



# Jenkins

[Introduction to Jenkins](#)

[What is Jenkins and why to use it?](#)

## Introduction to Jenkins

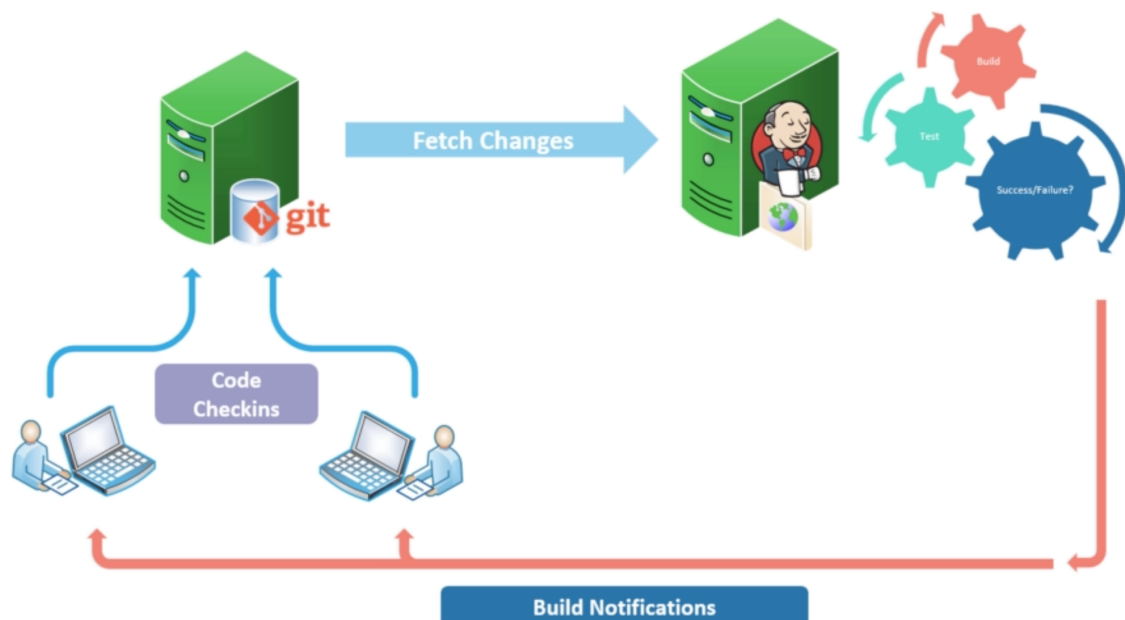
Before get into Jenkins let's try to understand the famous terminolgy CI/CD

### | What is Continuous Integration, Delivery and Deployment

- **Continuous Integration (CI)** is the practice which helps to merge all developer working copies to the central shared location.
- **Continuous Delivery (CD)** is the approach which helps teams to produce in short cycles, Ensures that software developed is more reliable. The main objective of CD is building, testing and releasing software faster and more frequent.
- **Continuous Deployment (CD)** Once the tests have been validated on the dev environment, it must be put into production. Continuous deployment, therefore, consists of automating deployment actions that were previously performed manually.

| Let's checkout how this works in Practical.

- As part of software development one or more developers used to develop the code regularly and push it to common source code repository (GIT)
- Then CI server (Jenkins) will fetches this code from GIT and then run the Build process which produces the bundle of softwares.
- These artifacts are stored in and central repository which is famously know Artifactory.
- Then CI server might also run several autoamted tests (Unit & Integration tests) on artifacts which could valuate the build.
- CI server will also might send the notification to developers about the sucess and failure of build.
- This cycle will continue to happen in an automated fashion once after every single commit done by developers in Git.



Why this approach becomes so popular these days ?

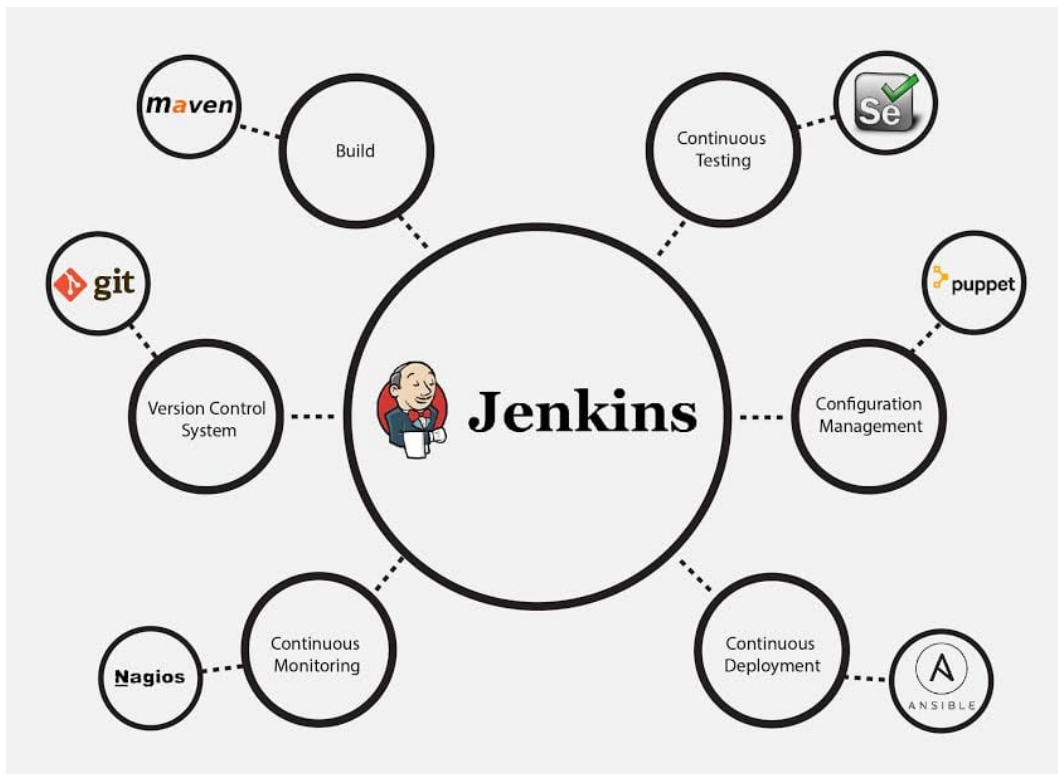
- Reduced risk on doing things manually
- Increased confidence about your product
- Better quality code
- Ready to ship code
- Time to market
- Reduced cost

## Tools available to perform CI/CD operations in market

- Bamboo
- Travis CI
- Cruise Control
- Jenkins

## What is Jenkins and why to use it?

- Jenkins is an open-source automation tool written in Java with plugins built for Continuous Integration purposes.
- Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.
- It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.
- Jenkins achieves Integration with the help of plugins. Plugins allow the integration of Various DevOps tools.
- If you want to integrate a particular tool, you need to install the plugins for that tool. For example Git, Maven, Ansible & Terraform.

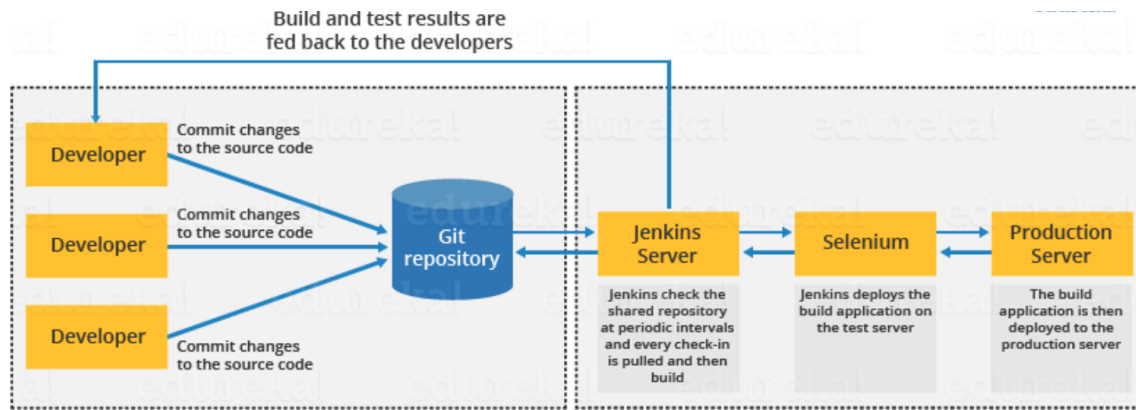


## Advantages of Jenkins include

- It is an open-source tool with great community support.
- It is easy to install.
- It has 1000+ plugins to ease your work. If a plugin does not exist, you can code it and share it with the community.
- It is free of cost.
- It is built with Java and hence, it is portable to all the major platforms.

Let's see how Jenkins will help to do achieve the tasks seamlessly.

- Let me first explain to you a generic flow diagram of Continuous Integration with Jenkins so that it becomes self-explanatory.



- First, a developer commits the code to the source code repository. Meanwhile, the Jenkins server checks the repository at regular intervals for changes.
- Soon after a commit occurs, the Jenkins server detects the changes that have occurred in the source code repository. Jenkins will pull those changes and will start preparing a new build.
- If the build fails, then the concerned team will be notified.
- If built is successful, then Jenkins deploys the built in the test server.
- After testing, Jenkins generates a feedback and then notifies the developers about the build and test results.
- It will continue to check the source code repository for changes made in the source code and the whole process keeps on repeating.