Gerrit introduction

- Department: ES1100
- Author: Andy Cheng

Outline

- Introduction
- Gerrit
- Git
- SSH public key
- Gerrit Code Review
- Git UI
- Git Commit Message
- Reference

Introduction

Purpose of this document:

We’ll give you presentation about Gerrit and how we work with it.

Gerrit

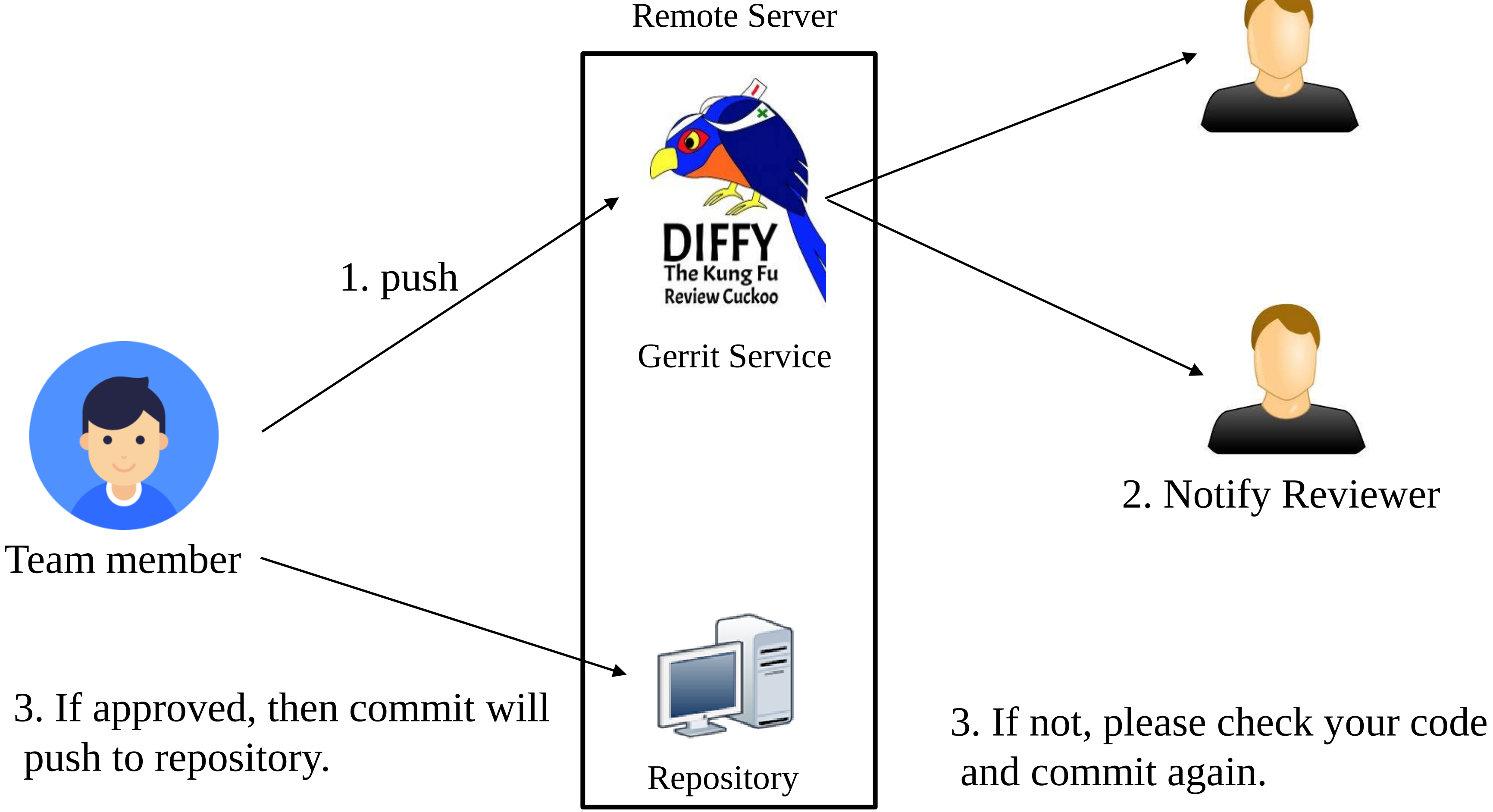
What is Gerrit ?

Gerrit is a web application based on git which is a tool for code review.

We work on local site and push commit to Gerrit, then it notifies all approvers to review your commit.

Gerrit

Gerrit work flow



Remote Server

Gerrit Service

Repository

Team member

1. push

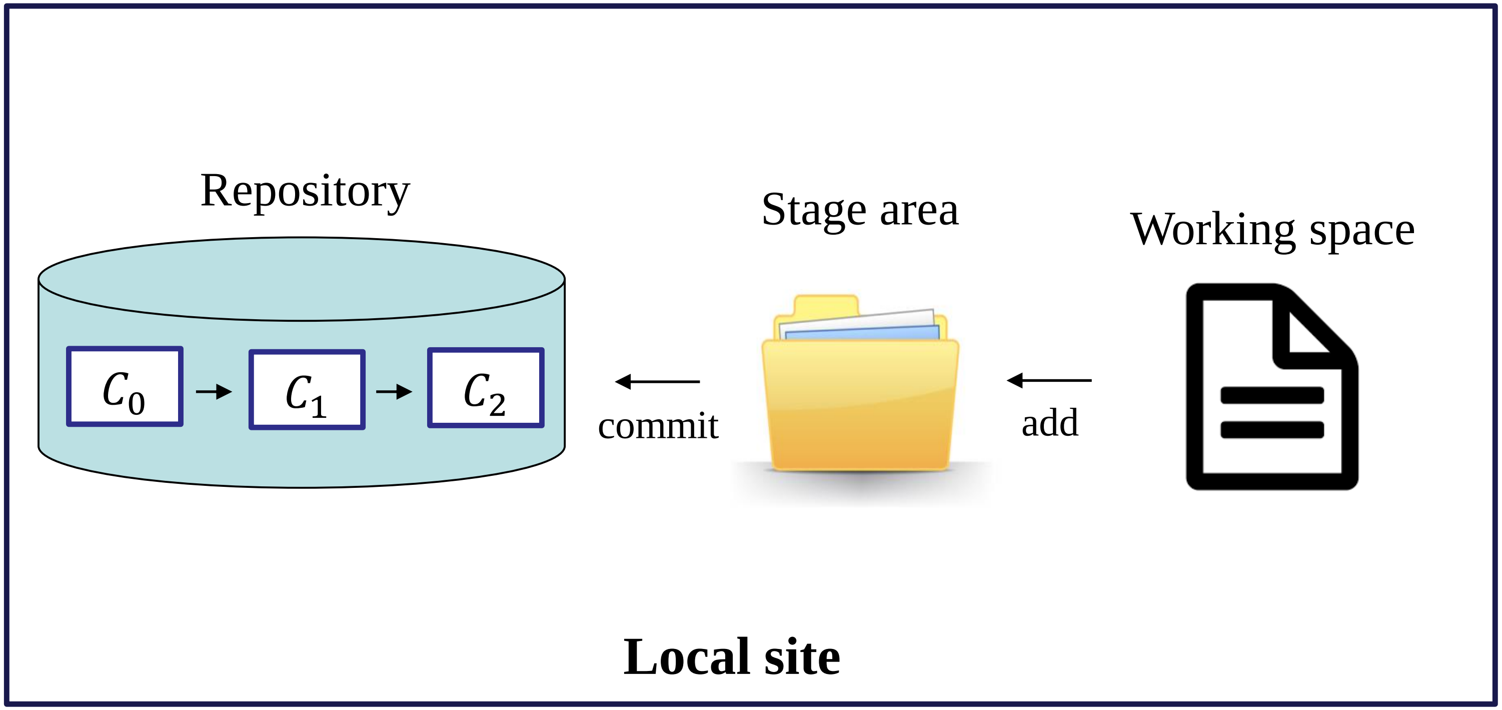
2. Notify Reviewer

3. If approved, then commit will push to repository.

3. If not, please check your code and commit again.

Git

Git Work Flow



Git

Basic Git Command

- **git clone**
  - git clone ssh://<UserID>@<ServerIP>:29418/<ProjectName>
- **git status**
  - git status: check the status of your git repo, it’s as same as “svn st”
- **git checkout**
  - git checkout <File>
  - git checkout <Branch>
- **git add**
  - git add -A : put all the new file you just add and file you modified to the stage area.
- **git commit**
  - git commit: commit your code to the local repository.
- **git push**
  - git push: push commit to remote repository.

Git

Basic Git Command

- **git stash**
  - git stash: temporarily store your code that doesn’t want to commit for now.
- **git pull**
  - git pull **:** this step is as same as “git fetch + git merge’. It will bring local branch up-to-date.
- **git fetch**
  - git fetch: It will fetch the update from remote server but doesn’t merge yet to local branch.

SSH public key

Generate public key

Before we start our code review, we need to provide our public key to Gerrit. This is because Gerrit need this authentication to push commit.

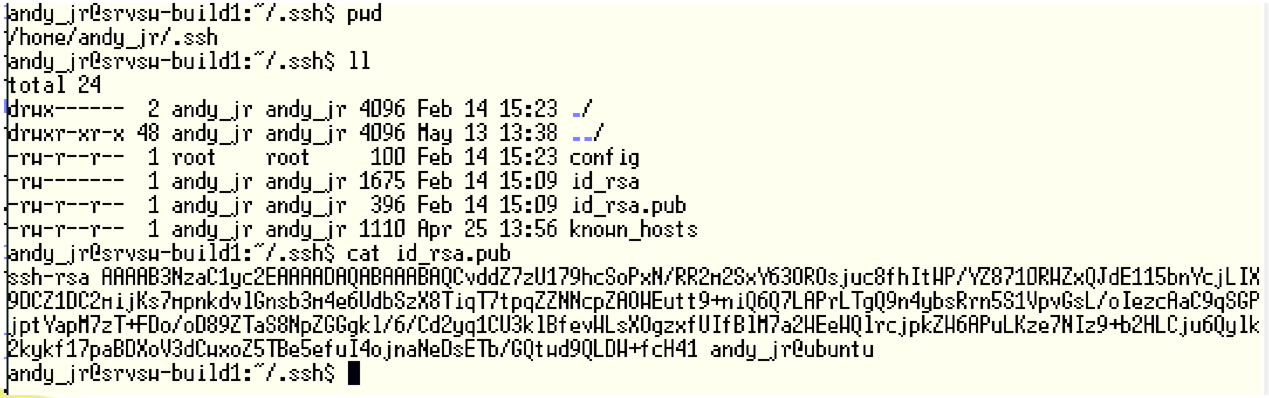
Step 1: generate ssh key on your host by command

- ssh-keygen -t rsa

Note: we use rsa as our algorithm.

Step 2: paste our public key to Gerrit site.

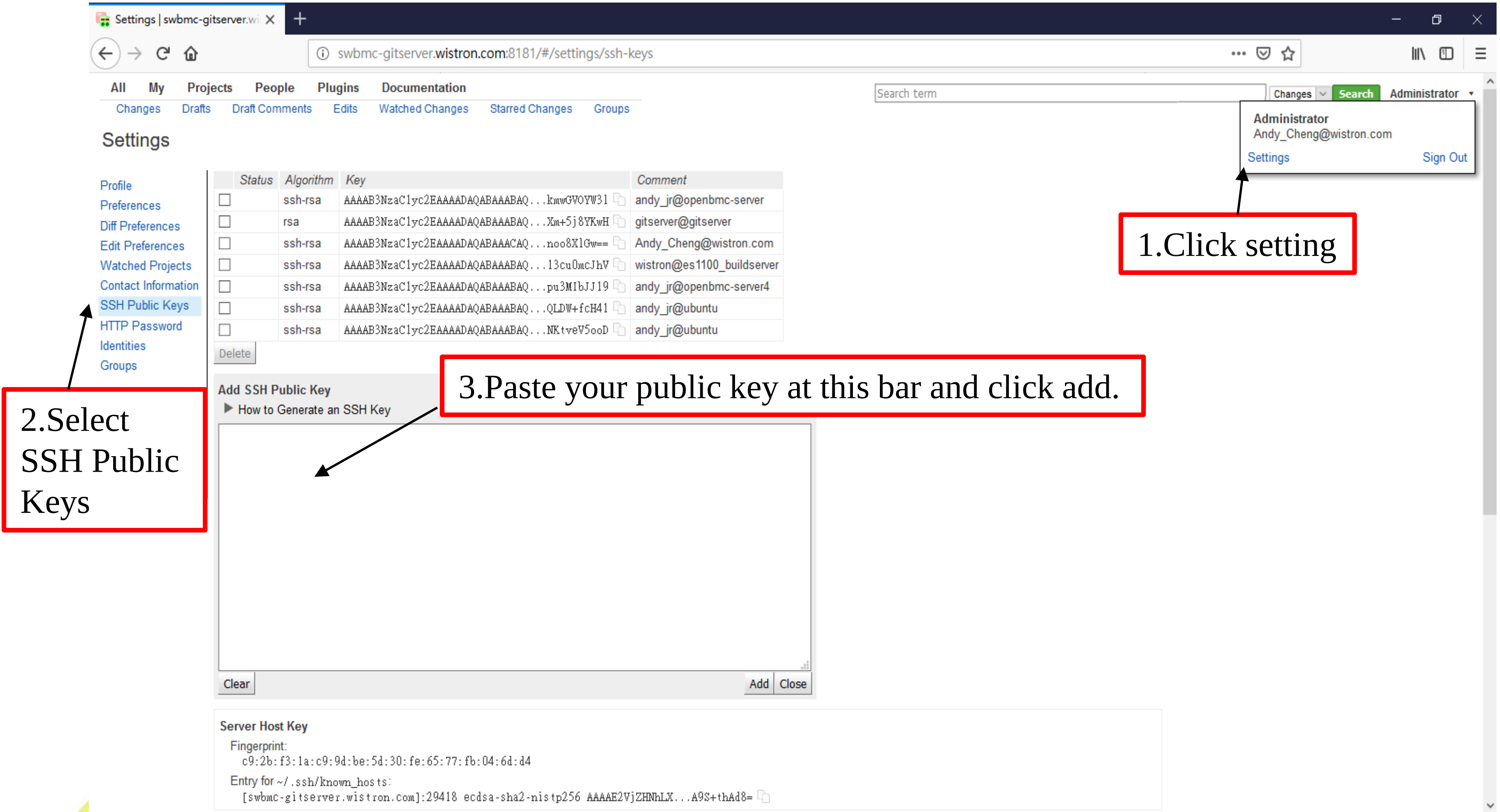
Note: Under your home directory, go to directory .ssh and it will include a pair of keys.



```
andy_jr@srvsw-build1:~/.ssh$ pwd
/home/andy_jr/.ssh
andy_jr@srvsw-build1:~/.ssh$ ll
total 24
drwx------  2 andy_jr andy_jr 4096 Feb 14 15:23 ./
drwxr-xr-x 48 andy_jr andy_jr 4096 May 13 13:38 ../
-rw-r--r--  1 root    root     100 Feb 14 15:23 config
-rw-------  1 andy_jr andy_jr 1675 Feb 14 15:09 id_rsa
-rw-r--r--  1 andy_jr andy_jr  396 Feb 14 15:09 id_rsa.pub
-rw-r--r--  1 andy_jr andy_jr 1110 Apr 25 13:56 known_hosts
andy_jr@srvsw-build1:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCvddZ7zU179hcSoPxN/RR2m2SxY630R0sjuc8fhItWP/YZ8710RWZxQJdE115bnYcjLIX
9DCZ1DC2mijKs7mpnkdvlGnsb3m4e6UdbSzX8TiqT7tpqZZNNcpZA0WEutt9+miQ6Q7LAPrLTgQ9n4ybsRrn5S1VpvGsL/oIezcAaC9qSGP
jptYapM7zT+FDo/oD89ZTaS8NpZGGgkl/6/Cd2yq1CU3klBfevWLsX0gzxfUIfBlM7a2WEeWQlrcjpkZW6APuLKze7NIz9+b2HLCju6Qylk
2kykf17paBDXoV3dCwxoZ5TBe5efuI4ojnaNeDsETb/GQtwd9QLDW+fcH41 andy_jr@ubuntu
andy_jr@srvsw-build1:~/.ssh$ █
```

SSH public key

Put public key on Gerrit



	Status	Algorithm	Key	Comment
□		ssh-rsa	AAAAB3NzaC1yc2EAAAADAQABAAABAQ...kmwGVOYW31	andy_jr@openbmc-server
□		rsa	AAAAB3NzaC1yc2EAAAADAQABAAABAQ...Xm+5j8YKwH	gitserver@gitserver
□		ssh-rsa	AAAAB3NzaC1yc2EAAAADAQABAAACAQ...noo8X1Gw==	Andy_Cheng@wistron.com
□		ssh-rsa	AAAAB3NzaC1yc2EAAAADAQABAAABAQ...13cu0mcJhV	wistron@es1100_buildserver
□		ssh-rsa	AAAAB3NzaC1yc2EAAAADAQABAAABAQ...pu3MIbJJ19	andy_jr@openbmc-server4
□		ssh-rsa	AAAAB3NzaC1yc2EAAAADAQABAAABAQ...QLDW+fcH41	andy_jr@ubuntu
□		ssh-rsa	AAAAB3NzaC1yc2EAAAADAQABAAABAQ...NKtveV5ooD	andy_jr@ubuntu

Server Host Key

Fingerprint:

```
c9:2b:f3:1a:c9:9d:be:5d:30:fe:65:77:fb:04:6d:d4
```

Entry for ~/.ssh/known_hosts:

```
[swbmc-gitserver.wistron.com]:29418 ecdsa-sha2-nistp256 AAAAE2VjZHNhLX...A9S+thAd8=
```

1.Click setting

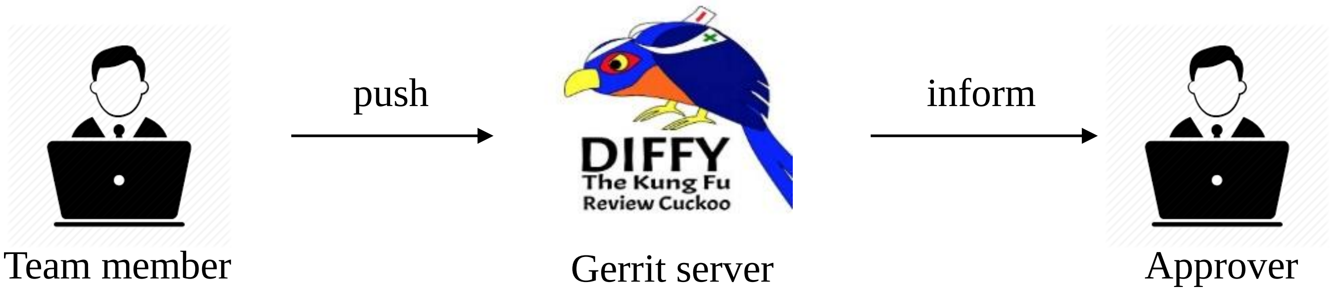
2.Select SSH Public Keys

3.Paste your public key at this bar and click add.

Gerrit Code Review

How to request a code review ?

Gerrit is an web application based on git. So when we want to request for a code review, we have to push to HEAD:refs/for/your-branch-name (e.g. refs/for/master) by Git push ssh://User_ID@gerrit.com:29418/project HEAD:refs/for/master. Then Gerrit will inform approver to check it.



Gerrit Code Review

Setup for code review.

Before getting start, we need to do following things first.

Step 1: Clone your project repository

```
git clone ssh://UserID@gerrit.com:29418/project.git
```

Step 2: Configure user.name and user.email

```
git config --global user.name andy
git config --global user.email Andy_Cheng@wistron.com
```

Step 3: Configure template of commit message.

```
git config --global commit.template ~/.gitmessage.txt
```

Gerrit Code Review

Setup for code review.

Step 4: Configure template of commit message.

Make sure you create a .gitmessage.txt like this.

```
Subject line (try to keep under 50 characters)

Multi-line description of commit,
feel free to be detailed.

[Ticket: X]
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   lib/test.rb
#
```

Gerrit Code Review

Setup for code review.

Step 5: Copy hook file from git repository

```
git rev-parse --git-dir
scp –p –P 29418 andy@gerrit.com:hooks/commit-msg .git/hooks
```

Every commit need an unique change-id which could automatically generate by hook file.

Gerrit Code Review

Procedure of code review

Step 1. git add -A

Step 2. git commit

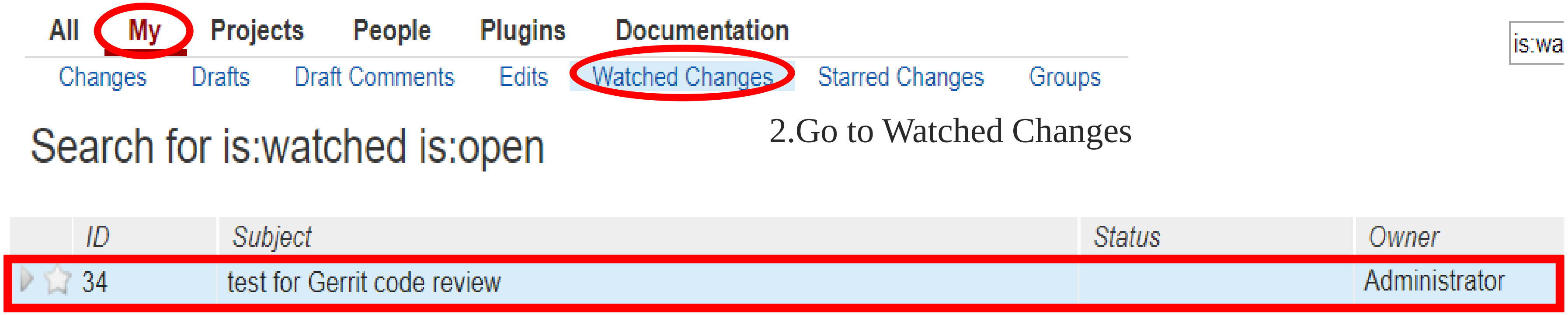
Step 3. git push ssh://User_ID@gerrit.com:29418/project HEAD:refs/for/master

Step 4. Then reviewer open up Gerrit for code review.

Gerrit Code Review

Procedure of code review

1.Select button My



All My Projects People Plugins Documentation

Changes Drafts Draft Comments Edits Watched Changes Starred Changes Groups

is:wa

Search for is:watched is:open

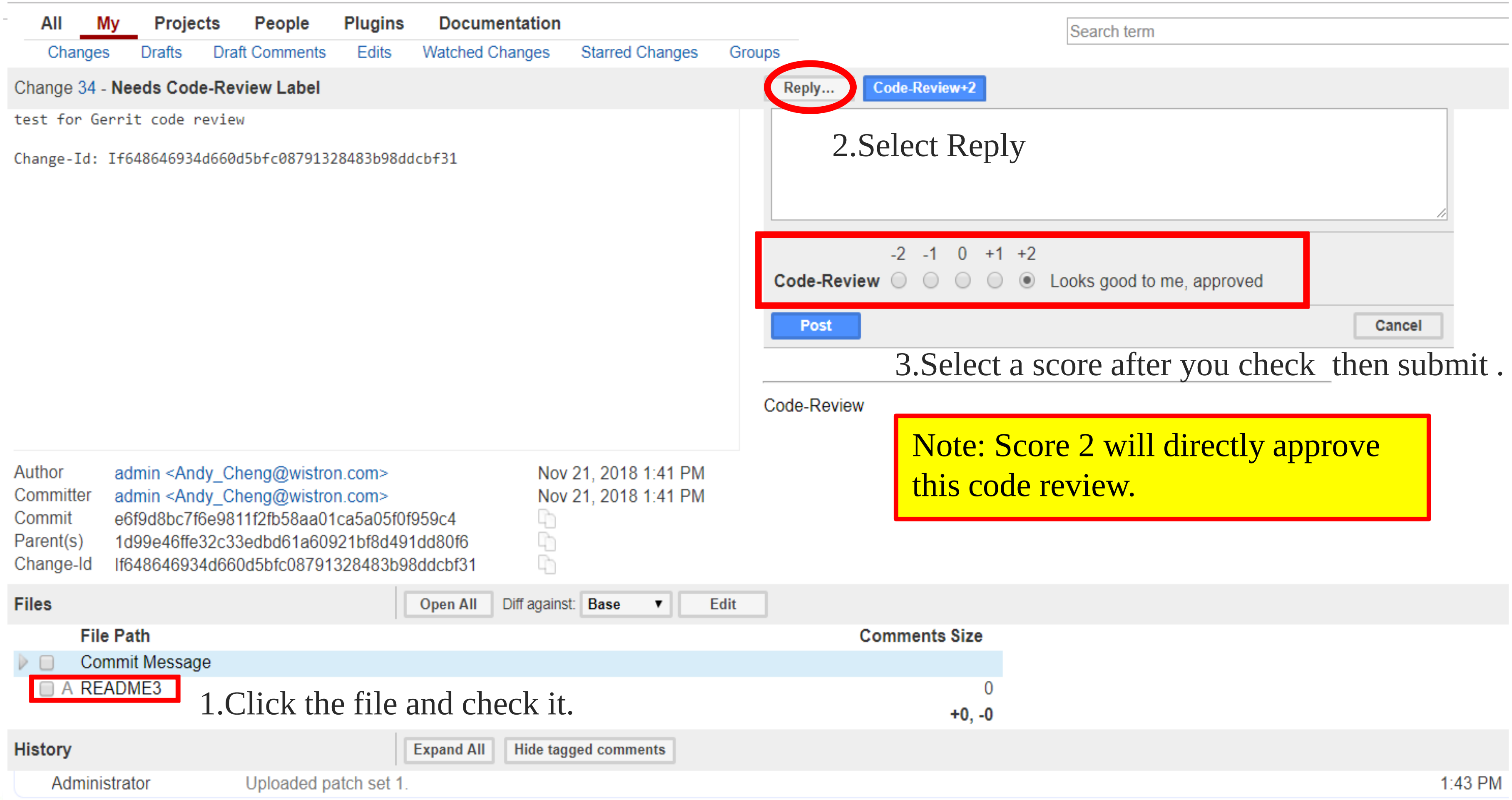
2.Go to Watched Changes

	ID	Subject	Status	Owner
	34	test for Gerrit code review		Administrator

3.Then you see a request for code review

Gerrit Code Review

Procedure of code review



All My Projects People Plugins Documentation

Changes Drafts Draft Comments Edits Watched Changes Starred Changes Groups

Search term

Change 34 - **Needs Code-Review Label**

Reply... Code-Review+2

```
test for Gerrit code review

Change-Id: If648646934d660d5bfc08791328483b98ddcbf31
```

2.Select Reply

	-2	-1	0	+1	+2	
Code-Review	○	○	○	○	●	Looks good to me, approved

Post Cancel

3.Select a score after you check then submit .

Code-Review

Note: Score 2 will directly approve this code review.

Author	admin <Andy_Cheng@wistron.com>	Nov 21, 2018 1:41 PM
Committer	admin <Andy_Cheng@wistron.com>	Nov 21, 2018 1:41 PM
Commit	e6f9d8bc7f6e9811f2fb58aa01ca5a05f0f959c4	
Parent(s)	1d99e46ffe32c33edbd61a60921bf8d491dd80f6	
Change-Id	If648646934d660d5bfc08791328483b98ddcbf31	

Files Open All Diff against: **Base** ▾ Edit

	File Path	Comments	Size
□	Commit Message		
□ A	README3		0
			+0, -0

1.Click the file and check it.

History Expand All Hide tagged comments

Administrator Uploaded patch set 1. 1:43 PM

Gerrit Code Review

What if we were rejected?

If our commit reject by approver, just do following step after you correct your mistake in local site.

Step 1. after you fix your error or anything else, just add to stage area again.

```
-  git add -A
```

Step 2. commit to repository by postfix “ –amend ” , it will change your latest commit in your repo.

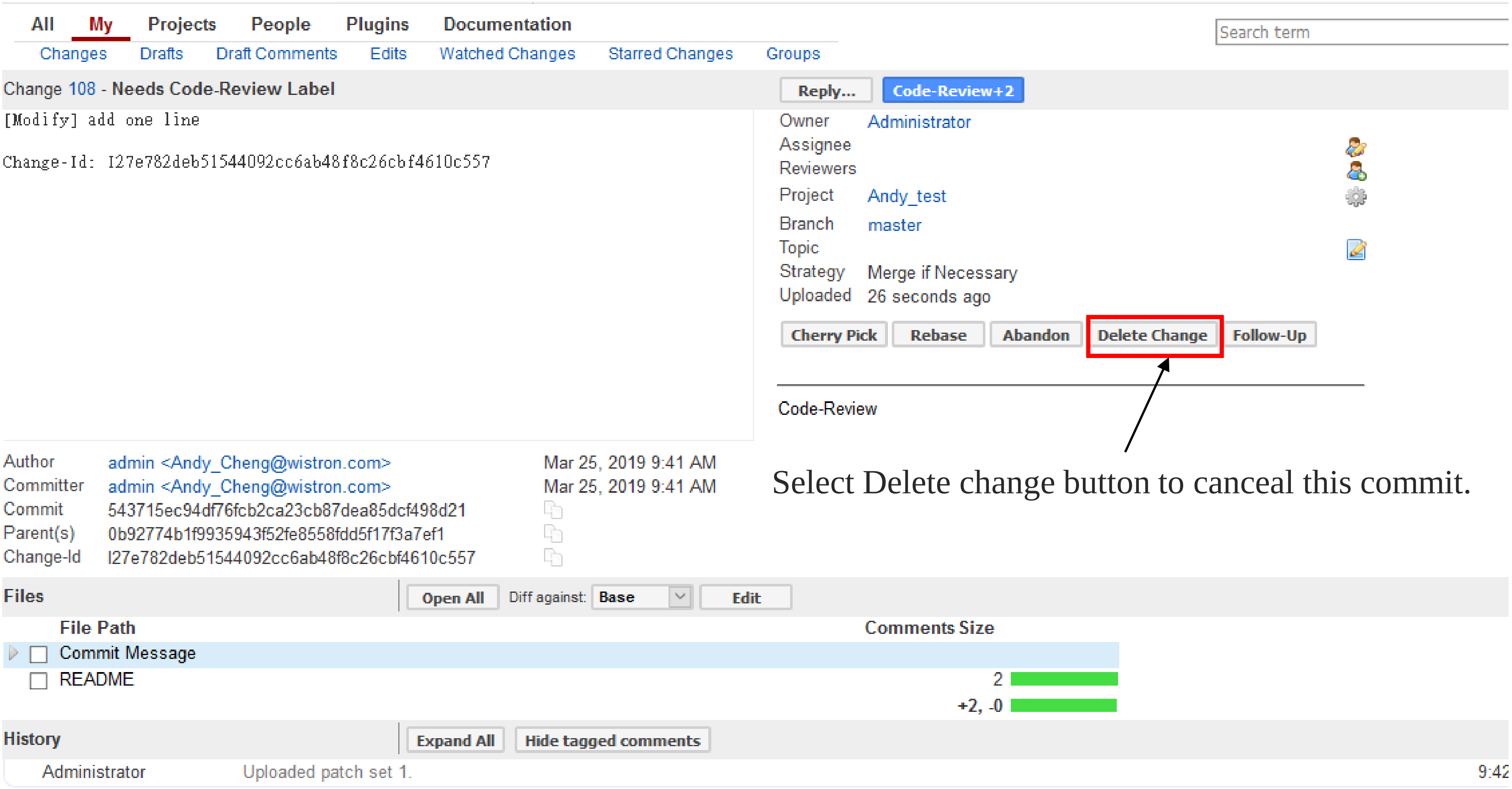
```
- git commit –amend
```

Step 3. Then push to Gerrit again.

```
- git push ssh://USER_ID@Gerrit.com:29418/PROJECT.git HEAD:refs/for/BRANCH.
```

Gerrit Code Review

What should approver do to reject commit ?



All My Projects People Plugins Documentation

Changes Drafts Draft Comments Edits Watched Changes Starred Changes Groups

Search term

Change 108 - **Needs Code-Review Label**

Reply... Code-Review+2

```
[Modify] add one line

Change-Id: I27e782deb51544092cc6ab48f8c26cbf4610c557
```

Owner	Administrator
Assignee	
Reviewers	
Project	Andy_test
Branch	master
Topic	
Strategy	Merge if Necessary
Uploaded	26 seconds ago

Cherry Pick Rebase Abandon Delete Change Follow-Up

Code-Review

Author	admin <Andy_Cheng@wistron.com>	Mar 25, 2019 9:41 AM
Committer	admin <Andy_Cheng@wistron.com>	Mar 25, 2019 9:41 AM
Commit	543715ec94df76fcb2ca23cb87dea85dcf498d21	
Parent(s)	0b92774b1f9935943f52fe8558fdd5f17f3a7ef1	
Change-Id	I27e782deb51544092cc6ab48f8c26cbf4610c557	

Files Open All Diff against: Base Edit

	File Path	Comments	Size
□	Commit Message		
□	README		2
			+2, -0

History Expand All Hide tagged comments

Administrator Uploaded patch set 1. 9:42

Select Delete change button to canceal this commit.

Git UI

- If you want to check history and every file on your git repository. Here are some tools about showing content by UI.

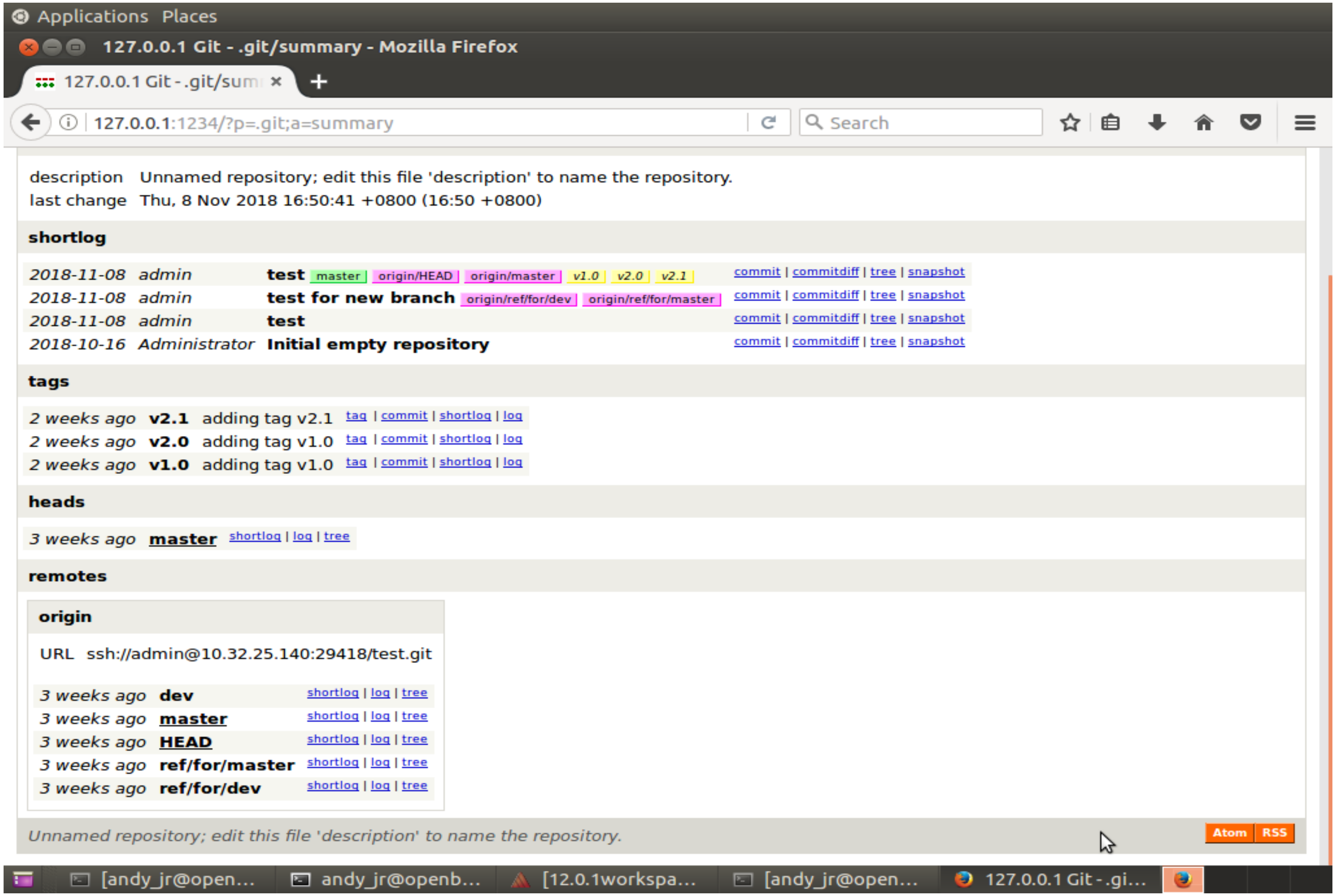
1.Using git command to generate httpd that listen port :1234

Step 1: git instaweb --httpd webrick

Step 2: Then open your browser and go to localhost:1234, it will show you all content of .git.

Git UI

Result after execution.



description	Unnamed repository; edit this file 'description' to name the repository.
last change	Thu, 8 Nov 2018 16:50:41 +0800 (16:50 +0800)

shortlog

2018-11-08	*admin*	**test** master origin/HEAD origin/master *v1.0* *v2.0* *v2.1*	commit | commitdiff | tree | snapshot
2018-11-08	*admin*	**test for new branch** origin/ref/for/dev origin/ref/for/master	commit | commitdiff | tree | snapshot
2018-11-08	*admin*	**test**	commit | commitdiff | tree | snapshot
2018-10-16	*Administrator*	**Initial empty repository**	commit | commitdiff | tree | snapshot

tags

2 weeks ago	**v2.1**	adding tag v2.1	tag | commit | shortlog | log
2 weeks ago	**v2.0**	adding tag v1.0	tag | commit | shortlog | log
2 weeks ago	**v1.0**	adding tag v1.0	tag | commit | shortlog | log

heads

3 weeks ago	**master**	shortlog | log | tree

remotes

origin

URL ssh://admin@10.32.25.140:29418/test.git

3 weeks ago	**dev**	shortlog | log | tree
3 weeks ago	**master**	shortlog | log | tree
3 weeks ago	**HEAD**	shortlog | log | tree
3 weeks ago	**ref/for/master**	shortlog | log | tree
3 weeks ago	**ref/for/dev**	shortlog | log | tree

Unnamed repository; edit this file 'description' to name the repository.

Atom RSS

Git UI

Note: If user want to view other git repository, please stop the deamon previously by following command.

Step 1: git instaweb --httpd webrick --stop

Step 2: go to project which you attempt to view.

Step 3: active the deamon by git instaweb --httpd webrick.

Git UI

2. Install gitg on your linux system.

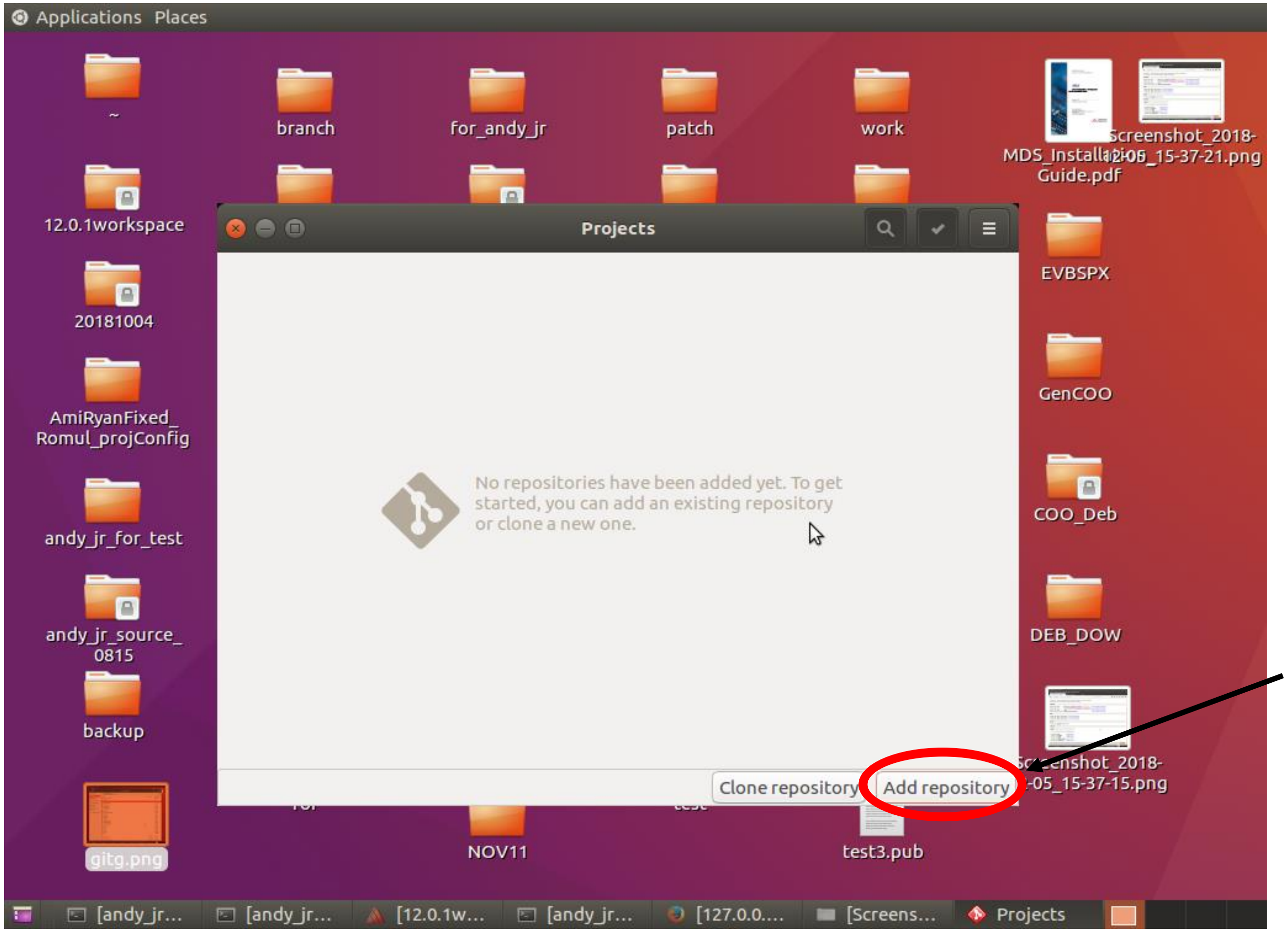
Step 1: download gitg from link.

Gitg: https://wiki.gnome.org/Apps/Gitg/

Step 2: Open gitg and add your .git repo.

Git UI

Start your gitg application.

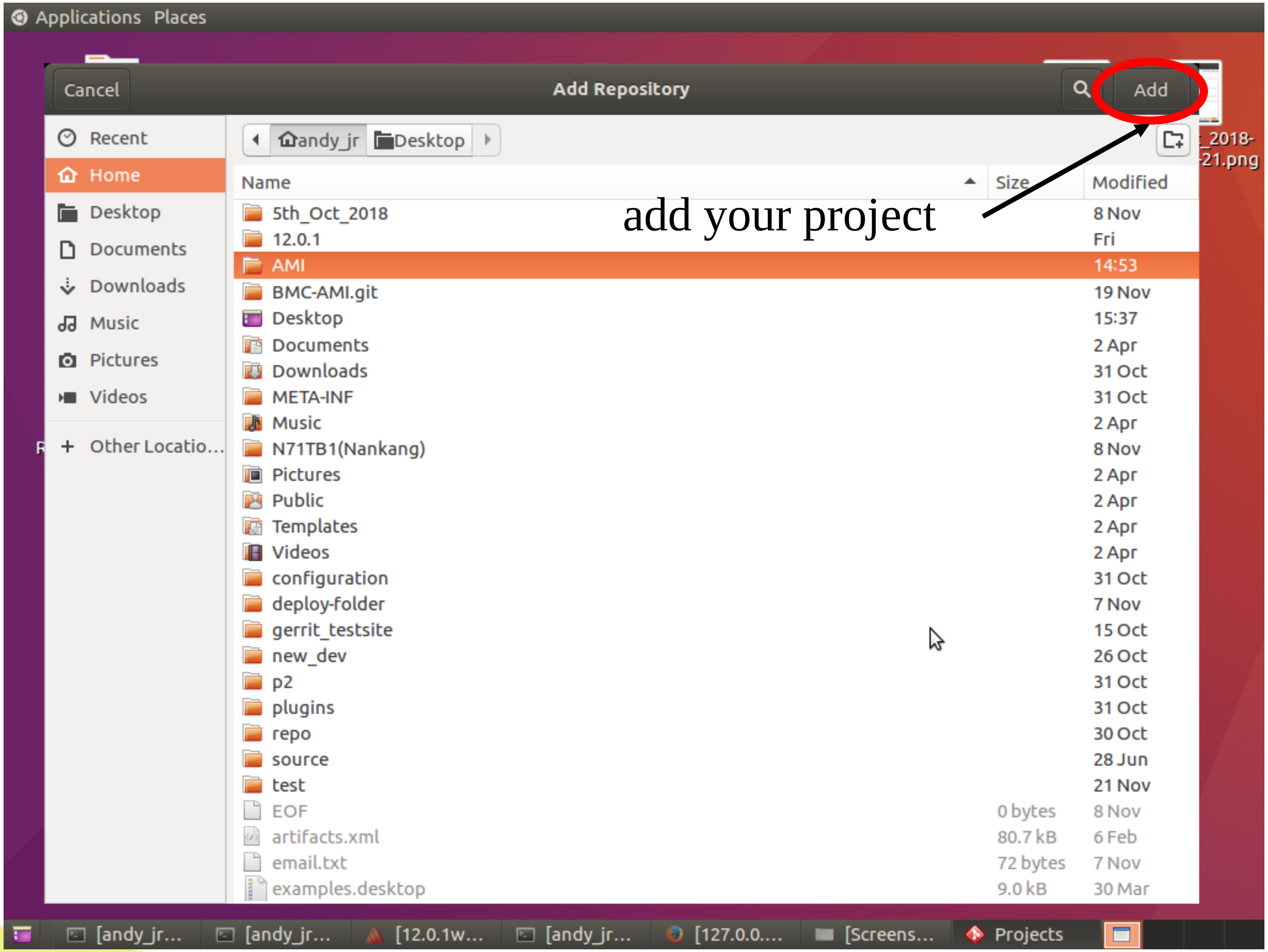


No repositories have been added yet. To get started, you can add an existing repository or clone a new one.

Click add repository.

Git UI

Select your git repository and click Add button.



add your project

Name	Size	Modified
5th_Oct_2018		8 Nov
12.0.1		Fri
AMI		14:53
BMC-AMI.git		19 Nov
Desktop		15:37
Documents		2 Apr
Downloads		31 Oct
META-INF		31 Oct
Music		2 Apr
N71TB1(Nankang)		8 Nov
Pictures		2 Apr
Public		2 Apr
Templates		2 Apr
Videos		2 Apr
configuration		31 Oct
deploy-folder		7 Nov
gerrit_testsite		15 Oct
new_dev		26 Oct
p2		31 Oct
plugins		31 Oct
repo		30 Oct
source		28 Jun
test		21 Nov
EOF	0 bytes	8 Nov
artifacts.xml	80.7 kB	6 Feb
email.txt	72 bytes	7 Nov
examples.desktop	9.0 kB	30 Mar

Git Commit Message

In order to let your commit clearly understand by every member, please follow the instruction below to formulate your commit message.

[Subject]:

with proper prefixed word at begin such as.

Feat: when you create new feature.

Fix: fix the bug.

Doc: upload some documents.

Style: reorganize your code.

Test: make test unit or benchmark for your code.

[Body]:

describe what and why you do this change in detail. Please make sure body is less than 72 characters.

Git Commit Message

Sample:

```
commit 42e769bdf4894310333942ffc5a15151222a87be
Author: andy<Andy_Cheng@wistron.com>
Date:   Fri Jan 01 00:00:00 1982 -02000

[Feat] Create fan control feature for Mihawk.

We create our oem fan control algorithm to make sure our motherboard won’t over heat.
```

Reference

- **Gerrit Code Review**
  - https://gerrit-review.googlesource.com/Documentation/index.html#_developer
- **Introduction for git**
  - https://git-scm.com/about
- **Gerrit/trouble shooting**
  - https://www.mediawiki.org/wiki/Gerrit/Troubleshooting
- **How to write a git commit message**
  - https://chris.beams.io/posts/git-commit/

Thanks for your participation