

Research Computing

ARC Centre of Excellence for Particle Physics at the Terascale

GITLab

Part 2 – Advanced Usage

Lucien Boland

Sean Crosby

Goncalo Borges

Research Computing Team

- **GIT in a Nutshell**

- Basic concepts

- **GITLab in a Nutshell**

- Authentication
- Create a project in GITLab
- Commit to a project in GITLab
- Checkout from a project in GITLAB

- **Revisit slides and talk**

- <https://rc.coepp.org.au/presentations>
- <https://www.youtube.com/watch?v=-ZYFTuZ-GNE&feature=youtu.be>

- **Project Groups**
 - Project permissions
- **Branches and Merge Requests**
 - Protected Branches
- **Tags, Issues and Milestones**
- **Web Hooks**

- **Project Groups**

- ▶ Projects are normally created in GitLab under the user namespace
- ▶ GitLab Project groups is a way to group several projects under the same namespace.
 - Allow to give permissions to users to act on a group of projects on a single go.
- ▶ Already created projects can be migrated to a new GitLab group namespace, if required

- **Projects Groups** (Demo)
 - Create a project under your namespace (Reminder)
 - Create a project group
 - Create a project under the newly project group
 - Migrate a project under your personal namespace to the group namespace

- **Permissions**

- Users have different abilities depending on their access level
 - Permissions are based on ROLES
 - If a user has roles both in a project group and in the project itself, the highest permission level is used.
- GitLab Roles
 - Guest: Create issues and leave comments
 - Reporter: Download project
 - Developer: Add tags/branches; push to non-protected branches
 - Master: Push to protected branches, Add new team members

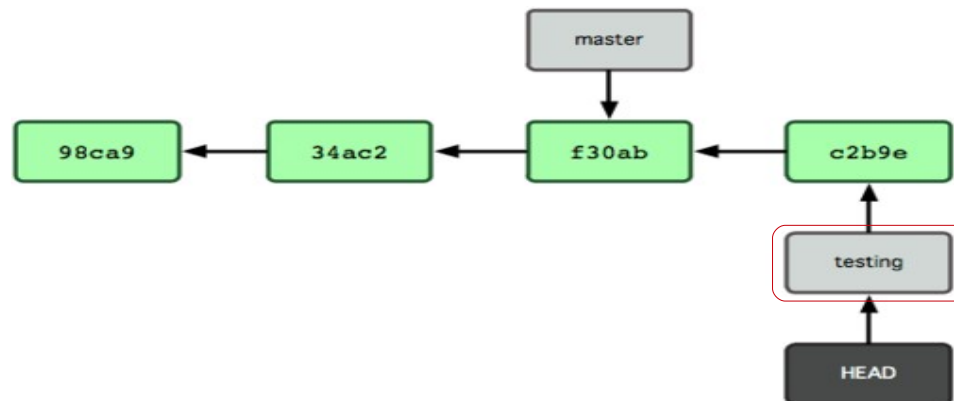
<https://gitlab.com/gitlab-org/gitlab-ce/blob/master/doc/permissions/permissions.md>

- **Permissions** (Demo)
 - Add users to a group
 - Attribute a default role to that user in the group
 - Change a specific user role for a specific project.

• Branches

- Is simply a lightweight movable pointer to specific commits.
 - As you initially make commits, you're given a master branch that points to the last commit you made
 - May mark development threads, done in parallel
 - You can create branches in GitLab (or via git command line)

- In GitLab there is the hability to protect brances users with roles bellow Master can not change



- **Branches** (Demo)
 - Create a branch (branch1) from the master
 - Commit a new file to branch1
 - Create a new branch (branch2) from branch1
 - Commit a new file to branch 2
 - Protect branch 2

- **Merge Requests**

- Eventually, the work developed in parallel may need to be inserted in another branch, or even in the master branch.
- GitLab provides a Merge Requests functionality
 - Merge Requests are a way to submit a piece of code from a branch of one repository to another branch of another or the same repository.
 - Merge Requests make sense in a collaborative environment where multiple authors work under the same code.
 - The author of the Merge Request is typically the author of the code, and submits the code for Review.
 - Depending on roles, different users can participate in the discussion

- **Merge Requests**(Demo)
 - Create a Merge Request to merge branch1 with master
 - Manage the Merge Request
 - Discuss it
 - Either accept it or reject it
 - Create a Merge Request to merge branch2 with master
 - Remember that branch2 is protected! You can not delete it.

- **Tags**

- Possibility to mark specific points in history as being important
- You can create tags in GitLab (or via git command line)

- **Issues**

- Possibility to track issues, problems to be fixed, proposals for enhancement and specific tasks in GitLab

- **Milestones**

- Possibility to define specific objectives to be accomplished at a given due date (Milestones)
 - It is possible to associate issues and merge requests to specific milestones

- **Tags, Issues and Milestone** (Demo)
 - Create a Tag in branch2
 - Create an issue
 - Create a Milestone
 - Associate issue with a Milestone

- **Web Hooks**

- Procedures you may want to execute triggered by
 - tag events
 - push events
 - issues events
 - merge requests events

- A bit less transparent than all the other functionalities
 - may require a little of json and ruby expertise
 - do not use it unless really necessary (keep it simple!)

- More information regarding Web Hooks:
https://gitlab.coepp.org.au/help/web_hooks/web_hooks

- **GitLab Community Edition**

- <http://doc.gitlab.com/ce/>

- **Questions / Doubts**

- rc@coepp.org.au