JENKINS

CONTINOUS INTEGRATION:

It is the combination of continuous builds and continuous test.

CONTINUOUS INTEGRATION = CONTINUOUS BUILD + CONTINUOUS TEST

CI SERVER: Here build, test and deploy all these activities are performed in a single CI server

CONTINUOUS INTEGRATION:

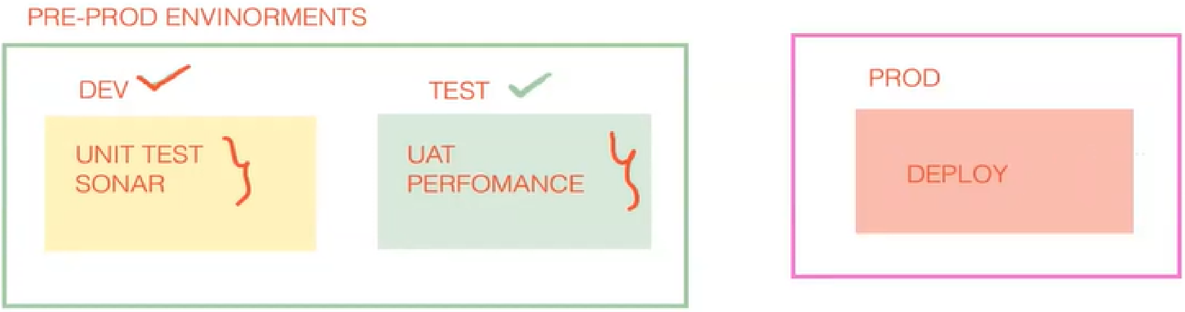
Whenever a developer commits the code using source code management like GIT, then the CI pip[eline gets the changes of the code runs automatically build and unit test.

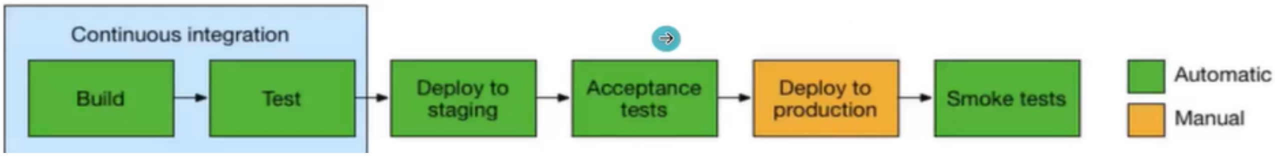
- Due to integrating the new code with old code, we can easily get to know the code is a success or failure
- it finds the errors more quickly.
- Delivery the products to client more frequently.
- Developers don't need to manual tasks.
- Reduces the developers time 20% to 30%.

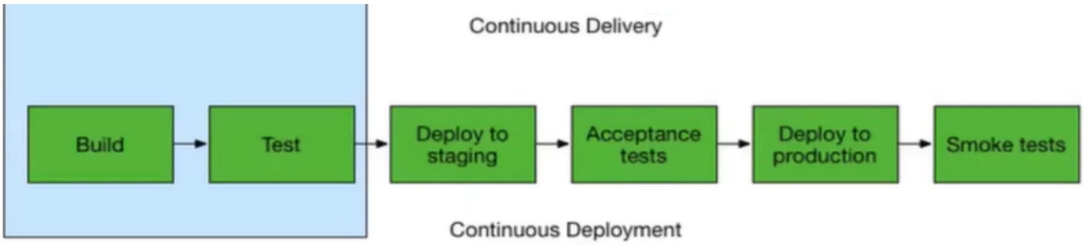
<u>CD: CONTINUOUS DELIVERY/DEPLOYMENT</u>

CONTINUOUS DELIVERY is making it available fo deployment. Anytime a new build artifact is available, the artifact is automatically placed in the desired environment and deployed.

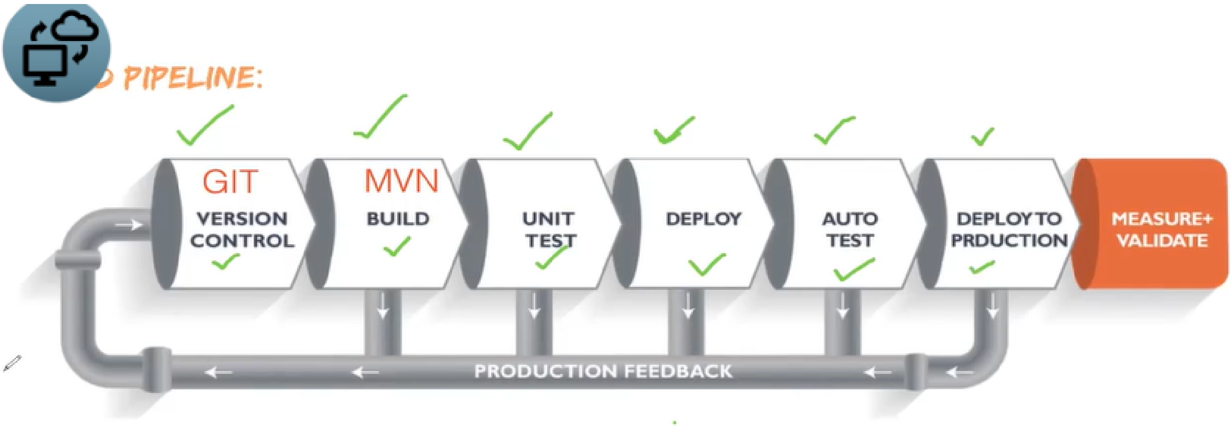
CONTINUOUS DEPLOYMENT is when you commit your code then it gets automatically tested, build and deploy on the production server.







CI/CD PIPELINE



It looks like Software Development Life Cycle, but let's see how it works.
Lets consider an example, if you are developing a web application

CI/CD PIPELINE

Auto Test: Once our code is working fine in testing servers, then we need to do Automation testing using Selenium or Junit.

Deploy to Production: If everything is fine then you can directly deploy your code in production server.

NOTE: If we have error in Code then it will give feedback and it will be corrected, if we have error in Build then it will give feedback and it will be corrected, Pipeline will work like this until it reaches Deploy.

Because of this pipeline, Bugs will be reported fast and get rectified so entire development is fast.

Version Control: Here developers need to write code for web applications. So it needs to be committed using version control system like GIT or SVN.

Build: Lets consider your code is written in java, it needs to be compiled before execution. In this build step code gets compiled.

Unit Test: If the build step is completed, then move to testing phase in this step unit step will be done.

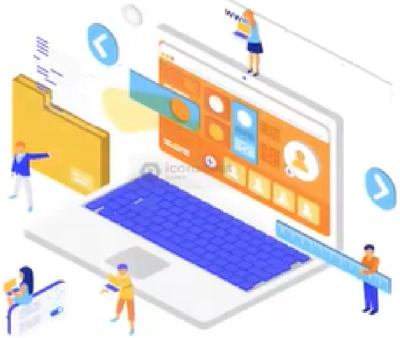
Deploy: If the test step is completed, then move to Deploy phase in this step you can deploy your code in dev, testing environment. Here you can see your application output.

KINS:

Jenkins is an **open source** project written in **java** by Kohsuke Kawaguchi it runs on the **Window, Linux and Mac OS**

It is community-supported, Free to use, and the First choice for Continuous Integration.

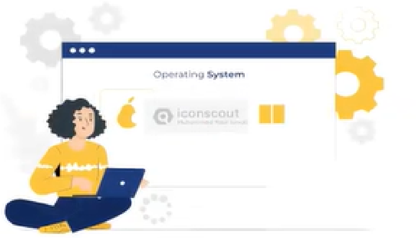
- Consist of Plugins
- Automates the Entire Software Development Life Cycle (SDLC).



- It was originally developed by Sun Microsystem in 2004 as HUDSON.
- Hudson was an enterprise Edition we need to pay for it.

1. The project was renamed Jenkins when Oracle brought the Microsystems.

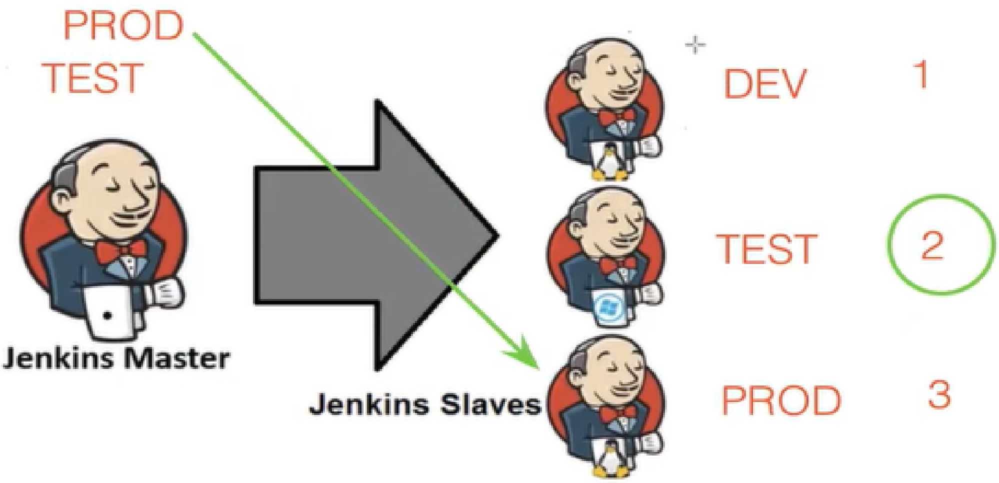
It can run on any major platform without Compatibility issues.



- Whenever developers write code, we integrate all the code of all developers at any point in time and we build, test, and deliver/deploy it to the client. This is called CI/CD.

ANTAGES:

- Jenkins follows Master-Slave Architecture.
- You can write your own plugin, can use the community plugin also.
- Can understand the process of what is going on.



- Jenkins master is going to assign a job to the slave..
- If Slaves are not available Jenkins itself does the job.
- By using the labels we can specify the jobs to the nodes.

JENKINS ALTERNATIVES:

Bamboo

GitLab

TC

circleci

Buddy

INS SETUP:

STEP-1: GETTING REPO (jenkins.io)

```
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
```

STEP-2: INSTALL JAVA

```
amazon-linux-extras install java-openjdk11 -y
```

STEP-3: INSTALL GIT MAVEN JENKINS

```
yum install git maven jenkins -y
```

STEP-4: STARTING JENKINS SERVEICE

```
systemctl start jenkins.service
systemctl status jenkins.service
```