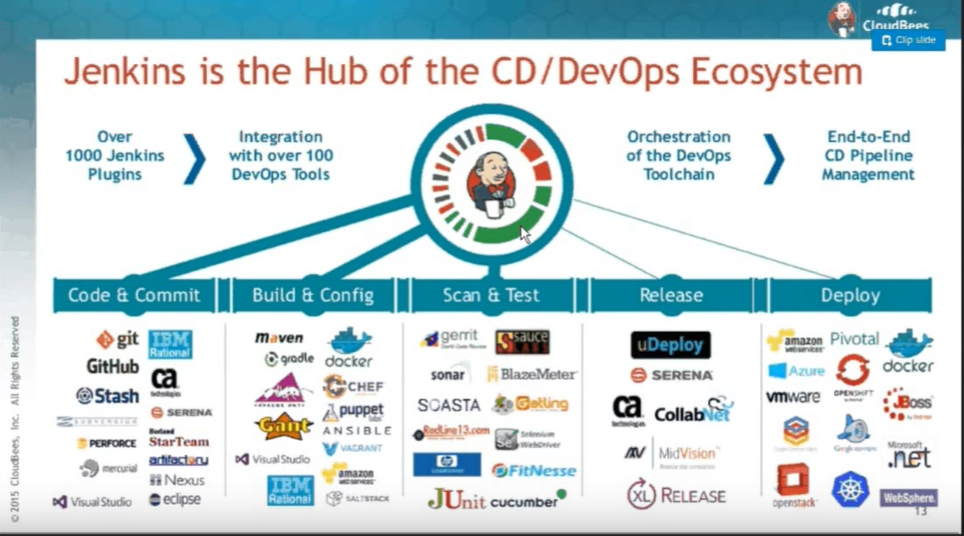
**Jenkin CI/CD pipeline**

Jenkin is continuous integration and continuous development tool through which I can make sure that all my code is continuously integrating and continuously deploying.

Under the Jenkins, there is complete pipeline is given. As soon as you commit your code, after that there is build and configuration happened at the Jenkins site and after that scanning and testing is done. It can be your Unit/Integration testing/other testing, then there is a release and then there is a deployment. UI testing is done after the deployment. Everything after 1st step i.e. commit happened automatically. Earlier it was a manual process. Jenkins will manage all these things on your own.

Jenkins are on cloud server which is also known as enterprise server, not reside on the local machine.

Similar to Jenkins, we have bamboo tool for the same purpose.



Jenkins is a web application, created in Java. Download the war file of Jenkins. Run the following command- “java -jar Jenkins.war”.

1. Install following plugins- testNG, Maven, Git
2. Give the local path of GIT, JAVA and Maven in Global configuration there.
3. Go to the new item> Give a name to your project> Click on Maven project
4. Give the description. Select Git under source code management and give the URL of your repository.
5. Click on credentials. Give your username and password. Add it.
6. Check the checkbox “build whenever snapshot dependency is built under build triggers.
7. I want whenever I commit the code, automatically my Jenkins will do the build and test my all testcases. Go to “Poll SCM”. We can poll our git that in every minute, my Jenkins will check if something is committed or not. It has 5 fields9 (Minute, hour, day of the month, month, day of the week).

Minute- 0 to 59

Hour- 0 to 23

Day of the month(DOM)- 1 to 31

Month- 1 to 12

DOW (day of week)- 0 to 7 ( where 0 and 7 are Sunday)

“\*\*\*\*\*” -> will keep on checking

“\*/2\*\*\*\*” -> build in every 2 minute

1. In Pre-shell, you can execute something from shell command or batch command
2. In the build, we need to give the path of pom.xml in the project. Mostly it is placed at the root folder. And give the goal- “clean install”
3. In enterprise Jenkins, step-1 and 2, we don’t have do. It is provided there.

**Jenkin pipeline**

It is plugin inside the Jenkin, you can create some workflows inside the Jenkin build. It is lot more similar to Jenkins jobs but you can control whole job workflow using Jenkin file.

In Jenkins, a pipeline is a group of events or jobs which are interlinked with one another in a sequence.

In simple words, Jenkins Pipeline is a combination of plugins that support the integration and implementation of **continuous delivery pipelines** using Jenkins. A pipeline has an extensible automation server for creating simple or complex delivery pipelines "as code," via pipeline DSL (Domain-specific Language).

In a Jenkins pipeline, every job or event has some sort of dependency on at least one or more events.

The picture above represents a continuous delivery pipeline in Jenkins. It contains a group of states called build, deploy, test and release. These events are interlinked with each other. Every state has its events, which work in a sequence called a continuous delivery pipeline.

A continuous delivery pipeline is an automated expression to display your process for getting software for version control. Thus, every change made in your software goes through a number of complex processes on its way to being released. It also involves developing the software in a reliable and repeatable manner, and progression of the built software through multiple stages of testing and deployment.