NOLSATU

GitLab Administration

GL-ADM

Keyword

GitLab, GitLab CI/CD,

Runner, Auto DevOps, Docker, Pipeline

References

GitLab Documentation: https://docs.gitlab.com/

GitLab

NOLSATU

GitLab

web-based Git-repository manager providing wiki, issue-tracking and CI/CD pipeline features, using an open-source license, developed by GitLab Inc. The software was created by Dmitriy Zaporozhets and Valery Sizov

NOLSATU

Why GitLab ?

GitLab is a single application for the entire DevOps lifecycle that allows teams to work together better and bring more value to your customers, faster.

GitLab does this by shortening your devops cycle time, bridging silos and stages, and taking work out of your hands. It also means a united workflow that reduces friction from traditionally separate activities like application security testing.

NOLSATU

Company use GitLab

NASA

SONY

redhat

Alibaba.com™

SPACEX

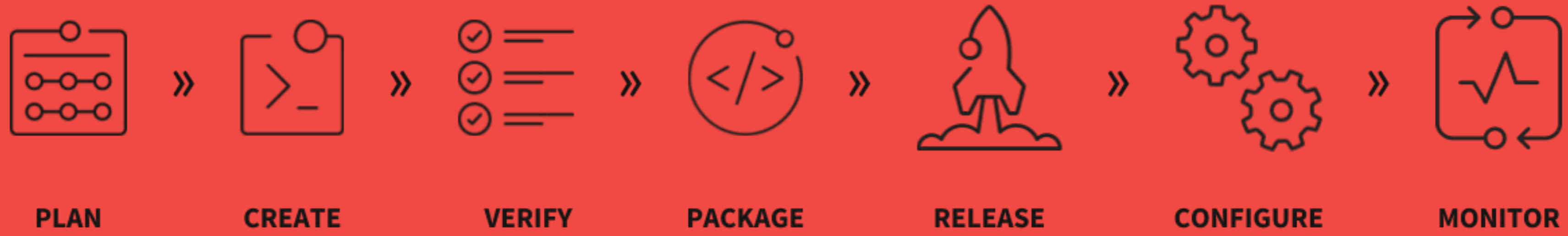
IBM

JUNIPER® NETWORKS

U.S. AIR FORCE

ERICSSON

Cycle Time GitLab



NOLSATU

Create Simple GIT Repository

To create a repository, create a directory for the project if it doesn’t exist, enter it, and run the command **git init**. The directory does not need to be empty.

This will create a **.git** directory in the **[project]** directory.

To create a commit, you need to do two things:

1. Tell Git which files to include in the commit, with **git add**. If a file has not changed since the previous commit (the “parent” commit), Git will automatically include it in the commit you are about to perform. Thus, you only need to add files that you have newly created or modified. Note that it adds directories recursively, so **git add** . will add everything that has changed.
2. Call **git commit** to create the commit object. The new commit object will have the current *HEAD* as its parent (and then, after the commit is complete, *HEAD* will point to the new commit object).

NOLSATU

GitLab Runner

GitLab Runner is the open source project that is used to run your jobs and send the results back to GitLab. It is used in conjunction with GitLab CI, the open-source continuous integration service included with GitLab that coordinates the jobs.

NOLSATU

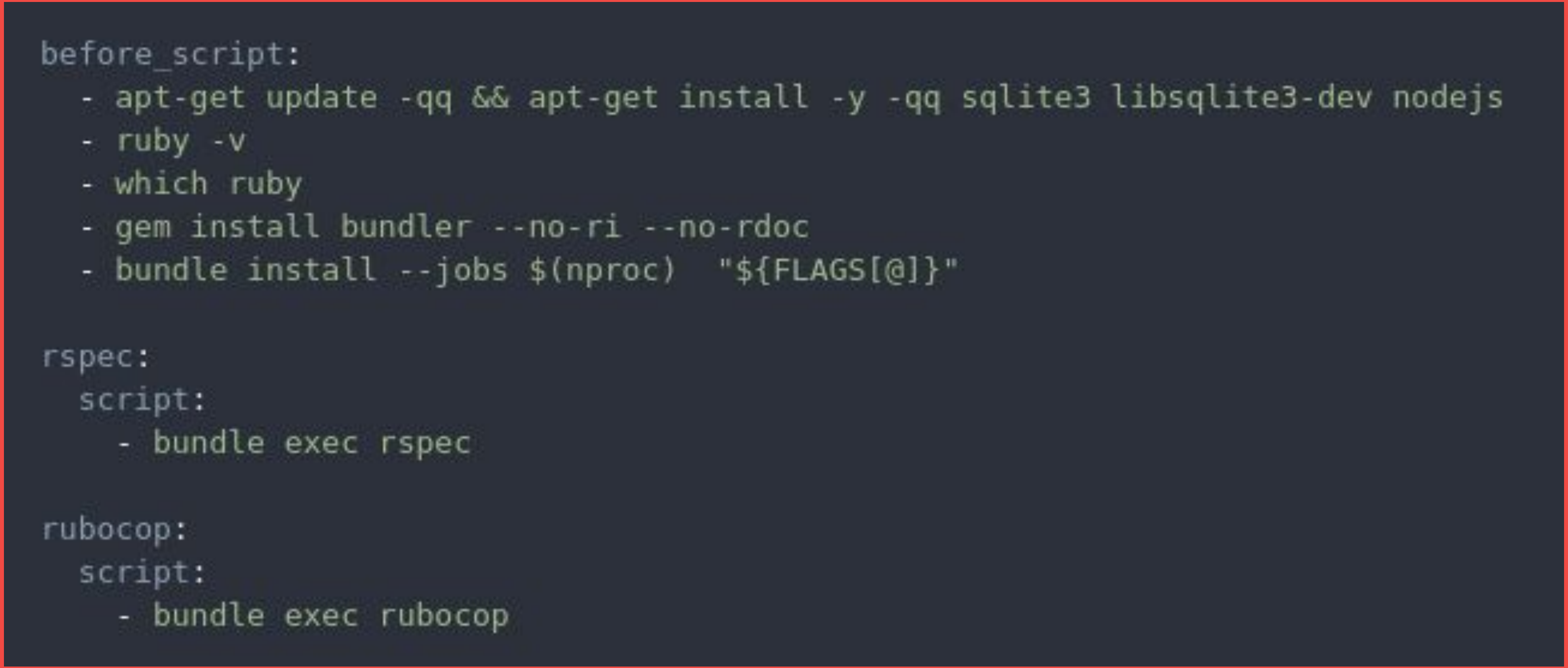
GitLab Runner

Steps needed to have a working CI can be summed up to:

1. Add .gitlab-ci.yml to the root directory of your repository
2. Configure a Runner

NOLSATU

Example .gitlab-ci.yml



```yaml
before_script:
  - apt-get update -qq && apt-get install -y -qq sqlite3 libsqlite3-dev nodejs
  - ruby -v
  - which ruby
  - gem install bundler --no-ri --no-rdoc
  - bundle install --jobs $(nproc)  "${FLAGS[@]}"

rspec:
  script:
    - bundle exec rspec

rubocop:
  script:
    - bundle exec rubocop
```

NOLSATU

Push .gitlab-ci.yml

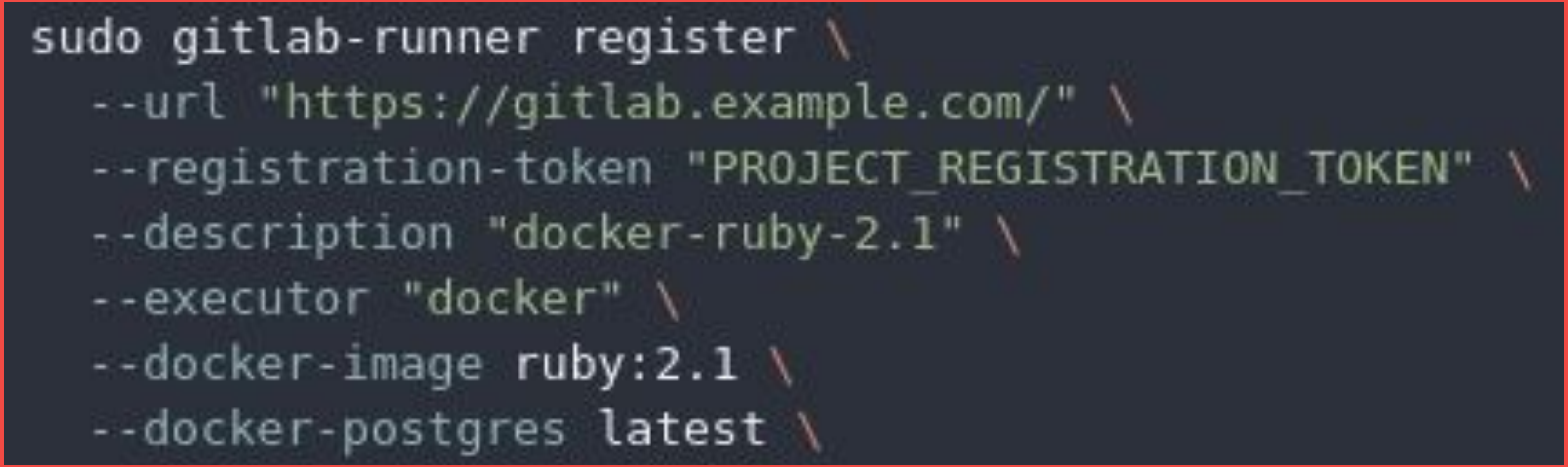
```
git add .gitlab-ci.yml

git commit -m “Add .gitlab-ci.yml”

git push origin master
```

NOLSATU

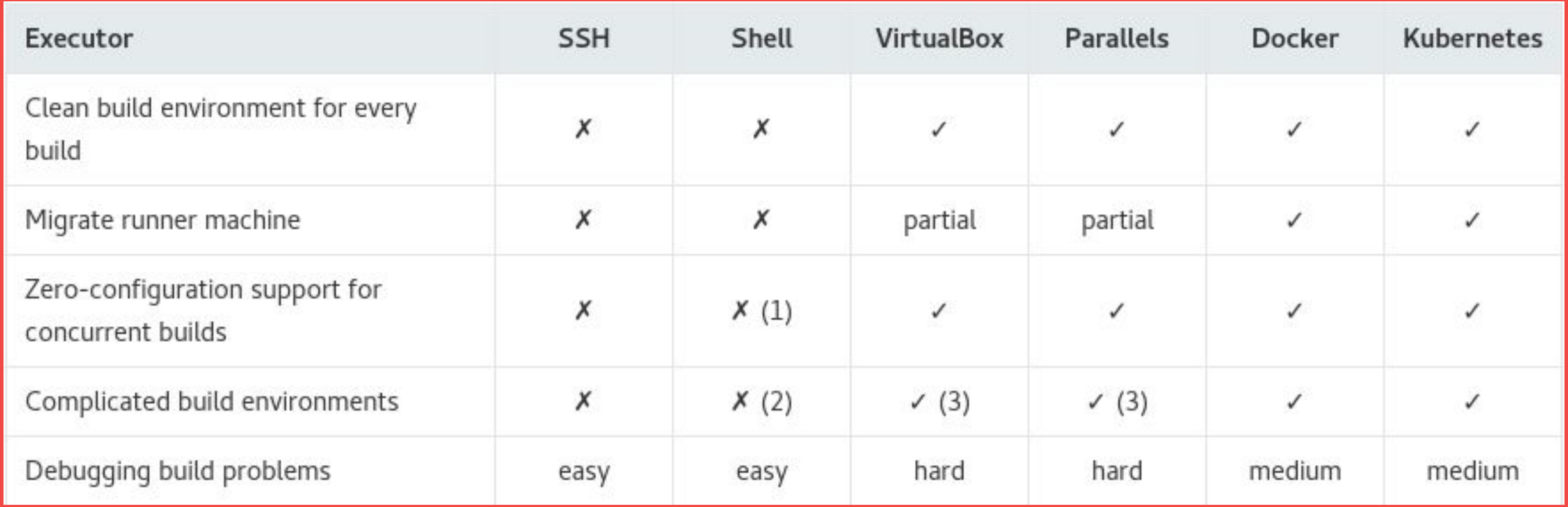
Configure a Runner



```
sudo gitlab-runner register \
  --url "https://gitlab.example.com/" \
  --registration-token "PROJECT_REGISTRATION_TOKEN" \
  --description "docker-ruby-2.1" \
  --executor "docker" \
  --docker-image ruby:2.1 \
  --docker-postgres latest \
```

NOLSATU

Selecting the Executor



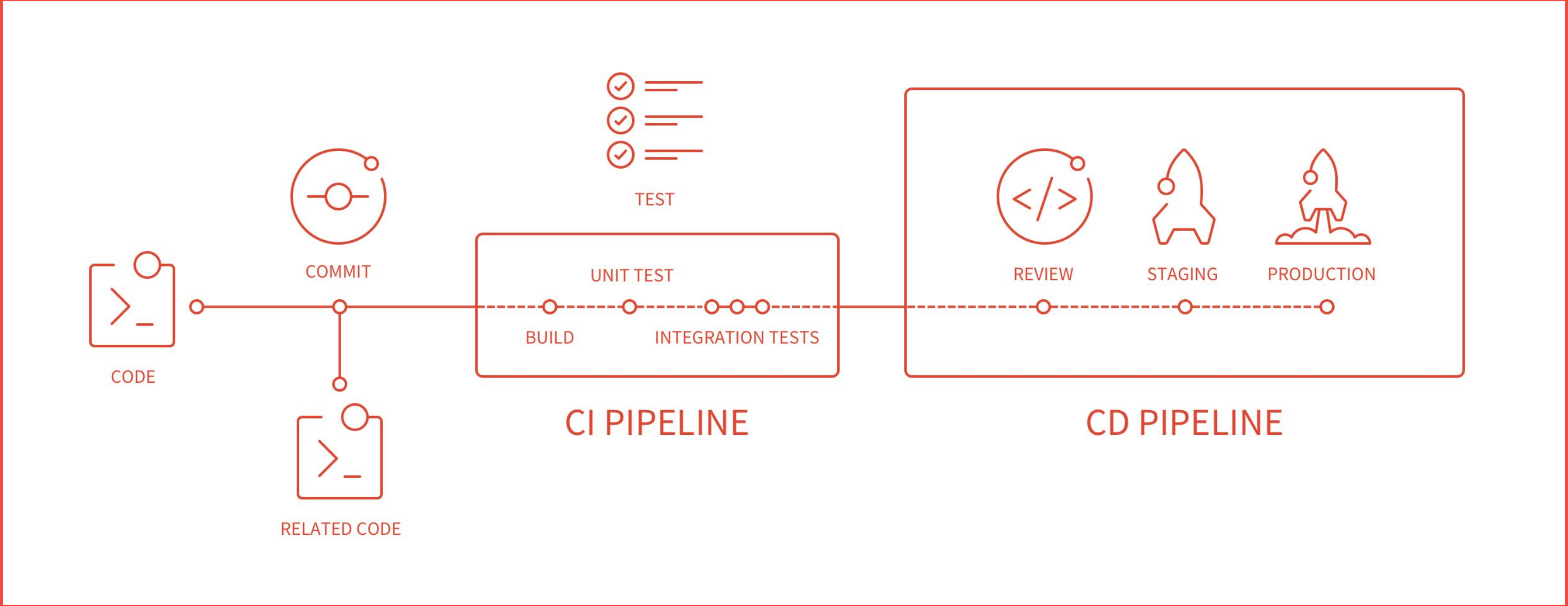
Executor	SSH	Shell	VirtualBox	Parallels	Docker	Kubernetes
Clean build environment for every build	✗	✗	✓	✓	✓	✓
Migrate runner machine	✗	✗	partial	partial	✓	✓
Zero-configuration support for concurrent builds	✗	✗ (1)	✓	✓	✓	✓
Complicated build environments	✗	✗ (2)	✓ (3)	✓ (3)	✓	✓
Debugging build problems	easy	easy	hard	hard	medium	medium

GitLab CI/CD

The benefits of Continuous Integration are huge when automation plays an integral part of your workflow. GitLab comes with built-in Continuous Integration, Continuous Deployment, and Continuous Delivery support to build, test, and deploy your application

NOLSATU

GitLab CI/CD



NOLSATU

CD Heroku

Heroku provides a variety of continuous delivery tools to help you deploy effortlessly and safely:

- Pipelines make it easy to maintain separate staging and production environments for your app.
- Review apps let you try out a GitHub pull request’s changes in an isolated and disposable environment.
- Heroku CI automatically runs your app’s test suite on new GitHub pull requests, or when code is merged into your repo’s master branch.

NOLSATU

GitLab CI/CD for Docker

GitLab CI in conjunction with GitLab Runner can use Docker Engine to test and build any application.

Docker is an open-source project that allows you to use predefined images to run applications in independent "containers" that are run within a single Linux instance. Docker Hub has a rich database of pre-built images that can be used to test and build your applications.

NOLSATU

Review Apps

Review Apps transform the way you and your team work. For developers this means you can kick off your feedback process much sooner. Submit a merge request and preview your changes in a live environment. When you push to master, you'll know your changes have gone through CI testing as well as a live review where the team can make sure things work as planned.

NOLSATU

Review Apps

Review Apps are a collaboration tool that takes the hard work out of providing an environment to showcase product changes.

Review Apps:

- Provide an automatic live preview of changes made in a feature branch by spinning up a dynamic environment for your merge requests.
- Allow designers and product manages to see your changes without needing to check out your branch and run your changes in a sandbox environment.
- Are fully integrated with the GitLab DevOps LifeCycle.
- Allow you to deploy your changes wherever you want.

NOLSATU

Auto DevOps

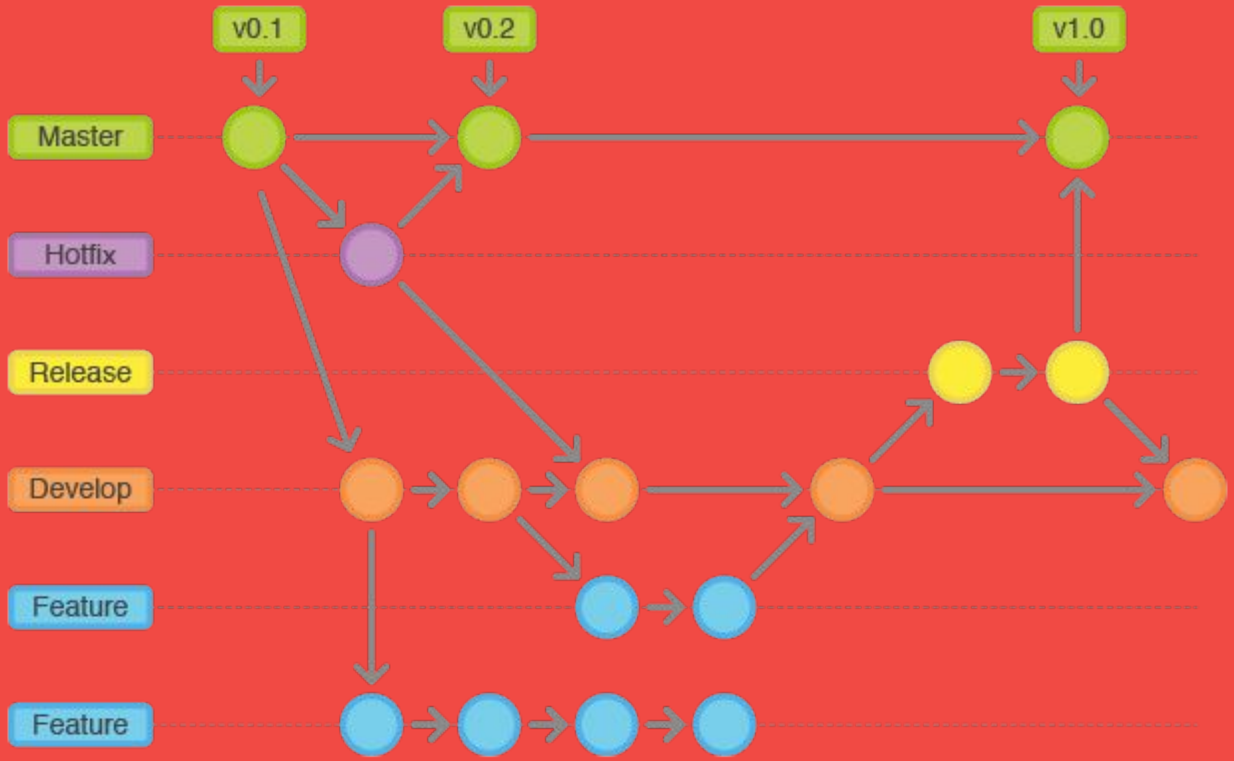
Auto DevOps provides pre-defined CI/CD configuration which allows you to automatically detect, build, test, deploy, and monitor your applications. Leveraging CI/CD best practices and tools, Auto DevOps aims to simplify the setup and execution of a mature & modern software development lifecycle.

NOLSATU

Git Flow

NOLSATU

Git Flow



NOLSATU

Git Branch

A branch represents an independent line of development. Branches serve as an abstraction for the edit/stage/commit process.

```
$ git branch
# list all branch
```

```
$ git checkout -b develop
# create branch develop and switch branch to develop
```

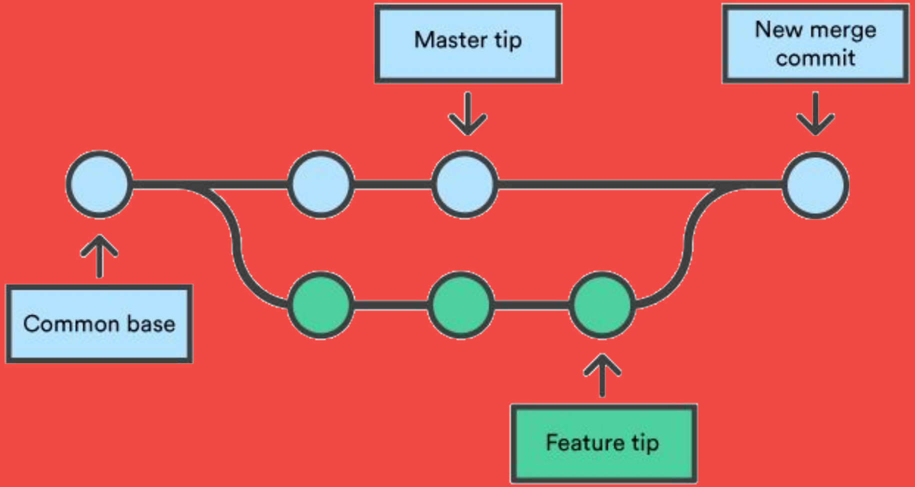
Git Checkout

The git checkout command lets you navigate between the branches created by git branch.

NOLSATU

Git Merge

Git merge will combine multiple sequences of commits into one unified history. In the most frequent use cases, git merge is used to combine two branches.



NOLSATU

Lab

GitLab CI/CD

NOLSATU

NolSatu.id

NOLSATU