

# Industry uses of Jenkins



Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

Some of the Uses of Jenkins :

- Software build using build systems such as **Gradle**, **Maven**, and more.
- Automation testing using test frameworks such as **Nose2**, **PyTest**, **Robot**, **Selenium**, and more.
- Execute test scripts (using Windows terminal, Linux shell, etc.

- Achieve test results and perform post actions such as printing test reports, and more.
- Execute test scenarios against different input combinations for obtaining improved test coverage.
- **Continuous Integration & Continuous Delivery** — As an extensible automation server, Jenkins can be used as a simple CI server or turned into the continuous delivery hub for any project.
- **Easy Installation** — Jenkins is a self-contained Java-based program, ready to run out-of-the-box, with packages for Windows, Mac, and other Unix operating systems.
- **Easy Configuration** — Jenkins can be easily set up and configured via its web interface.
- **Plugins** — Hundreds of plugins are available in its marketplace to easily integrate Jenkins with any tool.
- **Extensible** — It can be extended via its plugin architecture, providing nearly infinite possibilities for what Jenkins can do.
- **Distributed** — It can be easily distributed across multiple machines, helping drive builds, tests and deployments across multiple platforms faster.

## Continuous Integration (CI)

Continuous integration is a practice that forces developers to frequently integrate their code into a central repository. Instead of building out new features to the end without any quality measurement, every change is tested against the central repository in order to anticipate errors.

Every developer commits daily to a shared mainline and every commit triggers an automated process to build and test. If building or testing fails it can be detected and fixed within minutes without compromising the whole structure, workflow, and project. In that way, it is possible to isolate problems, solving them faster and provide higher-quality products.

## Continuous Delivery (CD)

Continuous delivery is the ability to make changes of all types — such as new features, configuration changes, error fixes, experiments — into production in a safe and efficient manner using short work cycles.

The main goal in continuous delivery is to make deployments predictable as routine activities that can be achieved upon request. To be successful, the code needs to always be in a deployable state even when there is a scenario with lots of developers working and making changes on a daily basis. All of the code progress and changes are delivered in a nonstop way with high quality and low risks. The end result is one or more artifacts that can be deployed to production.

# Continuous Deployment (CD)

Continuous deployment, also known as continuous implementation, is an advanced stage of continuous delivery that the automation process does not end at the delivery stage. In this methodology, every change that is validated at the automatic testing stage is later implemented at the production stage.

The *fail fast* strategy is always of the utmost importance when deploying to production. Since every change is deployed to production, it is possible to identify edge cases and unexpected behaviors that would be very hard to identify with automated tests. To fully take advantage of *continuous deployment*, it is important to have solid logging technology that allows you to identify the increasing error count on newer versions. In addition, a trustworthy orchestration technology like [Kubernetes](#) that will allow the new version to slowly be deployed to users until the full rollout or an incident is detected and the version is canceled.

## Automation

As a job executor, Jenkins can be used to automate repetitive tasks like backup/restore databases, turn on or turn off machines, collect statistics about a service and other tasks. Since every job can be scheduled, repetitive tasks can have a desired time interval (like once a day, once a week, every fifth day of the month, and so forth).

## Jenkins alternatives

Although Jenkins is a good option for an automated, CI/CD server, there are other options on the market such as Gitlab CI/CD, Circle CI, Travis or Bamboo.

### GitLab CI/CD

GitLab is a full-featured software development platform that includes a module called GitLab CI/CD to leverage the ability to build, test, and deploy without external requirements (such as Jenkins). It is a single application that can be used in all stages of the developers' work cycle on the same project: product, development, QA, security, and operations.

GitLab is a solution that enables teams to cooperate and work from a single step instead of managing thousands of threads across disparate tools. It provides a single data store, one user interface, and one permission model across the developers' life cycle. This permits teams to collaborate reducing cycle time and focusing on building software more quickly and efficiently.

Though Gitlab covers the CI/CD cycle thoroughly, it fails to do so for automation tasks since it does not have scheduling options. It can be a very good alternative since it integrates source code versioning and CI into the same tool.

Gitlab comes in a variety of flavors: there is a community, open-source edition that can be deployed locally, and some paid versions with an increasing number of features.

## Circle CI

Circle CI is a hosted continuous integration server. After Circle CI is authorized on GitHub or Bitbucket, every code change triggers tests in a clean container or VM. After this, an email is sent every time there is a successful test completed or a failure. Any project with a reporting library provides code test coverage results. Circle CI is simple to configure, has a comprehensive web interface, and can be integrated with multiple source code versioning tools.

## Bamboo CI

Bamboo is a solution for continuous integration, deployment, and delivery. Bamboo allows you to create a multi-stage build plan, set up triggers upon commits, and assign agents to builds and deployments. It also allows you to run automated tests in every code change which makes catching bugs easier and faster. Bamboo supports continuous deliveries as well.

Bamboo's brightest feature is its seamless integration with Atlassian products: Jira Software, Bitbucket, and Fisheye, and can be improved with hundreds of add-ons that are available at Atlassian marketplace.

## Travis CI

Travis is another open source solution that also offers a free hosted option for open source projects (paid for enterprise clients). It uses a solution similar to *Jenkins Pipelines*: you add a file called *.travis.yml* that describes the project's own build workflow. It also has parallel jobs builds but it does not have the same size of add-ons available for Jenkins.