Codefresh vs GitLab

GitLab compared to other DevOps tools

On this page

Summary

Codefresh is a CI/CD tool designed for containers and Kubernetes. Codefresh features aGitLab integration that allows you to use GitLab for version control and run Codefresh pipelines on your code. Codefresh has some features that make it more mature that GitLab for running pipelines across multiple projects. But it lacks the benefits of a single application.

Codefresh charges for builds per month, as well as concurrent builds. GitLab has no such limitations with the ability to elastically scale Runners to handle as many concurrent builds as needed on demand and then scale down so you aren't paying to keep inactive Runners up.

Codefresh only offers a self-managed option for Enterprise pricing. Free, Basic, and Pro tiers are for SaaS-only. GitLab offers self-managed and SaaS options at every price point.

Resources

- Codefresh homepage
- Comparison page on their site
- Codefresh GitLab integration

Comments/Anecdotes

- Codefresh makes some claims in their blog comparing themselves to GitLab that are not really accurate.
  - "GitlabCI isnâ€™t designed for micro-services since everything is tied to a single project"
    - Although we can improve our microservices support, this claim is not true. GitLab has multi project pipelines and can trigger pipelines for mulit-projects via API. In fact, The CI working group for CNCF chose GitLab to run their multi-project multi-cloud pipelines: CNCF case study, CNCF video
- We are missing some features that would bring us on par with Codefresh
  - The ability to define multiple pipelines
  - Support for monorepos with the ability to run pipelines only on specific directories
  - Group level Docker registry browser
  - Group level Kubernetes clusters
  - Make container building first class

Pricing

- Codefresh Pricing
- Codefresh prices per build and per concurrent build
- $299 Pro tier is only 3 concurrent builds, to get more you have to call for pricing

Comparison

FEATURES		
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. Documentation on Container Registry	✓	✓
Full Binary Repository A binary repository is a software repository for packages, artifacts and their corresponding metadata. It can be used to store binary files produced by an organization itself, such as product releases and nightly product builds, or for third party binaries which must be treated differently for both technical and legal reasons.	✗	✗
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. Learn more about Review Apps	✗	✓
A comprehensive API GitLab provides APIs for most features, allowing developers to create deeper integrations with the product. Read our API Documentation	✓	✓
CI/CD Horizontal Autoscaling GitLab CI/CD cloud native architecture can easily scale horizontally by adding new nodes if the workload increases. GitLab Runners can automatically spin up and down new containers to ensure pipelines are processed immediately and minimize costs. Learn more about GitLab CI/CD Horizontal Autoscaling	✓	✓
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	✓	✓

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


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


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 regex, 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 embedd in any HTML page or publish using GitLab Pages.

Learn how to generate and show code coverage information in GitLab


Manage JUnit reports created by CI jobs

Many languages use frameworks that automatically run tests on your code and create a report: one example is the JUnit format that is common to different tools. GitLab supports browsing artifacts and you can download reports, but we're still working on a proper way to integrate them directly into the product.

Read more on the issue


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.

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 to deploy your application.

Read more on the issue


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.

Read more on the issue


Support for multiple Kubernetes clusters

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

Read more on the issue

Easy Deployment of Helm, Ingress, and Prometheus on Kubernetes

Install Helm Tiller, Nginx Ingress, Prometheus and GitLab Runner directly into your cluster from the GitLab Web UI with one click.

Read through the documentation on installing applications on GKE clusters

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.

Learn more about Automatic Retry for Failed CI Jobs

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.

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

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] (https://docs.gitlab.com/ee/topics/autodevops/#auto-sast) to provide security-by-default.

Learn more about Static Application Security Testing

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](https://docs.gitlab.com/ee/topics/autodevops/#auto-dependency-scanning) 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](https://docs.gitlab.com/ee/topics/autodevops/#auto-container-scanning) 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](https://docs.gitlab.com/ee/topics/autodevops/#auto-sast) to provide security-by-default.

Learn more about application security for containers


Interactive Application Security Testing

[IAST](https://blogs.gartner.com/neil_macdonald/2012/01/30/interactive-application-security-testing/) 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.


Runtime Application Security Testing

RASP uses an agent to instrument the application to monitor library calls as the application is running in production. Unlike other security tools, RASP can take action to block threats in real-time, similar to a Web Application Firewall but from within the app's runtime environment rather than at the network layer. 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


Maven 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


Define multiple pipelines per repo

While GitLab has multi-project piplines that give you a view accross projects, each pipleine definition live in a YAML file tied to a specific project. Today, GitLab lacks the ability to define multiple pipelines in a single YAML file.

Read more on the issue


Explicit support for monorepos

The ability to execute jobs only/except when there are changes for a given path or file support monorepos where many microservices are contained in a single repo. Shipping in GitLab 11.4

Read more on the issue


Global Docker registry browser

A sinle UI view into images accross multiple repositories.

Read more on the issue


Global Kubernetes cluster configuration

A sinle UI view into Kubernetes cluster configuration accross multiple repositories.

Read more on the issue


First class container building

The ability to specify that a container should be built during a CI/CD job without needing to specify the implementation details.

Read more on the issue