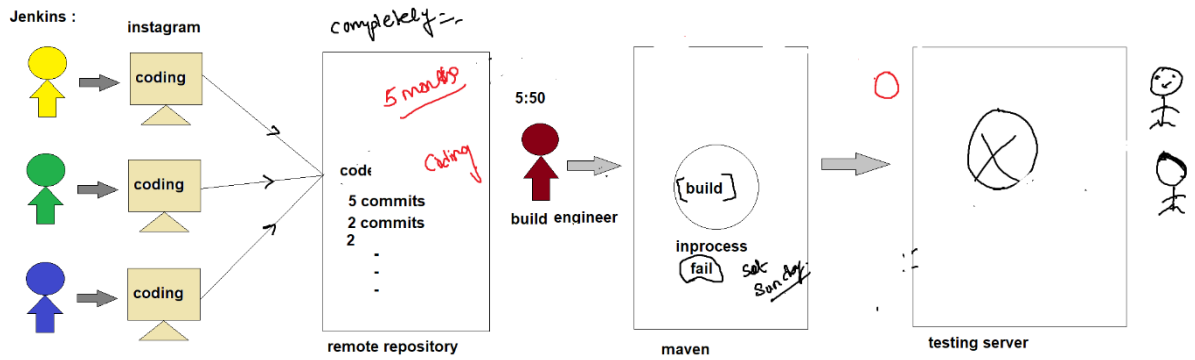
Jenkins

Traditional Scenario before Jenkins :



Drawbacks before Jenkins :

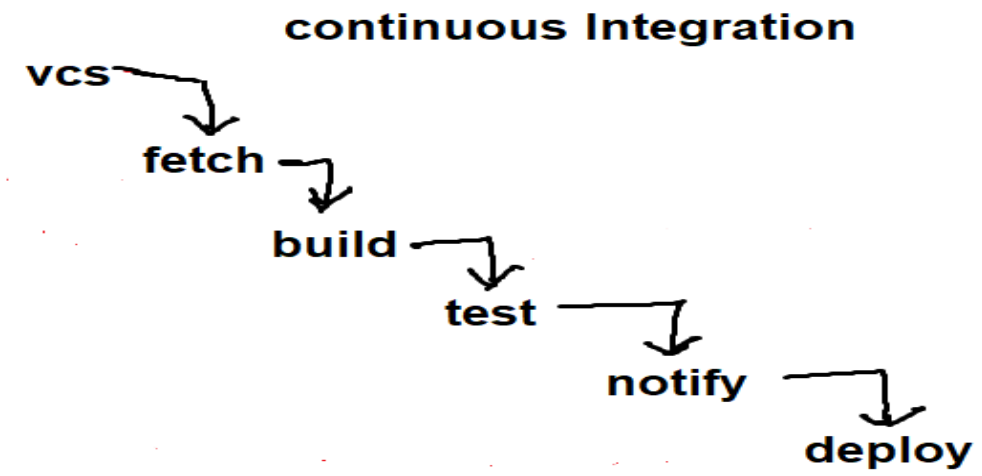
- Developers has to wait till the entire software code has build and tested to check errors.
- Software delivery will get delayed.
- Issues in Integration.
- Irregular commits.
- Delay in testing.
- Wastage of time.

Continuous Integration

Continuous integration is the practice of automating the integration of code changes from multiple contributors into a single software.

(Or)

Continuous integration is a primary devOps best practice, allowing developers to frequently merge code changes into a remote repository where builds and tests and deploy.



Continuous Integration tools :

1. Jenkins
2. Bamboo
3. Buddy
4. Build Bot
5. Circle CI
6. Travis CI

1. **Bamboo :** It is a continuous integration tool which was invented by Atlassian in 2007.
    - It supports git and maven
    - It is not an opensource tool.

2. **Buddy :** It is a Continuous Integration tool which automates the process of software development and delivery which was invented in the year 2012.
    - it is not Open source.

3. **Build Bot :** It is also a Continuous integration tool which was invented in the year 2003.
    - It was developed for python projects
    - It was written in python.
    - It will support only windows Operating system.
    - It got stable in the year 2022.

4. **Circle CI :** It is a continuous integration tool which was invented in the year 2011.
    - It was in use till 2019 by Facebook, Sony and Spotify.

5. **Travis CI :** It was invented in the year 2011 in Germany and it was developed using ruby language.
    - It was used for both continuous integration and continuous delivery.
    - It is not an open source tool.

Introduction to Jenkins :

- Jenkins is an automation tool which allows Continuous integration.
- It is an Opensource tool and it is free.
- It was written in java.
- Jenkins is dependent on java so, to install Jenkins we need JDK software to be installed.
- Jenkins can be used in any operating systems like Linux, Ubuntu, Redhat, windows and macOS etc.,
- It allows us to build, test the software project continuously which makes developers to integrate the changes to the project and find bugs easily and can solve them in a short period.
- It will allow continuous delivery of software from development to testing.
- Jenkins will achieve continuous integration with the help of plugins.

History of Jenkins :

- Jenkins was invented by Kosuke Kawauchi, he is a java developer.
- He was working in SUN Microsystems, he was tired of building the code and fixing the defects.
- In 2004, he created automation server called "**Hudson**" that automates build and test.
- In 2011, Oracle owned SUN microsystems and they had a dispute with Hudson opensource Community.
- So, because of dispute , Oracle took the source code of Hudson and they renamed it as "**Jenkins**" .

- Both Hudson and Jenkins continued to operate independently, but in a short spam of time Jenkins became more popular and Hudson is not maintained anymore.

Features of Jenkins :

- Easy Installation.
- Easy Configuration.
- Available plugins.
- Extensible.
- Easy distribution.
- Free and Opensource.
- Easy Integration.

Difference between Jenkins and Bamboo

Jenkins	Bamboo
1. It is an Open source Automation tool.	1.It is not an Opensource tool
2. Jenkins is free.	2.Bamboo is not free it will charge based on the agents using.
3. We can install Jenkins in any operating system like windows, red hat, Debian, Linux, macOS etc.,	3.Bamboo can be installed only on windows, Solaris, Linux and MacOS.
4. Jenkins will support databases like SQL server, MySQL, SQL lite etc.,	4.Bamboo will also support databases like MySQL, SQL lite etc.,
5. Jenkins will run on any browser like chrome, Firefox and internet explorer etc.,	5.Bamboo also supports all the browsers.
6. Jenkins will support a lot of plugins.	6.Bamboo will not support all the plugins compared to Jenkins.
7. Jenkins have a huge support from communities.	7.Bamboo does not have any support as compared to Jenkins.
8. Jenkins will support git and all types of VCS.	8.Bamboo will support git, mercurial and CVS.
9. Jenkins has almost 127 reporting plugins.	9.Only few reporting plugins are there in bamboo.
10. With the help of plugins we can do unit testing.	10.Using plugin installation we can do unit testing.

Continuous Delivery :

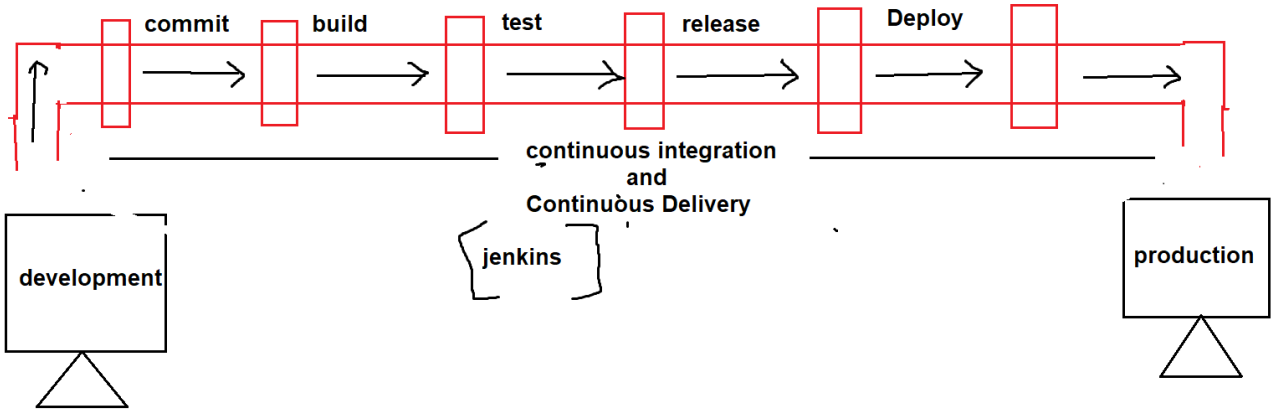
Continuous delivery is a devOps practice that refers to the building, testing and delivering improvements to the software code. The most important pat of continuous delivery is to assure the code is in deployable state.

Jenkins Pipeline :

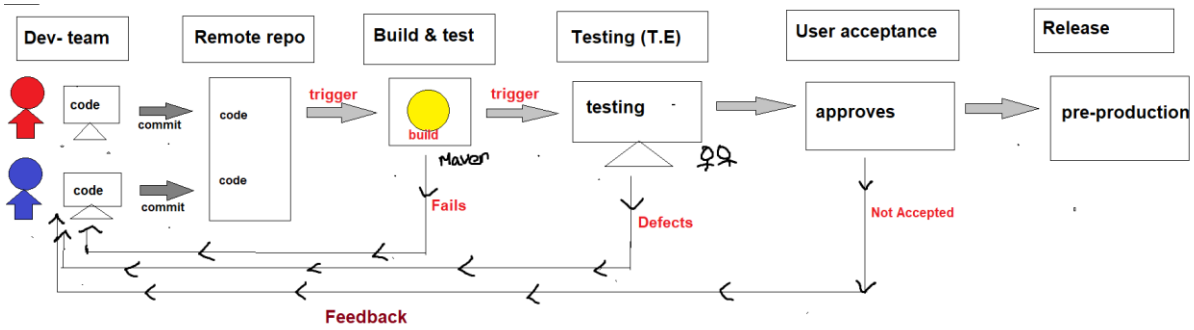
It is a plugin in Jenkins which is used to automate the continuous integration and continuous delivery using a file called Jenkins file.

Plugin : plugin is a piece of software which extends the functionality of an application.

- We need plugins in Jenkins to enhance the functionality of Jenkins for the user-specific needs.



Process :



Jenkins File :

A text file that stores the entire pipeline as a code.

- We can also store the Jenkins file in our remote repository.
- It enables the developers to access, edit and check the code all the time.

- It is written using groovy syntax.
- Jenkins file is based on two types of syntax called scripted and declarative.

There are two ways of writing Jenkins file :

1. **Scripted Pipeline.**
2. **Declarative Pipeline.**

1. Scripted Pipeline :

- It is the traditional way of writing the code in groovy.
- It is Strict way of writing the code.
- The code is defined within a node block.
- Scripted pipeline is written in Jenkins UI.

Syntax :

```
node {
        stage ( ‘ Build ’ ) {

        }
      stage ( ‘test’ ) {

        }
```

2. Declarative Pipeline :

- It follows simple groovy syntax.
- Code is written locally in a file called Jenkinsfile and commit it to the remote repo.
- The code is defined within a pipeline block.

Syntax :

```
Pipeline {
        agent any
        stages {
                stage ( 'Build' ) {
                        steps {
                        }
                }
                stage ( 'Test' ) {
                        steps {
                        }
                }
        }
}
```

Pipeline Concepts :

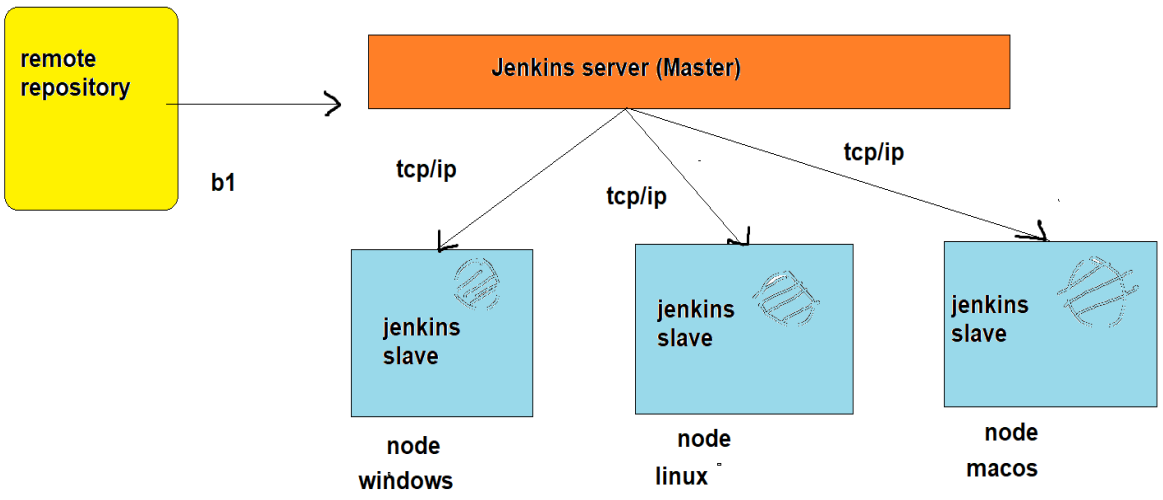
1. **Pipeline :** A user defined block which contains all the stages that has to be run by Jenkins and it is a key port of declarative syntax.
2. **Node :** A node is a "Machine" that executes an entire workflow of Jenkins and it is root element of scripted pipeline..
3. **Agent :** Agent is an executor which will run the entire pipeline, Agents can be defined for an entire pipeline or for each and every stage also.
4. **Stages :** It contains all the work and it contains each stage that performs a specific task.
5. **Steps :** Steps carried out in a sequence to execute a particular stage.

Why we need Jenkins Pipeline ?

- Using the declarative syntax makes easy to create complex pipelines as a code.
- Code will be stored in Jenkins file that can be triggered from remote repository.
- Supports complex projects by adding conditional loops.
- Improves user experience by integrating user feedback.

Master – slave Architecture :

- If we have larger project to build it on regular basis so running all the build simultaneously on the single Jenkins is not a good option.
- In this case we can use the distributed Jenkins to run all the builds of a project on a regular basis and also we can work with different environments using distributed Jenkins which will help to test the build in different environments.
- Using Distributed Jenkins we can create a master and a slave architecture where master Jenkins will distribute the loads of builds to different slaves which are connected and all the builds will run simultaneously.
- Each and every slave is considered as a “Node”.
- We can also connect different environments to the master Jenkins such as Linux, Windows, Ubuntu, MacOS etc.,
- The slave Jenkins will connect to the master Jenkins using TCP/IP protocols.



QSpiders

JSpiders