分布式版本控制Git(三)

自动化工具系列

大纲

- 搭建linux Git服务器

搭建Git服务器

GitHub是一个免费的服务器，这个网站就是提供Git仓库托管服务的，如果公司开发使用的话，会有几个问题

1 网络不稳定
2 代码是公开的，大家都可以查看

如果你需要一个私有的托管服务，支付每个月7美元购买5个私有库，想要更多的私有仓库则要交更多的钱

最好的方法就是在你的服务器上运行 Git。不仅你能够省去一笔钱，你还能够在你的服务器有更多的操作。

在linux上有两种方式部署git服务器
1 使用名为 GitLab 的 GUI 工具
2 运行一个纯 Git 服务器

搭建GitLab服务器

GitLab 是一个非常优秀的项目，这是一个开源项目，允许用户在自己的服务器上运行类似于 GitHub 的项目管理系统，使用 GitLab 为团队成员或公司运行类似于 GitHub 的服务。您可以使用 GitLab 开发私有项目。

安装之前，请确保你的服务器安装了ssh，防火墙，wget

1 安装 GitLab 之前，需要配置 SMTP 电子邮件服务器，以便 GitLab 可以在需要时随时推送电子邮件。官方推荐使用 Postfix。先在你的服务器上安装 Postfix

```
# yum install postfix                安装命令
# systemctl enable postfix           开机自启动
# systemctl start postfix            启动postfix
```

2添加GitLab镜像源，其他镜像下载地址：https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el7/

```
# wget https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el7/gitlab-ce-10.0.0-ce.0.el7.x86_64.rpm
```

3安装gitlab

```
# rpm -i gitlab-ce-10.0.0-ce.0.el7.x86_64.rpm
```

搭建GitLab服务器

4修改gitlab配置文件指定服务器ip和自定义端口

```
# vim  /etc/gitlab/gitlab.rb
  external_url `http://192.168.1.49:8900`   服务器ip和未被占用的端口
  unicorn['port'] = 8899                    默认8080，设置未被占用端口
```

5 重置gitlab配置文件

```
# gitlab-ctl reconfigure
```

6 重启gitlab

```
# gitlab-ctl restart
```

7 访问gitlab，直接输入上面配置的ip加端口访问即可
初始账户: root 密码:5iveL!fe

搭建GitLab服务器

8 设置gitlab发信功能，发信系统用的默认的postfix，smtp是默认开启的，两个都启用了，两个都不会工作，可以自己选择

a 设置关闭smtp，开启postfix

```
# vim /etc/gitlab/gitlab.rb
  gitlab_rails['smtp_enable'] = false
```

b 关闭postfix，设置开启smtp

https://docs.gitlab.com/omnibus/settings/smtp.html

测试是否可以邮件通知：登录并添加一个用户，我这里使用qq邮箱添加一个用户，登录qq邮箱，可以收到邮件通知

如果收不到，请查看垃圾邮箱或者检查邮件是否被拦截并删除，如果有请添加到白名单并删除用户再重新添加用户就可以收到了，否则请检查邮件日志并做好相关设置

tips：

1 如果gitlab报502错误，一般是权限问题，解决方法：# chmod -R 755 /var/log/gitlab

如果还不行，请检查你的内存，安装使用GitLab需要至少4GB可用内存,否则出现各种诡异的问题,而且在使用过程中也经常会出现500错误.

2 gitlab-ctl reconfigure:

Error executing action `run` on resource 'execute[/opt/gitlab/embedded/bin/initdb -D /var/xxxx ‘

文件/etc/passwd的权限是600,给予644权限后,解决问题

搭建Git服务器

下面我们来说下如何在linux下搭建Git服务器（也可以在windows下搭建git服务器，可以直接使用OpenSSH来完成，但是有一些更小的工具，比如CopSSH也可以进行Git服务器的搭建），但是帐号必须有sudo权限

第一步，安装git
$ sudo apt-get install git

第二步，创建一个git用户，用来运行git服务
$ sudo adduser git

第三步，创建证书登录：
收集所有需要登录的用户的公钥，就是他们自己的id_rsa.pub文件，把所有公钥导入到/home/git/.ssh/authorized_keys文件，一行一个。

搭建Git服务器

第四步，初始化Git仓库
先选定一个目录作为Git仓库，假定是/test/helloGit.git，在/test目录下输入命令：
$ sudo git init --bare helloGit.git
Git就会创建一个裸仓库，裸仓库没有工作区，因为服务器上的Git仓库纯粹是为了共享，所以不让用户直接登录到服务器上去改工作区，并且服务器上的Git仓库通常都以.git结尾。
然后，把owner改为git：
$ sudo chown -R git:git helloGit.git

第五步，禁用shell登录
出于安全考虑，第二步创建的git用户不允许登录shell，这可以通过编辑/etc/passwd文件完成。
找到类似下面的一行：
git:x:1001:1001:,,,:/home/git:/bin/bash
改为：
git:x:1001:1001:,,,:/home/git:/usr/bin/git-shell
这样，git用户可以正常通过ssh使用git，但无法登录shell，因为我们为git用户指定的git-shell每次一登录就自动退出

搭建Git服务器

第六步，克隆远程仓库：
现在，可以通过git clone命令克隆远程仓库了，在各自的电脑上运行：
$ git clone git@server :/test/helloGit.git
剩下的推送就简单了。

管理Git服务器

如果开发团队只有几十号人，把每个人的公钥收集起来放到服务器的/home/git/.ssh/authorized_keys文件里就是可行的。
如果开发团队有几百号人，就没法这么玩了，这时，可以用Gitosis来管理公钥。

因为Git是为Linux源代码托管而开发的，所以Git也继承了开源社区的精神，不支持权限控制。
如果想让Git有一套完善的权限控制，每个人是否有读写权限精确到每个分支甚至每个目录下,那么可以使用Git的钩子（hook），所以，可以在服务器端编写一系列脚本来控制提交等操作，达到权限控制的目的。可以使用Gitolite来管理权限。