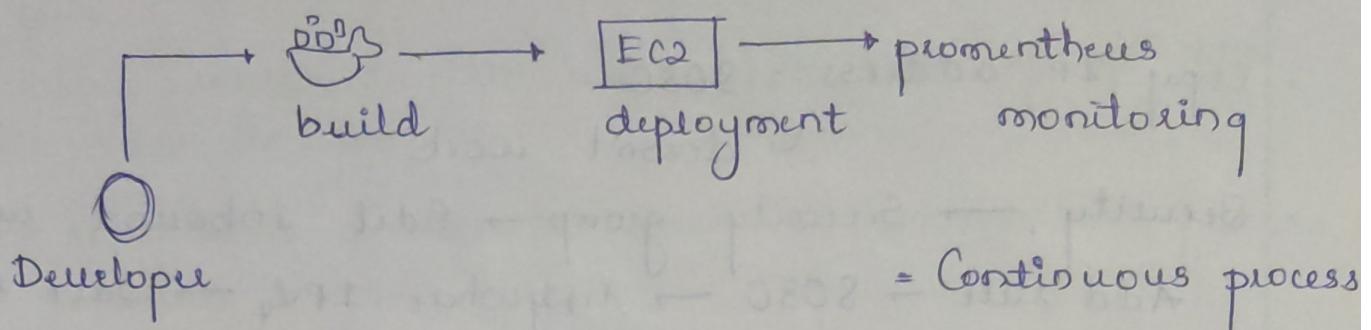


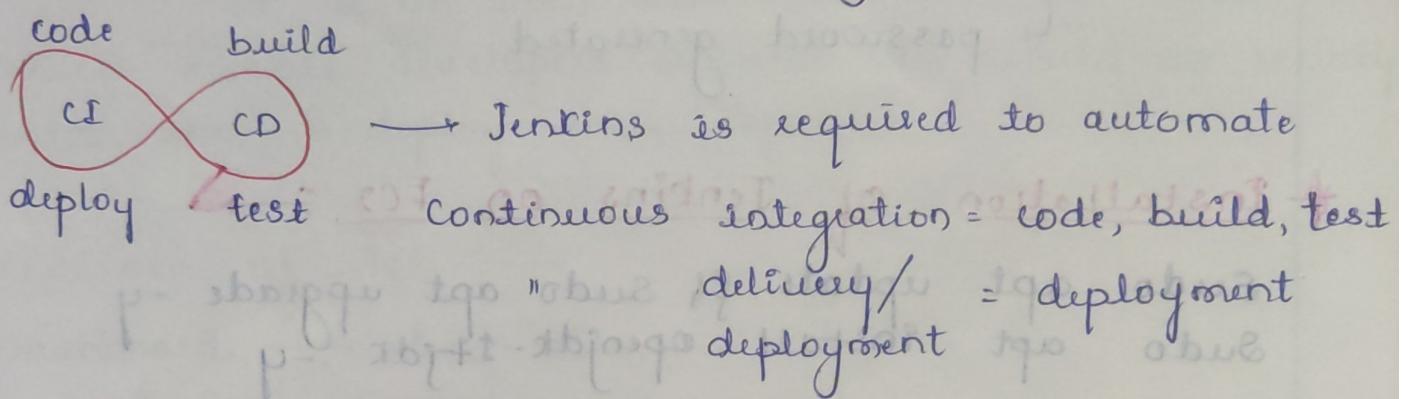
JENKINS

# Jenkins

Jenkins → to automate this process



- To make this process complete in less time we require devops engineer



Ex. Mobile update

- Click the update button = deployment

Jenkins is made of Java programming

- Before downloading Jenkins, download Java

+ [Jenkins.io/doc/book/installing/windows](https://jenkins.io/doc/book/installing/windows)

+ Create EC2 instance  
Connect → terminal

+ Sudo Systemctl enable jenkins  
↳ when system gets start the jenkins also get started.

## Install

- Install by using all the commands = Jenkins download + installation
- Copy IP address : 8080.
  - ↳ doesn't work.
- Security → Security group → Edit inbound rules → Add rule → 8080 → Anywhere IPv4 → Save rules
- Sudo cat paste
  - ↳ password generated

### \* Installation of Jenkins on EC2

Sudo apt update & sudo apt upgrade -y  
Sudo apt install openjdk-11-jdk -y

java -version.

Curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee /usr/share/keyrings/jenkins-keyring.gpg

Sudo systemctl enable jenkins

" " start "

" " status "

copy + : 8080

Sudo cat → paste

Install plugins

Jenkins is a GUI

\* Job: →  
the logo represents the waiter is waiting for the order to do job.

Job = represents a specific task or automated process within a CI/CD pipeline.

\* Jenkins: →

It is an open source automation server which enables developers around the world to reliably build, test & deploy their software.

\* Creation of Job: →

Dashboard → New item

↓ Freestyle project

First job

↓

Build step

echo "Maishrau is brilliant"

Save

↓

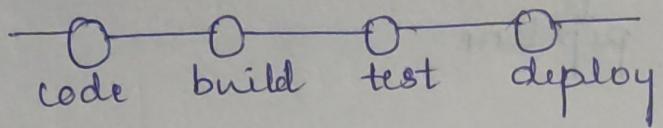
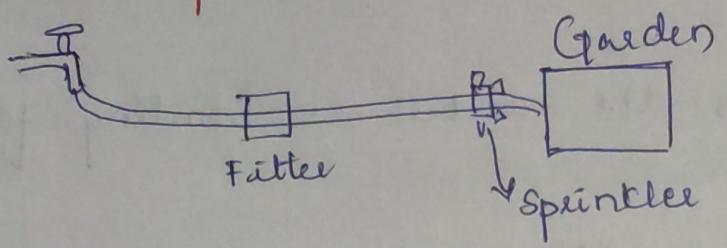
if u write here

mkdir -p devops

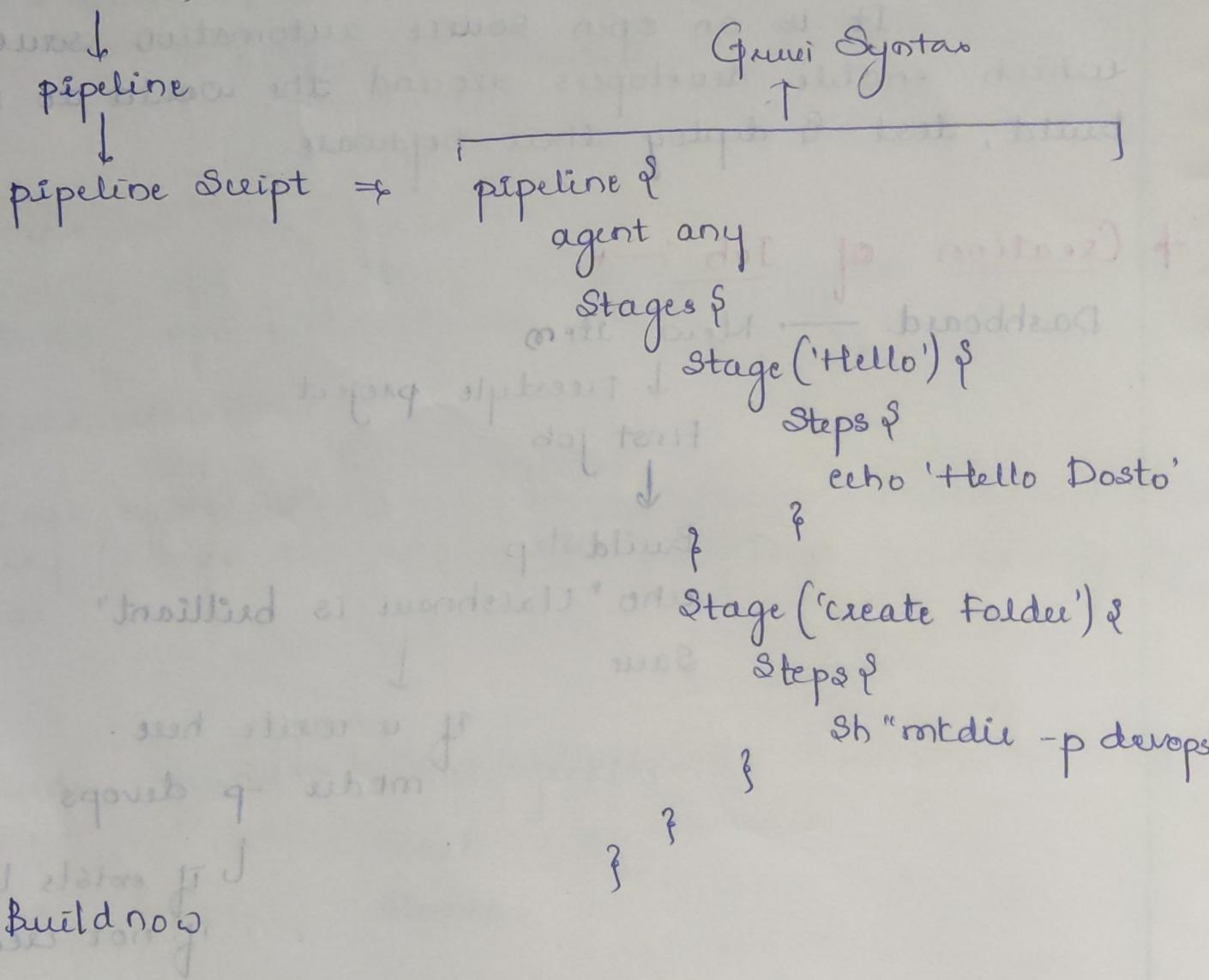
↳ if exists leave

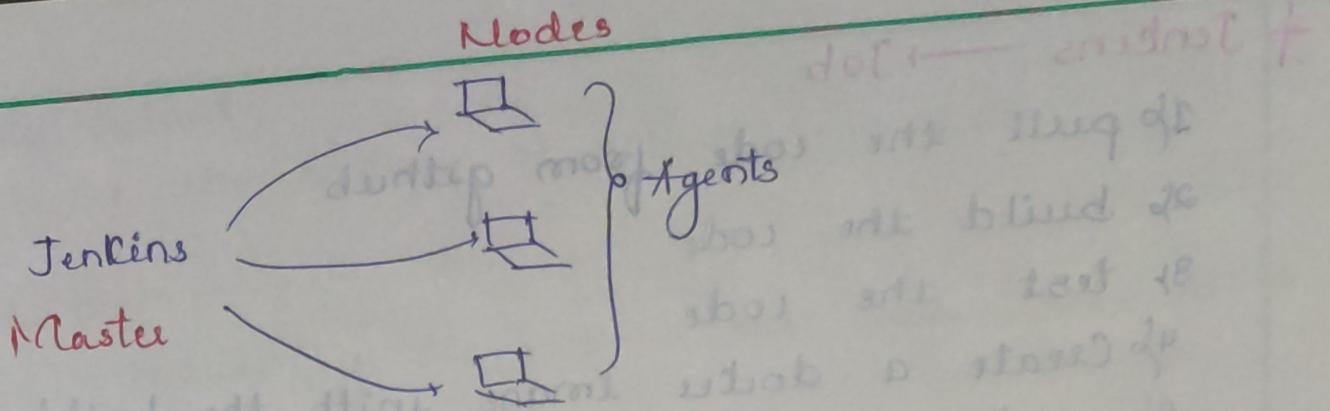
if not create

## \* Declarative pipeline:



New item



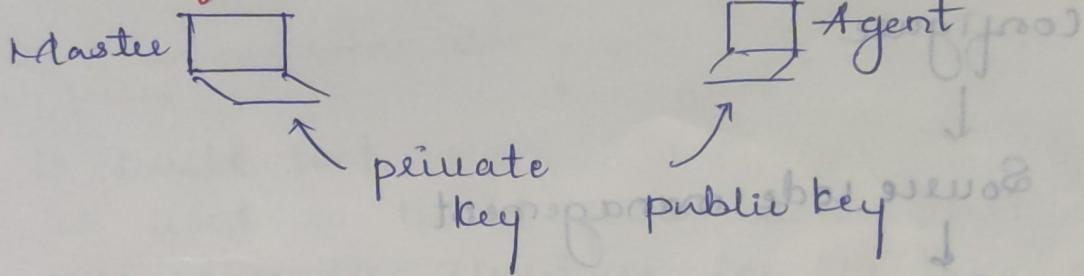


- Jenkins provide the jobs to Agents
- On Agents side no need to download the Jenkins
- On Agents we need java

For one another EC2 instance is created for agent connect

↳ install java

### \* Connection from master to Agent:



+ cd ~/.ssh

↳ ls

ssh-keygen

↳ Enter

↳ ls

↳ authorized\_keys

Master

id-ed25519

↓  
private

id-ed25519.pu  
b

Agent

id-ed25519.pu  
b

↓  
private

id-ed25519.pu  
b

cat id-ed25519.pu  
b

↳ id generates

↳ paste it in Agent

## Jenkins → Job

- 1) pull the code from github
- 2) build the code
- 3) test the code
- 4) Create a docker image with the build artifact
- 5) push the docker image to docker hub
- 6) deploy a container with the docker image

→ New item

↳ Name

↳ Freestyle project

OK.

Click on created job

↓  
Configure

Source code management

Git

C/P repository url

if github repo is private = Credentials

token ↓

username ↓

then " "

dwg.pizzabot Jenkins

↓  
username ↴

password = token

Branch

↳ main

dwg.pizzabot to

estwarp by ↴

target in the step ↴

\* Build

Console output

↳ see the history of all the execution

## \* Build tools

Java = maven, gradle, ant → these are build tools

python = pip

NodeJS = npm

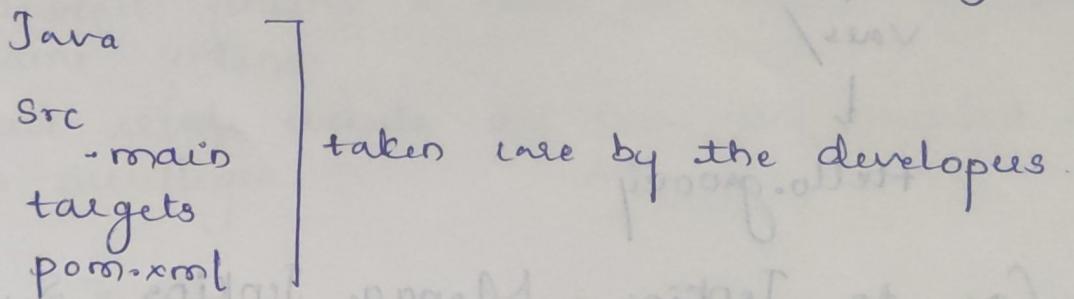
AngularJS/ReactJS/NextJS = npm

PHP = composer

- **Build** = Adding dependency related to the project & compiling the code
- **Artifact** = The output of a build is called a artifact

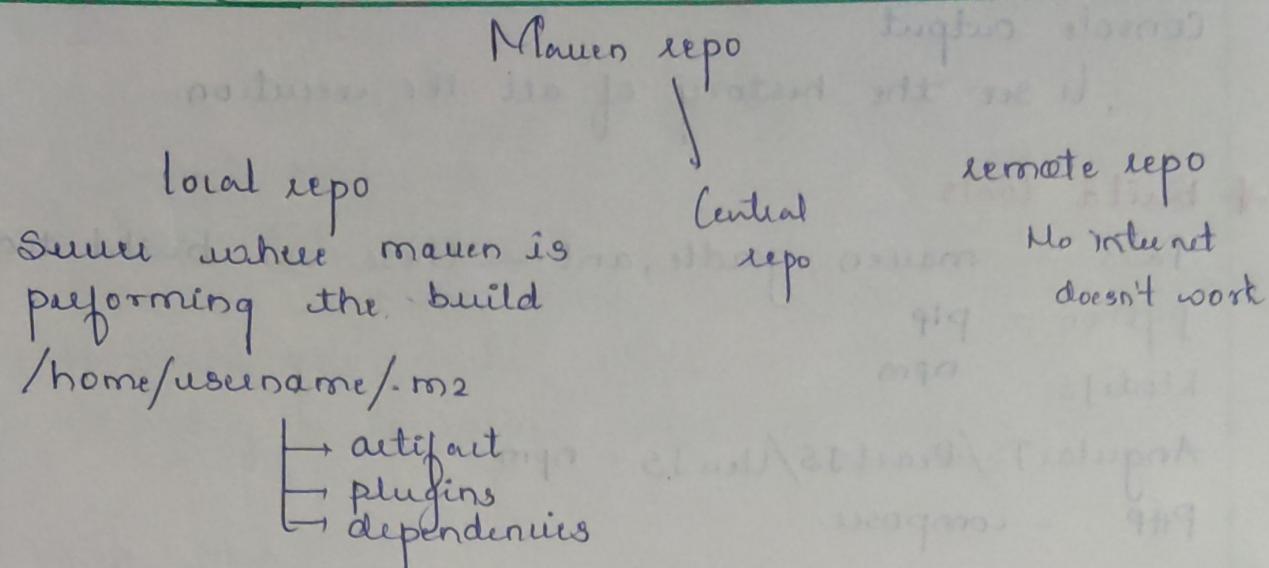
## \* Maven :

1. It is build tool
2. It is used to build java based applications
3. It enforces a dir structure for the java project



## \* Maven phases :

1. Validate → It checks if the project has necessary info or not
2. Compile → In this phase it compiles the code
3. Test → The compiled code is tested
4. package → The artifact is generated jar/war/ear
5. Install → The artifact will be deployed in local maven repo
6. Deploy → The artifact is deployed to remote maven repo



- If we don't get dependencies in local repo then it goes to remote repo.

## \* Shared libraries

groovy code → reuse

Github

↓  
New repo

→  
xars/

↓  
Hello.groovy

Go to Jenkins = Manage Jenkins = System =

Global trusted pipeline libraries

Git

↓  
paste url

open menu

## \* Maven Installation:

Sudo apt install maven -y.

To check version of maven = mvn --version.

maven package

```
└─ validate  
   └─ compile  
   └─ test  
   └─ package  
   └─ install  
   └─ deploy
```

ls

target ls

mvn install

## \* To integrate a Tool with Jenkins:

1. Install that tool on the jenkins Server
2. Install plugin related to that tool in jenkins

## \* Jenkins

1. When we install jenkins a user is added to the by the name jenkins
2. The job we create execute all jobs are executed with jenkins user privileges

A

- Sudo cat /etc/passwd

↳ jenkins user information displays

- who am i = to know which logged in

- Sudo su - jenkins
  - ↳ switch from ubuntu to jenkins
- whoami
- docker ps
- exit
- sudo /etc/group
  - ↳ displays the group
- sudo id jenkins
  - ↳ current user information of jenkins
- sudo usermod jenkins -g docker

- Git pull
- Maven install
- docker build
- " push