

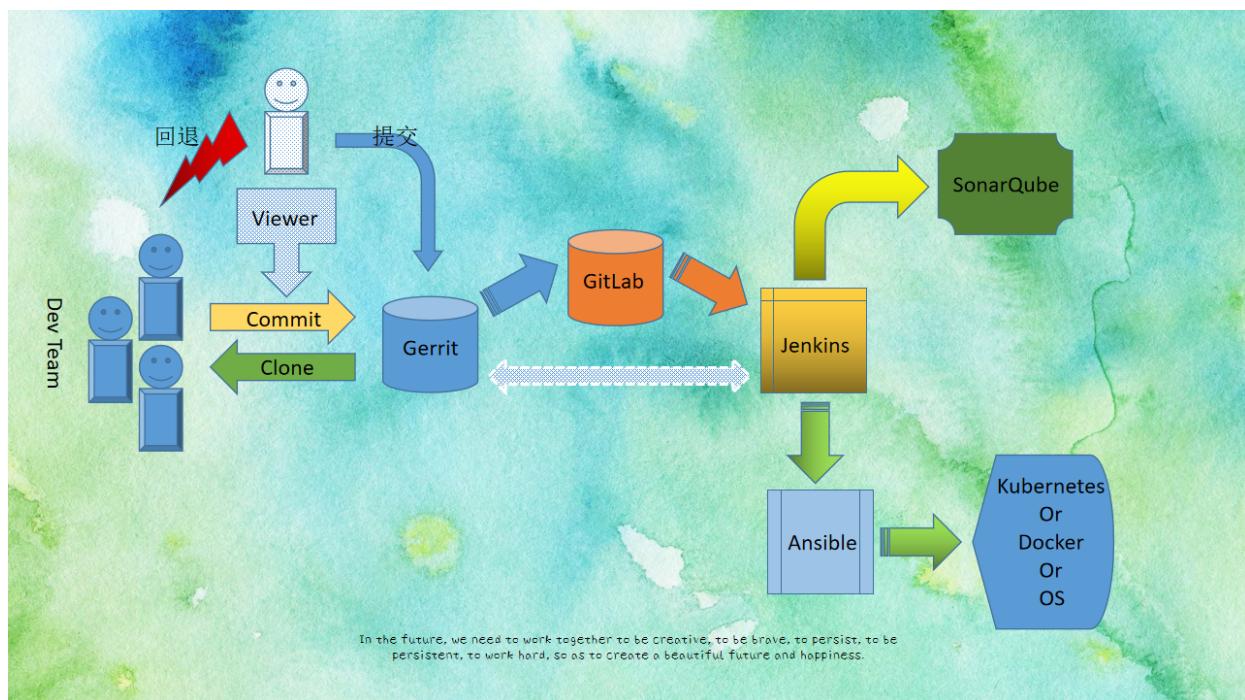
演示稿件

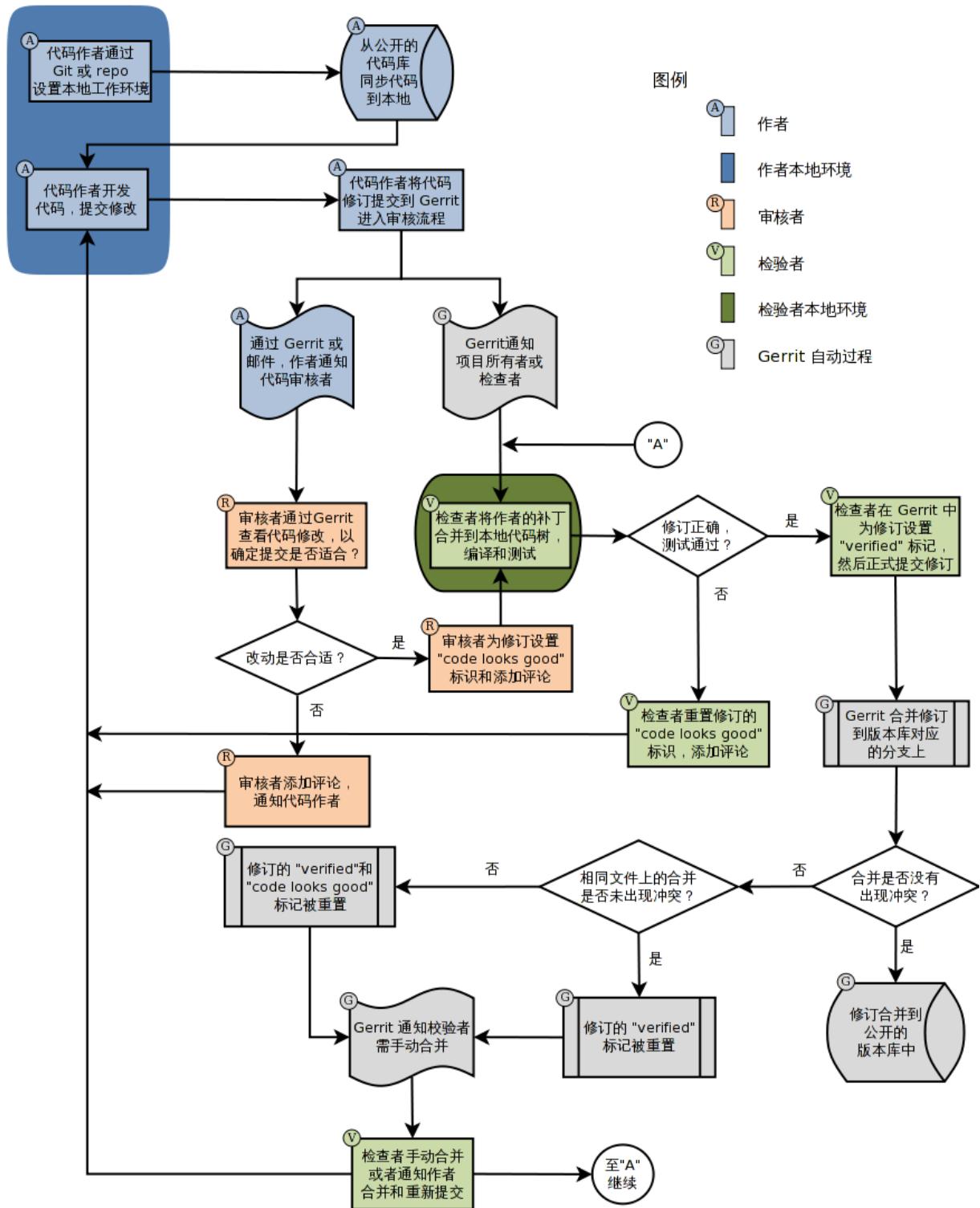


Gerrit使用手册.pptx
1.6MB



持续集成与部署.pptx
4.3MB





摘自：<http://source.android.com/source/life-of-a-patch.html>

服务器分配

192.168.21.144 Jenkins SonarQube Ansible

192.168.21.143 Gerrit

192.168.21.142 GitLab

个人经验，我认为Gerrit和Gitlab同一台服务器有坑，所以拆分开

<https://www.cnblogs.com/shengulong/p/10887510.html>

1 sysctl -p

```
2 net.core.netdev_max_backlog = 32768
3 net.core.somaxconn = 32768
4 net.core.wmem_default = 8388608
5 net.core.rmem_default = 8388608
6 net.core.rmem_max = 16777216
7 net.core.wmem_max = 16777216
8 net.ipv4.ip_local_port_range = 1024 65000
9 net.ipv4.route.gc_timeout = 100
10 net.ipv4.tcp_fin_timeout = 30
11 net.ipv4.tcp_keepalive_time = 1200
12 net.ipv4.tcp_timestamps = 0
13 net.ipv4.tcp_synack_retries = 2
14 #net.ipv4.tcp_syn_retries = 2
15 #net.ipv4.tcp_tw_reuse = 1
16 net.ipv4.tcp_mem = 94500000 915000000 927000000
17 net.ipv4.tcp_max_orphans = 3276800
18 net.ipv4.tcp_max_syn_backlog = 65536
19 fs.aio-max-nr = 4194304
20 vm.max_map_count = 655360
```

安装需要的rpm包

```
1 yum install -y tree net-tools vim lrzs unxz ntp wget telnet rsync curl
git policycoreutils openssh-server openssh-clients
2 rpm -ivh https://mirrors.ustc.edu.cn/epel/7/x86_64/Packages/e/epel-releas
e-7-11.noarch.rpm
```

关闭服务

```
1 systemctl stop firewalld
2 systemctl disable firewalld
3 systemctl stop NetworkManager
4 systemctl disable NetworkManager
5 sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
6
7 vim /etc/ssh/sshd_config
8 --- #PermitEmptyPasswords no
9 +++) PasswordAuthentication yes
10 --- #PermitRootLogin yes
11 +++) PermitRootLogin yes
12 --- #UseDNS yes
13 +++) UseDNS no
```

apache-ant-1.10.5
apache-maven-3.6.0
gerrit-2.16.8.war
sonarqube-7.7
sonar-scanner-3.3.0.1492
jdk1.8.0_212
events-log.jar
gitlab-ce-11.10.4-ce.0.el7.x86_64.rpm
node-v12.3.1-linux-x64

```
1 wget http://mirrors.tuna.tsinghua.edu.cn/apache/maven/maven-3/3.6.0/binaries/apache-maven-3.6.0-bin.tar.gz
2 wget http://mirrors.tuna.tsinghua.edu.cn/apache/ant/binaries/apache-ant-1.10.5-bin.zip
3 wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-7.7.zip
4 wget https://gerrit-releases.storage.googleapis.com/gerrit-2.16.8.war
5 wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-3.3.0.1492-linux.zip
6 wget https://gerrit-ci.gerritforge.com/job/plugin-events-log-stable-2.15/lastSuccessfulBuild/artifact/buck-out/gen/plugins/events-log/events-log.jar
7 wget https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el7/gitlab-ce-11.10.4-ce.0.el7.x86_64.rpm
8 wget http://services.gradle.org/distributions/gradle-5.4.1-all.zip
9 https://nodejs.org/dist/v12.3.1/node-v12.3.1-linux-x64.tar.xz
```

jenkins服务器操作

编辑profile的环境变量

```
1 vi /etc/profile
2 export JAVA_HOME=/usr/local/jdk1.8.0_212
3 export JRE_HOME=/usr/local/jdk1.8.0_212/jre
4 export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
5 export PATH=$PATH:$JAVA_HOME/bin
6
7 export SONAR_HOME=/usr/local/sonarqube-7.7
```

```
8 export PATH=$PATH:$SONAR_HOME/bin/linux-x86-64
9
10 export GRADLE_HOME=/usr/local/gradle-5.4.1
11 export PATH=$PATH:$GRADLE_HOME/bin
12
13 export SONAR_RUNNER_HOME=/usr/local/sonar-scanner-3.3.0.1492
14 export PATH=$PATH:$SONAR_RUNNER_HOME/bin
15 export SONAR_RUNNER_OPTS="-Xmx512m -XX:MaxPermSize=128m"
16
17 export MAVEN_HOME=/usr/local/apache-maven-3.6.0
18 export PATH=$PATH:$MAVEN_HOME/bin
19
20 export NODE_HOME=/usr/local/node-v12.3.1-linux-x64
21 export PATH=$PATH:$NODE_HOME/bin
22
23 export SONAR_SCANNER_OPTS="-Xmx512m"
```

使用Jenkins官方rpm源安装

```
1 wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
2 rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
3 yum install jenkins
```

修改配置文件

```
1 vi /etc/init.d/jenkins
2 /usr/local/jdk1.8.0_212/bin/java
```

启动Jenkins

```
1 systemctl daemon-reload
2 systemctl start jenkins
3 systemctl stop jenkins
4 /sbin/chkconfig jenkins on
```

免密登陆

```
1 su - jenkins
2 ssh-keygen -t rsa -C jenkins@bochtec.com
```

优化jenkins模块下载地址，切记本优化为应急，正常可安装插件不需要执行

```
1 sed -i 's#http://updates.jenkins-ci.org/download#http://ftp.tsukuba.wide.ad.jp/software/jenkins#g' updates/default.json
```

再次启动服务

```
1 systemctl start jenkins  
2 systemctl enable jenkins  
3 systemctl restart jenkins
```

编辑jenkins用户的环境变量

```
1 vi /var/lib/jenkins/.bash_profile  
2 export JAVA_HOME=/usr/local/jdk1.8.0_212  
3 export JRE_HOME=/usr/local/jdk1.8.0_212/jre  
4 export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH  
5 export PATH=$PATH:$JAVA_HOME/bin  
6  
7 export SONAR_HOME=/usr/local/sonarqube-7.7  
8 export PATH=$PATH:$SONAR_HOME/bin/linux-x86-64  
9  
10 export GRADLE_HOME=/usr/local/gradle-5.4.1  
11 export PATH=$PATH:$GRADLE_HOME/bin  
12  
13 export SONAR_RUNNER_HOME=/usr/local/sonar-scanner-3.3.0.1492  
14 export PATH=$PATH:$SONAR_RUNNER_HOME/bin  
15 export SONAR_RUNNER_OPTS="-Xmx512m -XX:MaxPermSize=128m"  
16  
17 export MAVEN_HOME=/usr/local/apache-maven-3.6.0  
18 export PATH=$PATH:$MAVEN_HOME/bin  
19  
20 export NODE_HOME=/usr/local/node-v12.3.1-linux-x64  
21 export PATH=$PATH:$NODE_HOME/bin  
22  
23 export SONAR_SCANNER_OPTS="-Xmx512m"
```

安装mysql5.7

```
1 wget https://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm  
2 rpm -ivh mysql57-community-release-el7-9.noarch.rpm  
3 yum install mysql-server  
4  
5 vi /etc/my.cnf  
6 [mysqld]
```

```
7 character_set_server=utf8
8 init_connect='SET NAMES utf8'
```

启动mysql

```
1 systemctl start mysqld
2 systemctl enable mysqld
```

授权sonar连接授权

```
1 grep 'temporary password' /var/log/mysqld.log
2 mysql -uroot -p
3 set password=password("XXXXXXXXXXXX");
4 CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8_general_ci;
5 GRANT ALL ON sonar.* TO 'sonar'@'127.0.0.1' IDENTIFIED BY 'XXXXXXXXXXXX';
6 FLUSH PRIVILEGES;
```

授权gerrit连接

```
1 mysql -uroot -pXXXXXXXXXXXXXX
2 CREATE DATABASE reviewdb CHARACTER SET utf8 COLLATE utf8_general_ci;
3 GRANT ALL ON reviewdb.* TO 'gerrit'@'192.168.21.143' IDENTIFIED BY 'XXXXXX
XXXXXXXX';
4 set global explicit_defaults_for_timestamp=1;
5 FLUSH PRIVILEGES;
```

配置sonarqube

```
1 chown jenkins:jenkins -R /usr/local/sonarqube-7.7
2 vi /usr/local/sonarqube-7.7/conf/sonar.properties
3
4 sonar.jdbc.username=sonar
5 sonar.jdbc.password=XXXXXXXXXXXXX
6 sonar.jdbc.url=jdbc:mysql://127.0.0.1:3306/sonar?useUnicode=true&characte
rEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance&useS
SL=false
```

启动sonarqube

```
1 vi /etc/passwd
2 jenkins:x:998:996:Jenkins Automation Server:/var/lib/jenkins:/bin/bash
3 su - jenkins
4 cd /usr/local/sonarqube-7.7/bin/linux-x86-64
5 ./sonar.sh start
```

gerrit服务器操作

安装jdk

```
1 vi /etc/profile
2 export JAVA_HOME=/usr/local/jdk1.8.0_212
3 export JRE_HOME=/usr/local/jdk1.8.0_212/jre
4 export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
5 export PATH=$PATH:$JAVA_HOME/bin
```

安装gitweb，一定要先安装这个

```
1 yum install -y gitweb
2 useradd gerrit
3 mkdir /usr/local/gerrit-2.15.13
4 chown gerrit:gerrit -R /usr/local/gerrit-2.15.13
5 su - gerrit
6 java -jar /opt/gerrit-2.16.8.war init -d /usr/local/gerrit-2.15.13
7 git config --file /usr/local/gerrit-2.15.13/etc/gerrit.config gitweb.cgi
     /var/www/git/gitweb.cgi
8 git config --file /usr/local/gerrit-2.15.13/etc/gerrit.config --unset git
     web.url
```

安装httpd作为代理

```
1 yum -y install httpd
2
3 vi /etc/httpd/conf/httpd.conf
4 <VirtualHost *:80>
5   ServerName localhost
6   ProxyRequests Off
7   ProxyVia Off
8   ProxyPreserveHost On
9   AllowEncodedSlashes On
10
11 <Proxy *:8080>
12 Order deny,allow
13 Allow from all
14 </Proxy>
15
16 <Location /login/>
```

```
17 AuthType Basic
18 AuthName "Gerrit Code Review"
19 Require valid-user
20 AuthUserFile /etc/httpd/conf/passwords
21 </Location>
22
23 ProxyPass / http://192.168.21.143:8080/ nocanon
24 </VirtualHost>
```

安装nginx作为代理

```
1 yum -y install httpd-tools nginx
2
3 vi /etc/nginx/nginx.conf
4 user nobody;
5 worker_processes auto;
6 error_log logs/error.log warn;
7 worker_cpu_affinity auto;
8 pid logs/nginx.pid;
9 worker_rlimit_nofile 65535;
10
11 events {
12     worker_connections 65535;
13     use epoll;
14 }
15
16 http {
17     include mime.types;
18     default_type text/plain;
19
20     log_format main '$remote_addr - $remote_user [$time_local] "$request" '
21     '$status $body_bytes_sent "$http_referer" '
22     '"$http_user_agent" "$http_x_forwarded_for"';
23
24     access_log logs/access.log main;
25     sendfile on;
26     keepalive_timeout 120;
27     client_header_buffer_size 128k;
28     client_max_body_size 2048m;
29     client_header_timeout 1m;
30     client_body_timeout 1m;
```

```
31 send_timeout 1m;
32 ssi on;
33 ssi_silent_errors on;
34 gzip on;
35 gzip_static on;
36 gzip_http_version 1.0;
37 gzip_vary on;
38 gzip_proxied any;
39 gzip_comp_level 9;
40 gzip_min_length 100;
41 gzip_types text/plain application/x-javascript text/css application/xml
text/javascript application/javascript;
42 server_tokens off;
43 proxy_ignore_client_abort on;
44
45 server {
46   listen 80;
47   server_name localhost;
48   location / {
49     auth_basic "Gerrit Code Review";
50     auth_basic_user_file /etc/nginx/passwords;
51     proxy_pass http://192.168.21.143:8080;
52     proxy_set_header X-Forwarded-For $remote_addr;
53     proxy_set_header Host $host;
54   }
55 }
56 }
```

创建http认证

切记一定要用jenkins用户（也就是免密用户，否则是一个坑）

这个用户会成为gerrit的管理者

```
1 htpasswd -c /etc/httpd/conf/passwords gerrit
```

```
1 systemctl start httpd
2 systemctl enable httpd
3 su - gerrit
4 cd /usr/local/gerrit-2.15.13/plugin
5 wget https://gerrit-ci.gerritforge.com/job/plugin-events-log-bazel-
stable-2.15/lastSuccessfulBuild/artifact/bazel-genfiles/plugins/events-
log/events-log.jar
```

配置Gerrit

```
1 cat /usr/local/gerrit-2.15.13/etc/gerrit.config
2 [gerrit]
3 basePath = git
4 serverId = 77ce6b6a-742e-4980-beb4-2b5f83100e9a
5 canonicalWebUrl = http://192.168.21.143:8080/
6 [database]
7 type = mysql
8 hostname = 192.168.21.144
9 database = reviewdb
10 username = gerrit
11 useSSL = false
12 [noteDb "changes"]
13 disableReviewDb = true
14 primaryStorage = note db
15 read = true
16 sequence = true
17 write = true
18 [index]
19 type = LUCENE
20 [auth]
21 type = HTTP
22 [receive]
23 enableSignedPush = false
24 [sendemail]
25 enable = true
26 smtpServer = smtp.bochtec.com
27 smtpServerPort = 465
28 smtpEncryption = SSL
29 sslVerify = true
30 smtpUser = baosong@bochtec.com
31 from = baosong@bochtec.com
32 [container]
33 user = gerrit
34 javaHome = /usr/local/jdk1.8.0_212/jre
35 [sshd]
36 listenAddress = *:29418
37 [httpd]
38 listenUrl = http://*:8080/
```

```
39 [cache]
40   directory = cache
41 [plugins]
42   allowRemoteAdmin = true
43 [plugin "events-log"]
44   maxAge = 20
45   returnLimit = 10000
46   storeUrl = jdbc:h2:/usr/local/gerrit-2.15.13/data/events-log
47   urlOptions = loglevel=INFO
48   urlOptions = logUnclosedConnections=true
49   copyLocal = true
50 [gitweb]
51   cgi = /var/www/git/gitweb.cgi
52   type = gitweb
```

```
1 cat /usr/local/gerrit-2.15.13/etc/secure.config
2 [database]
3   password = dBwcGEeC3BGx
4 [auth]
5   registerEmailPrivateKey = RSWM+n1/MxWdBwcGEeC3BGx+cS+t11Wlknc=
6 [sendemail]
7   smtpPass = dBwcGEeC3BGx
```

启动服务

```
1 cd /usr/local/gerrit-2.15.13/bin
2 ./gerrit.sh restart
```

免密登陆

```
1 su - gerrit
2 ssh-keygen -t rsa -C gerrit@bochtec.com
```

输入gerrit/jenkins公钥，并拷贝到相应Server Host Key到`~/.ssh/known_hosts`

Settings

Profile
Preferences
Diff Preferences
Edit Preferences
Watched Projects
Contact Information
SSH Public Keys
HTTP Password
Identities
Groups

Status	Algorithm	Key	Comment
<input checked="" type="checkbox"/>	ssh-rsa	AAA3Bf...ClyzEAAAQABAAQ...N0BPrzT8t...	gerit@bochtec.com
<input checked="" type="checkbox"/>	ssh-rsa	AAA3Bf...ClyzEAAAQABAAQ...A7Thv9Ay...	jenkins@bochtec.com

Add SSH Public Key

How to Generate an SSH Key

Server Host Key

Fingerprint:
9a:b6:32:05:8c:E2:2b:89:8d:a0:b1:97:f7:34:5e:18
Entry for '/.ssh/known_hosts':
[10.10.10.105] 29418 ssh-rsa AAAAE2VjZD95LJ3HvTTIbwlzB0a0...k9fT22p... □

Server Host Key

Fingerprint:
09:31:40:45:4c:c0:c2:23:09:7b:a0:15:10:64:d4:48
Entry for '/.ssh/known_hosts':
[10.10.10.105] 29418 ssh-rsa AAAAE2VjZD95LJ3HvTTIbwlzB0a0...k9fT22p... □

Server Host Key

Fingerprint:
9a:0b:11:a0:49:0e:5d:47:94:64:2e:c6:81:a0:ab:a0
Entry for '/.ssh/known_hosts':
[10.10.10.105] 29418 ssh-rsa AAAAE2VjZD95LJ3HvTTIbwlzB0a0...k9fT22p... □

Server Host Key

Fingerprint:
c3:9b:88:08:9a:ae:c4:15:29:40:9b:82:a0:5a:68
Entry for '/.ssh/known_hosts':
[10.10.10.105] 29418 ssh-rsa AAAAE2VjZD95LJ3HvTTIbwlzB0a0...k9fT22p... □

10.10.10.105

如果此用户和免密用户不同，会造成下面报错，在ssh -vv -p 29418 gerrit@10.10.10.105 测试时候会一直报错

```
1 no more authentication methods to try.
2 permission denied
```

验证登陆是否成功，并配置ssh-gerrit

```
1 # .bashrc
2
3 # Source global definitions
4 if [ -f /etc/bashrc ]; then
5   . /etc/bashrc
6 fi
7
8 # Uncomment the following line if you don't like systemctl's auto-paging
feature:
9 # export SYSTEMD_PAGER=
10
11 # User specific aliases and functions
12 alias ssh-gerrit='ssh -p 29418 -i ~/.ssh/id_rsa 192.168.21.143 -l gerri
t'
```

```
1 ssh-gerrit
2
3 **** Welcome to Gerrit Code Review ****
4
5 Hi 系统管理员，you have successfully connected over SSH.
```

```
6
7 Unfortunately, interactive shells are disabled.
8 To clone a hosted Git repository, use:
9
10 git clone ssh://gerrit@192.168.21.143:29418/REPOSITORY_NAME.git
11
12 Connection to 192.168.21.143 closed.
```

gitlab服务器操作

安装gitlab

使用yum安装本地包，可以将关联的其他包一起安装

```
1 wget https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el7/gitlab-ce-11.
10.4-ce.0.el7.x86_64.rpm
2 yum install gitlab-ce-11.10.4-ce.0.el7.x86_64.rpm postfix openssh-server
cronie git wget patch
3 gitlab-ctl reconfigure
4 gitlab-ctl status
5 sed -i "s#external_url 'http://gitlab.example.com'#external_url 'http://1
92.168.21.142#g" /etc/gitlab/gitlab.rb
6 gitlab-ctl reconfigure
7 gitlab-ctl restart
```

初始化密码

```
1 gitlab-rails console production
2 u=User.where(id:1).first
3 u.password='12345678'
4 u.password_confirmation='12345678'
5 u.save!
```

```
[root@gitlab-21.142 ~]# gitlab-rails console production
-----
GitLab:   11.10.0 (62c464651d2)
Rails:    5.2.4 (f1a2a2a2)
PostgreSQL: 9.6.11
-----
Loading production environment (Rails 5.0.7.2)
irb(main):001:0> u=User.where(id:1).first
=> #<User:0x0000000000000000>
irb(main):001:0> u.password='12345678'
=> "12345678"
irb(main):001:0> u.password_confirmation='12345678'
=> "12345678"
irb(main):004:0> u.save!
=> true
irb(main):005:0>
```

添加组

New group

Groups allow you to manage and collaborate across multiple projects. Members of a group have access to all of its projects.

Groups can also be nested by creating subgroups.

Projects that belong to a group are prefixed with the group namespace. Existing projects may be moved into a group.

Group name: bc

Group URL: http://10.10.10.104/bc

Group description (optional):

Group avatar: Choose file... No file chosen

The maximum file size allowed is 200KB.

Visibility level:

- Private: The group and its projects can only be viewed by members.
- Internal: The group and any internal projects can be viewed by any logged in user.
- Public: The group and any public projects can be viewed without any authentication.

Create group **Cancel**

正在等待 10.10.10.104 的响应... 10.10.10.104

新建项目

New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), among other things.

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

Information about additional Pages templates and how to install them can be found in our Pages getting started guide.

Tip: You can also create a project from the command line. Show command

Blank project	Create from template	Import project
Project name: test	Project slug: test	
Project URL: http://10.10.10.104/bc		
Want to house several dependent projects under the same namespace? Create a group .		
Project description (optional):		
Description format:		
Visibility Level <small>?</small>		
<input checked="" type="radio"/> Private: Project access must be granted explicitly to each user. <input type="radio"/> Internal: This project cannot be internal because the visibility of bc is private. To make this project internal, you must first change the visibility of the parent group. <input type="radio"/> Public: This project cannot be public because the visibility of bc is private. To make this project public, you must first change the visibility of the parent group.		
<input type="checkbox"/> Initialize repository with a README: Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.		
Create project		Cancel

10.10.10.104/groups/new 10.10.10.104

添加gerrit、jenkins公钥

The screenshot shows the GitLab User Settings interface. The left sidebar is titled 'User Settings' and includes options like Profile, Account, Applications, Chat, Access Tokens, Emails, Password, Notifications, SSH Keys (which is selected), GPG Keys, Preferences, Active Sessions, and Authentication log. The main content area is titled 'SSH Keys' and explains that SSH keys allow for secure connections between the user's computer and GitLab. It provides instructions for adding an SSH key, mentioning that it should start with 'ssh-rsa'. A text input field for 'Key' is shown with placeholder text 'Typically starts with "ssh-rsa ..."'. Below this is a 'Title' input field with placeholder 'e.g. My MacBook key' and a 'Name your individual key via a title' input field. A 'Add key' button is present. At the bottom, a section titled 'Your SSH keys (2)' lists two keys: 'jenkins@bochtec.com' (last used: n/a, created just now) and 'gerritt@bochtec.com' (last used: n/a, created just now). A status bar at the bottom right shows the IP address '10.10.10.104'.

jenkins服务器操作

jenkins安装插件

- 1 SonarQube Scanner
- 2 Gitlab
- 3 Build Authorization Token Root
- 4 Publish Over SSH
- 5 Gitlab Authentication
- 6 Maven Integration
- 7 Ant
- 8 Ansible
- 9 Email Extension Plugin
- 10 Locale plugin
- 11 SonarQube Scanner for Jenkins
- 12 Gerrit Trigger
- 13 NodeJS

配置Gerrit Trigger

Jenkins Gerrit Trigger > Check To Gerrit

Gerrit Connection Setting

Name: Check To Gerrit
No Connection On Startup:
Hostname: 10.10.10.105
Frontend URL: http://10.10.10.105/
SSH Port: 29418
Proxy:
Username: gerrit
E-mail:
SSH Keyfile: /var/lib/jenkins/.ssh/id_rsa
SSH Keyfile Password:
 Build Current Patches Only
Success:
Test Connection

Gerrit Reporting Values

Verify

Started	0
Successful	1
Failed	-1
Unstable	0
Not Built	0

Code Review

Code Review	10.10.10.103
-------------	--------------

这步可以消除黄色那句话，哈哈哈

Jenkins Gerrit Trigger > Check To Gerrit

Notification Level: All

Verdict Categories

Verdict Value: Code-Review Verdict Description: Code Review Add Verdict Category	Verdict Value: Verified Verdict Description: Verified Add Verdict Category
--	--

REST API

Use REST API
Gerrit HTTP Username: gerrit
Gerrit HTTP Password:

Test REST Connection

Enable Code-Review:

Replication Events

Block Build Until

Watchdog

Connection Watch
Allow Other Plugin

Settings

Username: gerrit
Password: a3tvt*#ta3v#44:84u9/t3t11F94ax9VduJ1g
Generate Password

192.168.21.143 //settings/http-password

The screenshot shows the Jenkins Global Configuration page. At the top, there are links for Back to Dashboard, Add New Server, and Diagnostics. The main section is titled "Global Configuration" and contains three input fields: "No. of Receiving Worker Threads" set to 3, "No. of Sending Worker Threads" set to 1, and "Replication Cache Expiration" set to 360. Below this is a "Save" button. The next section is titled "Gerrit Servers" and shows a table with one row for "Check To Gerrit" (version 2.15.13). The table has columns for Server Name, Version, Status, Edit, and Remove. The status column shows a green circle icon. The bottom right corner of the page displays the generation time: 生成页面: 2019-5-30 下午02时43分58秒 | 192.168.21.144 | Jenkins-2.15.13

gerrit服务器操作

检查插件

The screenshot shows the Gerrit Code Review user profile settings page. The top navigation bar includes links for All, My Projects, People, Plugins, and Documentation. Below this is a sub-navigation bar with links for List, General, Branches, Tags, Access, Dashboards, and Create New Project. A "Settings" sidebar on the left lists options like Profile, Preferences, Diff Preferences, Watched Projects, Contact Information, SSH Public Keys, HTTP Password, Identities, and Groups. The main content area displays the user's profile information: Username (gerrit), Full Name (系统管理员), Email Address (未设置), Registered (May 30, 2019 1:52 PM), and Account ID (1000000). On the right, there is a search bar, a "Changes" link, and a "Sign Out" button. The bottom right corner shows the IP address 192.168.21.143.

创建组Developer、Reviewer

Groups

Group Name	Description	Visible To All
Administrators	Gerrit Site Administrators	
Developer		
Non-Interactive Users	Users who perform batch actions on Gerrit	
Reviewer		

Powered by [Gerrit Code Review](#) (2.15.13) | [Switch to New UI](#) | Press "F" to view keyboard shortcuts

精简定制权限

Project All-Projects

Edits History: (githw)

Global Capabilities

Administrator Server	ALLOW ▾	Administrators	<input type="checkbox"/>	Exclusive
Priority	BATCH	Non-Interactive Users		
Stream Events	ALLOW ▾	Non-Interactive Users		

Reference: refs/*

Read	<input type="checkbox"/>	Exclusive
ALLOW ▾	Administrators	
ALLOW ▾	Anonymous Users	
Abandon	<input type="checkbox"/>	Exclusive
ALLOW ▾	Reviewer	
Forge Author Identity	<input type="checkbox"/>	Exclusive
ALLOW ▾	Developer	
ALLOW ▾	Reviewer	
Push	<input type="checkbox"/>	Exclusive
ALLOW ▾	Developer	
ALLOW ▾	Reviewer	
Label Code-Review	<input type="checkbox"/>	Exclusive
ALLOW ▾	-2 ▾ +2 ▾	Reviewer
ALLOW ▾	-1 ▾ +1 ▾	Developer
Label Verified	<input type="checkbox"/>	Exclusive
ALLOW ▾	-1 ▾ +1 ▾	Developer
Rebase	<input type="checkbox"/>	Exclusive
ALLOW ▾	Reviewer	
Submit	<input type="checkbox"/>	Exclusive
ALLOW ▾	Reviewer	

Reference: refs/heads/*

Create Reference	<input type="checkbox"/>	Exclusive
DENY ▾	Project Owners	
ALLOW ▾	Administrators	
Submit	<input type="checkbox"/>	Exclusive
DENY ▾	Project Owners	
ALLOW ▾	Administrators	

Reference: refs/meta/config

Read	<input checked="" type="checkbox"/>	Exclusive
ALLOW ▾	Administrators	



192.168.21.143

192.168.21.143

ALLOW ▾ Developer	Force Push
ALLOW ▾ Reviewer	Force Push
Label Code-Review	
ALLOW ▾ -2 ▾ +2 ▾ Reviewer	Exclusive
ALLOW ▾ -1 ▾ +1 ▾ Developer	Exclusive
Label Verified	
ALLOW ▾ -1 ▾ +1 ▾ Developer	Exclusive
Release	
ALLOW ▾ Reviewer	Exclusive
Submit	
ALLOW ▾ Reviewer	Exclusive
Reference: refs/heads/*	
Create Reference	Exclusive
DENY ▾ Project Owners	
ALLOW ▾ Administrators	
Submit	
DENY ▾ Project Owners	Exclusive
ALLOW ▾ Administrators	
Reference: refs/meta/config	
Read	<input checked="" type="checkbox"/> Exclusive
ALLOW ▾ Administrators	
ALLOW ▾ Project Owners	
Create Reference	
DENY ▾ Project Owners	Exclusive
ALLOW ▾ Administrators	
Push	
ALLOW ▾ Administrators	Exclusive
ALLOW ▾ Project Owners	Force Push
Label Code-Review	
DENY ▾ -2 ▾ +2 ▾ Change Owner	Exclusive
ALLOW ▾ -2 ▾ +2 ▾ Administrators	
ALLOW ▾ -1 ▾ +1 ▾ Project Owners	Exclusive
Submit	
DENY ▾ Project Owners	Exclusive
ALLOW ▾ Administrators	
Reference: refs/tags/*	
Create Reference	Exclusive
DENY ▾ Project Owners	
ALLOW ▾ Administrators	
Create Annotated Tag	
DENY ▾ Project Owners	Exclusive
ALLOW ▾ Administrators	
Create Signed Tag	
DENY ▾ Project Owners	Exclusive
ALLOW ▾ Administrators	

Powered by Gerrit Code Review (2.15.13) | Switch to New UI | Press '?' to view keyboard shortcuts | 192.168.21.143

开启邮件，更改日期显示，显示项目序号

All My Projects People Plugins Documentation

List Create New Project

Search term

Changes Search 系统管理员

Settings

Profile Preferences Diff Preferences Edit Preferences Watched Projects Contact Information SSH Public Keys HTTP Password Identities Groups

Display In Review Category:	None (default)
Maximum Page Size:	50 rows per page
Date/Time Format:	05-31-2019-05-31 10:22
Email Notifications:	Only Comments Left By Others
Email Format:	HTML and PlainText
Default Base For Merges:	First Parent
Diff View:	Side by Side
<input checked="" type="checkbox"/> Show Site Header / Footer <input checked="" type="checkbox"/> Highlight Changes Assigned To Me In Changes Table <input type="checkbox"/> Show Relative Dates In Changes Table <input checked="" type="checkbox"/> Show Change Sizes As Colored Bars <input checked="" type="checkbox"/> Show Change Number In Changes Table <input checked="" type="checkbox"/> Mute Common Path Prefixes In File List <input type="checkbox"/> Insert Signed-off-by Footer For Inline Edit Changes <input type="checkbox"/> Publish Draft Comments When A Change Is Updated by Push <input type="checkbox"/> Set all new changes work-in-progress by default	

My Menu

Name	URL
Changes	#/dashboard/self
Draft Comments	#/q/hassdraft
Edits	#/q/hassedit
Watched Changes	#/q/iswatched+isopen
Starred Changes	#/q/isstarred
Groups	#/groups/self

Save Changes

Powered by Gerrit Code Review (2.15.13) | Switch to New UI | Press '?' to view keyboard shortcuts

192.168.21.143

开启所有项目监听，发送邮件

Powered by Gerrit Code Review (2.15.13) | [Switch to New UI](#) | Press "?" to view keyboard shortcuts

192.168.21.143

在jenkins服务器测试

```
1 mkdir /opt/src
2 chown jenkins:jenkins /opt/src
3 su - gerrit
4 git config --global user.name "gerrit"
5 git config --global user.email "gerrit@bochtec.com"
6 git clone git@192.168.21.142:bc/test.git
7 touch README.md
8 git add README.md
9 git commit -m "add README"
10 git push -u origin master
11 vi .gitreview
12 =====
13 [gerrit]
14 host=192.168.21.143
15 port=29418
16 project=test.git
17 defaultbranch=develop #这里指向了 develop 分支， 默认不写master
18
19 =====
20 git add .gitreview
21 git commit .gitreview -m 'add .gitreview file by gerrit'
22 git push origin master
```

测试流程

```
1 git clone git@192.168.21.142:bc/test.git
2 Cloning into 'test'...
3 warning: You appear to have cloned an empty repository.
4 cd test/
5 ls
6 touch README.md
7 git add README.md
8 git commit -m "add README"
9 [master (root-commit) 32e5ff0] add README
10 1 file changed, 0 insertions(+), 0 deletions(-)
11 create mode 100644 README.md
12 git push -u origin master
13 Counting objects: 3, done.
14 Writing objects: 100% (3/3), 211 bytes | 0 bytes/s, done.
15 Total 3 (delta 0), reused 0 (delta 0)
16 To git@192.168.21.142:bc/test.git
17 * [new branch] master -> master
18 Branch master set up to track remote branch master from origin.
19 vi .gitreview
20 git add .gitreview
21 git commit .gitreview -m 'add .gitreview file by gerrit'
22 [master b19fac5] add .gitreview file by gerrit
23 1 file changed, 4 insertions(+)
24 create mode 100644 .gitreview
25 git push origin master
26 Counting objects: 4, done.
27 Delta compression using up to 2 threads.
28 Compressing objects: 100% (3/3), done.
29 Writing objects: 100% (3/3), 339 bytes | 0 bytes/s, done.
30 Total 3 (delta 0), reused 0 (delta 0)
31 To git@192.168.21.142:bc/test.git
32 32e5ff0..b19fac5 master -> master
```

在gerrit服务器配置项目

```
1 su - gerrit
2 ssh-gerrit gerrit create-project test
```

gerrit上的项目最好是从gitlab上git clone --bare过来，并且项目不要为空
使用gerrit用户操作

```

1 cd /usr/local/gerrit-2.15.13/git
2 rm -rf test.git
3 git clone --bare git@192.168.21.142:/bc/test.git

```

```

[gerrit@gerrit_10_105 git]$ pwd
/usr/local/gerrit-2.15.13/git
[gerrit@gerrit_10_105 git]$ cd /usr/local/gerrit-2.15.13/git
[gerrit@gerrit_10_105 git]$ ls
All-Projects.git All-Users.git test.git
[gerrit@gerrit_10_105 git]$ rm -rf test.git/
[gerrit@gerrit_10_105 git]$ ls
All-Projects.git All-Users.git
[gerrit@gerrit_10_105 git]$ git clone --bare git@10.10.10.104:/bc/test.git
Cloning into bare repository 'test.git'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (6/6), done.
[gerrit@gerrit_10_105 git]$ ls -al
total 0
drwxrwxr-x 5 gerrit gerrit 67 May 24 17:06 .
drwxr-xr-x 15 gerrit gerrit 167 May 24 14:20 ..
drwxrwxr-x 7 gerrit gerrit 119 May 24 13:56 All-Projects.git
drwxrwxr-x 7 gerrit gerrit 100 May 24 13:56 All-Users.git
drwxrwxr-x 7 gerrit gerrit 138 May 24 17:06 test.git
[gerrit@gerrit_10_105 git]$ ls test.git/
branches config description HEAD hooks info objects packed-refs refs
[gerrit@gerrit_10_105 git]$

```

配置gerrit和gitlab同步

remote "test"一定要和gitlab的同命名，后续可直接复制此段落即可添加新项目

```

1 vi /usr/local/gerrit-2.15.13/etc/replication.config
2 [remote "test"]
3   projects = test

```

```
4 url = git@10.10.10.104:bc/test.git
5 push = +refs/heads/*:refs/heads/*
6 push = +refs/tags/*:refs/tags/*
7 push = +refs/changes/*:refs/changes/*
8 timeout = 30
9 threads = 3
```

设置gerrit用户的 ~/.ssh/config

~/.ssh/config文件的权限一定要设置成600

```
1 vi /home/gerrit/.ssh/config
2 Host 10.10.10.104:
3   IdentityFile ~/.ssh/id_rsa
4   PreferredAuthentications publickey
```

与gitlab服务器免密

```
1 sh -c "ssh-keyscan -t rsa 192.168.21.142 >> /home/gerrit/.ssh/known_hosts"
2 sh -c "ssh-keygen -H -f /home/gerrit/.ssh/known_hosts"
```

重启gerrit服务

```
1 /usr/local/gerrit-2.15.13/bin/gerrit.sh restart
```

普通用户测试，切记一定要包含commit-msg hook

```
1 第二次创建用户把-c删除
2 htpasswd /etc/httpd/conf/passwords baosong
3 useradd baosong
4 su - baosong
5 ssh-keygen -t rsa -C baosong@bochtec.com
6 cat /home/baosong/.ssh/id_rsa.pub
7 将known_hosts添加gerrit里面的信息，做互信
```

访问gerrit服务，并激活注册邮箱，否则git-review的时候会报错

```
[user1@jenkins_10_103 test]$ ls
eee fff ggg README.md
[user1@jenkins_10_103 test]$ git-review
remote: Processing changes: refs: 1, done
remote: error: commit 6e155c3: email address user1@bochtec.com is not registered in your account, and you lack 'forge committer' permission.
remote: You have not registered any email addresses.
remote: To register an email address, visit:
remote: http://10.10.10.105:8080/#settings/contact
remote:
remote:
remote: Pushing to refs/publish/* is deprecated, use refs/for/* instead.
To ssh://user1@10.10.10.105:29418/test.git
 ! [remote rejected] HEAD -> refs/publish/master (commit 6e155c3: invalid committer)
error: failed to push some refs to 'ssh://user1@10.10.10.105:29418/test.git'
[user1@jenkins_10_103 test]$
```

```
1 git clone "http://gerrit@192.168.21.143:8080/a/test" && (cd "test" && mkdir .git/hooks && curl -Lo `git rev-parse --git-dir`/hooks/commit-msg ht
```

```

1 p://gerrit@192.168.21.143:8080/tools/hooks/commit-msg; chmod +x `git rev-parse --git-dir`/hooks/commit-msg)
2 git config --global user.name "gerrit"
3 git config --global user.email "baosong@bochtec.com"
4 touch ddd
5 git add ddd
6 git commit -m "baosong add ddd"
7 git review

```

```

[test@gerrit_10_105 test]$ git-review
Creating a git remote called "gerrit" that maps to:
  ssh://test@10.10.105:29418/test.git
remote: Processing changes: new: 1, done
remote:
remote: New Changes:
remote:   http://10.10.10.105:8080/#/c/test/+/41 test add bbb
remote:
remote: Pushing to refs/publish/* is deprecated, use refs/for/* instead.
To ssh://test@10.10.10.105:29418/test.git
 * [new branch]      HEAD -> refs/publish/master
[test@gerrit_10_105 test]$

```

jenkins配置

The screenshot shows the Jenkins configuration interface for a job named 'test'. The 'Source Management' tab is selected, displaying two repository configurations:

- Repository 1:** URL: ssh://gerrit@10.10.105:29418/test, Credentials: None.
- Repository 2:** URL: git@10.10.10.104:bc/test.git, Credentials: None.

The 'Build Environment' tab is visible below, followed by the 'Post-Build Actions' tab at the bottom.

Windows操作

服务器操作创建相应人员私密、公钥

`ssh-keygen -t rsa -C daibing@bochtec.com`

```
[gerrit@gerrit_21_143 ssh_key]$ ssh-keygen -t rsa -C daibing@bochtec.com
Generating public/private rsa key pair.
Enter file in which to save the key (/home/gerrit/.ssh/id_rsa): /home/gerrit/ssh_key/daibing
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gerrit/ssh_key/daibing.
Your public key has been saved in /home/gerrit/ssh_key/daibing.pub.
The key fingerprint is:
SHA256:fJLITBcU0498212/ao0vTczvjVYrgiUtmBC3LHuJelo daibing@bochtec.com
The key's randomart image is:
+---[RSA 2048]---+
 000*.=0  o*
 00.* .. o.+
 ..0+ . o .+
 ++=.= .o.
 o+oS + .o|
 .. + o ..o|
 . E    . . o|
 +       ...o|
 .       ...o|
+---[SHA256]---+
[gerrit@gerrit_21_143 ssh_key]$ ls
daibing  daibing.pub
```

将daibing改名发送给相应人员，重命名为id_rsa，存放在/c/User/dell/.ssh下

激活邮件

All My Projects People Plugins Documentation

Open Merged Abandoned

Settings

Profile Preferences Diff Preferences Edit Preferences Watched Projects Contact Information SSH Public Keys HTTP Password Identities Groups

Username: Full Name: 系统管理员 Preferred Email: [Register New Email...](#)

Mail Filters [Save Changes](#)

Powered by Gerrit Code Review (2.15.13) | [Switch to New UI](#) | Press "?" to view keyboard shortcuts

192.168.21.143/#/settings/contact

192.168.21.143

安装git

 **git** --distributed-is-the-new-centralized

[Search entire site...](#)

[About](#) [Documentation](#) [Downloads](#) [Community](#)

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Downloads

 Mac OS X  Windows  Linux/Unix

Older releases are available and the Git source repository is on GitHub.

GUI Clients

Git comes with built-in GUI tools (`git-gui`, `gitk`), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

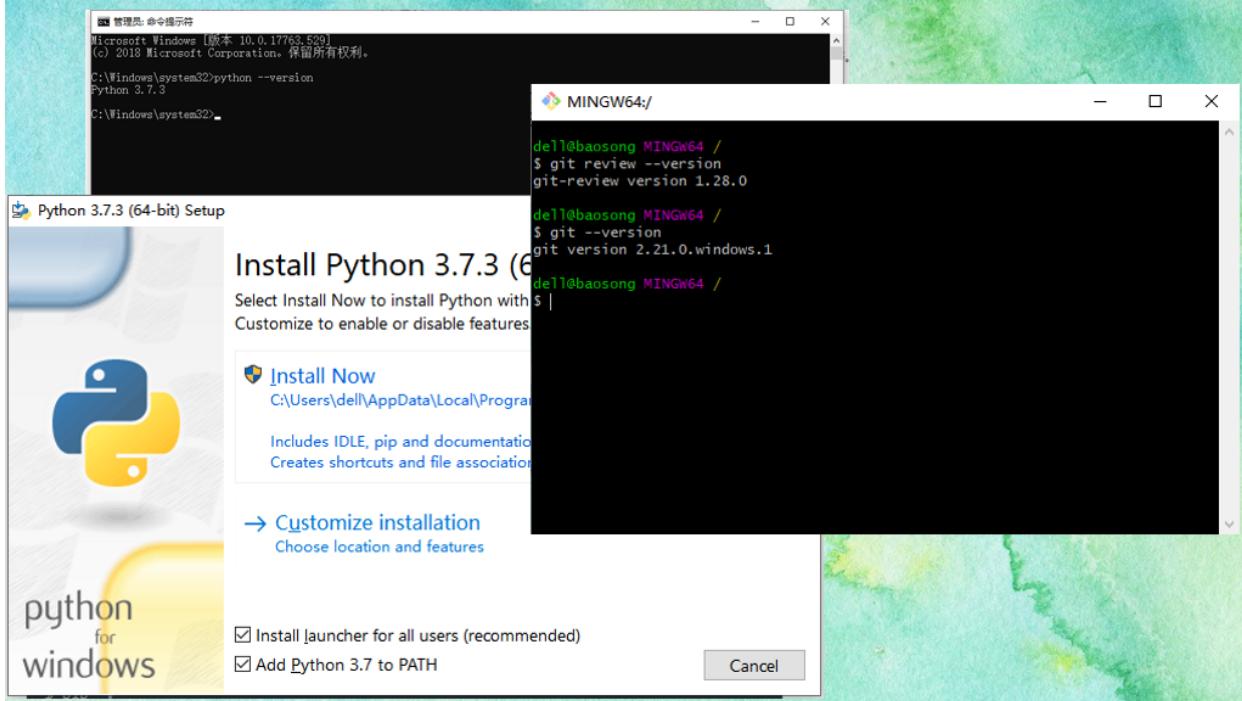
```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the web interface.

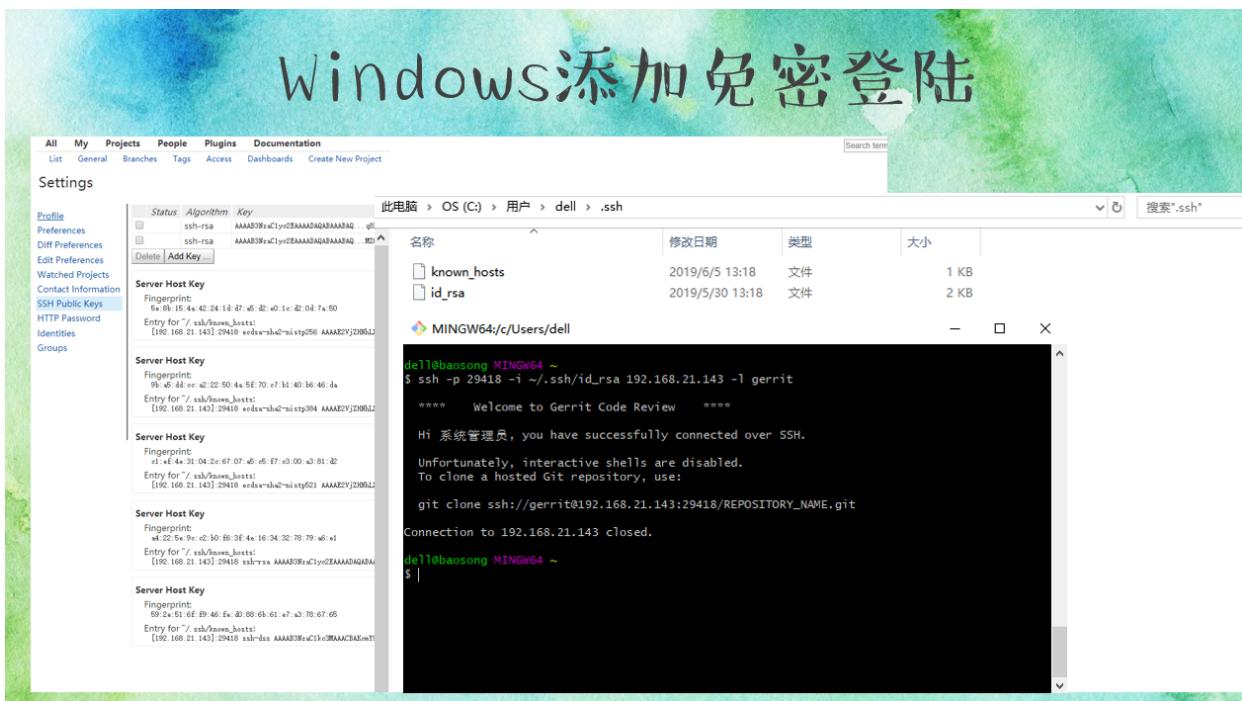
https://blog.csdn.net/qq_32786873/article/details/80570783

安装Python、pip、git-review

```
python -V  
pip -V  
pip install git-review  
git --version  
git review --version
```



免密



```
git config --global user.name "gerrit"  
git config --global user.email "gerrit@bochtec.com"
```

```
git add .
```

```
git commit -m "gerrit add cps"
```

```
git review
```

The terminal window shows the following sequence of commands and their output:

```
git config --global user.name "gerrit"
git config --global user.email "gerrit@bochtec.com"
git add .
git commit -m "gerrit add cps"
git review
```

Output:

```
dell@baosong MINGW64 ~/cps (master)
$ git config --global user.name "daibing"
dell@baosong MINGW64 ~/cps (master)
$ git config --global user.email "daibing@bochtec.com"
dell@baosong MINGW64 ~/cps (master)
$ git add .
dell@baosong MINGW64 ~/cps (master)
$ git commit -m "daibing add cps"
[master 1623072] daibing add cps
 4 files changed, 14 insertions(+), 0 deletions(-)
  create mode 100644 cps.iml
  create mode 100644 pom.xml
  create mode 100644 source/pom.xml
  create mode 100644 source/resource.iml
$ git review
ssh://dell@192.168.21.143:29418/cps.git did not work. Description: dell@192.168.21.143: Permission denied (publickey).
fatal: Could not read from remote repository.

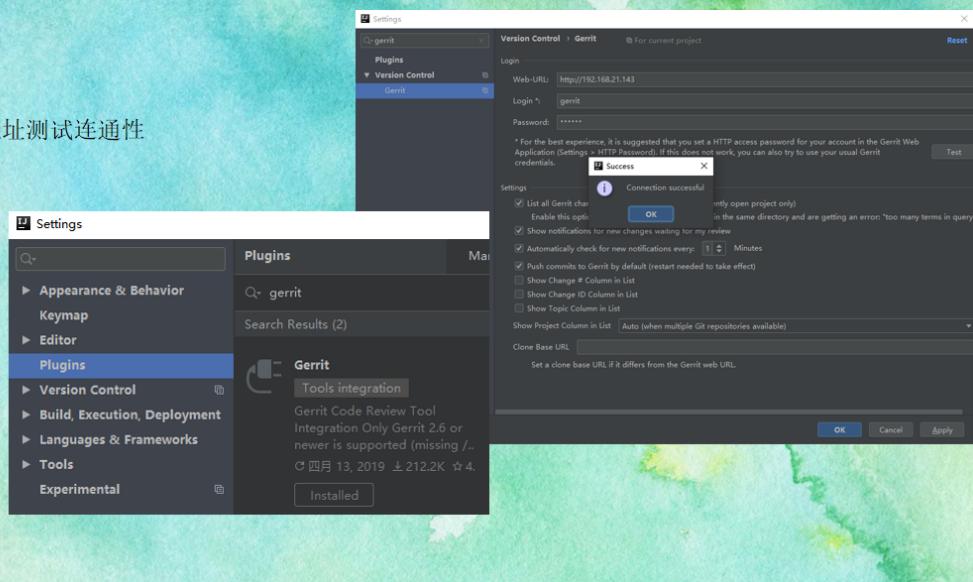
Please make sure you have the correct access rights
and the repository exists.
Could not connect to gerrit.
Enter your gerrit username: daibing
Trying again with ssh://daibing@192.168.21.143:29418/cps.git
Creating a git remote called 'gerrit' that maps to:
  ssh://daibing@192.168.21.143:29418/cps.git

This repository is now set up for use with git-review. You can set the
default username for future repositories with:
  git config --global --add gitreview.username "daibing"

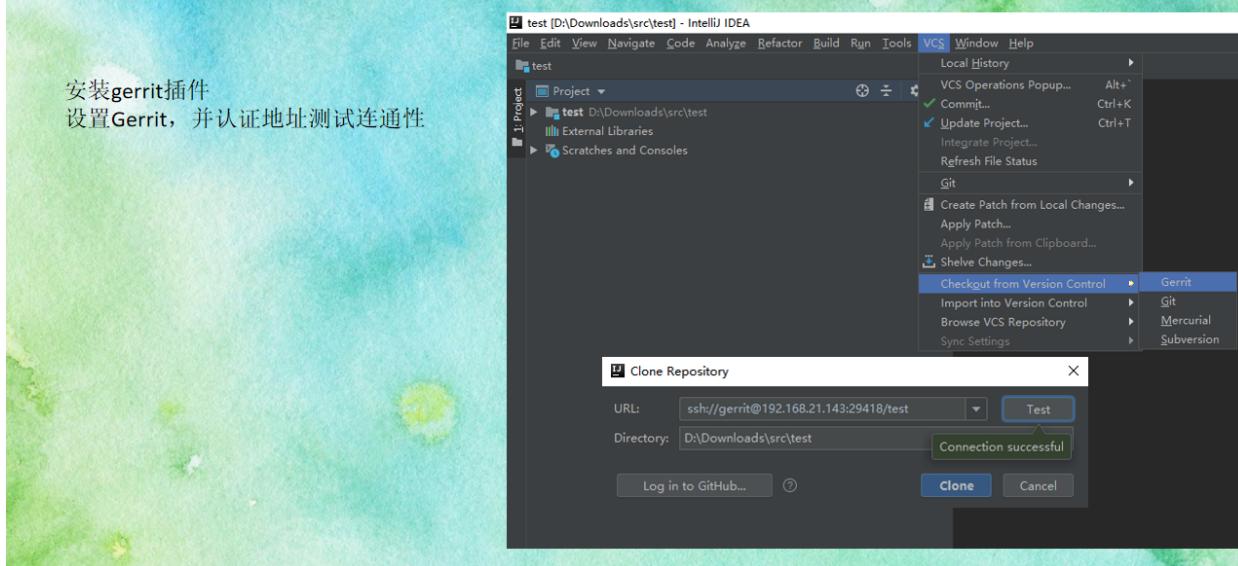
remote:
remote: Processing changes: new: 1 (.)
remote: Processing changes: new: 1 (.)
remote: Processing changes: new: 3 (.)
remote: Processing changes: new: 1 (.)
remote: Processing changes: new: 1 (-)
remote: Processing changes: new: 1, done
remote:
remote: New Changes:
remote:   http://192.168.21.143:8080/#/c/cps/+/41 daibing add cps
remote:
To ssh://192.168.21.143:29418/cps.git
 * [new branch]      HEAD -> refs/for/master
dell@baosong MINGW64 ~/cps (master)
```

IDEA与Gerrit结合

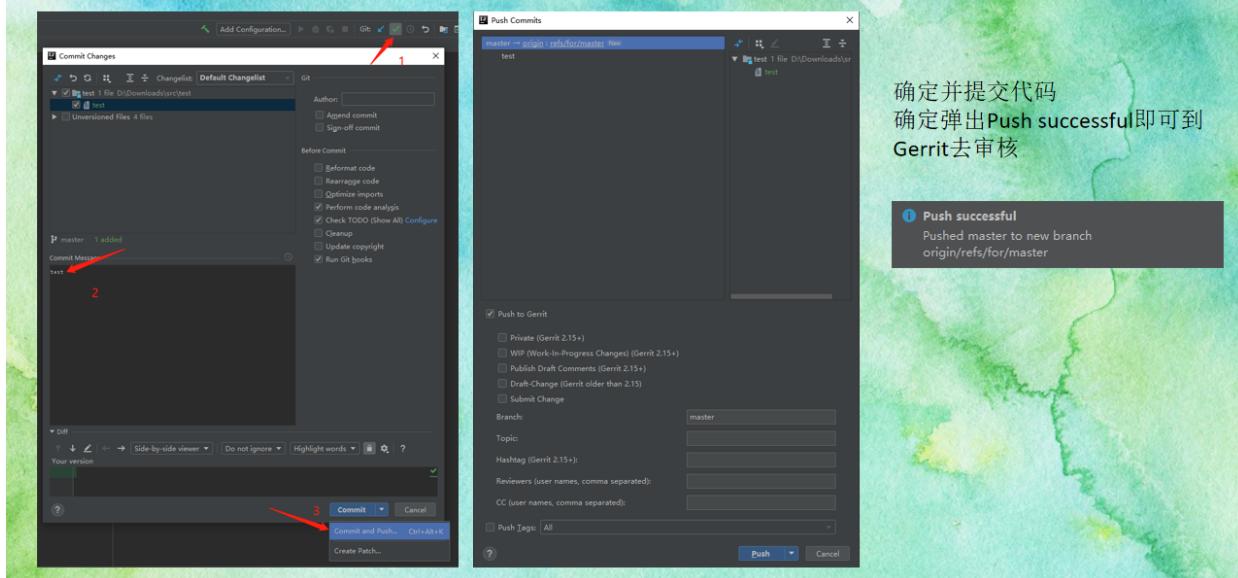
安装gerrit插件
设置Gerrit，并认证地址测试连通性



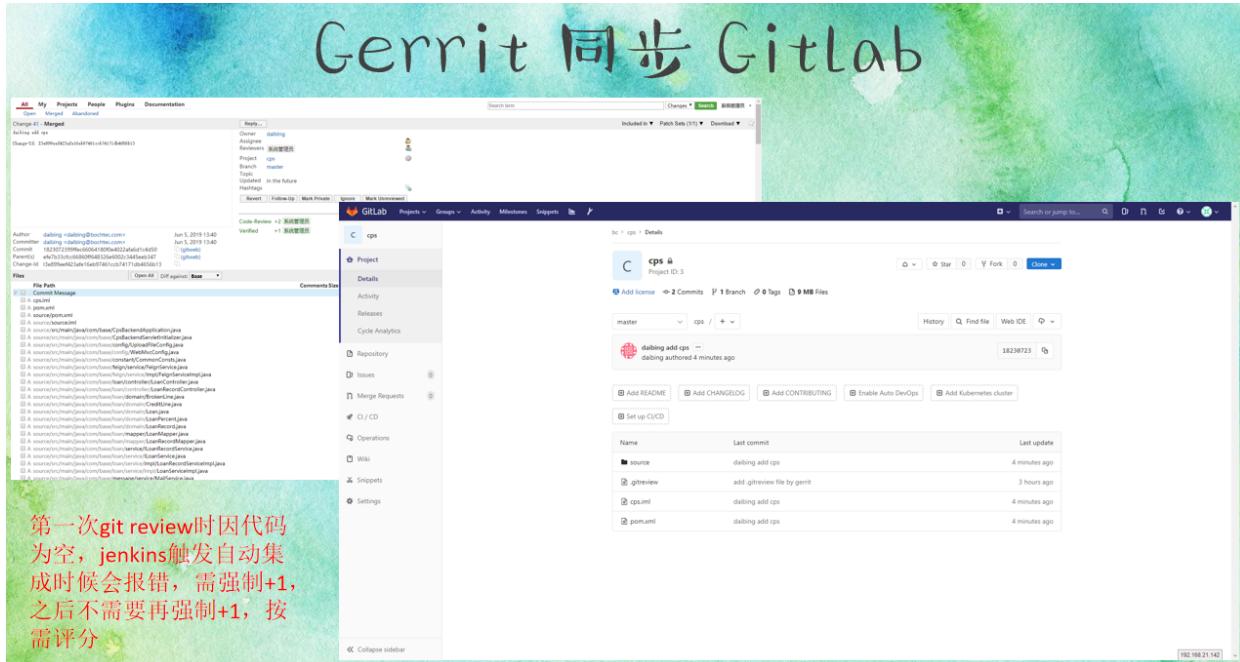
IDEA新建任务



IDEA提交代码



第一次git review时因代码为空，jenkins触发自动集成时候会报错，需强制+1，之后不需要再强制+1，按需评分



sonar配置——系统管理——系统设置

sonarqube-配置-权限-用户-创建令牌

SonarQube servers

Environment variables	<input checked="" type="checkbox"/> Enable injection of SonarQube server configuration as build environment variables If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.						
SonarQube installations	<table border="1"> <tr> <td>Name</td> <td>SonarQube</td> </tr> <tr> <td>Server URL</td> <td>http://192.168.20.15:9000</td> </tr> <tr> <td>Server authentication token</td> <td>..... SonarQube authentication token. Mandatory when anonymous access is disabled.</td> </tr> </table>	Name	SonarQube	Server URL	http://192.168.20.15:9000	Server authentication token SonarQube authentication token. Mandatory when anonymous access is disabled.
Name	SonarQube						
Server URL	http://192.168.20.15:9000						
Server authentication token SonarQube authentication token. Mandatory when anonymous access is disabled.						
<input type="button" value="高级..."/> <input type="button" value="Delete SonarQube"/>							
<input type="button" value="Add SonarQube"/>							

List of SonarQube installations

jenkins添加凭据

- 1、ssh的与gitlab的公钥私钥，建议与ansible共用
- 2、gitlab的api token

全局凭据 (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

名称	
gitlab	SSH Username with private key
GitLab API token	GitLab API token

图标: 小虫 大

gitlab设置——系统管理——系统设置

gitlab添加token (administrator-setting-access tokens)

Gitlab

Enable authentication for '/project' end-point

GitLab connections

Connection name	Jenkins
Gitlab host URL	http://192.168.20.131:10080/
Credentials	GitLab API token <input type="button" value="添加"/>

API Token for accessing Gitlab

高级... **Test Connection** **删除** **新增**

邮件配置——系统管理——系统设置

Jenkins Location

Jenkins URL: http://192.168.20.15:8080/

系统管理员邮件地址: yjdev@yunjingit.com

Extended E-mail Notification

SMTP server: smtp.bochtec.com

Default user E-mail suffix: @bochtec.com

Use SMTP Authentication

User Name: baosong@bochtec.com

Password:

Advanced Email Properties

Use SSL:

SMTP port: 465

Charset: UTF-8

Additional accounts:

Default Content Type: HTML (text/html)

Use List-ID E-mail Header:

Add 'Precedence: bulk' E-mail Header:

Default Recipients: baosong@bochtec.com

Reply To List:

Emergency reroute:

Allowed Domains:

Excluded Recipients:

Default Subject: [构建通知] \$PROJECT_NAME - Build # \$BUILD_NUMBER - \$BUILD_STATUS!

保存 **应用**

[构建通知]: \$PROJECT_NAME - Build # \$BUILD_NUMBER - \$BUILD_STATUS!

```
<!DOCTYPE html>
<head>
<meta charset="UTF-8">
<title>${ENV, var="JOB_NAME"}-第${BUILD_NUMBER}次构建日志</title>
</head>
<body leftmargin="8" marginwidth="0" topmargin="8" marginheight="4" offset="0">
<table width="95%" cellpadding="0" cellspacing="0" style="font-size: 11pt; font-family: Tahoma, Arial, Helvetica, sans-serif">
<tr>
<td><br/>
本邮件由系统自动发出，无需回复！<br/>
各位同事，大家好，以下为${PROJECT_NAME}项目构建信息<br/>
<td><font color="#CC0000">构建结果 - ${BUILD_STATUS}</font></td>
</tr>
<tr>
<td><br />
<b><font color="#0B610B">代码检测</font></b>
<hr size="2" width="100%" align="center" /></td>
</tr>
</table>

```

保存 **应用**

```

1 [构建通知]: $PROJECT_NAME - Build # $BUILD_NUMBER - $BUILD_STATUS!
2
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>${ENV, var="JOB_NAME"}-第${BUILD_NUMBER}次构建日志</title>
8 </head>
9
10 <body leftmargin="8" marginwidth="0" topmargin="8" marginheight="4"
11 offset="0">
12 <table width="95%" cellpadding="0" cellspacing="0" style="font-size: 11
pt; font-family: Tahoma, Arial, Helvetica, sans-serif">
13
14 <tr>
15 本邮件由系统自动发出，无需回复！<br/>
16 各位同事，大家好，以下为${PROJECT_NAME}项目构建信息<br/>
17 <td><font color="#CC0000">构建结果 - ${BUILD_STATUS}</font></td>
18 </tr>
19
20 <tr>
21 <td><br />
22 <b><font color="#0B610B">代码检测</font></b>
23 <hr size="2" width="100%" align="center" /></td>
24 </tr>
25
26 <tr>
```

```
27 <td>
28 <ul>
29 <li>Sonar Url: <a href="http://192.168.21.144:9000/dashboard?id=${PROJECT_NAME}">http://192.168.21.144:9000/dashboard?id=${PROJECT_NAME}</a></li>
30 </ul>
31
32 <tr>
33 <td><br />
34 <b><font color="#0B610B">构建信息</font></b>
35 <hr size="2" width="100%" align="center" /></td>
36 </tr>
37
38 <tr>
39 <td>
40 <ul>
41 <li>项目名称 : ${PROJECT_NAME}</li>
42 <li>构建编号 : 第${BUILD_NUMBER}次构建</li>
43 <li>触发原因: ${CAUSE}</li>
44 <li>构建状态: ${BUILD_STATUS}</li>
45 <li>构建日志: <a href="${BUILD_URL}console">${BUILD_URL}console</a>
</li>
46 <li>构建 Url : <a href="${BUILD_URL}">${BUILD_URL}</a></li>
47 <li>工作目录 : <a href="${PROJECT_URL}ws">${PROJECT_URL}ws</a></li>
48 <li>项目 Url : <a href="${PROJECT_URL}">${PROJECT_URL}</a></li>
49 </ul>
50
51 <h4><font color="#0B610B">失败用例</font></h4>
52 <hr size="2" width="100%" />
53 $FAILED_TESTS<br/>
54
55 <h4><font color="#0B610B">最近提交(#$GIT_PREVIOUS_SUCCESSFUL_COMMIT)</font></h4>
56 <hr size="2" width="100%" />
57 <ul>
58 ${CHANGES_SINCE_LAST_SUCCESS, reverse=true, format="%c", changesFormat="
<li>%d [%a] %m</li>"}
59 </ul>
60 详细提交: <a href="${PROJECT_URL}changes">${PROJECT_URL}changes</a><br/>
61
62 </td>
63 </tr>
```

```
64  </table>
65  </body>
66  </html>
```

The screenshot shows the Jenkins 'Global Mail' configuration page. It includes sections for 'Additional groovy classpath' (with a 'New...' button), 'Content Token Reference', and '邮件通知' (Email Notification). Under '邮件通知', there are fields for 'SMTP服务器' (smtp.bochtec.com), '用户名' (@bochtec.com), '密码' (baosong@bochtec.com), '使用SSL协议' (selected), 'SMTP端口' (465), 'Reply-To Address' (empty), '字符集' (UTF-8), and 'Test e-mail recipient' (baosong@bochtec.com). A message 'Email was successfully sent' is displayed below the test recipient field. Buttons for '保存' (Save) and '应用' (Apply) are at the bottom.

全局设置

🔒 全局安全配置

The screenshot shows the Jenkins 'Global Security' configuration page. It has sections for '启用安全' (Enable Security), '不要记住我' (Remember Me), and '访问控制'. Under '访问控制', there is a '安全性' (Security) section with 'Jenkins专有用户数据库' (selected) and '允许用户注册' (Allow User Registration) checked. This section is highlighted with a red rectangle. Below it is a '授权策略' (Authorization Strategy) section with '登录用户可以做任何事' (Selected) and '匿名用户具有可读权限' (Anonymous users have read permission) checked. This section is also highlighted with a red rectangle. Other options like 'Gitlab Authentication Plugin', 'LDAP', 'Servlet容器代理', and 'Unix user/group database' are listed but not selected.

全局工具设置

JDK

JDK 安装

JDK
别名

JAVA_HOME

自动安装

系统下JDK安装列表

Git

Git installations

Name

Path to Git executable

自动安装

SonarQube Scanner

SonarQube Scanner 安装

SonarQube Scanner
Name

SONAR_RUNNER_HOME

自动安装

系统下SonarQube Scanner 安装列表

Ant

Ant 安装

Ant
Name

ANT_HOME

自动安装

系统下Ant 安装列表

Maven

Maven 安装

Maven
Name

MAVEN_HOME

自动安装

系统下Maven 安装列表

NodeJS

NodeJS 安装

NodeJS
别名

安装目录

自动安装

系统下NodeJS 安装列表

新建自由风格任务

添加回滚、发布选项

The screenshot shows the Jenkins configuration interface for adding a build step. The 'Status' parameter is selected, and its configuration is displayed. The 'Deploy' and 'Rollback' options are listed under the 'Options' section. The 'Description' field contains the text: 'Deploy : 【发布】' and 'Rollback : 【回滚】'. Below the parameter configuration, there is a preview section and a link to 'Remove Parameter'.

添加git地址，ansible为秘钥登录授权与gitlib

The screenshot shows the Jenkins configuration interface for a Git repository. The 'Repository URL' is set to 'ssh://git@192.168.20.131:10022/contract/paas/config-server.git'. The 'Credentials' dropdown is set to 'ansible', with a red arrow pointing to it. The 'Branches to build' section shows a branch specifier of '*dev'. The 'Source Code Browser' section is set to '(自动)'.

与gitlab设置web hook

Build when a change is pushed to GitLab. GitLab webhook URL: <http://192.168.20.15:8080/project/T-config-server>

Enabled GitLab triggers	Push Events <input checked="" type="checkbox"/>
	Opened Merge Request Events <input type="checkbox"/>
	Accepted Merge Request Events <input checked="" type="checkbox"/>
	Closed Merge Request Events <input type="checkbox"/>
	Rebuild open Merge Requests <input type="checkbox"/> Never
	Approved Merge Requests (EE-only) <input checked="" type="checkbox"/>
	Comments <input checked="" type="checkbox"/>
	Comment (regex) for triggering a build <input type="text"/> Jenkins please retry a build

Enable [ci-skip]

Ignore WIP Merge Requests

Set build description to build cause (e.g. Merge request or Git Push)

Build on successful pipeline events

Pending build name for pipeline

Cancel pending merge request builds on update

Allowed branches

- Allow all branches to trigger this job
- Filter branches by name
 - Include dev. Matching 1 branch.
 - Exclude
- Filter branches by regex
- Filter merge request by label bb0028900b3d7f79dad826e10b714ca5

保存 应用

gitlab设置webhook

The screenshot shows the GitLab Admin Area interface. The left sidebar is expanded, showing various sections like Admin Area, Monitoring, System Hooks, Applications, Abuse Reports, Deploy Keys, Service Templates, Labels, Appearance, Settings (which is currently selected), General, Integrations, Repository, CI/CD, Reporting, Metrics and profiling, Network (which is also selected), and Preferences. The main content area is titled "Admin Area > Network". It contains three main sections: "Performance optimization" (with a "Save changes" button), "User and IP Rate Limits" (with an "Expand" button), and "Outbound requests" (with a "Collapse" button). At the bottom right of the main content area, there is a small text box containing the IP address "192.168.21.149".

The screenshot shows the 'Integrations' page in GitLab. On the left sidebar, 'Integrations' is selected under 'Settings'. The main area displays configuration for a webhook named 'test'. The 'Push events' trigger is active, with the URL set to 'http://192.168.21.149:8080/project/test'. Other triggers like 'Tag push events', 'Comments', and 'Merge request events' are listed but inactive. SSL verification is enabled. At the bottom, there's a green 'Add webhook' button.

一切使用shell执行

sonar、mvn、ansible

```

# 执行 shell
命令
#source /etc/profile
export JAVA_HOME=/usr/local/jdk1.8.0_202
export JRE_HOME=/usr/local/jdk1.8.0_202/jre
export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
export PATH=$PATH:$JAVA_HOME/bin

export SONAR_HOME=/usr/local/sonarqube-7.7
export PATH=$PATH:$SONAR_HOME/bin/linux-x86-64

export SONAR_RUNNER_HOME=/usr/local/sonar-scanner-3.3.0.1492
export PATH=$PATH:$SONAR_RUNNER_HOME/bin
export SONAR_RUNNER_OPTS="-Xmx512m -XX:MaxPermSize=128m"

export MAVEN_HOME=/usr/local/apache-maven-3.6.0
export PATH=$PATH:$MAVEN_HOME/bin

DIR_Backup=/backup/Test
DIR_Upfile=/data/paas/Test

case $Status in
Deploy)
echo "Status:$Status"
cd ${DIR_Backup}
if [ -d ${JOB_NAME} ]
then
echo "The files is already exists"
else
mkdir ${DIR_Backup}/${JOB_NAME}
fi
cd ${JOB_NAME}
if [ -d ${BUILD_NUMBER} ]
then
echo "The files is already exists"
else
mkdir ${DIR_Backup}/${JOB_NAME}/${BUILD_NUMBER}
fi
cd ${WORKSPACE}
/usr/local/apache-maven-3.6.0/bin/wv sonar:sonar -Dsonar.projectKey=${JOB_NAME} -Dsonar.projectName=${JOB_NAME} -Dsonar.host.url=http://192.168.20.15:9000 -Dsonar.java.binaries=/usr/local/apache-maven-3.6.0/bin/mvn -B -f ${WORKSPACE}/sonar-project.properties clean install -Dmaven.test.skip=true -Ptest
/bin/cp ${WORKSPACE}/target/*.jar ${DIR_Upfile}/$[JOB_NAME]/server/
/bin/cp ${WORKSPACE}/target/*.jar ${DIR_Backup}/$[JOB_NAME]/$[BUILD_NUMBER]/
sudo su - ansible -c "cd /etc/ansible/playbook/commit && ./commit.sh ${JOB_NAME}"
echo "Completin!"
;;
Rollback)
echo "Status:$Status"
echo "Version:$Version"
cp ${DIR_Backup}/$[JOB_NAME]/*.jar ${DIR_Upfile}/$[JOB_NAME]/server/
sudo su - ansible -c "cd /etc/ansible/playbook/commit && ./commit.sh ${JOB_NAME}"
echo "Completin!"
;;
*)
exit
;;
esac

```

```

1 #source /etc/profile
2 export JAVA_HOME=/usr/local/jdk1.8.0_202
3 export JRE_HOME=/usr/local/jdk1.8.0_202/jre
4 export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
5 export PATH=$PATH:$JAVA_HOME/bin
6

```

```

7 export SONAR_HOME=/usr/local/sonarqube-7.7
8 export PATH=$PATH:$SONAR_HOME/bin/linux-x86-64
9
10 export SONAR_RUNNER_HOME=/usr/local/sonar-scanner-3.3.0.1492
11 export PATH=$PATH:$SONAR_RUNNER_HOME/bin
12 export SONAR_RUNNER_OPTS="-Xmx512m -XX:MaxPermSize=128m"
13
14 export MAVEN_HOME=/usr/local/apache-maven-3.6.0
15 export PATH=$PATH:$MAVEN_HOME/bin
16
17 DIR_Backup=/backup/Test
18 DIR_Upfile=/data/paas/Test
19
20 case $Status in
21 Deploy)
22 echo "Status:$Status"
23 cd ${DIR_Backup}
24 if [ -d ${JOB_NAME} ]
25 then
26 echo "The files is already exists"
27 else
28 mkdir ${DIR_Backup}/${JOB_NAME}
29 fi
30
31 cd ${JOB_NAME}
32 if [ -d ${BUILD_NUMBER} ]
33 then
34 echo "The files is already exists"
35 else
36 mkdir ${DIR_Backup}/${JOB_NAME}/${BUILD_NUMBER}
37 fi
38
39 cd ${WORKSPACE}
40 /usr/local/apache-maven-3.6.0/bin/mvn sonar:sonar -
Dsonar.projectKey=${JOB_NAME} -Dsonar.projectName=${JOB_NAME} -
Dsonar.host.url=http://192.168.20.15:9000 -Dsonar.java.binaries=. -Dsonar.login=652860bd890396ccb88f3f2fbbea017514bbd85a
41 /usr/local/apache-maven-3.6.0/bin/mvn -B -f ${WORKSPACE}/pom.xml clean
install -Dmaven.test.skip=true -Ptest
42 /bin/cp ${WORKSPACE}/target/*.jar ${DIR_Upfile}/${JOB_NAME}/server/
43 /bin/cp ${WORKSPACE}/target/*.jar ${DIR_Backup}/${JOB_NAME}/${BUILD_NUM-
BER}/

```

```

44 sudo su - ansible -c "cd /etc/ansible/playbook/commit && ./commit.sh
${JOB_NAME}"
45 echo "Completin!"
46 ;;
47 Rollback)
48 echo "Status:${Status}"
49 echo "Version:${Version}"
50 cp ${DIR_Backup}/${JOB_NAME}/${Version}/*.jar ${DIR_Upfile}/${JOB_NAM
E}/server/
51 sudo su - ansible -c "cd /etc/ansible/playbook/commit && ./commit.sh
${JOB_NAME}"
52 echo "Completin!"
53 ;;
54 *)
55 exit
56 ;;
57 esac

```

企业微信通知

微信配置通知	
工程名(有空格请用%20填充)	config-server(测试环境)
CorpId(企业ID)	ww01147c25dfe45977
Secret(应用秘钥)	eQxnVJJfTPF9xPV12gSWCKEC9i-fGpqxXgKP6AtLLM
推送部门(填写部门id)	0
AgentId(应用ID)	1000008
另外需要推送人员	BaoSong ZhangYanMing KongDeHai LuLiLij WuKaiYan WangQuan

cd /var/lib/jenkins/plugins, 更新下面包



newplugon.zip
4.3MB

```
[jenkins@jekins jenkins]$ git clone ssh://jenkins@192.168.21.145:29418/test && scp -p -P 29418 jenkins@192.168.21.145:hooks/commit-msg test/.git/hooks/
Cloning into 'test'...
The authenticity of host '[192.168.21.145]:29418 ([192.168.21.145]:29418)' can't be established.
ECDSA key fingerprint is SHA256:bNRdsW9BtMView3dde2pm9Pm9R1Pk3dq32wqEKJFSy1k.
ECDSA key fingerprint is MD5:72:a2:05:5c:5a:8a:1b:3a:a6:f4:5d:5d:ae:4c:d9:8f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[192.168.21.145]:29418' (ECDSA) to the list of known hosts.
remote: Counting objects: 16, done
remote: Finding sources: 100% (16/16)
Receiving objects: 100% (16/16), 1.50 Kib | 0 bytes/s, done.
Resolving deltas: 100% (3/3), done.
remote: Total 16 (delta 3), reused 16 (delta 3)
commit-msg
[jenkins@jekins jenkins]$ ls
test
[jenkins@jekins jenkins]$ ls
test
[jenkins@jekins jenkins]$ cd test/
[jenkins@jekins test]$ ls
README.md
[jenkins@jekins test]$ touch swht.txt
[jenkins@jekins test]$ ls
README.md swht.txt
[jenkins@jekins test]$ git add .
[jenkins@jekins test]$ git commit -m "add the test file"
[master bc0e803] add the test file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 swht.txt
[jenkins@jekins test]$ git push origin master:refs/for/master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 318 bytes | 0 bytes/s, done.
Total 2 (delta 0), reused 0 (delta 0)
remote: Processing changes: new: 1, refs: 1, done
remote:
remote: New Changes:
remote:   http://192.168.21.145:8080/1 add the test file
remote:
To ssh://jenkins@192.168.21.145:29418/test
 * [new branch]      master -> refs/for/master
[jenkins@jekins test]$ █
```

git clone ssh://jenkins@192.168.21.145:29418/test && scp -p -P 29418

jenkins@192.168.21.145:hooks/commit-msg test/.git/hooks/

cd test

touch swht.txt

git add .

git commit -m "add the test file"

git push origin master:refs/for/master

git push origin master:refs/for/master

ssh-keyscan -t rsa 192.168.21.149 >> known_hosts

ssh-keygen -H -f known_hosts

这句话最重要

```
1 sonar.projectKey=eureka-server
2 sonar.projectName=eureka-server
```

```
3 sonar.host.url=http://192.168.20.15:9000
4 sonar.login=4f66dcb3a1e61b288977a753bdd9af1c8160ae84
5 sonar.projectVersion=1.0-SNAPSHOT
6 sonar.sourceEncoding=UTF-8
7 sonar.modules=java-module
8
9 java-module.sonar.projectName=eureka-server
10 java-module.sonar.language=java
11 java-module.sonar.sources=./app/src
12 java-module.sonar.projectBaseDir=.
13 sonar.exclusions=**/*test*/
14 sonar.java.source=9
15 sonar.java.binaries=.
```

