



## **Gitlab VS Jenkins**

Has been a popular debate among developers for a long time. While most of the coders like Jenkins, developers cast their votes for Gitlab. Both tools come with certain strengths and weaknesses. Without further ado, let's dive into the ultimate comparison of Gitlab and Jenkins.



# **Pricing Models**

Gitlab was initially offered for free and open-source under the MIT license. Currently, the company offers three packages. The first one is the free version that offers 400 CI/CD minutes per month. The premium version charge is \$19 per user per month billed annually and offers 10,000 CI/CD minutes per month. It includes additional features such as project management, ops insights, release controls, etc. The Ultimate version costs \$99 per user per month and contains 50,000 CI/CD minutes per month. In addition to all the features of the Premium Edition, the Ultimate edition provides you with advanced features such as portfolio management, compliance or value stream management, advanced security controls, and some other extras.



#### **Core features**

When it comes to installation, Jenkins is an easy tool to set up. All you need is JRE installed on your machine. It supports Windows, Unix, and Mac platforms. You can run it as a java servlet in GlassFish, Apache Tomcat, or any other Java container; it also works with Docker. You just need to download the file and run the normal installation process. Jenkins is self-hosted and available on-premise too. The configuration of the tool is pretty simple. You can easily configure it using the web interface. The upgrade process is also a no-brainer. Extensive documentation is available for every task.



#### **Plugin Support**

As for the Gitlab vs Jenkins plugin support, Jenkins offers an extensive bundle of plugins for every type of CI/CD task. There are more than 1700 plugins available for users that can be easily customized. The plugins are regularly updated. You can integrate these plugins with other DevOps tools as well. Jenkins is significantly extensible owing to its huge plugin portfolio. Whether you want to perform a single task of integrating build processes or manage the entire product development lifecycle, Jenkins makes it easy.



#### **Continuous Integration / Continuous Deployment**

Gitlab comes with a built-in CI/CD feature that helps you quickly set up CI/CD pipelines. Auto DevOps is another powerful feature of Gitlab that makes software delivery easy, efficient, and automated. It uses predefined default CI/CD templates that scan your code language and automatically build, test, and deploy tasks. It also includes vulnerability testing and secure auditing that proactively scans the project for vulnerabilities, licensing issues, security flaws, etc. Auto DevOps can be applied at project level, group level, or instance level. Gitlab gets an edge here in the GitLab vs Jenkins battle.



## **Auto Scaling**

When it comes to Jenkins, the auto-scale option is rather complicated to make changes to the Jenkins architecture workflow. A script is required to install software on Jenkins runners to provide an automated way to register with Jenkins Primary and then disconnect before termination. In addition, it is important to set up and monitor auto-scale triggering events using a tool like CloudWatch.



# **Technical Support**

One of the key reasons for the success of Jenkins is its community support. Right from installation to usage, the Jenkins community is ready to offer support for any type of task. On the other hand, I must mention that while documentation is available, there is no dedicated technical support team.



## **Parallel Execution**

By allowing you to concurrently run multiple builds across multiple virtual machines, Jenkins makes it easy for developers to continuously and quickly test their code. Gitlab also facilitates parallel execution. It clones the job and runs it in parallel in the background.



#### **REST API Support**

Jenkins REST API supports XML, Python, and JSON, enabling you to extend its functionality.

Gitlab offers a REST API for projects, groups, and standalone ones.



## Work Distribution

Jenkins makes it easy for you to perform CI/CD tasks as you can distribute workloads such as builds, tests, and deployments across multiple machines, platforms, and regions. With distributed workloads and parallel execution, it has faster lifecycles

with reduced time frames.



However, the initial costs and configuration complexity can also become an issue. It all comes down to your project requirements, in-house proficiency, budgets, etc.

