

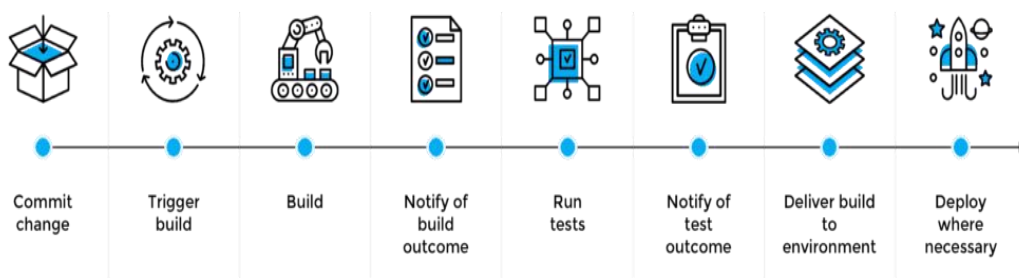
## -What is Jenkins ?

=Jenkins is an open source automation tool written in java with plugins built for Continuous Integration purposes, Its used to build and test your software projects continuously making it easier to developers to integrate changes to the project , also it allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

Here is an example of the life cycle of the CI/CD Pipeline:

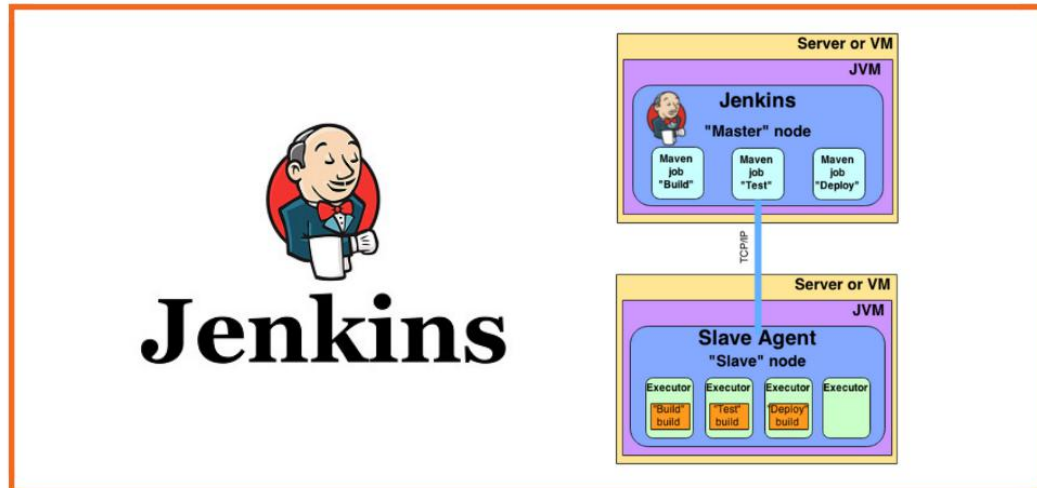
(When a a developer ensure about his code and wants to commit it to a version control tool (ex: github), he commits the source code change into a version control repository , then the CI server (jenkins) detects a new commit , compiles the code and runs unit tests , then CI server deploying the artifacts “The files generated from a build” to the automated test ,Once all tests are successful then the code is ready to deploy into the production line , In case of there is a failure in any stage the workflow will stop and send a feedback to the developer.

### CI/CD Pipeline



## What is Jenkins agent ?

Jenkins agent is an executable , residing on a node that is tasked by the controller to run a jenkins job . the jenkins agents can be managed by the master node.



## What is jenkins executor ?

Is one of the basic building blocks which allow a build to run on a node/agent

## Explain Jenkins master-slave architecture

### Jenkins Master:

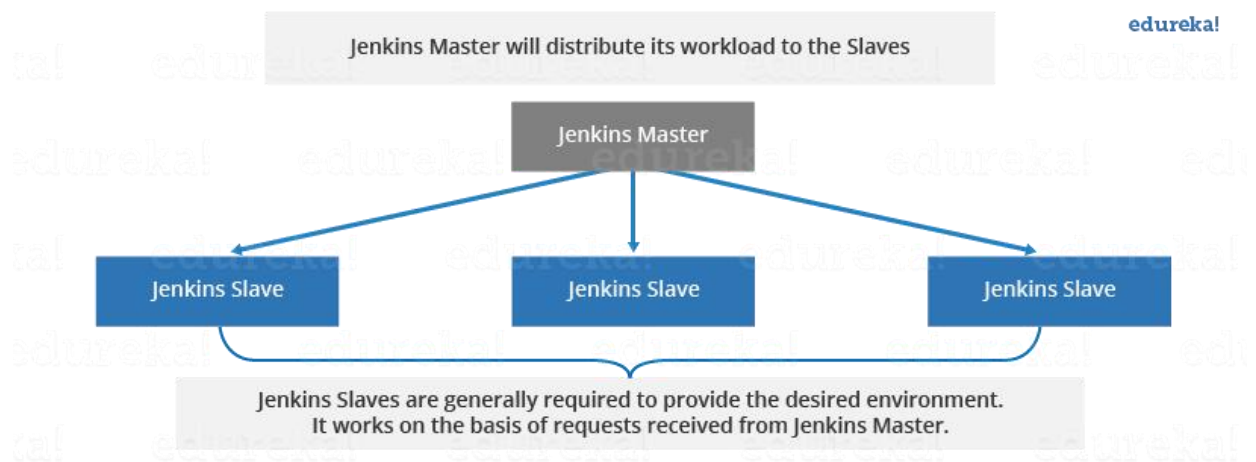
Is your main Jenkins server is the Master and its job is to handle:

- 1) Scheduling build jobs
- 2) Dispatching builds to the slaves for the actual execution
- 3) Monitor slaves
- 4) Recording and Presenting the build results
- 5) It can execute builds jobs directly

### Jenkins Slaves:

Slave is a java executable that runs on a remote machine following the characteristics of slaves :

- It hears requests from the Master instance.
- Slaves can run on a variety of OS.
- You can configure a project to always run on a particular slave machine.



- Mention three security mechanisms used by Jenkins to authenticate users

**- Mention ways that a job can be run/scheduled in Jenkins**

## -Install Jenkins with docker image

### -Create a new user

Includes all known "users", including login identities which the current security realm can enumerate, as well as people mentioned in commit messages in recorded changelogs.

User ID	Name	Last Commit Activity <sup>1</sup>	On
 amirh	Amir Hossam	N/A	


- Create read role and assign it to the new user

# Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Role-based Authorization Strategy

 Success

Loading plugin extensions

 Success

Authorization

☐ Anyone can do anything ?

☐ Legacy mode ?

☐ Logged-in users can do anything ?

☐ Matrix-based security ?

☐ Project-based Matrix Authorization Strategy ?

☒ Role-Based Strategy ?

Role	Overall				
	Administer	Read	Create	Delete	N
admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Developer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

## Assign Roles Assign Roles

Global roles

User/group		Read	admin		
	 No type prefix: amirh	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
	Anonymous	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Create free style project and link it to private git repo(inside it create directory and create file with "hello world").

Project HelloWorld

Hello World

 [Edit description](#)

[Disable Project](#)



[Workspace](#)



[Recent Changes](#)

Permalinks