
GitLab Development Standards Documentation

Release 0.1

Peng Xiao

Nov 22, 2016

1	Table of Contents	3
1.1	Git Usage	3
1.2	GitLab Tutorial	3
1.3	Testing	6
1.4	Release Management	6
1.5	Continuous Integration	6
2	Support	7
3	Indices and tables	9

This guide is for small software development team which use GitLab ¹ as the only one tool to manage their daily work. If you are in a team like this, this documentation may help you and your team, no matter you are a project member or project owner.

GitLab provides Git repository management, code reviews, issue tracking, activity feeds and wikis. GitLab itself is also free software. GitLab also have integrated CI and CD to test, build and deploy your code.

Note: This guide is summarized from many many open sourced projects, and apply them to GitLab. Some of them maybe not so professional and perfect, but if you have no guide for your team at this moment, then this one will be a good one for you, if you have a better one, please give us some suggestions. Thanks.

¹ <https://about.gitlab.com/>

Table of Contents

1.1 Git Usage

1.1.1 Branch

1.1.2 Commit

1.1.3 Merge

1.2 GitLab Tutorial

GitLab is a open sourced software, and it has two versions: community version and enterprise version. gitlab.com is hosted by gitlab.org, and everyone can put their code on it. Or you can deploy gitlab in your own machine.

1.2.1 Administration

There is for GitLab Administrator. How to install/configure/upgrade/maintain GitLab. If you chose gitlab.com to host your source code, then you can ignore this section.

Install

Go to <https://about.gitlab.com/downloads/> and select operating system, follow the guide. After that, open the browser and type the address.

Configuration

Edit `/etc/gitlab/gitlab.rb` then run `gitlab-ctl reconfigure`

Upgrade

Update is very easy, if the operating system is CentOS.

```
$ sudo yum install gitlab-ce
```

Maintain

For backup: please use the scripts: <https://github.com/gitlabadm/auto-gitlab-backup>

1.2.2 Project

Project is a git repository, and you can create or import your project from popular Git services.

Create an empty project

Import from GitHub/Bitbucket

1.2.3 Group

A Group is a collection of projects and all the projects in the same group are under the same namespace. If you already (or will) have more than one related git repository, please group them together and treat them as a group.

More information about group like memeber management can get from the screenshot below.

New Group

Group path	<input type="text" value="https://as-gitlab.cisco.com/ open-source"/>
Description	<div><div></div><div></div></div>
Group avatar	<div><div>Choose File ...</div><div>File name...</div></div> <div>The maximum file size allowed is 200KB.</div>
Visibility Level ?	<div><div><input type="radio"/> Private</div><div>The group and its projects can only be viewed by members.</div></div> <div><div><input type="radio"/> Internal</div><div>The group and any internal projects can be viewed by any logged in user.</div></div> <div><div><input checked="" type="radio"/> Public</div><div>The group and any public projects can be viewed without any authentication.</div></div>
	<ul style="list-style-type: none">• A group is a collection of several projects• Members of a group may only view projects they have permission to access• Group project URLs are prefixed with the group namespace• Existing projects may be moved into a group
<div><div>Create group</div><div>Cancel</div></div>	

1.2.4 Issue

1.2.5 Merge Request

Branch should be merged to master when you have finished some work, for example, you have fixed an issue or implemented a feature. You can create *Merge Request* in GitLab, and assign it to somebody in your team. Actually, you also can create a merge request which is working in progress.

You can create a MR by yourself, or use the templates.

Templates

There are issue template and merge request template. How to create merge request template? Just like issue template, you should create a markdown file in your git repository. The location should be in `project_folder/.gitlab/merge_request_templates/xxx.md`, `xxx` will be the name of the template.

Here is what a template look like ¹.

```
$ more .gitlab/merge_request_templates/default.md
**1. What this MR does / why we need it:**

-
-

**2. Make sure that you've checked the boxes below before you submit MR:**

- [ ] I have read [Contribution guidelines](https://gitlab.com/test/demo/blob/master/
↪CONTRIBUTING.md)
- [ ] I have run `tox` locally and there is no error.
- [ ] no conflict with master branch.

**3. Which issue this PR fixes (optional)**

**4. CHANGELOG/Release Notes (optional)**

Thanks for your MR, you're awesome! :+1:
```

When you create a merge request and chose the template, it looks like this:

New Merge Request

From **test-mr** into **master** [Change branches](#)

Title

default

This is a test

Start the title with **WIP:** to prevent a **Work In Progress** merge request from being merged before it's ready.

Description

Write Preview

1. What this MR does / why we need it:

- Add docker-compose for auto deployment

2. Make sure that you've checked the boxes below before you submit MR:

☐ I have read [Contribution guidelines](#)
☐ I have run `tox` locally and there is no error.
 ☐ no conflict with master branch.

3. Which issue this PR fixes (optional)

fix #5

4. CHANGELOG/Release Notes (optional)

Thanks for your MR, you're awesome! 🍌

Assignee

Assignee

Milestone

Milestone


Labels

Labels

¹ https://github.com/avelino/awesome-go/blob/master/github/PULL_REQUEST_TEMPLATE.md

WIP

For the merge request which is working in the progress, please add *WIP:* in the beginning of the title. in that case, this merge request can not be merged unless you remove *WIP:* from the title.

 Pipeline #172 passed for 19cbc9cd. [View details](#)

Coverage 67.00%

This merge request is currently a Work In Progress
When this merge request is ready, [remove the WIP:](#) prefix from the title to allow it to be merged.

✓ Accepting this merge request will close issue #175.

Other Information

For *Assignee*, *Milestone*, *Labels*, this can be setted by the creator or the project manager.

How to Merge

The merge request can be merged by assignee when:

- The GitLab CI passed
- Code Review by a group of people, anyone can do code review.
- Check out the branch and do the functional test if needed.
- Other requirements

After Merge

There are some operations after merge:

- Remove source branch (optional)
- Check related issues and make sure close them
- Check master branch's build status
- Others

Reference

1.2.6 GitLab CI

1.3 Testing

1.4 Release Management

1.5 Continuous Integration

Support

There are many content need to be added. Please give us some suggestion through issues or PRs. Thanks.

Reference

Indices and tables

- `genindex`
- `modindex`
- `search`