Continuous Integration

What, Why, How, When, Who

What

- Development methodology
- Continuous integration of new commits/branches to the main code base
- Automated builds
- Automated testing

Why

- Frequent merges reduce the number and the complexity of conflicts
- Developers are up to date with latest changes
- Bad code is spotted early on
  - coding standards
  - static analysis
- Stability (unit testing, code coverage, integration tests, etc.)
- Lower entry level for new developers
- Automation
  - Parallelization of development and testing
  - Effortless multi-platform testing

How

- Jenkins (FOSS)
- TravisCI
- CircleCI
- TeamCity
- Bamboo
- Gitlab CI
- ...

https://en.wikipedia.org/wiki/Comparison_of_continuous_integration_software

When

- Before a commit
- Before a push
- After a push
- At a rebase (this is essentially the same as the before commit)
- On pull requests
- Based on a schedule (in theory this is not part of the CI, but who cares...)
- ...

Who

The tools!!!

DOs

- Create new self-contained branches for each change
- Commit and push often
- Setup notifications for failed integrations
- Setup git hooks to fail early
- Fix issues on the spot
- Create and keep artifacts

DEMO