

XX 公司

Logo 通过 gitolite 架设 git 仓库

客户

- 项目代码 -

提交至: 客户
所有者: 杨彬
作者: 杨彬
编辑者: 杨彬
版本: 1.0
日期: 2014-04-29

© 2014 XX 公司 版权所有 (保留所有权力)

本文件的内容涉及商业机密, 未经 XX 公司允许, 不得泄露给任何第三方。

免责声明

本文件所有权属于 XX 公司。未经 XX 公司允许，任何个人或组织不得将其复制，保存，转发或用于任何商业用途。本文件内容如有任何改动，将不做特别通知。

本文件只用于信息沟通目的，不能作为 XX 公司对于商品，服务的质量承诺书。XX 公司对本文件的一切内容拥有最终解释权。

注册商标

文件修改履历

版本	修改日	修改人	注释
1.0	2014/01/06	杨彬	初稿

目录

1 安装 Git.....	4
2 Gitolite 服务器端配置.....	4
2.1 添加 git 用户组.....	4
2.2 Gitolite 安装与配置.....	4
2.3 配置管理员密钥.....	5
2.4 权限管理.....	6
2.5 创建代码仓库.....	6
2.6 删除代码仓库.....	7
3 Gitolite 授权控制.....	7
4 附录.....	8
4.1 词汇表.....	8
4.2 引用.....	8

1 安装 Git

Git 详细安装配置请查看 [git_基础教程](#)。

2 Gitolite 服务器端配置

2.1 添加 git 用户组

切换到 root 用户：

添加 git 用户组：

设置 git 用户组的密码：

```
su root
adduser git
sudo passwd git
```

2.2 Gitolite 安装与配置

切换到 git 用户进行以下操作：

```
su git
cd ~
```

递归创建目录：

```
mkdir -p /home/git/bin/gitolite/share/gitolite/conf/ /home/git/bin/gitolite/share/gitolite/hooks/
```

从 Gitolite 远端 git 库下载 Gitolite：

```
git clone git://github.com/ossxp-com/gitolite.git
```

会出现一下提示：

```
Cloning into 'gitolite'...
remote: Counting objects: 6194, done.
remote: Compressing objects: 100% (2709/2709), done.
remote: Total 6194 (delta 3419), reused 6194 (delta 3419)
Receiving objects: 100% (6194/6194), 2.87 MiB | 516 KiB/s, done.
Resolving deltas: 100% (3419/3419), done.
```

进入 /gitolite/src/ 目录，执行以下命令：

```
cd gitolite/src/
./gl-system-install /home/git/bin/gitolite /home/git/bin/gitolite/share/gitolite/conf/
/home/git/bin/gitolite/share/gitolite/hooks/
```

执行后可以看见以下提示信息：

```
***** WARNING *****
gl-setup is not in your $PATH.

Since gl-setup MUST be run from the PATH (and not as src/gl-setup or
such), you must fix this before running gl-setup. Just add

PATH=/home/git/bin/gitolite:$PATH

to the end of your bashrc or similar file. You can even simply do that
manually each time you log in and want to run a gitolite command.
```

修改配置：

```
vi /home/git/bin/gitolite/gl-setup
```

找到：

```
GL_PACKAGE_CONF=/usr/share/gitolite/conf
```

修改为：

```
GL_PACKAGE_CONF=/home/git/bin/gitolite/share/gitolite/conf/
```

添加系统变量并使之生效：

```
echo "PATH=/home/git/bin/gitolite:$PATH" >> ~/.bashrc
source ~/.bashrc
```

2.3 配置管理员密钥

切换到客户端，并在客户端生成 ssh 密钥：

```
cd ~
ssh-keygen
```

注意：生成密钥的时候不需要填写任何数据，一直回车就可以

查看并把 ssh 公钥上传到服务器：

```
cd ~/.ssh
scp id_rsa.pub git@<ip address>:~/
```

切换到服务器做以下操作：

```
su git
cd ~
mv id_rsa.pub <name>.pub
```

注意：为了防止重名，我们以 个人的名字 进行命名。

把公钥添加到 Gitolite:

```
cd /home/git/bin/gitolite/  
./gl-setup ~/<name>.pub
```

敲回车键, 进行编辑自动打开的 .gitolite.rc 文件

找到:

```
$GL_PACKAGE_HOOKS = "/usr/share/gitolite/hooks";
```

修改为:

```
$GL_PACKAGE_HOOKS = "/home/git/bin/gitolite/share/gitolite/hooks";
```

保存后继续执行, 出现以下提示信息:

```
Initialized empty Git repository in /home/git/repositories/gitolite-admin.git/  
creating testing...  
Initialized empty Git repository in /home/git/repositories/testing.git/  
[master (root-commit) e496467] gl-setup /home/git/binyang.pub  
2 files changed, 26 insertions(+)  
create mode 100644 conf/gitolite.conf  
create mode 100644 keydir/binyang.pub
```

2.4 权限管理

从服务器端 clone 一份 gitolite-admin 仓库来管理代码和用户:

```
git clone git@<id_address>:gitolite-admin.git
```

可以看到如下内容:

```
Initialized empty Git repository in /home/deovo/share/gitolite-admin/.git/  
remote: Counting objects: 6, done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 6 (delta 0), reused 0 (delta 0)  
Receiving objects: 100% (6/6), done.
```

注意:

如果提示 Can't locate Time/HiRes.pm in @INC...

则进行如下操作:

```
yum install perl-Time-HiRes
```

我们可以 clone 自带的一个代码仓库进行测试:

```
git clone git@<ip_address>:testing.git
```

因为它是一个空的仓库, 所以我们会看到以下信息:

```
Initialized empty Git repository in /home/deovo/share/testing/.git/
```

warning: You appear to have cloned an empty repository.

2.5 创建代码仓库

我们这了要创建一个叫做 tutorial 的代码仓库：

打开 gitolite-admin 的配置文件

```
cd ~/gitolite-admin
```

```
vi conf/gitolite.conf
```

添加以下配置：

```
repo          tutorial
              RW+  =    @admin
              R    =    @all
```

添加保存后进行提交：

```
git add .
```

```
git commit -m "<注释>"
```

```
git push
```

配置完成后，进入 tutorial 目录，并执行以下操作：

```
cd tutorial
```

```
git init
```

```
git remote add origin git@<id_address>:tutorial.git
```

```
git add .
```

```
git commit -m "<注释>"
```

```
git push origin master
```

注意：

第一次提交需要使用，`git push origin master`，后面则直接使用 `git push` 就可以。

完成以上操作后一个版本仓库就创建好了，其他客户端可以进行 clone 工作了，clone 命令是：

```
git clone git@<ip_address>:tutorial.git
```

2.6 删除代码仓库

在这里我们删除上面创建的 tutorial 仓库：

删除代码仓库需要两步：

1. 编辑 gitolite-admin/conf/gitolite.conf

```
vi gitolite-admin/conf/gitolite.conf
```

删除 tutorial 相关的 repo 代码，然后进行：

```
git add .
```

```
git commit -m "注释"  
git push
```

2. 登陆服务器, 把/home/git/repositories 下 tutorial 相关的目录删除, 库对应的目录应该是 tutorial.git

```
cd /home/git/repositories  
rm -rf tutorial.git
```

3 Gitolite 授权控制

权限解释:

1. @admin = king yangbin
解释定义了用户组@admin, 用户组包含两个用户 king 和 yangbin
2. repo tutorial
RW+ = king
定义了 tutorial 版本库, 并制定了只有用户 king 才能访问, 并且拥有强制更新的权限
3. repo tutorial
RW = @admin
定义了 tutorial 版本库, 并制定了只有用户组@admin 才能访问, 并且拥有读写的权限

4 附录

4.1 词汇表

词汇	解释

附录 1: 词汇表

4.2 引用

引用	注释
通过 gitolite 架设 git 仓库	http://my.oschina.net/u/183217/blog/127141
Gitolite 的授权介绍	http://jiangli.easymorse.com/?p=477

附录 2: 引用