

JENKINS

Page No.

Date

" Jenkins To Automation "

- Jenkins is an open source project written in Java that runs on windows, macOS and other Unix like operating system. It is free community supported and might be your first choice tool for CI. Jenkins automated the entire software development lifecycle.

• Jenkins was originally developed by Sun Microsystems in 2004 under the name Hudson.

- The project was later named Jenkins when Oracle bought Microsystems.

• It can run on any major platform without any compatibility issue.

• Whenever a developer writes code, we integrate all that code of all developers at that point of time and we build, test and deliver/deploy to the client. This process is called CI/CD.

• Jenkins helps us to achieve this.

- Because of CI, Now bugs will be reported fast and get rectified fast. So the entire software development happens fast.

“ Workflow of Jenkins ”

- We can attach git, maven, selenium and Artifactory plugins to jenkins.
- once developer puts code in github, jenkins pull that code and send to maven for build
- once build is done, jenkins pull that code and send to selenium for testing.
- Once testing is done, jenkins will pull that code and send to artifactory as per requirement and so on till the task is done.
- we can also deploy with jenkins.

“ Advantages of Jenkins ”

- It has lots of plugins available.
- you can write your own plugins.
- you can use community plugins.
- jenkins is not just a tool. It is a framework i.e. you can do whatever you want. All you need is plugins.
- we can attach slaves(nodes) to jenkins master. It instruct others(slaves) to do job. If slaves are not available, jenkins itself does the job.

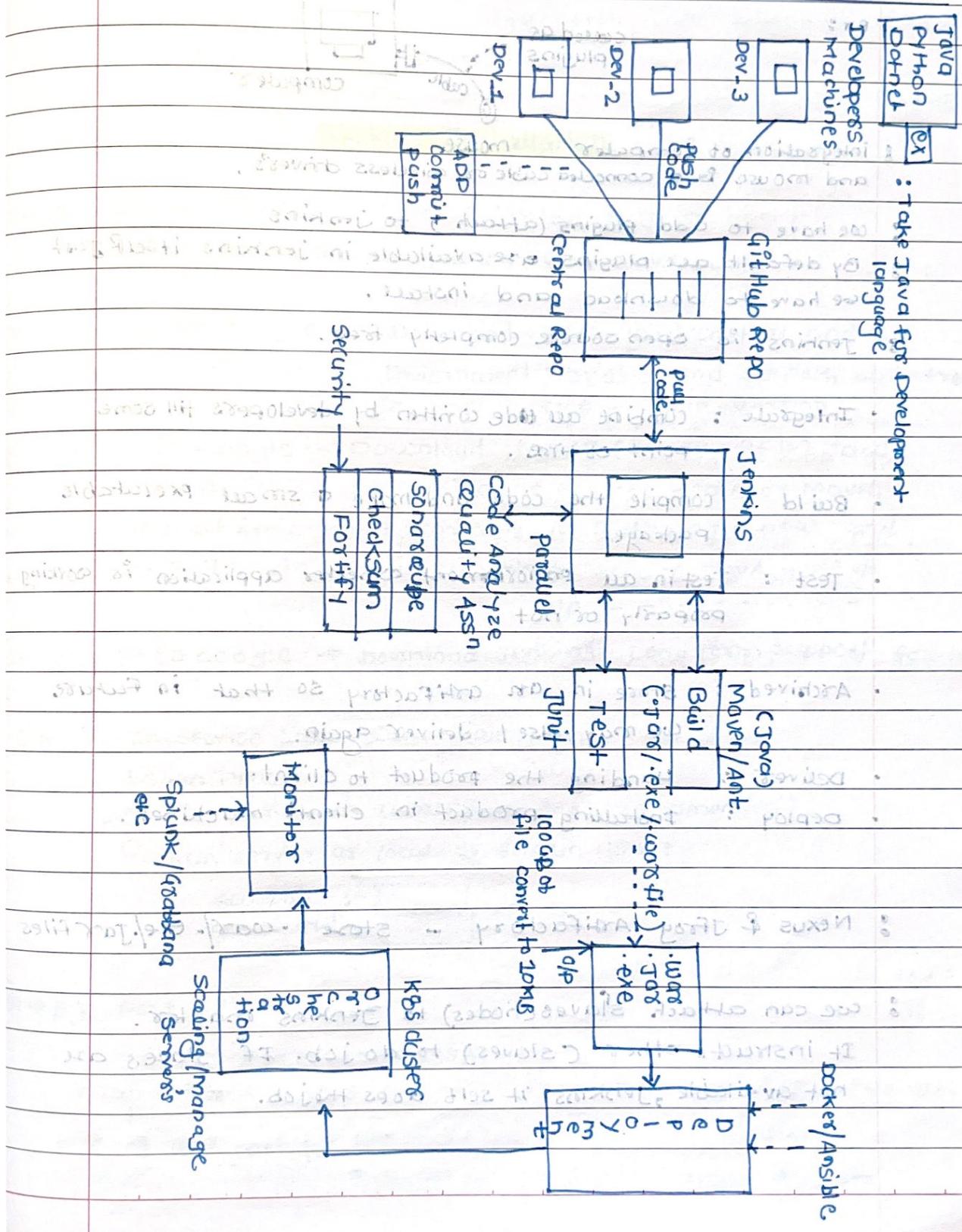
- Jenkins also behave as crone server Replacement i.e. can do scheduled task.
- It can create labels.

Jenkins

Page No.

Date

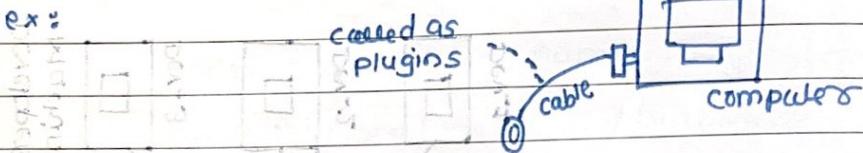
CI - CD - Continuous Integration - Continuous Delivery
+ taking (Deployment) »



* plugin : is acting as integration / mediator component

at (downloading) point.

ex :



: integration of computer and mouse is a connected cable or wireless driver's .

: we have to add plugins (attach) to jenkins

: By default all plugins are available in jenkins itself just we have to download and install .

: jenkins is open source completely free .

- Integrate : combine all code written by developers till some point of time .

- Build : compile the code and make a small executable package .

- Test : Test in an environment whether application is working properly or not .

- Archived : store in an artifactory so that in future we may use / deliver again .

- Deliver : handing the product to client .

- Deploy : installing product in client's machines .

- Nexus & Jfrog Artifactory - store - war/-exe/.jar files

- we can attach slaves(nodes) to Jenkins master .

It instructs other (slaves) to do job . If slaves are not available , Jenkins it self does the job .

Jenkins also act as crone server replacement. i.e. can do repeated task automatically

- Running some scripts sequentially before bedtime (8 AM)
- Eg: Automatic daily alarm.

Jenkins Installation

Step 1) Dependencies we have to install : Java. libraries
And for continuous Integration Git, Maven, etc. (04:47:12)

Step 2) Go To Google - Download Java and Install and set Environment Variable and set path and verify in cmd. cmd = mvn -version

Step 3) Go To Google - Download Maven (binary zip file) and create folder in c drive as ex: M2_HOME or Maven and set Environment Variable in system as a new and Set Path and verify in cmd. echo %JAVA_HOME%
O/P -----

Step 4) Go To google → download Jenkins Long Term support for windows.

Step 5) In service Logon Credentials
Logon Type
 Run service as local system (not recommended)
 Run service as local or domain user:
 Account : []
 password : []

Step 5) Port selection

Port Number (1 - 65535):

8080 If enable port forward at port 8080

Test port

AKASH A. KALE

Step 7) Set Java Location - Top of Java jenkins is installed

Step 8) Finished Installation

Step 9) Go to browser (Any browser)

localhost:8080/login

my comp
ip add.

Step 10) Getting Started

To be unlock jenkins bootstrap - step 9 of 10

first boot time be bmc address terminal

path `/ - / - / - / - 4M3 . 0m . ni`

For Administrator password bootstrap - step 9 of 10

password `graphism` : as the admin password

Step 11) Go to above path copy password and put or paste as administrator password.

Step 12) Customize Jenkins

<input checked="" type="checkbox"/> Recommended	<input type="checkbox"/> Install suggested plugins	<input type="checkbox"/> Select plugins to install

Recommended : Java 11 or 18
 Docker
 Git
 Maven

Step 13) Getting Started

✓ ✓ ✓ ✓

Step 14) If error msg is shown and not proceed

next : java 11 or 18 compatible to the

jenkins is not available in this system.

- Then install Java 11 and set Environment variable and path OR check to installation is forward or not.

Fixed Jenkins Installation problem.

Error

Message: Failed to find compatible Java version (11.0.8/17) in
To install C:\Program Files\Java\Java 11
Jenkins.

(Installation)

Step 15) Set Username, Password & Email for login Jenkins.

IF Above Step's NOT WORKING in your

Machine The follow Below process

For Installation & Login
jenkins

Download .war file from Jenkins download (google.com)

Create folder in dry drive in our machine

ex: C:/Software/jenkins war got it now

ABC wrote to got it 08:08 8 feb 2020

Open CMD on our windows machine.

Run command or go to jenkins folder with the help

of Cd -l -l -l

In jenkins directory run command

java -jar jenkins.war

Automatically load all jenkins war file files of folder

secret key is available here for our administrator password.

put Ap and proceed next step.

put Ap and proceed next step.

After complete loading step open port automatically jenkins localhost:8080

Login (username / password)

If you stop server then run command

↓

control + C

↓

localhost:8080 (Stop / Not Running).

IF relogin then go to jenkins director and

run command `java -jar jenkins.war`

Relogin

Automatically Installed all plugins

b) If you restart your jenkins by

localhost:8080/restart

If you stop your jenkins

localhost:8080/stop or start

oxo

Run command at terminal with this path

if CA - 11-1

To Jenkins directory and run command

java -jar Jenkins.war

No boot was successful

Deployment war file failed to load

See logs for details

for more information about error

↑ After this message press enter

Akash Akale

* Create one sample job in Jenkins

Jenkins Dashboard

New Item

Enter an item name

FirstJob

Freestyle Project

If you don't know what to

Pipeline

Select then choose Freestyle

Multi configuration project

For ex build job

Folder

Github Organization

etc

Dashboard on see in Jenkins Configuration Page

OK

dashboard - FirstJob

Description

General

Source code management

Build Triggers

Build

Post

Environment

Build Action

For Ex: Go to Build section



Add one command ex: echo "Hello"



Apply → save

Automatically

Come back to Jenkins server page

Back to Dashboard

Save and See

Here you can see last build.

All

S W Name Last success Last failure Last duration
watch logo

* For start job we have three options in jenkins
Dashboard bar (execution process)

Project FirstJob

① Click on watch logo

② FirstJob

Build click on

③ Left hand menu we see option

Build Now click on

- When build FirstJob then see on Dashboard Status of this job & other information.

First job successfully executed.
 green colour

* "second job"

New Item

Second Job

Build section

In-Description add wrong command

Save and Apply

Back to Dashboard

Second Job ↗

Delete Job

 Build Now ✓

P.201609.07.001

- ↳ Executing Second Job → build failed at step 00
- ↳ new job → =
- ↳ job Failed (wrong command)

cloud logo
for failed job

Red colour

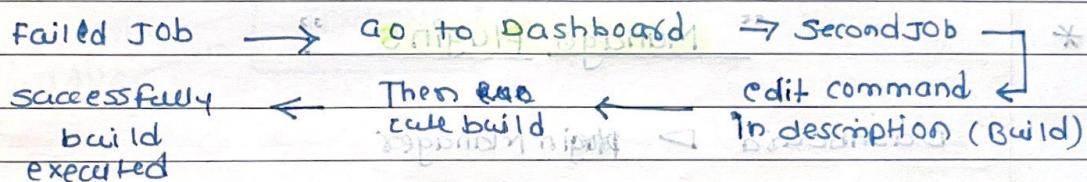
EPIC 2020 : Headless

Feature: New Job + Test Job Help



How To Troubleshoot

*



Dashboard | Raindrop | Job | second job | build 001 of step 001 ↑
end of step 001 ↓

* "I want to create Third Job"

Dashboard ▷ All ▷ build 001
 build 002
 build 003

Third Job

Name: NotanI

→ notanI@notanI.com



→ notanI@notanI.com



If you want to create a new item from other existing,

you can use this option:

notanI@notanI.com

① Copy from

Previous Job

ex: First or Second Job

notanI@notanI.com

OK

→ Apply

→ Save

→ Dashboard

→ Build

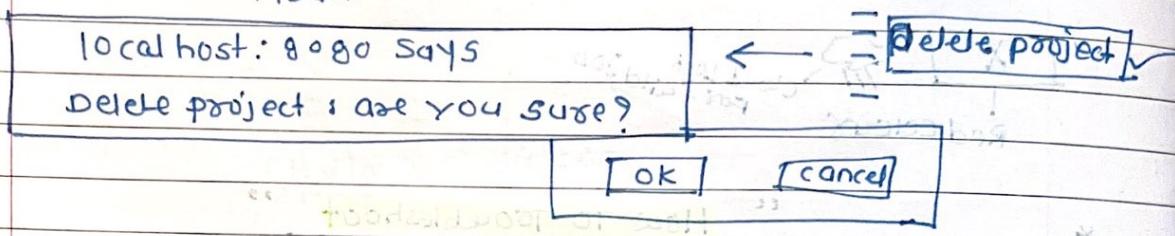
Delete Job

How to Delete ?



Go to Dashboard → Jobs → select job
dropdown ↓

MSG...



Manage Plugins

Dashboard → plugin Manager

↑ Back to Dashboard | Available | Installed | Advanced
Manage Jenkins

Search in Available Plugins: Maven inv

For Example

Available

Install: Name

dot brain



Maven Integration



Maven Invoker

Install without Restart

Download Now and install after restart

Check Now

Update information obtained: 1 day to obtain
1 hour ago

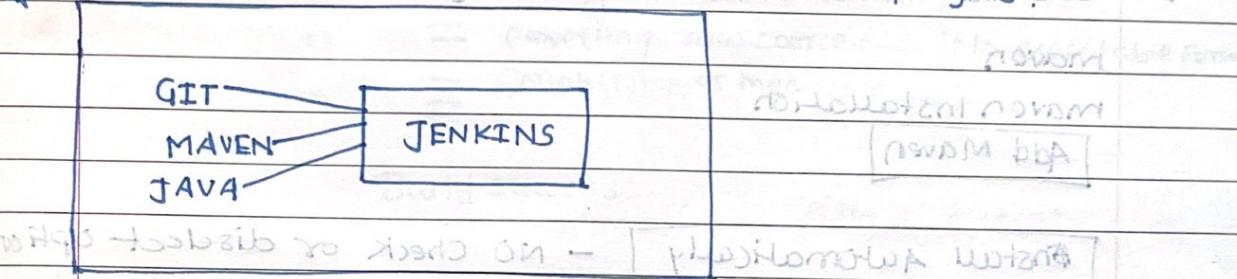
→ Installing Plugins / Upgrades

Dependancies ✓ Plugins downloads from internet
 Software ✓ and install itself by jenkins.

→ When installing plugins jenkins restart automatically.

→ After installing plugins we see on dashboard particular this plugin as a project.

Ex: Maven Project — because we installed maven plugins.



For Jenkins we have download & installed this s/w as a prerequisites.

Dashboard: Global Tool Configuration

JDK

JDK Installations

Add JDK

By default (Install automatically) ✓

↑ if not install Java in our machine

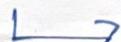
↓ if install already then disable or uncheck this option



JDK

Name

JAVA



JAVA_HOME

Path :

- Git

Git Installations

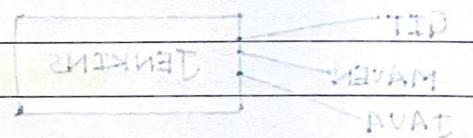
- git

{ Small size that why no need to do anything

- we have installed Maven Plugins to display maven

maven Installation

Add Maven



Install automatically - NO Check or deselect option

Name : Maven

Maven

Maven-Home

Path -

Add Maven

Apply

Save

* Plugins are integration points

" Manual Build / Night Time Build "

Page No. _____
Date _____

- In CMD Windows we clone one repo ex: xyz.repo
 - Goto this repo directory using cd -/xyz.repo
 - In this repo we will see all files & folders
- * • C:\repo\name> mvn clean package

↑ Manual Build using Maven

Used some Maven command.

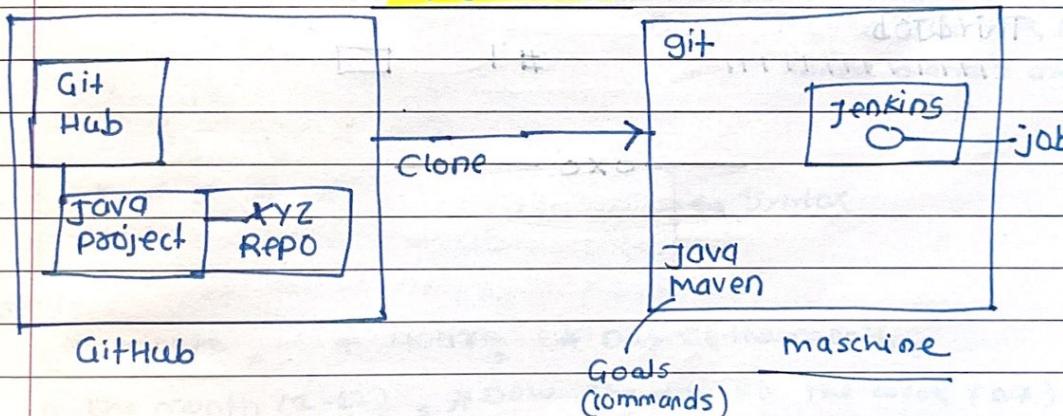
|| Doing the build Activity

|| Converting raw source code into executable format

|| (NightTime or Manual Build) logic

Build Success

- * " Same Execution process we will do with the help of Jenkins "



New Item

[This job] > configuration page



[Maven project]

↳ Source Code Management

Git

Repository

Resositor URL

copy from github
as a clone and
past here

Build *

Root POM - In cmd window we can see this command

POM.XML - This file defines the project structure

Goals and Options / command - These are used to build the project

Clean package - Maven automatically understand our project

Save Apply

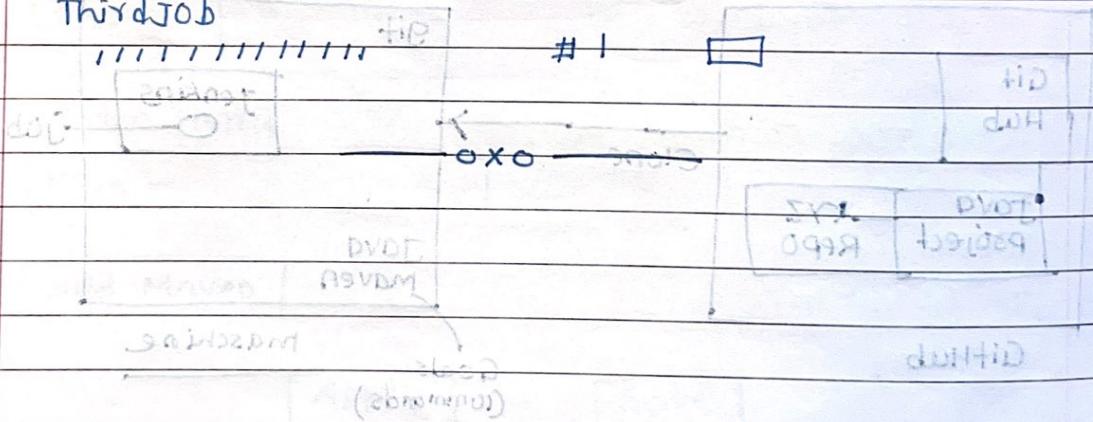
Third job - automatically pull code from GitHub

repository - it will take help from Maven (Java)

execute build

Build Executor status success

Third Job



Configure options page - dotnet

[Kotlin project]

source code management

git

repository URL

Akash A. Kale

" Scheduled Projects "

Run a job →

↓
dotted

Repeatedly (continuously)

1) Build Periodically

2) Poll SCM (Source code Management)

↳ (Pull)

+ →

Crontab / cronjob / cronool (Linux concept)

↓

Jenkins Dashboard

↓
Job (code repository, notification, etc.)

↓
configure → Build periodically

↓
Build Triggers

Minutes Minute (0-59)

Hour Hour (0-23)

DOM Day of the Month

MONTH The month (1-12)

DOW The day of the week (0-7)

Where 0 and 7 are Sunday.

Schedule

* * * * *

↳ Syntax

Represent

* minute, * hour, * day of the month,

* the month (1-12), * DOW The day of the week (0-7)

↓ Subtasks

Save ✓

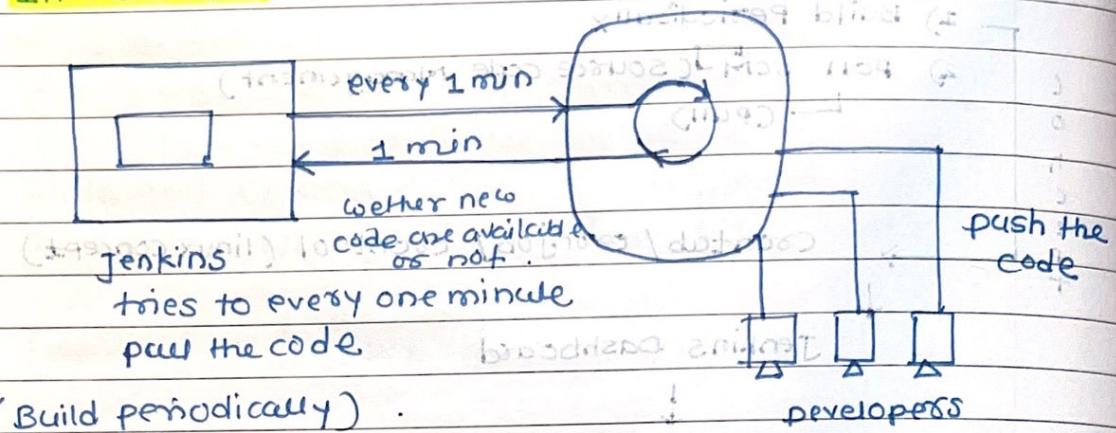
APPLY ✓

O/P: Automatically build jobs and

- If you want to execute any job frequent interval you just mention the time. (Build periodically).

* In Maven Job

Git Hub



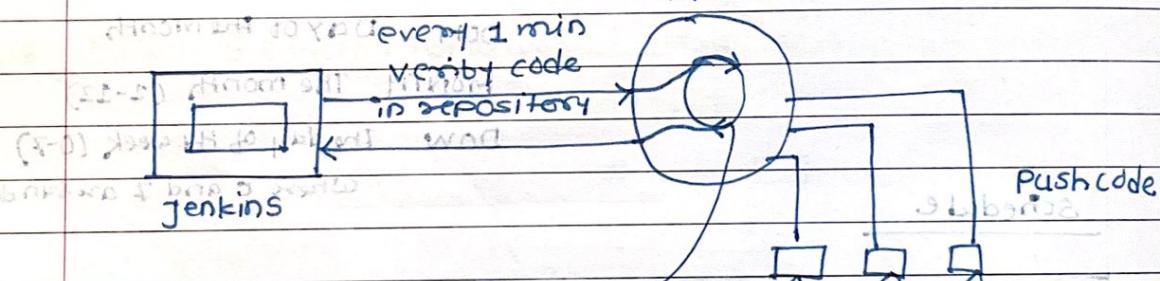
(Build periodically)

Time mention (schedule box)

* Poll SCM

Build Trigger → Poll SCM

GITHUB

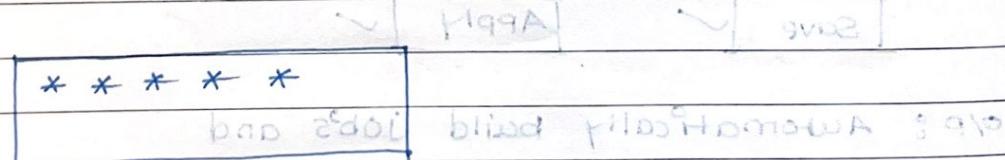


Note: If at all new

Changes are available that time will be execute the job.

Otherwise it will not do anything.

Schedule



Save

Apply

for example of this how to apply presubmit build

Maven Job, scheduled Task, poll SCM

Go to google chrome → local host: 8080 → login → now

Go to Manage Jenkins on left side of Jenkins dashboard →

Manage Plugins → Available → select Maven Integration & Greenballs

→ Install without Restart

Now go to New Item → Maven Project

Now go to Manage Jenkins → Global tool configuration goto Add JDK

uncheck this install Automatically option

Name → JAVA → choose → click on Add → Java

JAVA-HOME → c:\programfiles\Java\Jdk-blind

Now go to Maven

Name → Maven

MAVEN-HOME → C:\DevTools\apache-maven

MAVEN PROJECT (By Name)maven

go to https://github.com/fethub/account-name / Repository-Name

click on zip (repository code)

fork → to copy this zip to Jenkins home directory of your own

sign-in into your github account

click on (repo-name)

clone

go to C drive

git clone <url of repo-name> & add file at your own

cd repo-name

C:\repo-name> mvn clean package

Maven project (by Jenkins)

- ① Now go to Jenkins → New Item → Enter name → My Maven project
- ② Then select Maven project → OK
- ③ source code management → GitHub → Repository URL → github
- ④ Build option → Root POM → pom.xml → Goals & option → clean package
- ⑤ Go to Jenkins home page → click on My Maven project → Build now.

Scheduled project

- ① Click on any project → configure → Build → triggers → build → periodically
- ② * * * * * → save
- ③ can see automated builds after every 1 minute.
- ④ You can manually trigger build as well.

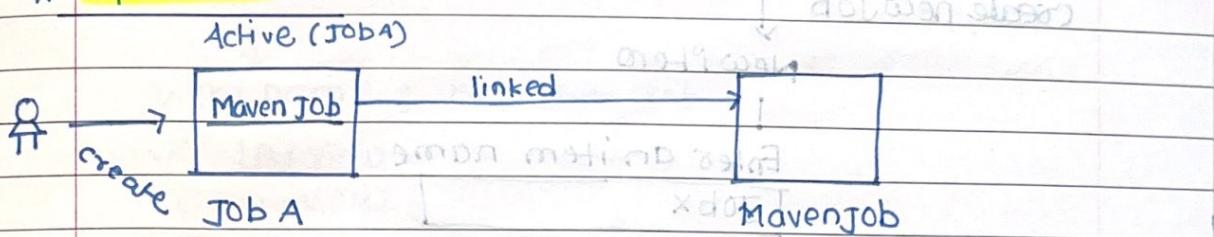
source code polling (Poll SCM)

- ① Now go to Jenkins Home page
- ② go to My Maven project → configure
- ③ Now go to Build trigger
- ④ Poll SCM
- ⑤ scheduled → save
- ⑥ Now go to GitHub account → do some changes in README.md → commit changes.
- ⑦ you can see, after 1 min, it builds automatically.

- Code(new) commit in present Repo then automatically SCM is working means pull the code from GIT Repo, and build the code.

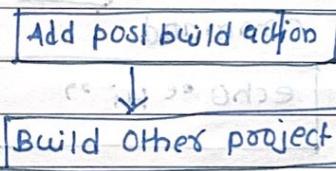
Linked / Related Jobs

* Upstream



- Assume job A is already in our project
- Maven Job already exist in our project

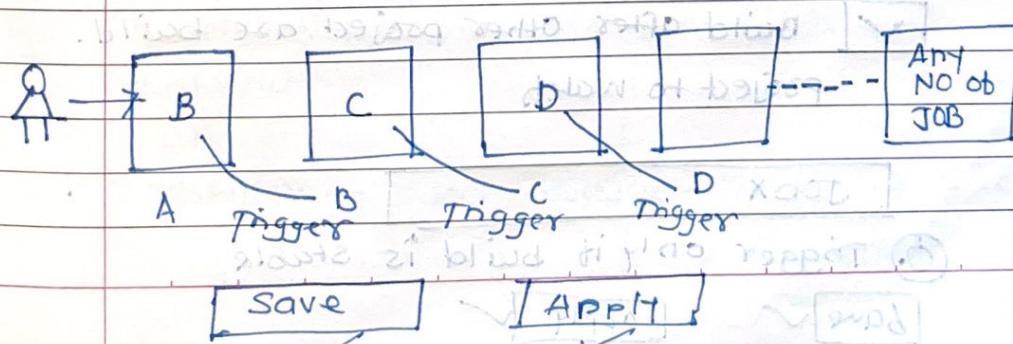
Go to Dashboard → post-build Action



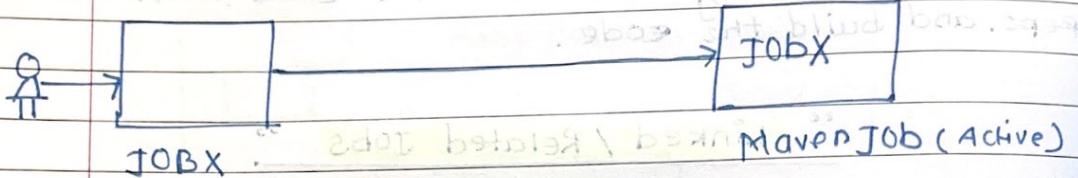
project to Build

Mavenjob

- Trigger only if build is stable
- Trigger when if the build is unstable (warning msg)
- Trigger even if the build fails



* Downstream



Create new job

New Item

Enter an item name

JobX

OK

echo ee Hi "

Save ✓ or Apply ✓

Inside of this MavenJob I would like to give JobX name

Go to MavenJob

Build Triggers

Build after other project are build.

-- project to watch

JOBX

① Trigger only if build is stable

Save ✓

Apply ✓

Jobx execute & Maven Job executes itself.

"Views"

Jenkins Dashboard

New View

↓
view name : FreestyleJob

- List View
- My View

OK

Name

FreestyleJob

Description : Example Job

Jobfilters

2 } Select those you want

3

etc

OK

Save **Apply**

- we can create another job

MavenJob

List View

OK

Jobfilter → Select MavenJob → OK → Save → Apply.

Jobfilter based

- If you want to delete view
Select view → Right Left Hand → Delete view.
→ yes.

"Jenkins User Administration"

Create users → Go to Manage Users

username:	<input checked="" type="text"/> ok
password:	<input type="text"/>
confirm password:	<input type="text"/> now
Full Name:	<input type="text"/> dotnet2027
E-mail Address:	<input type="text"/> dev@dotnet2027.com

- I would like to login as new user in administrator login.
- By default logged in users can see everything like Admin users.
- Install plugins - Available

Role-based Authorization strategy

install without restart

restart jenkins

- we have install plugins as a administrator
- After install role-based Authorization strategy we can see in dashboard

Configurable global security

⑥ Role-Based strategy APPLY OK

We try to login as another user as newuser.

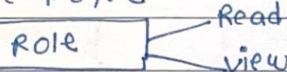
In dashboard we will see Access Denied.

(-) newuser is missing the overall / Read permission.

- * o I will give limited permission to user

Global Role

employee



← I will attach newuser to this employee role.

- * Project Specific Role

Create Role

Developer

→ Full access for developer job

Ex: At every project, I want to have developer role & tester role.

Tester

- full access for test related job

This Project

- I will try to attach developer role to developer user's.
- I will try to attach tester role to tester user's.

Go to Manage Jenkins

↓
Security

↓
Manage and Assign Roles

↓
Manage Roles

◦ Role to add

employee → and give to permission only read & view.

◦ Item Role



Role to add

Developer

pattern

Dev.*

→ Assign full permission.

Item Role

↓ find scope see what can be done in add

Role to add (what can be done in add) → Manage →

Tester

PATTERN

Test, *

↓ find scope what can be done in add → Manage →

Give full permission ✓

Save

APPLY

Now I am going to attach user's to Developer Role and Tester Role.

Manage and Assign Roles

User/group to add

Employee ABC and XYZ users attached to employee role.

Item Roles



user/group	developer	tester
Anonymous		
ABC	✓	
XYZ		✓

Apply

OK

- * Create new job as a DevProject → Add build step
Execute windows batch command
command

echo ee Hi''

Save

Apply

- * Create new job as a TestProject → Add build step
Execute windows batch command
command

echo ee Hello''

Save

Apply

- * As an administrator we can see all function & job in jenkins dashboard.

- * When assign role and attach user then see function and job only devrelated and testrelated project.

How To Install Jenkins on Linux

Install Jenkins on Linux system

Prerequisites

EC2 with port 80:80 open

Install Java on EC2

yum -y install java-1.8.0-openjdk (any version)

java -version

— java.home (Path) copy & Paste to terminal

— Setting envn path.

PATH=\$PATH:\$JAVA_HOME

— source ~/.bash-profile

yum -y install wget

wget -o (jenkins)

yum -y install jenkins

Start Jenkins

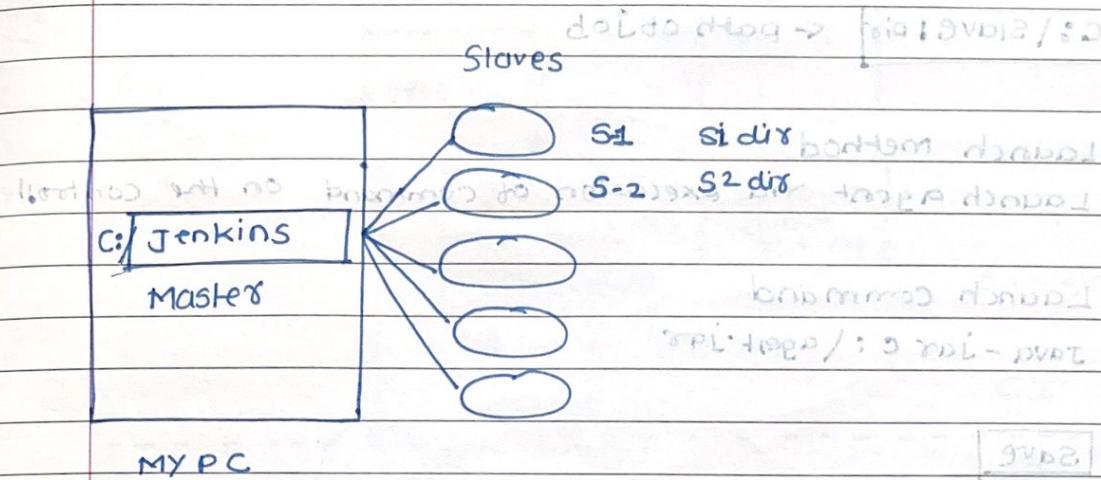
Systemctl start jenkins / service jenkins start

Systemctl enable jenkins / chkconfig jenkins on

(http://your-server-public-IP:8080) in browser.

EC2 - Public IP copy : 8080 ✓ in browser

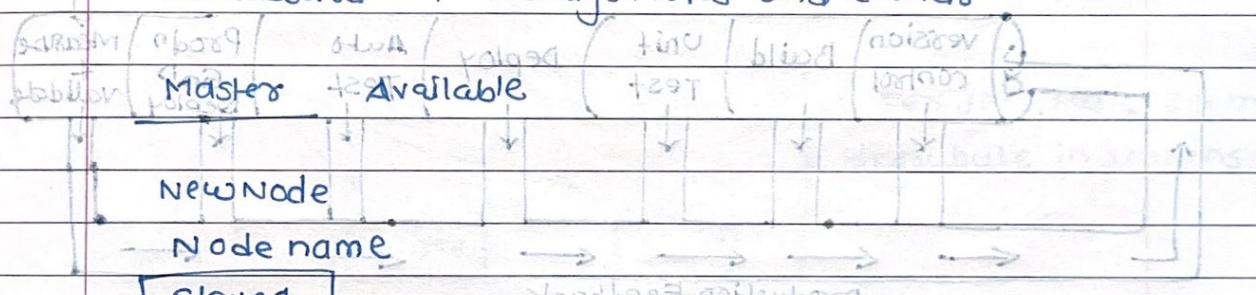
Master And Slave Configuration



* All those job creating only master and executing

• upstream and downstream disable

Dashboard → Manage Nodes and clouds



④ permanent Agent ~~Demand~~

OK	✓	Configuration Details	Description	Number of executor	Test	minimize time
1		I	change as per our req?			

Remote root directory

C:\Slave\Dir ← path of job

Launch method

Launch agent via execution of command on the controller

Launch command

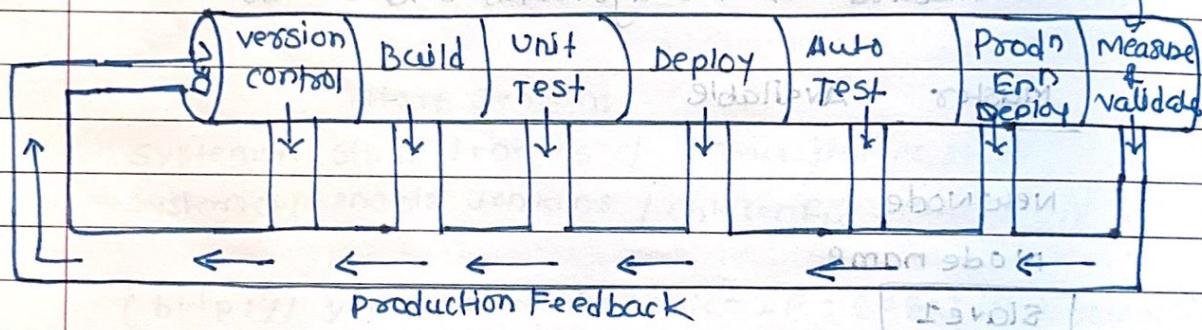
java -jar c:\agent.jar

Save

*

Jenkins - opensource

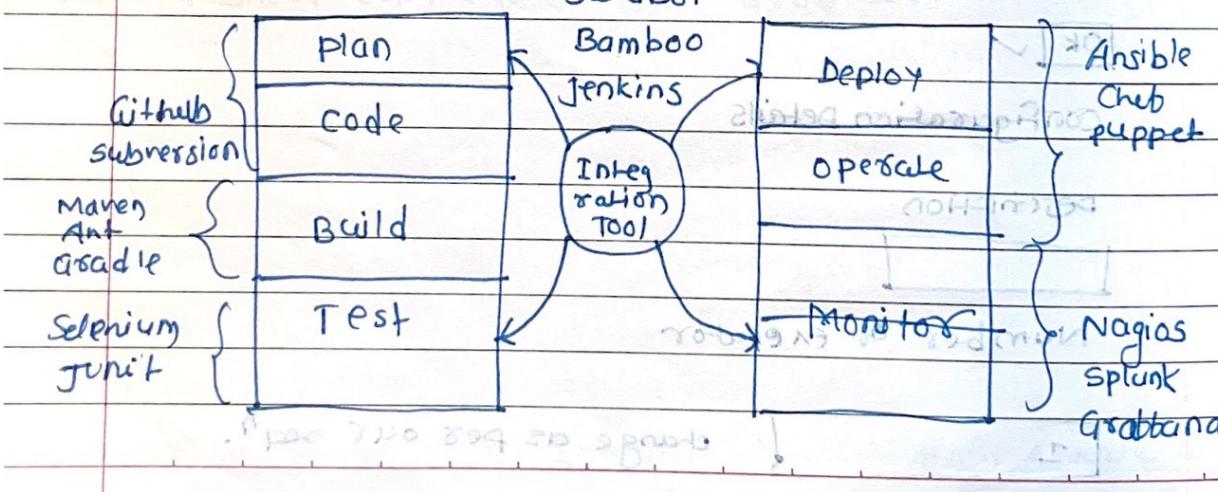
"CI-CD pipeline"

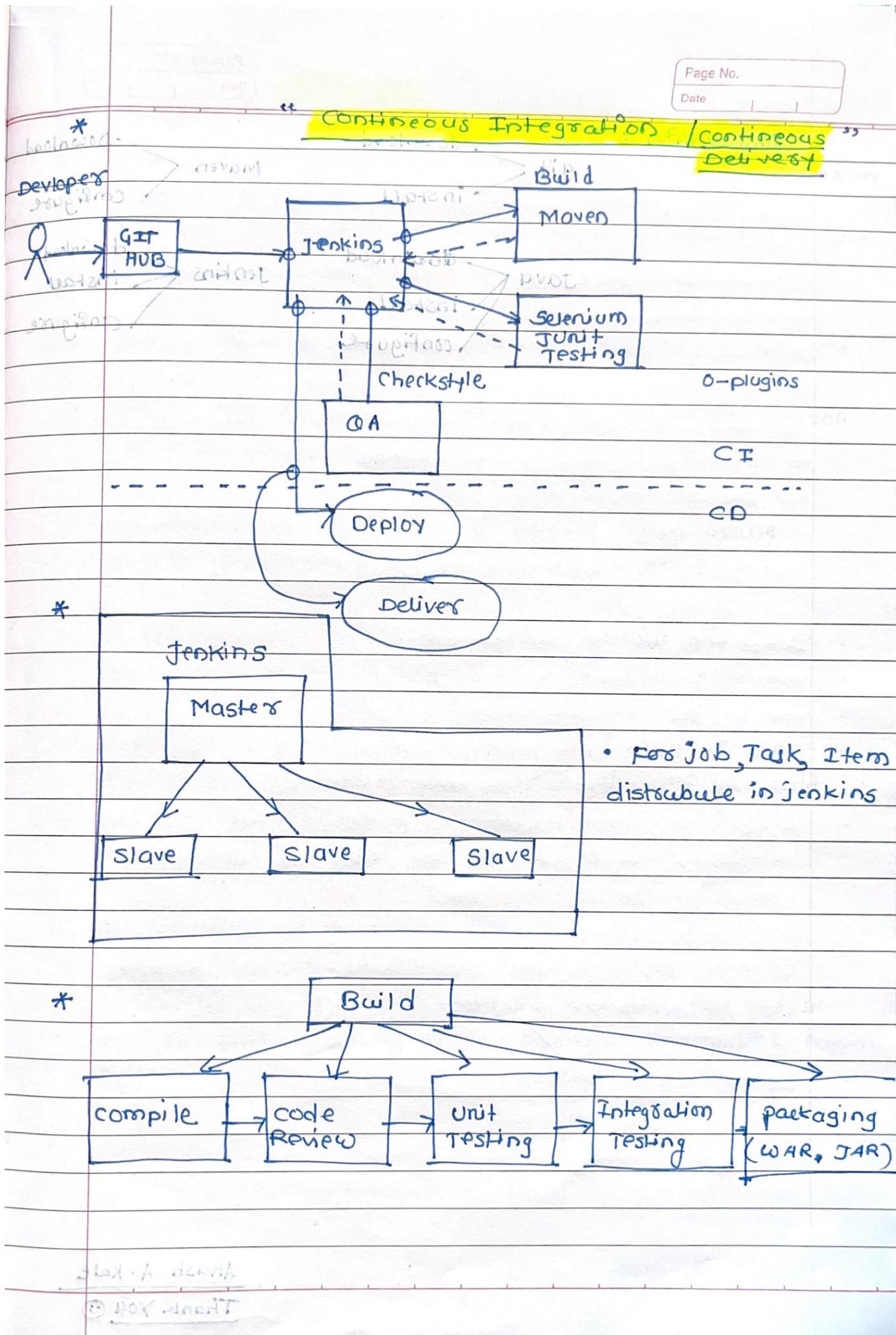


*

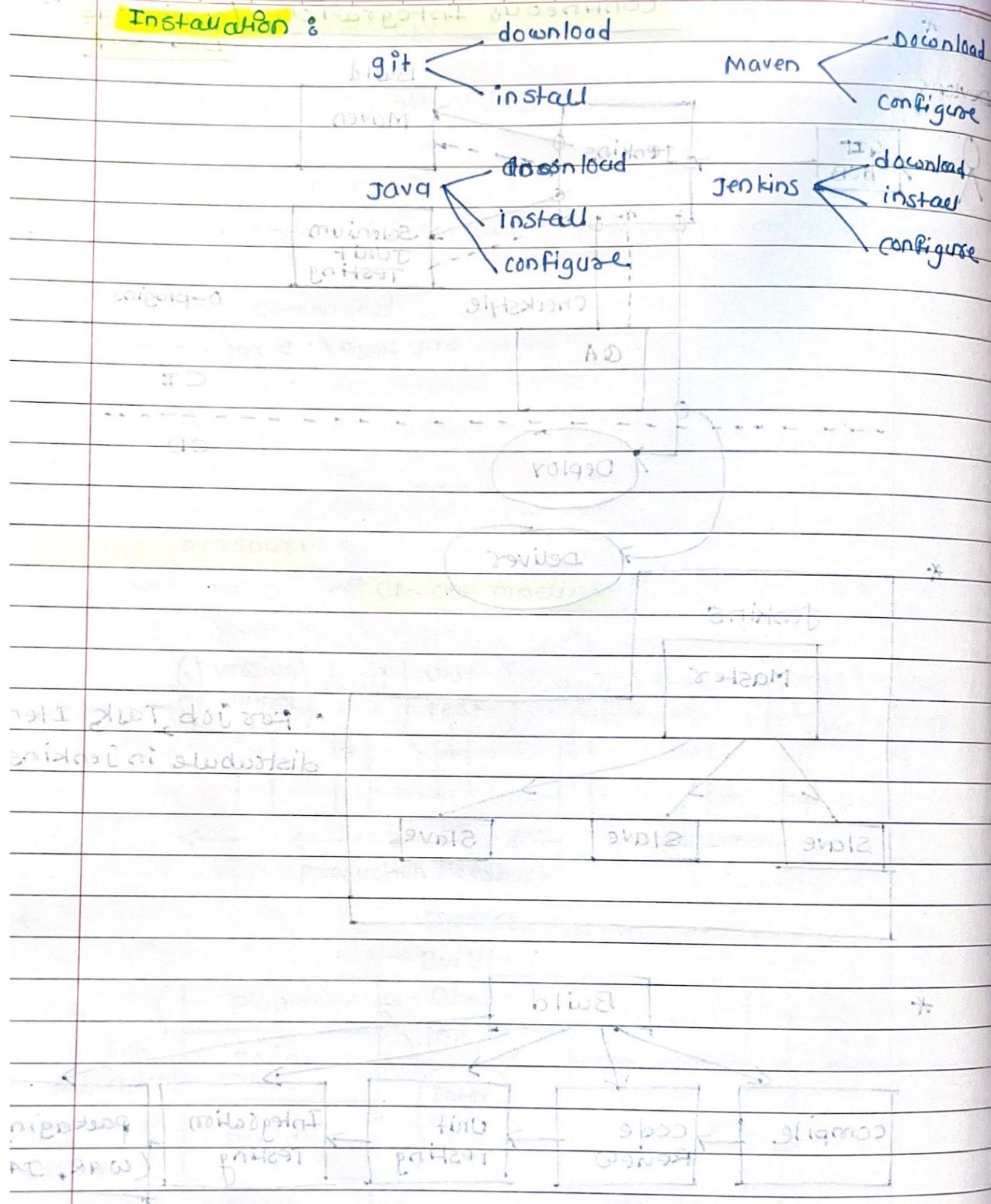
Travis CI

Buildbot





Installation :



AKash A. Kale

Thank You 😊