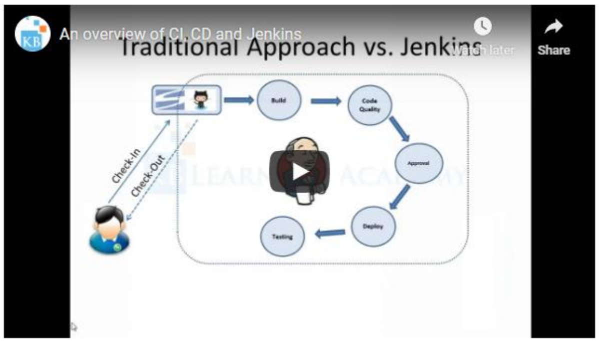
View the lesson (What is Jenkins?)

What is Jenkins?



Jenkins is a self-contained, open-source automation server that can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software. It helps to achieve the Continuous Integration process.

Jenkins is free and written entirely in Java. It is a commonly used worldwide application which is growing day by day. Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with a Java Runtime Environment (JRE) installed.

It is a server-based system that runs in a web server such as Apache Tomcat. With Jenkins, organizations can speed up the software development process by automation. Jenkins combines all sorts of life-cycle development processes including build, document, test, package, stage, deploy, static analysis, and much more.

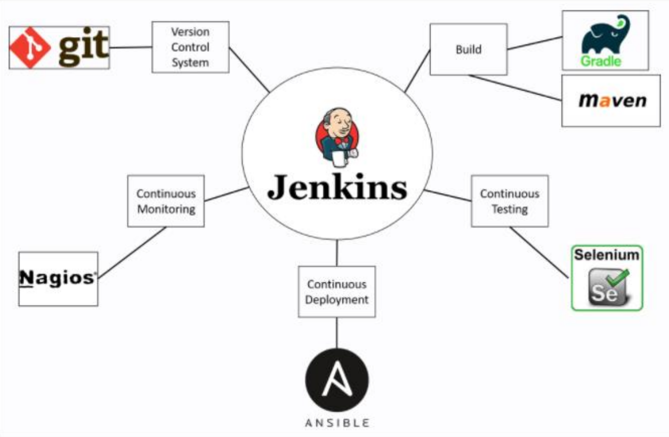
Continuous Integration

Continuous integration (CI) is the practice of automating the integration of code changes from all developers into shared mainline several times a day. The software is immediately built and tested in Continuous Integration after a code commit. If the test is passed, build is tested for deployment. The code is moved to production if the deployment is a success. All of these is a continuous process and thus the name continuous integration.

Before Jenkins, once the assigned coding tasks had been completed by all Developers, they used to commit their code simultaneously. The build is later tested and deployed. As the code was built all at once, some developers would have to wait until other developers had finished coding to test their build. Isolating, detecting, and fixing errors on several commits is not easy. When all bugs are fixed and tested the code is deployed. So, the Development Cycle is sluggish.

With Jenkins, after a code commit, the software is built and tested immediately. Since the code is built after every single developer commits, it's easy to detect whose code caused the build to fail. After each successful build and test, the code is deployed. So, it accelerates the software development process.

Through using plugins, Jenkins achieves Continuous Integration. Plugins allow various DevOps stages to be integrated. When you want to integrate a specific application, the plugins for that application must be installed.



Jenkins Continues Integration

Jenkins History

Jenkins was originally developed as the Hudson project. Hudson's creation started in the summer of 2004 at Sun Microsystems. It was first released in java.net in Feb, 2005.

In 2011, after the acquisition of Sun Microsystems by Oracle, it is forked and renamed as Jenkins.

On February 1, 2011, Oracle said that they intended to continue development of Hudson, and considered Jenkins a fork rather than a rename. Jenkins and Hudson therefore continued as two independent projects. As of June 2019, the Jenkins organization on GitHub had 667 project members and around 2,200 public repositories, compared with Hudson's 28 project members and 20 public repositories with the last update in 2016. With time, Jenkins became more popular, and Hudson is not maintained anymore.

Jenkins Terminology

Name	Description
Agent	An agent is typically a machine, or container, which connects to a Jenkins master and executes tasks when directed by the master.
Build	Result of a single execution of a Project
Core	The primary Jenkins application (jenkins.war) which provides the basic web UI, configuration, and foundation upon which Plugins can be built.
Job/Project	Jenkins seems to use these terms interchangeably. They all refer to runnable tasks that are controlled/monitored by Jenkins.
Master	The central, coordinating process which stores configuration, loads plugins, and renders the various user interfaces for Jenkins.
Node	A machine that is part of the Jenkins environment and capable of executing Pipelines or Projects. Both the Master and Agents are considered to be Nodes.
Pipeline	A suite of plugins that supports implementing and integrating continuous delivery pipelines into Jenkins.
Plugin	An extension to Jenkins functionality provided separately from Jenkins Core.
Publisher	Part of a Build after the completion of all configured Steps which publishes reports, sends notifications, etc.
Step	A single task; fundamentally steps tell Jenkins what to do inside of a Pipeline or Project.

Complementary Interactive Lesson about What Jenkins is

What Is Jenkins? | What Is Jenkins And How It Works? | Jenkins Tutorial For Beginners | Simplilearn
Watch later
Share
simplilearn
WHAT IS JENKINS?
JENKINS TUTORIAL