Travis CI is a hosted, distributed continuous integration service used to build and test software projects hosted at GitHub. Travis CI also offers a self-hosted version called Travis CI Enterprise which requires either a GitHub Enterprise installation or account on GitHub.com. In contrast, GitLab.com and GitLab self-hosted versions offer both source code management, issue tracking, continuous integration, and many more DevOps toolchain requirements in a single application, while still also working with GitHub.

When Travis CI has been activated for a given repository, GitHub will notify it whenever new commits are pushed to that repository or a pull request is submitted. Travis CI will then check out the relevant branch and run the commands specified in .travis.yml, which usually build the software and run any automated tests. When that process has completed, Travis notifies the developer(s) in the way it has been configured to do so.

Although the Travis CI source is technically free software and available piecemeal on GitHub under permissive licenses, the company notes that it is unlikely that casual users could successfully integrate it on their own platforms. (ref: Wikipedia). In contrast, GitLab is open source and open core and available for everyone to contribute.

**Feature Comparison**

**FEATURES**

**Application performance monitoring**

<table>
<thead>
<tr>
<th>PLAN</th>
<th>PREMIUM</th>
<th>ULTIMATE</th>
</tr>
</thead>
</table>

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

*Monitoring performance alerts*
GitHub 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

**GitHub Self-monitoring**

GitHub comes out of the box enabled for Prometheus monitoring with extensive instrumentation, making it easy to ensure your GitHub deployment is responsive and healthy.

Learn more about GitHub self-monitoring

**Value Stream Analytics**

GitHub provides a dashboard that lets teams measure the time it takes to go from planning to monitoring. GitHub 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**

GitHub provides a group dashboard that lets teams measure the time it takes to go from planning to monitoring. GitHub 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

**Built-in Container Registry**

GitHub Container Registry is a secure and private registry for Docker images. It allows for easy upload and download of images from GitHub CI. It is fully integrated with GitHub 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 GitHub 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

**Built for using containers and Docker**

GitHub ships with its own Container Registry, Docker CI Runner, and is ready for a complete CI/CD container workflow. There is no need to install, configure, or maintain additional plugins.

Read the docs

**Cloud Native**

GitHub and its CI/CD is Cloud Native, purpose built for the cloud model. GitHub can be easily deployed on Kubernetes and used to deploy your application to Kubernetes with support out of the box.

**Kubernetes integration**

**Container debugging with an integrated web terminal**

Easily debug your containers in any of your environments using the built-in GitHub Web Terminal. GitHub can open a terminal session directly from your environment if your application is deployed on Kubernetes. This is a very powerful feature where you can quickly debug issues without leaving the comfort of your web browser.

Learn more about the web terminal

**Comprehensive pipeline graphs**
Pipelines can be complex structures with many sequential and parallel jobs. To make it a little easier to see what is going on, you can view a graph of a single pipeline and its status.

Learn more about pipeline graphs

**Browsable artifacts**

With GitLab CI you can upload your job artifacts in GitLab itself without the need of an external service. Because of this, artifacts are also browsable through GitLab’s web interface.

Learn more about using job artifacts in your project

**Latest artifacts locked to prevent deletion**

The latest artifact of a successful job and pipeline on any active branch, MR, or tag is automatically locked to prevent being deleted. This makes it possible to set an aggressive expiration policy to clean up older artifacts, reduce disk space consumption, and ensure the latest artifact is always available.

Learn more about job artifacts expiration

**Scheduled triggering of pipelines**

You can make your pipelines run on a schedule in a cron-like environment.

Learn how to trigger pipelines on a schedule in GitLab

**Code Quality MR Widget**

Code Quality reports are available in the merge request widget area, giving you early insights into how the change will affect the health of your code before deciding if you want to accept it.

Learn more about Code Quality

**Code Quality Reports**

Full Code Quality reports are available on the pipeline page, showing areas of the codebase that do not meet the organization’s preferred style or standards.

Learn more about Code Quality reports

**Multi-project pipeline graphs**

With multi-project pipeline graphs you can see how upstream and downstream pipelines are linked together for projects that are linked to others via triggers as part of a more complex design, as it is for micro-services architecture.

Learn more about multi-project pipeline graphs

**Protected variables**

You can mark a variable as “protected” to make it available only to jobs running on protected branches, therefore only authorized users can get access to it.

Learn how to use protected variables

**Environments and deployments**

GitLab CI is capable of not only testing or building your projects, but also deploying them in your infrastructure, with the added benefit of giving you a way to track your deployments. Environments are like tags for your CI jobs, describing where code gets deployed.

Learn more about environments

**Environments history**

Environments history allows you to see what is currently being deployed on your servers, and to access a detailed view for all the past deployments. From this list you can also re-deploy the current version, or even rollback an old stable one in case something went wrong.

Learn more about history of an environment

**Environment-specific variables**
Limit the environment scope of a variable by defining which environments it can be available for.

Learn how to configure environment-specific variables

Group-level variables

Define variables at the group level and use them in any project in the group.

Learn how to configure variables

Customizable path for CI/CD configuration

You can define a custom path into your repository for your CI/CD configuration file.

Learn how to configure a custom CI/CD configuration file

Run CI/CD jobs on Windows

GitLab Runner supports Windows and can run jobs natively on this platform. You can automatically build, test, and deploy Windows-based projects by leveraging PowerShell or batch files.

Install GitLab Runner on Windows

Run CI/CD jobs on macOS

GitLab Runner supports macOS and can run jobs natively on this platform. You can automatically build, test, and deploy for macOS-based projects by leveraging shell scripts and command line tools.

Install GitLab Runner on macOS

Run CI/CD jobs on Linux ARM

GitLab Runner supports Linux operating systems on ARM architectures and can run jobs natively on this platform. You can automatically build, test, and deploy for Linux ARM-based projects by leveraging shell scripts and command line tools.

Install GitLab Runner on Linux ARM

Run CI/CD jobs on FreeBSD

GitLab Runner supports FreeBSD and can run jobs natively on this platform. You can automatically build, test, and deploy for FreeBSD-based projects by leveraging shell scripts and command line tools.

Install GitLab Runner on FreeBSD

Show code coverage rate for your pipelines

GitLab is able to parse job output logs and search, via a customizable regular expression, any information created by tools like SimpleCov to get code coverage. Data is automatically available in the UI and also as a badge you can embed in any HTML page or publish using GitLab Pages.

Learn how to generate and show code coverage information in GitLab

Details on duration for each command execution in GitLab CI/CD

Other CI systems show execution time for each single command run in CI jobs, not just the overall time. We’re reconsidering how job output logs are managed in order to add this feature as well.

Read more on the issue

Auto DevOps

Auto DevOps brings DevOps best practices to your project by automatically configuring software development lifecycles by default. It automatically detects, builds, tests, deploys, and monitors applications.

Read more about Auto DevOps in the documentation

Protected Runners
Protected Runners allow you to protect your sensitive information, for example deployment credentials, by allowing only jobs running on protected branches to access them.

**Easy Integration of existing Kubernetes clusters**

Add your existing Kubernetes cluster to your project, and easily access it from your CI/CD pipelines to host Review Apps and to deploy your application.

**GitLab Kubernetes Agent**

Manage the deployments and connection to your Kubernetes clusters in a secure and compliant way, driven by code.

**Easy creation of Kubernetes clusters on GKE**

Create a Kubernetes cluster on GKE directly from your project, just connecting your Google Account and providing some information. The cluster can be used also by Auto DevOps to deploy your application.

**Support for multiple Kubernetes clusters**

Easily deploy different environments, like Staging and Production, to different Kubernetes clusters. This allows to enforce strict data separation.

**Easy Deployment of applications for Kubernetes clusters**

Install Helm, Ingress, Cert-Manager, Prometheus, GitLab Runner, JupyterHub, and Knative directly into your cluster from the GitLab Web UI with one click.

**Canary Deployments**

GitLab Premium can monitor your Canary Deployments when deploying your applications with Kubernetes.

**Automatic Retry for Failed CI Jobs**

You can specify a retry keyword in your .gitlab-ci.yml file to make GitLab CI/CD retry a job for a specific number of times before marking it as failed.

**Pipelines security**

The ability of running CI/CD pipelines on protected branches is checked against a set of security rules that defines if you’re allowed or not. It includes creating new pipelines, retrying jobs, and perform manual actions.

**Include external files in CI/CD pipeline definition**

You can include external files in your pipeline definition file, using them as templates to reuse snippets for common jobs.
### Static Application Security Testing

GitLab allows easily running Static Application Security Testing (SAST) in CI/CD pipelines; checking for vulnerable source code or well known security bugs in the libraries that are included by the application. Results are then shown in the Merge Request and in the Pipeline view. This feature is available as part of Auto DevOps to provide security-by-default.

Learn more about Static Application Security Testing

### Secret Detection

GitLab allows you to perform Secret Detection in CI/CD pipelines; checking for unintentionally committed secrets and credentials. Results are then shown in the Merge Request and in the Pipeline view. This feature is available as part of Auto DevOps to provide security-by-default.

Learn more about Secret Detection

### Dependency Scanning

GitLab automatically detects well known security bugs in the libraries that are included by the application, protecting your application from vulnerabilities that affect dependencies that are used dynamically. Results are then shown in the Merge Request and in the Pipeline view. This feature is available as part of Auto DevOps to provide security-by-default.

Learn more about Dependency Scanning

### Container Scanning

When building a Docker image for your application, GitLab can run a security scan to ensure it does not have any known vulnerability in the environment where your code is shipped. Results are then shown in the Merge Request and in the Pipeline view. This feature is available as part of Auto DevOps to provide security-by-default.

Learn more about Container Scanning

### Dynamic Application Security Testing

Once your application is online, GitLab allows running Dynamic Application Security Testing (DAST) in CI/CD pipelines; your application will be scanned to ensure threats like XSS or broken authentication flaws are not affecting it. Results are then shown in the Merge Request and in the Pipeline view. This feature is available as part of Auto DevOps to provide security-by-default.

Learn more about Dynamic Application Security Testing

### Interactive Application Security Testing

IAST combines elements of static and dynamic application security testing methods to improve the overall quality of the results. IAST typically uses an agent to instrument the application to monitor library calls and more. GitLab does not yet offer this feature.

### Browser Performance Testing

Easily detect performance regressions for web apps, prior to merging into master. Browser Performance Testing is included in Auto DevOps, providing automatic performance analytics of the root page with zero configuration.

Learn more about Browser Performance Testing

### Load Performance Testing

Easily detect performance regressions for software under load, prior to merging into master.

Learn more about Load Performance Testing

### Automated Accessibility scanning of Review Apps

Performing accessibility testing is important in order to ensure you’re serving all the users who use your products. In GitLab you can generate Accessibility reports automatically prior to merging into master.

Learn more about Automated Accessibility scanning
Step folding for CI/CD logs

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Monthly</td>
<td>Shared</td>
<td>Fixed</td>
</tr>
</tbody>
</table>

Collapse the job log output for each command.

Documentation

View Kubernetes pod logs

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Monthly</td>
<td>Shared</td>
<td>Fixed</td>
</tr>
</tbody>
</table>

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

Windows Container Executor

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Monthly</td>
<td>Shared</td>
<td>Fixed</td>
</tr>
</tbody>
</table>

With this feature you are able to use Docker containers on Windows directly, in much the same way as if they were on Linux hosts. This enables more advanced kinds of pipeline orchestration and management for users of Microsoft platforms.

Learn more about the Windows Container Executor

Visual Reviews

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Monthly</td>
<td>Shared</td>
<td>Fixed</td>
</tr>
</tbody>
</table>

Visual Review allow users to give feedback on a proposed change in a Merge Request directly from the Review App itself. This feature enables designers, Product Managers, and other stakeholders to comment on the changes to the look and feel / user experience of a change just as easily and quickly as developers working in the MR.

Learn more about Visual Reviews

On-demand Dynamic Application Security Testing

<table>
<thead>
<tr>
<th>Core</th>
<th>Starter</th>
<th>Premium</th>
<th>Ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Monthly</td>
<td>Shared</td>
<td>Fixed</td>
</tr>
</tbody>
</table>

"There's no reason to wait for the next CI pipeline run to find out if your site is vulnerable or to reproduce a previously found vulnerability. GitLab offers scanning your running application with On-demand Dynamic Application Security Testing (DAST), independent of code changes or merge requests.%"  

Learn more about On-demand DAST