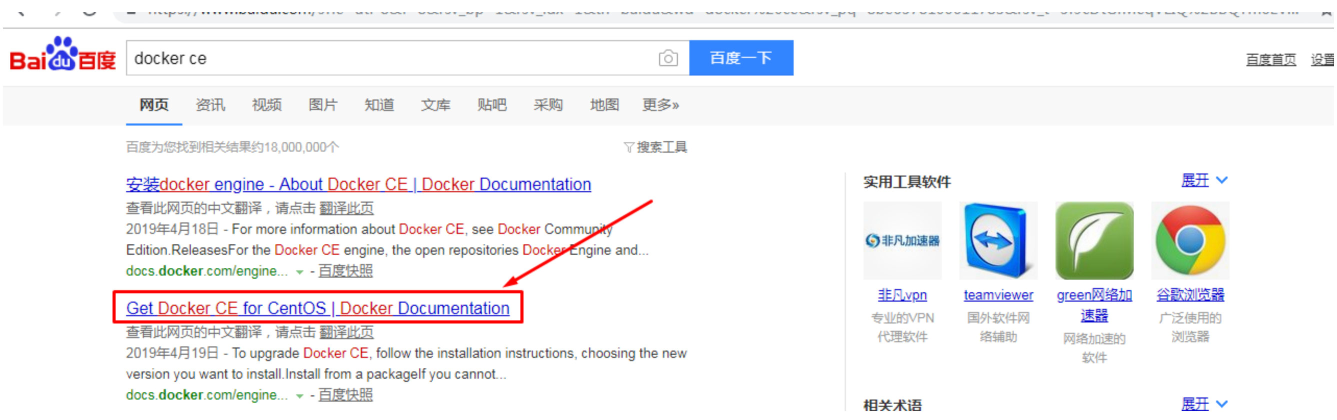
maven

```
[root@jenkins-server ~]# ls
anaconda-ks.cfg                apache-tomcat-8.5.40.tar.gz
apache-maven-3.6.1-bin.tar.gz  jdk-8u191-linux-x64.tar.gz
[root@jenkins-server ~]# tar xf apache-maven-3.6.1-bin.tar.gz
[root@jenkins-server ~]# ls
anaconda-ks.cfg     apache-maven-3.6.1-bin.tar.gz  jdk-8u191-linux-x64.tar.gz
apache-maven-3.6.1  apache-tomcat-8.5.40.tar.gz
[root@jenkins-server ~]# mv apache-maven-3.6.1 /usr/local/maven

#配置环境变量
[root@jenkins-server ~]# tail -3 /etc/profile
export JAVA_HOME=/usr/local/jdk
export MAVEN_HOME=/usr/local/maven
export PATH=${JAVA_HOME}/bin:${MAVEN_HOME}/bin:$PATH

[root@jenkins-server ~]# source /etc/profile
```

git

```
[root@jenkins-server ~]# yum -y install git
```

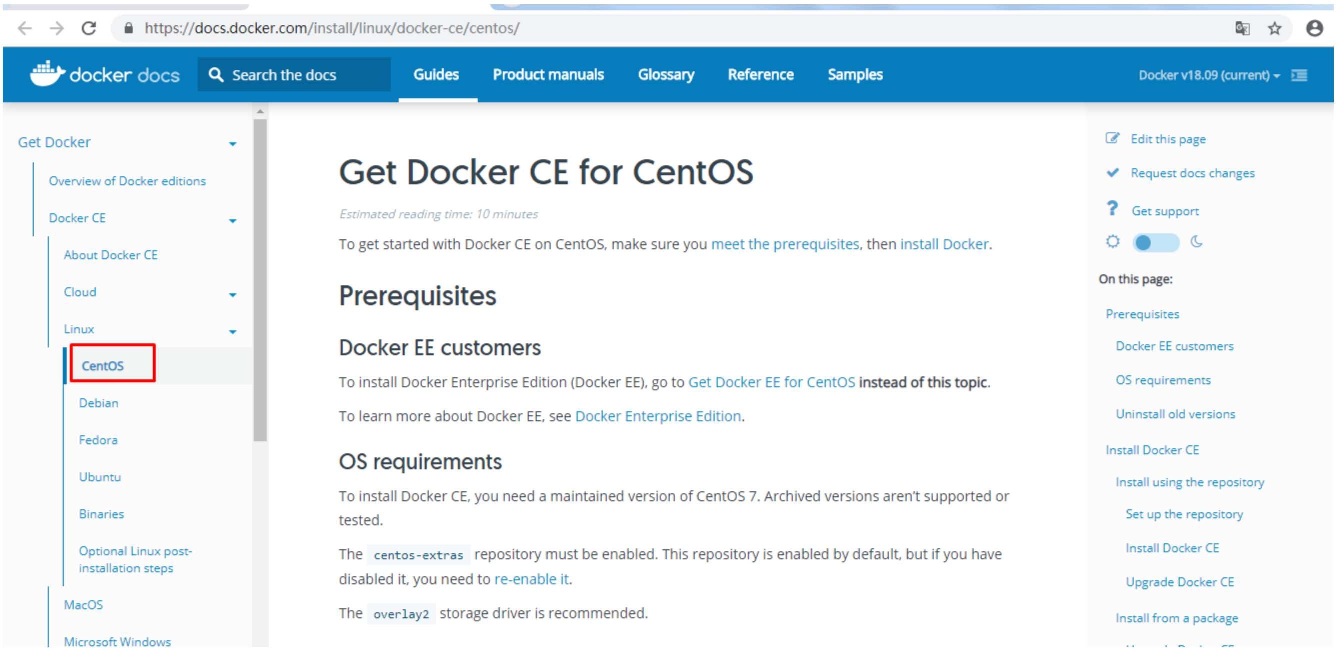
docker



百度为您找到相关结果约18,000,000个

安装docker engine - About Docker CE | Docker Documentation
查看此网页的中文翻译，请点击 翻译此页
2019年4月18日 - For more information about Docker CE, see Docker Community Edition.ReleasesFor the Docker CE engine, the open repositories Docker Engine and...
docs.docker.com/engine... - 百度快照

Get Docker CE for CentOS | Docker Documentation
查看此网页的中文翻译，请点击 翻译此页
2019年4月19日 - To upgrade Docker CE, follow the installation instructions, choosing the new version you want to install.Install from a packageIf you cannot...
docs.docker.com/engine... - 百度快照



Get Docker CE for CentOS

Estimated reading time: 10 minutes

To get started with Docker CE on CentOS, make sure you meet the prerequisites, then install Docker.

Prerequisites

Docker EE customers

To install Docker Enterprise Edition (Docker EE), go to Get Docker EE for CentOS **instead of this topic.**

To learn more about Docker EE, see Docker Enterprise Edition.

OS requirements

To install Docker CE, you need a maintained version of CentOS 7. Archived versions aren't supported or tested.

The `centos-extras` repository must be enabled. This repository is enabled by default, but if you have disabled it, you need to re-enable it.

The `overlay2` storage driver is recommended.

```
[root@jenkins-server ~]# yum install -y yum-utils   device-mapper-persistent-data lvm2

```

```
[root@jenkins-server ~]# yum-config-manager     --add-repo https://download.docker.com/linux/centos/docker-ce.repo

[root@jenkins-server ~]# yum -y install docker-ce

#修改docker.service文件

[root@jenkins-server ~]# cat /usr/lib/systemd/system/docker.service


[Service]
...
ExecStart=/usr/bin/dockerd
...



#添加daemon.json
[root@jenkins-server ~]# systemctl daemon-reload
[root@jenkins-server ~]# systemctl start docker

[root@jenkins-server ~]# cat /etc/docker/daemon.json
{
        "insecure-registries": ["http://192.168.122.8"]
}

[root@jenkins-server ~]# systemctl restart docker


```

6.4 安装harbor、docker

在harbor-server安装

harbor

```
#epel-release
[root@harbor-server ~]# yum -y install epel-release

#pip工具
[root@harbor-server ~]# yum -y install python2-pip

```

```
[root@harbor-server ~]# pip install --upgrade pip

#docker-compose工具
[root@harbor-server ~]# pip install docker-compose

#部署harbor
[root@harbor-server ~]# ls
anaconda-ks.cfg  harbor-offline-installer-v1.7.5.tgz
[root@harbor-server ~]# tar xf harbor-offline-installer-v1.7.5.tgz

[root@harbor-server ~]# ls
anaconda-ks.cfg  harbor  harbor-offline-installer-v1.7.5.tgz
[root@harbor-server ~]# cd harbor/
[root@harbor-server harbor]# ls
common                          docker-compose.yml    LICENSE
docker-compose.chartmuseum.yml  harbor.cfg            open_source_license
docker-compose.clair.yml        harbor.v1.7.5.tar.gz  prepare
docker-compose.notary.yml       install.sh
[root@harbor-server harbor]# vim harbor.cfg

hostname = 192.168.122.8

#如果需要在harbor-server主机上打包容器应用镜像并上传，需要修改docker daemon
#本例修改，请参照docker部署部分


#修改docker daemon继续启动harbor
[root@harbor-server harbor]# pwd
/root/harbor
[root@harbor-server harbor]# ./prepare
Generated and saved secret to file: /data/secretkey
Generated configuration file: ./common/config/nginx/nginx.conf
Generated configuration file: ./common/config/adminserver/env
Generated configuration file: ./common/config/core/env
Generated configuration file: ./common/config/registry/config.yml
Generated configuration file: ./common/config/db/env
Generated configuration file: ./common/config/jobservice/env
Generated configuration file: ./common/config/jobservice/config.yml
Generated configuration file: ./common/config/log/logrotate.conf
Generated configuration file: ./common/config/registryctl/env
Generated configuration file: ./common/config/core/app.conf
Generated certificate, key file: ./common/config/core/private_key.pem, cert file: ./common/config/registry/root.crt
The configuration files are ready, please use docker-compose to start the service.
[root@harbor-server harbor]# ./install.sh

#验证harbor是否可用
[root@harbor-server harbor]# docker login http://192.168.122.8
Username: admin #输入用户名
Password: Harbor12345 #输入密码
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
```

```

Login Succeeded


```

docker

```
[root@harbor-server ~]# yum install -y yum-utils   device-mapper-persistent-data lvm2

[root@harbor-server ~]# yum-config-manager     --add-repo https://download.docker.com/linux/centos/docker-ce.repo

[root@harbor-server ~]# yum -y install docker-ce

#修改docker daemon使用本地harbor仓库
[root@harbor-server harbor]# systemctl start docker

[root@harbor-server harbor]# cat /usr/lib/systemd/system/docker.service

[Service]
...
ExecStart=/usr/bin/dockerd
...

#添加daemon.json文件
[root@harbor-server harbor]# cat /etc/docker/daemon.json
{
        "insecure-registries": ["http://192.168.122.8"]
}

[root@harbor-server harbor]# systemctl daemon-reload
[root@harbor-server harbor]# systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[root@harbor-server harbor]# systemctl restart docker

```

6.5 安装docker

在web-server安装

```
[root@web-server ~]# yum install -y yum-utils   device-mapper-persistent-data   lvm2
```

```

[root@web-server ~]# yum-config-manager     --add-repo https://download.docker.com/linux/centos/docker-ce.repo

[root@web-server ~]# yum -y install docker-ce

#启动
[root@web-server ~]# systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.

[root@web-server ~]# systemctl start docker

#修改docker.service
[root@web-server harbor]# cat /usr/lib/systemd/system/docker.service

[Service]
...
ExecStart=/usr/bin/dockerd
...

#修改daemon.json
[root@web-server ~]# cat /etc/docker/daemon.json
{
        "insecure-registries": ["http://192.168.122.8"]
}

#重启docker
[root@web-server ~]# systemctl daemon-reload
[root@web-server ~]# systemctl restart docker

#验证是否可以使用harbor

[root@web-server ~]# docker login http://192.168.122.8
Username: admin
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

```

七、配置

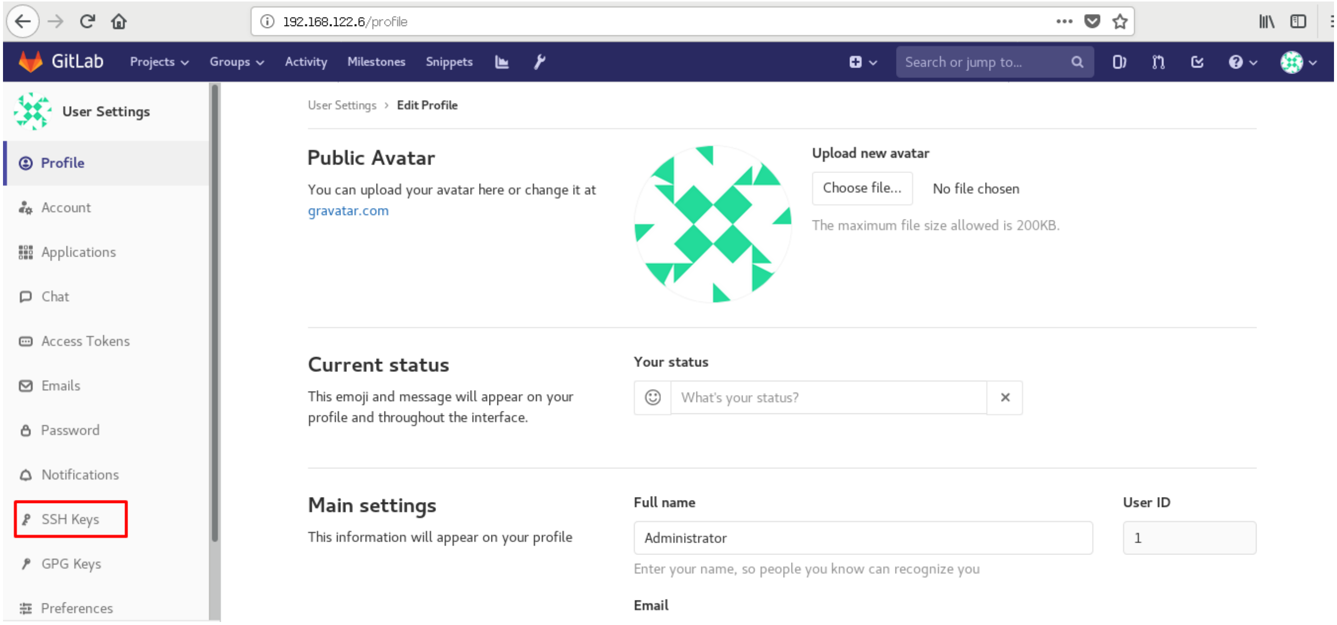
7.1 配置开发人员主机密钥至gitlab

生成密钥

```
[root@dev ~]# ssh-keygen -t rsa -f /root/.ssh/id_rsa -N ''
Generating public/private rsa key pair.
Created directory '/root/.ssh'.
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:DBOkXzfLmS1y2o8DCrZGL/BRlg8zjaLDCuPcgU+fQOI root@dev
The key's randomart image is:
+---[RSA 2048]----+
|     .o          |
|     . .         |
|    . o+. o      |
|   E..O=.o *     |
| .o. +.=S B .    |
|o.=+=.  o= .     |
|+o+*==.....      |
|.o o=+o   .o     |
|   . ..   ...    |
+----[SHA256]-----+
[root@dev ~]# ls /root/.ssh
id_rsa  id_rsa.pub


#生成便于标识的开发者密钥

[root@dev ~]# ssh-keygen -t rsa -f /root/.ssh/id_rsa -C "dev@aiops.net.cn" -N ''
Generating public/private rsa key pair.
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:99eshaDdkSQ664cP2bszq5L4+VFtLQ6X0s/f36bGuDA dev@aiops.net.cn
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|                 |
|            . .  |
|           . = + |
|        S + = X .|
|         . X O O |
|       . .E.+o= *|
|      . oo.==ooo=|
|       .oo++*Boo*|
+----[SHA256]-----+
[root@dev ~]# ls /root/.ssh/
id_rsa  id_rsa.pub

[root@dev ~]# cat /root/.ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDMO8YSAnh4gHDNa/nGK4OZrl2UdjoYj1fEzGpuDQ0Ck8kK6dKa0aDKGa
1ON/kBifNqGKewd7LjK7QzC2gmGVFZ+7dXdRYYk3Z/ud5HEBTAvUWh5yegD3YlZxzOssYe4xhG4FN2PYUErQUP
svM1YvAQolVZJXnSuDJJd7ZFsFObGVjm4RqYc4nLUlQjaUH2c8lozmdAVpte3DLu0HJuJl/B8IoF12QpeMIKa9
gTjumJSIBR4jWMMDqw3HIaiw3uC/EP+9T5dXB4U/r1fVzjwGeSVM3Hc9cM0xBFBUbNDS94DKqkS9eAa3+G2vN6
SsRoVUSsNotaWKo1xnNMEmxlwEsn dev@aiops.net.cn
```

gitlab页面添加开发者密钥

1557019926865



Public Avatar

You can upload your avatar here or change it at gravatar.com

Upload new avatar

Choose file... No file chosen

The maximum file size allowed is 200KB.

Current status

This emoji and message will appear on your profile and throughout the interface.

Your status

What's your status?

Main settings

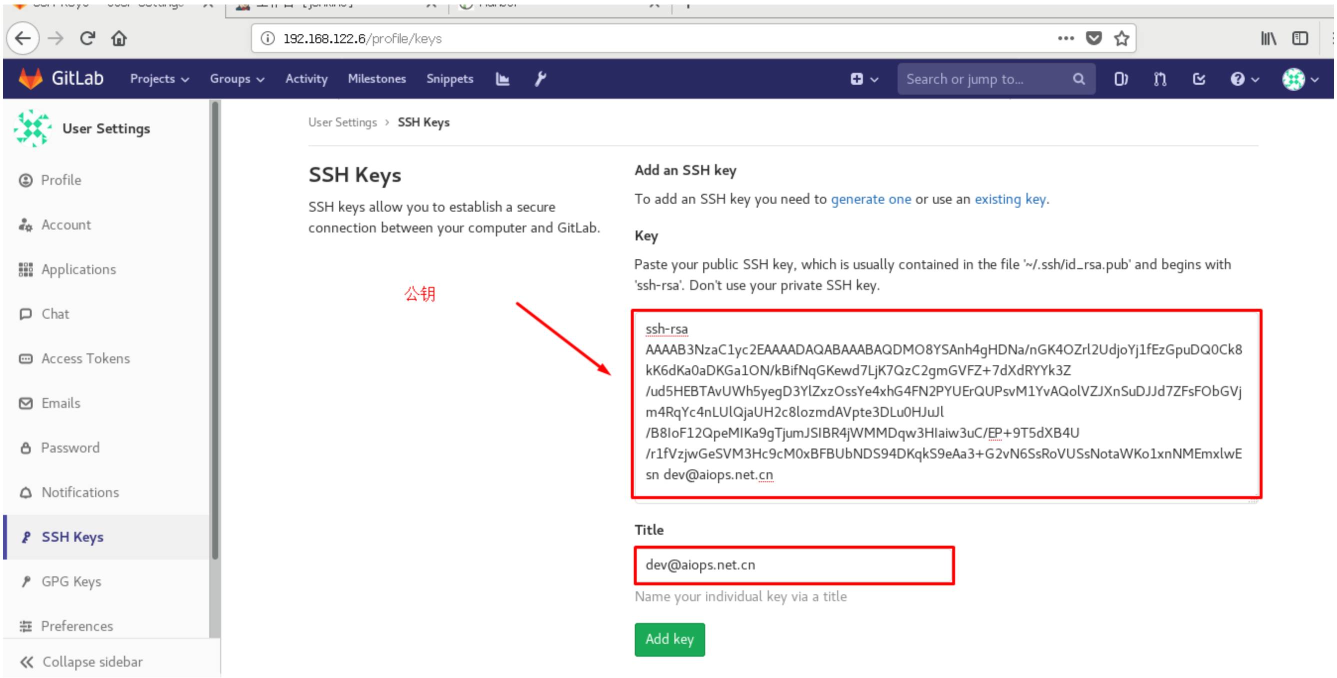
This information will appear on your profile

Full name: Administrator

Enter your name, so people you know can recognize you

User ID: 1

Email



SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

To add an SSH key you need to generate one or use an existing key.

Key

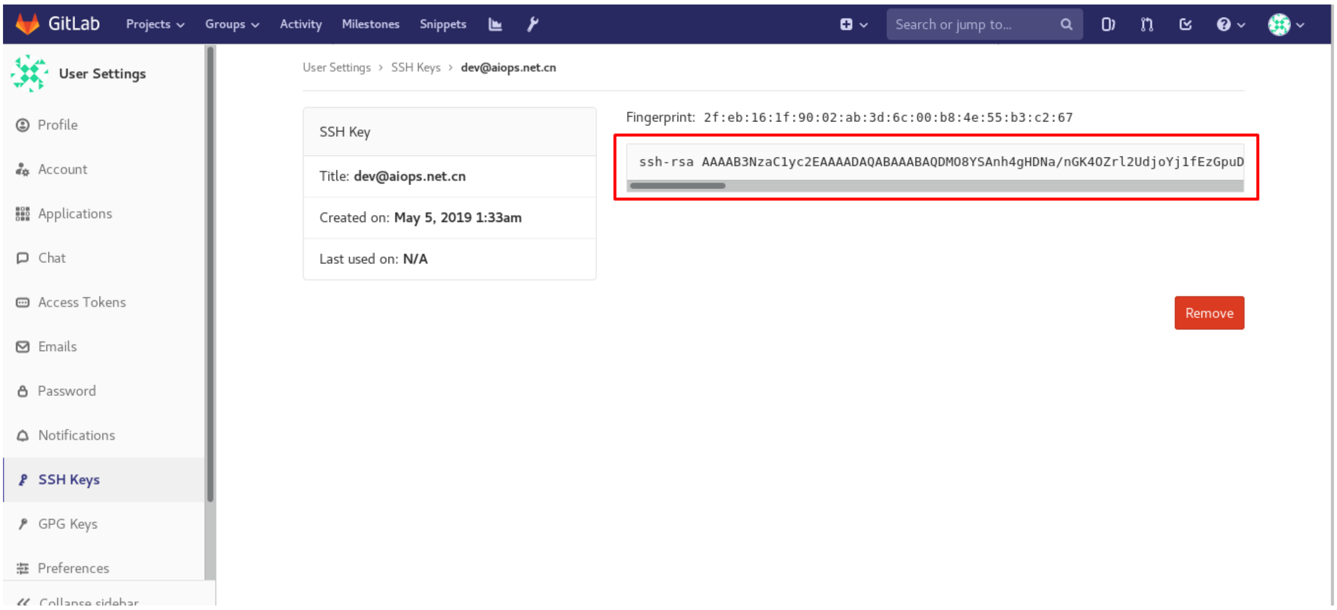
Paste your public SSH key, which is usually contained in the file '~/.ssh/id_rsa.pub' and begins with 'ssh-rsa'. Don't use your private SSH key.

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDMO8YSAnh4gHDNa/nGK4OZrl2UdjoYj1fEzGpuDQ0Ck8
kK6dKa0aDKGa1ON/kBifNqGKewd7LjK7QzC2gmGVFZ+7dXdRYYk3Z
/ud5HEBTAvUWh5yegD3YlZxzOssYe4xhG4FN2PYUErQUPsvM1YvAQolVZJXnSuDJJd7ZFsFObGVj
m4RqYc4nLUlQjaUH2c8lozmdAVpte3DLu0HJuJl
/B8IoF12QpeMIKa9gTjumJSIBR4jWMMDqw3HIaiw3uC/EP+9T5dXB4U
/r1fVzjwGeSVM3Hc9cM0xBFBUbNDS94DKqkS9eAa3+G2vN6SsRoVUSsNotaWKo1xnNMEmxlwE
sn dev@aiops.net.cn
```

Title: dev@aiops.net.cn

Name your individual key via a title

Add key



SSH Key

Title: **dev@aiops.net.cn**

Created on: **May 5, 2019 1:33am**

Last used on: **N/A**

Fingerprint: `2f:eb:16:1f:90:02:ab:3d:6c:00:b8:4e:55:b3:c2:67`

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDMO8YSAnh4gHDNa/nGK4OZrl2UdjoYj1fEzGpuD
```

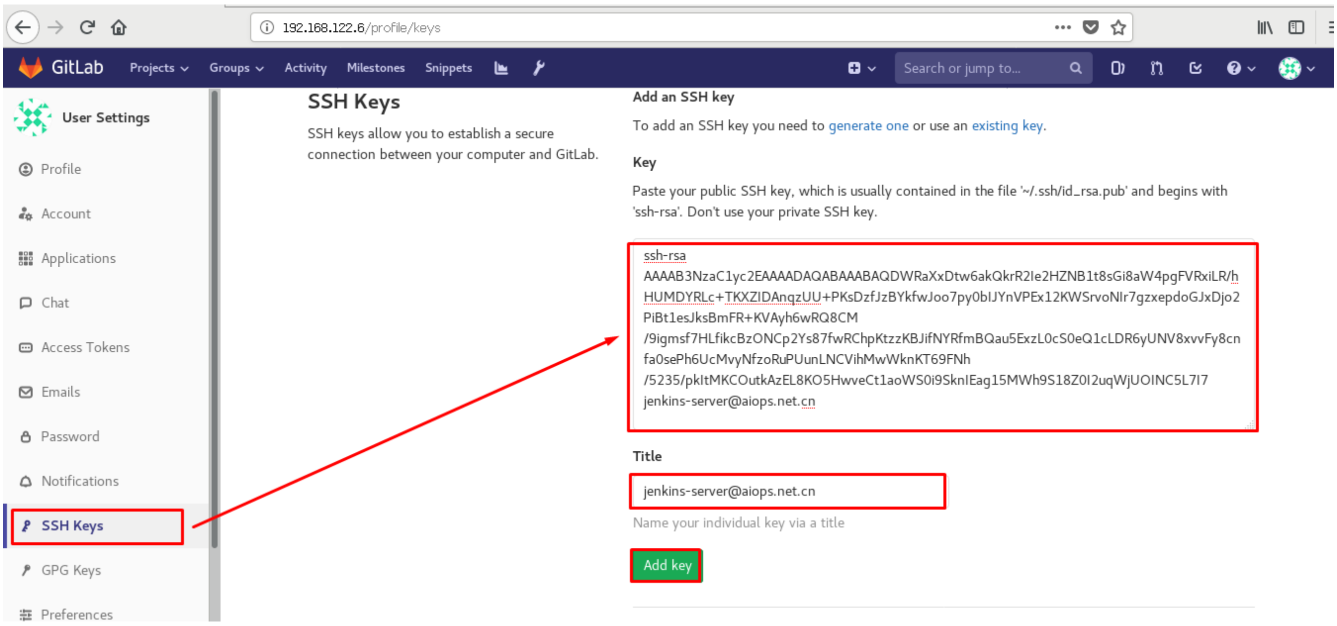
7.2 配置jenkins-server主机密钥至gitlab

7.2.1 生成密钥对

```
[root@jenkins-server ~]# ssh-keygen -t rsa -f /root/.ssh/id_rsa -C "jenkins-server@aiops.net.cn" -N ''
Generating public/private rsa key pair.
Created directory '/root/.ssh'.
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:TuwOapIXWPrSGlFBkvuKmTnzzlRQXAZ421ATSP+e9tg jenkins-server@aiops.net.cn
The key's randomart image is:
+---[RSA 2048]----+
|  .**=*.         |
|  oo=+ .         |
|  .o.+.          |
|  .oo .o         |
|  .=.   S        |
|  ooo  = .       |
| =o= .. *        |
|Bo=.=. + +       |
| ==*.   o E      |
+----[SHA256]-----+
[root@jenkins-server ~]# ls /root/.ssh
id_rsa  id_rsa.pub

[root@jenkins-server ~]# cat /root/.ssh/id_rsa.pub
```

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDWRaXxDtw6akQkrR2Ie2HZNB1t8sGi8aW4pgFVRxiLR/hHUMDYRLc+TK
XZIDAnqzUU+PKsDzfJzBYkfwJoo7pyObIJYnVPEx12KWSrvoNIr7gzxepdoGJxDjo2PiBt1esJksBmFR+KVAyh
6wRQ8CM/9igmsf7HLfikcBzONCp2Ys87fwRChpKtzzKBJifNYRfmBQau5ExzL0cS0eQ1cLDR6yUNV8xvvFy8cn
fa0sePh6UcMvyNfzoRuPUunLNCVihMwWknKT69FNh/5235/pkItMKCOutkAzEL8KO5HwveCt1aoWS0i9SknIEa
g15MWh9S18Z0I2uqWjUOINC5L7I7 jenkins-server@aiops.net.cn
```

7.2.2 在gitlab-server上添加公钥



SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

To add an SSH key you need to generate one or use an existing key.

Key

Paste your public SSH key, which is usually contained in the file '~/.ssh/id_rsa.pub' and begins with 'ssh-rsa'. Don't use your private SSH key.

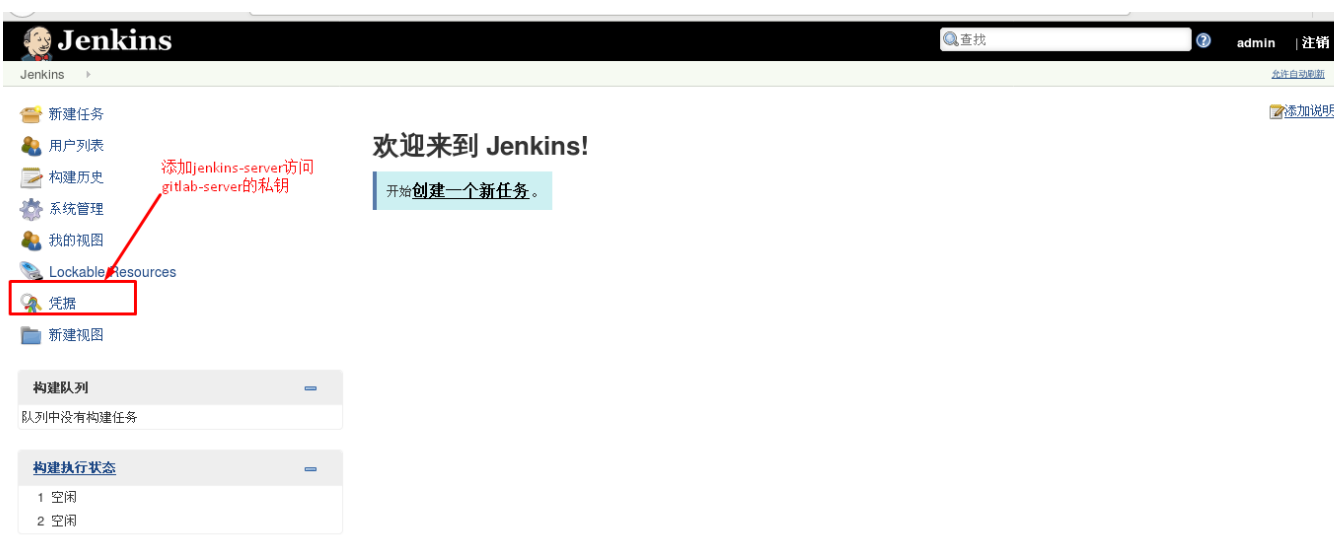
```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDWRaXxDtw6akQkrR2Ie2HZNB1t8sGi8aW4pgFVRxiLR/h
HUMDYRLc+TKXZIDAnqzUU+PKsDzfJzBYkfwJoo7pyObIJYnVPEx12KWSrvoNIr7gzxepdoGJxDjo2
PiBt1esJksBmFR+KVAyh6wRQ8CM
/9igmsf7HLfikcBzONCp2Ys87fwRChpKtzzKBJifNYRfmBQau5ExzL0cS0eQ1cLDR6yUNV8xvvFy8cn
fa0sePh6UcMvyNfzoRuPUunLNCVihMwWknKT69FNh
/5235/pkItMKCOutkAzEL8KO5HwveCt1aoWS0i9SknIEag15MWh9S18Z0I2uqWjUOINC5L7I7
jenkins-server@aiops.net.cn
```

Title

jenkins-server@aiops.net.cn

Name your individual key via a title

7.3 配置jenkins-server主机的私钥到凭据列表



添加jenkins-server访问gitlab-server的私钥

欢迎来到 Jenkins!

开始创建一个新任务。

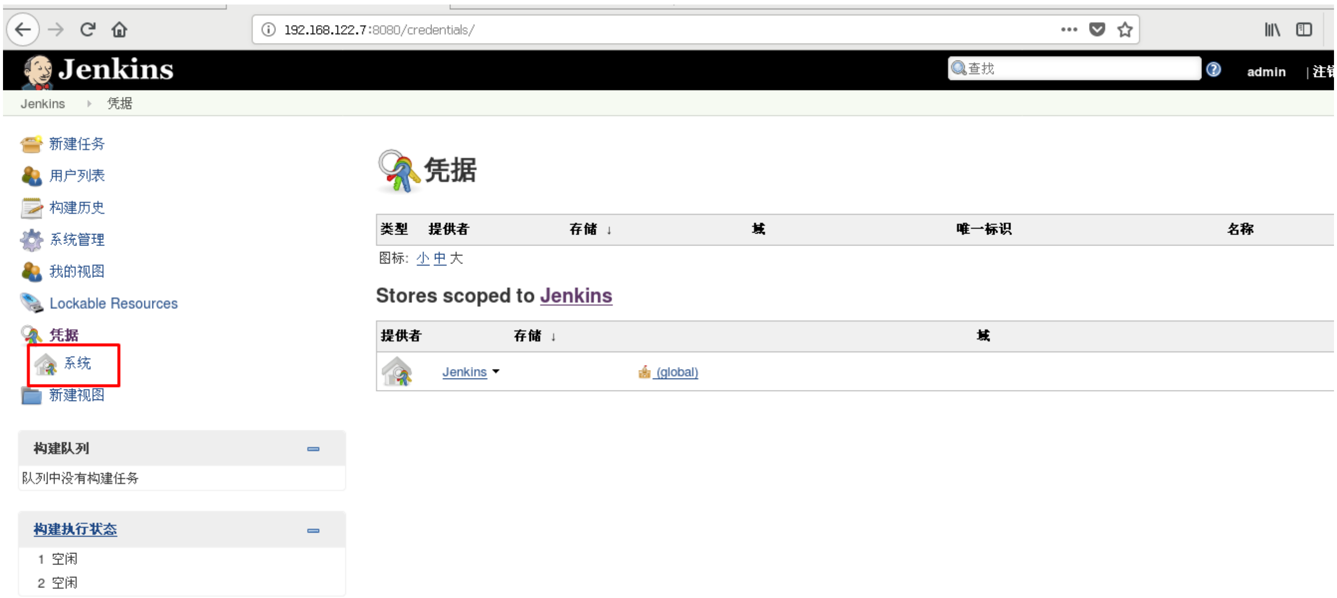
构建队列

队列中没有构建任务

构建执行状态

1 空闲

2 空闲



凭据

类型	提供者	存储 ↓	域	唯一标识	名称

图标: 小 中 大

Stores scoped to Jenkins

提供者	存储 ↓	域
Jenkins	(global)	

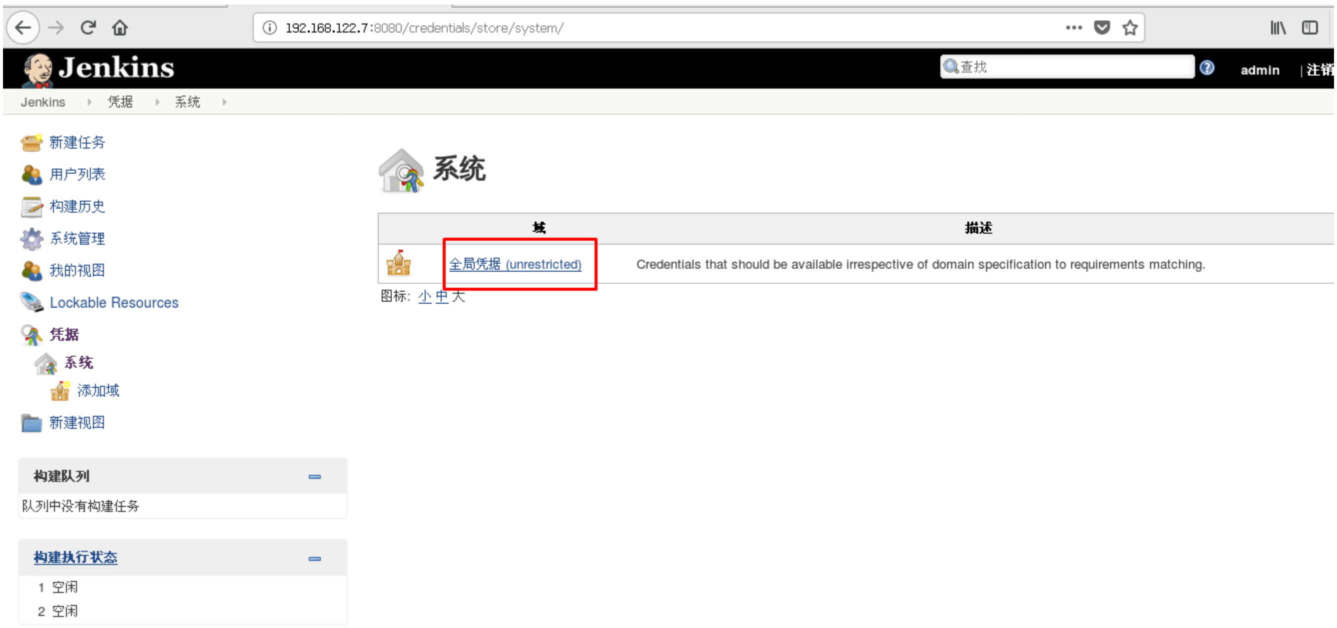
构建队列

队列中没有构建任务

构建执行状态

1 空闲

2 空闲



192.168.122.7:8080/credentials/store/system/

Jenkins

查找 admin | 注销

Jenkins › 凭据 › 系统 ›

- 新建任务
- 用户列表
- 构建历史
- 系统管理
- 我的视图
- Lockable Resources
- 凭据
  - 系统
    - 添加域
- 新建视图

构建队列

队列中没有构建任务

构建执行状态

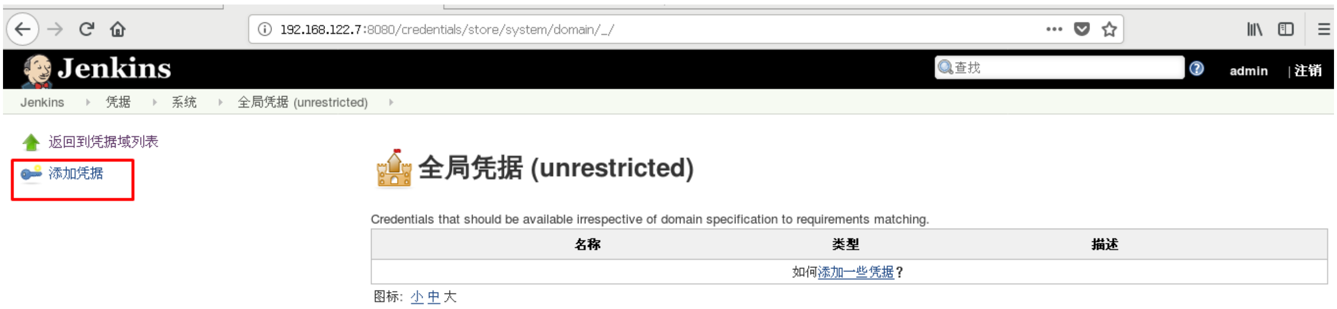
1 空闲

2 空闲

系统

域	描述
全局凭据 (unrestricted)	Credentials that should be available irrespective of domain specification to requirements matching.

图标: 小 中 大



192.168.122.7:8080/credentials/store/system/domain/_/

Jenkins

查找 admin | 注销

Jenkins › 凭据 › 系统 › 全局凭据 (unrestricted) ›

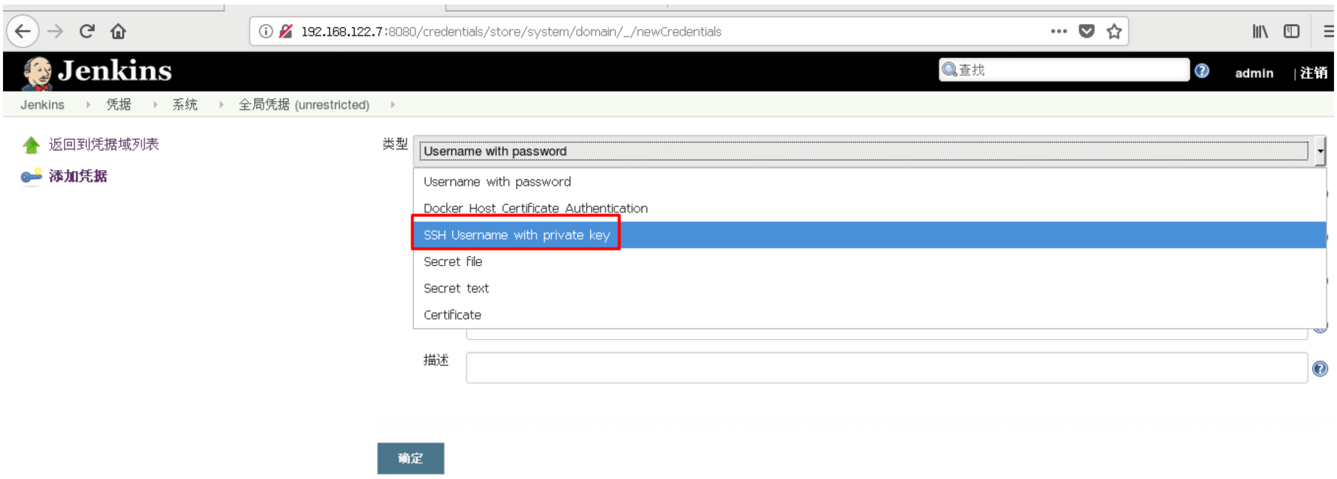
- 返回到凭据域列表
- 添加凭据

全局凭据 (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

名称	类型	描述
如何添加一些凭据?		

图标: 小 中 大



192.168.122.7:8080/credentials/store/system/domain/_/newCredentials

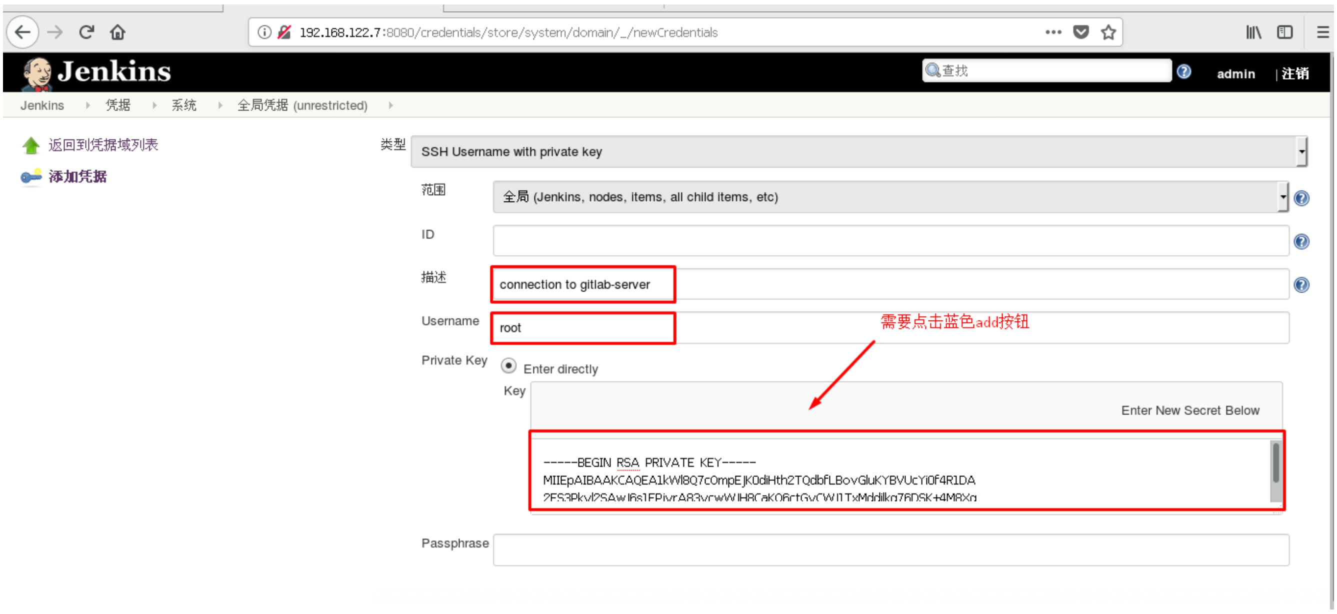
Jenkins › 凭据 › 系统 › 全局凭据 (unrestricted)

类型: Username with password

- Username with password
- Docker Host Certificate Authentication
- SSH Username with private key
- Secret file
- Secret text
- Certificate

描述

```
[root@jenkins-server ~]# cat /root/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA1kWl8Q7cOmpEJK0diHth2TQdbfLBovGluKYBVUcYi0f4R1DA
2ES3Pkyl2SAwJ6s1FPjyrA83ycwWJH8CaKO6ctGyCWJ1TxMddilkq76DSK+4M8Xq
XaBicQ46Nj4gbdXrCZLAZhUfilQMoesEUPAjP/YoJrH+xy34pHAczjQqdmLPO38E
QoaSrc8ygSYnzWEX5gUGruRMcy9HEtHkNXCw0eslDVfMb7xcvHJ32tLHj4elHDL8
jX86Ebj1LpyzQlYoTMFpJyk+vRTYf+dt+f6ZCLTCgjrrZAMxC/CjuR8L3grdWqFk
tIvUpJyBGoNeTFofUtfGdCNrqlo1DiDQuS+yOwIDAQABAoIBAArpBvcMQ6hxysB3
VB6j8aqGnc0AZF2wojiRs1WtRhGpe3neIcOhVBdG/dbdbbZHYG+N8YVTTQroamQ3
V18OnyrDhKY3rjN1jLV8jBS2oaas09tHA5T62qhZChvC9BHDp7EYGNXZ035OoOoH
VnPslk7mcMrvm0J3E8cDsmPJgjP+4hpauDNbzg3pPB5r8G5Oo6czNevpeTAWtLx1
eIeqLQG0Dj7SNdtqoW95tpqvbySwS7OX4U18Kdzdt0w1jsdQ4KUpLN/M4PPqdLks
kdshqRalNDGGaCl3qS95FZ5vtb8f2F52NSexmeAHs3wv4GqsU4ZWP0LFOjkiJuej
UMOYuJECgYEA+XnOqX1fcPE4xOAj6tn+Pz7XpM/xKsfHYYvhRt1eIOzDQoaOBoAm
j2Doy4at7dJ8ZpEznH6iThuV566MQUDTLuQsit1JKsiUz+uNDIElGCFJEEJV4W8t
/2I3EKmcQ9YyTea/gcPUKkTf3yGf4zsvWyYIXMaKxuuGsF/KJ5YuGYkCgYEA2+Ao
miGYQXEnax+paagzDjZ+E+seAgUEkDTqP4KsGGGpE/DZ1NjqqTdksKvADzMm/ETp
Xabgp3PAsNViElnqFUDbSmH7o0aN8lxAXzGKi9nR0S/QqBOvoUYgmLKVQXvyrrql
imsFVbVOqOKdAm2NaEuq2a56uZWZ5uSi4pp98KMCgYEAplHawYxgEMABBOAZkfz7
T9bplWs8cTT3a8UBz2jN1E2Eb9mc3iCZlIUbLnT/h8oIPakYK4ERW9lwOyFXSDmb
kWt1dq7IlMlczLFRO8DtpgSq3TgcYUrp85Ta+TTahS9MIjtv+WdBD1OTk4KrQaa3
j6DvekqzQqLLDNMPxwyMd2ECgYBdJPmuSbsJRhkvzUrkYoe2ZqlEYN7mdh+3w2LU
otOqxHxu4SI2g+ns43V5TljfV4kVZ1ABB3e2GFgx0UNpiMOcW0xKl3WVdocde60d
ZoxNsGmtF3dqLGTDikS2yzhldE+ba6BDIhExfMwkyVNAOw6jdyKacsq0ocZ/xK2o
F1BwxwKBgQC9wUqNzt984lHYhQeGstxThJ+h3mUJv8YNNGkYCJ5osccXV//Lp5x1
fJUGCwTzjMTvO9EfnBJ1Yh2cEN1IUYqS/Mo4rr9+/VDSbc4wqMUpa4lHy7bvJvGR
DP2orU9tWxKmwB3Fe8JUnG7PO/4gYF9TulSa/VQV5l3iZatgjUc7MQ==
-----END RSA PRIVATE KEY-----

```



192.168.122.7:8080/credentials/store/system/domain/_/newCredentials
Jenkins
查找
admin
注销
Jenkins › 凭据 › 系统 › 全局凭据 (unrestricted) ›
返回到凭据域列表
添加凭据

类型 SSH Username with private key

范围 全局 (Jenkins, nodes, items, all child items, etc)

ID

描述 connection to gitlab-server

Username root

需要点击蓝色add按钮

Private Key ● Enter directly

Key

Enter New Secret Below

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA1kWl8Q7cOmpEJK0diHth2TQdbfLBovGluKYBVUcYi0f4R1DA
2FS3Pkvl2SAwl6s1FPivrA83vcwWJH8CaK06ctGvCWJ1TxMddilkq76DSK+4M8Xq
```

Passphrase

7.4 配置jenkins使用docker

验证系统中是否有jenkins用户

```
[root@jenkins-server ~]# grep jenkins /etc/passwd
jenkins:x:997:995:Jenkins Automation Server:/var/lib/jenkins:/bin/false
```

验证系统中是否有docker用户及用户组

```
[root@jenkins-server ~]# grep docker /etc/group
docker:x:993:
```

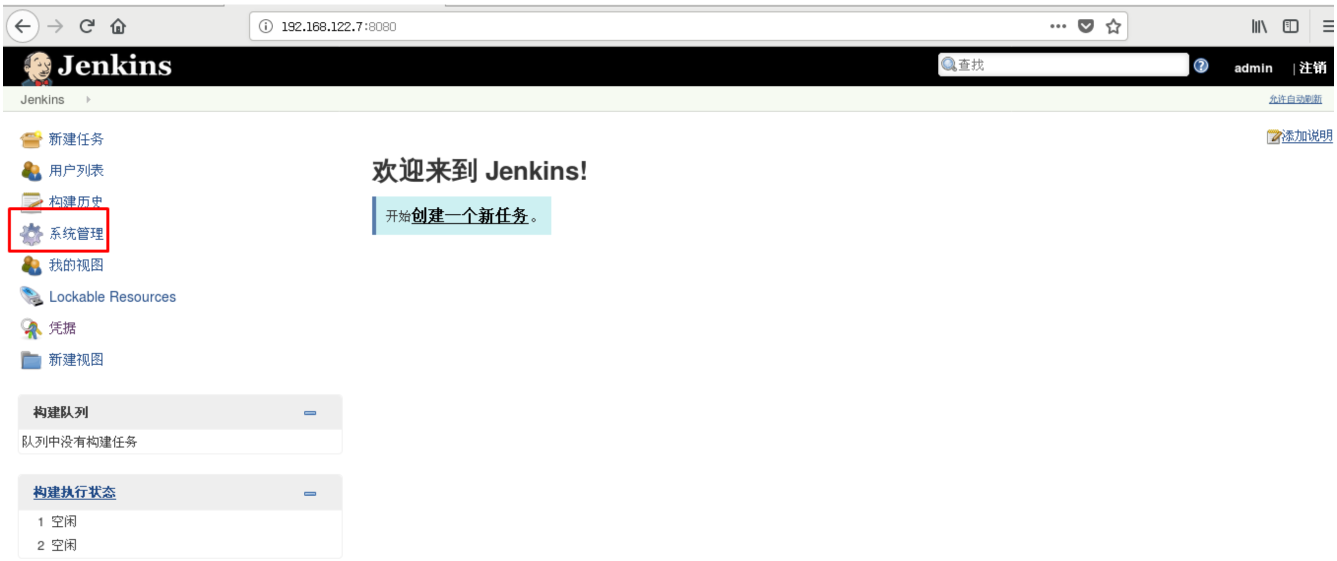
添加jenkins用户到docker用户组

```
[root@jenkins-server ~]# usermod -G docker jenkins
[root@jenkins-server ~]# grep docker /etc/group
docker:x:993:jenkins
```

7.5 jenkins-server添加插件

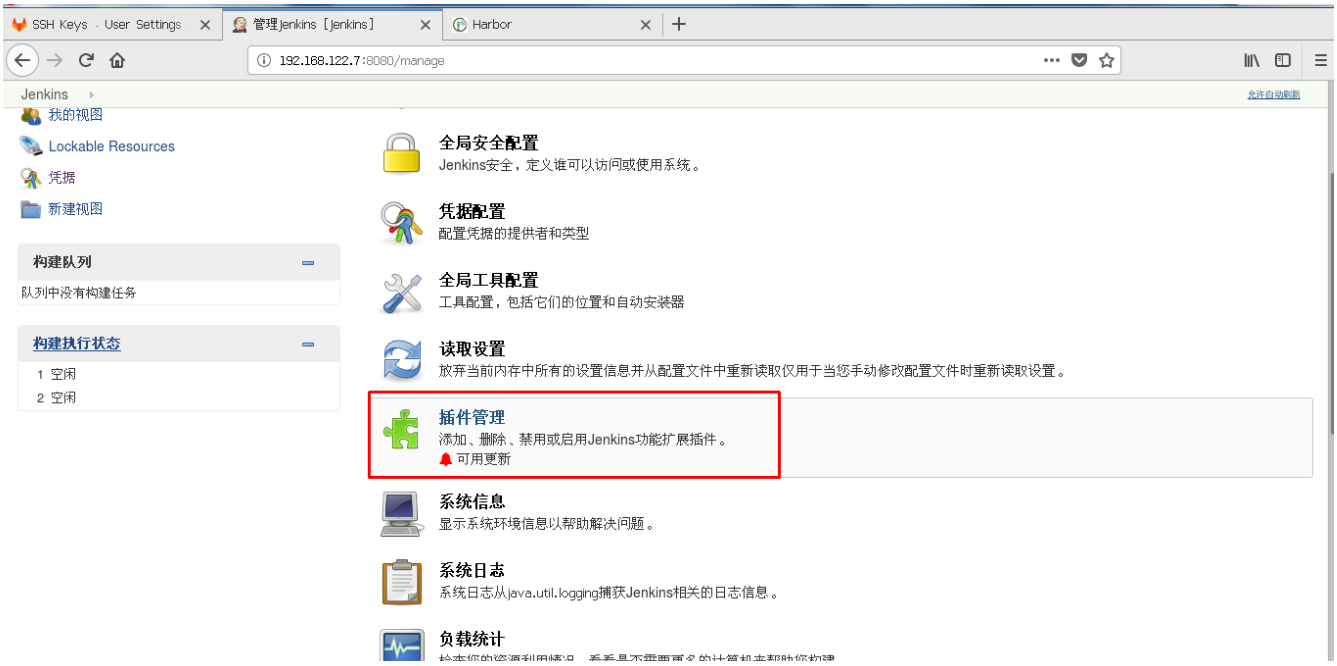
安装的插件有：

- ssh 用于jenkins-server对web-server进行操作
- git parameter 用于git版本提交进行参数构建
- gitlab 用于jenkins-server拉取项目
- gitlab hook 用于项目自动构建
- maven integration 用于编译



欢迎来到 Jenkins!

开始创建一个新任务。



全局安全配置
Jenkins安全，定义谁可以访问或使用系统。

凭据配置
配置凭据的提供者和类型

全局工具配置
工具配置，包括它们的位置和自动安装器

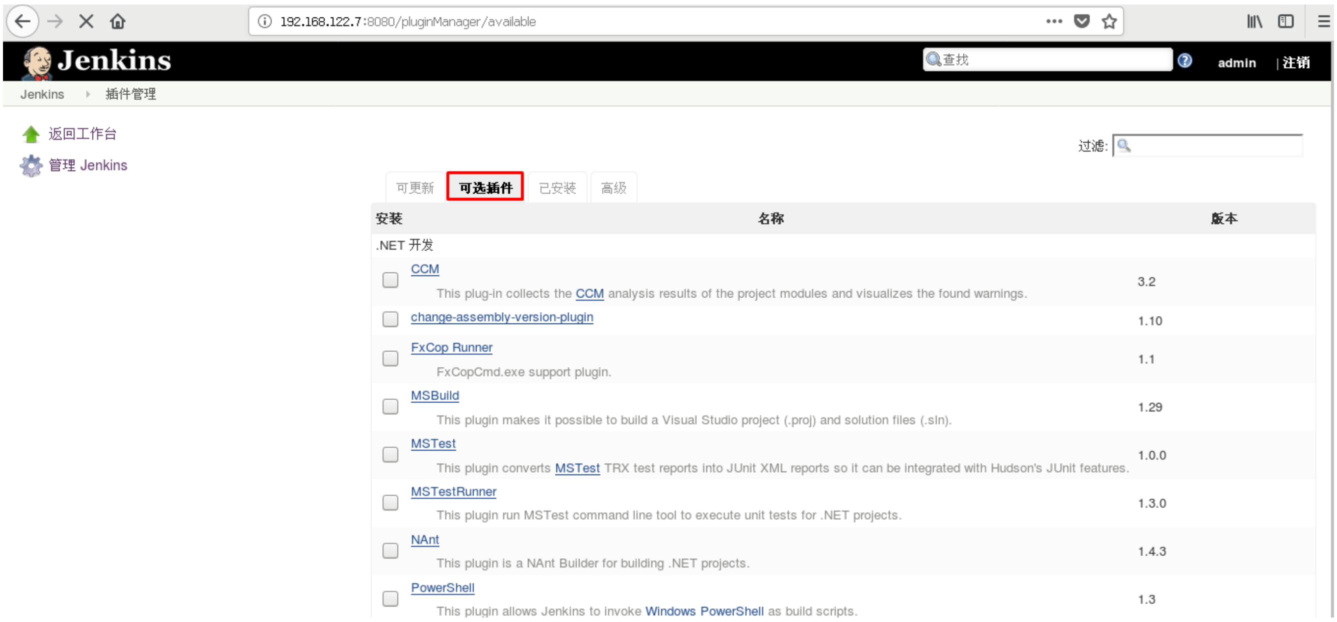
读取设置
放弃当前内存中所有的设置信息并从配置文件中重新读取仅用于当您手动修改配置文件时重新读取设置。

插件管理
添加、删除、禁用或启用Jenkins功能扩展插件。
可用更新

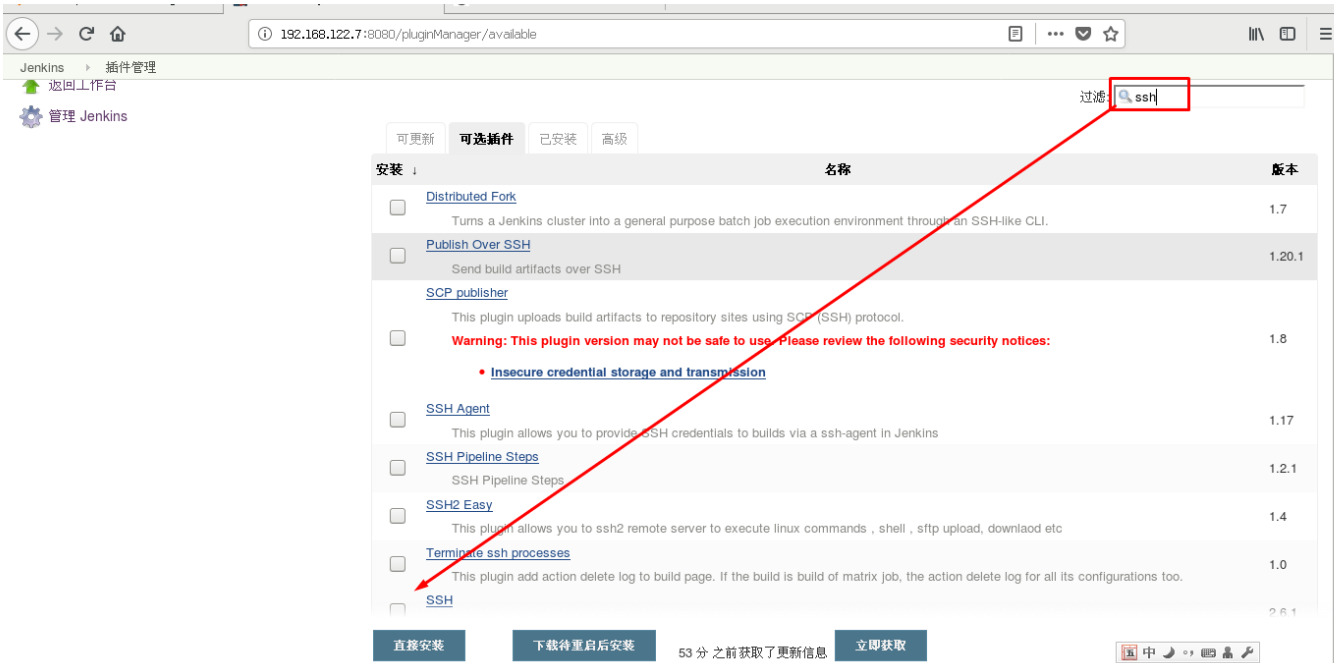
系统信息
显示系统环境信息以帮助解决问题。

系统日志
系统日志从java.util.logging捕获Jenkins相关的日志信息。

负载统计

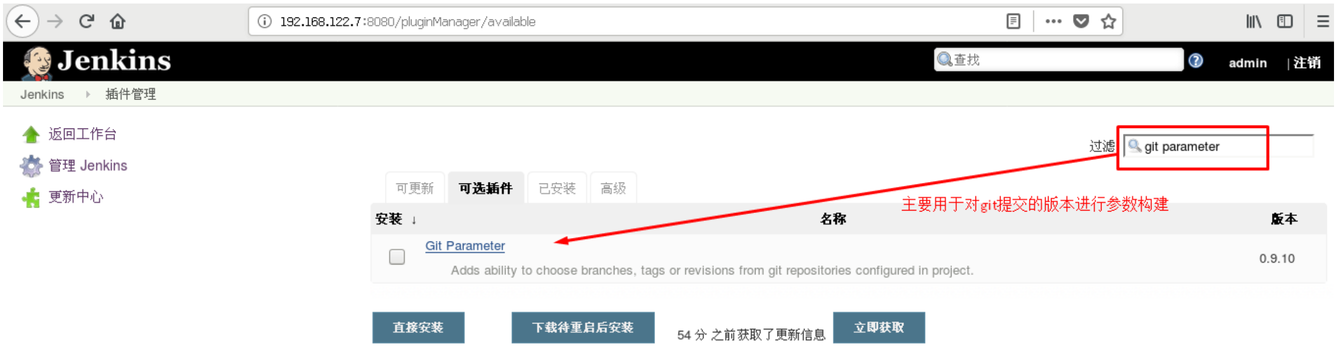


安装	名称	版本
	.NET 开发	
□	CCM This plug-in collects the CCM analysis results of the project modules and visualizes the found warnings.	3.2
□	change-assembly-version-plugin	1.10
□	FxCop Runner FxCopCmd.exe support plugin.	1.1
□	MSBuild This plugin makes it possible to build a Visual Studio project (.proj) and solution files (.sln).	1.29
□	MSTest This plugin converts MSTest TRX test reports into JUnit XML reports so it can be integrated with Hudson's JUnit features.	1.0.0
□	MSTestRunner This plugin run MSTest command line tool to execute unit tests for .NET projects.	1.3.0
□	NAnt This plugin is a NAnt Builder for building .NET projects.	1.4.3
□	PowerShell This plugin allows Jenkins to invoke Windows PowerShell as build scripts.	1.3



安装 ↓	名称	版本
□	Distributed Fork Turns a Jenkins cluster into a general purpose batch job execution environment through an SSH-like CLI.	1.7
□	Publish Over SSH Send build artifacts over SSH	1.20.1
□	SCP publisher This plugin uploads build artifacts to repository sites using SCP (SSH) protocol. **Warning: This plugin version may not be safe to use. Please review the following security notices:** • **Insecure credential storage and transmission**	1.8
□	SSH Agent This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins	1.17
□	SSH Pipeline Steps SSH Pipeline Steps	1.2.1
□	SSH2 Easy This plugin allows you to ssh2 remote server to execute linux commands , shell , sftp upload, downlaod etc	1.4
□	Terminate ssh processes This plugin add action delete log to build page. If the build is build of matrix job, the action delete log for all its configurations too.	1.0
□	SSH	2.6.1

53 分 之前获取了更新信息



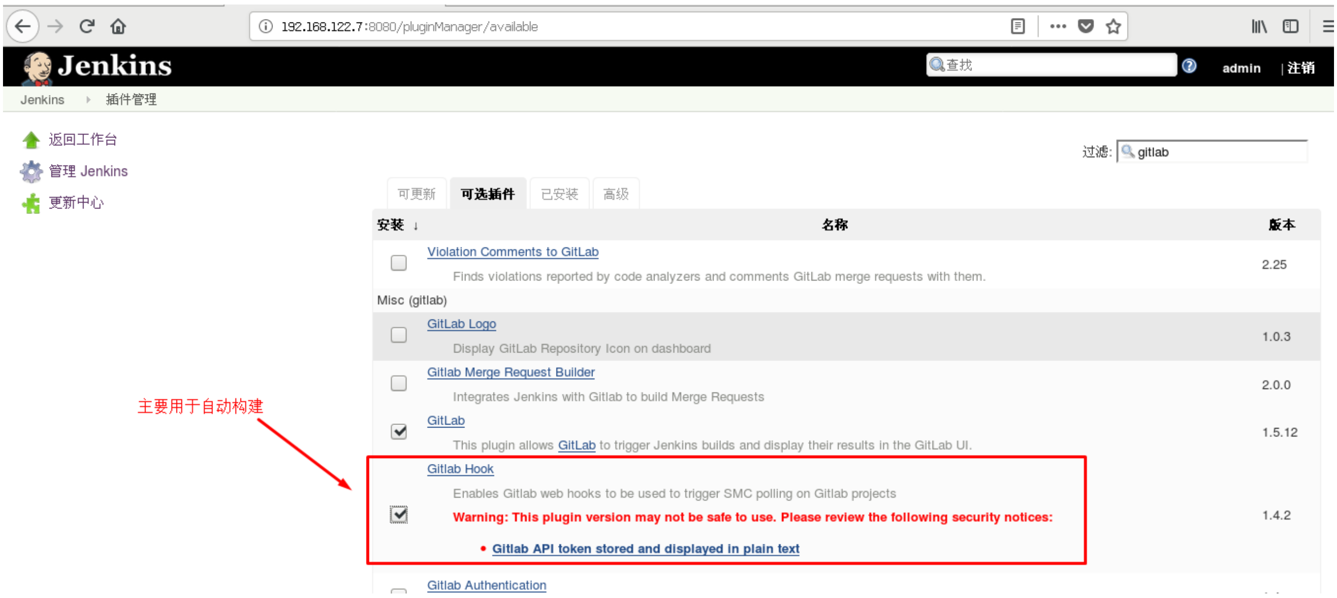
192.168.122.7:8080/pluginManager/available

过滤 git parameter

主要用于对git提交的版本进行参数构建

安装 ↓	名称	版本
□	Git Parameter Adds ability to choose branches, tags or revisions from git repositories configured in project.	0.9.10

54 分 之前获取了更新信息

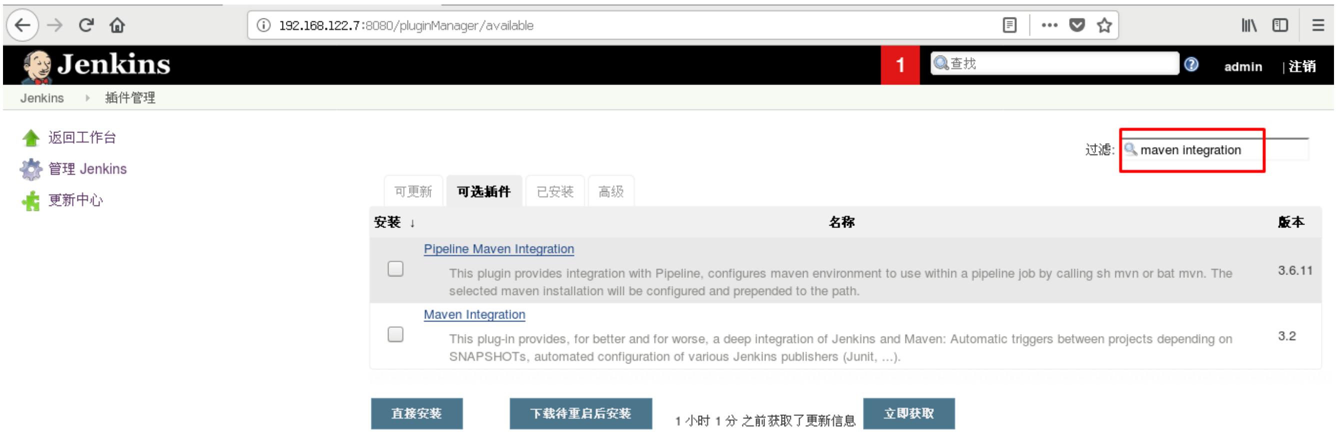


192.168.122.7:8080/pluginManager/available

过滤: gitlab

主要用于自动构建

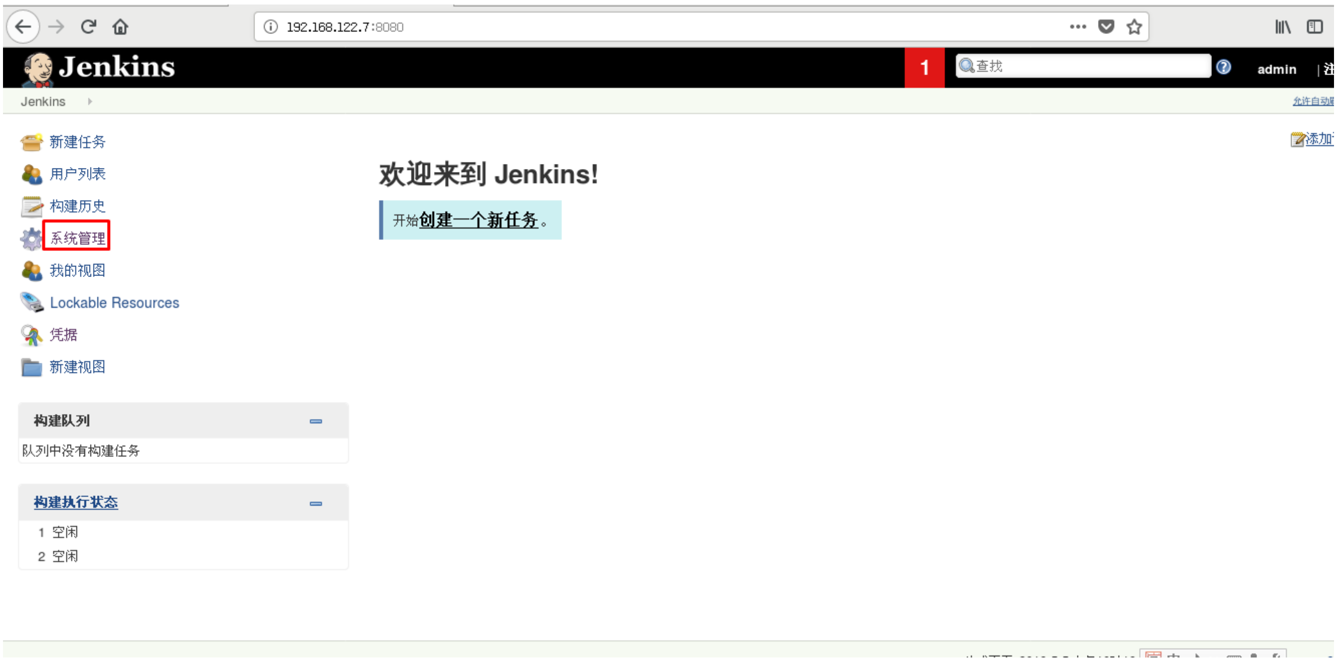
安装 ↓	名称	版本
□	Violation Comments to GitLab Finds violations reported by code analyzers and comments GitLab merge requests with them.	2.25
Misc (gitlab)		
□	GitLab Logo Display GitLab Repository Icon on dashboard	1.0.3
□	Gitlab Merge Request Builder Integrates Jenkins with Gitlab to build Merge Requests	2.0.0
☑	GitLab This plugin allows GitLab to trigger Jenkins builds and display their results in the GitLab UI.	1.5.12
☑	Gitlab Hook Enables Gitlab web hooks to be used to trigger SMC polling on Gitlab projects **Warning: This plugin version may not be safe to use. Please review the following security notices:** • **Gitlab API token stored and displayed in plain text**	1.4.2
□	Gitlab Authentication	

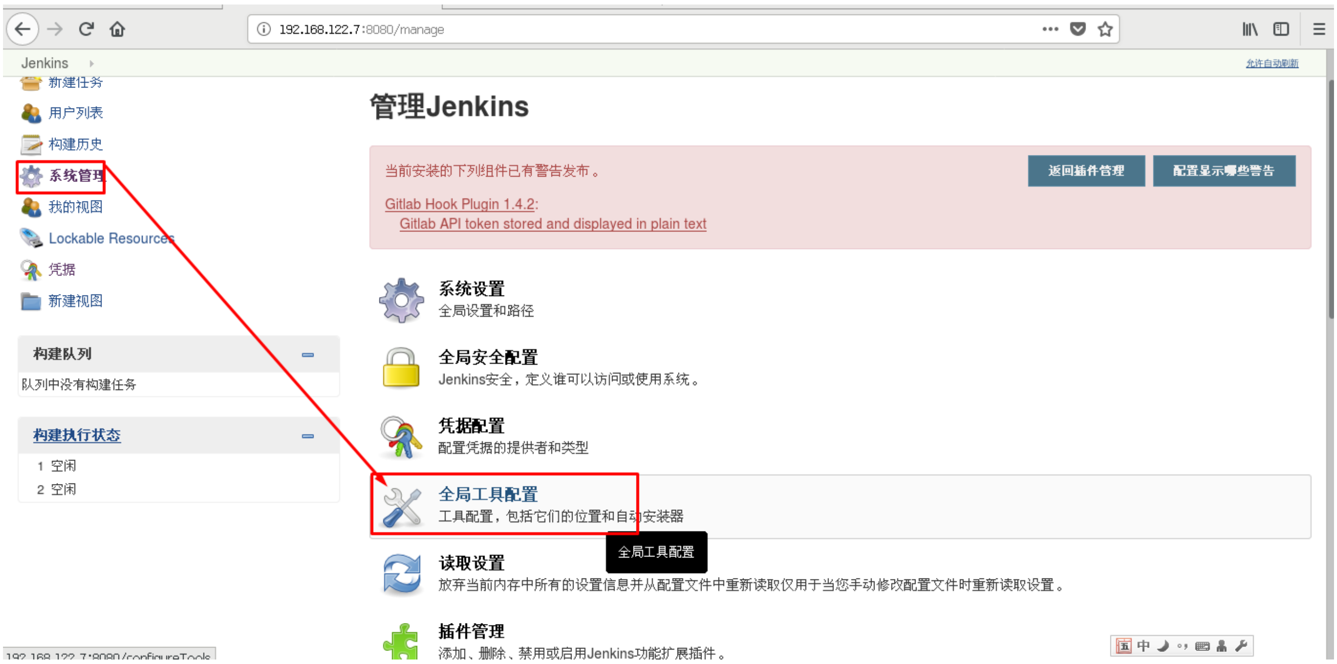


安装 ↓	名称	版本
□	Pipeline Maven Integration This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path.	3.6.11
□	Maven Integration This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTs, automated configuration of various Jenkins publishers (Junit, ...).	3.2

7.6 jenkins全局配置

7.6.1 jenkins全局工具配置





管理Jenkins

当前安装的下列组件已有警告发布。

Gitlab Hook Plugin 1.4.2:
Gitlab API token stored and displayed in plain text

系统设置
全局设置和路径

全局安全配置
Jenkins安全，定义谁可以访问或使用系统。

凭据配置
配置凭据的提供者和类型

全局工具配置
工具配置，包括它们的位置和自动安装器

读取设置
放弃当前内存中所有的设置信息并从配置文件中重新读取仅用于当您手动修改配置文件时重新读取设置。

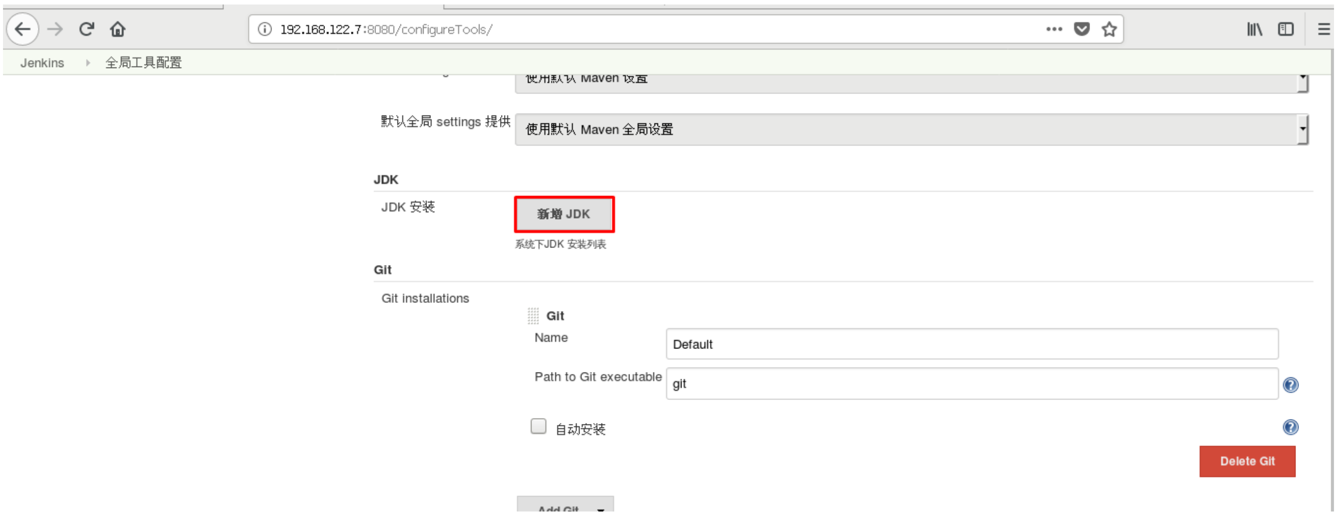
插件管理
添加、删除、禁用或启用Jenkins功能扩展插件。

JDK配置

确认系统中jdk目录

```
[root@jenkins-server ~]# java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)
[root@jenkins-server ~]# echo $JAVA_HOME
/usr/local/jdk
```

添加jdk



使用默认 Maven 设置

默认全局 settings 提供: 使用默认 Maven 全局设置

JDK

JDK 安装

Git

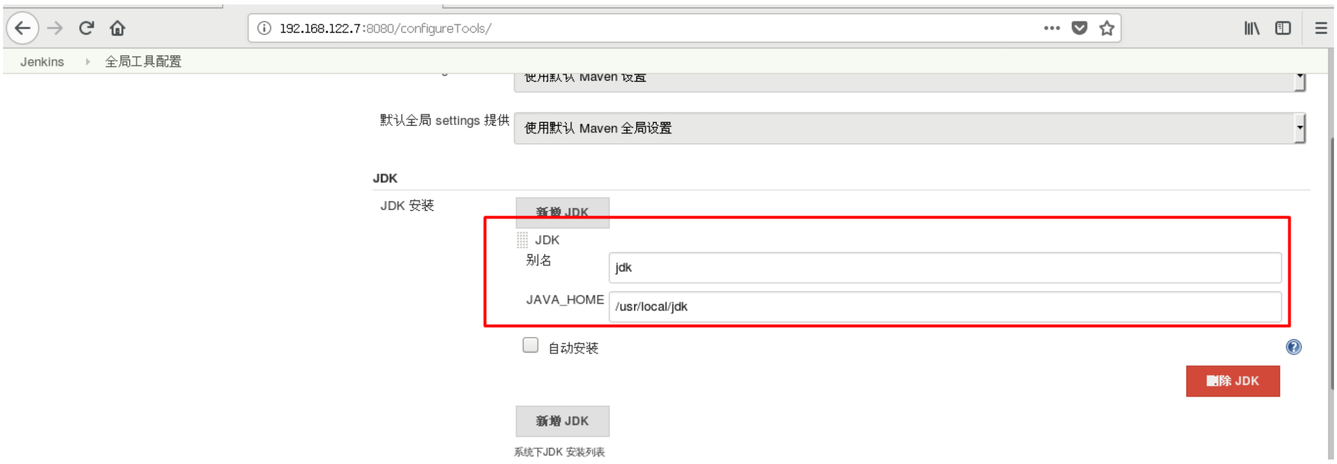
Git installations

Git

Name: Default

Path to Git executable: git

□ 自动安装



使用默认 Maven 设置

默认全局 settings 提供: 使用默认 Maven 全局设置

JDK

JDK 安装

JDK

别名: jdk

JAVA_HOME: /usr/local/jdk

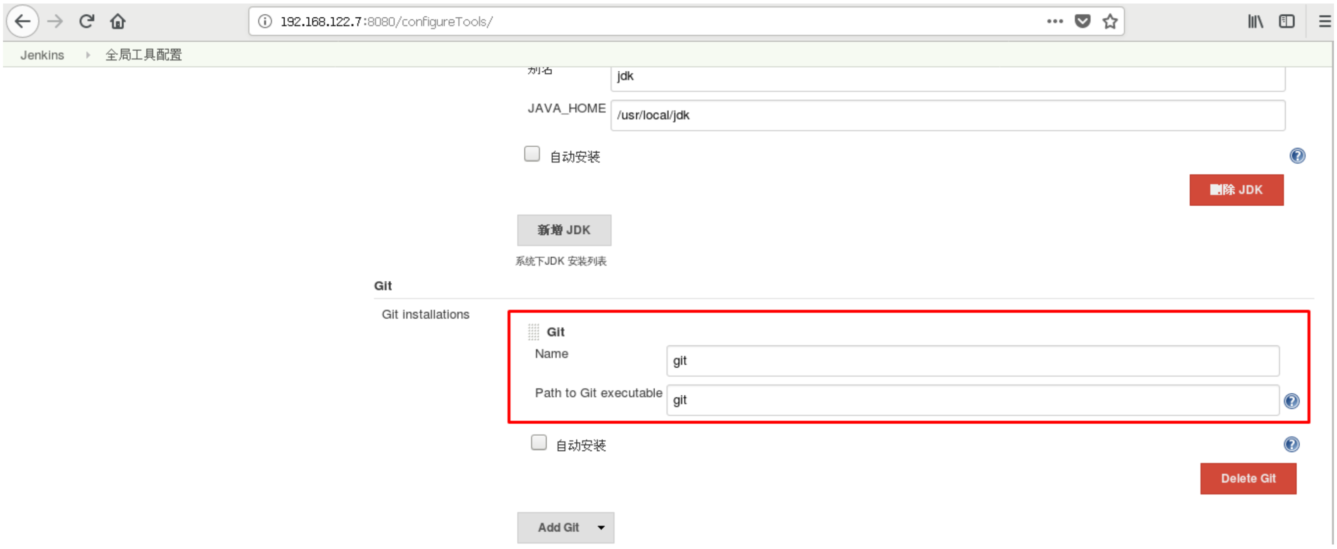
□ 自动安装

Git配置

确认系统中git是否安装

```
[root@jenkins-server ~]# git version
git version 1.8.3.1
```

添加git



192.168.122.7:8080/configureTools/

Jenkins › 全局工具配置

别名 jdk

JAVA_HOME /usr/local/jdk

□ 自动安装

系统下JDK 安装列表

Git

Git installations

Git

Name git

Path to Git executable git

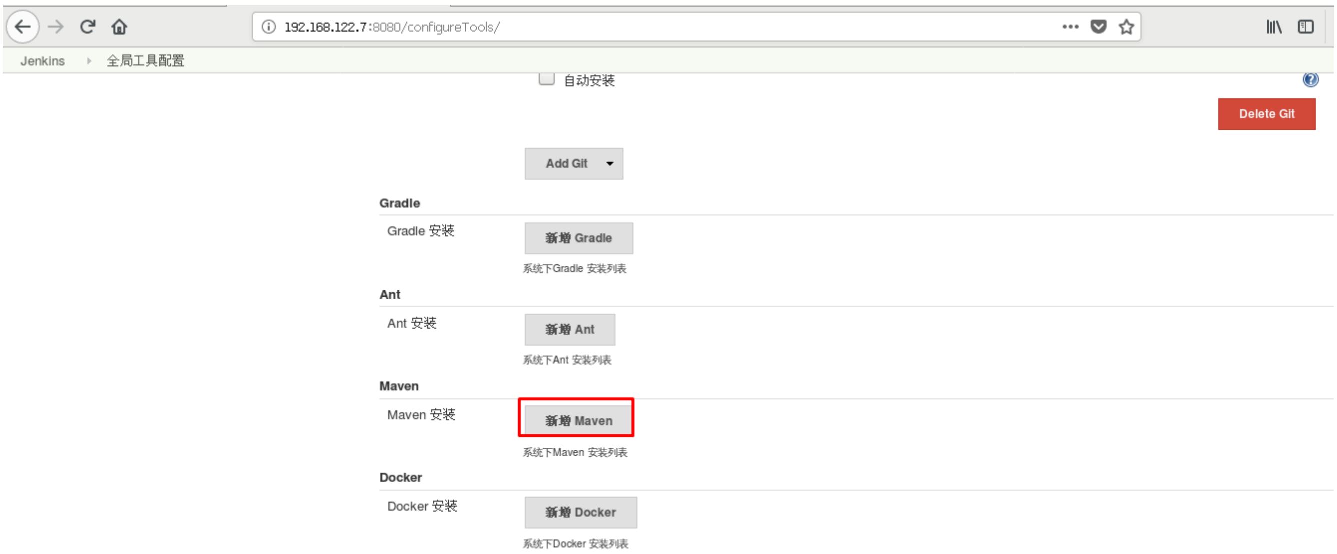
□ 自动安装

Maven配置

确认maven是不安装

```
[root@jenkins-server ~]# mvn -v
Apache Maven 3.6.1 (d66c9c0b3152b2e69ee9bac180bb8fcc8e6af555; 2019-04-05T03:00:29+08:00)
Maven home: /usr/local/maven
Java version: 1.8.0_191, vendor: Oracle Corporation, runtime: /usr/local/jdk/jre
Default locale: zh_CN, platform encoding: UTF-8
OS name: "linux", version: "3.10.0-957.el7.x86_64", arch: "amd64", family: "unix"

```

添加maven



192.168.122.7:8080/configureTools/

Jenkins › 全局工具配置

□ 自动安装

Gradle

Gradle 安装

系统下Gradle 安装列表

Ant

Ant 安装

系统下Ant 安装列表

Maven

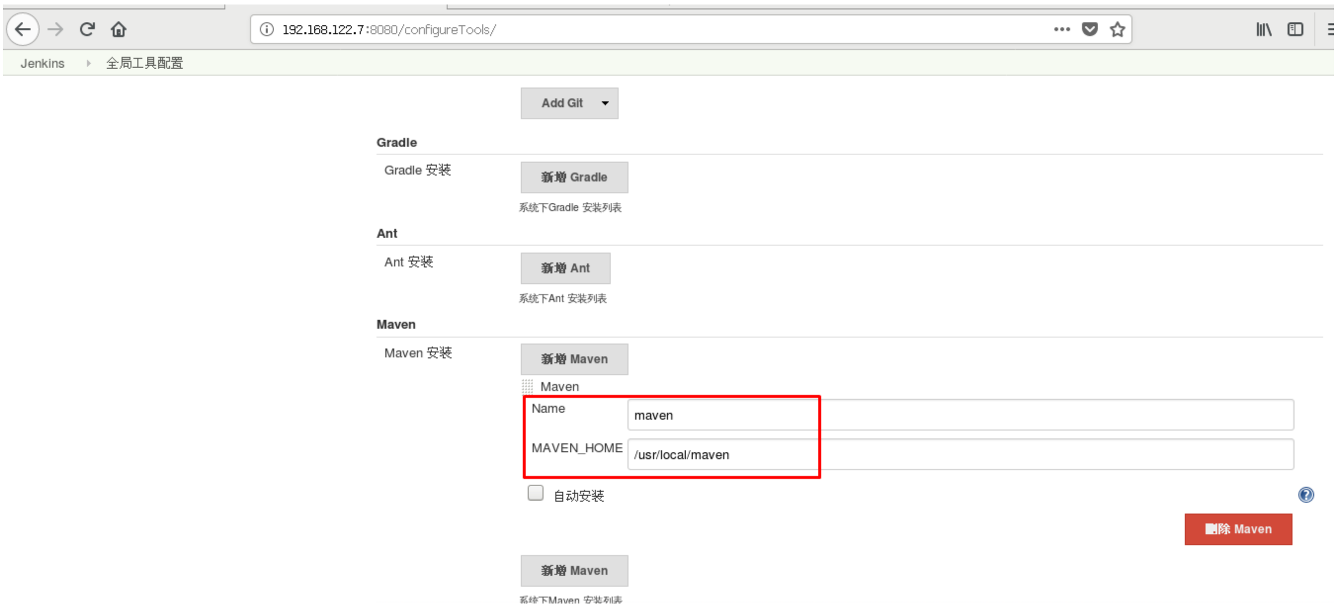
Maven 安装

系统下Maven 安装列表

Docker

Docker 安装

系统下Docker 安装列表



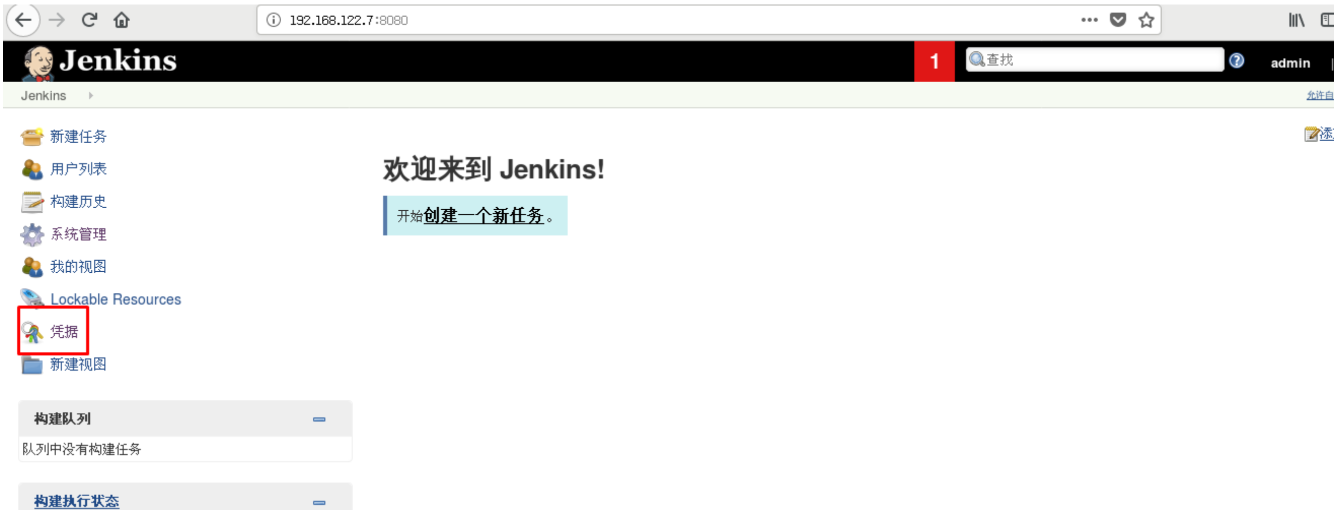
Name: maven

MAVEN_HOME: /usr/local/maven

7.6.2 jenkins系统配置

主要配置ssh插件，用于jenkins操作web-server,让web-server执行命令。

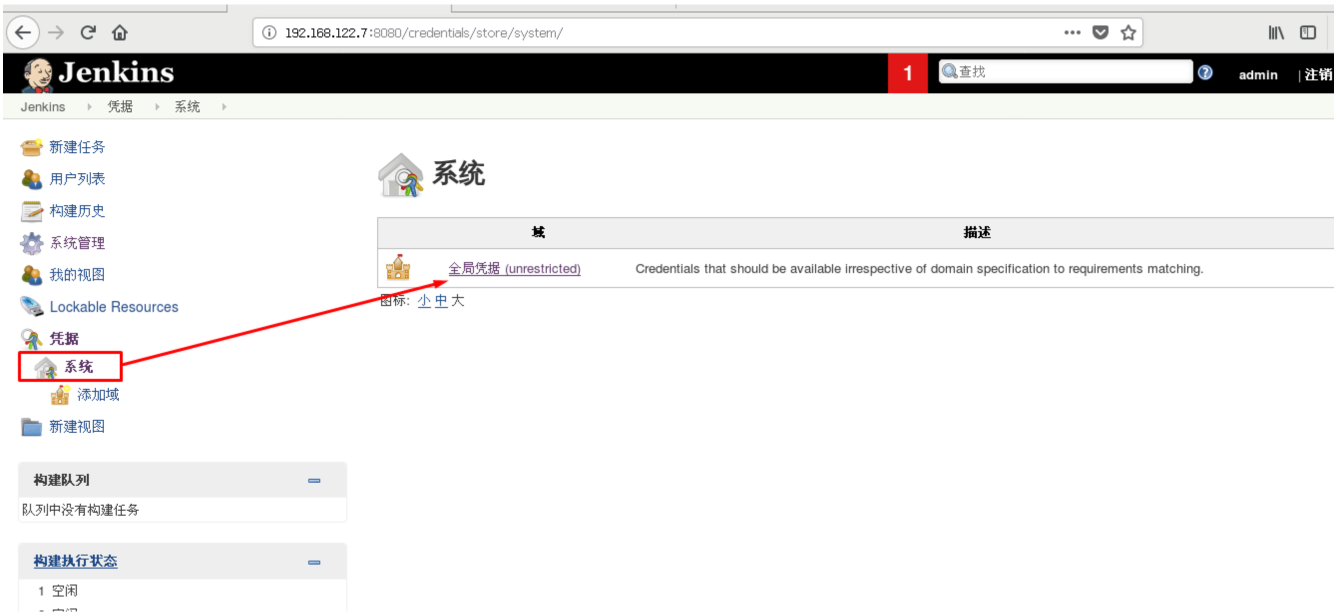
7.6.2.1 添加凭据



欢迎来到 Jenkins!

开始创建一个新任务。

队列中没有构建任务



192.168.122.7:8080/credentials/store/system/

Jenkins 1 查找 admin | 注销

Jenkins › 凭据 › 系统 ›

- 新建任务
- 用户列表
- 构建历史
- 系统管理
- 我的视图
- Lockable Resources
- 凭据
  - 系统
  - 添加域
- 新建视图

构建队列

队列中没有构建任务

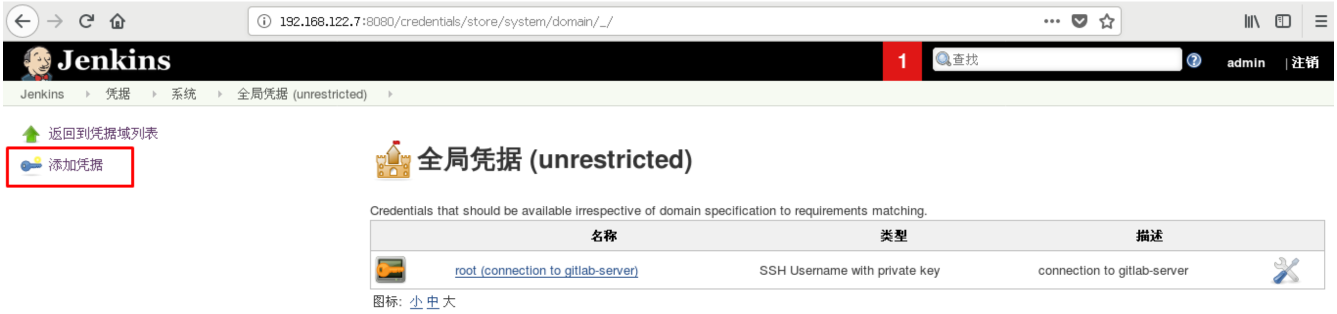
构建执行状态

1 空闲

系统

域	描述
全局凭据 (unrestricted)	Credentials that should be available irrespective of domain specification to requirements matching.

图标: 小 中 大



192.168.122.7:8080/credentials/store/system/domain/_/

Jenkins 1 查找 admin | 注销

Jenkins › 凭据 › 系统 › 全局凭据 (unrestricted) ›

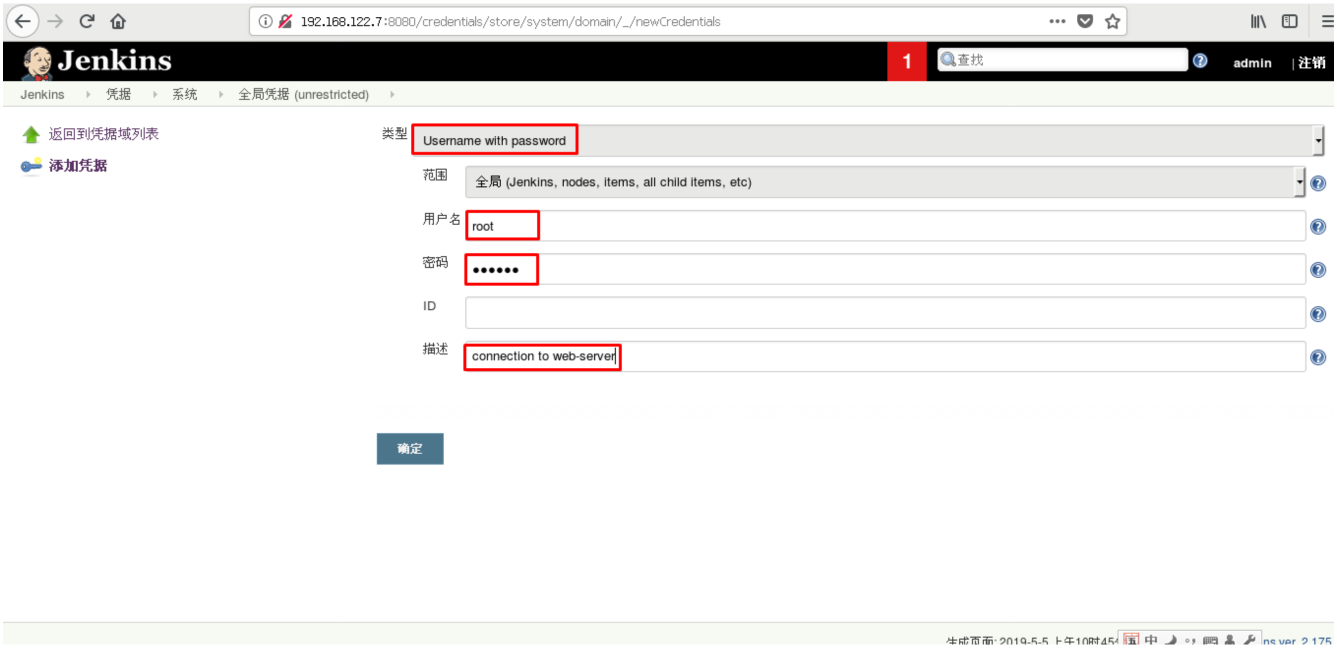
- 返回到凭据域列表
- 添加凭据

全局凭据 (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

名称	类型	描述
root (connection to gitlab-server)	SSH Username with private key	connection to gitlab-server

图标: 小 中 大



192.168.122.7:8080/credentials/store/system/domain/_/newCredentials

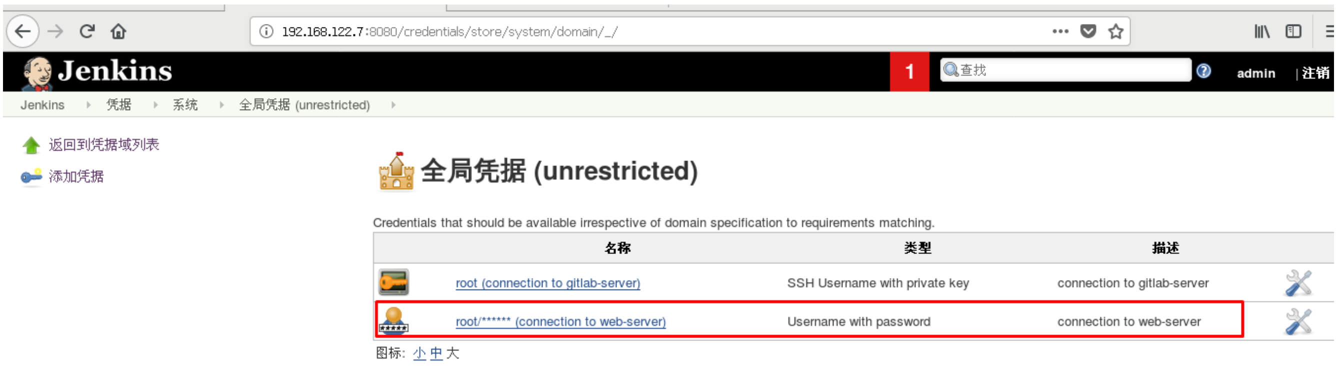
Jenkins › 凭据 › 系统 › 全局凭据 (unrestricted) ›

返回到凭据域列表

添加凭据

类型	Username with password
范围	全局 (Jenkins, nodes, items, all child items, etc)
用户名	root
密码	••••••
ID	
描述	connection to web-server

确定



192.168.122.7:8080/credentials/store/system/domain/_/

Jenkins › 凭据 › 系统 › 全局凭据 (unrestricted) ›

返回到凭据域列表

添加凭据

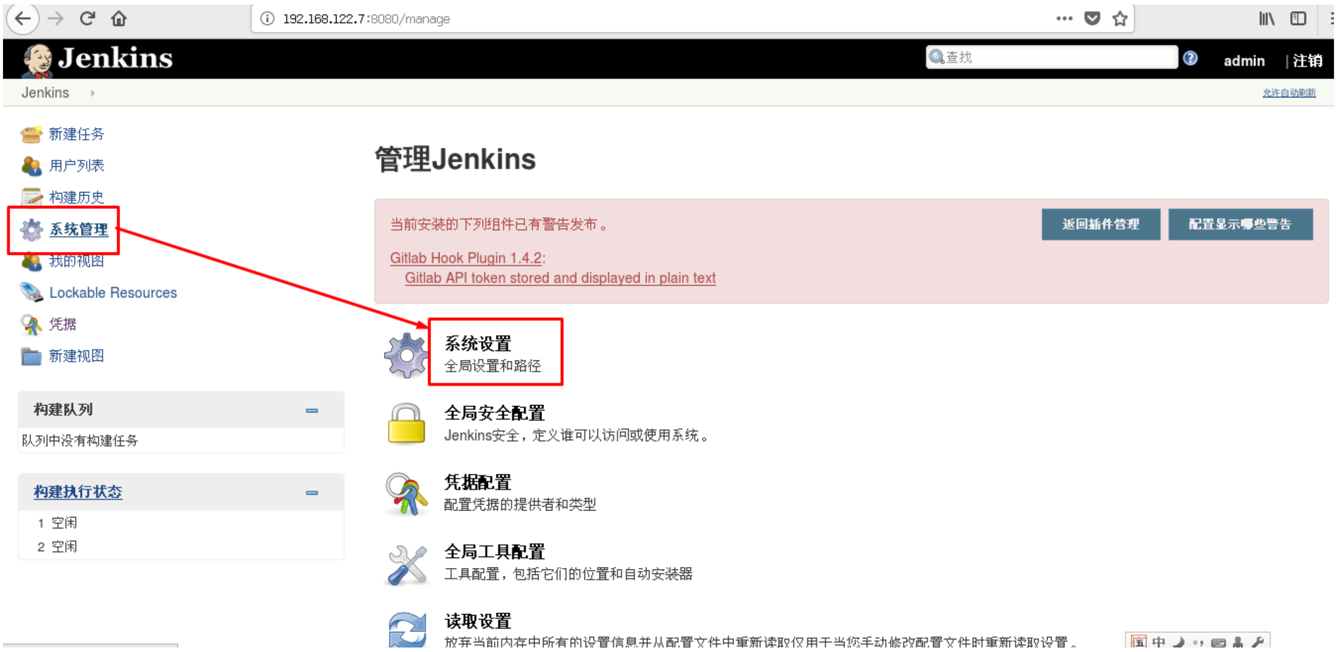
全局凭据 (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

名称	类型	描述
root (connection to gitlab-server)	SSH Username with private key	connection to gitlab-server
root/****** (connection to web-server)	Username with password	connection to web-server

图标: 小 中 大

7.6.2.2 配置ssh插件



192.168.122.7:8080/manage

Jenkins

查找

admin | 注销

Jenkins

允许自动刷新

- 新建任务
- 用户列表
- 构建历史
- 系统管理
- 我的视图
- Lockable Resources
- 凭据
- 新建视图

构建队列

队列中没有构建任务

构建执行状态

1 空闲

2 空闲

管理Jenkins

当前安装的下列组件已有警告发布。

返回插件管理 | 配置显示哪些警告

Gitlab Hook Plugin 1.4.2:

Gitlab API token stored and displayed in plain text

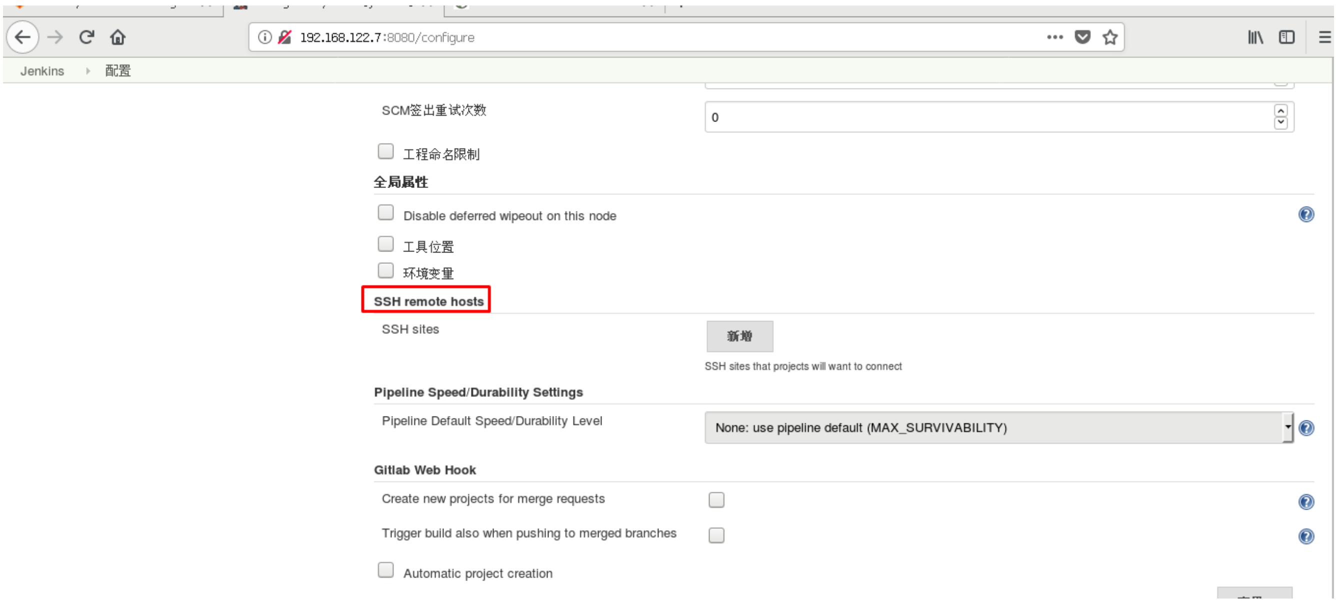
系统设置
全局设置和路径

全局安全配置
Jenkins安全，定义谁可以访问或使用系统。

凭据配置
配置凭据的提供者和类型

全局工具配置
工具配置，包括它们的位置和自动安装器

读取设置
放弃当前内存中所有的设置信息并从配置文件中重新读取仅用于当您手动修改配置文件时重新读取设置。



192.168.122.7:8080/configure

Jenkins › 配置

SCM签出重试次数: 0

□ 工程命名限制

全局属性

□ Disable deferred wipeout on this node

□ 工具位置

□ 环境变量

SSH remote hosts

SSH sites: 新增

SSH sites that projects will want to connect

Pipeline Speed/Durability Settings

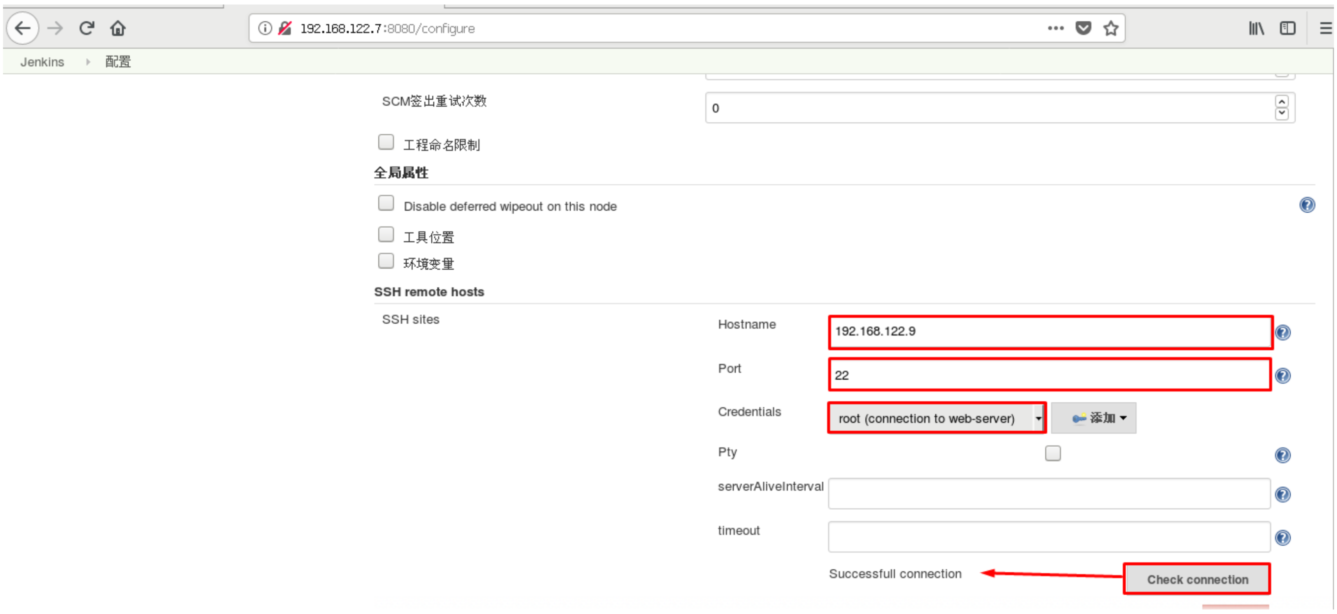
Pipeline Default Speed/Durability Level: None: use pipeline default (MAX_SURVIVABILITY)

Gitlab Web Hook

Create new projects for merge requests □

Trigger build also when pushing to merged branches □

□ Automatic project creation



192.168.122.7:8080/configure

Jenkins › 配置

SCM签出重试次数: 0

□ 工程命名限制

全局属性

□ Disable deferred wipeout on this node

□ 工具位置

□ 环境变量

SSH remote hosts

SSH sites

Hostname: 192.168.122.9

Port: 22

Credentials: root (connection to web-server)

Pty: □

serverAliveInterval:

timeout:

Successfull connection

八、项目发布

8.1 项目代码获取

```
[root@dev ~]# git clone --recurse-submodules https://gitee.com/dl88250/solo.git
正克隆到 'solo'...
remote: Enumerating objects: 43707, done.
remote: Counting objects: 100% (43707/43707), done.
remote: Compressing objects: 100% (18606/18606), done.
remote: Total 43707 (delta 24446), reused 38212 (delta 19553)
接收对象中: 100% (43707/43707), 88.55 MiB | 621.00 KiB/s, done.
处理 delta 中: 100% (24446/24446), done.
子模组 'src/main/webapp/skins' (https://github.com/b3log/solo-skins) 已为路径 'src/main/webapp/skins' 注册
正克隆到 'src/main/webapp/skins'...
remote: Enumerating objects: 1110, done.
remote: Counting objects: 100% (1110/1110), done.
remote: Compressing objects: 100% (684/684), done.
remote: Total 11660 (delta 785), reused 709 (delta 425), pack-reused 10550
接收对象中: 100% (11660/11660), 23.56 MiB | 556.00 KiB/s, done.
处理 delta 中: 100% (8743/8743), done.
子模组路径 'src/main/webapp/skins'：检出 '895d3cfa4c522932070377f8d19f4eae559d2de1'

```

8.2 项目代码修改

主用修改项目如何连接数据库

```
[root@dev ~]# ls
anaconda-ks.cfg  solo
[root@dev ~]# cd solo
[root@dev solo]# ls
CHANGE_LOGS.html  gulpfile.js  package.json       pom.xml    scripts
Dockerfile        LICENSE      package-lock.json  README.md  src
[root@dev solo]# cd src
[root@dev src]# ls
main  test
[root@dev src]# cd main/
[root@dev main]# pwd
/root/solo/src/main
[root@dev main]# ls
java  resources  webapp
[root@dev main]# cd resources/
[root@dev resources]# pwd
/root/solo/src/main/resources
[root@dev resources]# ls
docker                 lang_zh_CN.properties  log4j.properties  solo.properties
etc                    latke.properties       opensearch.xml
lang_en_US.properties  local.properties       repository.json

[root@dev resources]# cat local.properties
...
#### MySQL runtime ####
runtimeDatabase=MYSQL
jdbc.username=root
jdbc.password=123456
jdbc.driver=com.mysql.cj.jdbc.Driver
jdbc.URL=jdbc:mysql://192.168.122.9:3306/solo?useUnicode=yes&characterEncoding=UTF-8&useSSL=false&serverTimezone=UTC

...

```

8.3 安装项目数据库

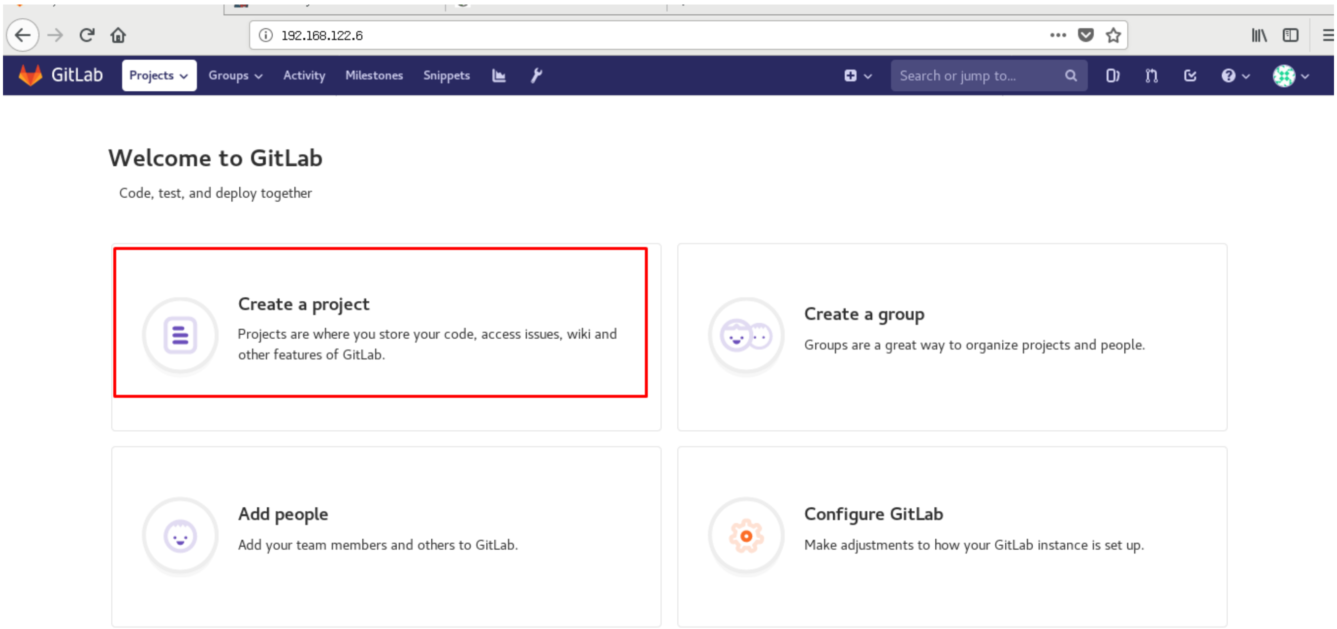
在web-server安装

```
[root@web-server ~]# systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/lib/systemd/system/mariadb.service.
[root@web-server ~]# systemctl start mariadb
```

```

[root@web-server ~]# mysqladmin -uroot password "123456"

[root@web-server ~]# mysql -uroot -p123456

MariaDB [(none)]> create database if not exists solo default charset utf8 collate utf8_general_ci;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| solo               |
| test               |
+--------------------+
5 rows in set (0.00 sec)

MariaDB [(none)]> grant all on solo.* to 'root'@'%' identified by "123456";


```

8.4 项目代码上传到gitlab-server

8.4.1 创建项目仓库



Welcome to GitLab

Code, test, and deploy together

Create a project

Projects are where you store your code, access issues, wiki and other features of GitLab.

Create a group

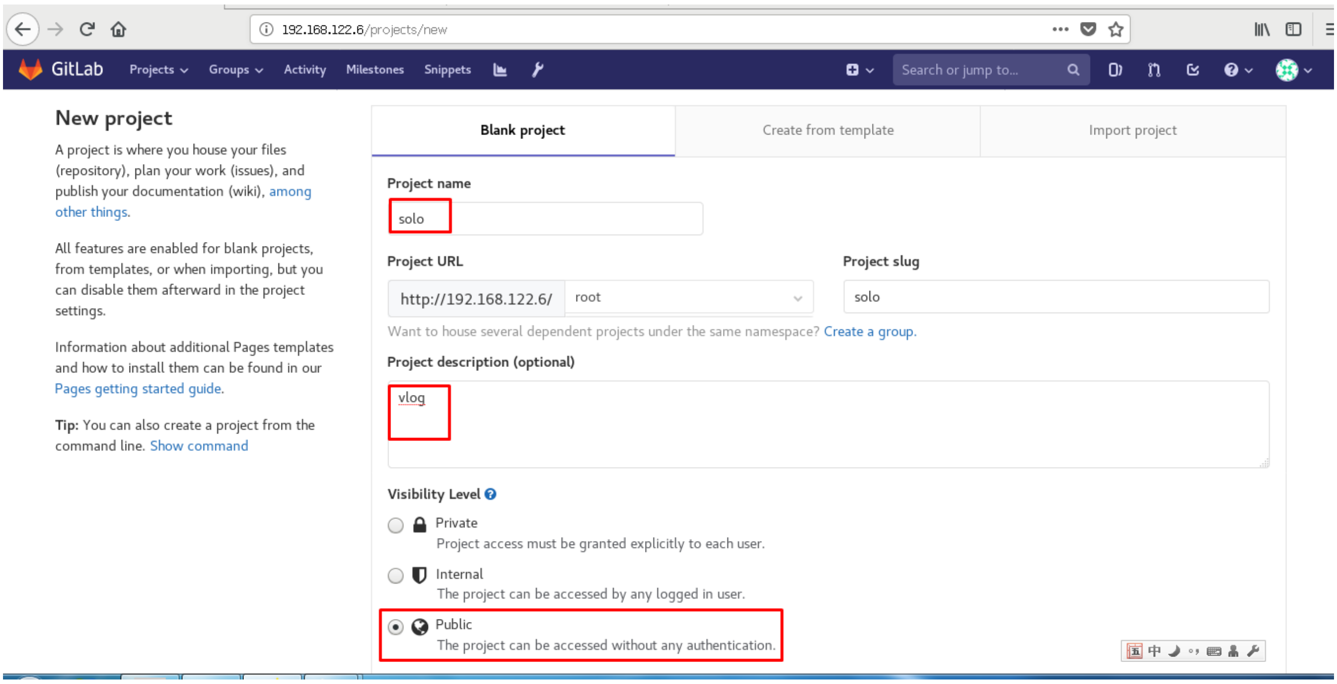
Groups are a great way to organize projects and people.

Add people

Add your team members and others to GitLab.

Configure GitLab

Make adjustments to how your GitLab instance is set up.



New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), among other things.

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

Information about additional Pages templates and how to install them can be found in our Pages getting started guide.

Tip: You can also create a project from the command line. Show command

Project name: solo

Project URL: http://192.168.122.6/ root

Project slug: solo

Want to house several dependent projects under the same namespace? Create a group.

Project description (optional): vlog

Visibility Level

○ Private
Project access must be granted explicitly to each user.

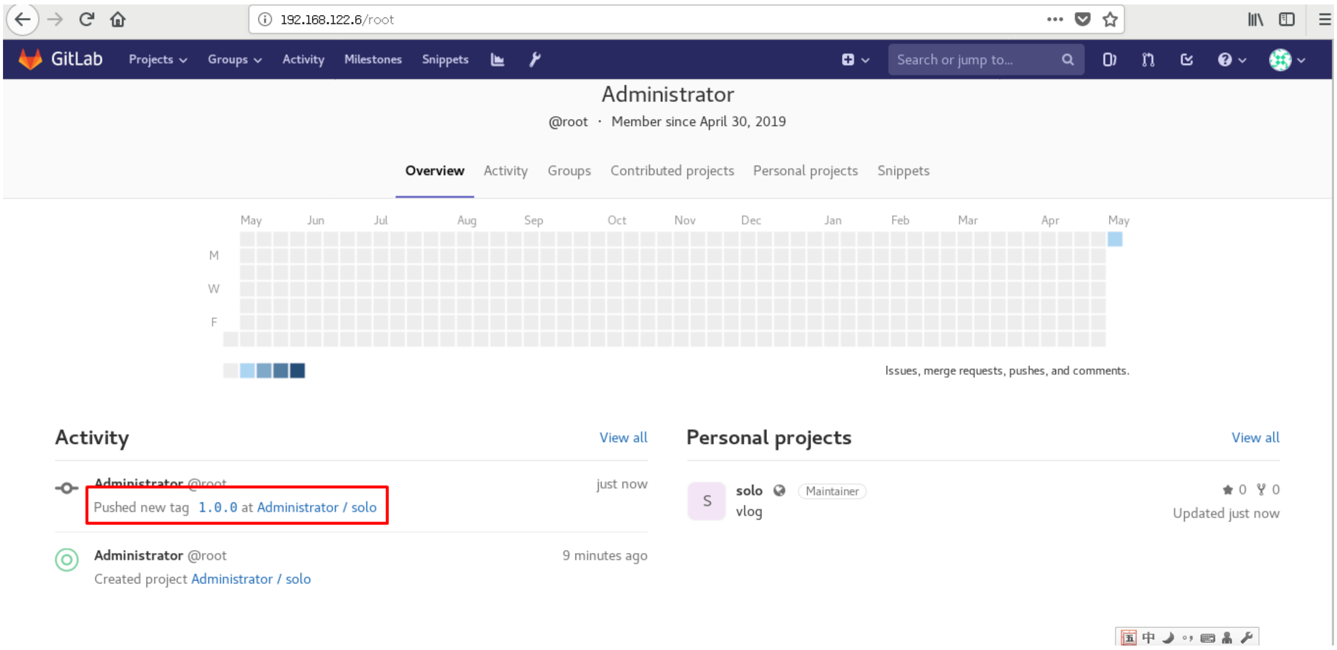
○ Internal
The project can be accessed by any logged in user.

● Public
The project can be accessed without any authentication.

8.4.2 上传项目代码

```
[root@dev ~]# git config --global user.name "dev"
[root@dev ~]# git config --global user.email "dev@aiops.net.cn"

[root@dev solo]# git remote remove origin

[root@dev solo]# git remote add origin git@192.168.122.6:root/solo.git

[root@dev solo]# git add -A .

[root@dev solo]# git commit -m "new"
[master c644bd4] new
 1 file changed, 1 insertion(+), 1 deletion(-)

[root@dev solo]# git tag 1.0.0

[root@dev solo]# git push origin 1.0.0
The authenticity of host '192.168.122.6 (192.168.122.6)' can't be established.
ECDSA key fingerprint is SHA256:b0Dbv+011dgg4r62fTbQsl4KdZ4dtrXMyY2pOWo+fws.
ECDSA key fingerprint is MD5:53:89:e0:9a:95:05:ee:54:08:7c:43:62:2e:1e:ec:da.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.122.6' (ECDSA) to the list of known hosts.
Counting objects: 43707, done.
Compressing objects: 100% (13713/13713), done.
Writing objects: 100% (43707/43707), 88.55 MiB | 19.88 MiB/s, done.
Total 43707 (delta 24446), reused 43707 (delta 24446)
remote: Resolving deltas: 100% (24446/24446), done.
To git@192.168.122.6:root/solo.git
 * [new tag]         1.0.0 -> 1.0.0


```

gitlab-server web页面进行验证



Administrator

@root · Member since April 30, 2019

Issues, merge requests, pushes, and comments.

Activity

Administrator @root just now
Pushed new tag 1.0.0 at Administrator / solo

Administrator @root 9 minutes ago
Created project Administrator / solo

Personal projects

solo Maintainer ★ 0 0
vlog
Updated just now

8.5 创建项目运行的基础应用镜像

主要是tomcat容器应用镜像

- 使用Dockerfile
- 在jenkinks-server主机

8.5.1 创建项目目录

```
[root@jenkins-server ~]# mkdir tomcatdir

[root@jenkins-server ~]# cd tomcatdir/
```

8.5.2 生成Dockerfile

```
[root@jenkins-server tomcatdir]# cat Dockerfile
FROM centos:latest
MAINTAINER "aiops<admin@aiops.net.cn>"

```

```
ENV VERSION=8.5.38
ENV JAVA_HOME=/usr/local/jdk

RUN yum -y install wget

RUN wget http://mirror.bit.edu.cn/apache/tomcat/tomcat-8/v${VERSION}/bin/apache-to
mcat-${VERSION}.tar.gz

RUN tar xf apache-tomcat-${VERSION}.tar.gz

RUN mv apache-tomcat-${VERSION} /usr/local/tomcat

RUN rm -rf apache-tomcat-${VERSION}.tar.gz /usr/local/tomcat/webapps/*

RUN mkdir /usr/local/tomcat/webapps/ROOT

ADD ./jdk /usr/local/jdk

RUN echo "export TOMCAT_HOME=/usr/local/tomcat" >> /etc/profile

RUN echo "export JAVA_HOME=/usr/local/jdk" >> /etc/profile

RUN echo "export PATH=$TOMCAT_HOME/bin:$JAVA_HOME/bin:$PATH" >> /etc/profile

RUN echo "export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar" >> /etc/profile

RUN source /etc/profile

EXPOSE 8080

CMD ["/usr/local/tomcat/bin/catalina.sh","run"]
```

```
[root@jenkins-server tomcatdir]# cp -r /usr/local/jdk /root/tomcatdir/
[root@jenkins-server tomcatdir]# ls
Dockerfile  jdk
```

8.5.3 使用docker build创建镜像

```
[root@jenkins-server tomcatdir]# docker build -t 192.168.122.8/library/tomcat:8538 .

[root@jenkins-server tomcatdir]# docker build -t 192.168.122.8/library/tomcat:8540 .
Sending build context to Docker daemon  397.8MB
Step 1/18 : FROM centos:latest
 ---> 9f38484d220f
```

```
Step 2/18 : MAINTAINER "aiops<admin@aiops.net.cn>"
 ---> Using cache
 ---> b7650836fa50
Step 3/18 : ENV VERSION=8.5.40
 ---> Running in ee48ae979f03
Removing intermediate container ee48ae979f03
 ---> b433d22b1965
Step 4/18 : ENV JAVA_HOME=/usr/local/jdk
 ---> Running in d351dda0414a
Removing intermediate container d351dda0414a
 ---> c26ddbf87fa0
Step 5/18 : RUN yum -y install wget
 ---> Running in 55d05fb7543c
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: mirrors.nwsuaf.edu.cn
 * extras: mirrors.nwsuaf.edu.cn
 * updates: mirrors.nwsuaf.edu.cn
Resolving Dependencies
--> Running transaction check
---> Package wget.x86_64 0:1.14-18.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package         Arch              Version                Repository       Size
================================================================================
Installing:
 wget            x86_64            1.14-18.el7            base            547 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 547 k
Installed size: 2.0 M
Downloading packages:
warning: /var/cache/yum/x86_64/7/base/packages/wget-1.14-18.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID f4a80eb5: NOKEY
Public key for wget-1.14-18.el7.x86_64.rpm is not installed
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Importing GPG key 0xF4A80EB5:
 Userid     : "CentOS-7 Key (CentOS 7 Official Signing Key) <security@centos.org>"
 Fingerprint: 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5
 Package    : centos-release-7-6.1810.2.el7.centos.x86_64 (@CentOS)
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : wget-1.14-18.el7.x86_64                                      1/1
install-info: No such file or directory for /usr/share/info/wget.info.gz
```

```
  Verifying  : wget-1.14-18.el7.x86_64                                    1/1

Installed:
  wget.x86_64 0:1.14-18.el7

Complete!
Removing intermediate container 55d05fb7543c
 ---> dc9b4cc6cb8d
Step 6/18 : RUN wget http://mirror.bit.edu.cn/apache/tomcat/tomcat-8/v${VERSION}/bin/apache-tomcat-${VERSION}.tar.gz
 ---> Running in f93ff8e6cbf0
--2019-05-05 03:59:46--  http://mirror.bit.edu.cn/apache/tomcat/tomcat-8/v8.5.40/bin/apache-tomcat-8.5.40.tar.gz
Resolving mirror.bit.edu.cn (mirror.bit.edu.cn)... 202.204.80.77, 219.143.204.117, 2001:da8:204:2001:250:56ff:fea1:22
Connecting to mirror.bit.edu.cn (mirror.bit.edu.cn)|202.204.80.77|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9690027 (9.2M) [application/octet-stream]
Saving to: 'apache-tomcat-8.5.40.tar.gz'

     0K .......... .......... .......... .......... ..........  0%  307K 31s
    50K .......... .......... .......... .......... ..........  1%  725K 22s
   100K .......... .......... .......... .......... ..........  1%  526K 20s
   150K .......... .......... .......... .......... ..........  2% 1.01M 17s
   200K .......... .......... .......... .......... ..........  2%  931K 16s
   250K .......... .......... .......... .......... ..........  3% 1.07M 15s
   300K .......... .......... .......... .......... ..........  3% 1.03M 14s
   350K .......... .......... .......... .......... ..........  4%  895K 13s
   400K .......... .......... .......... .......... ..........  4%  834K 13s
   450K .......... .......... .......... .......... ..........  5% 1.14M 12s
   500K .......... .......... .......... .......... ..........  5% 1.04M 12s
   550K .......... .......... .......... .......... ..........  6%  905K 12s
   600K .......... .......... .......... .......... ..........  6%  700K 12s
   650K .......... .......... .......... .......... ..........  7%  909K 11s
   700K .......... .......... .......... .......... ..........  7% 1.15M 11s
   750K .......... .......... .......... .......... ..........  8%  862K 11s
   800K .......... .......... .......... .......... ..........  8%  932K 11s
   850K .......... .......... .......... .......... ..........  9%  811K 11s
   900K .......... .......... .......... .......... .......... 10%  935K 11s
   950K .......... .......... .......... .......... .......... 10%  756K 11s
  1000K .......... .......... .......... .......... .......... 11% 1.41M 10s
  1050K .......... .......... .......... .......... .......... 11%  765K 10s
  1100K .......... .......... .......... .......... .......... 12% 1008K 10s
  1150K .......... .......... .......... .......... .......... 12%  742K 10s
  1200K .......... .......... .......... .......... .......... 13%  681K 10s
  1250K .......... .......... .......... .......... .......... 13%  937K 10s
  1300K .......... .......... .......... .......... .......... 14% 1000K 10s
  1350K .......... .......... .......... .......... .......... 14% 1.00M 10s
  1400K .......... .......... .......... .......... .......... 15% 1.56M 9s
  1450K .......... .......... .......... .......... .......... 15% 1.35M 9s
  1500K .......... .......... .......... .......... .......... 16% 1.37M 9s
  1550K .......... .......... .......... .......... .......... 16% 1.25M 9s
  1600K .......... .......... .......... .......... .......... 17% 1.27M 9s
  1650K .......... .......... .......... .......... .......... 17% 1.24M 9s
  1700K .......... .......... .......... .......... .......... 18% 1.31M 9s
  1750K .......... .......... .......... .......... .......... 19%  336K 9s
  1800K .......... .......... .......... .......... .......... 19%  545K 9s
  1850K .......... .......... .......... .......... .......... 20% 1.55M 9s
  1900K .......... .......... .......... .......... .......... 20%  918K 9s
  1950K .......... .......... .......... .......... .......... 21% 1.20M 9s
  2000K .......... .......... .......... .......... .......... 21%  870K 9s
  2050K .......... .......... .......... .......... .......... 22%  961K 8s
  2100K .......... .......... .......... .......... .......... 22% 1.00M 8s
  2150K .......... .......... .......... .......... .......... 23% 1.32M 8s
  2200K .......... .......... .......... .......... .......... 23%  904K 8s
  2250K .......... .......... .......... .......... .......... 24% 6.58M 8s
  2300K .......... .......... .......... .......... .......... 24%  654K 8s
  2350K .......... .......... .......... .......... .......... 25% 1.44M 8s
  2400K .......... .......... .......... .......... .......... 25%  475K 8s
  2450K .......... .......... .......... .......... .......... 26% 1.05M 8s
  2500K .......... .......... .......... .......... .......... 26%  813K 8s
  2550K .......... .......... .......... .......... .......... 27%  916K 8s
  2600K .......... .......... .......... .......... .......... 28% 1.31M 8s
  2650K .......... .......... .......... .......... .......... 28%  926K 8s
  2700K .......... .......... .......... .......... .......... 29% 1.42M 8s
  2750K .......... .......... .......... .......... .......... 29%  528K 8s
  2800K .......... .......... .......... .......... .......... 30%  879K 7s
  2850K .......... .......... .......... .......... .......... 30%  926K 7s
  2900K .......... .......... .......... .......... .......... 31% 1.16M 7s
  2950K .......... .......... .......... .......... .......... 31%  800K 7s
  3000K .......... .......... .......... .......... .......... 32% 1.03M 7s
  3050K .......... .......... .......... .......... .......... 32%  707K 7s
  3100K .......... .......... .......... .......... .......... 33%  903K 7s
  3150K .......... .......... .......... .......... .......... 33%  867K 7s
  3200K .......... .......... .......... .......... .......... 34% 1.40M 7s
  3250K .......... .......... .......... .......... .......... 34% 1.05M 7s
  3300K .......... .......... .......... .......... .......... 35% 1.19M 7s
  3350K .......... .......... .......... .......... .......... 35% 1.03M 7s
  3400K .......... .......... .......... .......... .......... 36%  925K 7s
  3450K .......... .......... .......... .......... .......... 36%  720K 7s
  3500K .......... .......... .......... .......... .......... 37% 1.09M 7s
  3550K .......... .......... .......... .......... .......... 38%  386K 7s
  3600K .......... .......... .......... .......... .......... 38%  109M 7s
  3650K .......... .......... .......... .......... .......... 39%  624K 6s
  3700K .......... .......... .......... .......... .......... 39%  758K 6s
  3750K .......... .......... .......... .......... .......... 40% 1.08M 6s
  3800K .......... .......... .......... .......... .......... 40%  885K 6s
  3850K .......... .......... .......... .......... .......... 41%  609K 6s
  3900K .......... .......... .......... .......... .......... 41%  843K 6s
  3950K .......... .......... .......... .......... .......... 42%  269K 6s
  4000K .......... .......... .......... .......... .......... 42%  717K 6s
  4050K .......... .......... .......... .......... .......... 43% 1.12M 6s
  4100K .......... .......... .......... .......... .......... 43% 1.11M 6s
  4150K .......... .......... .......... .......... .......... 44%  410K 6s
  4200K .......... .......... .......... .......... .......... 44% 1.49M 6s
  4250K .......... .......... .......... .......... .......... 45%  720K 6s
  4300K .......... .......... .......... .......... .......... 45%  571K 6s
  4350K .......... .......... .......... .......... .......... 46%  646K 6s
  4400K .......... .......... .......... .......... .......... 47%  586K 6s
  4450K .......... .......... .......... .......... .......... 47%  580K 6s
  4500K .......... .......... .......... .......... .......... 48%  524K 6s
  4550K .......... .......... .......... .......... .......... 48%  360K 6s
  4600K .......... .......... .......... .......... .......... 49%  368K 6s
  4650K .......... .......... .......... .......... .......... 49%  932K 6s
  4700K .......... .......... .......... .......... .......... 50%  254K 6s
  4750K .......... .......... .......... .......... .......... 50%  737K 6s
  4800K .......... .......... .......... .......... .......... 51%  375K 6s
  4850K .......... .......... .......... .......... .......... 51%  389K 6s
  4900K .......... .......... .......... .......... .......... 52%  523K 6s
  4950K .......... .......... .......... .......... .......... 52%  752K 6s
  5000K .......... .......... .......... .......... .......... 53%  435K 6s
  5050K .......... .......... .......... .......... .......... 53%  386K 6s
  5100K .......... .......... .......... .......... .......... 54%  496K 6s
  5150K .......... .......... .......... .......... .......... 54%  522K 6s
  5200K .......... .......... .......... .......... .......... 55%  431K 6s
  5250K .......... .......... .......... .......... .......... 56%  185K 6s
  5300K .......... .......... .......... .......... .......... 56%  362K 6s
  5350K .......... .......... .......... .......... .......... 57%  446K 6s
  5400K .......... .......... .......... .......... .......... 57%  490K 6s
  5450K .......... .......... .......... .......... .......... 58%  533K 6s
  5500K .......... .......... .......... .......... .......... 58%  323K 6s
  5550K .......... .......... .......... .......... .......... 59%  429K 6s
  5600K .......... .......... .......... .......... .......... 59%  901K 5s
  5650K .......... .......... .......... .......... .......... 60%  386K 5s
  5700K .......... .......... .......... .......... .......... 60%  425K 5s
  5750K .......... .......... .......... .......... .......... 61%  675K 5s
  5800K .......... .......... .......... .......... .......... 61%  436K 5s
  5850K .......... .......... .......... .......... .......... 62%  458K 5s
  5900K .......... .......... .......... .......... .......... 62%  865K 5s
  5950K .......... .......... .......... .......... .......... 63%  295K 5s
  6000K .......... .......... .......... .......... .......... 63%  330K 5s
  6050K .......... .......... .......... .......... .......... 64%  548K 5s
  6100K .......... .......... .......... .......... .......... 64%  337K 5s
  6150K .......... .......... .......... .......... .......... 65%  402K 5s
  6200K .......... .......... .......... .......... .......... 66%  772K 5s
  6250K .......... .......... .......... .......... .......... 66%  236K 5s
  6300K .......... .......... .......... .......... .......... 67%  375K 5s
  6350K .......... .......... .......... .......... .......... 67%  479K 5s
  6400K .......... .......... .......... .......... .......... 68%  791K 5s
  6450K .......... .......... .......... .......... .......... 68%  364K 5s
  6500K .......... .......... .......... .......... .......... 69%  628K 5s
  6550K .......... .......... .......... .......... .......... 69%  366K 4s
  6600K .......... .......... .......... .......... .......... 70%  532K 4s
  6650K .......... .......... .......... .......... .......... 70%  415K 4s
  6700K .......... .......... .......... .......... .......... 71%  464K 4s
  6750K .......... .......... .......... .......... .......... 71%  542K 4s
  6800K .......... .......... .......... .......... .......... 72%  263K 4s
  6850K .......... .......... .......... .......... .......... 72%  439K 4s
  6900K .......... .......... .......... .......... .......... 73%  750K 4s
  6950K .......... .......... .......... .......... .......... 73%  624K 4s
  7000K .......... .......... .......... .......... .......... 74%  610K 4s
  7050K .......... .......... .......... .......... .......... 75% 2.76M 4s
  7100K .......... .......... .......... .......... .......... 75%  400K 4s
  7150K .......... .......... .......... .......... .......... 76%  643K 4s
  7200K .......... .......... .......... .......... .......... 76%  389K 4s
  7250K .......... .......... .......... .......... .......... 77%  780K 3s
  7300K .......... .......... .......... .......... .......... 77%  243K 3s
  7350K .......... .......... .......... .......... .......... 78%  762K 3s
  7400K .......... .......... .......... .......... .......... 78%  425K 3s
  7450K .......... .......... .......... .......... .......... 79% 50.2M 3s
  7500K .......... .......... .......... .......... .......... 79% 1.05M 3s
  7550K .......... .......... .......... .......... .......... 80% 1.22M 3s
  7600K .......... .......... .......... .......... .......... 80%  818K 3s
  7650K .......... .......... .......... .......... .......... 81%  306K 3s
  7700K .......... .......... .......... .......... .......... 81%  693K 3s
  7750K .......... .......... .......... .......... .......... 82%  346K 3s
  7800K .......... .......... .......... .......... .......... 82%  821K 3s
  7850K .......... .......... .......... .......... .......... 83%  740K 3s
  7900K .......... .......... .......... .......... .......... 84%  364K 2s
  7950K .......... .......... .......... .......... .......... 84% 1.44M 2s
  8000K .......... .......... .......... .......... .......... 85%  744K 2s
  8050K .......... .......... .......... .......... .......... 85%  452K 2s
  8100K .......... .......... .......... .......... .......... 86%  901K 2s
  8150K .......... .......... .......... .......... .......... 86%  955K 2s
  8200K .......... .......... .......... .......... .......... 87%  326K 2s
  8250K .......... .......... .......... .......... .......... 87%  603K 2s
  8300K .......... .......... .......... .......... .......... 88%  301K 2s
  8350K .......... .......... .......... .......... .......... 88% 2.98M 2s
  8400K .......... .......... .......... .......... .......... 89%  453K 2s
  8450K .......... .......... .......... .......... .......... 89%  744K 2s
  8500K .......... .......... .......... .......... .......... 90%  613K 1s
  8550K .......... .......... .......... .......... .......... 90%  403K 1s
  8600K .......... .......... .......... .......... .......... 91% 1.11M 1s
  8650K .......... .......... .......... .......... .......... 91%  822K 1s
  8700K .......... .......... .......... .......... .......... 92% 1.34M 1s
  8750K .......... .......... .......... .......... .......... 92%  696K 1s
  8800K .......... .......... .......... .......... .......... 93%  374K 1s
  8850K .......... .......... .......... .......... .......... 94%  371K 1s
  8900K .......... .......... .......... .......... .......... 94%  198K 1s
  8950K .......... .......... .......... .......... .......... 95%  273K 1s
  9000K .......... .......... .......... .......... .......... 95%  243K 1s
  9050K .......... .......... .......... .......... .......... 96%  430K 1s
  9100K .......... .......... .......... .......... .......... 96%  527K 1s
  9150K .......... .......... .......... .......... .......... 97%  382K 0s
  9200K .......... .......... .......... .......... .......... 97%  281K 0s
  9250K .......... .......... .......... .......... .......... 98%  731K 0s
  9300K .......... .......... .......... .......... .......... 98%  500K 0s
  9350K .......... .......... .......... .......... .......... 99%  667K 0s
  9400K .......... .......... .......... .......... .......... 99%  812K 0s
  9450K .......... ..                                         100% 53.8M=16s

```

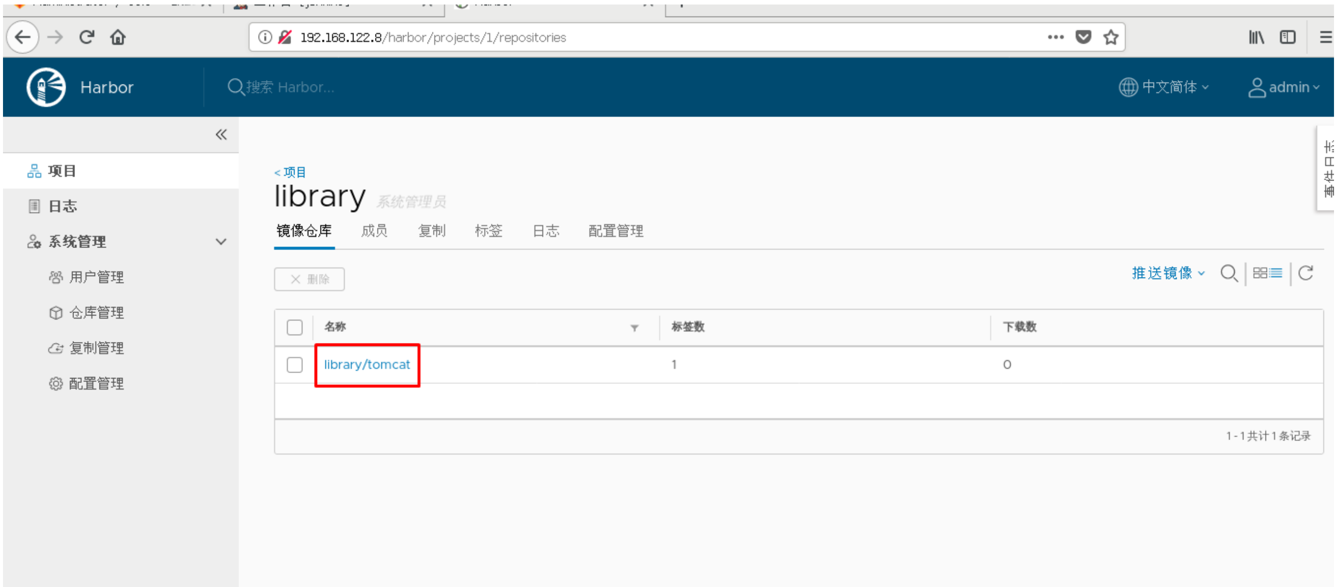
```
2019-05-05 04:00:02 (594 KB/s) - 'apache-tomcat-8.5.40.tar.gz' saved [9690027/9690027]

Removing intermediate container f93ff8e6cbf0
 ---> 47a6a11aa578
Step 7/18 : RUN tar xf apache-tomcat-${VERSION}.tar.gz
 ---> Running in 5b991b7dabc7
Removing intermediate container 5b991b7dabc7
 ---> e96d749db67a
Step 8/18 : RUN mv apache-tomcat-${VERSION} /usr/local/tomcat
 ---> Running in 5d0a3c55f3fa
Removing intermediate container 5d0a3c55f3fa
 ---> d90d5cb19ec7
Step 9/18 : RUN rm -rf apache-tomcat-${VERSION}.tar.gz /usr/local/tomcat/webapps/*
 ---> Running in 0bb777acab17
Removing intermediate container 0bb777acab17
 ---> fd5add8a0088
Step 10/18 : RUN mkdir /usr/local/tomcat/webapps/ROOT
 ---> Running in 50fb05774360
Removing intermediate container 50fb05774360
 ---> 17a8f3e9d68a
Step 11/18 : ADD ./jdk /usr/local/jdk
 ---> eef922c51c4d
Step 12/18 : RUN echo "export TOMCAT_HOME=/usr/local/tomcat" >> /etc/profile
 ---> Running in 8b0d0b866ec3
Removing intermediate container 8b0d0b866ec3
 ---> 5e6da5e35d20
Step 13/18 : RUN echo "export JAVA_HOME=/usr/local/jdk" >> /etc/profile
 ---> Running in 68e1a9548e6a
Removing intermediate container 68e1a9548e6a
 ---> 99227d4f27ff
Step 14/18 : RUN echo "export PATH=$TOMCAT_HOME/bin:$JAVA_HOME/bin:$PATH" >> /etc/profile
 ---> Running in 087f6b6a386a
Removing intermediate container 087f6b6a386a
 ---> addf2b7943a3
Step 15/18 : RUN echo "export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar" >> /etc/profile
 ---> Running in 0c5a3f38c890
Removing intermediate container 0c5a3f38c890
 ---> bacedda825ad
Step 16/18 : RUN source /etc/profile
 ---> Running in 2a0cc68511ee
Removing intermediate container 2a0cc68511ee
 ---> eeb8c789791a
Step 17/18 : EXPOSE 8080
 ---> Running in e3eabe49efb2
Removing intermediate container e3eabe49efb2
 ---> 38dd3bcb0fc8
Step 18/18 : CMD ["/usr/local/tomcat/bin/catalina.sh","run"]
 ---> Running in f73a8e53c8bc
Removing intermediate container f73a8e53c8bc
 ---> bb17fd9e88fd
```

```
Successfully built bb17fd9e88fd
Successfully tagged 192.168.122.8/library/tomcat:8540

```

8.5.4 上传到harbor镜像

```
[root@jenkins-server tomcatdir]# docker images
REPOSITORY                     TAG                 IMAGE ID            CREATED             SIZE
192.168.122.8/library/tomcat   8540                bb17fd9e88fd        56 seconds ago      726MB

[root@jenkins-server ~]# docker login http://192.168.122.8
Username: admin
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[root@jenkins-server ~]# docker push 192.168.122.8/library/tomcat:8540
The push refers to repository [192.168.122.8/library/tomcat]
822588c6f03b: Pushed
36d19227df3b: Pushed
13c98ca3d3c2: Pushed
4e5d8a69b140: Pushed
f6b227267eda: Pushed
b5cc28a7cb82: Pushed
d219e1115c86: Pushed
8669b387dc4e: Pushed
05720cf5d863: Pushed
f6f37927944a: Pushed
b86975510122: Pushed
d69483a6face: Pushed
8540: digest: sha256:cc373950dee499be449c167859fcca69bc9722734da186254019ad22c34df5e9 size: 2833
```

8.5.5 harbor仓库验证



名称	标签数	下载数
library/tomcat	1	0

1 - 1 共计 1 条记录

8.5.6 tomcat镜像是否可用

```
[root@web-server ~]# docker run -d 192.168.122.8/library/tomcat:8540
Unable to find image '192.168.122.8/library/tomcat:8540' locally
8540: Pulling from library/tomcat
8ba884070f61: Pull complete
b2355315a96b: Pull complete
135c4351a789: Pull complete
fcdd5340ca84: Pull complete
f993509b6844: Pull complete
58ec4a771a1c: Pull complete
5656fbb82114: Pull complete
3b3f26a0a444: Pull complete
0c9837669885: Pull complete
2eda638caa06: Pull complete
61cc01ec979c: Pull complete
039334651d42: Pull complete
Digest: sha256:cc373950dee499be449c167859fcca69bc9722734da186254019ad22c34df5e9
Status: Downloaded newer image for 192.168.122.8/library/tomcat:8540
99ccbe2cbd8e3b87bf84b7f0ee1d7d615865e04e5d283bb09805ae7bad3966da


[root@web-server ~]# docker ps
CONTAINER ID        IMAGE                               COMMAND
CREATED             STATUS              PORTS               NAMES
99ccbe2cbd8e        192.168.122.8/library/tomcat:8540   "/usr/local/tomcat/b…"   18
seconds ago      Up 17 seconds       8080/tcp            jovial_edison

[root@web-server ~]# curl http://172.17.0.2:8080
```

8.6 创建构建任务

第一步：jenkins获取项目代码

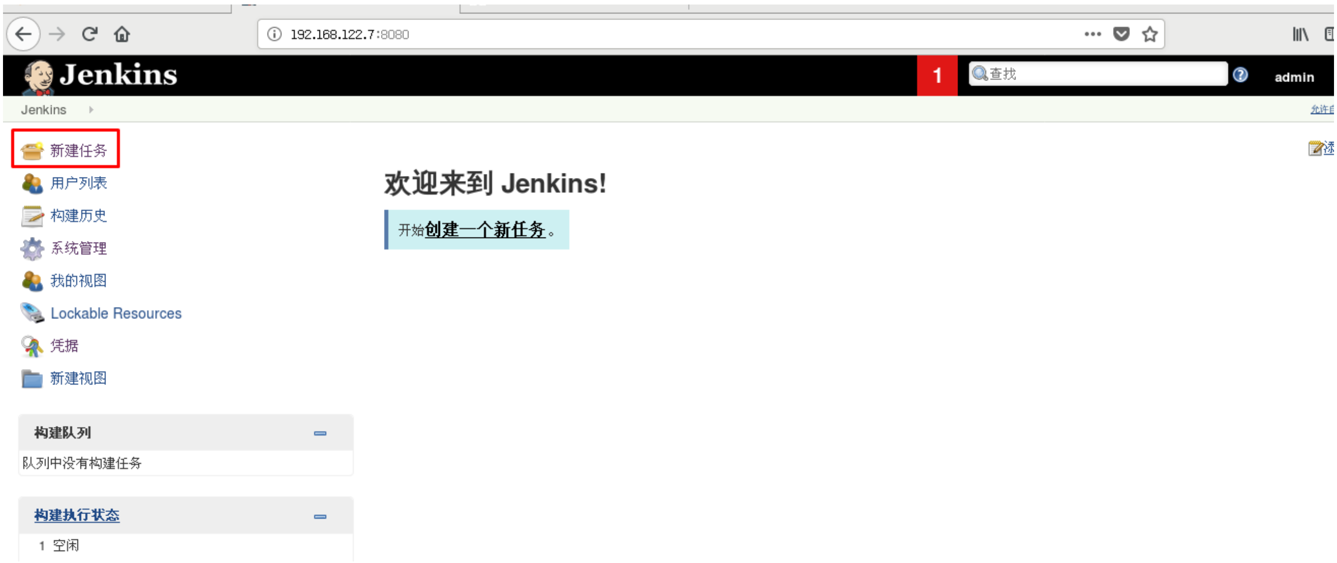
第二步：jenkins对项目代码编译，由maven完成

第三步：jenkins使用docker对编译完成的项目代码进行打包，打包成容器应用镜像

第四步：jenkins把打包的容器应用镜像上传到harbor

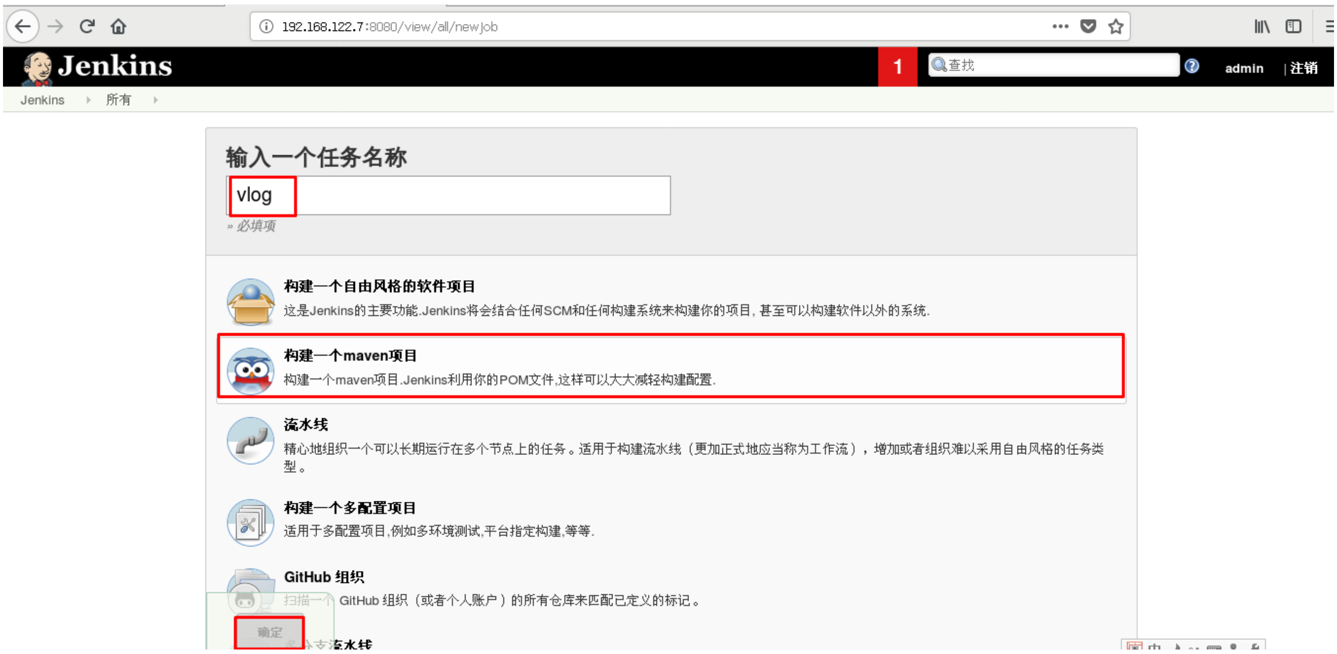
第五步：jenkins通过ssh插件完成对web-server进行运行容器应用镜像的操作

在jenkins web页面中创建



欢迎来到 Jenkins!

开始**创建一个新任务**。



输入一个任务名称

vlog

» 必填项

构建一个自由风格的软件项目
这是Jenkins的主要功能.Jenkins将会结合任何SCM和任何构建系统来构建你的项目, 甚至可以构建软件以外的系统.

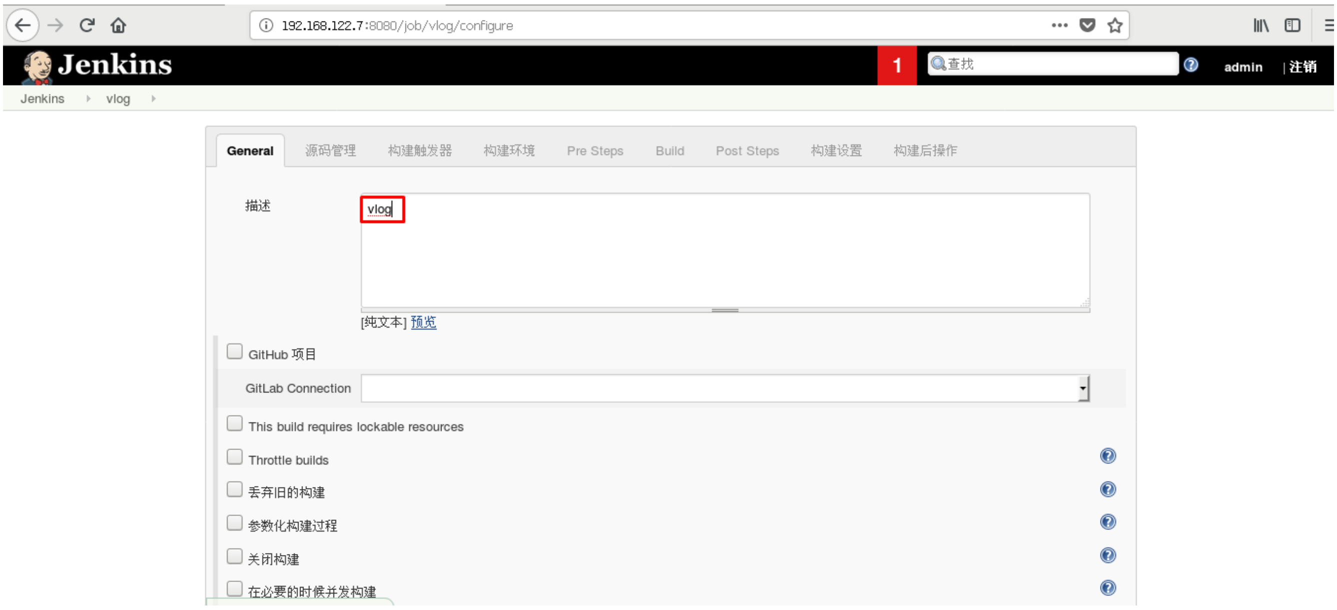
构建一个maven项目
构建一个maven项目.Jenkins利用你的POM文件,这样可以大大减轻构建配置.

流水线
精心地组织一个可以长期运行在多个节点上的任务。适用于构建流水线（更加正式地应当称为工作流），增加或者组织难以采用自由风格的任务类型。

构建一个多配置项目
适用于多配置项目,例如多环境测试,平台指定构建,等等.

GitHub 组织
扫描一个 GitHub 组织（或者个人账户）的所有仓库来匹配已定义的标记。

多分支流水线



192.168.122.7:8080/job/vlog/configure

Jenkins › vlog

描述: vlog

[纯文本] 预览

□ GitHub 项目

GitLab Connection

□ This build requires lockable resources

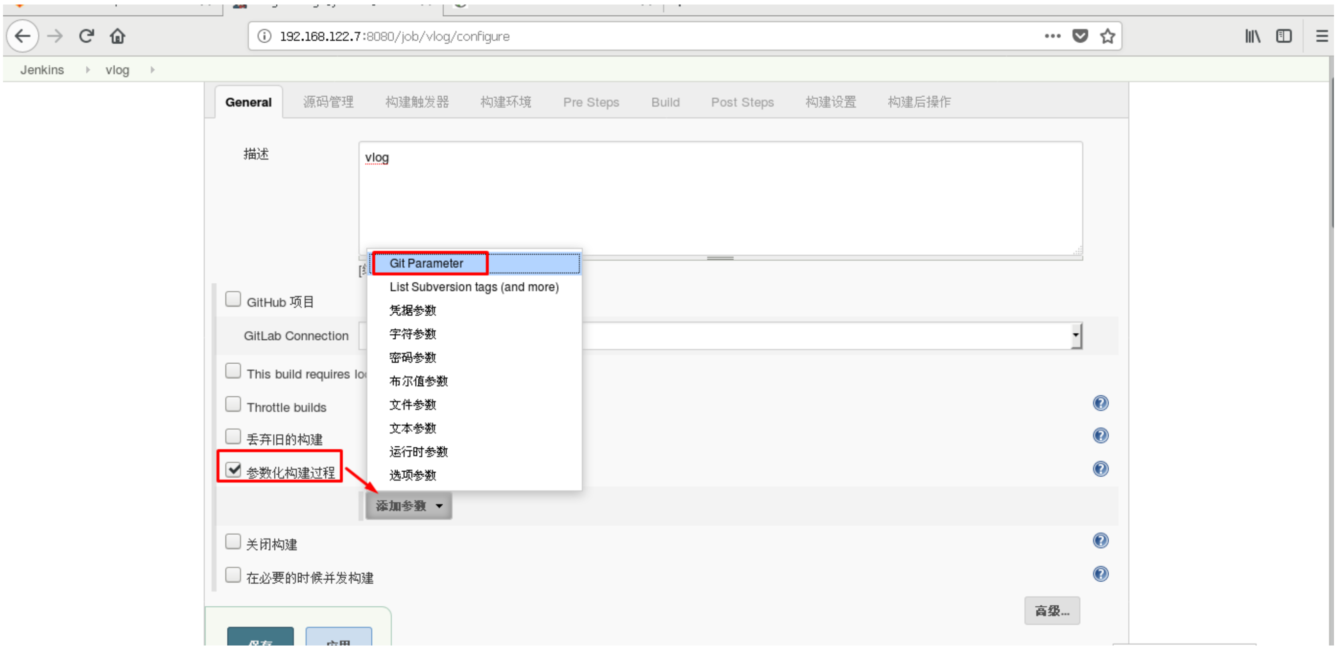
□ Throttle builds

□ 丢弃旧的构建

□ 参数化构建过程

□ 关闭构建

□ 在必要的时候并发构建



192.168.122.7:8080/job/vlog/configure

Jenkins › vlog

描述: vlog

□ GitHub 项目

GitLab Connection

□ This build requires lo

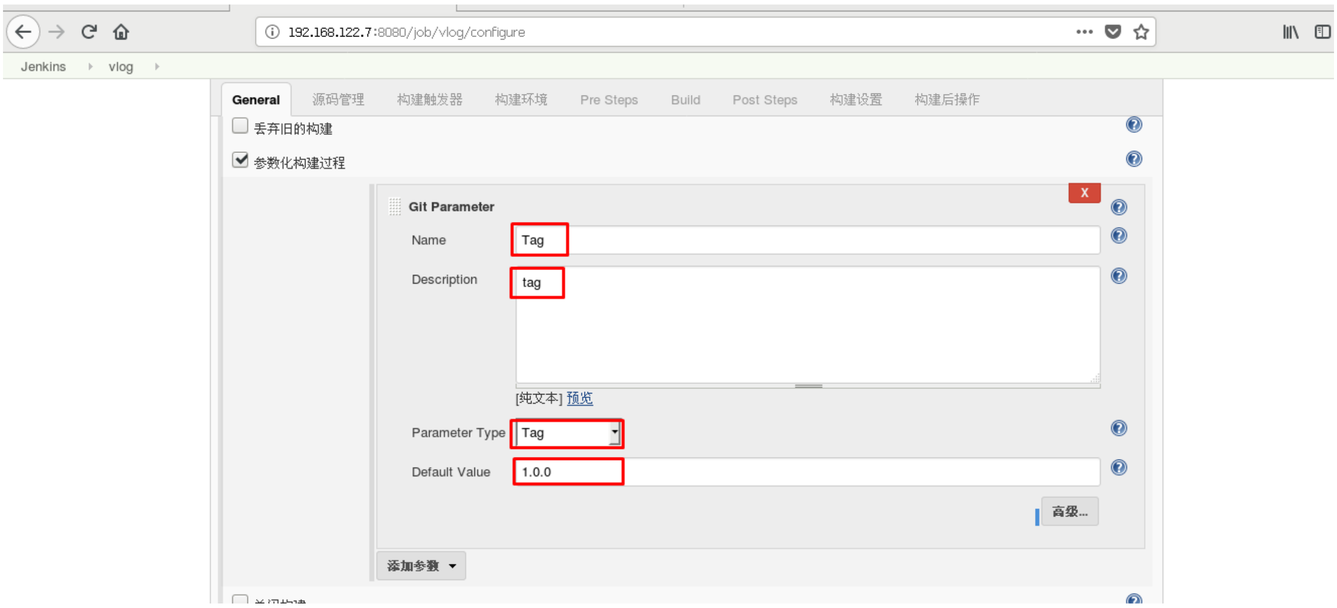
□ Throttle builds

□ 丢弃旧的构建

☑ 参数化构建过程

□ 关闭构建

□ 在必要的时候并发构建



192.168.122.7:8080/job/vlog/configure

Jenkins ▸ vlog ▸

□ 丢弃旧的构建

☑ 参数化构建过程

Git Parameter

Name: Tag

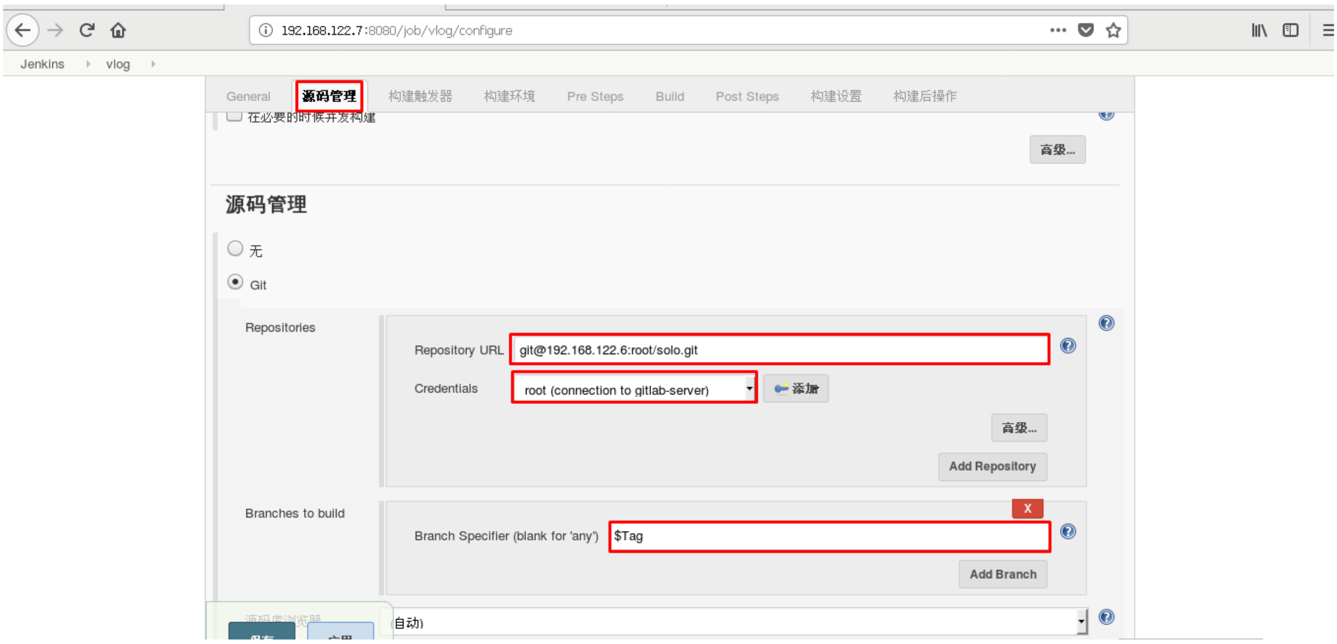
Description: tag

[纯文本] 预览

Parameter Type: Tag

Default Value: 1.0.0

□ 关闭构建



192.168.122.7:8080/job/vlog/configure

Jenkins ▸ vlog ▸

□ 在必要的时候并发构建

源码管理

○ 无

● Git

Repositories

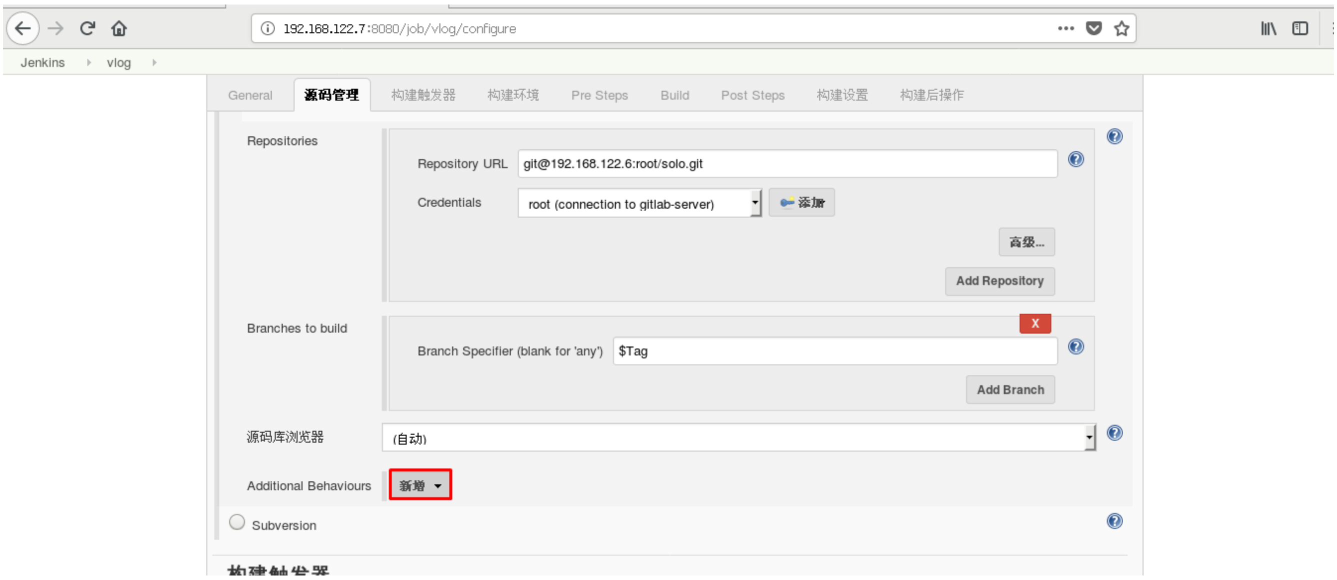
Repository URL: git@192.168.122.6:root/solo.git

Credentials: root (connection to gitlab-server)

Branches to build

Branch Specifier (blank for 'any'): $Tag

源码库浏览器: (自动)



192.168.122.7:8080/job/vlog/configure

Jenkins › vlog ›

Repositories

Repository URL: git@192.168.122.6:root/solo.git

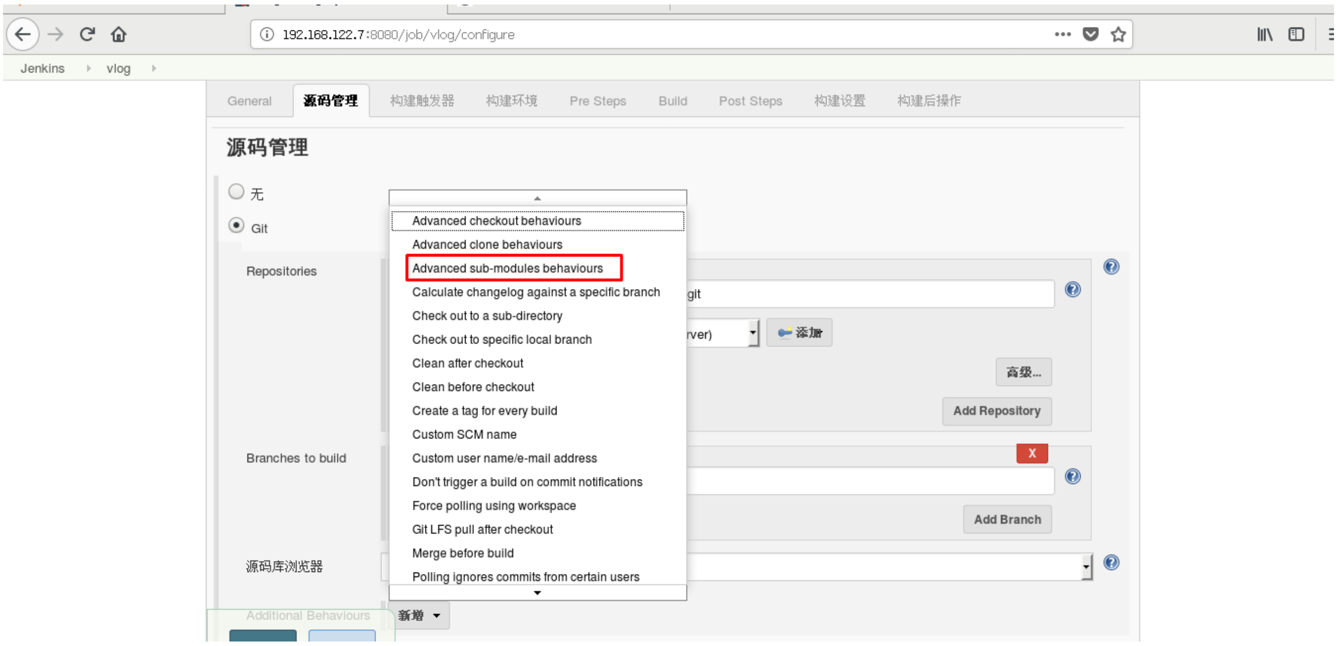
Credentials: root (connection to gitlab-server)

Branches to build

Branch Specifier (blank for 'any'): $Tag

源码库浏览器: (自动)

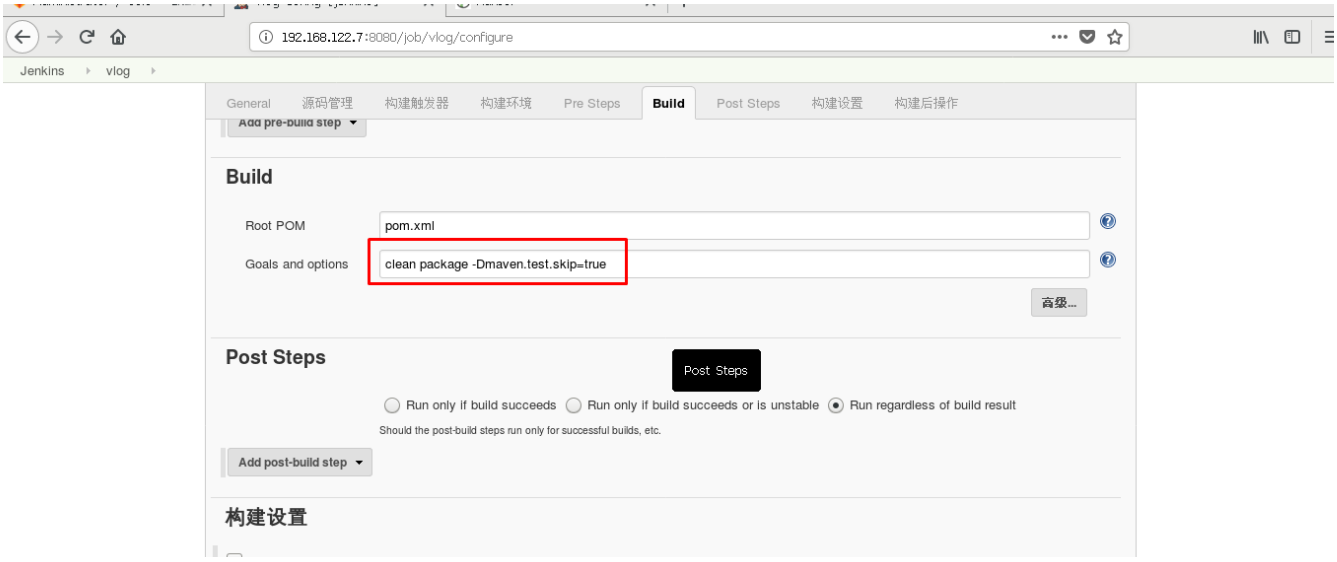
Additional Behaviours



192.168.122.7:8080/job/vlog/configure

Jenkins › vlog ›

General
源码管理
构建触发器
构建环境
Pre Steps
Build
Post Steps
构建设置
构建后操作
源码管理
○ 无
● Git
Repositories
Branches to build
源码库浏览器
Additional Behaviours
新增
Advanced checkout behaviours
Advanced clone behaviours
Advanced sub-modules behaviours
Calculate changelog against a specific branch
Check out to a sub-directory
Check out to specific local branch
Clean after checkout
Clean before checkout
Create a tag for every build
Custom SCM name
Custom user name/e-mail address
Don't trigger a build on commit notifications
Force polling using workspace
Git LFS pull after checkout
Merge before build
Polling ignores commits from certain users
添加
高级...
Add Repository
X
Add Branch



Build

Root POM: pom.xml

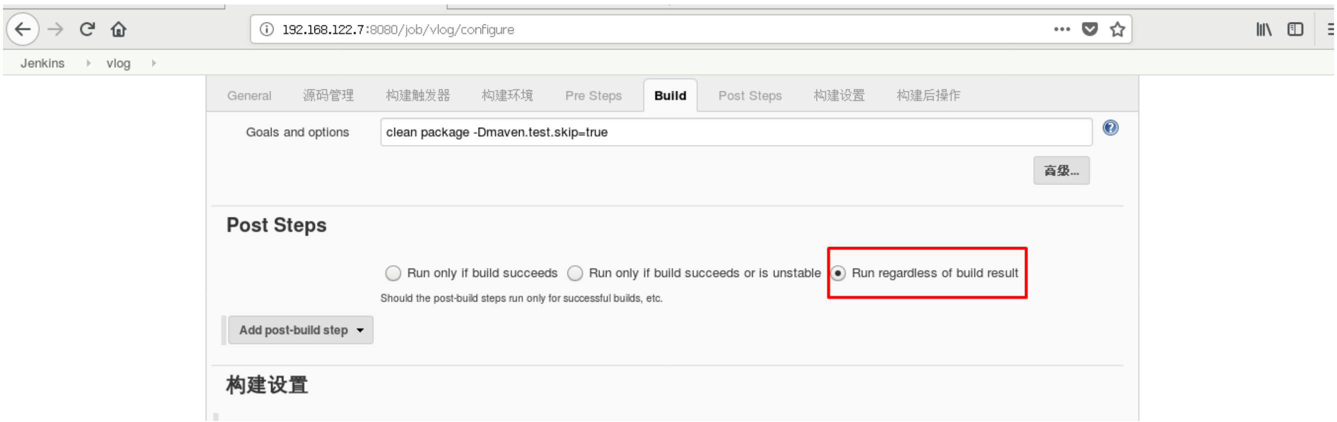
Goals and options: clean package -Dmaven.test.skip=true

Post Steps

○ Run only if build succeeds ○ Run only if build succeeds or is unstable ● Run regardless of build result

Should the post-build steps run only for successful builds, etc.

构建设置



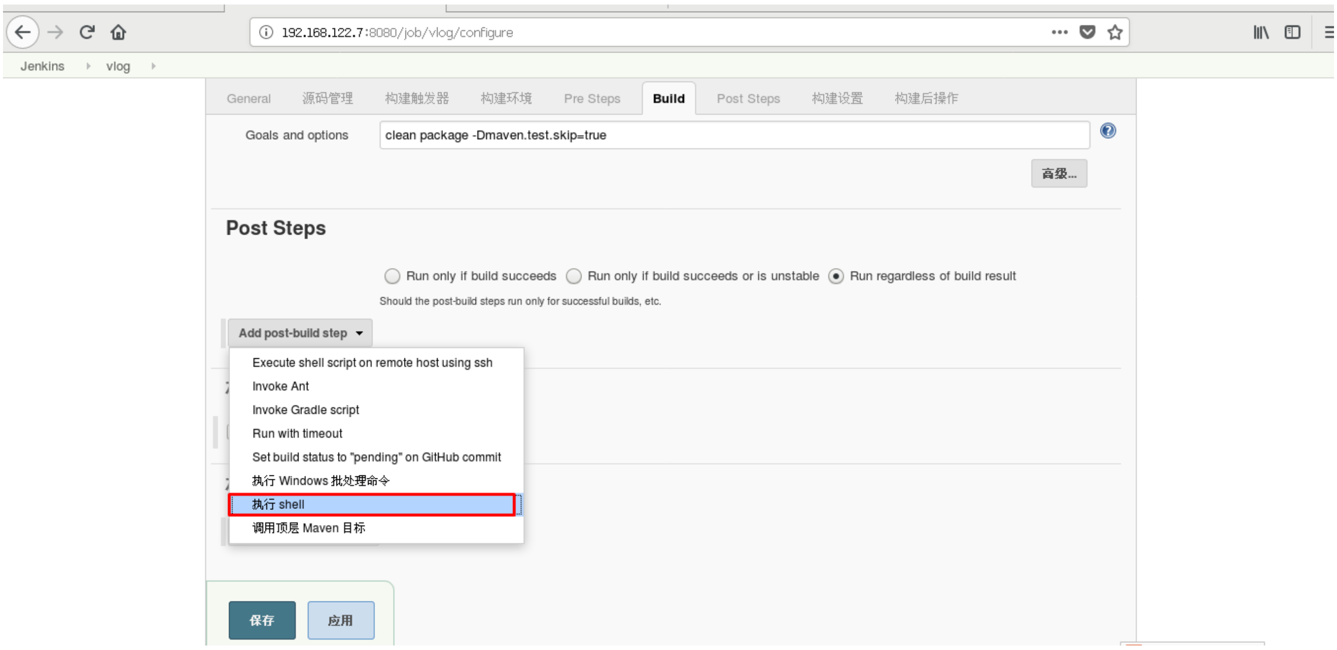
Goals and options: clean package -Dmaven.test.skip=true

Post Steps

○ Run only if build succeeds ○ Run only if build succeeds or is unstable ● Run regardless of build result

Should the post-build steps run only for successful builds, etc.

构建设置

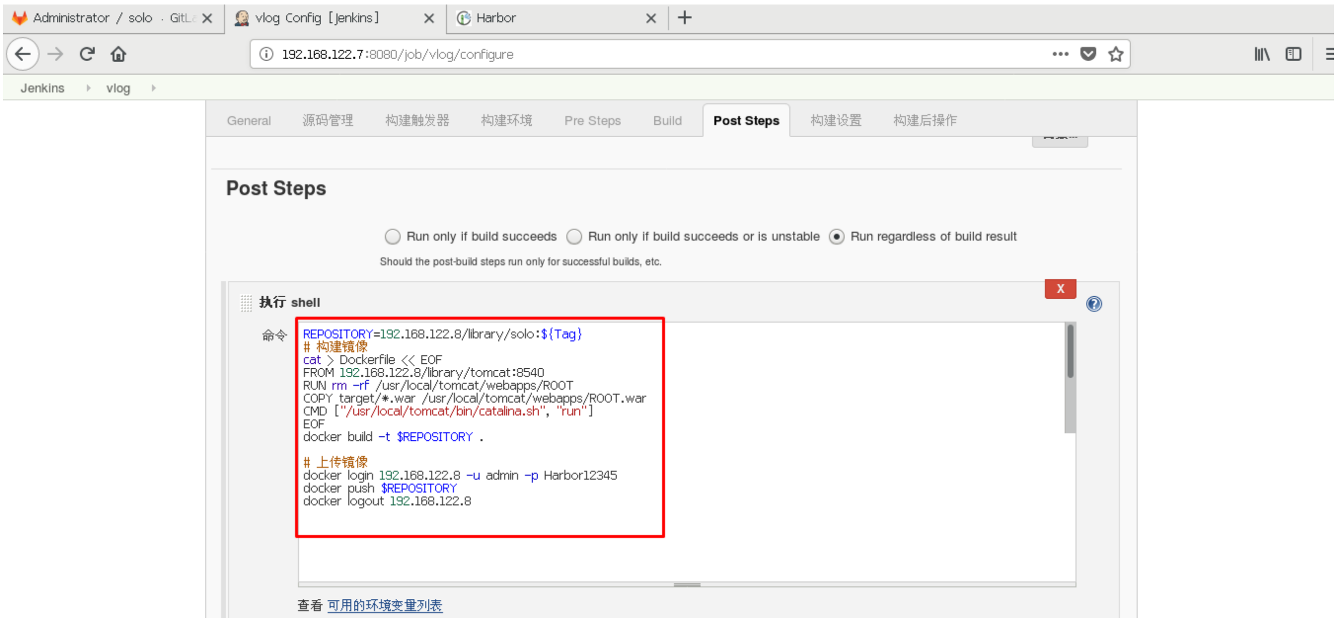


Goals and options: clean package -Dmaven.test.skip=true

Post Steps

○ Run only if build succeeds ○ Run only if build succeeds or is unstable ● Run regardless of build result

Should the post-build steps run only for successful builds, etc.



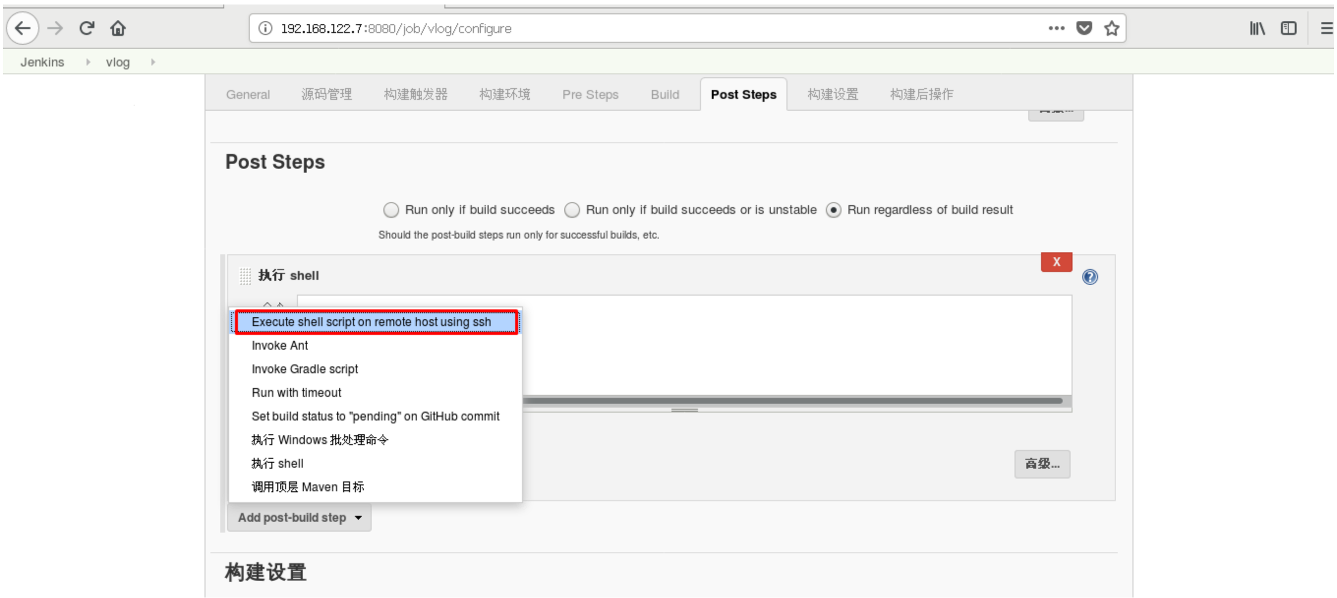
Post Steps

○ Run only if build succeeds ○ Run only if build succeeds or is unstable ● Run regardless of build result

Should the post-build steps run only for successful builds, etc.

```
REPOSITORY=192.168.122.8/library/solo:${Tag}
# 构建镜像
cat > Dockerfile << EOF
FROM 192.168.122.8/library/tomcat:8540
RUN rm -rf /usr/local/tomcat/webapps/ROOT
COPY target/*.war /usr/local/tomcat/webapps/ROOT.war
CMD ["/usr/local/tomcat/bin/catalina.sh", "run"]
EOF
docker build -t $REPOSITORY .

# 上传镜像
docker login 192.168.122.8 -u admin -p Harbor12345
docker push $REPOSITORY
docker logout 192.168.122.8
```

```
REPOSITORY=192.168.122.8/library/solo:${Tag}
# 构建镜像
cat > Dockerfile << EOF
FROM 192.168.122.8/library/tomcat:8540
RUN rm -rf /usr/local/tomcat/webapps/ROOT
COPY target/*.war /usr/local/tomcat/webapps/ROOT.war
CMD ["/usr/local/tomcat/bin/catalina.sh", "run"]
EOF
docker build -t $REPOSITORY .

# 上传镜像
docker login 192.168.122.8 -u admin -p Harbor12345
docker push $REPOSITORY
docker logout 192.168.122.8
```

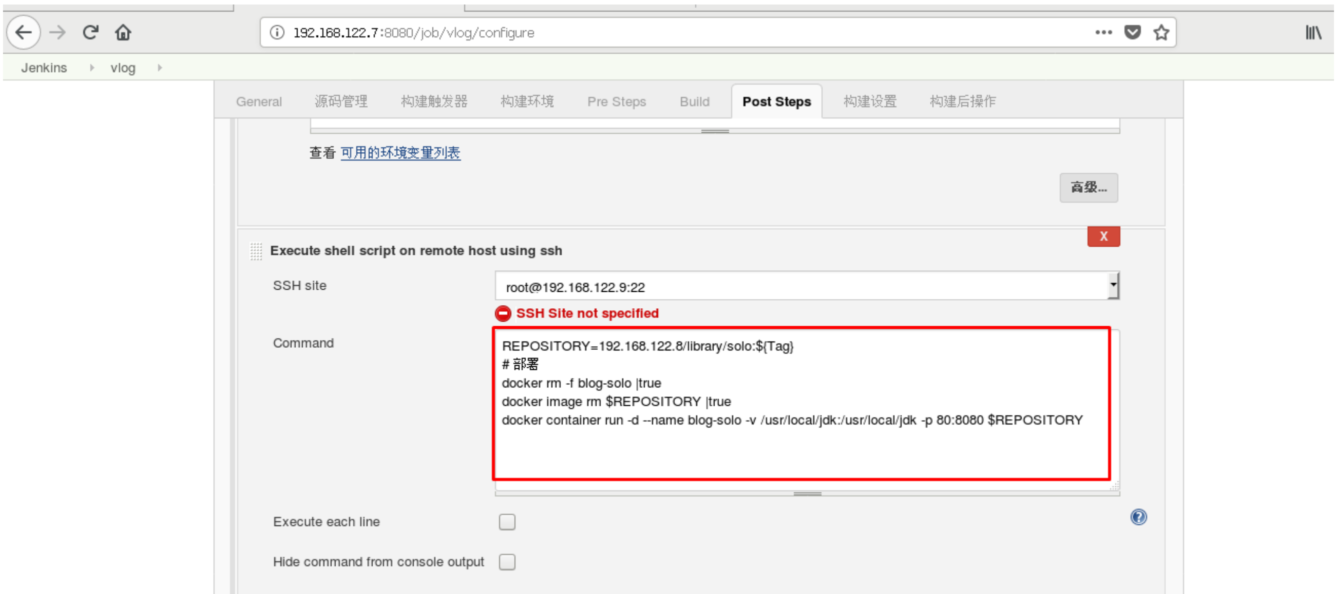


Post Steps

○ Run only if build succeeds ○ Run only if build succeeds or is unstable ● Run regardless of build result

Should the post-build steps run only for successful builds, etc.

执行 shell



Execute shell script on remote host using ssh

SSH site: root@192.168.122.9:22

SSH Site not specified

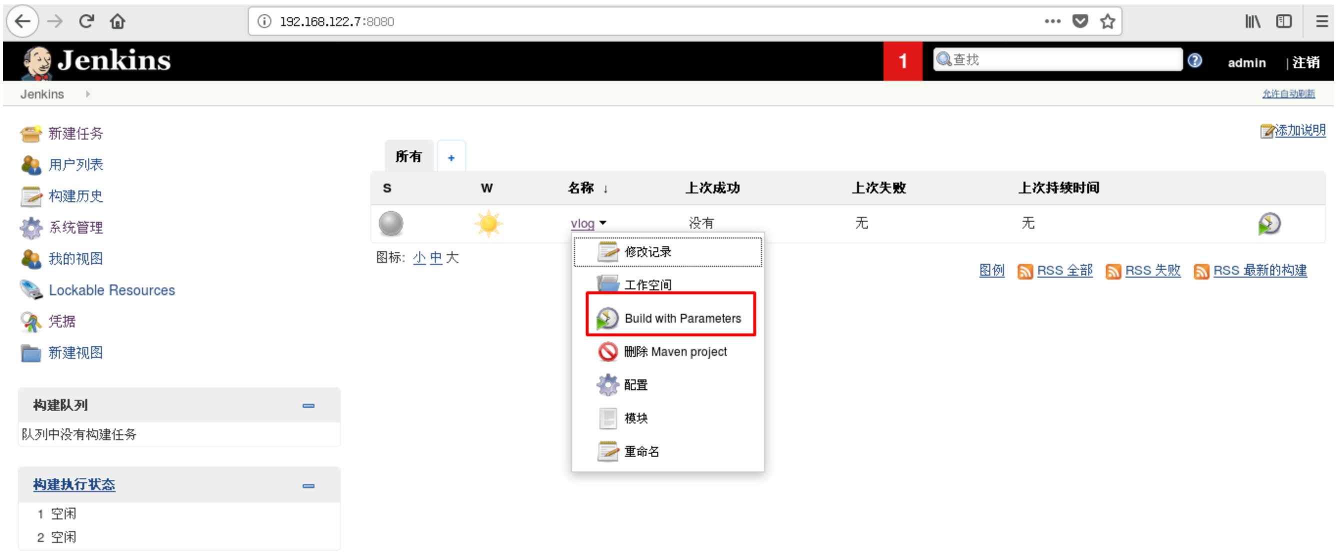
Command:

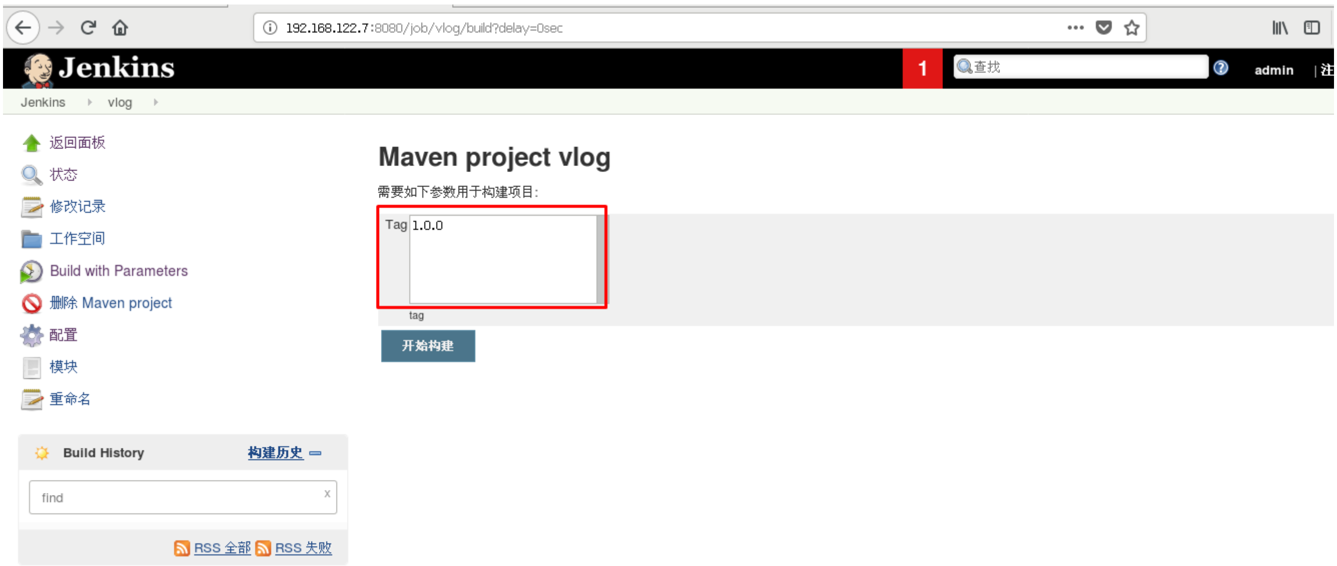
```
REPOSITORY=192.168.122.8/library/solo:${Tag}
# 部署
docker rm -f blog-solo |true
docker image rm $REPOSITORY |true
docker container run -d --name blog-solo -v /usr/local/jdk:/usr/local/jdk -p 80:8080 $REPOSITORY
```

Execute each line □

Hide command from console output □

```
REPOSITORY=192.168.122.8/library/solo:${Tag}
# 部署
docker rm -f blog-solo |true
docker image rm $REPOSITORY |true
docker container run -d --name blog-solo -v /usr/local/jdk:/usr/local/jdk -p 80:8080
$REPOSITORY
```



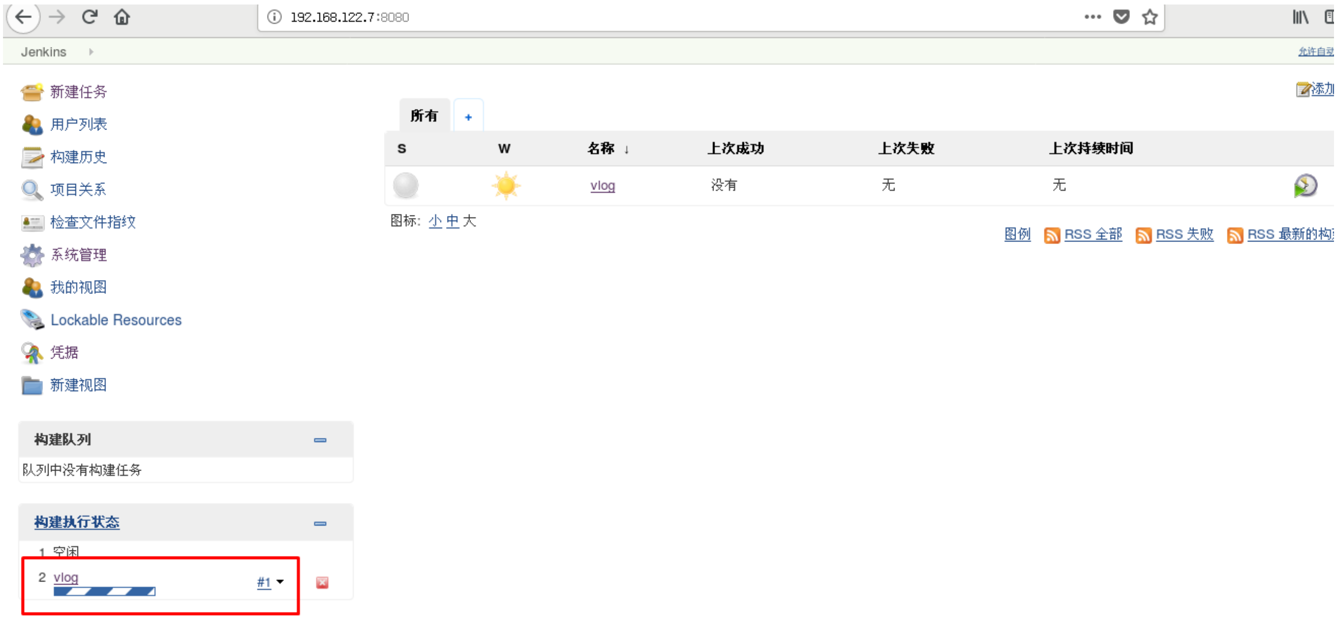

192.168.122.7:8080/job/vlog/build?delay=0sec

Maven project vlog

需要如下参数用于构建项目:

Tag: 1.0.0

tag



192.168.122.7:8080

Jenkins

新建任务
用户列表
构建历史
项目关系
检查文件指纹
系统管理
我的视图
Lockable Resources
凭据
新建视图

构建队列
队列中没有构建任务

构建执行状态
1 空闲
2 vlog #1

所有 +

S	W	名称 ↓	上次成功	上次失败	上次持续时间
○	☀	vlog	没有	无	无

图标: 小 中 大

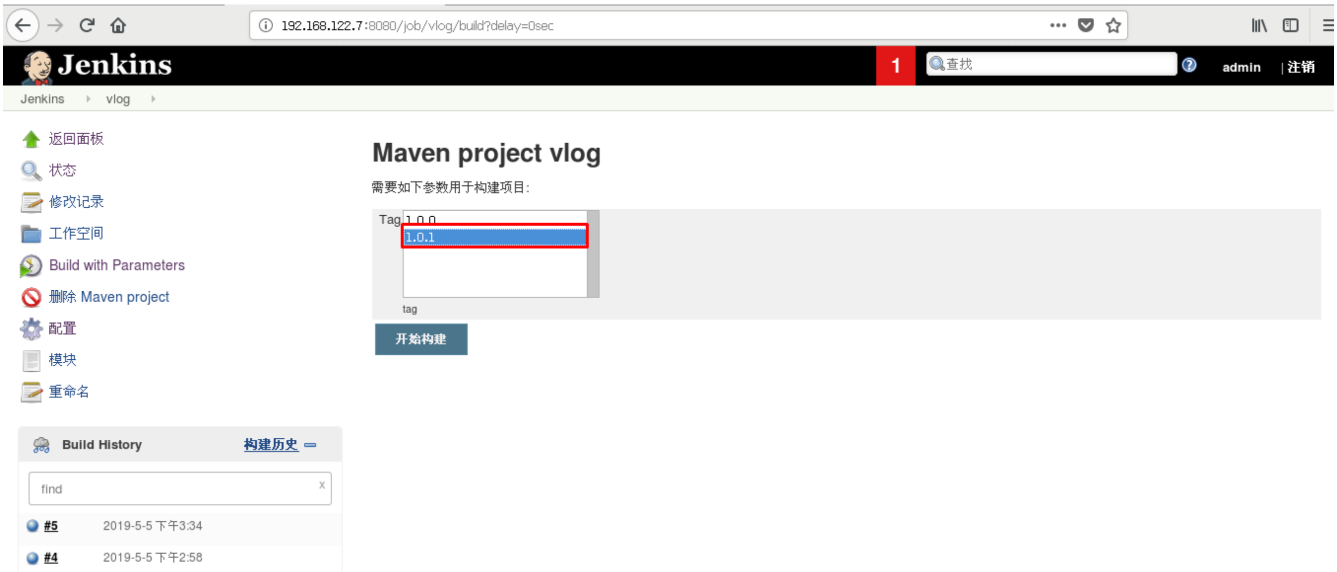
图例 RSS 全部 RSS 失败 RSS 最新的构

查看编译后状态变化

```
[root@jenkins-server jenkins]# pwd
/var/lib/jenkins
[root@jenkins-server jenkins]# ls

workspace

[root@jenkins-server jenkins]# ls workspace/
vlog  vlog@tmp
[root@jenkins-server jenkins]# ls workspace/vlog
CHANGE_LOGS.html  gulpfile.js  package.json       pom.xml    scripts
Dockerfile        LICENSE      package-lock.json  README.md  src



[root@jenkins-server workspace]# ls vlog
CHANGE_LOGS.html  gulpfile.js  package.json       pom.xml    scripts  target
Dockerfile        LICENSE      package-lock.json  README.md  src

[root@jenkins-server vlog]# ls target/
classes  generated-sources  maven-archiver  maven-status  solo  solo.war


```

重新发布新版本

```
[root@dev solo]# git tag 1.0.1
[root@dev solo]# git push origin 1.0.1
Counting objects: 11, done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 524 bytes | 0 bytes/s, done.
Total 6 (delta 3), reused 0 (delta 0)
To git@192.168.122.6:root/solo.git
 * [new tag]         1.0.1 -> 1.0.1

```



192.168.122.7:8080/job/vlog/build?delay=0sec

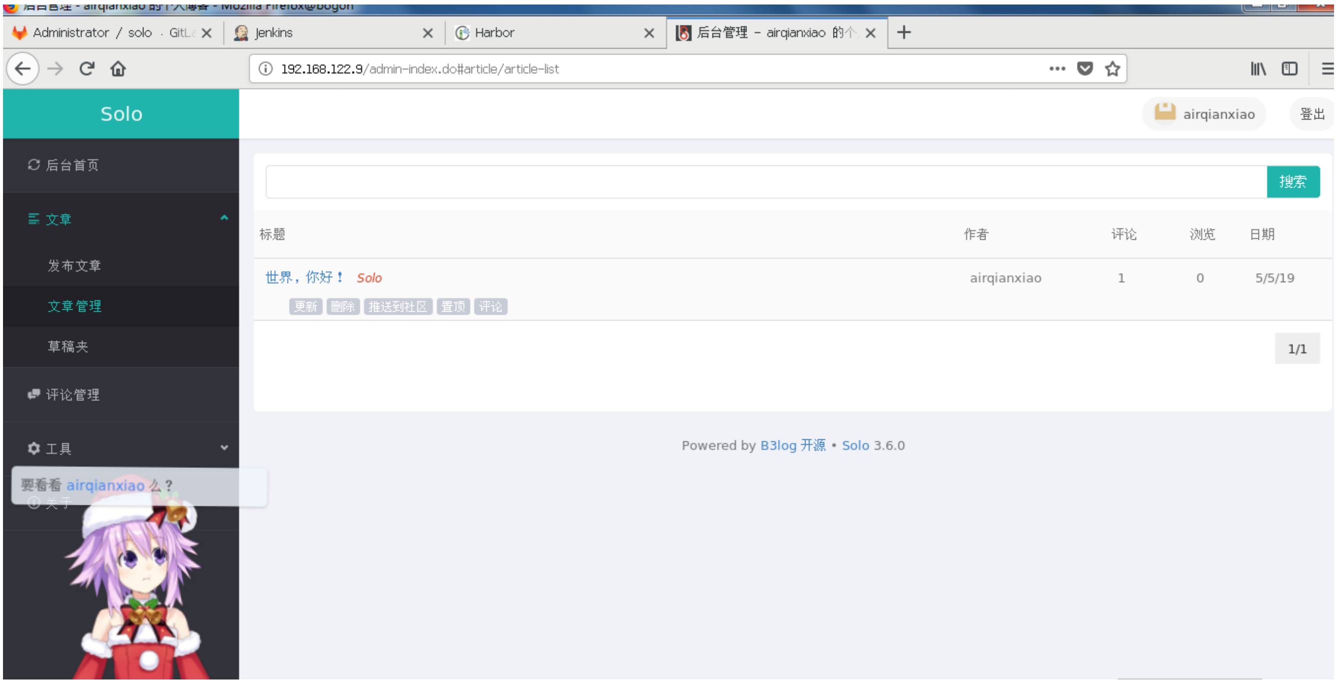
Maven project vlog

需要如下参数用于构建项目:

Tag 1.0.0
1.0.1

tag

访问验证



192.168.122.9/admin-index.do#article/article-list

标题	作者	评论	浏览	日期
世界，你好！ Solo	airqianxiao	1	0	5/5/19

Powered by B3log 开源 • Solo 3.6.0