Gogs is a lightweight Git server written in Go which is designed to be simple to set up and operate and can be run on just about anything. It is 100% open source under the MIT OSS license and provided only in self-managed form. Gogs offers repository file viewing and editing, project issue tracking, and a built-in wiki for project documentation.

### Feature Comparison

#### FEATURES

<table>
<thead>
<tr>
<th>Feature</th>
<th>GitLab</th>
<th>Gogs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in CI/CD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
GitLab has built-in Continuous Integration/Continuous Delivery, for free, no need to install it separately. Use it to build, test, and deploy your website (GitLab Pages) or webapp. The job results are displayed on merge requests for easy access.

Learn more about CI/CD

Runs with less memory and consumes less CPU power

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Bronze</td>
<td>Silver</td>
<td>Gold</td>
</tr>
</tbody>
</table>

Uses little memory, it runs fine with 512MB. Uses little CPU power since Go is a compiled language

Issues

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Bronze</td>
<td>Silver</td>
<td>Gold</td>
</tr>
</tbody>
</table>

Quickly set the status, assignee or milestone for multiple issues at the same time or easily filter them on any properties. See milestones and issues across projects.

Learn more about the Issue Tracker

Labels

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Bronze</td>
<td>Silver</td>
<td>Gold</td>
</tr>
</tbody>
</table>

Labels provide an easy way to categorize issues, merge requests, or epics based on descriptive titles as ‘bug’ or ‘documentation’.

Learn more about GitLab Labels

Milestones

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Bronze</td>
<td>Silver</td>
<td>Gold</td>
</tr>
</tbody>
</table>

Create and manage milestones at both the project and group levels, viewing all the issues for the milestone you’re currently working on, representing an Agile program increment or a release.

Learn more about Milestones

Iterations

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Bronze</td>
<td>Silver</td>
<td>Gold</td>
</tr>
</tbody>
</table>

Create and manage iterations at the group level, view all the issues for the iteration you’re currently working on within your group or project, and enable all subgroups and projects to stay in sync on the same cadence.

Learn more about Iterations

Issue Due Dates

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Bronze</td>
<td>Silver</td>
<td>Gold</td>
</tr>
</tbody>
</table>

In GitLab, you can set a due date for individual issues. This is very convenient if you have small tasks with a specific deadline.

Due dates documentation

Move Issue to Another Project

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Bronze</td>
<td>Silver</td>
<td>Gold</td>
</tr>
</tbody>
</table>

You can move issues between projects in GitLab. All links, history and comments will be copied and the original issue will reference the newly moved issue. This makes working with multiple issue trackers much easier.

Learn more about moving issues between projects

Mark Issue as Duplicate

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Bronze</td>
<td>Silver</td>
<td>Gold</td>
</tr>
</tbody>
</table>
Mark an issue as a duplicate of another issue, closing it.

Learn more about marking duplicate issues

**Todos**

When a user is mentioned in or assigned to an issue or merge request it will be included in the user Todos, making the development workflow faster and easier to track.

Learn more about Todos

**Project Issue Board**

GitLab has Issue Boards, each list of an Issue Board is based on a label that exists in your issue tracker. The Issue Board will therefore match the state of your issue tracker in a user-friendly way.

Learn more about GitLab Issue Boards

**Time Tracking**

Time Tracking in GitLab lets your team add estimates and record time spent on issues and merge requests.

Learn more about Time Tracking

**Repository pull mirroring**

Mirror a repository from a remote Git server to your local server, making it easy to keep local forks and replicas up to date.

Learn more about repository pull mirroring

**Create new branches from issues**

In GitLab, you can quickly create a new branch from an issue on the issue tracker. It will include the issue number and title automatically, making it easy to track which branch belongs to which issue.

See how in our documentation

**Application performance monitoring**

GitLab collects and displays performance metrics for deployed apps, leveraging Prometheus. Developers can determine the impact of a merge and keep an eye on their production systems, without leaving GitLab.

Learn more about monitoring deployed apps

**Application performance alerts**

GitLab allows engineers to seamlessly create service level indicator alerts and be notified of any desired events, all within the same workflow where they write their code.

Learn more about creating SLI alerts

**Value Stream Analytics**

GitLab provides a dashboard that lets teams measure the time it takes to go from planning to monitoring. GitLab can provide this data because it has all the tools built-in: from the idea, to the CI, to
code review, to deploy to production.

Learn more about Value Stream Analytics

Group Level Value Stream Analytics

GitLab provides a group dashboard that lets teams measure the time it takes to go from planning to monitoring. GitLab can provide this data because it has all the tools built-in: from the idea, to the CI, to code review, to deploy to production.

Learn more about Value Stream Analytics

Create merge request from email

Create a merge request from email by sending in the merge request title, description, and source branch name. Alternatively use patch files to create a merge request without first pushing a branch.

Built-in Container Registry

GitLab Container Registry is a secure and private registry for Docker images. It allows for easy upload and download of images from GitLab CI. It is fully integrated with Git repository management. (Codefresh will be ending their support for private docker registries as of May 1, 2020)

Documentation on Container Registry

Preview your changes with Review Apps

With GitLab CI/CD you can create a new environment for each one of your branches, speeding up your development process. Spin up dynamic environments for your merge requests with the ability to preview your branch in a live environment. Review Apps support both static and dynamic URLs.

Learn more about Review Apps

Environments Auto-stop

This feature allows users to configure an optional expiration date which can be set for review app environments.

Learn more about Environments auto-stop

New features every month

GitLab is updated with new features and improvements every month on the 22nd.

One integrated tool

Other tools require the integration of multiple 3rd party tools to complete the software development lifecycle. GitLab has a completely integrated solution that covers the entire development lifecycle.

AD / LDAP integration

Sync groups, manage SSH-keys, manage permissions, authentication and more. You can manage an entire GitLab instance through the LDAP / AD integration.

More information about AD / LDAP integration
## Remote repository push mirroring

Mirror a repository from your local server to elsewhere. Push mirroring is supported via HTTP and SSH using password authentication, and using public-key authentication with SSH.

Learn more about repository push mirroring

---

## File Locking

Working with multiple people on the same file can be a risk. Conflicts when merging a non-text file are hard to overcome and will require a lot of manual work to resolve. With GitLab Premium, File Locking helps you avoid merge conflicts and better manage your binary files by preventing everyone, except you, from modifying a specific file or entire directory.

Learn more about File Locking

---

## Support for Scaled GitLab

GitLab Premium includes support for scaling GitLab services across multiple nodes to manage demands on your system and provide redundancy. GitLab has developed reference architectures so you can easily determine the optimal architecture for your needs.

Learn more about scaling GitLab

---

## Deploy Boards

GitLab Premium ships with Deploy Boards offering a consolidated view of the current health and status of each CI/CD environment running on Kubernetes. The status of each pod of your latest deployment is displayed seamlessly within GitLab without the need to access Kubernetes.

Learn more about Deploy Boards

---

## Automatically close issue(s) when a merge request is merged

With GitLab, you can use specific keywords to close one or more issues as soon as a merge request is merged.

Learn more about automatically closing issues

---

## Work in Progress merge requests (WIP)

Prevent merge requests from accidentally being accepted before they're completely ready by marking them as Work In Progress (WIP). This gives you all the code review power of merge requests, while protecting unfinished work.

Learn more about WIP MRs

---

## Git LFS 2.0 support

Manage large files such as audio, video and graphics files with the help of Git LFS. Git LFS 2.0 file locking support helps large teams work with binary assets and is integrated with our native file locking feature.

Learn more about Git LFS support in GitLab

---

## View Kubernetes pod logs

---
The monitoring of servers, application, network and security devices via generated log files to identify errors and problems for analysis. GitLab makes it easy to view the logs of running pods in connected Kubernetes clusters. By displaying the logs directly in GitLab, developers can avoid having to manage console tools or jump to a different interface. Learn more about viewing Kubernetes pod logs

Leverage virtual package registries to simplify package management workflows.

A virtual registry is a collection of local, remote and other virtual registries accessed through a single logical URL.

GitLab Epic detailing the issues required to add this functionality.

Forward requests for packages not found in GitLab to npmjs.com

By default, when an NPM package is not found in the GitLab NPM Registry, the request is forwarded to npmjs.com

Check out the docs to learn more

Conan (C/C++) Repository

Conan is an open source, decentralized and multi-platform C/C++ Package Manager for developers to create and share native binaries.

Documentation on the Conan Repository

Maven (Java) Repository

GitLab’s Maven repository makes it easier to publish and share Java libraries across an organization, and ensure dependencies are managed correctly. It is fully integrated with GitLab, including authentication and authorization.

Documentation on the Maven Repository

NPM (node) Registry

GitLab’s NPM repository makes it easier to publish and share NPM packages across an organization, and ensure dependencies are managed correctly. It is fully integrated with GitLab, including authentication and authorization.

Documentation on the NPM Registry

NuGet (.NET) Repository

GitLab’s NuGet Repository allows C#/.NET developers to create, publish and share packages using the NuGet client or visual studio.

Documentation on the NuGet Repository

PyPi (Python) Repository

Python developers can set up GitLab as a remote PyPi repository and build, publish, and share packages using the PyPi client or GitLab CI/CD.

Documentation for the PyPi Repository

RPM (Linux) Repository
This planned feature will enable Linux developers to build, publish and share RPM packages alongside their source code and pipelines.

Check out the issue for additional details on implementation and timing

**Debian (Linux) Repository**

This planned feature will enable Linux developers to build, publish and share Debian packages alongside their source code and pipelines.

Check out the issue for additional details on implementation and timing

**RubyGems (Ruby) Repository**

This planned feature will enable Ruby developers to setup GitLab as a remote RubyGems repository and to build, publish and share packages using the command line or GitLab CI/CD. This will also be a valuable feature for GitLab and help with dogfooding.

Check out the issue for additional details on implementation and timing

**Go Proxy**

This planned feature will enable Go developers to publish and share their packages right alongside their source code and pipelines. This will also be a valuable feature for GitLab and help with dogfooding.

Read the Go Proxy docs

**Composer (PHP) Repository**

This planned feature will enable PHP developers to build, publish and share their packages right alongside their source code and pipelines.

Check out the docs

**Use the Package Registry through REST API**

Enables support for automation and integration of the GitLab Package Registry through a REST API.

**Documentation on API**

**Git protocol v2 support**

Git's wire protocol defines how clones, fetches and pushes are communicated between the client and server. Git protocol v2 improves performance of fetch commands and enables future protocol improvements.

Learn more about Git protocol v2