FairCoop and its DevOps Tools

Up to Code

Open source has become mandatory in FairCoop; actually, a general check of all tools developed/used within the ecosystem is being performed, taking care of having all projects in the appropriate environment.

A git-based devops system is or will be adopted not only with software but also with projects, texts, etc. with the aim to synchronize all different contributions.

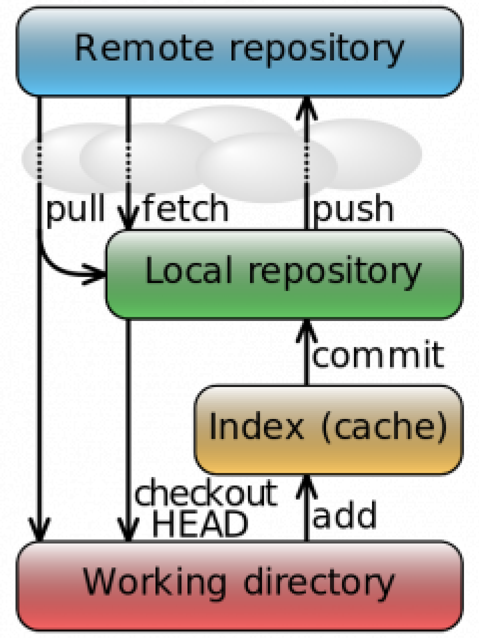
The decision on which between widespread github/gitlab and/or alternative git for self-hosting like Gogs and Gitea is left to devs/operators, besides a dedicated gitlab service is hosted by Fairkom, whereFairCoop group and FairCoin group operate .

A git git world

- Git was invented by Linus Torwald, who claimed he loved to give his name to everything he created (git means idiot in Eglish) for source code management (SCM), with an emphasis on speed.
- Git is a distributed non-linear workflow which allows Version Control System (VCS), Snapshot, Branches, Tags, Merges, Staging, Diff

Git workflow

(women at work! Be patient, Rome wasn't built in a day ;)



If branching in git is “just fast” (a local copy with all the code is created) the really great feature is merging, allowing easy merge between branches and dealing with merge conflicts.

How to contribute

- For **a pull request** upstream need to be forked, so a new branch needs to be created, then follow those steps:
- Create a local clone of your fork with *git clone*
- Create a local branch for your changes
- Make your changes and commit them to your local branch with *git commit*, ensuring to include a descriptive commit message
- Push the branch to your GitHub fork using *git push*
- Go to the page for the upstream repository go to the pull requests tab
- Click the “New Pull Request” Button
- Select the branch you want to submit, and write a summary of what your change explaining what it is intended to do and how it is implemented

Applying a Pull Request

Once a *pull request* has been submitted by a contributor the project maintainers evaluates the proposal and, if appropriate, merge it into the upstream repository. Usually common steps are:

- Evaluate the value of the changes
- Check out the changed code and run any test suites against it
- Pull the changes into the upstream code

a Toy Story

An attempt to apply a three environments methodology took place with FC Grav based website, which was created in developed/tested/released.

Even if http://fair.coop can't be considered a stable release (still many fixes need to be performed, especially with the blog) it was a nice attempt and we still believe in this methodology, perhaps for different scale of projects (such as a complete SO).

A test/production environment is possibly broad enough for dev purpose and it's the one mainly adopted within hackers' community.