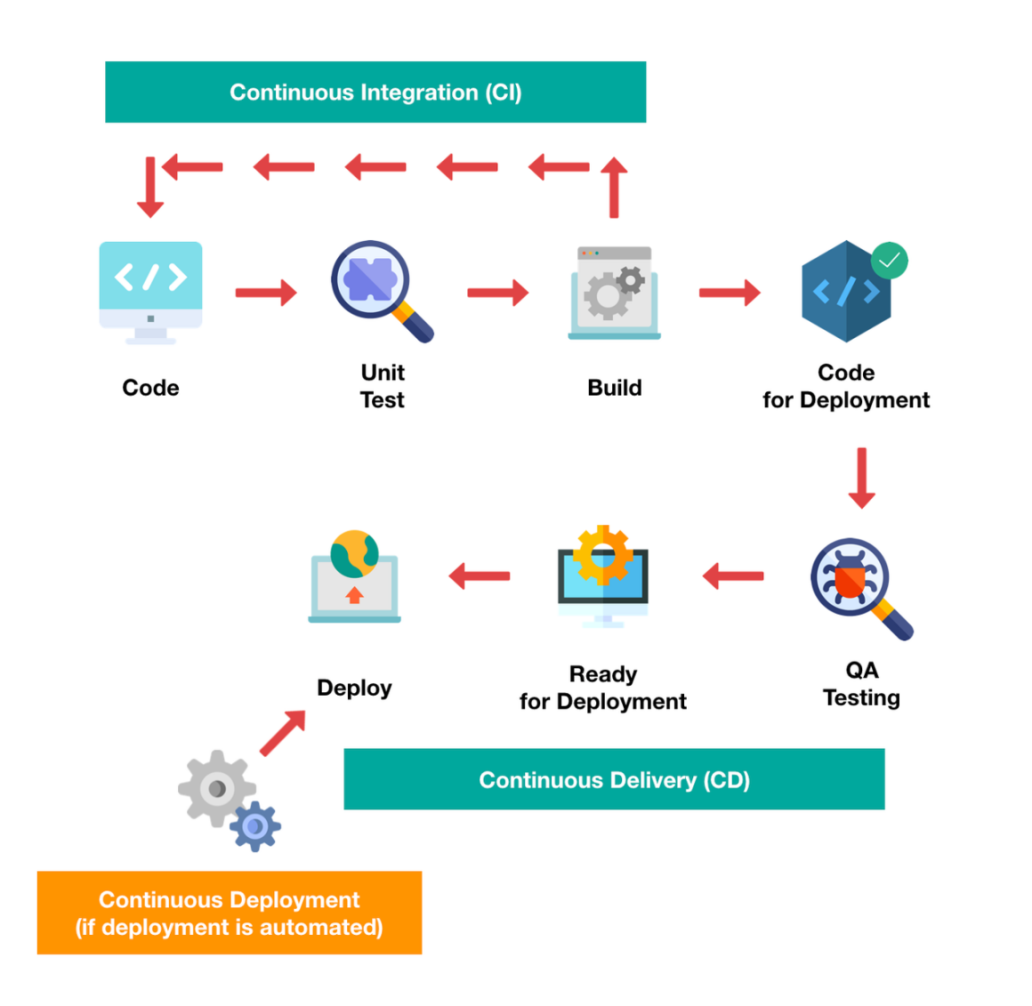
**CI/CD**



**Continuous Integration (CI)** is a development practice in which the developers are needs to commit changes to the source code in a shared repository at regular intervals. Once a developer’s changes are merged, those changes are validated by automatically building the application and running different levels of automated testing, typically unit and integration tests, to ensure the changes haven’t broken the app.

**Continuous Delivery** the validated code will be pushed to the repository/lower QA environments manually.

**Continuous Deployment** is a step up from Continuous Delivery in which every change in the source code is deployed to production automatically.

**Some tools for CI/CD**

-Jenkins

-Team City

-Bamboo

-Circle CI

-Travis CI

**Jenkins:** Jenkins is an open-source Automation server(8080 port) that is written entirely in Java which enable developers to build , test, and deploy their software.

- It is cross-platform and can be used on Windows, Mac OS, Linux etc.,



**Architecture Of Jenkins**

- Developers do the necessary modifications in the source code and commit the changes to the repository.

- The repository is continuously checked by Jenkins CI server for any changes (either in the form of code or libraries) and changes are pulled by the server.

- The Build server performs a build with the code and an executable is generated if the build process is successful. In case of a build failure, an automated email with a link to build logs and other build artifacts is sent to the developer.

-In case of a successful build, the built application (or executable) is deployed to the test server and the newly built executable goes through a series of automated tests. Developers are alerted in case the changes have caused any breakage in functionality.

-If there is no issues in the testing then tested application are automatically deployed to the Prod/Production server.

**Master- Slave Architecture In Jenkins :** is used for managing distributed builds. It will be used when single server of Jenkins is not enough.

**Master :** The main server in Jenkins is the Master.

jobs handled by Jenkins Master:

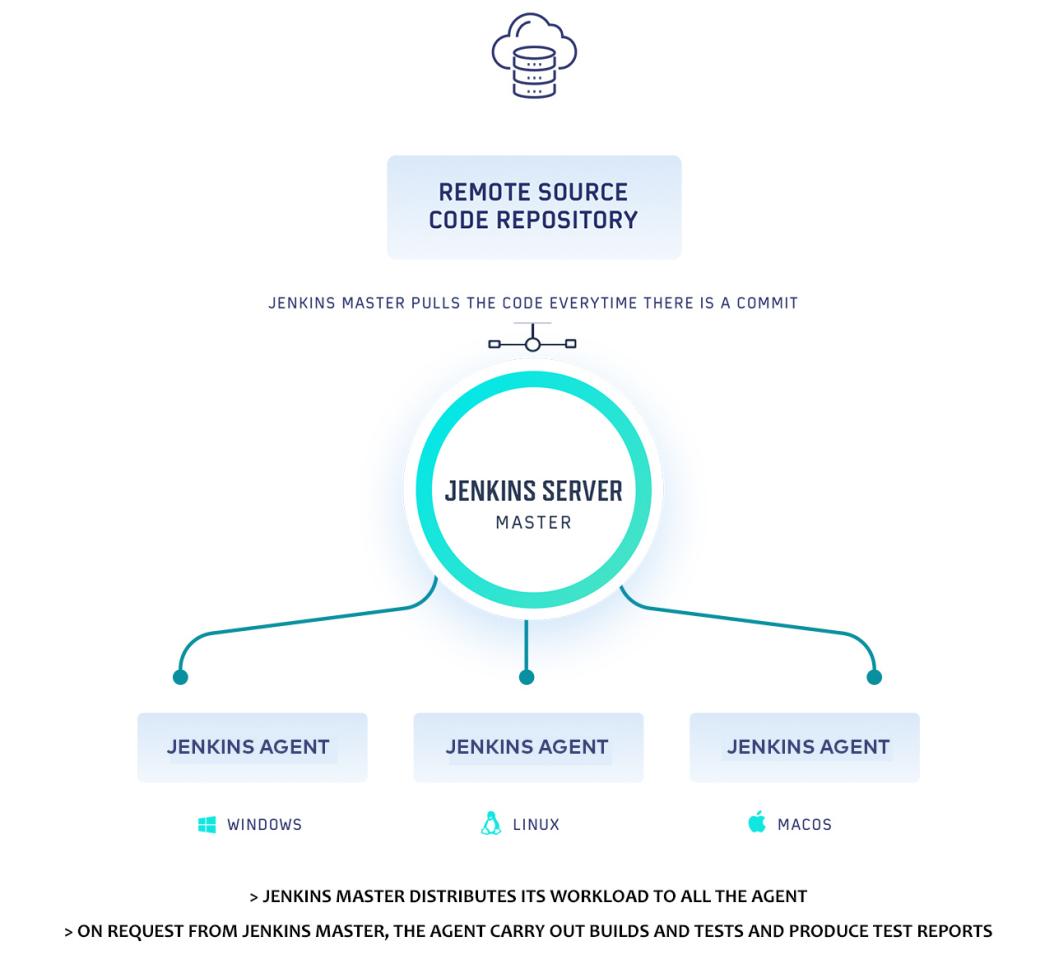
-Schedule build jobs

-Choosing the appropriate agent

-Monitor agents and take them online/offline as and when required.

-Presenting the build results (and reports) to the developer.

**Slave/Agent/Node :** is a remote machine that is connected to the Master.Its job is to execute build that are dispatched by the Master.



**Jenkins Pipeline :**

A pipeline is defined as a series of events or tasks which are interconnected in a particular order. In simple terms, the Jenkins pipeline is a set of modules or plugins which enable the implementation and integration of Continuous Delivery pipelines within Jenkins.

There are four states of **Continuous Delivery** in Jenkins pipeline-

Build

Deploy

Test

Release

**JenkinsFile :** Jenkins Pipeline can be defined by a text file called JenkinsFile. You can implement pipeline as code using JenkinsFile, and this can be defined by using a DSL (Domain Specific Language). With the help of JenkinsFile, you can write the steps required for running a Jenkins Pipeline.

**How to install jenkins?**

<https://www.lambdatest.com/blog/what-is-jenkins/>

**Source Code Management Tools Supported by Jenkins,**

Git

Subversion

Mercurial

Clearcase

**Aborting a build**

BUILD ID URL/stop

BUILD ID URL/term

BUILD ID URL/kill

**To Stop jenkins**

http://localhost:8080/exit

http://localhost:8080/restart

http://localhost:8080/reload

CMD command to stop jenkins

Ctrl + C

