**代码服务器搭建步骤**

# 1：名词解释

S ：服务器端(Server)

C ：客户端(Client)

# 2：安装必要软件

1. 安装所需软件(S)



sudo apt-get install git-core openssh-server openssh-client

2、安装配置gisosis(该步可不配置)

2.1、下载gitosis(S)

mkdir ~/gitosis\_setup

cd ~/gitosis\_setup

git clone git://eagain.net/gitosis

如果克隆失败，用下面的地址：

git clone http://github.com/res0nat0r/gitosis.git

2.2、安装gitosis(S)

cd gitosis

sudo python setup.py install

如果python setup.py install失败，需要安装python-setuptools：

sudo apt-get install python-setuptools

2.3、为gitosis创建系统用户(S)

创建一个禁用密码的用户git:

sudo adduser --system --shell /bin/sh -gecos ‘git SCM user’ --group --disabled-password --home /home/git git

2.4、创建配置gitosis管理员(C)

不要用服务器上的用户作为管理员，这里可以使用team leader作为管理员，来管理用户权限配置，这里使用team leader的admin用户作为gitosis的管理员。

2.4.1、生成admin端的公钥(C)

ssh-keygen -t rsa

根据提示输入一些信息之后，会在/home/admin/.ssh/下生成id\_rsa.pub，如下图：



然后将id\_rsa.pub拷贝到服务器(S)的tmp目录下：

scp ~/.ssh/id\_rsa.pub sykean@192.168.1.252:/tmp/id\_rsa\_admin.pub

2.4.2、执行gitosis-init(S)

cd /tmp

sudo chmod 777 id\_rsa\_admin.pub

sudo -H -u git gitosis-init < id\_rsa\_admin.pub

此时，会在/home/git目录下生成一些目录，如果其他用户需要clone gitosis-admin.git的权限，需要执行以下操作：

sudo chmod 755 /home/git/repositories/gitosis-admin.git/hooks/post-update

2.4.3、从服务器(S)克隆gitosis-admin.git

此操作在客户端完成

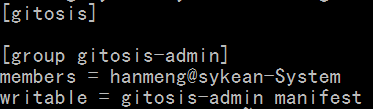
mkdir -p ~/git/

cd ~/git/

git clone [git@192.168.1.252:gisosis-admin.git](mailto:git@192.168.1.252:gisosis-admin.git)

cd gitosis-admin.git

其中，gitosis.conf是gitosis的配置文件，用于配置用户和用户所拥有的权限，keydir是所有组成员的公钥。如果要添加用户，则根据2.4.1章节生成其公钥拷贝到keydir目录下，查看gitosis.conf配置文件如下：



gitosis-admin指示管理员用户组，有用户hanmeng@sykean-System，其可以读写gitosis-admin和manifest仓库。manifest后面介绍。

至此，gitosis配置完成，后面介绍如何添加组成员。

# 配置git-daemon-run

git-daemon-run是一个脚本管理工具，用来启动git-daemon

1. 安装git-daemon-run(S)

apt-get install git-daemon-run

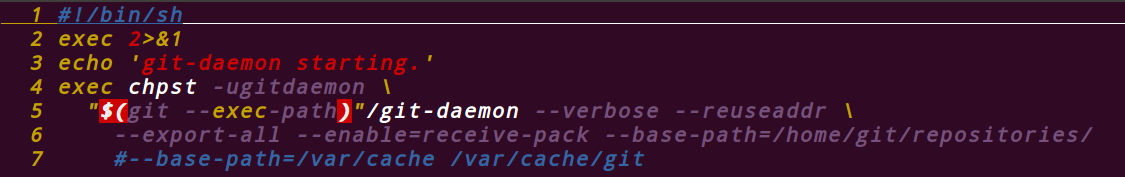
1. 配置git-daemon-run(S)

vim /etc/sv/git-daemon/run

将最后一行修改为：

--export-all --enable=receive-pack --base-path=/home/git/repositories/

修改完结果如下：



1. 重启git daemon(S)

sudo sv stop git-daemon

sudo sv start git-daemon

或者：

sudo sv restart git-daemon

# 搭建repo服务器

1. 配置repo

1.1、下载git-repo.git(C)

mkdir -p ~/gitCfg

cd ~/gifCfg

git clone <http://gerrit.googlesource.com/git-repo> git-repo.git

如果上面的地址无法连接，可以尝试下面的地址：

git clone <http://review.mfunz.com/git-repo> git-repo.git

1.2、配置repo(C)

1.2.1、将git-repo.git目录下的repo拷贝到/usr/bin下面：

cd git-repo.git

sudo cp repo /usr/bin/repo

1.2.2、修改repo

1. 创建版本根目录(S)

进入repositories初始化manifest仓库

cd repositories

git init --bare manifest.git

1. 配置default.xml文件(C)

default.xml是repo用来管理多个git仓库的。

3.1、克隆manifest.git

mkdir ~/repo

cd ~/repo

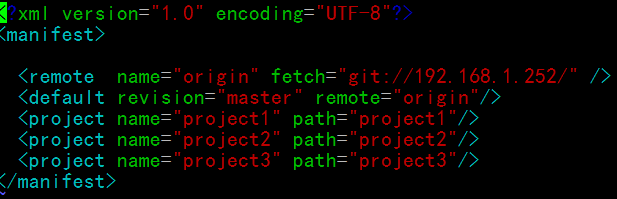
git clone git@192.168.1.252:manifest.git

3.2、配置default.xml

cd manifest

vim default.xml

这里配置如下：(创建三个测试仓库：project1、project2、project3)



将default.xml保存push到远程仓库：

git add default.xml

git commit -m “Init default.xml” default.xml

git push origin master

1. 建立测试仓库(S)

cd /home/git/repositories/

git init --bare project1.git

git init --bare project1.git

git init --bare project1.git

修改仓库的文件属性：

sudo chmod 777 -R　/home/git/repositories/project1.git project2.git project3.git

1. 上传代码到仓库(C)

cd /home/git/repositories/

git clone [git@192.168.1.252:project1.git](mailto:git@192.168.1.252:project1.git)

git clone git@192.168.1.252:project2.git

git clone git@192.168.1.252:project3.git

然后在对应仓库添加代码，并推送到远程服务器

1. 测试repo服务器

mkdir ~/repotest

cd ~/repotest

repo init -u [git@192.168.1.252:manifest.git](mailto:git@192.168.1.252:manifest.git)

repo sync

这样就会根据default.xml配置的project拉下来三个代码仓库。

# Gerrit搭建

1. 建立gerrit账户(S)

sudo adduser gerrit

sudo passwd gerrit

su gerrit

1. 配置java环境

oracle官网下载对应平台的jdk并安装，这里我下载：jdk-7u45-linux-i586.tar.gz

使用下面命令安装：

sudo tar zxvf ./jdk-7u45-linux-i586.tar.gz -C /opt

然后配置环境变量：

vim ~/.bashrc

添加如下内容：

JAVA\_HOME=/opt/jdk1.7.0\_45

export JRE\_HOME=$JAVA\_HOME/jre

export CLASSPATH=$JAVA\_HOME/lib:$JRE\_HOME/lib:$CLASSPATH

export PATH=$JAVA\_HOME/bin:$JRE\_HOME/bin:$PATH

1. 下载gerrit(S)

这里我直接在百度搜的某个固定版本，gerrit的war包，通常包名为gerrit-xxx.war

,xxx为版本号，这里我下载gerrit-2.11.war

1. 安装gerrit

java -jar gerrit-2.11.war init -d review\_site

基本上一路回车就可以，注意其中身份验证时输入http，此认证方式需要配置apache的反向代理，并在apache中配置web站点的口令认证，通过口令认证后gerrit在创建账号的过程中会询问用户的邮件地址并发送邮件。gerrit数据库选择默认H2即可，

Behind reverse proxy           [y/N]? y

这里配置反向代理，选择y

1. 建立存放代码的目录

mkdir ~/GerritSrc

1. 配置gerrit

vim review\_site/etc/gerrit.config

[gerrit]

basePath = git

canonicalWebUrl = http://192.168.1.252:8081/

[database]

type = h2

database = db/ReviewDB

[index]

type = LUCENE

[auth]

type = HTTP

[sendemail]

smtpServer = smtp.exmail.qq.com

smtpServerPort = 465

smtpEncryption = SSL

smtpUser = hanmeng@sykean.com

smtpPass = Sunny1214

from = hanmeng@sykean.com

[container]

user = gerrit

javaHome = /opt/jdk1.7.0\_45/jre

[sshd]

listenAddress = \*:29418

[httpd]

listenUrl = proxy-http://\*:8081/

[cache]

directory = cache

1. 安装apache2服务器

sudo apt-get install --reinstall apache2  apache2.2-common

1. 配置apache2服务器

8.1、开启代理功能、复写功能

cd /etc/apache2/mods-enabled  
ln -s ../mods-available/proxy.load   
ln -s ../mods-available/proxy.conf  
ln -s ../mods-available/proxy\_http.load  
ln -s ../mods-available/proxy\_balancer.conf  
ln -s ../mods-available/proxy\_balancer.load  
ln -s ../mods-available/rewrite.load  
ln -s ../mods-available/ssl.conf  
ln -s ../mods-available/ssl.load

8.2、配置apache2的监听端口

vim /etc/apache2/port.conf

NameVirtualHost \*:80

NameVirtualHost \*:8080 //添加

Listen 80

Listen 8080 //添加

8.3、配置反向代理

sudo vim /etc/apache2/httpd.conf

<VirtualHost \*:8080>

ServerName 192.168.1.252

ProxyRequests Off

ProxyVia Off

ProxyPreserveHost On

AllowEncodedSlashes On

RewriteEngine On

RewriteRule ^/(.\*) http://192.168.1.252:8081/$1 [NE,P]

<Proxy \*>

Order deny,allow

Allow from all

</Proxy>

<Location /login/>

AuthType Basic

AuthName "Gerrit Code Review"

Require valid-user

AuthBasicProvider file

AuthUserFile /home/gerrit/review\_site/etc/passwd

</Location>

ProxyPass / http://192.168.1.252:8081/

</VirtualHost>

8.4、主机配置文件中加入httpd.conf，使其生效

sudo vim /etc/apache2/apache2.conf

加入include httpd.conf 如果有就不用加了

1. 设置第一个gerrit用户的账号和密码

touch ./review\_site/etc/passwd

htpasswd -b ./review\_site/etc/passwd admin admin

(后续再添加gerrit用户可使用: $: htpasswd -b ./review\_site/etc/passwd UserName PassWord )

1. 开启gerrit服务器

./review\_site/bin/gerrit.sh start

1. 重启apache2服务器

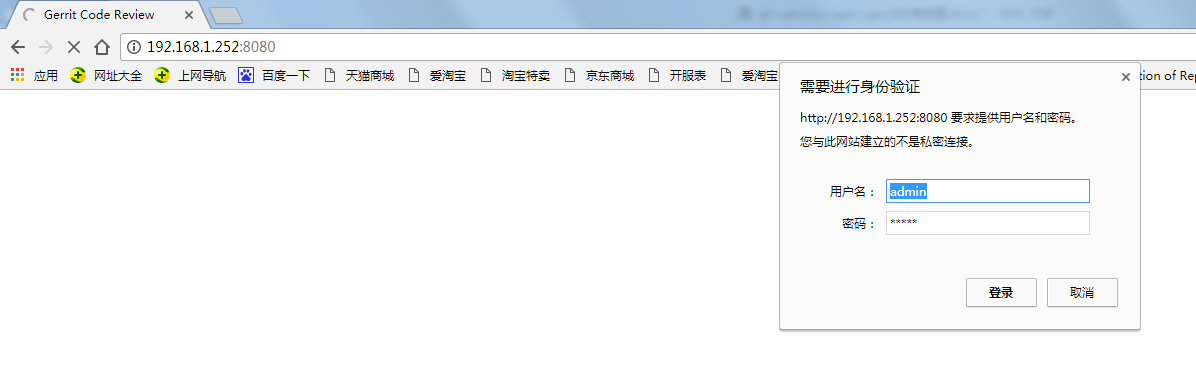
sudo /etc/init.d/apache2 restart

1. 访问http://192.168.1.252:8080/,用刚才创建的admin用户登录，系统自动会将该用户注册为gerrit的账户。(注：第一个登陆的用户自动注册为管理员（gerrit账号ID为1000000的账户）)

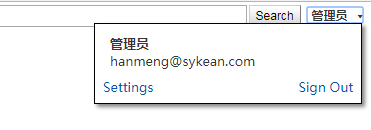
注意：用htpasswd命令创建HTTP认证时，并没有在gerrit数据库中创建账号信息，当登陆成功后，gerrit会自动创建同名的gerrit用户。如果在htpasswd命令创建的账号之前，gerrit数据库中已经存在同名的gerrit用户（如：通过直接操作数据库添加的），那么web登录时，不会将htpasswd命令创建的账号和gerrit数据库中同名用户关联起来，而是自动新创建一个gerrit用户与之关联（即一个新的gerrit用户ID），而且会要求你手动输入gerrit用户名。

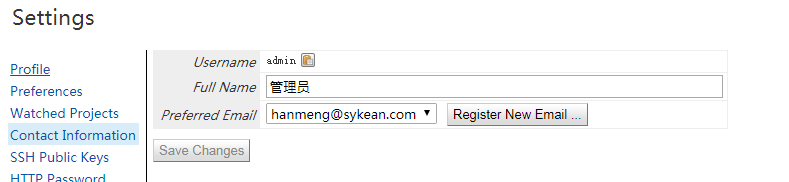
gerrit网页端配置：

1：登录，浏览器输入192.168.1.252:8080回车验证身份



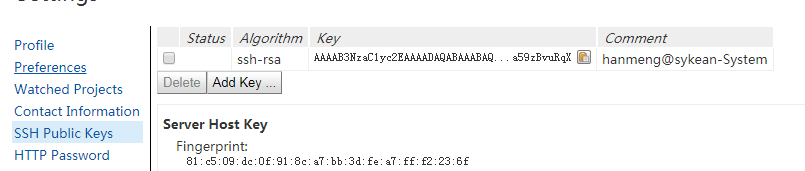
2：账户信息配置，点击右上角Settings





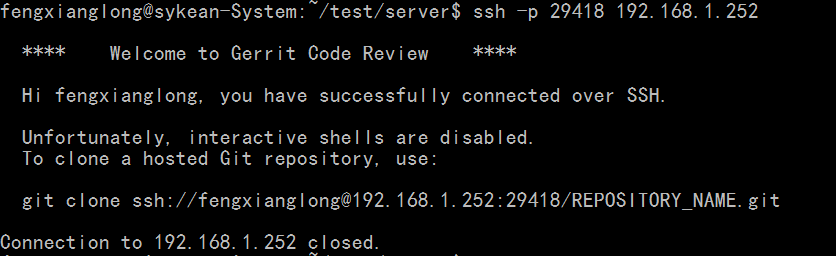
Contact Information配置全名和邮箱，全名会在界面右上角显示，邮箱用户后续操作中的提示。

添加个人公钥：



本地段操作：

添加完公钥之后，可以通过:ssh -p 29418 192.168.1.252来测试ssh是否可以连接，如果连接成功，那么就可以做接下来的操作了。成功提示如下：



创建项目：

ssh -p 29418 user@localhost gerrit create-project --empty-commit --name demo-project

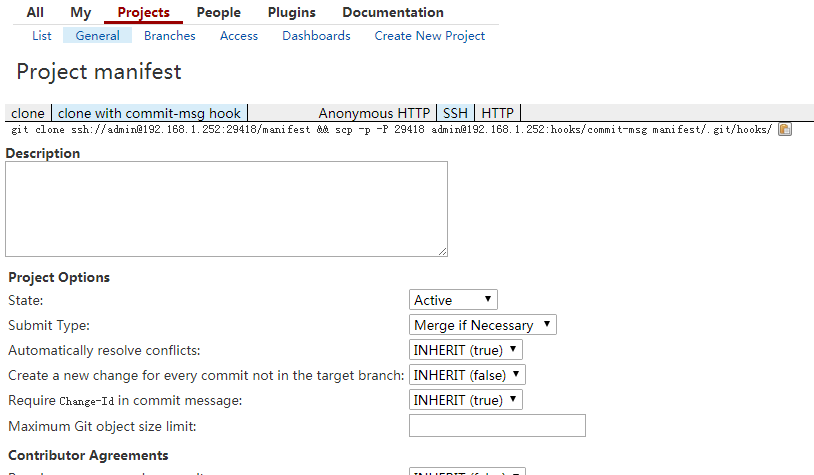
例如创建清单库和测试库：

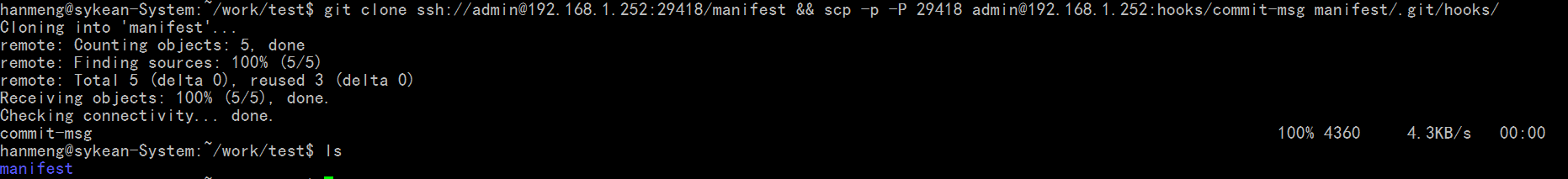
ssh -p 29418 [admin@192.168.1.252](mailto:admin@192.168.1.252) gerrit create-project --empty-commit --name manifest

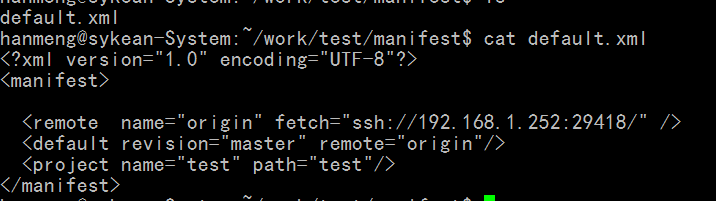
ssh -p 29418 [admin@192.168.1.252](mailto:admin@192.168.1.252) gerrit create-project --empty-commit --name test

这个时候可以在远程服务器的gerrit服务器的git目录下看到manifest.git和test.git两个仓库，在gerrit网页端的projects->List下也可以看到这两个仓库。

repo默认是通过清单库的default.xml来管理其他项目的，这里单独clone下manifest仓库，添加default.xml文件







然后推送default.xml到远程manifest仓库

git add default.xml

git commit -m “add default.xml” default.xml

git push orgin HEAD:refs/for/master //这句话是推送到gerrit，可以通过管理员审核，然后合并

然后本地创建目录，通过以下操作，同时管理多个git仓库

repo init -u ssh://fengxianglong@192.168.1.252:29418/manifest.git

//这里会在当前目录下生成.repo目录，里面会包含完整的repo和清单库

repo sync //这里会根据清单库的xml文件来同步下载xml里面的project标签定义的所以项目