

Mac 下配置 Git 服务器

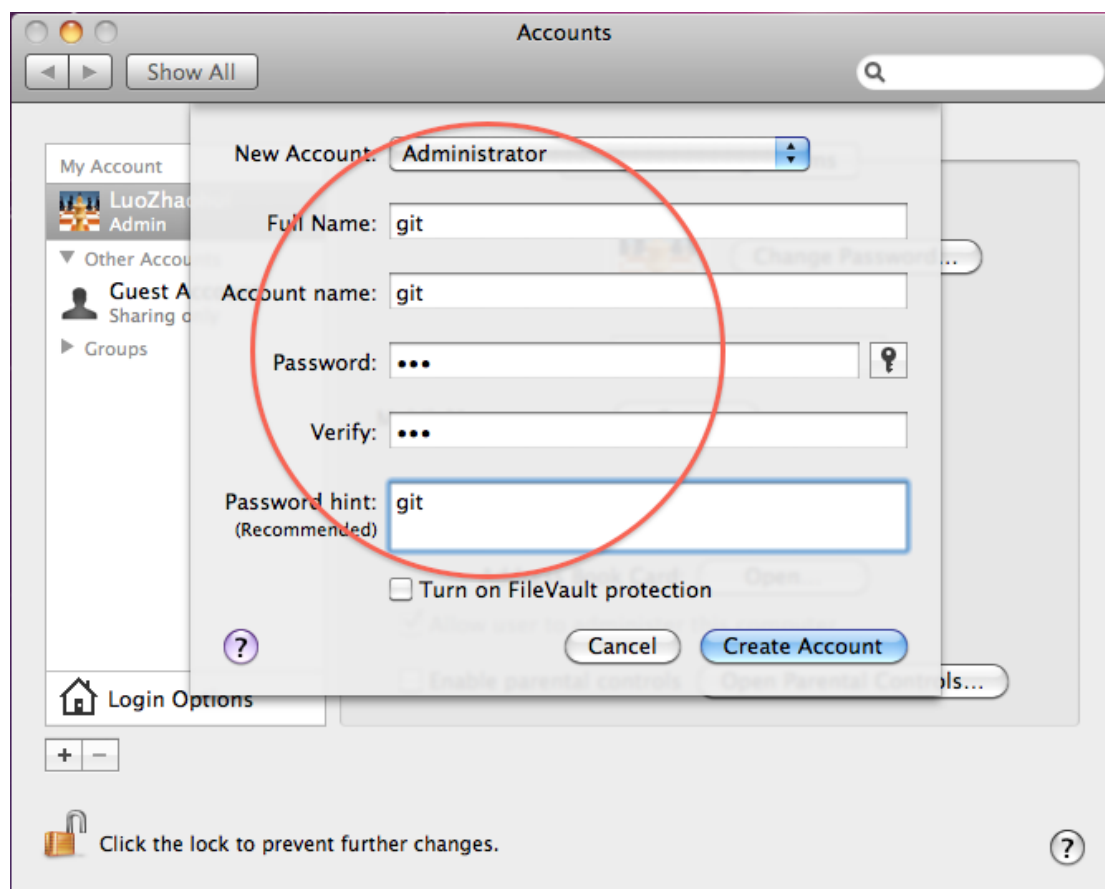
罗朝辉(kesalin@gmail.com)

环境: Mac 版本: Snow 10.6

思路: XCode 默认支持 Git 作为代码仓库, 当我们新建一个仓库的时候, 可以勾选创建默认仓库, 只不过这个仓库是在本地的。本文介绍如何在 mac 机器上创建 Git 服务器, 总体思路是: 使用 [gitosis](#) 来简化创建过程, 在用做服务器的机器上创建一个名为 git 的账户来创建 git 服务器, 其他客户端通过 ssh 机制访问 git 服务器。

一, 创建 git 账户

- 1, 在用做服务器的机器 Server 上创建 git 账户。我们可以通过 System Preferences->accounts 来添加。在这里我添加一个 git 的 administrator 账户, administrator 不是必须的, 在这里仅仅为了方便。



- 2, 设置远程访问

logout 当前账户, 使用 git 账户登录; 在 System Preferences->Sharing 中, 勾选: Web Sharing 和 Remote Logig。

二, 下载安装 gitosis

- 1, Mac Snow 默认已经为我们安装了 Git 和 Python, 可以使用如下命令查看其版本信息:

```
yourname:~ git$ git --version
git version 1.7.3.4
yourname:~ git$ python --version
Python 2.6.1
```

2, 通过命令 "git clone git://eagain.net/gitosis.git" 来下载 gitosis

```
yourname:~ git$ git clone git://eagain.net/gitosis.git

Cloning into gitosis ...

remote: Counting objects: 614, done.
remote: Compressing objects: 100% (183/183), done.
remote: Total 614 (delta 434), reused 594 (delta 422)
Receiving objects: 100% (614/614), 93.82 KiB | 45 KiB/s, done.
Resolving deltas: 100% (434/434), done.
```

3, 进入 gitosis 目录, 使用命令 "sudo python setup.py install" 来执行 python 脚本来安装 gitosis。

```
yourname:~ git$ cd gitosis/
yourname:gitosis git$ sudo python setup.py install
running install
running bdist_egg
running egg_info
creating gitosis.egg-info
.....
Using /Library/Python/2.6/site-packages/setuptools-0.6c9-py2.6.egg
Finished processing dependencies for gitosis==0.2
```

三, 制作 ssh rsa 公钥

1, 回到 client 机器上, 制作 ssh 公钥。在这里我的使用同一台机器上的另一个账户作为 client。如果作为 client 的机器与作为 server 的机器不是同一台, 也是类型的流程: 制作公钥, 放置到服务的 /tmp 目录下。只不过在同一台机器上, 我们可以通过开启另一个 terminal, 使用 su 切换到 local 账户就可以同时操作两个账户。

```
yourname:~ git$ su local_account
Password:
bash-3.2$ cd ~
bash-3.2$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/local_account/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/local_account/.ssh/id_rsa.
Your public key has been saved in /Users/local_account/.ssh/id_rsa.pub.

bash-3.2$ cd .ssh
bash-3.2$ ls
id_rsa      id_rsa.pub
bash-3.2$ cp id_rsa.pub /tmp/yourame.pub
```

在上面的命令里，首先通过 `su` 切换到 `local` 账户（只有在同一台机器上才有效），然后进入到 `local` 账户的 `home` 目录，使用 `ssh-keygen -t rsa` 生成 `id_rsa.pub`，最后将该文件拷贝放置到 `/tmp/yourname.pub`，这样 `git` 账户就可以访问 `yourname.pub` 了，在这里改名是为了便于在 `git` 中辨识多个 `client`。

四，使用 `ssh` 公钥初始化 `gitosis`

1，不论你是以那种方式（邮件，`usb` 等等）拷贝 `yourname.pub` 至服务器的 `/tmp/yourname.pub`。下面的流程都是一样，登入服务器机器的 `git` 账户，进入先前提到 `gitosis` 目录，进行如下操作初始化 `gitosis`，初始化完成后，会在 `git` 的 `home` 下创建 `repositories` 目录。

```
yourname:gitosis git$ sudo -H -u git gitosis-init < /tmp/yourname.pub
Initialized empty Git repository in /Users/git/repositories/gitosis-admin.git/
Reinitialized existing Git repository in /Users/git/repositories/gitosis-admin.git/
```

2，在这里，会将该 `client` 当做认证受信任的账户，因此在 `git` 的 `home` 目录下会有记录，文件 `authorized_keys` 的内容与 `yourname.pub` 差不多。

```
yourname:~ git$ cd ~
yourname:~ git$ cd .ssh
yourname:.ssh git$ ls
authorized_keys
```

我们需要将 `authorized_keys` 稍做修改，用文本编辑器打开它，删除里面的 `"command="gitosis-serve yourname",no-port-forwarding,no-X11-forwarding,no-agent-forwarding,no-pty "` 这一行：

```
yourname:.ssh git$ open -e authorized_keys
```

3，然后，我们对 `post-update` 赋予可写权限，以便 `client` 端可以提交更改。

```
yourname:gitosis git$ sudo chmod 755 /Users/git/repositories/gitosis-admin.git/hooks/post-update
Password:
yourname:.ssh git$ cd ~
yourname:~ git$ cd repositories/
yourname:repositories git$ ls
gitosis-admin.git
yourname:repositories git$
```

在上面的命令中可以看到，`gitosis` 也是作为仓库的形式给出，我们可以在其他账户下 `checkout`，然后对 `gitosis` 进行配置管理等等，而无需使用服务器的 `git` 账户进行。

4，最后一步，修改 `git` 账户的 `PATH` 路径。

```
yourname:gitosis git$ touch ~/.bashrc
yourname:gitosis git$ echo PATH=/usr/local/bin:/usr/local/git/bin:$PATH > .bashrc
yourname:gitosis git$ echo export PATH >> .bashrc
yourname:gitosis git$ cat .bashrc
PATH=/usr/local/bin:/usr/local/git/bin:$PATH
export PATH
```

至此，服务器的配置完成。

五、client 配置

1, 回到 local 账户，首先在 terminal 输入如下命令修改 local 的 git 配置:

```
bash-3.2$ git config --global user.name "yourgitname"
bash-3.2$ git config --global user.email "yourmail@yourcom.com"
```

2, 测试服务器是否连接正确，将 10.1.4.211 换成你服务的名称或服务器地址即可。

```
yourname:~ local_account$ ssh git@10.1.4.211
Last login: Mon Nov 7 13:11:38 2011 from 10.1.4.211
```

3, 在本地 clone 服务器仓库，下面以 gitosis-admin.git 为例:

```
bash-3.2$ git clone git@10.1.4.211:repositories/gitosis-admin.git

Cloning into gitosis-admin ...
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 5 (delta 0)
Receiving objects: 100% (5/5), done.
bash-3.2$ ls
Desktop      InstallApp  Music       Sites
Documents    Library     Pictures    gitosis-admin
Downloads    Movies      Public
bash-3.2$ git
```

在上面的输出中可以看到，我们已经成功 clone 服务器的 gitosis-admin 仓库至本地了。

4, 在本地管理 gitosis-admin:

进入 gitosis-admin 目录，我们来查看一下其目录结构: gitosis.conf 文件是一个配置文件，里面定义哪些用户可以访问哪些仓库，我们可以修改这个配置; keydir 是存放 ssh 公钥的地方。

```
bash-3.2$ cd gitosis-admin/
bash-3.2$ ls
gitosis.conf keydir
bash-3.2$ cd keydir/
bash-3.2$ ls
yourname.pub
```

我们只需要将其他 client 产生的 ssh 公钥添加到 keydir 目录下，并在 gitosis.conf 文件中配置这些用户可以访问的仓库(用户名与放置在 keydir 下 sh 公钥名相同，这就是在前面我们要修改 ssh 公钥名的原因)，然后将改动提交至服务器，这样就可以让其他的 client 端访问服务器的代码仓库了。