Gitlab CI/CD

# 1持续集成的原则

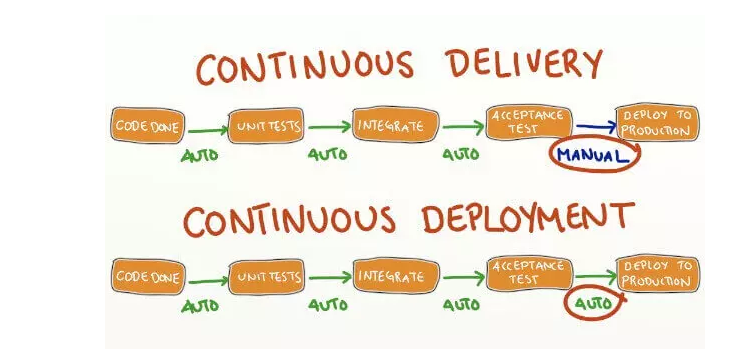
* 需要版本控制软件保障团队成员提交的代码不会导致集成失败
* 开发人员必须及时向版本控制库中提交代码，也必须经常性地从版本控制库中更新代码到本地；
* 需要有专门的集成服务器来执行集成构建。
* 根据项目的具体实际，集成构建可以被软件的修改来直接触发，也可以定时启动，如每半个小时构建一次；必须保证构建的成功。
* 如果构建失败，修复构建过程中的错误是优先级最高的工作。

一旦修复，需要手动启动一次构建。

# 2 持续集成系统的组成

一个完整的构建系统必须包括：

* 一个自动构建过程，包括自动编译、分发、部署和测试等。
* 一个代码存储库，即需要版本控制软件来保障代码的可维护性，同时作为构建过程的素材库。
* 一个持续集成服务器。



# 3 安装& 注册GitLab-Runner

<https://blog.csdn.net/qq_27520051/article/details/80552220>

## 3.1 install

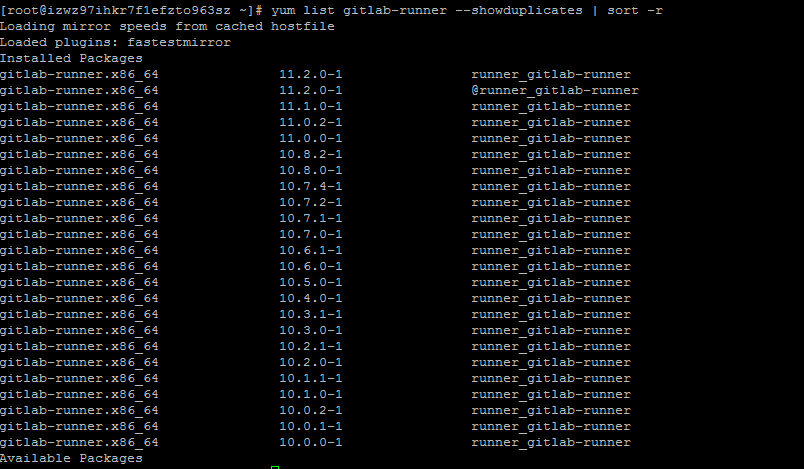
$curl -L https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.rpm.sh | sudo bash

安装最新版的gitlab-runner

$ yum install gitlab-runner

查看版本：

$yum list gitlab-runner --showduplicates | sort -r



## 3.2 Gitlab-Ci运行原理

由以下两个模块组成:

gitlab-ci server

gitlab-ci-runner

其中，gitlab-ci server负责调度、触发Runner,以及获取返回结果. 而gitlab-ci-runner则是主要负责来跑自动化CI（测试，编译，打包等）。

基本流程是： 用户提交代码->检查是否有.gitlab-ci.yml文件->如果无，则结束；-> 如果有，则调用runner执行脚本->获取返回的结果。

# 4 登陆Gitlab

点击“ 重置Runner注册令牌”



图4.1 Runner注册令牌

# 5　运行CI两步：

* Add .gitlab-ci.yml to the root directory of your repository

.gitlab-ci.yml文件内容可以参考项目

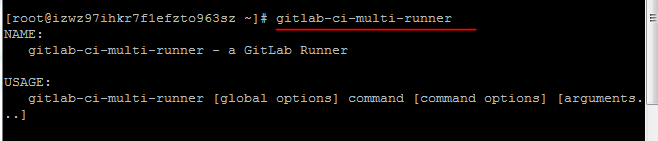
https://gitlab.com/gitlab-org/gitlab-ci-yml

* Configure a Runner

参考文档：

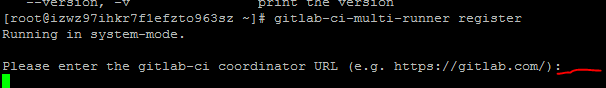
<https://docs.gitlab.com/ee/ci/quick_start/README.html>

Step1 运行：gitlab-ci-multi-runner



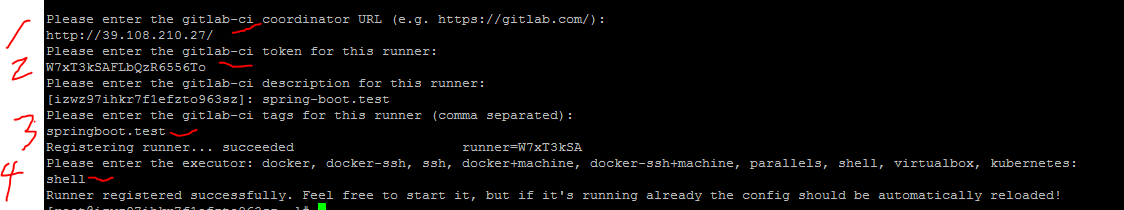
Step2:

$gitlab-ci-multi-runner register

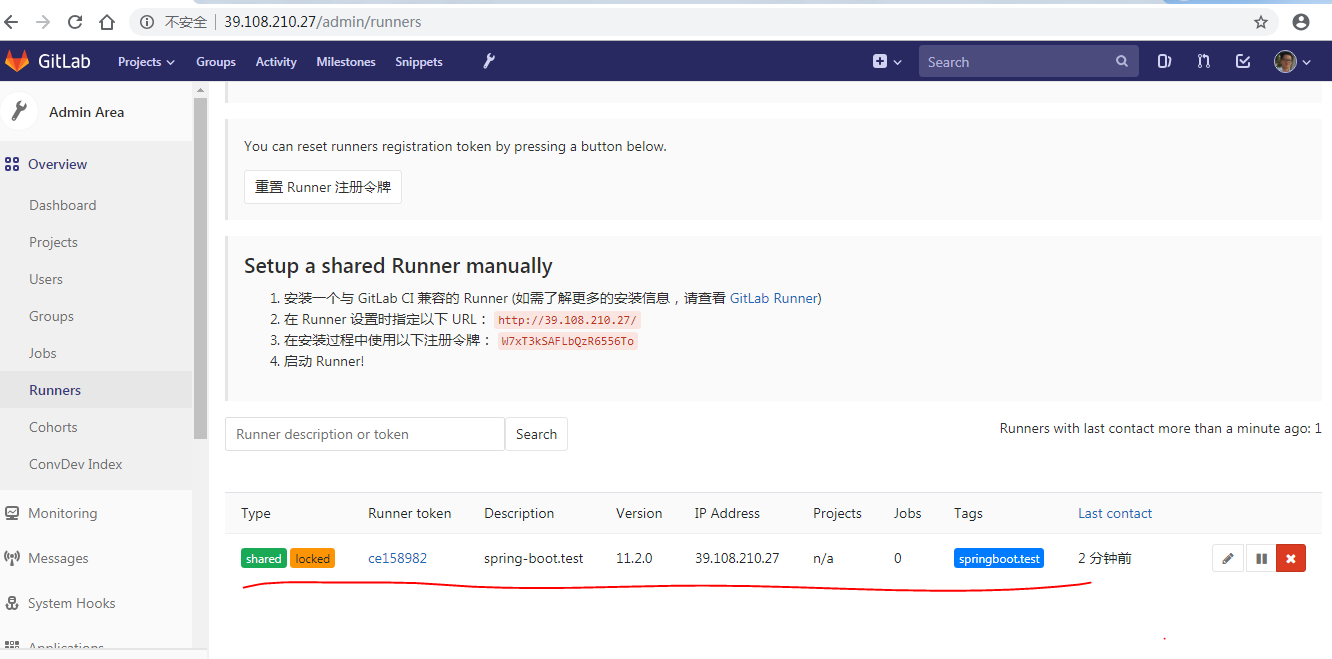


<http://39.108.210.27>

Step3 填写与图4.1 Runner注册令牌一致的Token

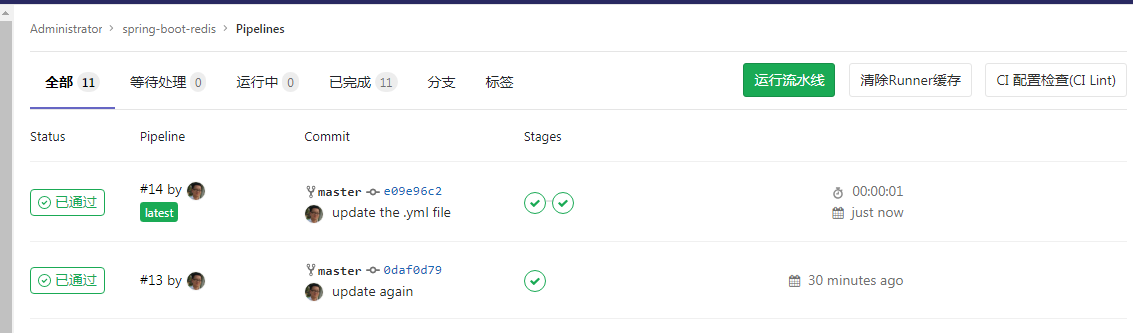


刷新GITLAB管理页面：



<https://www.cnblogs.com/xiaoer/p/8283266.html>

如图所示： 注册了一个Tag为tag-spring-redis的作业。



.gitlab-ci.yml 文件示例：

