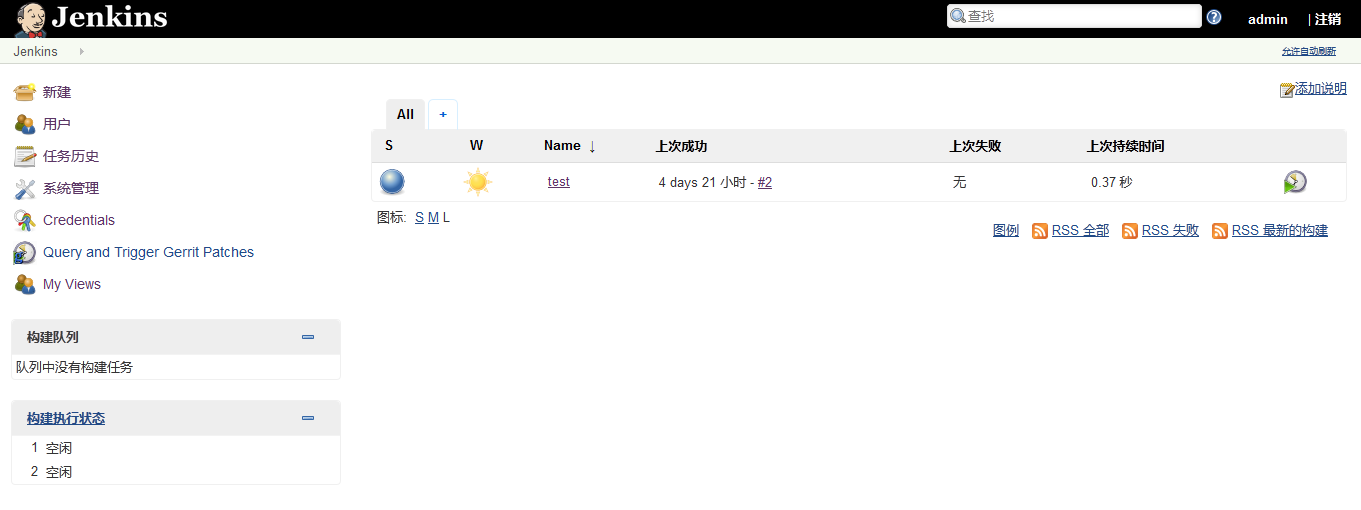


Jenins 工作目录 /var/lib/Jenkins/

登录web端:http://localhost:8080

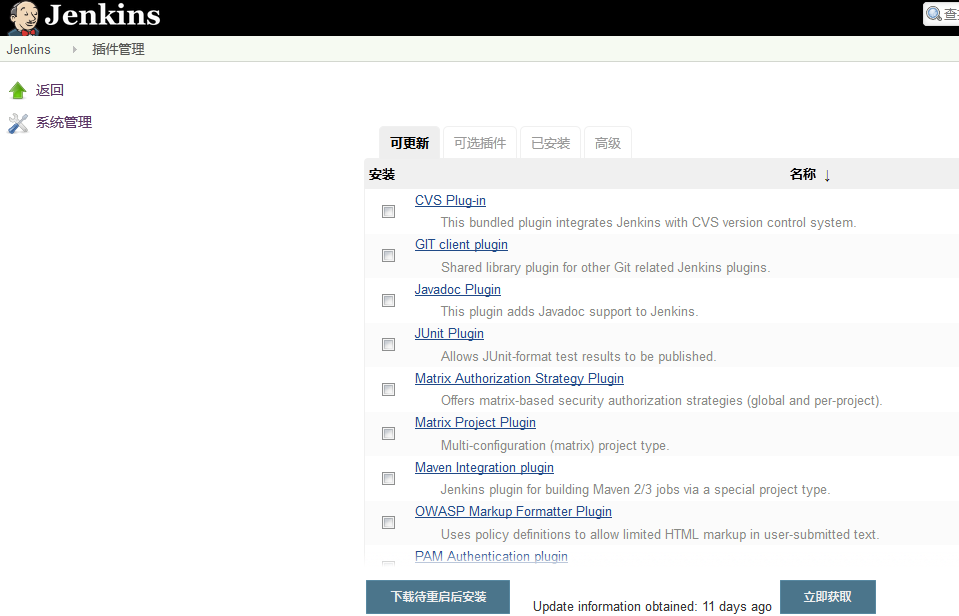


首先要进行向相关的配置

安装plugins 【系统管理】🡪 【管理插件】



选择需要的插件

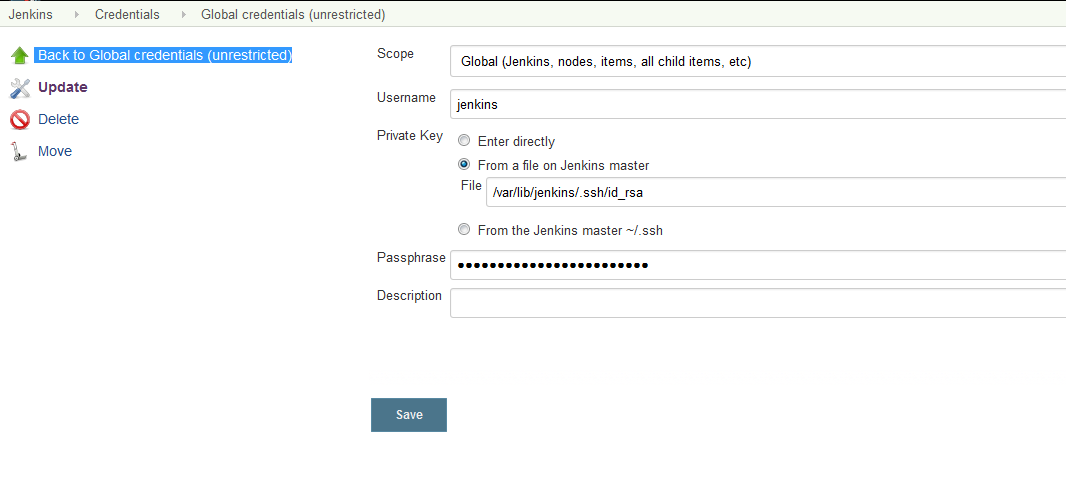


在这里由于我们使用gerrit，所以选择与之相关的plugin



主要用的是上面四个。

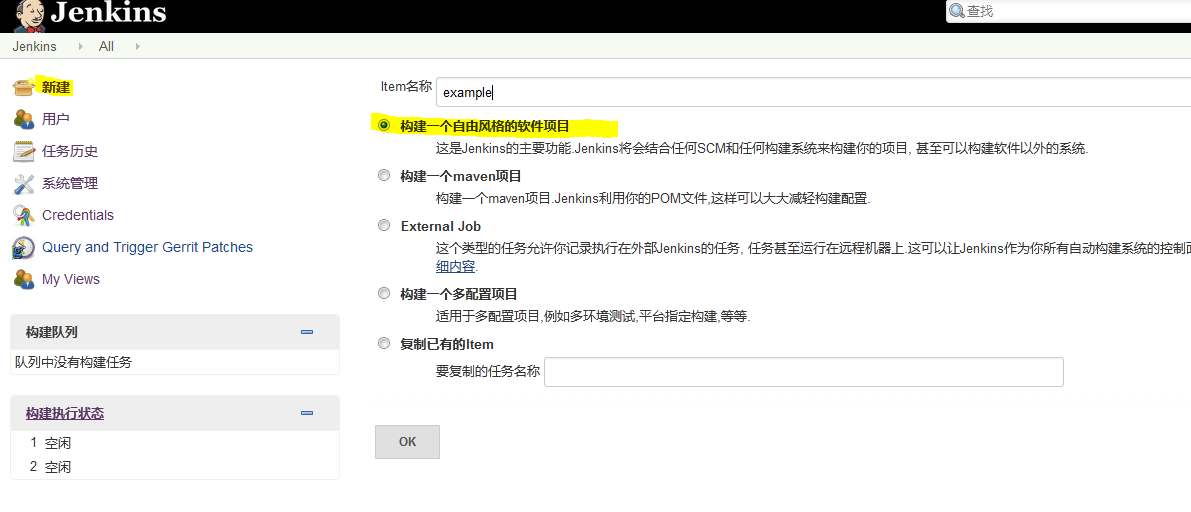
接下来就是配置Credentials



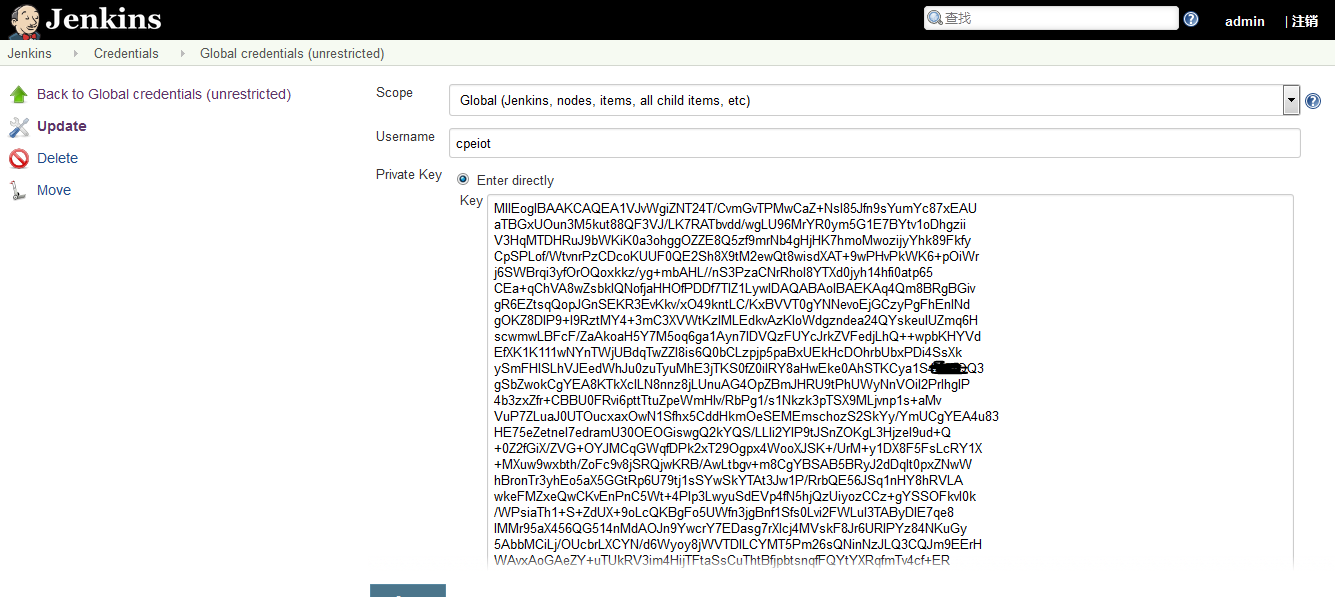
在这里添加的是在我们安装jenkins主机上的ssh rsa。生成方法不再赘述。但是生成的时候要切换到jenkins这个用户下。

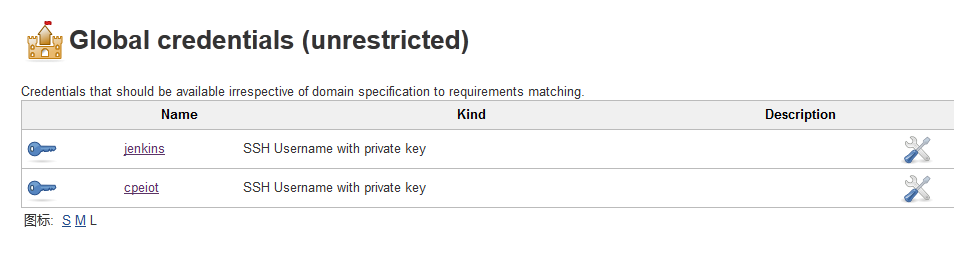
完成上面配置就开始实现和gerrit的结合。

在jenkins web上创建新的job，为了方便演示，我们创建一个free-style job

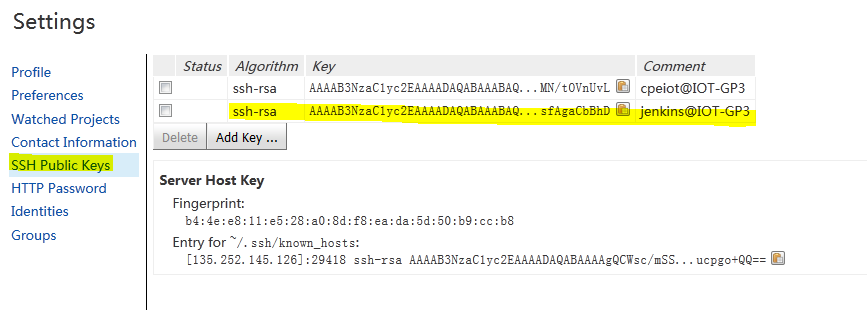


为了实现jenkins和gerrit关联，需要添加gerrit用户的private key 和jenkins 的private key

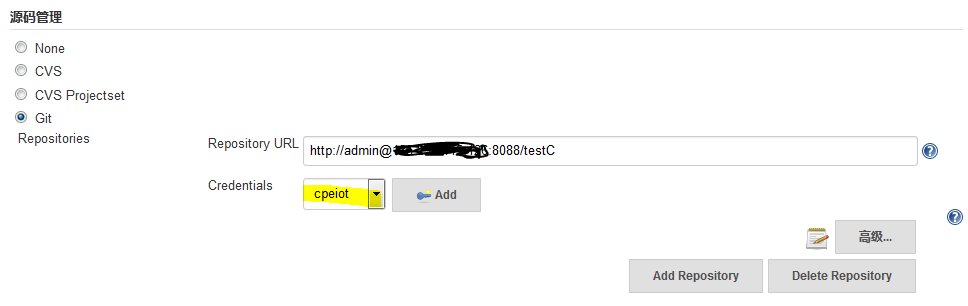




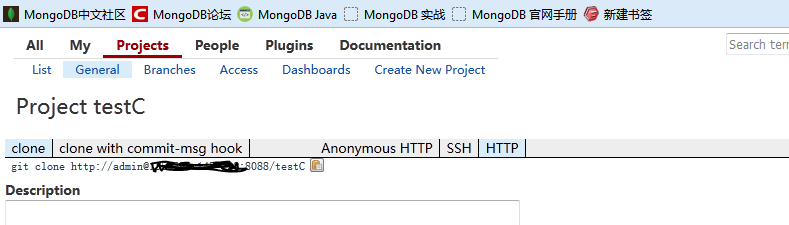
此外还要在jenkins关联的gerrit用户的key中添加jenkins的public key。



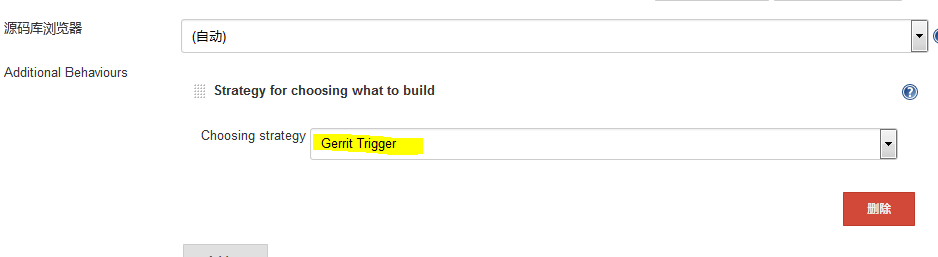
接下来就是对该job的配置



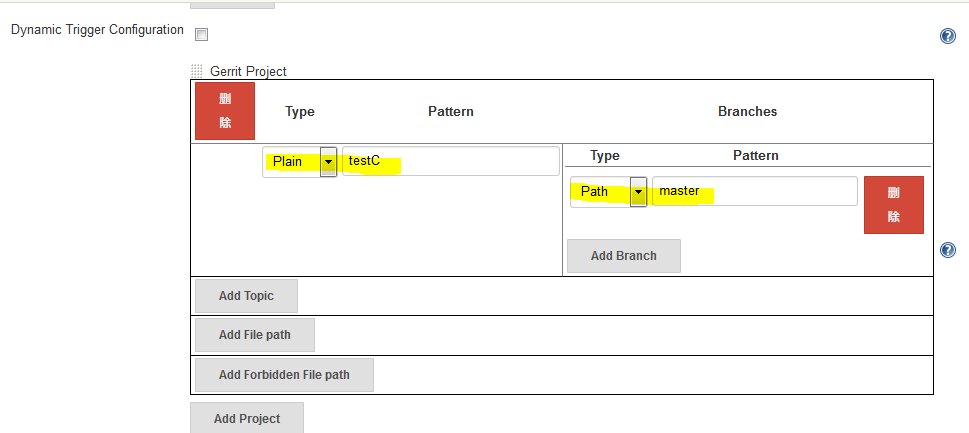
上面的URL可以在gerrit相关的项目找到。

直接拷贝即可。

我们使用gerrit trigger来触发。



进行触发器的配置



选择项目，选择分支。

其他可以选择默认设置。

接下来开始配置gerrit server



在管理Jenkins里面找到Gerrit Trigger

新建Server，进行相关配置如下：



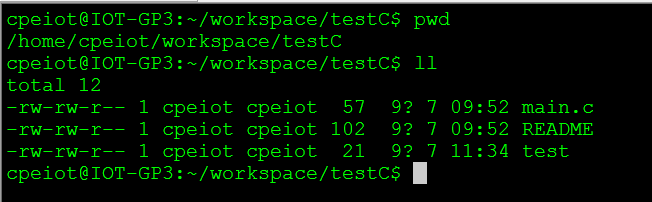
上面的SSH Keyfile ，在创建的时候会有默认值，直接默认即可。这个是jenkins用户的private key。

配置完成，点击【Test Connetcion】 出现success即可。

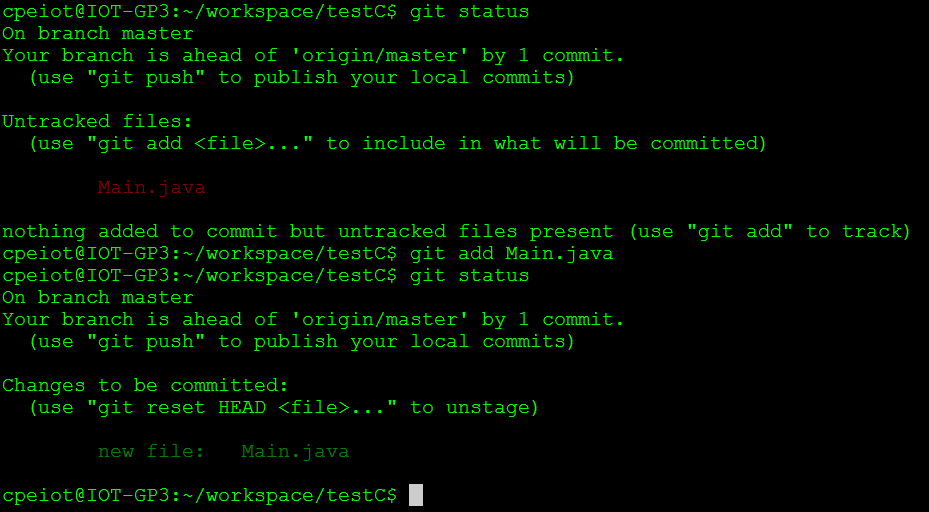
这样就完成了所有的操作。

接下来测试：

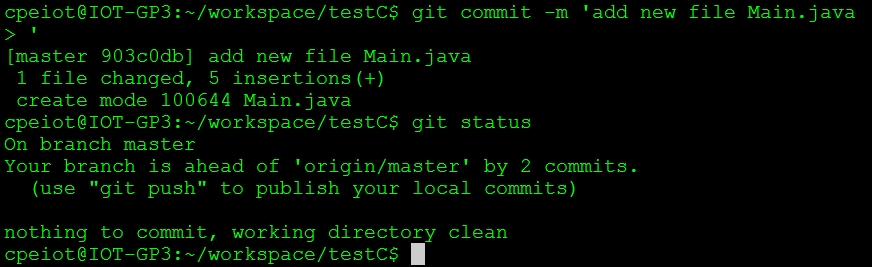
前提：已完成了gerrit项目的clone。



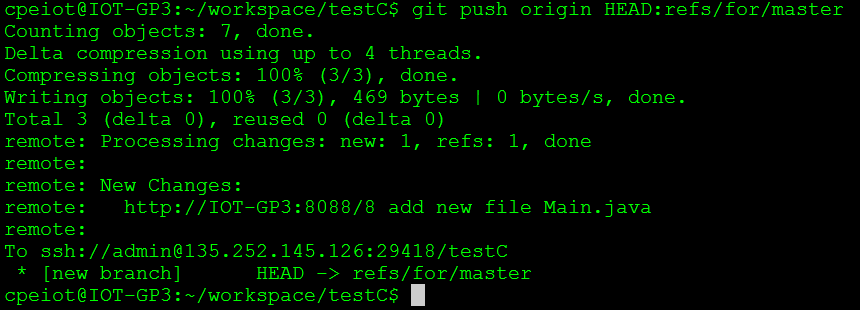
添加文件 Main.java



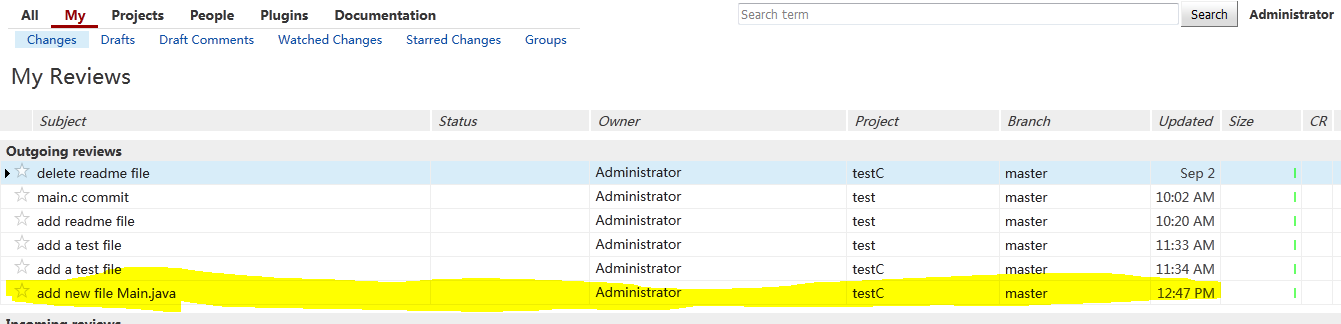
提交文件



这样整个项目就产生了变更。



查看gerrit:

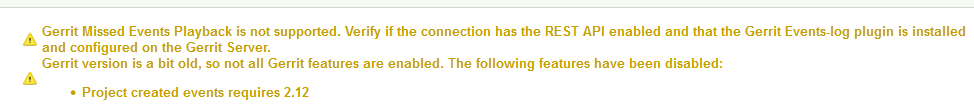


查看jenkins ，是否成功build



这样就全部实现了整个过程。

补充：



为了方便实现Project created events，添加插件 gerrit-plugin-api-2.12-SNAPSHOT.jar