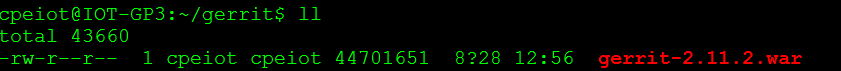
Gerrit 安装

安装之前准备：

JDK1.6 +

从<http://code.google.com/p/gerrit/>下载到Gerrit。



在工作目录中执行 java –jar gerrit-2.11.2.war init –d review-site

cpeiot@IOT-GP3:~/gerrit$ java -jar gerrit-2.11.2.war init -d review-siet1

Using secure store: com.google.gerrit.server.securestore.DefaultSecureStore

\*\*\* Gerrit Code Review 2.11.2

\*\*\*

Create '/home/cpeiot/gerrit/review-siet1' [Y/n]?

\*\*\* Git Repositories

\*\*\*

Location of Git repositories [git]:

\*\*\* SQL Database

\*\*\*

Database server type [h2]:

\*\*\* Index

\*\*\*

Type [LUCENE/?]:

\*\*\* User Authentication

\*\*\*

Authentication method [OPENID/?]: development\_become\_any\_account

\*\*\* Review Labels

\*\*\*

Install Verified label [y/N]?

\*\*\* Email Delivery

\*\*\*

SMTP server hostname [localhost]:

SMTP server port [(default)]:

SMTP encryption [NONE/?]:

SMTP username :

\*\*\* Container Process

\*\*\*

Run as [cpeiot]:

Java runtime [/usr/lib/jvm/java-7-openjdk-amd64/jre]:

Copy gerrit-2.11.2.war to /home/cpeiot/gerrit/review-siet1/bin/gerrit.war [Y/n]?

Copying gerrit-2.11.2.war to /home/cpeiot/gerrit/review-siet1/bin/gerrit.war

\*\*\* SSH Daemon

\*\*\*

Listen on address [\*]:

Listen on port [29418]:

Gerrit Code Review is not shipped with Bouncy Castle Crypto SSL v151

If available, Gerrit can take advantage of features

in the library, but will also function without it.

Download and install it now [Y/n]?

Downloading http://www.bouncycastle.org/download/bcpkix-jdk15on-151.jar ... !! FAIL !!

error: www.bouncycastle.org

Please download:

http://www.bouncycastle.org/download/bcpkix-jdk15on-151.jar

and save as:

/home/cpeiot/gerrit/review-siet1/lib/bcpkix-jdk15on-151.jar

Press enter to continue

Continue without this library [Y/n]?

Generating SSH host key ... rsa(simple)... done

\*\*\* HTTP Daemon

\*\*\*

Behind reverse proxy [y/N]?

Use SSL (https://) [y/N]?

Listen on address [\*]:

Listen on port [8080]: 8088

Canonical URL [http://IOT-GP3:8088/]:

\*\*\* Plugins

\*\*\*

Installing plugins.

Install plugin reviewnotes version v2.11.2 [y/N]?

Install plugin replication version v2.11.2 [y/N]?

Install plugin download-commands version v2.11.2 [y/N]?

Install plugin singleusergroup version v2.11.2 [y/N]?

Install plugin commit-message-length-validator version v2.11.2 [y/N]?

Initializing plugins.

No plugins found with init steps.

\*\*\* Gerrit Administrator

\*\*\*

Create administrator user [Y/n]? y

username [admin]:

name [Administrator]:

email [admin@example.com]:

HTTP password [secret]:

Initialized /home/cpeiot/gerrit/review-siet1

Executing /home/cpeiot/gerrit/review-siet1/bin/gerrit.sh start

Starting Gerrit Code Review: OK

Waiting for server on IOT-GP3:8088 ... OK

Opening http://IOT-GP3:8088/#/admin/projects/ ...OK

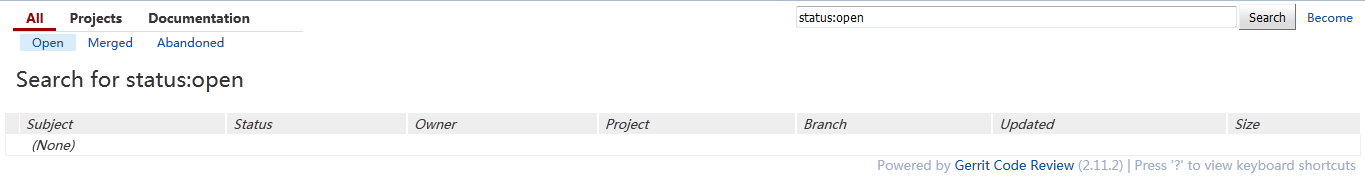
安装完成之后，gerrit是自动启动的，当需要时可以实现手动开启/关闭

在gerrit目录下bin

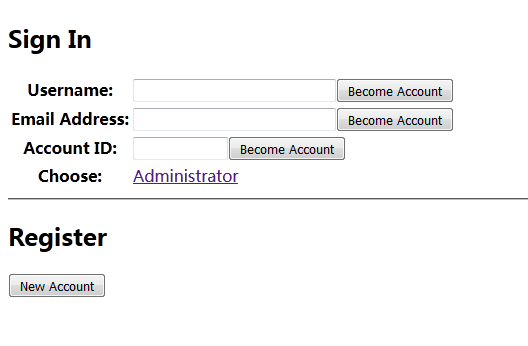
./gerrit.sh start/stop/restart 这样实现开启/停止/重启。

到此为止已经安装成功。

打开浏览器输入 <http://localhost:8088> 根据个人设置的端口来配置。

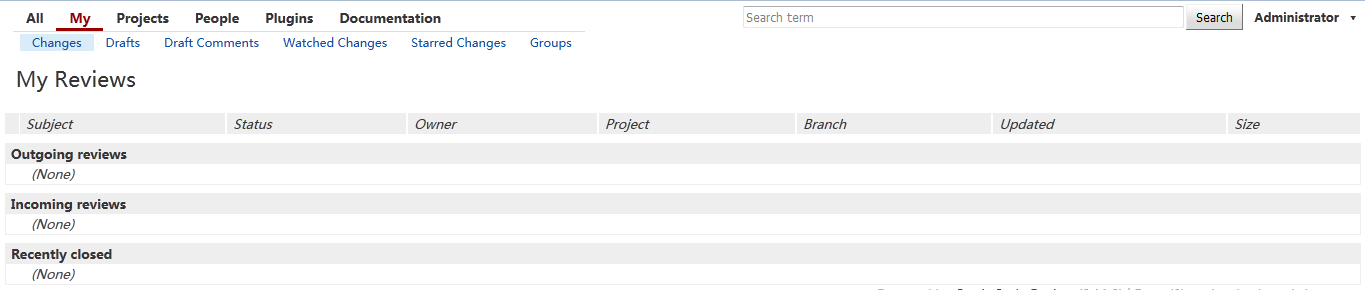


因为在安装的时候，已经注册了管理员用户，在这里可以直接使用管理员账号；否则在web端，注册的第一个账户为管理员账号。

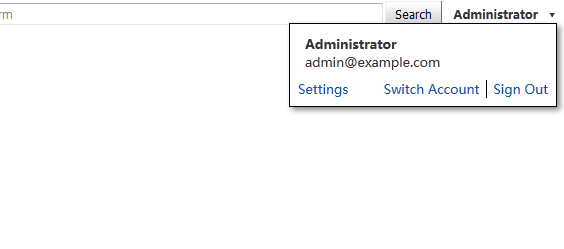


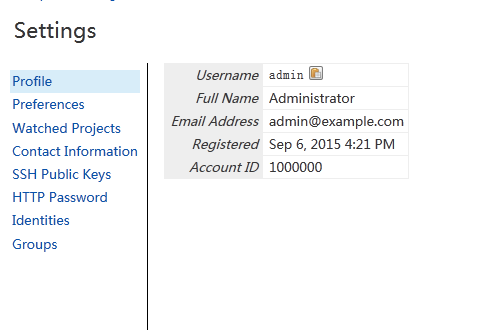
选择 【Administrator】

这样就进入了gerrit主界面。



点击用户名—> 【Setting】



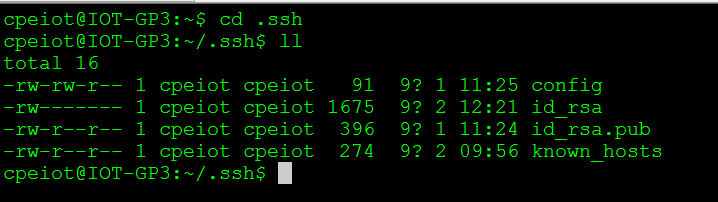


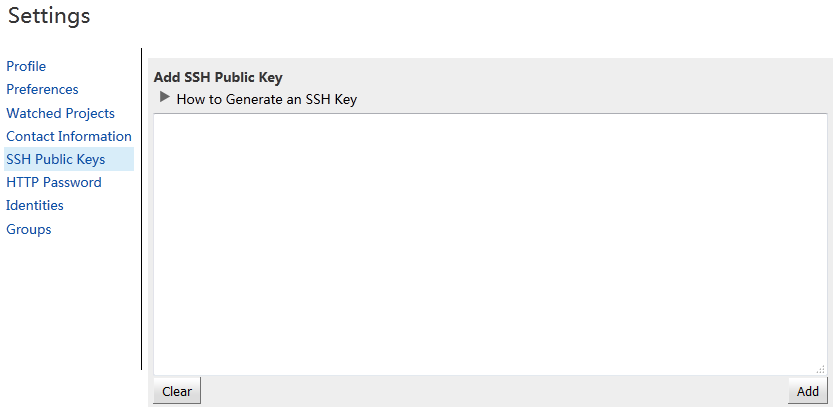
在这里的设置主要是【SSH Public Keys】 ， 将注册用户的ssh public key (~/.ssh/id\_rsa.pub文件) 加入此处。

获得ssh key 的方法：

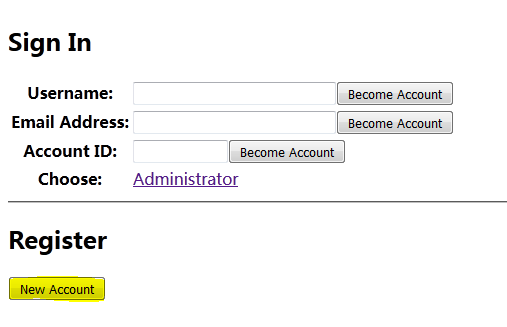
$ ssh-keygen –t rsa –b 2048 –c “username” –f “filename”或者默认设置

$ ssh-keygen –t rsa

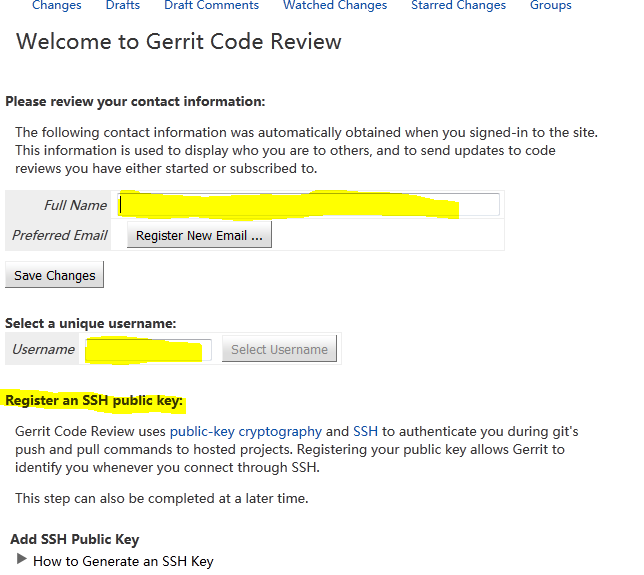




添加用户操作：



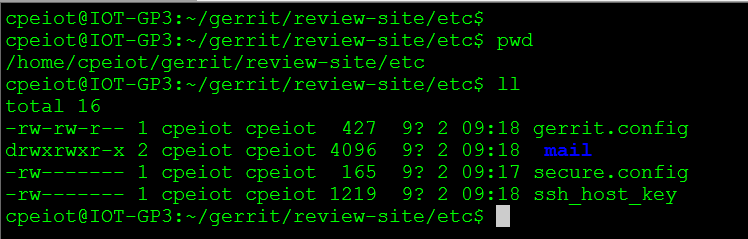
点击【New Account】



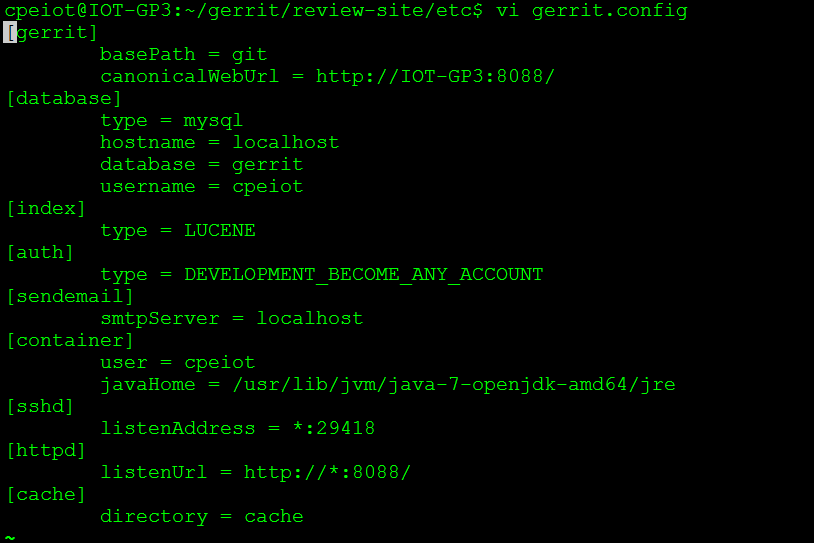
完成之后，输入用户的SSH public key.

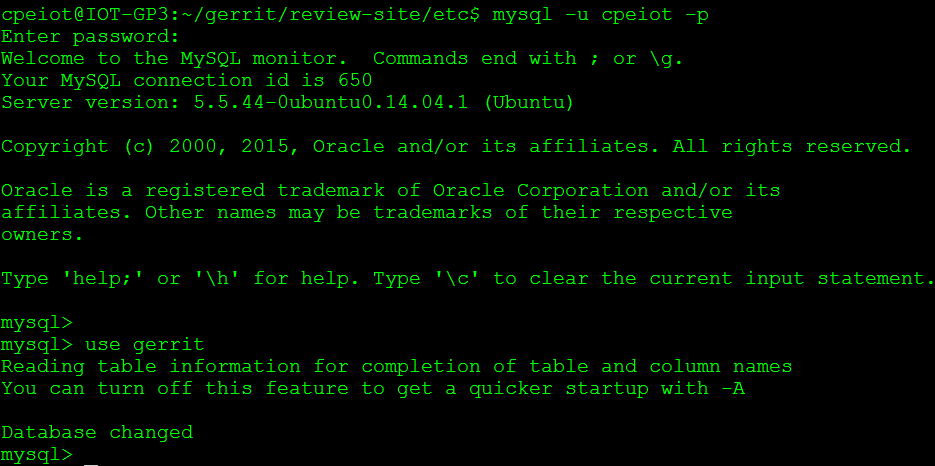
作为leader，对用户、项目的管理建议使用其他数据库，而非默认的H2。

配置MySQL，数据库的方法：

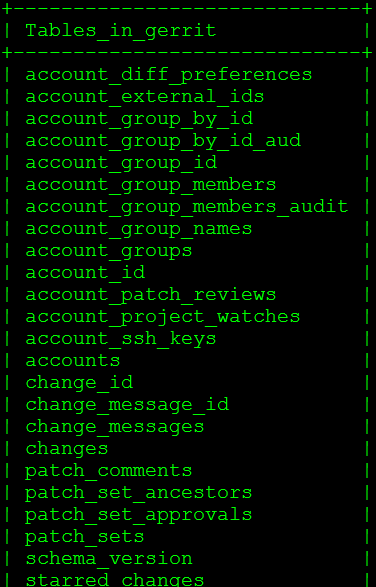


在gerrit-config中完成配置，配置信息如下：



进入MySQL，

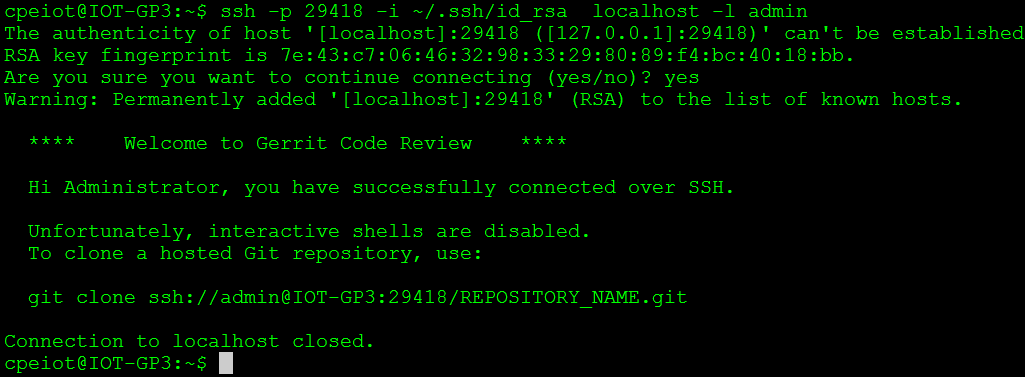
查看数据库中的表，



这样一来就可以完成对用户、项目等的管理。

到此就完成了基本配置。

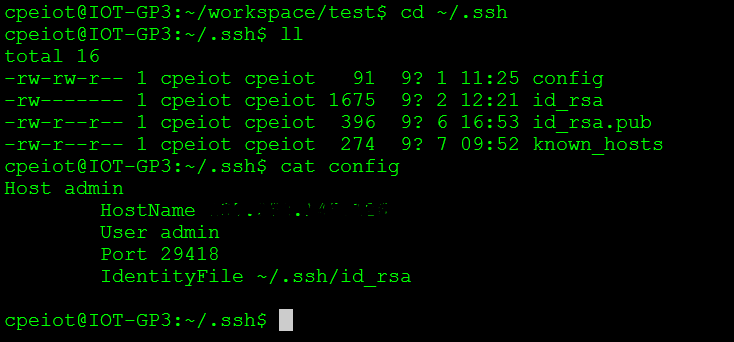
验证配置：



出现“\*\*\*\* Welcome to Gerrit Code Review \*\*\*\*” ，那么配置成功。

为了以后不用输入如此冗长的字母，配置一个host。

在~/.ssh中新建文件config



HostName : gerrit所在主机IP

User : 用户在gerrit上注册的用户名

Port : gerrit 的端口

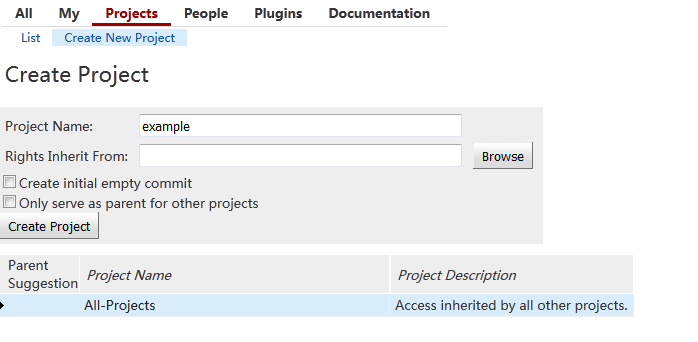
IdentityFile : 需要和gerrit上public key 进行验证的private key.

这样就可以使用ssh admin进行相关操作。

新建项目：

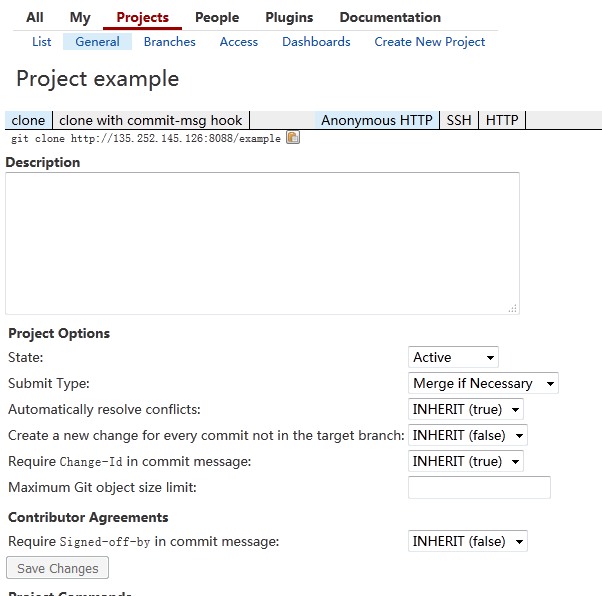


点击【create new projects】



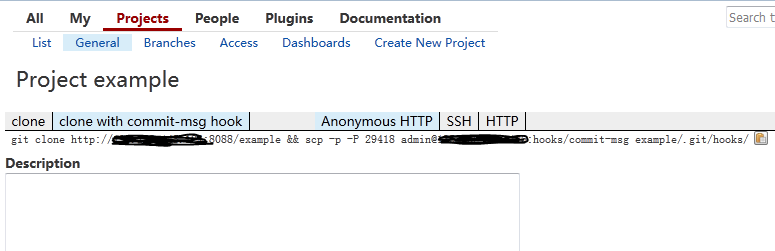
在这里只是一个普通的项目example 点击【Create Project】

这样就完成了项目的创建。



现在我们需要把项目clone到本地。

选择【clone with commit-msg hook】



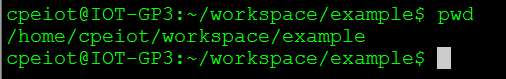
复制上面的clone地址，在本地的工作目录进行clone。

如果在这个过程中出现



会在push的时候，出现错误，不过现在可以忽略，下面有解决办法。

进入工作目录

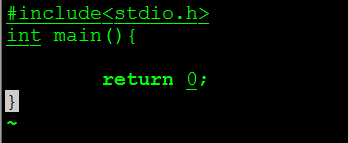


就可以coding。

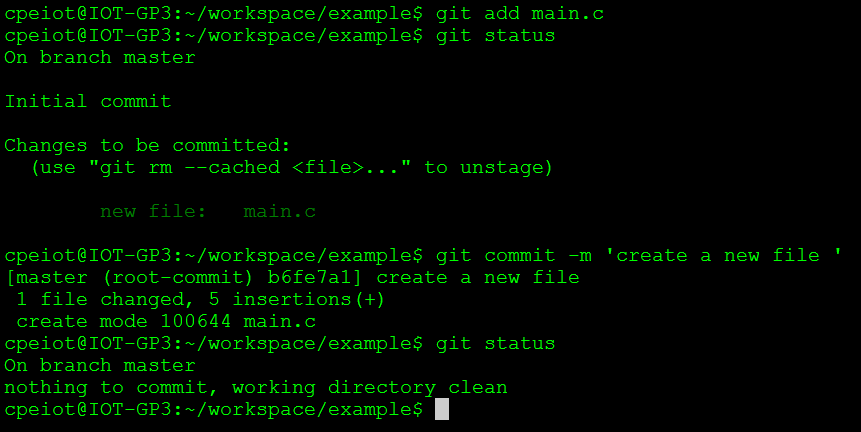
在这里做一个小小的测试



添加一个main.c文件

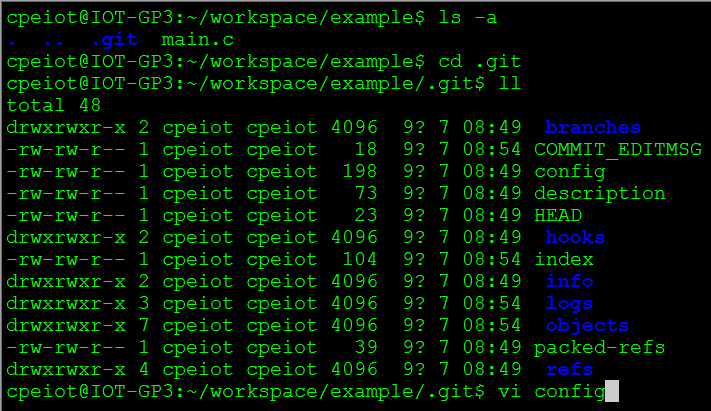


使用git 相关命令进行提交。

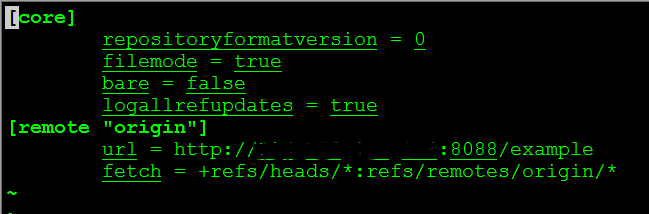


这样已经在本地仓库实现了提交，现在要做的是将其提交到远程仓库，此处与普通提交有点差异。

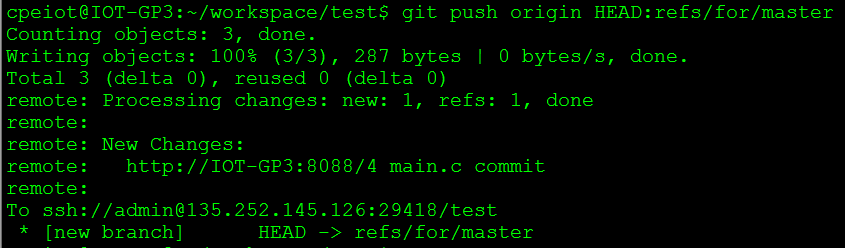
首先查看git的配置信息



在clone的时候，会自动生成一个默认的origin



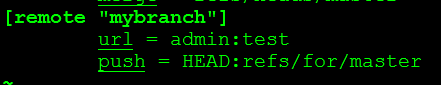
现在开始提交



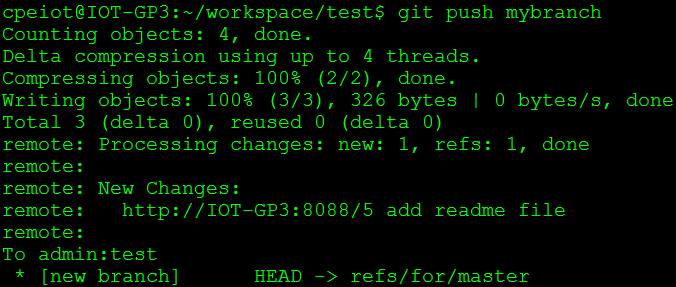
此时注意提交方式：

git push origin HEAD:refs/for/master

当然也可以指定简洁的方式，在.git/config文件中添加



此时可以使用



到此为止，已经完成了本地仓库到远程仓库的提交。

登录web ,查看上面我们提交引起的changes。