

title: Android编译集成之gitolite

date: 2018-06-14 15:01:14

tags: [CI,gitolite]

在Git服务管理领域，主要有三种流行的方案：

Gitosis - 轻量级，开源项目，使用SSH公钥认证，只能做到库级的权限控制，目前项目已经停止开发，不再维护。

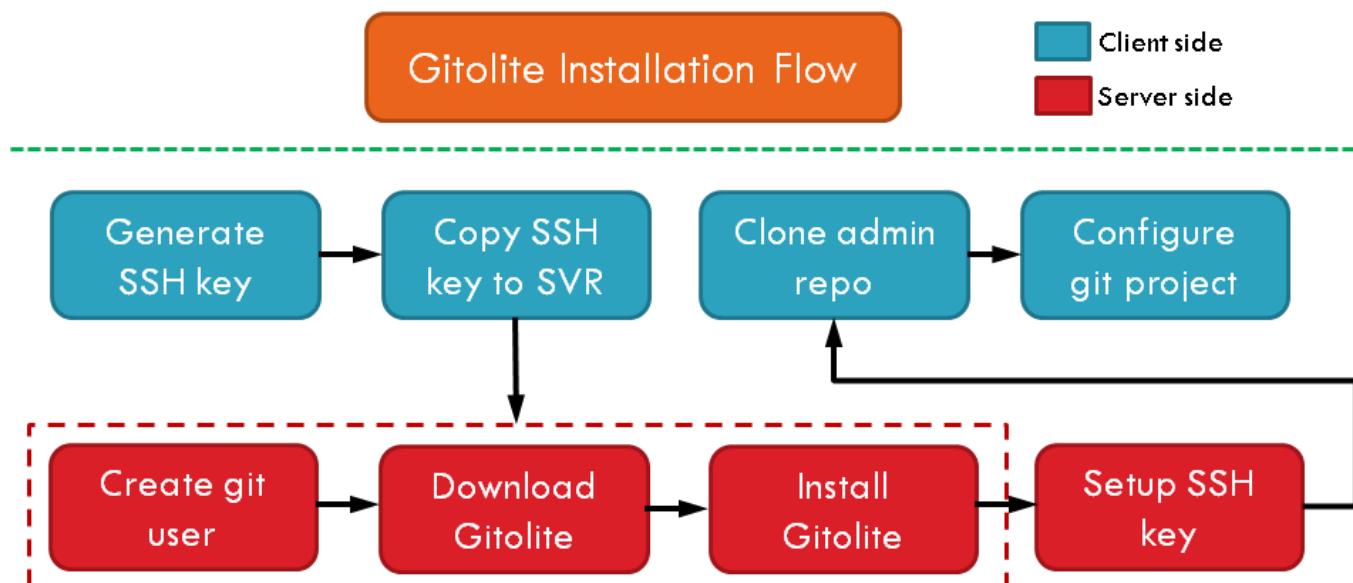
Gitolite - 轻量级，开源项目，使用SSH公钥认证，能做到分支级的权限控制。

Git + Repo + Gerrit - 超级重量级，集版本控制，库管理和代码审核为一身，可管理大型及超大型项目。

Gitolite是在Git之上的一个授权层，依托sshd或者httpd来进行认证。Gitolite允许你定义访问许可不只作用于仓库，同样可作用于仓库中的每个branch和tag name，可确切定义用户(或一组人)只能push特定的“refs”(或者branches或者tags)。

安装

Gitolite的安装需要分别在客户端和服务端操作，搭建Gitolite的步骤如下图所示。



客户端

1. 安装openssh

```
$ sudo apt-get -y install openssh-client
```

2. 安装git

```
sudo apt-get -y install git
```

3. 生成SSH key

```
$ ssh-keygen -t rsa -C HF
```

运行命令后，一直按回车键直到SSH key生成。

4. 上传SSH public key到服务器

```
$ mv ~/.ssh/id_rsa.pub username.pub  
$ scp username.pub servername@server:/tmp
```

username及server根据实际情况填写。

服务器

1. 安装openssh

```
$ sudo apt-get -y install openssh-server openssh-client
```

2. 安装git

```
$ sudo apt-get -y install git
```

3. 安装gitolite

```
$ sudo apt-get -y install gitolite
```

4. 创建gitolite用户

```
$ sudo adduser --system --shell /bin/bash --group --disabled-password  
--home /home/gitolite gitolite
```

5. 初始化服务

```
$ sudo chmod 777 /tmp/username.pub  
$ sudo su - gitolite  
$ gl-setup /tmp/username.pub
```

上面的命令执行作用：

- a)建立目录“~/repositories”，用来存储所有git仓库。
- b)把公钥文件username.pub添加到 ~/.ssh/authorized_keys。

配置

gitolite的配置文件本身就是一个git仓库，gitolite的配置修改就是通过git实现的，先clone下来修改，再push上去。只要push上去，gitolite会立刻监视到配置文件的修改并应用。

管理库

首先，在管理客户端clone管理库到本地。

```
$ git clone gitolite@server:gitolite-admin.git
```

管理库里的目录里有conf/和keydir/两个目录。

- conf/gitolite.conf 用于Git项目配置，访问权限设置。
- keydir/ 用于存储用户的SSH public key(公钥)。

如下所示，“username”(之前用gl-setup命令时候的pubkey名称)具有读写权限而且在gitolite-admin仓库里有一个同名的公钥文件。

```
$ cat conf/gitolite.conf

repo gitolite-admin
    RW+               = username
```

配置

1. 用户/仓库分组

```
@oss_repos      = linux perl rakudo git gitolite
@secret_repos   = fenestra pear

@admin          = scott
@interns        = ashok
```

```
@engineers      = sitaram dilbert wally alice
@staff          = @admins @engineers @interns
```

在下面的例子里，实习生可以push“int”分支。工程师可以push任何有“eng-”开头的branch，还有refs/tags下面用“rc”开头的后面跟数字的。而且管理员可以随便更改(包括rewind)对任何参考名。

```
repo @oss_repos
  RW int$              = @interns
  RW eng-              = @engineers
  RW refs/tags/rc[0-9] = @engineers
  RW+                  = @admins
```

2.GL_WILDREPOS

在gitolite里修改权限，添加正则表达式后提交

```
repo test/.*$
R = test
```

提示 warning：

remote：bad reponame ‘test/.*\$’ or you forgot to set \$GL_WILDREPOS

修改服务器/home/git/.gitolite.rc设置\$GL_WILDREPOS=0为\$GL_WILDREPOS=1即可