GitLab vs. CircleCI

**Summary**

Founded in 2011 and headquartered in San Francisco, CA, CircleCI provides a service that automates the Continuous Integration stage of the Software Development Life Cycle (SDLC). This CI service offering can be hosted in the cloud or on a private server. CI jobs are built within five different environments: a Source repo, a Linux VM, a MacOS VM, or a Windows VM. They demonstrate their support of the Open Source Community by providing organizations with free credits for Open Source projects.

**CircleCI Features**

CircleCI can provide automated services for other stages of the Software Development Life Cycle (SDLC) using their CI platform. They call it "GitLab". CircleCI can provide automated services for other stages of the Software Development Life Cycle (SDLC) using their CI platform. They call it "GitLab". They define a set of reusable CI-as-a-service packages of YAML configurations that contain expected steps of Config to a single line of code. In other words, CircleCI as a platform that is included in the YAML configuration file that authorizes a piece of code during the build process that performs as a function. The tags are focused on an open-source code library.

**Strengths**

- **YAML Pipeline**: CI pipelines are configured in a YAML file.
- **Security**: CircleCI is very robustly secured and is compliant with SOC 2 standards.
- **Forrester Report**: CircleCI was recognized as a Leader in The Forrester Wave™: Cloud-Native Continuous Integration Tools. CI is used to scale, scale, and scale, enabling CI/CD to be automated.
- **Catering to Large Enterprises**: CircleCI offers a large number of preconfigured environments, which are highly favored by enterprises.
  - CircleCI offers support for building and testing CI/CD projects in a cloud native environment, which is now available on CircleCI Cloud only, with cost incurred on self-hosted installations.
  - CircleCI is actively working on integrating this functionality, more details can be found here.

**Feature Comparison**

**Features**

- [ ] Application performance monitoring
- [ ] Traceability
  - [ ] GitLab
  - [ ] CircleCI

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.
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 pipelines 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, or it is for microservices architecture.

Learn more about multi-project pipeline graphs

### Protected variables

You can mark a variable as "protected" to make it available only to jobs in a single or protected branch, 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 into your infrastructure, with the added benefit of giving you away 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-trigger the current version, or even rollback an old deploy in case something went wrong.

Learn more about history of an environment

### Environments specific variables

Limit the environment scope of a variable by defining which environments it can be available for.

Learn how to configure an environment's specific variables

### Group-level variables

Define variables at the group level and reuse 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

### Ran CI/CD jobs on Windows

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

Install GitLab Runner on Windows

### Ran CI/CD jobs on macOS

GitLab Runner supports macOS and can run jobs automatically 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

### Ran CI/CD jobs on Linux ARM

GitLab Runner supports Linux, operating systems on ARM architecture 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

### Ran 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 regex, any information created by tools like `SimpleCov` to get code coverage. Data is automatically available in the UI and also as a widget you can embed into 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Duration (ms)</th>
<th>Run CI on</th>
<th>Run QA on</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install</td>
<td>123</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Test</td>
<td>456</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Build</td>
<td>789</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Deploy</td>
<td>1011</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

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.

Read more on the issue:

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 deploy your applications.

Read more on the issue:

Easy creation of Kubernetes clusters on GitLab

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

Read more on the issue:

Support for multiple Kubernetes clusters

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

Read more on the issue:

Easy Deployment of applications for Kubernetes clusters

Install Helm, Tensorflow, CoreOS Cluster, Paperless, GitLab Runner, JupyterLab, and Kubernetes directly into your cluster from the GitLab WebUI with one click.

Read through the documentation on installing applications on Kubernetes clusters

Canary Deployments

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

Learn more about configuring Canary Deployments

Automatic Retry for Failed CI Jobs

You can specify a retry handler in your GitLab CI.yml file to make GitLab CI retry a job for a specific number of times before marking it as failed.

Learn more about Automatic Retry for Failed CI Jobs

Pipeline 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 denying new pipelines, depending jobs, and performing manual actions.

Learn more about pipeline security

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.

Learn more about including external files
Static Application Security Testing

GitHub allows easy running of 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

GitHub allows you to perform secret detection in CI/CD pipelines, checking for intentionally committed secrets and vulnerabilities. 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

GitHub automatically detects and 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, GitHub can run a container 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, GitHub allows running of 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 application security for containers

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. GitHub does not yet offer this feature.

Learn more about IAST

Browser Performance Testing

Easily detect performance regressions for web apps, prior to merging into master. Browser Performance Testing is enabled in Auto DevOps, providing automatic performance analysis 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 product. In GitHub, you can generate Accessibility reports automatically prior to merging into master.

Learn more about Automated Accessibility scanning

Step Folding for CI/CD jobs

Consolidate the job log output for each command.

Documentation

View Kubernetes pod logs

This enables users, applications, network and security devices to generate logs to identify access and permission for analysis. GitHub makes it easy to view the logs of running pods in the Kubernetes cluster. By displaying the logs directly in GitHub, developers can avoid having to manage intrusive tools or jump to a different interface.

Learn more about monitoring Kubernetes pods.
With this feature you can build and run Docker containers on Windows directly, in much the same way as if they were on Linux hosts. This enables more advanced kinds of CI/CD pipelines and deployments for users on Microsoft platforms.

Learn more about the Windows Container Collector

Visual Reviews

Visual Reviews allow users to give feedback on proposed changes in a Merge Request directly from the Review App itself. This feature enables designers, product managers, and other stakeholders to comment on the changes as they 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

“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 a scanning your running application with On-demand Dynamic Application Security Testing (DAST), independent of code changes or merge request.”

Learn more about On-demand DAST