GitLab

CREATE, COLLABORATE AND DEPLOY

What is Continuous Integration/Delivery (CI/CD)?

Continuous Integration is the practice of integrating code into a shared repository and building/testing each change automatically, as early as possible - usually several times a day.

Continuous Delivery adds that the software can be released to production at any time, often by automatically pushing changes to a staging system.

Continuous Deployment goes further and pushes changes to production automatically.

Why Continuous Integration/Delivery (CI/CD)?

Continuous Integration

- Detects errors as quickly as possible
  - Fix while fresh in your mind
- Reduces integration problems
  - Smaller problems are easier to digest
  - Don’t compound problems
- Allows teams to develop faster, with more confidence

Continuous Delivery

- Ensures that every change to the system is releasable
- Lowers risk of each release - makes releases “boring”
- Delivers value more frequently
- Get fast feedback on what users care about

Why GitLab CI/CD?

- **Integrated**: GitLab CI is part of GitLab. You can use it for free on GitLab.com, comes with CE/EE.
- **Easy to learn**: Simple to get started, powerful when you need it. See our Quick Start guide.
- **Beautiful**: GitLab CI offers the same great experience as GitLab. Familiar, easy to use, and beautiful.
- **Scalable**: Tests run distributed on separate machines of which you can add as many as you want or even scale automatically.
- **Faster results**: Each build can be split in multiple jobs that run in parallel on multiple machines.
- **Open source**: CI is included with both the open source GitLab Community Edition and the proprietary GitLab Enterprise Edition.

GitLab CI/CD Features

- **Multi-platform:** you can execute builds on Unix, Windows, OSX, and any other platform that supports Go.
- **Multi-language:** build scripts are command line driven and work with Java, PHP, Ruby, C, and any other language.
- **Parallel builds:** GitLab CI splits builds over multiple machines, for fast execution.
- **Autoscaling:** you can automatically spin up and down VM's or Kubernetes pods to make sure your builds get processed immediately while minimizing costs.
- **Realtime logging:** a link in the merge request takes you to the current build log that updates dynamically.
- **Versioned tests:** a .gitlab-ci.yml file that contains your tests, allowing developers to contribute changes and ensuring every branch gets the tests it needs.

GitLab CI/CD Features

- **Pipeline:** define multiple jobs per stage and even trigger other pipelines.
- **Build artifacts:** upload binaries and other build artifacts to GitLab and browse and download them.
- **Test locally**: reproduce tests locally using `gitlab-runner exec`.
- **Docker support**: use custom Docker images, spin up services as part of testing, build new Docker images, run on Kubernetes.
- **Container Registry**: built-in container registry to store, share, and use container images.
- **Continuous Delivery (CD)**: Continuous delivery and deployment are easy with multiple types of jobs, and secure environmental variables.
- **Environments**: define multiple environments including temporary Review Apps, see deployment history for every environment.