#Introduction

GitLab is a web-based DevOps lifecycle tool that provides a Git-repository manager providing wiki, issue-tracking and CI/CD pipeline features, using an open-source license, developed by GitLab Inc.

The code was originally written in Ruby, with some parts later rewritten in Go, initially as a source code management solution to collaborate with his team on software development. It later evolved to an integrated solution covering the software development life cycle, and then to the whole DevOps life cycle. The current technology stack includes Go, Ruby on Rails and Vue.js.

It follows an open-core development model where the core functionality is released under an open source (MIT) license while the additional functionality is under a proprietary license.

#Why use Gitlab

- Free CI/CD pipeline included

GitLab comes with a powerful and well-integrated Continuous Integration (CI) and Continuous Deployment (CD) pipeline. With GitHub, we would have to pay for and manually integrate an external CI service such as CircleCI.

- Free container registry

We’re big believers in dockerizing everything. GitLab has a built-in docker registry for each repository, no configuration required. With GitHub, we’d again have to set up and pay for a third-party service to manage our docker images.

- Kubernetes integration

We love how Kubernetes simplifies our infrastructure management. So we were more than excited when GitLab released their Kubernetes integration because it’s a natural fit for our workflow.

- Review apps

We deploy every branch on every commit into its own review app. This is an immense improvement to our QA workflow because we can conduct all quality assurance directly in the actual feature branch before merging it back to the master branch. We’ve found that this process brought our QA and Dev team closer together and made the collaboration much more efficient.

- Better project management

While GitHub does offer some project management tools, we find GitLab’s boards and issue management more extensive and better suited to our workflows. Some examples would be burndown charts, group milestones, issue due dates, moving issues between projects, or the issue CSV export.

- Superior data portability

GitLab can import projects and issues from more sources than GitHub and it allows us to export our projects to other systems.

- Integrated monitoring

GitLab helps us collect performance metrics for both servers as well as our applications. It quickly lets us determine the impact of merging a particular branch and keep an eye on our production systems without ever leaving GitLab.

- Quick actions

We love GitLab’s quick actions and frequently catch ourselves trying to use them on GitHub as well — only to find that they don’t work.

- Inline commenting and discussion resolution

We find reviewing merge requests (“MRs”) with GitLab more effective due to the discussion resolution feature, which allows us to keep a more manageable overview about which code review feedback has already been addressed, and what’s still left to do