

基于CentOS7搭建GitLab

亲测有效，转载：<https://www.jianshu.com/p/9baa47a31f46>

一、简介



Git Lab

[GitLab](#)是利用 Ruby on Rails 一个开源的版本管理系统，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。它拥有与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。团队成员可以利用内置的简单聊天程序(Wall)进行交流。它还提供一个代码片段收集功能可以轻松实现代码复用，便于日后有需要的时候进行查找。

二、GitLab和GitHub的区别



image

相同点:

两个都是基于web的Git仓库,而且GitLab在使用方式上和GitHub基本上是一样的,都提供了存储、分享、发布和合作开发项目的中心化云存储场所.

不同点:

GitHub同时提供公共仓库和私有仓库,但如果需要使用私有仓库,是需要收费的.GitLab让开发团队对他们的代码仓库拥有更多的控制,相比于GitHub,它有不少的特色: 允许免费设置仓库权限;允许用户选择分享一个project的部分代码;允许用户设置project的获取权限,进一步的提升安全性;可以设置获取到团队整体的改进进度;通过innersourcing让不在权限范围内的人访问不到该资源。从代码私有性方面来看,有时公司并不希望员工获取到全部的代码,这个时候GitLab无疑是更好的选择。但对于开源项目而言, GitHub依然是代码托管的首选。

三、安装GitLab

我们是基于CentOS7来搭建的,所以同学们先准备好CentOS的环境,能上网就可以了.其实安装步骤在GitLab的官网上都有很详细的介绍,同学们可以到官网按照这步骤来实现就可以了。附上官网安装地址:<https://about.gitlab.com/install/#centos-7>

1.安装和配置必要的依赖

在CentOS 7 (and RedHat/Oracle/Scientific Linux 7),下面这些命令是在防火墙中开放HTTP和SSH的访问

```
sudo yum install -y curl policycoreutils-python openssh-server
sudo systemctl enable sshd
sudo systemctl start sshd
sudo firewall-cmd --permanent --add-service=http
sudo systemctl reload firewalld
```

接着,安装Postfix来实现邮件通知的功能,如果你想使用其他的邮件服务器,可以跳过下面这些命令,你可以再GitLab安装完毕后自行安装其他的邮件服务器。[如何配置其他的邮件服务器文档地址](#)

```
sudo yum install postfix
sudo systemctl enable postfix
sudo systemctl start postfix
```

2.配置GitLab的仓库地址并下载安装包

添加仓库地址

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

安装GitLab

```
yum install -y gitlab-ee
```

执行结果


```
Recipe: <Dynamically Defined Resource>
  * service[gitlab] action restart
    - restart service service[gitlab]
Recipe: gitlab::enable
  * runit_service[gitlab] action hup
    - send hup to runit_service[gitlab]
Recipe: <Dynamically Defined Resource>
  * service[gitlab-workhorse] action restart
    - restart service service[gitlab-workhorse]
  * service[node-exporter] action restart
    - restart service service[node-exporter]
  * service[gitlab-exporter] action restart
    - restart service service[gitlab-exporter]
  * service[redis-exporter] action restart
    - restart service service[redis-exporter]
  * service[prometheus] action restart
    - restart service service[prometheus]
Recipe: monitoring::prometheus
  * execute[reload prometheus] action run
    - execute /opt/gitlab/bin/gitlab-ctl hup prometheus
Recipe: <Dynamically Defined Resource>
  * service[alertmanager] action restart
    - restart service service[alertmanager]
  * service[postgres-exporter] action restart
    - restart service service[postgres-exporter]
  * service[grafana] action restart
    - restart service service[grafana]

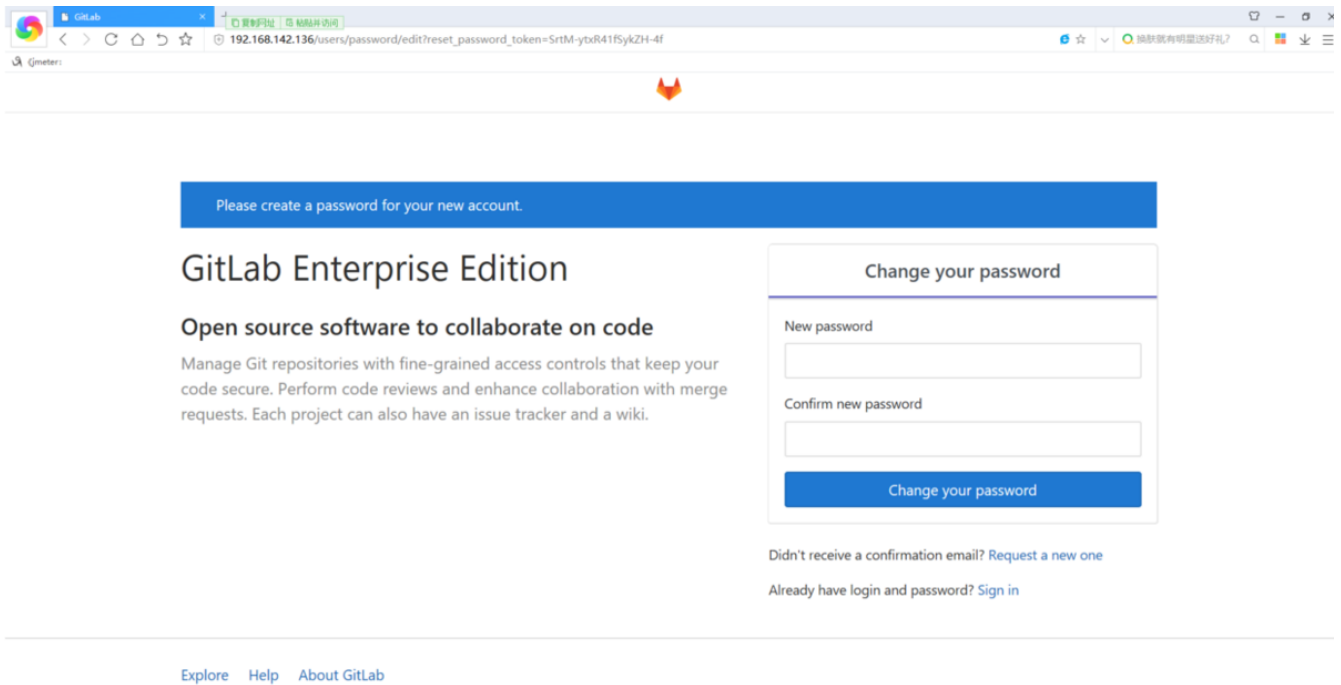
Running handlers:
Running handlers complete
Chef Client finished, 527/1451 resources updated in 04 minutes 47 seconds
gitlab Reconfigured!
[root@localhost ~]#
```

四、其他

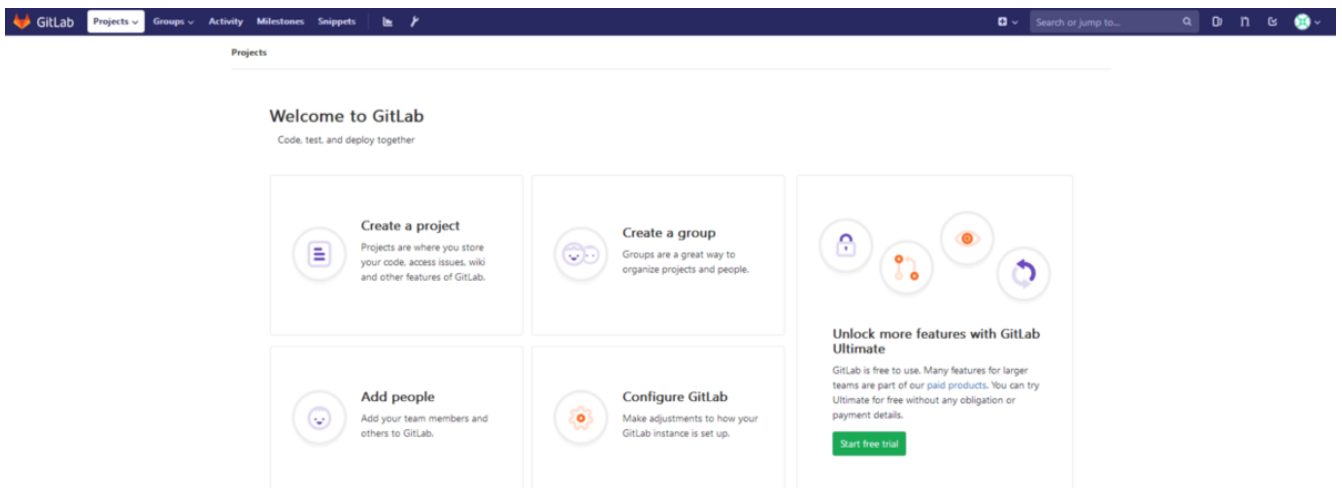
直接在浏览器中访问**虚拟机的地址**即可,会出现如下的界面,说明已经搭建成功了.

GitLab默认账户为 root

给root设置新密码, 然后直接访问虚拟机地址, 就可以到GitLab登录页面。



该页面是让管理员设置初始密码,设置完成之后,通过账户 `root` 就可以登录到GitLab的管理页面了。



image

关于更多的GitLab详细配置可以参考官方文档. <https://docs.gitlab.com/omnibus/README.html#installation-and-configuration-using-omnibus-package>