1. **GitLab概述**

GitLab是一个利用Runby on Rails开发的开源应用程序，实现一个自托管的Git项目仓库，可通过Web界面进行访问公开的或者私人项目。

Runby on Rails是一个可以使开发、部署、维护web应用程序变得简单的框架。

GitLab拥有与GitHub类似的功能，能够浏览源代码，管理缺陷和注释。可以管理团队对仓库的访问，它非常易于浏览提交过的版本并提供一个文件历史库。它还提供一个代码片段收集功能可以情商实现代码复用，便于日后有需要的时候进行查找。

1. **搭建GitLab平台**

课题使用的系统是CentOS7 64位。GitLab的下载地址是 <https://packages.gitlab.com/gitlab/gitlab-ce/>

* 1. **安装环境初始化**

yum install curl policycoreutils openssh-server openssh-clients postfix -y

**2.2.默认，使用 Postfix 发送邮件**

service postfix start

chkconfig postfix on

**2.3.关闭防火墙**

iptables -F #清空规则

systemctl stop firewalld #关闭系统防火墙

systemctl disable firewalld #禁用系统防火墙

**2.4.安装gitlab-ce**

官方地址下载比较慢，使用清华的镜像库

wget --content-disposition https://packages.gitlab.com/gitlab/gitlab-ce/packages/el/7/gitlab-ce-11.0.3-ce.0.el7.x86\_64.rpm/download.rpm

rpm -ivh gitlab-ce-11.0.3-ce.0.el7.x86\_64.rpm

**2.5.配置并启动GitLab**

gitlab-ctl reconfigure #重新配置应用程序。相当于初始化一下

netstat -nultp |grep :80

vim /etc/gitlab/gitlab.rb

改：13 external\_url 'http://gitlab.example.com'

为：13 external\_url 'http://172.17.100.11'#本机IP

**2.6.gitlab的管理**

关闭gitlab： # gitlab-ctl stop

启动gitlab： # gitlab-ctl start

重启gitlab： # gitlab-ctl restart

1. **docker安装GitLab**
   1. **拉取镜像**

# gitlab-ce为稳定版本，后面不填写版本则默认pull最新latest版本

$ docker pull gitlab/gitlab-ce

**3.2. 运行GitLab镜像**

$ docker run -d -p 443:443 -p 80:80 -p 222:22 --name gitlab --restart always -v /home/gitlab/config:/etc/gitlab -v /home/gitlab/logs:/var/log/gitlab -v /home/gitlab/data:/var/opt/gitlab gitlab/gitlab-ce

# -d：后台运行

# -p：将容器内部端口向外映射

# --name：命名容器名称

# -v：将容器内数据文件夹或者日志、配置等文件夹挂载到宿主机指定目录

* 1. **配置**

按上面的方式，gitlab容器运行没问题，但在gitlab上创建项目的时候，生成项目的URL访问地址是按容器的hostname来生成的，也就是容器的id。作为gitlab服务器，我们需要一个固定的URL访问地址，于是需要配置gitlab.rb（宿主机路径：/home/gitlab/config/gitlab.rb）。

# gitlab.rb文件内容默认全是注释

$ vim /home/gitlab/config/gitlab.rb

# 配置http协议所使用的访问地址,不加端口号默认为80

external\_url 'http://192.168.199.231'

# 配置ssh协议所使用的访问地址和端口

gitlab\_rails['gitlab\_ssh\_host'] = '192.168.199.231'

gitlab\_rails['gitlab\_shell\_ssh\_port'] = 222

# 此端口是run时22端口映射的222端口:wq #保存配置文件并退出

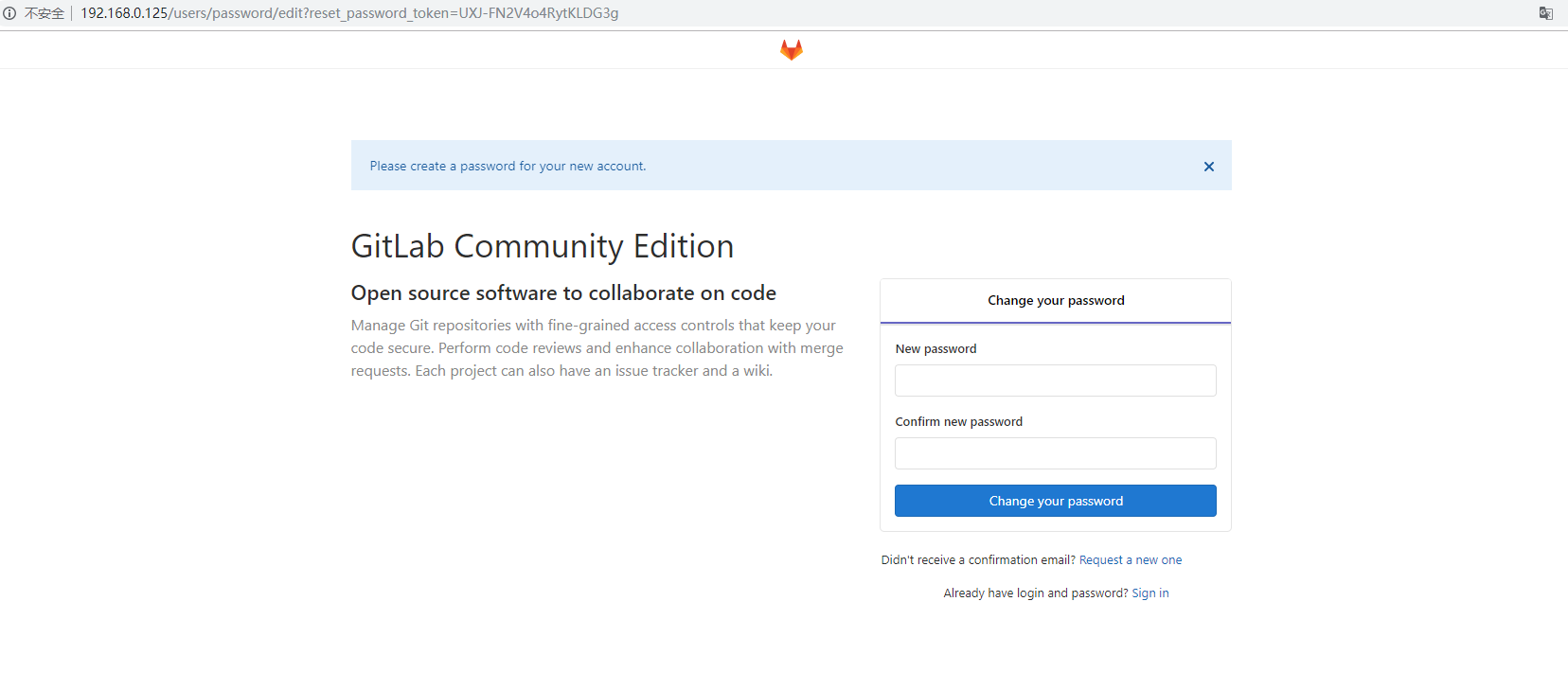
# 重启gitlab容器

$ docker restart gitlab

此时项目的仓库地址就变了。如果ssh端口地址不是默认的22，就会加上ssh:// 协议头  
打开浏览器输入ip地址(因为我的gitlab端口为80，所以浏览器url不用输入端口号，如果端口号不是80，则打开为：ip:端口号)

* 1. **验证**

在浏览器中输入IP地址，出现如下界面则安装成功，首次登陆需要配置root账号的密码。



* 1. **移动镜像**

查看gitlab的容器ID

docker ps -a



打包镜像

docker commit 264510355cde gitlab

docker save gitlab > /tmp/gitlab.tar

将打包好的镜像放入docker

docker load < /urs/local/src/mynewimage.tar

启动gitlab

docker run -d -p 443:443 -p 80:80 -p 222:22 --name gitlab --restart always -v /home/gitlab/config:/etc/gitlab -v /home/gitlab/logs:/var/log/gitlab -v /home/gitlab/data:/var/opt/gitlab gitlab

修改配置

# gitlab.rb文件内容默认全是注释

$ vim /home/gitlab/config/gitlab.rb

# 配置http协议所使用的访问地址,不加端口号默认为80

external\_url 'http://192.168.199.231'

# 配置ssh协议所使用的访问地址和端口

gitlab\_rails['gitlab\_ssh\_host'] = '192.168.199.231'

gitlab\_rails['gitlab\_shell\_ssh\_port'] = 222

# 此端口是run时22端口映射的222端口:wq #保存配置文件并退出

# 重启gitlab容器

$ docker restart gitlab