Gerrit 安装与使用

> [Gerrit 官方文档](#)

Gerrit 安装

在Linux服务器上安装相关软件

1. 安装Jdk
2. 安装Git
3. 安装httpd

创建Gerrit目录

```
mkdir /root/gerrit
```

下载Gerrit安装包

```
cd /root/gerrit
wget https://gerrit-releases.storage.googleapis.com/gerrit-3.2.7.war
```

启动Gerrit

```
java -jar gerrit-3.2.7.war init -d review_site

# Location of Git repositories   [git]: /root/gerrit/gitwork
# Authentication method          [openid/?]: HTTP
# Canonical URL                  [http://localhost:8080/]:
http://39.105.47.81:8080/
# Install plugin codemirror-editor version v3.2.7 [y/N]? y
# Install plugin commit-message-length-validator version v3.2.7 [y/N]? y
#  Install plugin delete-project version v3.2.7 [y/N]? y
# Install plugin download-commands version v3.2.7 [y/N]? y
# Install plugin gitiles version v3.2.7 [y/N]? y
# Install plugin hooks version v3.2.7 [y/N]? y
# Install plugin plugin-manager version v3.2.7 [y/N]? y
# Install plugin replication version v3.2.7 [y/N]? y
# Install plugin reviewnotes version v3.2.7 [y/N]? y
#  Install plugin singleusergroup version v3.2.7 [y/N]? y
# Install plugin webhooks version v3.2.7 [y/N]? y

# 以上选项设置相关值，其他默认Enter
```

查看Gerrit配置

```
vim /root/gerrit/review_site/etc/gerrit.config
```

```
[gerrit]
    basePath = /root/gerrit/gitwork
    canonicalWebUrl = http://39.105.47.81:8080/
    serverId = 6718b3aa-2b8e-4a4a-bf5b-ac9738b42353
[container]
```

```
        javaOptions = "-
Dflogger.backend_factory=com.google.common.flogger.backend.log4j.Log4jBackendFac
tory#getInstance"
        javaOptions = "-
Dflogger.logging_context=com.google.gerrit.server.logging.LoggingContext#getInst
ance"
        user = root
        javaHome = /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.272.b10-3.el8_3.x86_64/jre
[index]
        type = lucene
[auth]
        type = HTTP
[receive]
        enableSignedPush = false
[sendemail]
        smtpServer = localhost
[sshd]
        listenAddress = *:29418
[httpd]
        listenUrl = http://*:8080/
[cache]
        directory = cache
```

Gerrit 命令行操作

```
# 查看gerrit状态
/root/gerrit/review_site/bin/gerrit.sh status
# 重启
/root/gerrit/review_site/bin/gerrit.sh restart
# 启动
/root/gerrit/review_site/bin/gerrit.sh start
# 关闭
/root/gerrit/review_site/bin/gerrit.sh stop
```

添加gerrit登录用户

```
# "-c"参数为创建，仅限第一次添加用户时使用，用户名和密码均为admin
htpasswd -cb /etc/httpd/passwords admin admin
# 第二次创建时不要加"-c"参数
htpasswd -b /etc/httpd/passwords user1 123456
```

设置httpd反向代理，http认证

```
vim /etc/httpd/conf/httpd.conf
```

```
Listen 8090
<VirtualHost *:8090>
    ServerName localhost
    ProxyRequests off
    ProxyVia off
    ProxyPreserveHost on

    <Proxy *>
        Order deny,allow
        Allow from all
```

```
        </Proxy>

        <Location /login/>
                AuthType Basic
                AuthName "Gerrit Code Review"
                Require valid-user
                AuthUserFile /etc/httpd/passwords
        </Location>


        AllowEncodedSlashes On
        ProxyPass / http://39.105.47.81:8080/ # 代理地址为Gerrit配置中canonicalWebUrl的
值
</VirtualHost>
```

```
systemctl restart httpd # 重启httpd
```

访问gerrit页面

http://39.105.47.81:8090/

使用 admin登录

设置SSH密钥

```
# 本地生成密钥
ssh-keygen -t ed25519 -C "Setting SSH keys in Gerrit"

# 如果您使用的是不支持 Ed25519 算法的旧系统，请使用以下命令：
ssh-keygen -t rsa -b 4096 -C "Setting SSH keys in Gerrit"

cat ~/.ssh/id_ed25519.pub # 复制密钥
```


Gerrit 基本使用

创建项目

克隆项目到本地
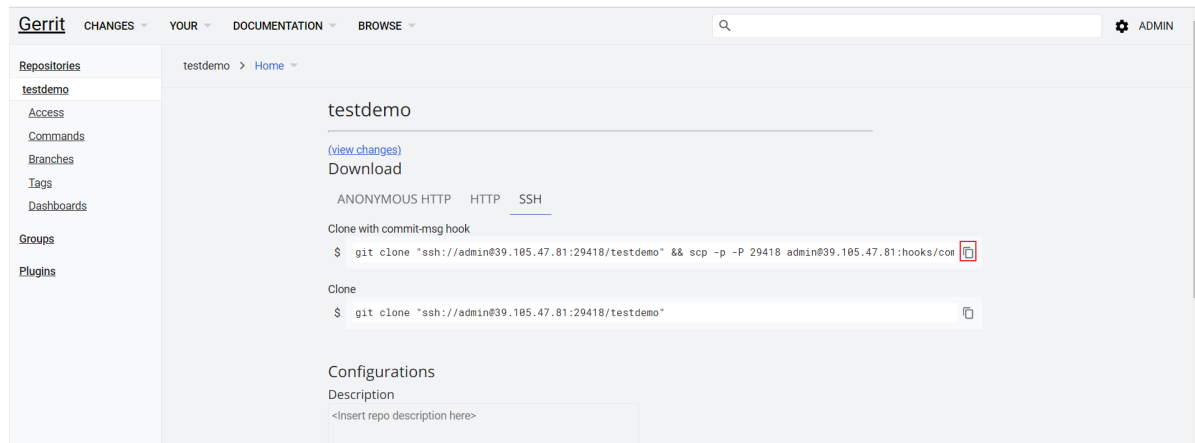

```
# admin为项目所有者  testdemo为项目名称
git clone "ssh://admin@39.105.47.81:29418/testdemo" && scp -p -P 29418
admin@39.105.47.81:hooks/commit-msg "testdemo/.git/hooks/"
```

git 提交修改到Gerrit

```
vim hello.md
git add .
git commit -m 'add hello'
git push origin HEAD:refs/for/master
```

查看修改内容是否提交到Gerrit

查看修改内容是否提交到Git仓库

```
# 重新找个目录
git clone ssh://admin@39.105.47.81:29418/testdemo
# 修改内容没有被提交到Git仓库
```

查看修改内容

对代码进行建议

> +2 直接可以进行提交

提交代码

发现代码已经合并

继续查看Git仓库

```
git pull
# 发现代码已经在Git仓库
```

Gerrit 权限管理

> code review +2 表示代码可以合并，有一个+2即可
>
> code review +1 表示代码还行，但我不能让你合并
>
> code review 0 表示没有任何意见，希望其他人处理
>
> code review -1 表示代码还有问题，但是不影响其他人给你+2进行合并
>
> code review -2 表示代码不能合并，其他人给你+2也不能合并
>
> code review +1 +1 并不等于 +2

创建用户

```
# 创建dev，review两个用户
htpasswd -b /etc/httpd/passwords dev dev
htpasswd -b /etc/httpd/passwords review review
```
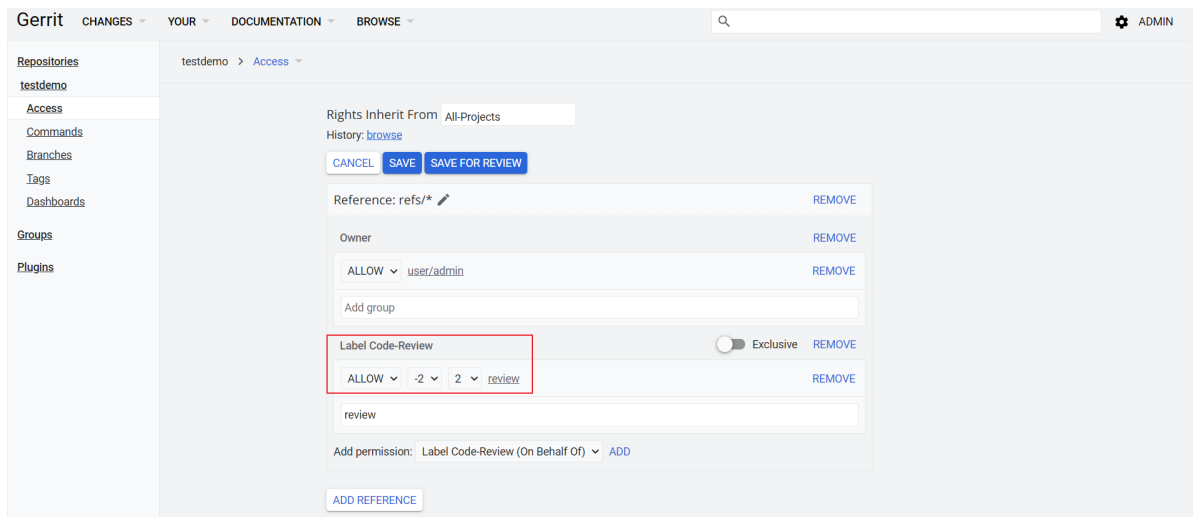
将用户注册到Gerrit上，需要用户登录一次Gerrit

> 退出登录，清除cookie即可

创建用户组

添加用户到用户组

设置项目的reviewer

代码review

1. 项目owner可以直接code review +2 然后submit 代码合并

2. 项目owner可以添加reviewers code review 0

   reviewer code review +2

   owner submit

3. 提交者 git push origin HEAD:refs/for/master%r=review 指定review

   owner submit

设置所有项目不能git push origin master
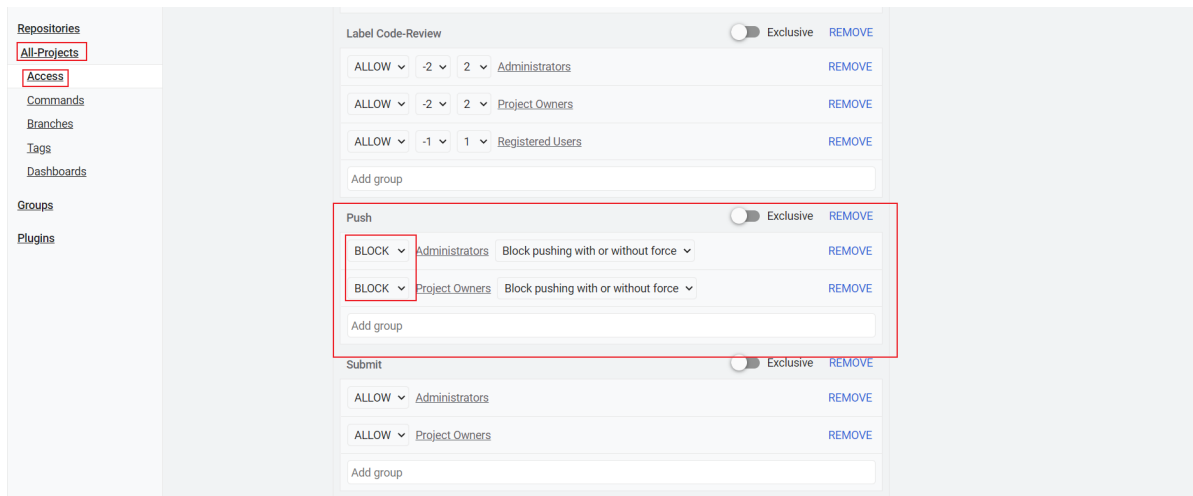
设置用户组有合并代码权限

遇到相关问题解决方案

在git push时未成功推送

```
[root@iZ2zeewce1gkyyw1ttadd5Z test2]# git push origin HEAD:refs/for/master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 253 bytes | 253.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Processing changes: refs: 1, done
remote: ERROR: commit a9ab45a: missing Change-Id in message footer
remote:
remote: Hint: to automatically insert a Change-Id, install the hook:
remote:   gitdir=$(git rev-parse --git-dir); scp -p -P 29418 admin@39.105.47.81:hooks/commit-msg ${gitdir}/hooks/
remote: and then amend the commit:
remote:   git commit --amend --no-edit
remote: Finally, push your changes again
remote:
To ssh://39.105.47.81:29418/test2
 ! [remote rejected] HEAD -> refs/for/master (commit a9ab45a: missing Change-Id in message footer)
error: failed to push some refs to 'ssh://39.105.47.81:29418/test2'
```
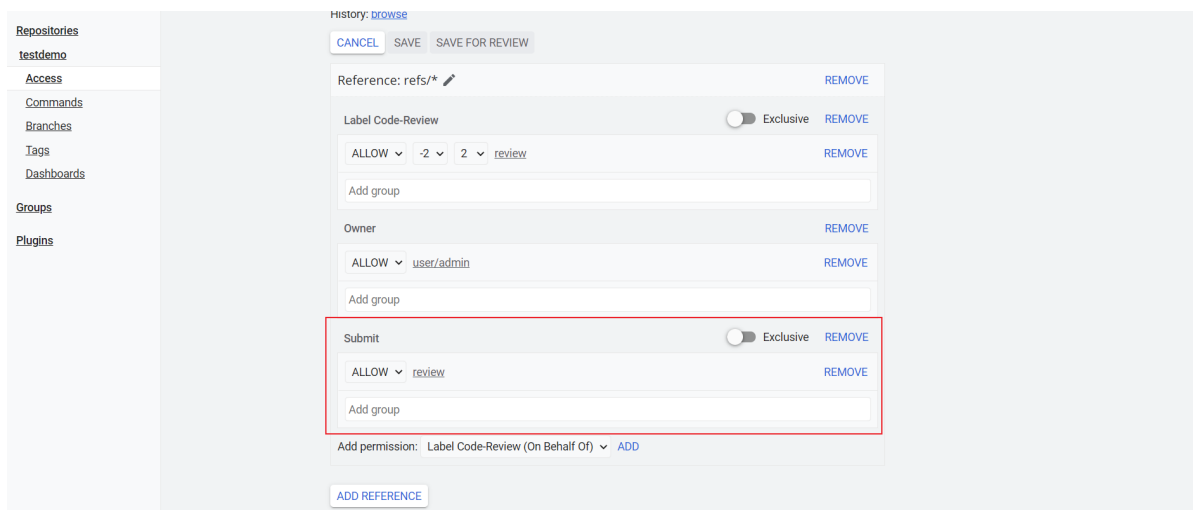
```
# 执行提示内容
gitdir=$(git rev-parse --git-dir); scp -p -P 29418
admin@39.105.47.81:hooks/commit-msg ${gitdir}/hooks/
git commit --amend --no-edit

# 重新提交
git push origin HEAD:refs/for/master
```

在git commit时报错

```
[root@VM-0-4-centos test2]# git commit -m 'helll 222'
git: 'interpret-trailers' 不是一个 git 命令。参见 'git --help'。
cannot insert change-id line in .git/COMMIT_EDITMSG
```

安装git 2.0之后版本

```
# 卸载原来的git
yum remove git
# Git第三方仓库安装
url https://setup.ius.io | sh
# 查看git版本
yum search git
# 安装git224
yum install git224
```

小技巧

git 别名设置

```
vim ~/.gitconfig
```

添加内容

```
[alias]
        cr = "!git push origin HEAD:refs/for/\"$1\" #"
```

提交到到gerrit时，可以用git cr master 代替git push origin HEAD:refs/for/master

提交代码时设置reviewer

```
# reviewer1为审查者
git cr master%r=reviewer1

# 以上命令等价于
git push origin HEAD:refs/for/master%r=reviewer1
```

提交代码时设置reviewer

```
# reviewer1为审查者
git cr master%r=reviewer1

# 以上命令等价于
git push origin HEAD:refs/for/master%r=reviewer1
```