Short intro to Git and Github

Anton Alkin

anton.alkin@cern.ch

8th of November 2025, ALICE Tutorial preparation day

Outline

Useful links:

O2 Analysis tutorial MM channel, O2 Analysis MM channel
Git, Git Book, GitHub features
AliBuild, ALICE software installation, alidist recipes
ALICE O^2 coding guidelines, C++ reference

Introduction

What is this about?

- This is *an addition to* a step-by-step instruction
- The focus here are concepts, software components and their interconnections
- Collaborative code development
- Introduction to repositories, commits and pull requests

Get a copy of the slides to use the links to additional information

Git and GitHub

Git

- Decentralized Version Control System (VCS)
- Set of command-line[a] software tools
- Stores repository metadata in file system together with the tracked files
- Provides baseline for collaborative development
- Provides tools exporting and importing sets of commits

[a] 3rd party GUI apps exist

GitHub

- Centralized web-service that manages Git repositories
- Stores repositories on remote servers and provides access control
- Provides tools to enhance collaborative development
- Provides tools for setting up Continuous Integration (CI)

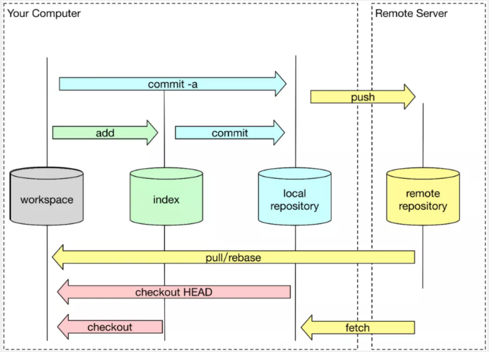
Git is not GitHub

Git

- Decentralized Version Control System (VCS)
- Set of command-line[a] software tools
- Stores repository metadata in file system together with the tracked files
- Provides baseline for collaborative development
- Provides tools exporting and importing sets of commits

[a] 3rd party GUI apps exist

Repositories



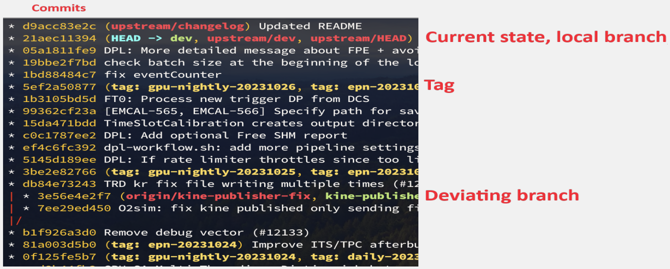
- Everyone has a copy of the repository - a fork
- Repositories can refer to other repositories – remotes
- You have your local copy and your GitHub fork
- Your local copy has the fork and the original repository as remotes
- Traditionally, the remote, pointing to the original repo, is called upstream and the remote pointing to the fork - origin

Commits, branches and history

Core notions

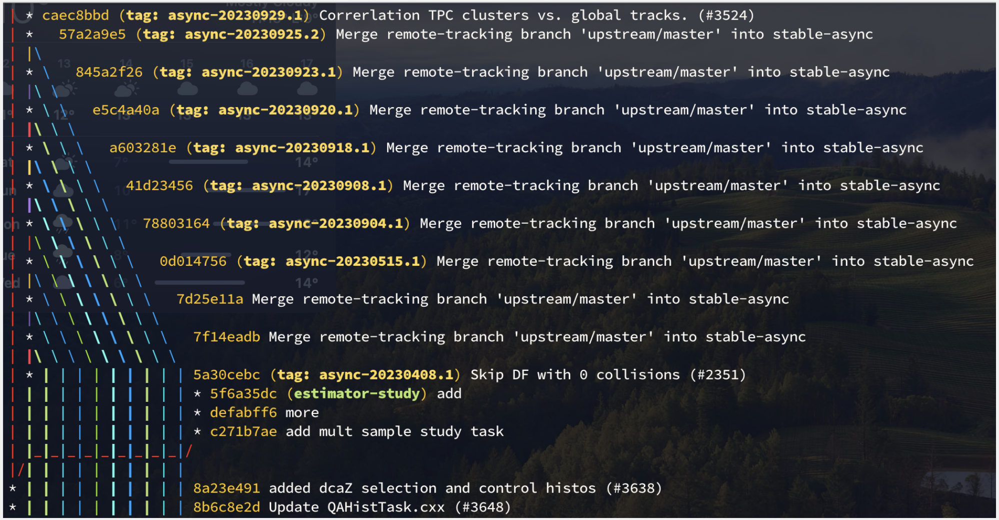
Commit	is a self-contained set of changes to the state of tracked files
Branch	is a sequence of commits, literally a branch of a repository tree
History	is a chronological sequence of commits, optionally arranged by branches
HEAD	is a pointer to the current state of the tracked files

Example of git history



```
* d9acc83e2c (upstream/changelog) Updated README
* 21aec11394 (HEAD -> dev, upstream/dev, upstream/HEAD)
* 05a1811fe9 DPL: More detailed message about FPE + avoi
* 19bbe2f7bd check batch size at the beginning of the lo
* 1bd88484c7 fix eventCounter
* 5ef2a50877 (tag: gpu-nightly-20231026, tag: epn-20231
* 1b3105bd5d FT0: Process new trigger DP from DCS
* 99362cf23a [EMCAL-565, EMCAL-566] Specify path for sav
* 15da471bdd TimeSlotCalibration creates output director
* c0c1787ee2 DPL: Add optional Free SHM report
* ef4c6fc392 dpl-workflow.sh: add more pipeline settings
* 5145d189ee DPL: If rate limiter throttles since too li
* 3be2e82766 (tag: gpu-nightly-20231025, tag: epn-20231
* db84e73243 TRD kr fix file writing multiple times (#12
| * 3e56e4e2f7 (origin/kine-publisher-fix, kine-publishe
| * 7ee29ed450 O2sim: fix kine published only sending fi
|/
* b1f926a3d0 Remove debug vector (#12133)
* 81a003d5b0 (tag: epn-20231024) Improve ITS/TPC afterbu
* 0f125fe5b7 (tag: gpu-nightly-20231024, tag: daily-2023
```

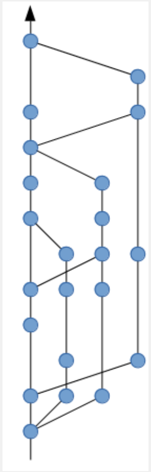
Merge and rebase



```
| * caec8bbd (tag: async-20230929.1) Correrlation TPC clusters vs. global tracks. (#3524)
| *   57a2a9e5 (tag: async-20230925.2) Merge remote-tracking branch 'upstream/master' into stable-async
| |\
| * \   845a2f26 (tag: async-20230923.1) Merge remote-tracking branch 'upstream/master' into stable-async
| |\ \
| * \ \   e5c4a40a (tag: async-20230920.1) Merge remote-tracking branch 'upstream/master' into stable-async
| |\ \ \
| * \ \ \   a603281e (tag: async-20230918.1) Merge remote-tracking branch 'upstream/master' into stable-async
| |\ \ \ \
| * \ \ \ \   41d23456 (tag: async-20230908.1) Merge remote-tracking branch 'upstream/master' into stable-async
| |\ \ \ \ \
| * \ \ \ \ \   78803164 (tag: async-20230904.1) Merge remote-tracking branch 'upstream/master' into stable-async
| |\ \ \ \ \ \
| * \ \ \ \ \ \   0d014756 (tag: async-20230515.1) Merge remote-tracking branch 'upstream/master' into stable-async
| |\ \ \ \ \ \ \
| * \ \ \ \ \ \ \   7d25e11a Merge remote-tracking branch 'upstream/master' into stable-async
| |\ \ \ \ \ \ \ \
| * \ \ \ \ \ \ \ \   7f14eadb Merge remote-tracking branch 'upstream/master' into stable-async
| |\ \ \ \ \ \ \ \ \
| * | | | | | | | | | 5a30cebc (tag: async-20230408.1) Skip DF with 0 collisions (#2351)
| | | | | | | | | | * 5f6a35dc (estimator-study) add
| | | | | | | | | | * defabff6 more
| | | | | | | | | | * c271b7ae add mult sample study task
| |_|_|_|_|_|_|_|_|/
|/| | | | | | | | |
* | | | | | | | | | 8a23e491 added dcaZ selection and control histos (#3638)
* | | | | | | | | | 8b6c8e2d Update QAHistTask.cxx (#3648)
```

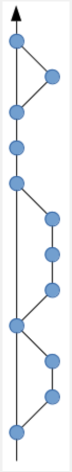
`git log --graph --decorate --oneline --abbrev-commit --all`

Merge and rebase (cont.)



Non-linear history

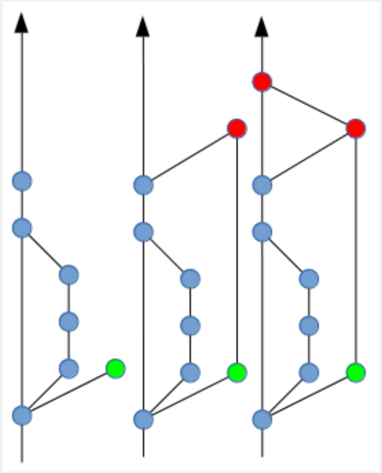
- Histories of several parallel developments in separate repositories are preserved
- Combined history in complicated, reverting breaking change may be not simple
- Distributed projects with several forks being actively used



Linear history

- Histories of non-main copies are ignored
- Combined history is easy to read, breaking changes can be reverted in a straightforward way
- Centralized projects, with a single main source

Merge and rebase (cont.)

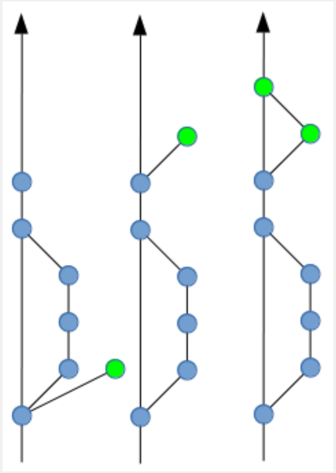


←Merging

commits from the tree being merged are re-created on top of the current head

a merge commit is created, the original commits from the tree being merged are unchanged

Rebasing →



GitHub is not Git

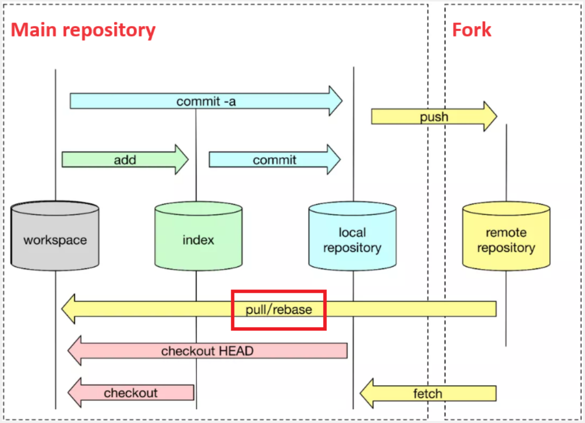
GitHub

- Centralized web-service that manages Git repositories
- Stores repositories on remote servers and provides access control
- Provides tools to enhance collaborative development
- Provides tools for setting up Continuous Integration (CI)

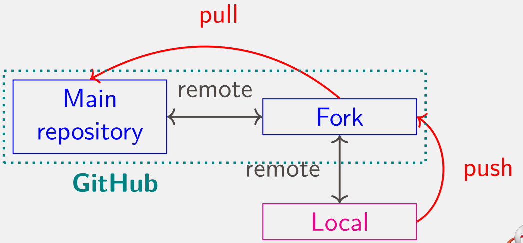
Typical workflow

- Fork the repository into which you are planning to contribute code
- Clone the repository locally, set the remotes remote (see Github access tokens)
- Regularly update your local copy from upstream
- Create a branch, add code, push the branch to your fork
- Test thoroughly and update your branch
- Create a PR

Pull Requests



- Your forked copy of the repository is a remote for the main one
- Main repository can pull the commits from a branch in your repository
- GitHub provides a convenient interface to request, test and confirm the pull



Best practice

Official commands cheat-sheet

Keep an eye on the repository state

- `git status` will display the summary
- You can add Git info to your shell prompt
- Or you can use one of Git GUI clients

Don't do work on the main branch

- Before you start coding - synchronize your local repo with upstream
- Create a branch for your changes and switch to it:
  `git checkout -b new-branch-name`[a]
- Avoid committing directly into **master** branch - it is supposed to be synchronized with upstream

[a]you can do that even after you started changing files

Continuous Integration

Definition

Continuous integration is the practice of automating the integration of code changes from multiple contributors into a single software project

O2Physics CI checks

Formatting:	Adherence of the new code to ALICE coding guidelines is verified in addition to some other formatting requirements.
Linter:	Various checks are performed to ensure code readability, standard-compliance and to spot simple potential bugs.
Compilation:	The software project is built from scratch including the changes that are being tested[a].

[a]Please, test the compilation yourself before creating a PR, the less build checks are run in CI, the faster the PRs are being merged

Summary

Recap of the main points

- Git is a VCS, GitHub is a service that uses Git as a backend
- It is your responsibility to manage local repository and the fork of the main repo - keep the main branch clean and up-to-date, develop in separate branches
- Test your code locally before creating a PR

Questions?