



FREE eBook

LEARNING

gerrit

Free unaffiliated eBook created from
Stack Overflow contributors.

#gerrit

Table of Contents

About	1
Chapter 1: Getting started with gerrit	2
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
Chapter 2: Gerrit Plugins	3
Remarks.....	3
Examples.....	3
Replication plugin.....	3
Events-log plugin.....	3
Chapter 3: Gerrit workflow	4
Examples.....	4
Merging straight to master.....	4
Use rebase instead of merge.....	4
Working on a feature branch.....	5
Chapter 4: Review a change on Gerrit	6
Remarks.....	6
Examples.....	6
Accept and submit changes.....	6
Comment changes.....	6
Credits	7

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [gerrit](#)

It is an unofficial and free gerrit ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official gerrit.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with gerrit

Remarks

This section provides an overview of what gerrit is, and why a developer might want to use it.

It should also mention any large subjects within gerrit, and link out to the related topics. Since the Documentation for gerrit is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Detailed instructions on getting gerrit set up or installed.

Read [Getting started with gerrit online](https://riptutorial.com/gerrit/topic/5715/getting-started-with-gerrit): <https://riptutorial.com/gerrit/topic/5715/getting-started-with-gerrit>

Chapter 2: Gerrit Plugins

Remarks

- Gerrit plugins are not provided compiled. Sources can be download from [Gerrit Plugin code site](#). Nevertheless, you can find **compiled version** on the [community Gerrit compiled plugins website](#).
- Install plugin : <https://gerrit-review.googlesource.com/Documentation/cmd-plugin-install.html>

Examples

Replication plugin

It is used to duplicate a git repository from gerrit to anyway. Configuration file is

```
$GERRIT_INSTALL/etc/replication.config.
```

Config file example to clone `MyRepo` from `gerrit` to `backupServer`

```
[remote "backup"]
  url = ProjectUrlOnBackupServer/${name} #Example backup.some.org:/pub/git/${name}.git
  push = +refs/heads/*:refs/heads/*
  push = +refs/tags/*:refs/tags/*
  projects = MyRepo
```

To reload the plugin, restart Gerrit server or execute following command if `install plugin` is installed:

```
ssh -p 29418 localhost gerrit plugin reload replication
```

Full documentation : [Official Replication Plugin Documentation](#)

Events-log plugin

This plugin is mandatory to make Gerrit able to receive message from other services. As an example, it has to be install to use `Sonar Gerrit` plugin

- Install `jar` file under `plugins` folder
- Default configuration is sufficient

Read Gerrit Plugins online: <https://riptutorial.com/gerrit/topic/7555/gerrit-plugins>

Chapter 3: Gerrit workflow

Examples

Merging straight to master

If you want to make a change that you want to merge with master, the best way is to first create a topic branch

```
git checkout -b foo
```

make a single commit with your feature

```
git commit -m "Made the thing X finally work"
```

and push that branch to review via

```
git push origin foo:refs/for/master/foo
```

Avoid working directly on master, as you will have problems resolving conflicts if someone else pushed to master before you. If you have your work on a separate branch, then resolving conflicts is as simple as

```
git checkout master
git pull origin master
git checkout foo
git rebase master
git push origin foo:refs/for/master/foo
```

Use rebase instead of merge

Because of how gerrit relies on change-ids, in order to resolve conflicts (pull changes to your topic branch) the best practice is to rebase topic branch onto master/other branch you want to push to. This way you preserve the change-id without having to amend the merge commit. For example if you submit foo to `refs/for/master` with history as follows:

```
  a < foo
 /
---s < master
```

Then gerrit will create a change with change-id taken from the commit `a`. But say after you passed review but before you submitted to master someone submitted their change:

```
  a < foo
 /
---s---x < master
```

If you rebase `foo` onto `master` you'll have

```
      a < foo
     /
---s---x < master
```

And you can easily push `a` again with the same change-id.

Working on a feature branch

As mentioned in the other example, you should use rebase instead of merge. But if you're working on a feature branch with your team then you'll run into the problem of pulling rewritten history. So the best way to work on a feature branch `foo` is to locally create tracking branch `foo` that you use only for pulling others' changes and create one more branch e.g. `dev_foo` that you use only for pushing. Then use the basic workflow as if `foo` was `master` and `dev_foo` your topic branch.

For example your history might look like this before submitting a change to review

```
      x < dev_foo
     /
a---b < foo
 /
---s---t---u < master
```

To push commit `x` to review and have it submitted to `foo` use

```
git push origin dev_foo:refs/for/foo/dev_foo
```

Read Gerrit workflow online: <https://riptutorial.com/gerrit/topic/6708/gerrit-workflow>

Chapter 4: Review a change on Gerrit

Remarks

Documentation sources : [Gerrit Review documentation](#)

Examples

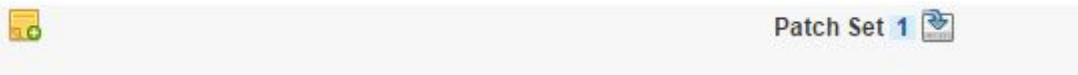
Accept and submit changes

Details to explain, with picture

Comment changes

During a review, user has the possibility to add comments to explain his point of view. There is several kind of comments

1. **Comment a file** : The yellow symbol allow user to add comment on entire file



2. **Line comment** : Click the line number to comment it
3. **Word selection** : The last option is to comment words by selection them and press 'c' keypad.

Shortcut

Press 'Esc' to cancel comment edit

Press 'Ctrl + s' to save draft comment

Read Review a change on Gerrit online: <https://riptutorial.com/gerrit/topic/6497/review-a-change-on-gerrit>

Credits

S. No	Chapters	Contributors
1	Getting started with gerrit	Community
2	Gerrit Plugins	Flows
3	Gerrit workflow	Dunno
4	Review a change on Gerrit	Flows