# gitlab-cicd流水线搭建

Gitlab流水线搭建过程使用docker compose 部署所需容器，需要提前在服务器上安装好docker和docker compose。此处忽略。

Docker 加速

添加到 /etc/docker/daemon.json并重启docker使其生效 sudo systemctl restart docker

{

"registry-mirrors": [

"https://docker.m.daocloud.io"

]

}

另外需要安装git

sudo yum install -y git curl

## 安装gitlab-ce

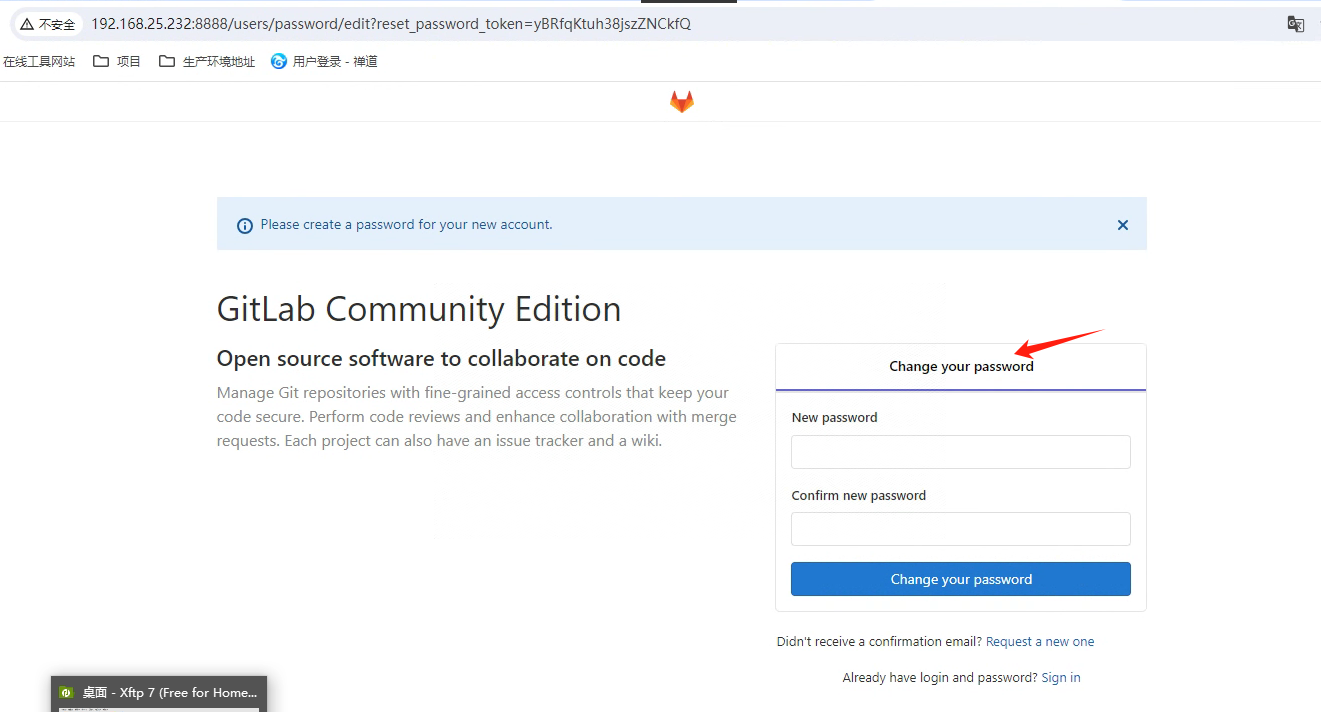
### 执行docker-compose.yml 文件

docker-compose up -d

|  |
| --- |
| version: "3.3"  services:  gitlab:  image: gitlab/gitlab-ce:12.10.14-ce.0  privileged: false  restart: always  ports:  - 8888:8888  - '10022:22'  - 10443:443  volumes:  - ./config:/etc/gitlab  - ./logs:/var/log/gitlab  - ./data:/var/opt/gitlab  environment:  - GITLAB\_OMNIBUS\_CONFIG=external\_url 'http://192.168.25.232:8888/'; gitlab\_rails['lfs\_enabled'] = true; gitlab\_rails['gitlab\_shell\_ssh\_port'] = 10022  deploy:  resources:  limits:  memory: 10240M |

### **登录gitlab**

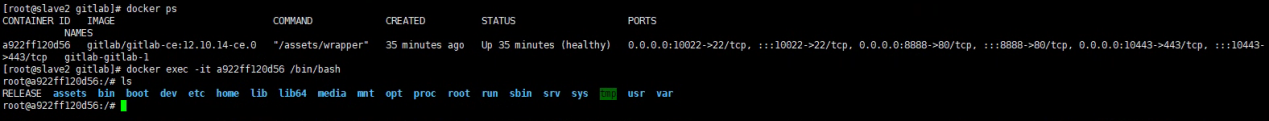
首次安装完成并访问gitlab 在登录页会提示修改root 的密码。修改后以root用户登录



### **手动重置root登录密码**

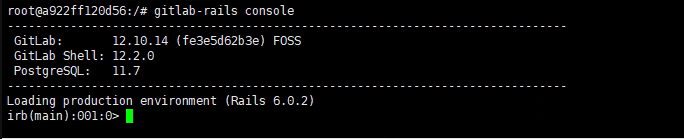
使用SSH登录到容器内部：

docker exec -it gitlab\_gitlab\_1 bash



重置密码：

sudo gitlab-rails console



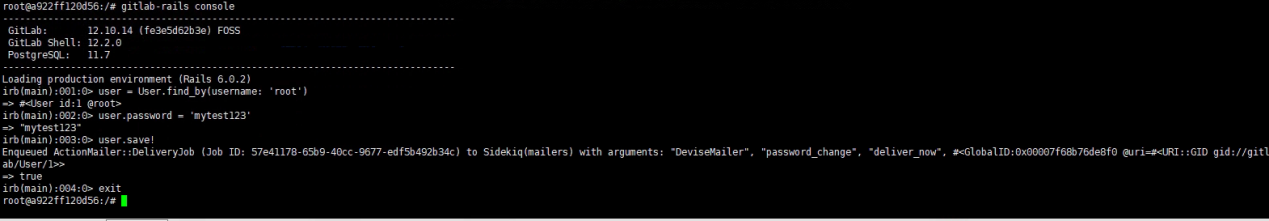
在控制台中执行：

user = User.find\_by(username: 'root')

user.password = 'new\_password'

user.save!

exit



## 安装gitlab-runner和注册runner

以下是在 CentOS 7 下安装 GitLab Runner 的步骤

### **使用官方提供的安装脚本进行安装**

curl -L https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.rpm.sh | sudo bash

### **安装 GitLab Runner**

sudo yum install -y gitlab-runner

### **注册 Runner**

sudo gitlab-runner register

按照提示输入 GitLab 实例的 URL 和注册令牌，并选择适当的运行器类型（如 “Shell”）。

项目配置时以tag进行区分。

### **2.4启动 Runner**

1.启动服务： sudo systemctl start gitlab-runner

2.查看状态： sudo systemctl status gitlab-runner

3.设置开机自启：systemctl enable gitlab-runner

## 3.安装jdk

### 3.1.下载 Java 8 JDK

yum install java-1.8.0-openjdk-devel

### 3.2.检查 Java 安装

java -version

## 4.安装maven

### 4.1**从apache-maven官网下载**

<https://downloads.apache.org/maven/maven-3/3.8.8/source/>

后上传到服务器上进行解压 tar -zxvf [apache-maven-3.8.8-src.tar.gz](https://downloads.apache.org/maven/maven-3/3.8.8/source/apache-maven-3.8.8-src.tar.gz)

### 4.2设置环境变量

echo "export PATH=/opt/apache-maven-3.6.3/bin:$PATH" >> /etc/profile

### 4.3使环境变量生效

source /etc/profile

### 4.4检查 Maven 安装

mvn -version

## 添加 gitlab-runner 用户

1.由于gitlab-runner 作业运行的时候是使用的gitlab-runner命名的用户，所以需要在服务器上添加该用户。

useradd gitlab-runner 这里不需要给该用户设置密码。

passwd newusername 用户设置密码

2.给该用户生成ssh密钥对

sudo su - gitlab-runner

ssh-keygen -t rsa -b 4096 -C ["your\_email@example.com"](mailto:\"your_email@example.com\")

gitlab-runner用户对其主目录及.ssh目录有正确的权限。通常，~/.ssh目录的权限应为700，私钥文件id\_rsa的权限应为600

3.将公钥添加到远程服务器

sudo su - gitlab-runner

ssh-copy-id [root@192.168.25.232](mailto:root@192.168.25.232)

如果ssh的服务器密钥变量或者服务器重置了，使用

ssh-keygen -R 192.168.25.232 命令来删除与该主机相关的条目。然后重新连接以接受新的主机密钥。

### 5.1设置setting.xml

后端项目打包使用到maven,故maven的setting.xml文件需要放在这个用户的主目录下的 .m2 文件夹中。

### 5.2确认 gitlab-runner 用户的主目录

sudo grep gitlab-runner /etc/passwd

假设输出为 gitlab-runner:x:1001:1001::/home/gitlab-runner:/bin/bash，那么主目录就是 /home/gitlab-runner。

### 5.3创建 .m2 目录并复制或创建 settings.xml 文件

### 5.4检查gitlab-runner 用户有执行mvn命令的权限没有

sudo -u gitlab-runner mvn -version

### 5.5需要注意修改下setting.xml localRepository 指定的目录的权限

这里以 /data/repository 本地仓库目录为例

chomd -R 777 repository/

### 5.6 如果gitlab-runner用户不在 docker 组，将其添加到 docker 组

# 查看当前用户所属组

id gitlab-runner

# 如果用户不在 docker 组，将其添加到 docker 组

sudo usermod -aG docker gitlab-runner

## 安装make

Centos yum update && yum install -y make

## 使用nvm来安装node和npm

1.先更新系统

sudo yum update

2.安装nvm

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash

source ~/.bashrc

3.安装node

nvm install 16

# Nexus安装

# Registry安装

添加到 /etc/docker/daemon.json并重启docker使其生效 sudo systemctl restart docker

{

"registry-mirrors": [

"https://docker.m.daocloud.io"

],

"insecure-registries": ["192.168.25.230:5000", "192.168.25.72:5000"]

}