Devops Session — 3

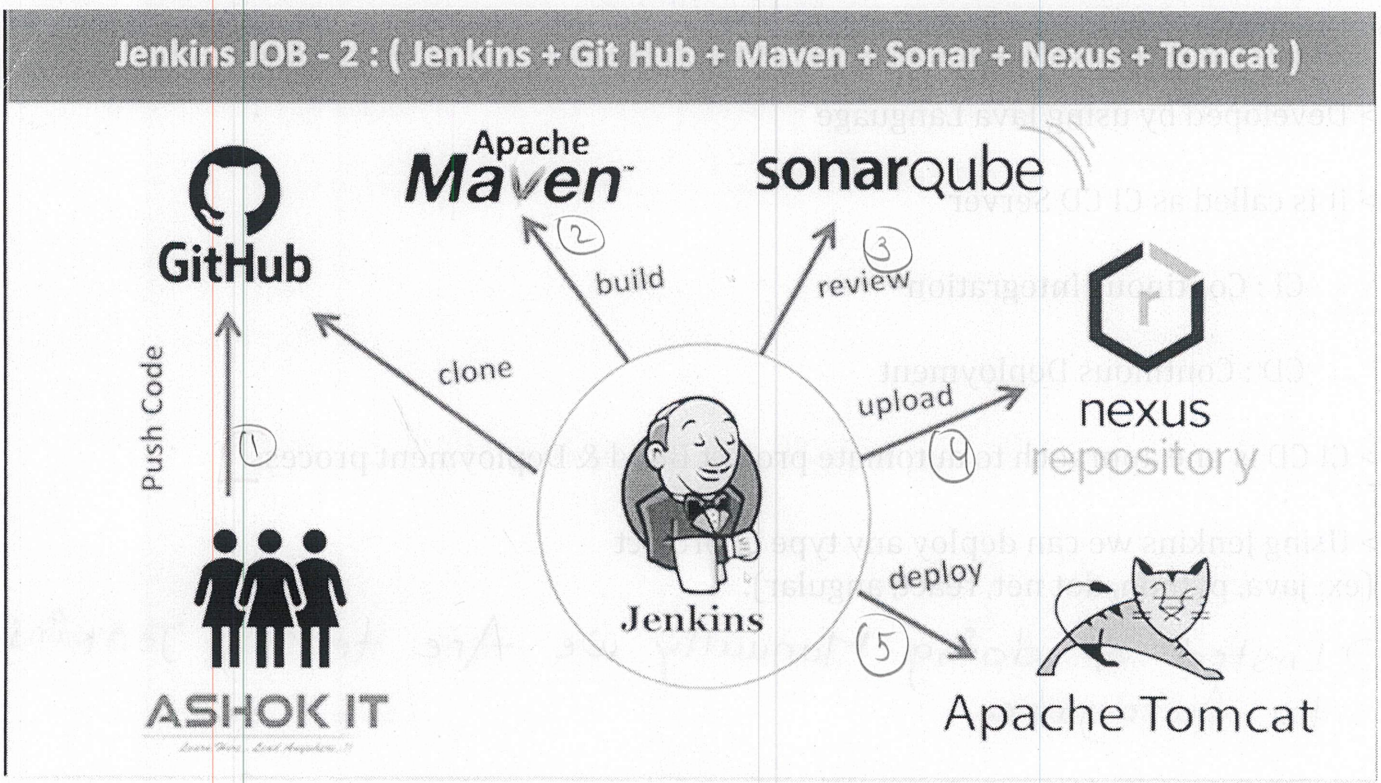
Jenkins
Alternate Argocd / github Actions

13-DevOps -28-OCT-24

==========

What is Build & Deployment

==========



=> Take latest code from Git Hub Repo

=> Build Source code using Maven

=> Perform Code Review Using Sonar

=> Upload Project Artifact into Nexus

=> Deploy code into server.

=> In single day multipe times code will be committed to git hub repository from Development team so multiple times we have to perform build and deployment process.

Note: If we do build and deployment process manually then it is time taking process and error prone.

=> To overcome above problems, we need to automate Project Build and Deployment process.

=> To automate project build and deployment process we will use "JENKINS".

Jenkins

JENKINS

=> Open source Software & free of cost

=> Developed by using Java Language

=> It is called as CI CD Server

CI : Continous Integration

CD : Continous Deployment

=> CI CD is one appraoch to automate project Build & Deployment process.

=> Using Jenkins we can deploy any type of project
(ex: java, python, dot net, react, angular).

=> Insted of doing Manually we Are telling Jenkins to Automate.

Jenkins Setup

Git Repo : https://github.com/ashokitschool/DevOps-Documents/blob/main/01-Jenkins-Server-Setup.md

↳ Refer Document for Next Process

Administrator Passwd

↳ Sudo cat /var/lib/jenkins/secrets/initialAdminPasswd

==========

what is job in jenkins ?

==========

=> JOB means set of steps that we are giving to jenkins to perform the task

Step-1 : Take code from git repo

Step-2 : Perform maven build

Step-3 : Perform code review using sonar

Step-4 : Upload artifact into nexus

Step-5 : Deploy war file into tomcat server

==========

Jenkins Job with with GIT Hub Repo + Maven - Integeration

==========

Step-1 : Configure Maven as gloabl tool in Jenkins

(Jenkins Dashboard -> Manage Jenkins --> Global Tools Configuration -> Add maven)

name : Maven-3.9.9

Step-2 : Create Jenkins job with "free style project"

-> Select New item & Enter Job Name
-> Select Free Style project
-> Goto "source code mgmt" tab and select git
-> Configure project git repo url and branch name
(https://github.com/ashokitschool/maven-web-app.git)
-> Goto "build step" and select "Invoke Top level maven targets"
-> Select Maven version we configured as global tool
-> Enter maven goals => clean pacakge
-> Click on 'Apply & Save'

Note: With above steps we have created JENKINS Job

Step-3 : Run Jenkins Job with "Build Now" option

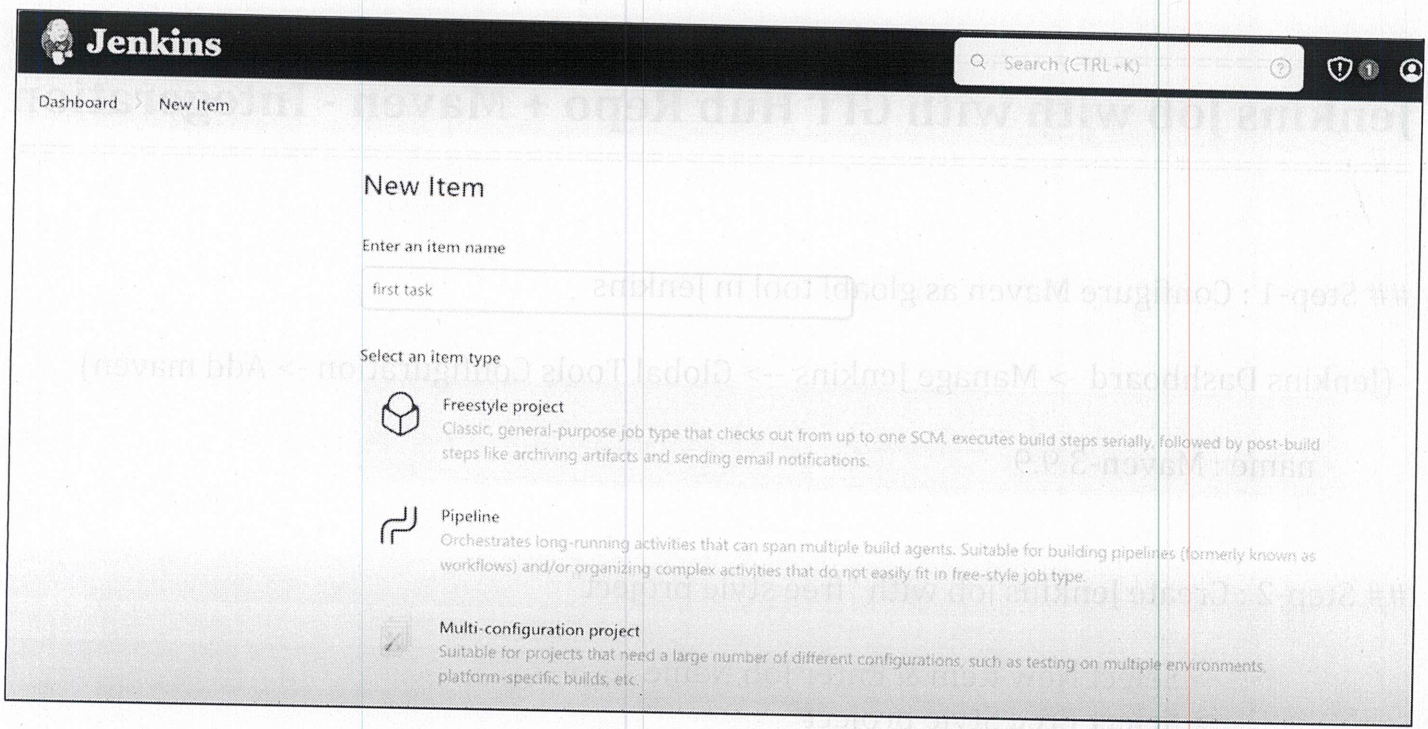
Step-4 : Click on 'Build Number' and then click on 'Console Ouput' to see job execution details.

=> Jenkins Home Directory in EC2 : /var/lib/jenkins/workspace/

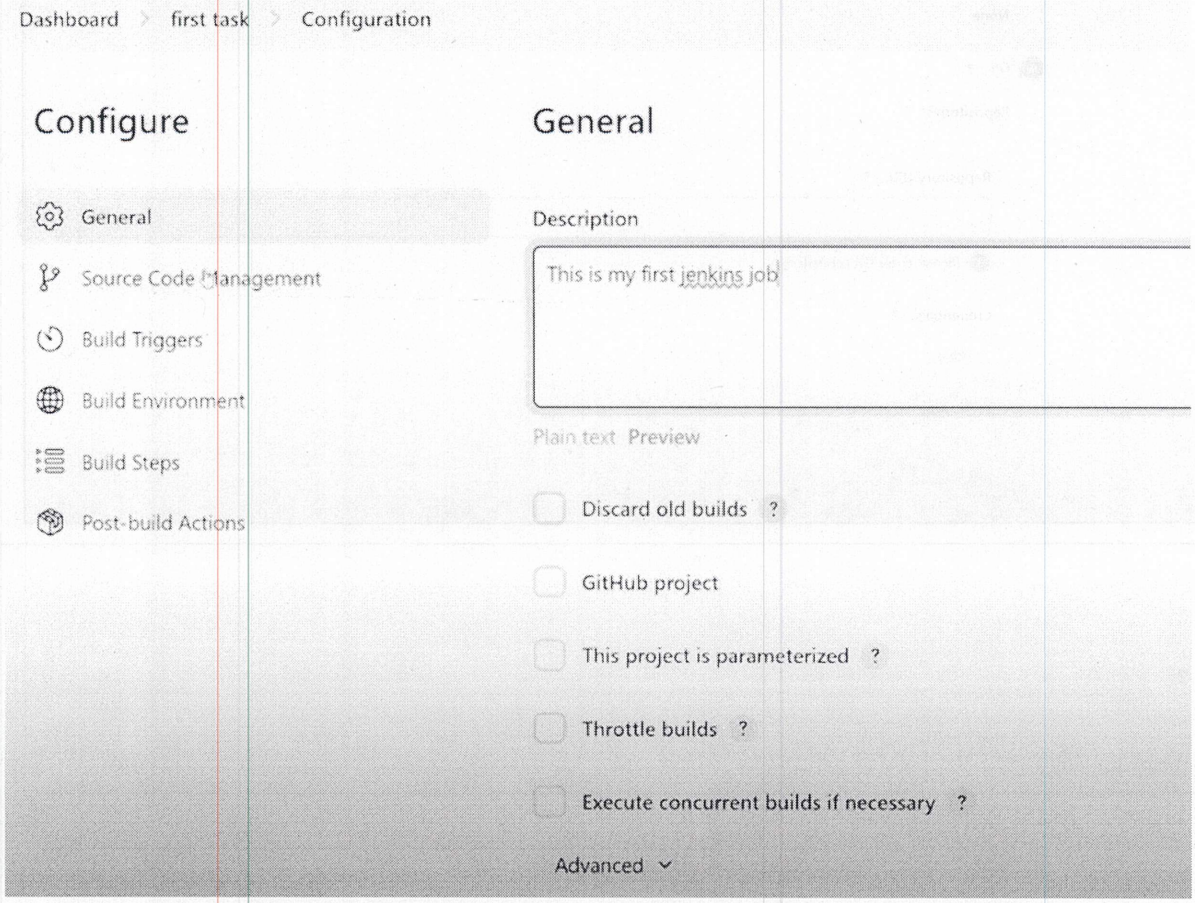
=> Go to jenkins workspace and then go to job folder then go to target folder there we see jar/war file created.

==========

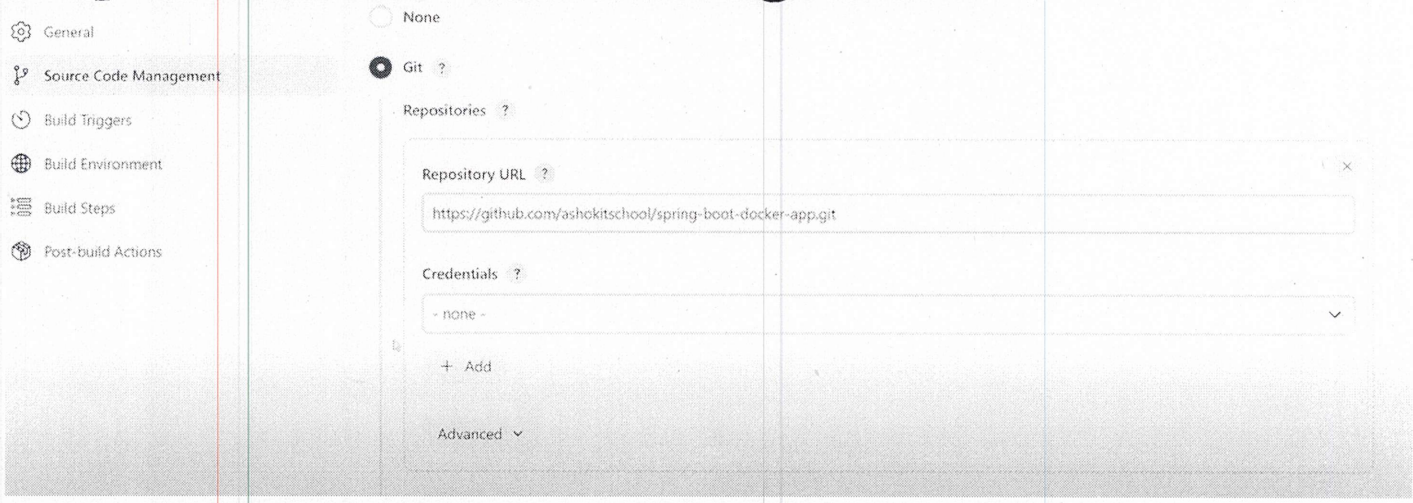
Creating job in Jenkins

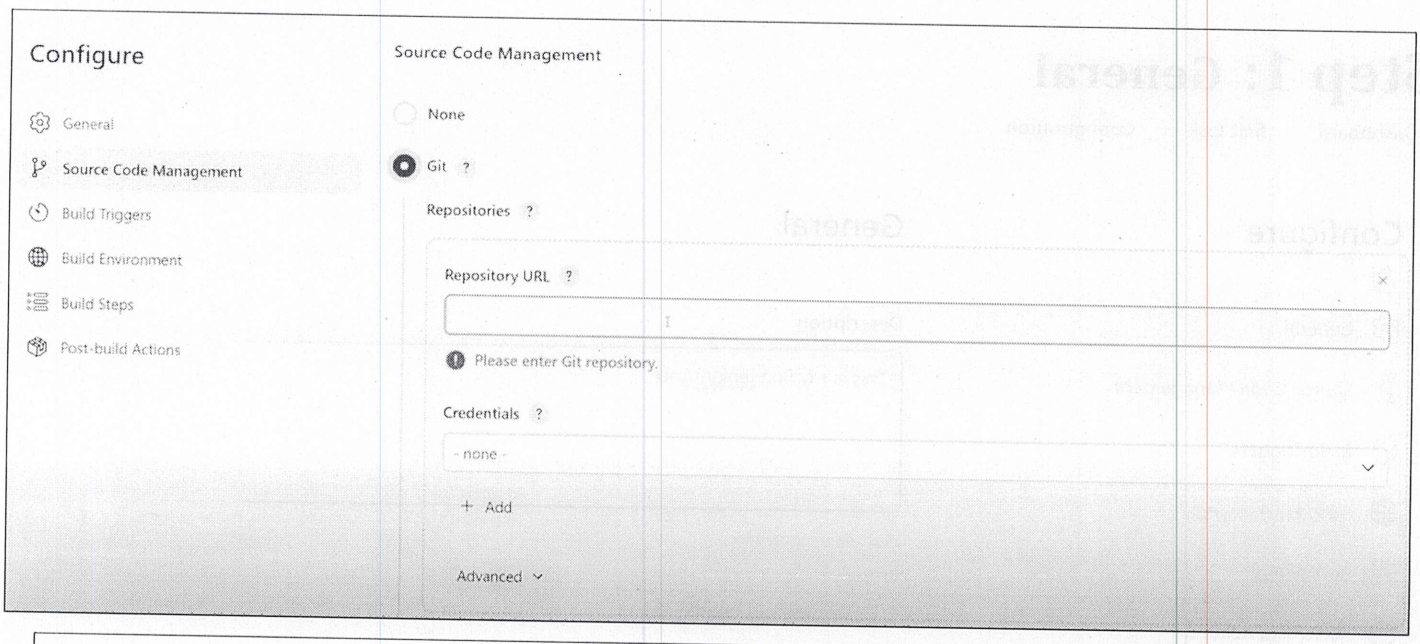


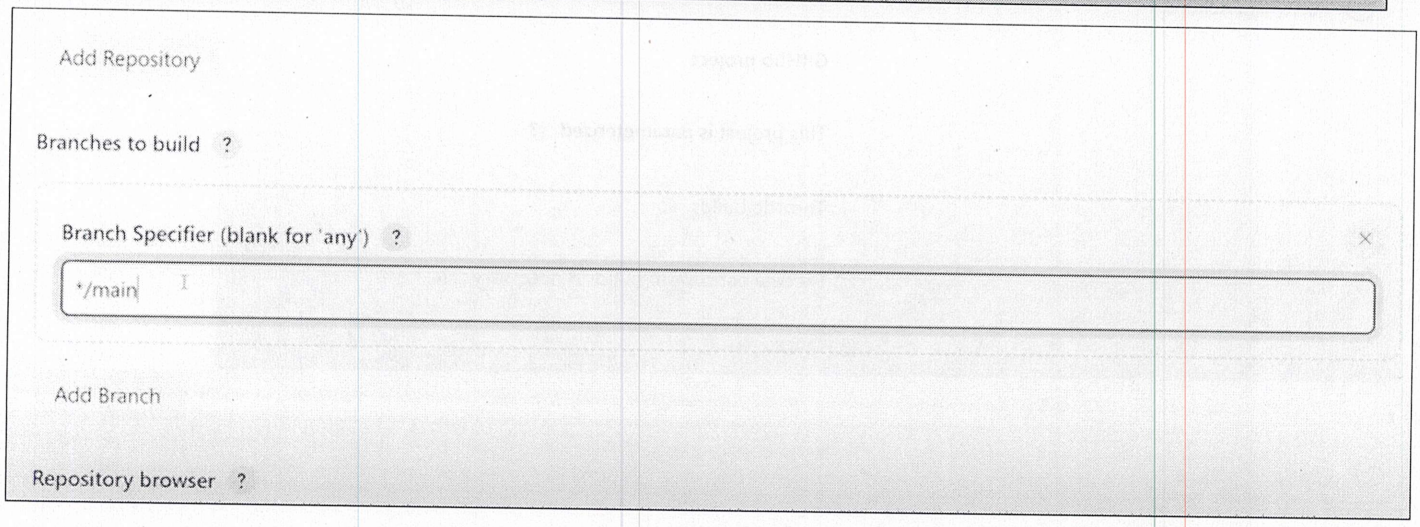
Step 1: General



Step 2 : Source code management

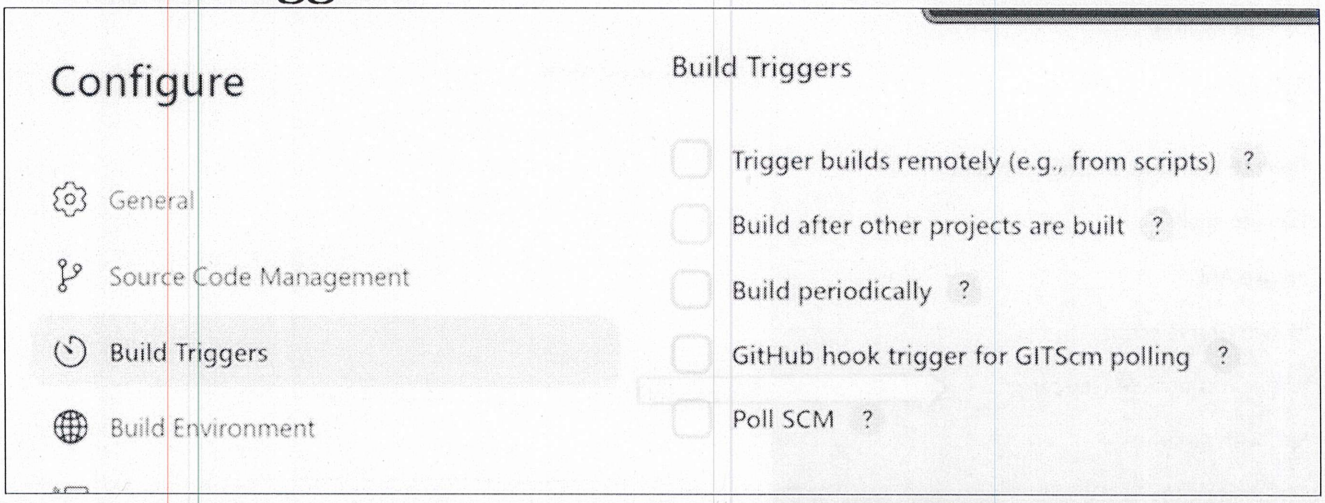






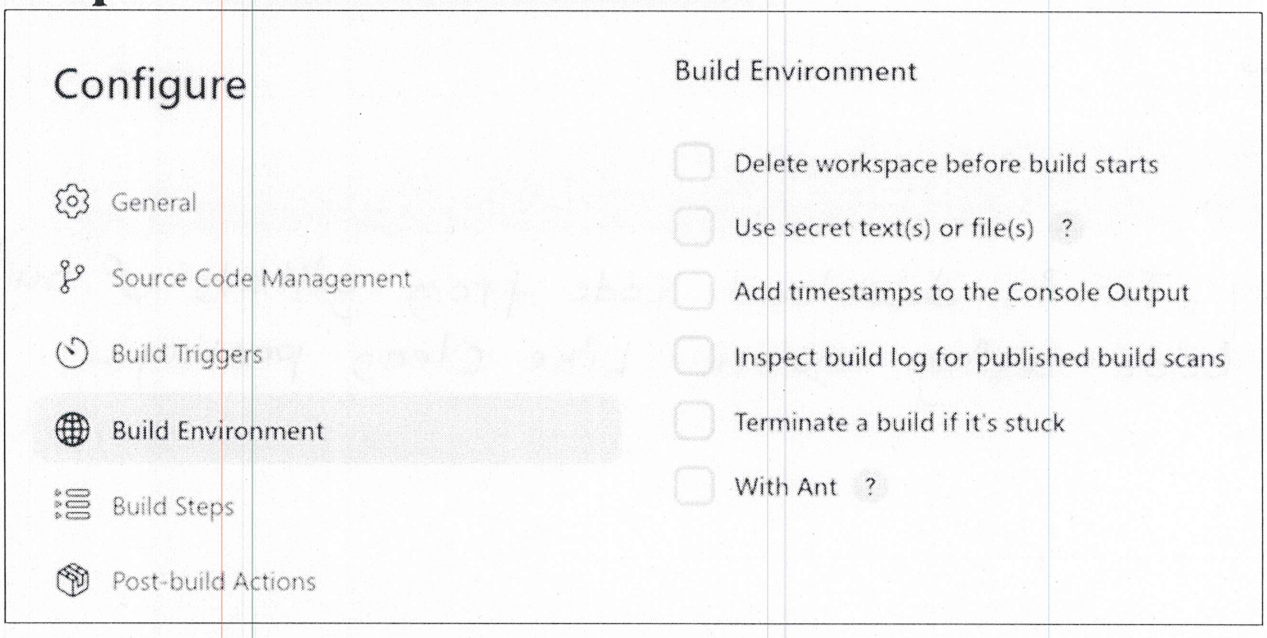
- we can clone git-hub repository , if github repository is private repo then we have to configure account credentials also .

3 – Build Triggers



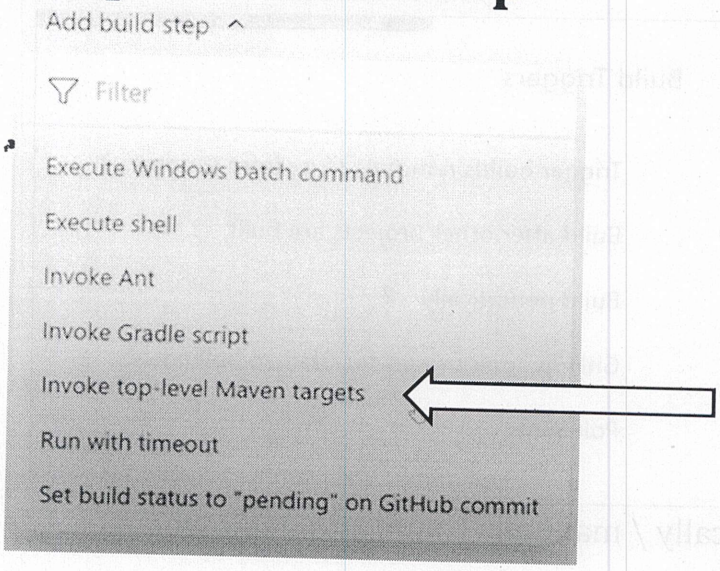
- When we want to run our job , automatically / manually (build and deployment)
- Do you want to run our on specific time
- Do you want to run our job whenever there is code change in github that is done by using build triggers.
- Build triggers is basically when our job is executed.

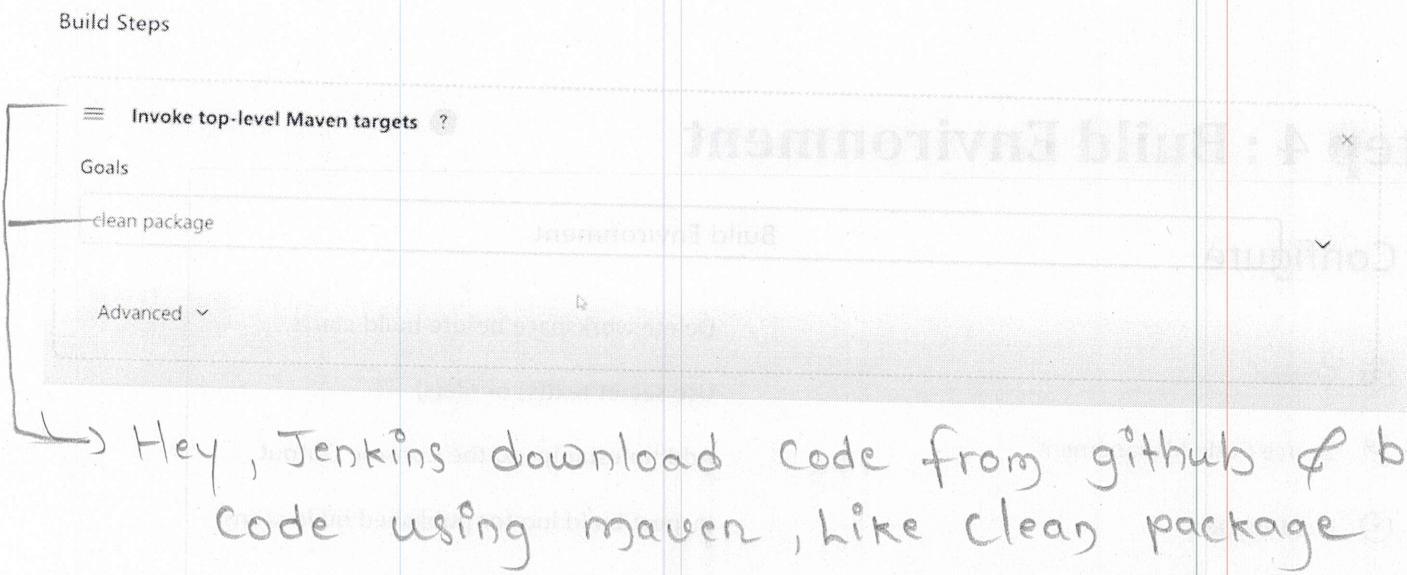
Step 4 : Build Environment



- read above properties that we need to configure according to our needs.

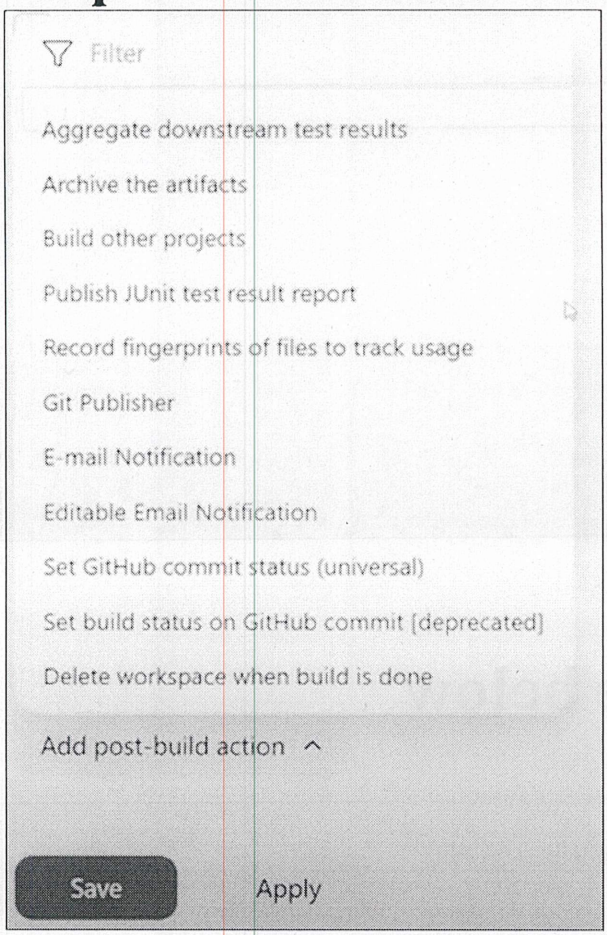
Step 5 : Build Step





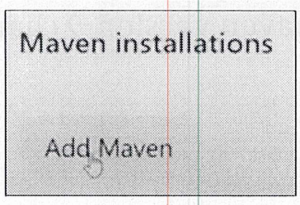
Hey, Jenkins download code from github & build code using maven, like clean package

Step 6 : Post-build Actions

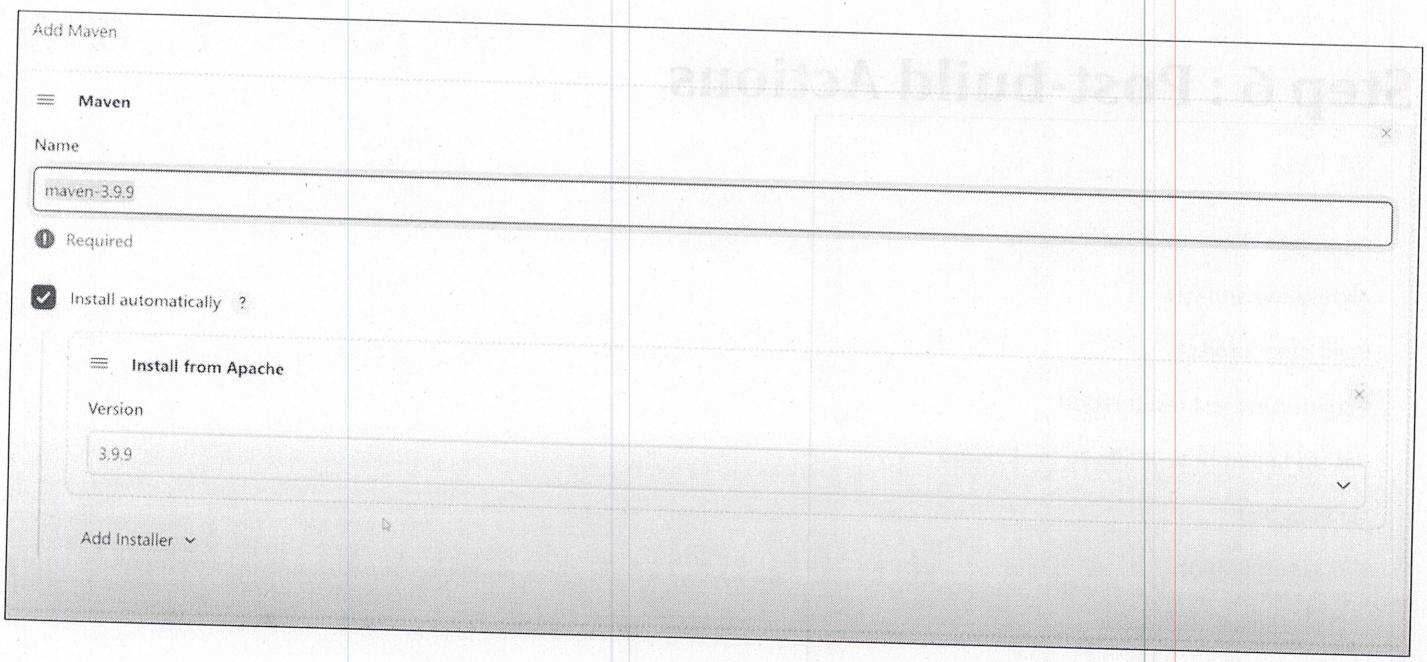


Add Maven

- If build is failed and maven is not available then click on →manage Jenkins →tools



-



Issue-If You got an issue like below

```
Commit message: "Create Dockerfile-New"
 > git rev-list --no-walk 6adc84273bfaebe746c0fce58a142cdc509d23f9 # timeout=10
[Job 1] $ mvn clean package
FATAL: command execution failed
java.io.IOException: error=2, No such file or directory
        at java.base/java.lang.ProcessImpl.forkAndExec(Native Method)
        at java.base/java.lang.ProcessImpl.<init>(ProcessImpl.java:314)
        at java.base/java.lang.ProcessImpl.start(ProcessImpl.java:244)
        at java.base/java.lang.ProcessBuilder.start(ProcessBuilder.java:1110)
Caused: java.io.IOException: Cannot run program "mvn" (in directory "/var/lib/jenkins/workspace/Job 1"): error=2, No
such file or directory
```

Ans- go to the manage jenkis →tools → add maven → add maven version save and apply

Go to dashboard → job 1 →configure → go to build step →change maven version→change default to the latest maven version →save & apply

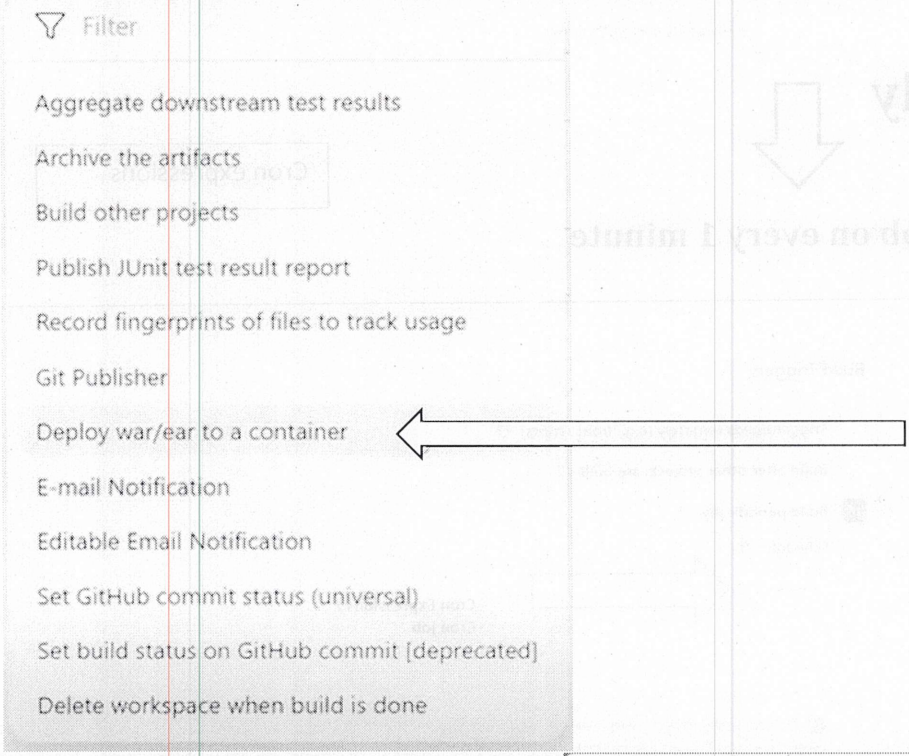
- Our build get success.

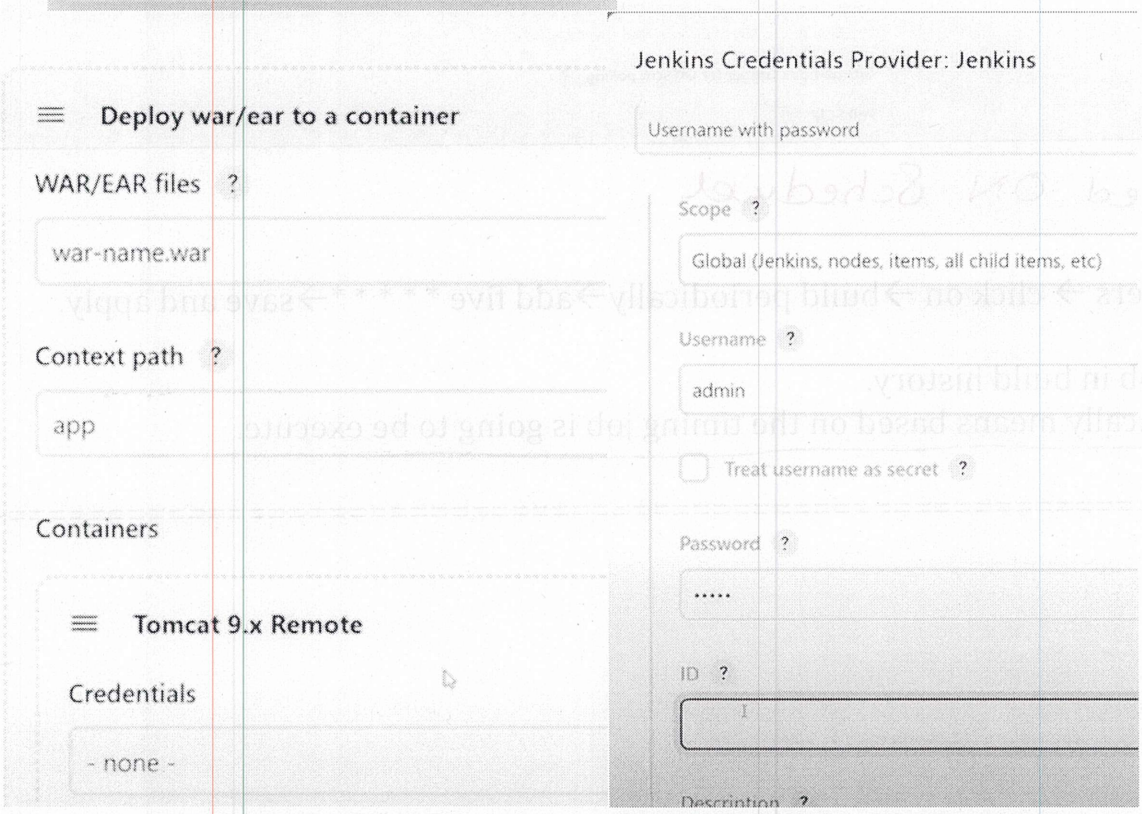
```
ubuntu@ip-172-31-14-139:~$ cd /var/lib/jenkins
ubuntu@ip-172-31-14-139:/var/lib/jenkins$ ll
```

- To check our war file is created or not.

Our build process is completed Now we need to deploy our project using tomcat server ,so we need to add tomcat server in our Jenkins

- Manage Jenkins → click on plugins → available plugins →search for deploy to container → select first one and add →now go to the project →click on configure→go to the post build action → click on war to container→





Click on credentials →add username and password→type id as tomcat server→add

What jenkins do is copying war file from Jenkins to the tomcat webapps folder

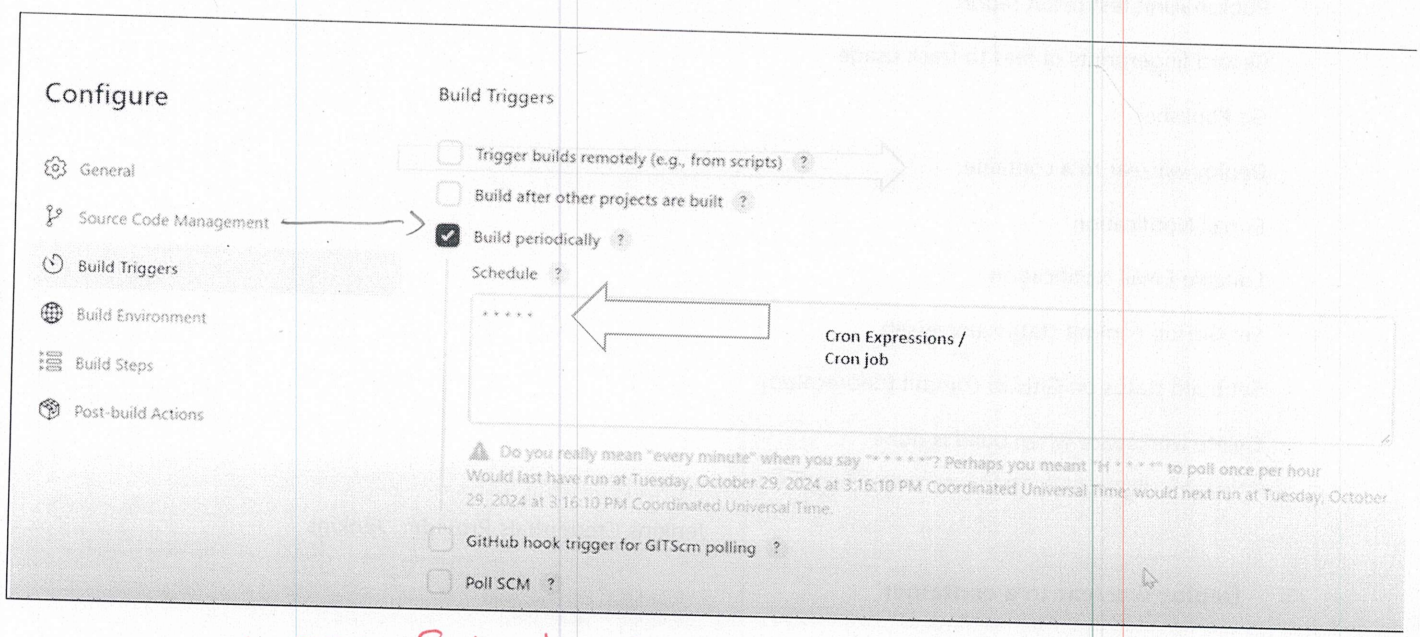
- Here container is nothing but is tomcat server

Note –Once we stop Jenkins vm and again we start Jenkins server it will be very slow then copy ip address of Jenkins vm and :8080/manage/configure→update ip address in Jenkins url →then our Jenkins server will be quick

Build periodically

Cron expressions

- **If I want to run my job on every 1 minute**



→ Based ON Schedule

Go to build triggers → click on →build periodically→add five * * * * *→save and apply.

You can check job in build history.

- Build periodically means based on the timing job is going to be execute.

==========

Poll Scm (Executes only when Code Change in github)

QUE - If I want to run my job after whenever code change in my job
ANS – Poll Scm means for every 1 minute Jenkins server should talk to github repo for any code changes happen in the github or not.

- After every 1 minute my Jenkins will communicate to the github hey github is there any code change , if is there any code change then job will execute.
- If no commit happen then job will not execute.

Note – for poll scm we need to form repo and copy our fork url to the source code management save and apply.

Need forking for this

==========

Note –
We can not directly commit changes in someone else github repository , if we want to commit changes in someone elses github repository then we need to fork the repository

==========

==========
Assignment
==========

=> Create Jenkins Job to perform below operation

1) Take source code from git repo

Git Repo : https://github.com/ashokitschool/maven-web-app.git

2) Build that code using maven

3) Deploy war file into tomcat server (diff linux vm)

Note: We need to install "deploy to container plugin" to deploy war file into tomcat server using jenkins.

Go to Jenkins Dashboard -> Manage Jenkins --> Manage Plugins -> Goto Available Tab -> Search For "Deploy To Container" Plugin -> Install without restart.

==========

Issue – Redeploy failed

==========

If you got below error

```
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/New-job1/target/maven-web-app.war to container Tomcat 9.x
Remote with context maven-web-app
ERROR: Build step failed with exception
org.codehaus.cargo.container.ContainerException: Failed to redeploy [/var/lib/jenkins/workspace/New-job1/target/maven-
web-app.war]
```

Ans- stop instance and restart again →update configure →post build action

Note: We need to configure "tomcat container" as "post build action" in jenkins job.

==========

Jenkins Pipeline 15-DevOps -30-OCT-24

==========

=> Jenkins pipeline is a way to define CI CD process as a code.

=> The whole CI CD workflow we will define as a code in pipeline.

=> when we are dealing with complex CI CD process then pipelines are highly recommended.

=> Pipeline contains set of stages to perform CI CD

Stage-1: Clone Git Repo

Stage-2: Maven Build

State-3: Code Review

Stage-4 : Artifact Upload

Stage-5: Build Docker Image

Stage-6: Push Image to Registry

Stage-7: Deploy App in K8S

Stage-8: Send Email Notification

=> We can create jenkins pipelines in 2 ways

1) Declarative Pipeline

2) Scripted (groovy) Pipeline

Jenkins Declarative Pipeline

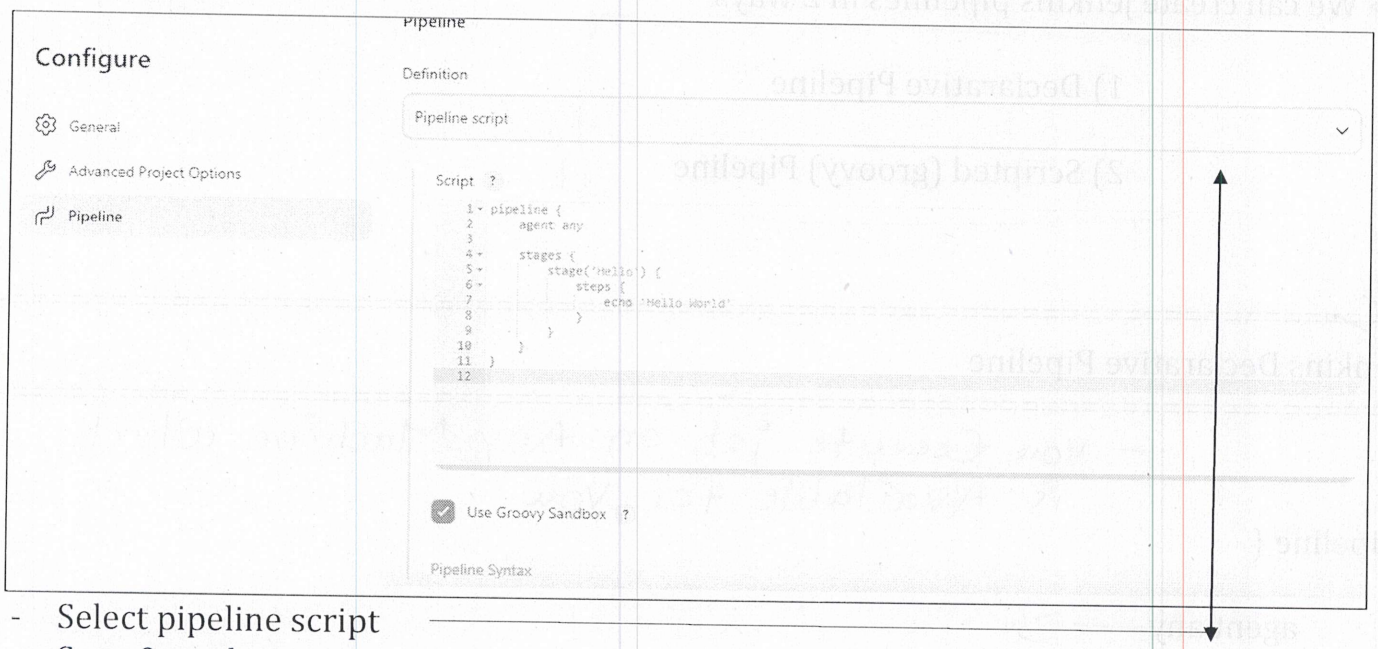
— You Execute job on Any Machine which is Available for you

```
pipeline {

    agent any ——>

    tools{
        maven "maven-3.9"
    }

    stages {
        stage('Git Clone'){
            steps{
                echo 'cloning git repo'
            }
        }
        stage('Maven Build'){
            steps{
                echo 'Maven Build'
            }
        }
        stage('Deploy'){
            steps{
                echo 'Tomcat Deployment'
            }
```

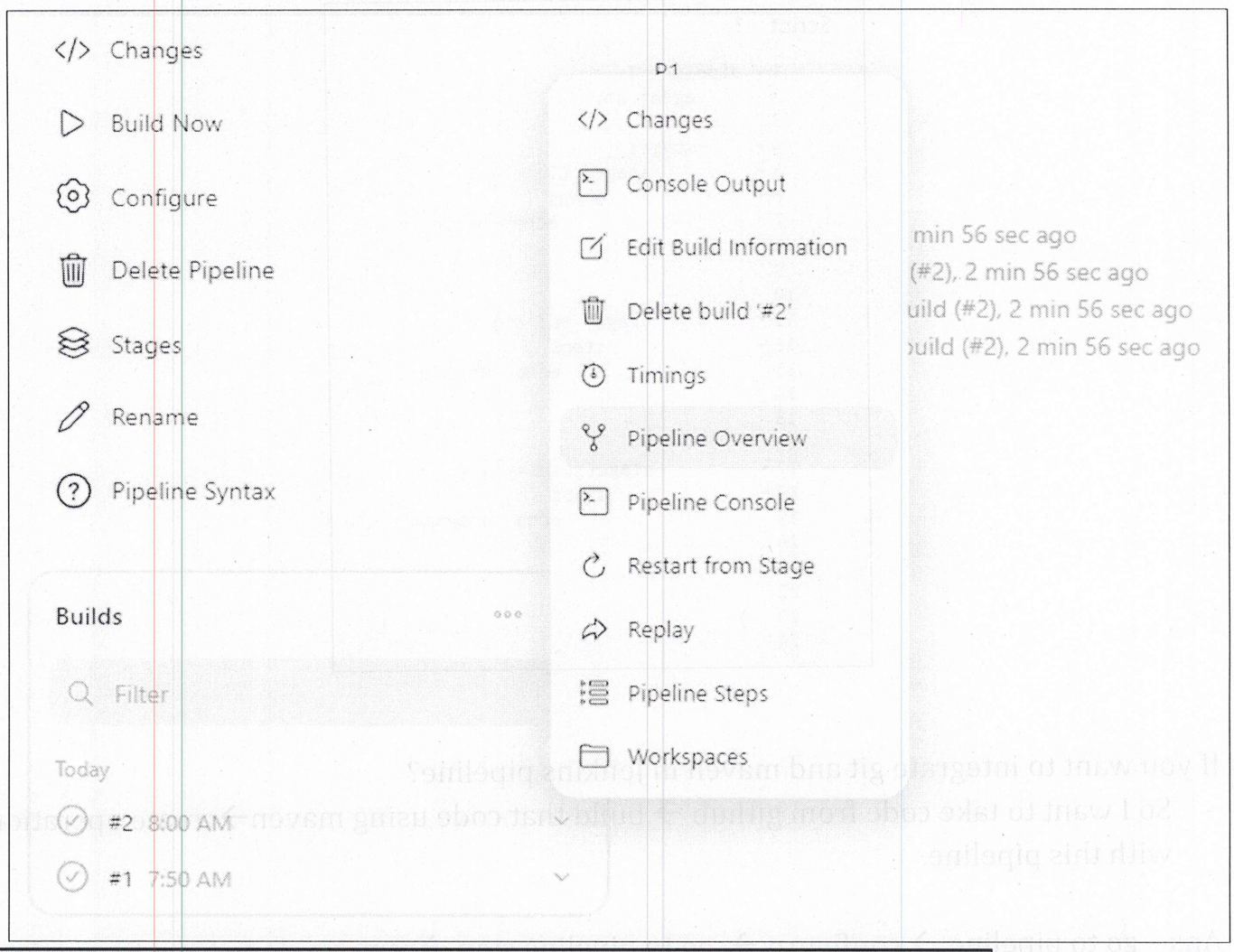
```
        }
    }
}
```

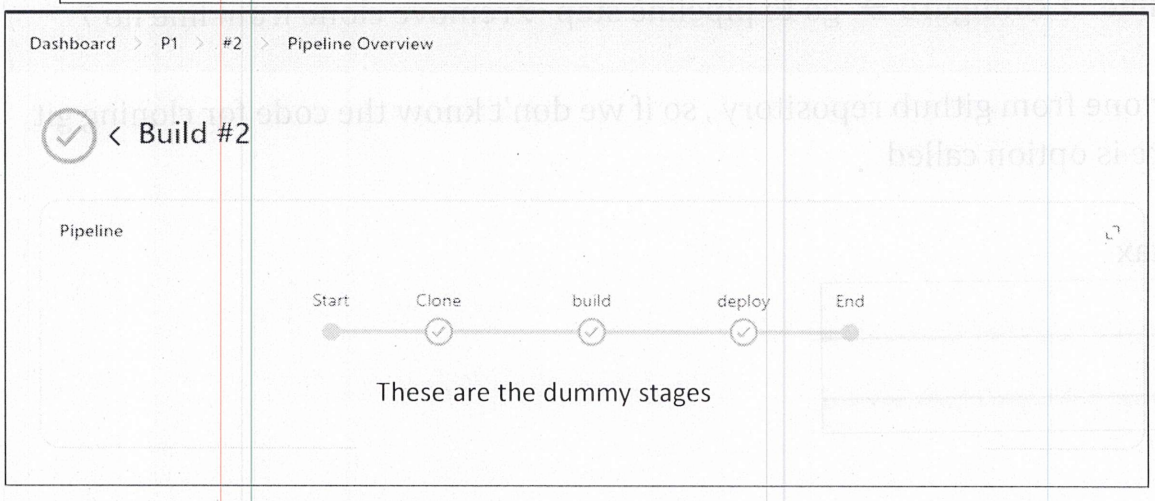
How to create simple pipeline in Jenkins ?

- Create new job
- go to pipeline



- Select pipeline script
- Save & apply





Additional Script

Script ?

```
1  pipeline {
2      agent any
3
4      stages {
5          stage('Clone') {
6              steps {
7                  echo 'Hello'
8              }
9          }
10
11         stage('build') {
12             steps {
13                 echo 'World'
14             }
15         }
16
17         stage('deploy') {
18             steps {
19                 echo 'program'
20             }
21         }
22     }
23 }
24
```

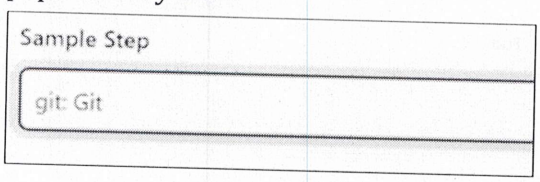
If you want to integrate git and maven in Jenkins pipeline?

- So I want to take code from github → build that code using maven→create operation with this pipeline.

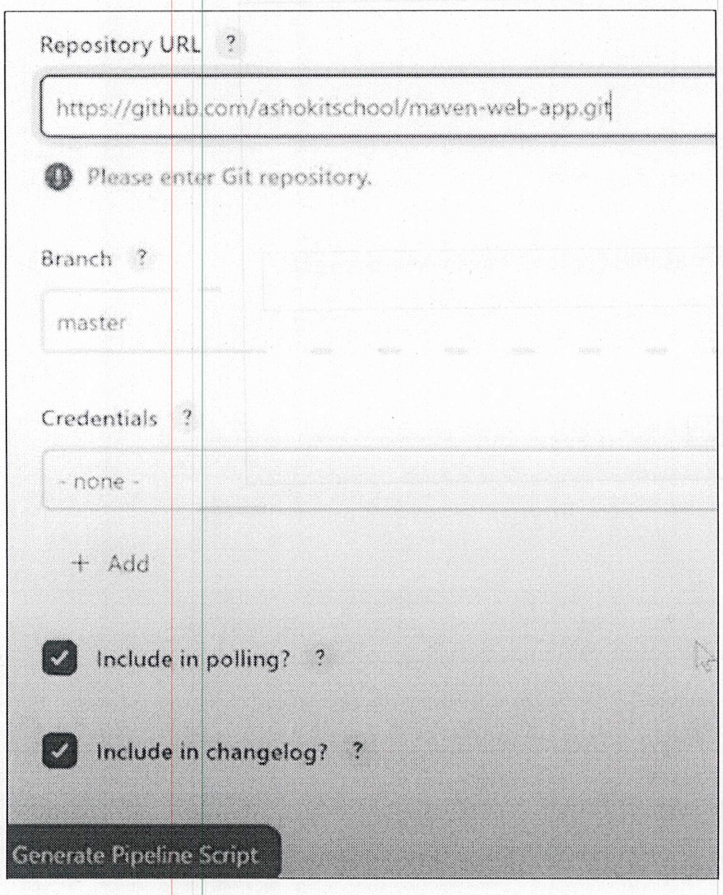
Ans – go to pipeline → configure → go to pipeline step →remove clone from line no 7

- We need to clone from github repository , so if we don't know the code for cloning git repo the there is option called

- pipeline syntax

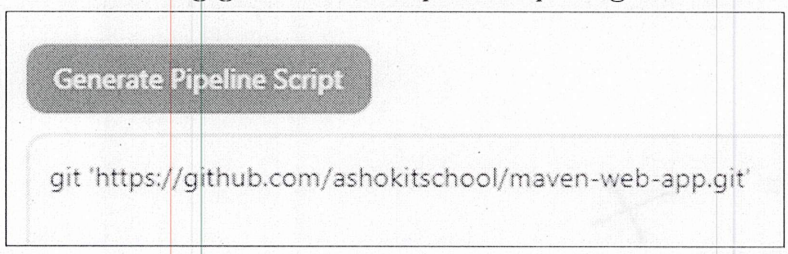


-

- 

- Now click on generate script

- After clicking generate script , script is generated

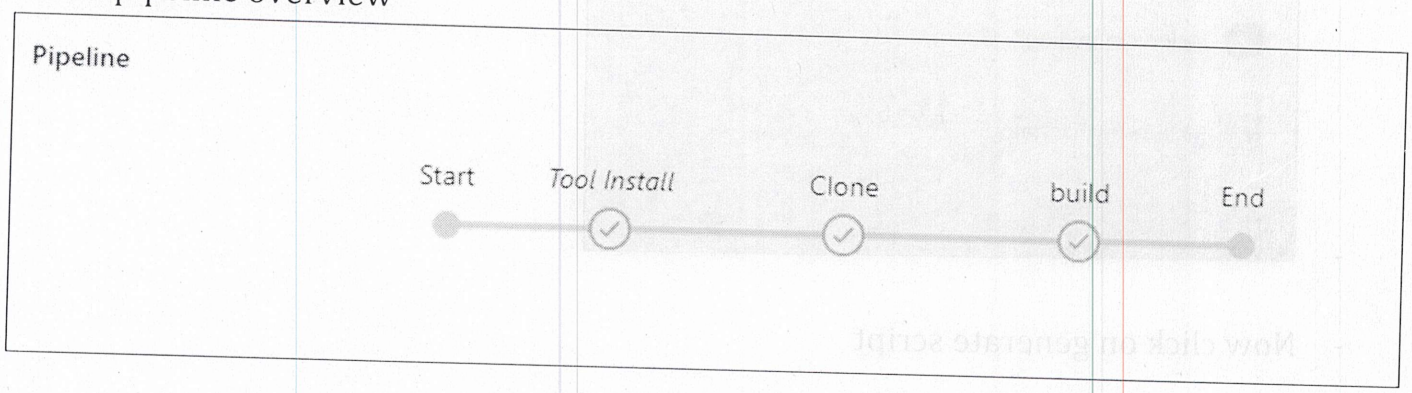
- 

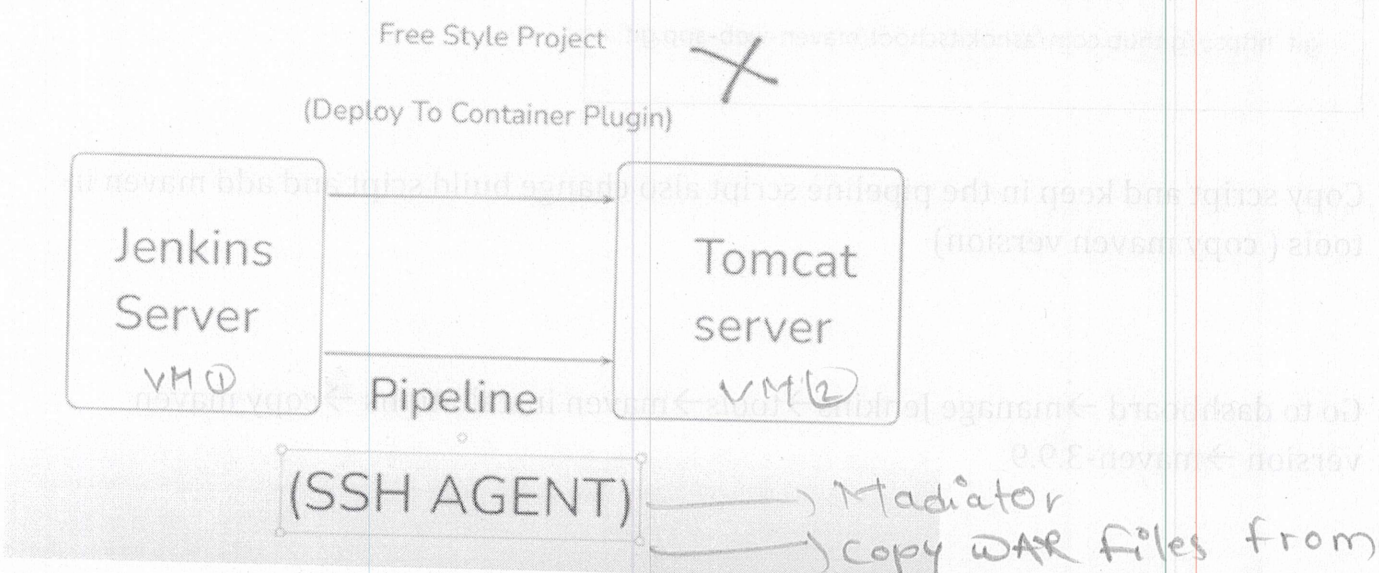
- Copy script and keep in the pipeline script also change build scipt and add maven in tools (copy maven version)

- Go to dashboard →manage Jenkins→tools→maven installations →copy maven version →maven-3.9.9

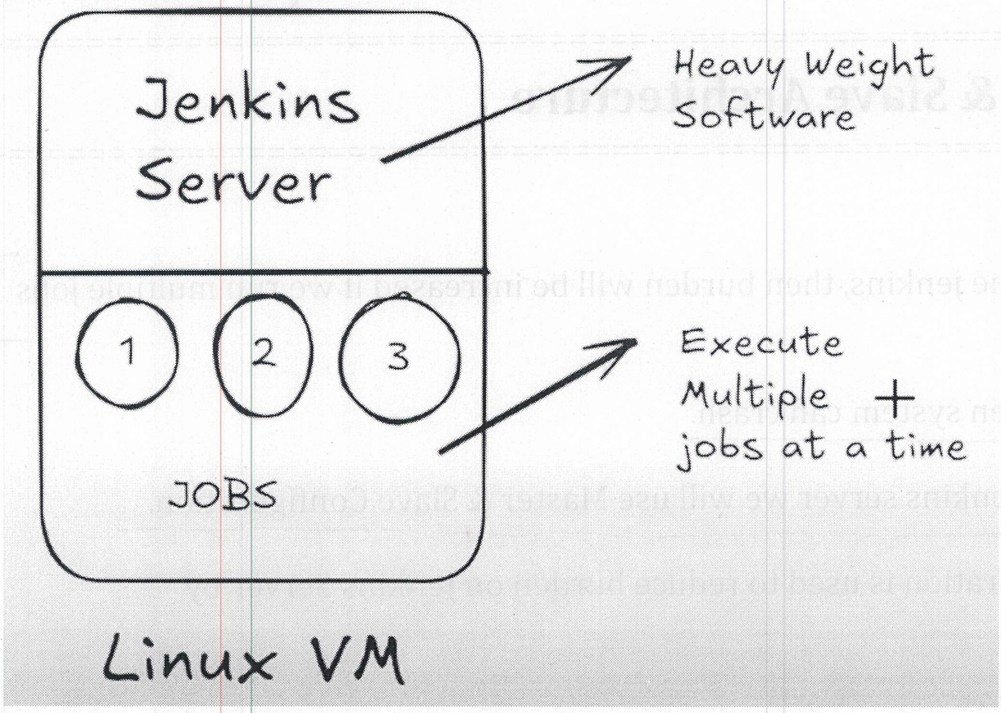
```
pipeline {
    agent any
    tools{
        maven "maven-3.9.9"
    }

    stages {
        stage('Clone') {
            steps {
                git 'https://github.com/ashokitschool/maven-web-app.git'
            }
        }

        stage('build') {
            steps {
                sh 'mvn clean package'
            }
```

- Check pipeline overview





Installation → Plugins → SSH Agent



Que - What if Jenkins server is crashed ?

Ans- we are going to loose all our jobs (Projects) to avoid this problem we are going to learn master and slave architecture.

17-DevOps -06-NOV-24

Jenkins - Master & Slave Architecture

=> If we use single machine jenkins, then burden will be increased if we run multiple jobs at a time.

=> If burden increased then system can crash.

=> To reduce burden on jenkins server we will use Master & Slave Configuration.

=> Master & Slave configuration is used to reduce burden on Jenkins Server by distributing tasks/load.

—> To Avoid burden And Problem will use Master & Slave Architecture.

Jenkins Master

—for One Master node we Can add any no. of Slave Nodes.

=> The machine which contains Jenkins s/w is called as Jenkins Master machine.

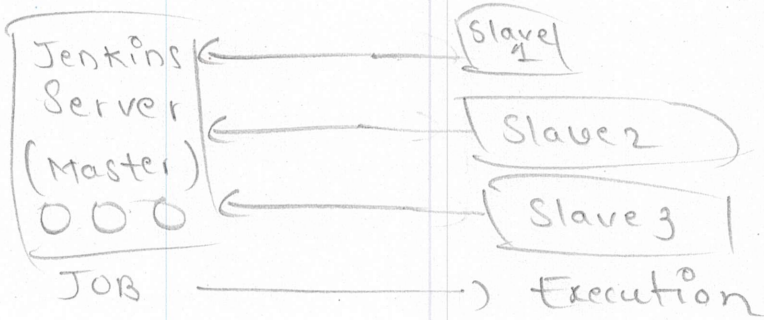
=> It is used to create the jobs —> ONLY

=> It is used to schedule the jobs
- which job should be executed on which slave machine.

=> It is responsible to distribute Jobs execution to slave machines.

Note: We can run jobs on Jenkins Master machine directley but not recommended.

Note - We can run multiple slaves on single master , but in one pipeline we can run only single slave.



==========

Jenkins Slave

==========

=> The machine which is connected with 'Jenkins Master' machine is called as 'Jenkins-Slave' machine.

=> Slave Machine will recieve task from 'Master Machine' for job execution.

→ If you want to run job on slave
Ex - agent "slave-1"

→ One pipeline only one slave to be used.

==========

Step-1 : Create Jenkins Master vm → Installation Steps

==========

1) Launch Linux VM (t2.medium) also micro is sufficient
2) Install Java s/w
3) Install Jenkins s/w

==========

Step-2 : Create Jenkins Slave vm

==========

1) Create EC2 instance (Ubuntu with t2.micro)
2) Connect to EC2 using ssh client
3) Change hostname for readability

```
$ sudo hostname jenkins-slave
$ exit and connect back
```

- After connecting back you will get below output

```
user@C1-GR-LAP-37 MINGW64 /d/Devops 27/Key-Pairs
$ ssh -i "keypair.pem" ubuntu@ec2-13-53-91-241.eu-north-1.compute.amazonaws.com
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Nov 11 05:01:29 UTC 2024

  System load:  0.05                Temperature:             -273.1 C
  Usage of /:   35.0% of 6.71GB     Processes:               111
  Memory usage: 26%                 Users logged in:         0
  Swap usage:   0%                  IPv4 address for ens5: 172.31.33.145


Expanded Security Maintenance for Applications is not enabled.

42 updates can be applied immediately.
22 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Mon Nov 11 04:57:08 2024 from 152.58.1.57
ubuntu@jenkins-slave:~$
```

3) Install Java Software

`sudo apt install default-jre`

$ If you encounter below error use command- `sudo apt update`

- Apt is a package manager for Ubuntu machine.

```
 Found [IP: 13.48.13.7 80]
E: Failed to fetch http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/pool/main/m/mesa/li
ot Found [IP: 13.48.13.7 80]
E: Failed to fetch http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/pool/main/m/mesa/li
Found [IP: 13.48.13.7 80]
E: Failed to fetch http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/pool/main/m/mesa/me
4  Not Found [IP: 13.48.13.7 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
ubuntu@jenkins-slave:~$ java -version
Command 'java' not found, but can be installed with:
sudo apt install openjdk-17-jre-headless  # version 17.0.12+7-1ubuntu2~24.04, or
sudo apt install openjdk-21-jre-headless  # version 21.0.4+7-1ubuntu2~24.04
sudo apt install default-jre              # version 2:1.17-75
sudo apt install openjdk-11-jre-headless  # version 11.0.24+8-1ubuntu3~24.04.1
sudo apt install openjdk-8-jre-headless   # version 8u422-b05-1~24.04
sudo apt install openjdk-19-jre-headless  # version 19.0.2+7-4
sudo apt install openjdk-20-jre-headless  # version 20.0.2+9-1
sudo apt install openjdk-22-jre-headless  # version 22~22ea-1
ubuntu@jenkins-slave:~$ sudo apt update
```

4) Create one directory in /home/ubuntu

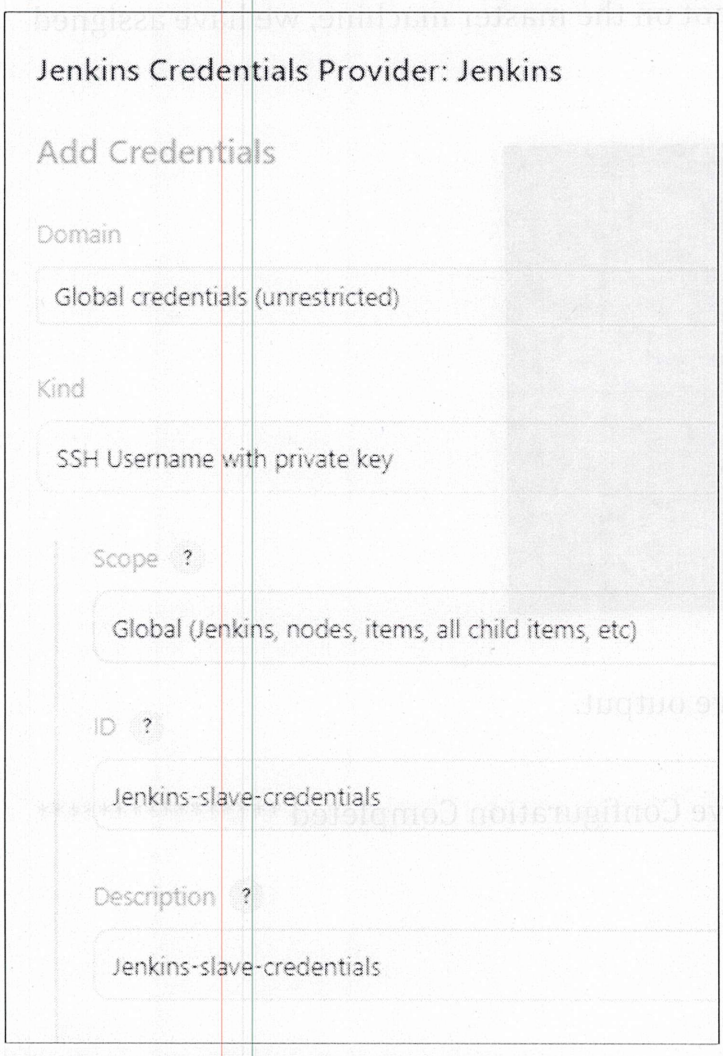
$ mkdir slave-node

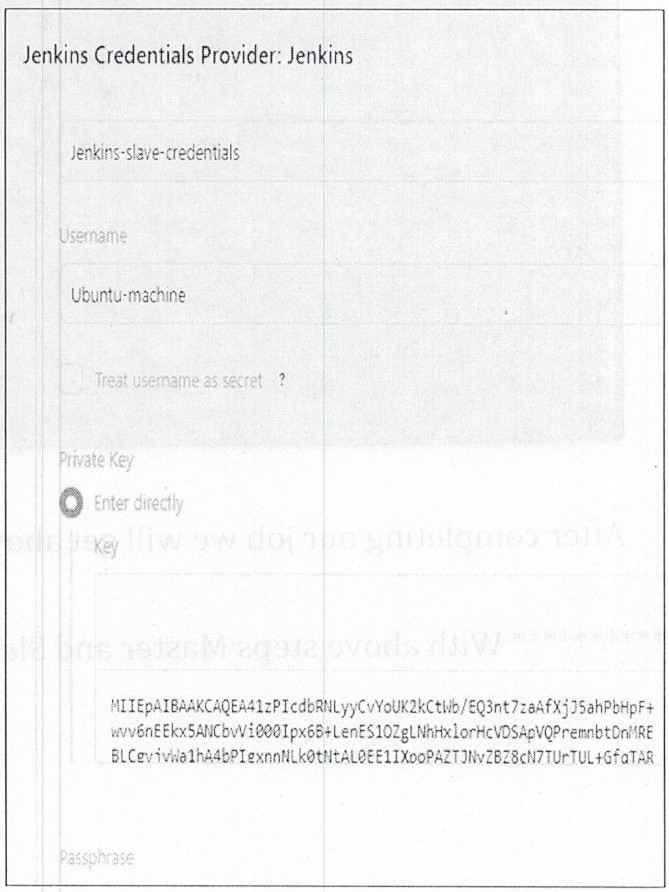
==========

Step-3: Configure Slave Node in Jenkins Master Node

==========

1) Go to Jenkins Dashboard
2) Go to Manage Jenkins
3) Select Nodes option
4) Click on 'New Node' -> Enter Node Name (slave-1)-> Select Permanent Agent→Create
5) Enter Remote Root Directory (/home/ubuntu/slavenode)
6) Enter Label name as "Slave-1"
7) Select Launch Method as 'Launch Agent Via SSH'
8) Give Host as 'Slave VM DNS URL'
9) Add Credentials (Select Kind as : SSH Username with private key)





10) Enter Username as : ubuntu
11) Select Private Key as Enter Directley and add private key

Note: Open gitbash and read pem file content and copy content add add it.

12) Select Host Key Strategy as 'Manually Trusted Key Verification Strategy'

13) Click on Apply and Save (We can see configured slave)

==========

Now we need to run our job on slave node

- Go to the job and edit pipeline script

```
pipeline {
    agent {
        label "Slave-1"
    }
    tools{
```

- My job should be run on slave machine not on the master machine, we have assigned the task to the Slave-1 machine.

```
ubuntu@jenkins-slave:~/slave-node$ ll
total 1388
drwxrwxr-x 6 ubuntu ubuntu    4096 Nov 11 15:44 ./
drwxr-x--- 6 ubuntu ubuntu    4096 Nov 11 15:44 ../
drwxrwxr-x 3 ubuntu ubuntu    4096 Nov 11 15:44 caches/
drwxrwxr-x 4 ubuntu ubuntu    4096 Nov 11 15:43 remoting/
-rw-rw-r-- 1 ubuntu ubuntu 1393083 Nov 11 15:43 remoting.jar
drwxrwxr-x 3 ubuntu ubuntu    4096 Nov 11 15:44 tools/
drwxrwxr-x 4 ubuntu ubuntu    4096 Nov 11 15:44 workspace/
ubuntu@jenkins-slave:~/slave-node$ cd workspace
ubuntu@jenkins-slave:~/slave-node/workspace$ ll
total 16
drwxrwxr-x 4 ubuntu ubuntu 4096 Nov 11 15:44  ./
drwxrwxr-x 6 ubuntu ubuntu 4096 Nov 11 15:44  ../
drwxrwxr-x 5 ubuntu ubuntu 4096 Nov 12 04:06  P1/
drwxrwxr-x 2 ubuntu ubuntu 4096 Nov 12 04:06 'P1@tmp'/
ubuntu@jenkins-slave:~/slave-node/workspace$
```

- After completing our job we will get above output.

********** With above steps Master and Slave Configuration Completed **********

Issue

If you get below error that means you have not authenticate properly.

```
Started by user admin
[Pipeline] Start of Pipeline (hide)
[Pipeline] node
Still waiting to schedule task
‘Slave-1’ is offline
```

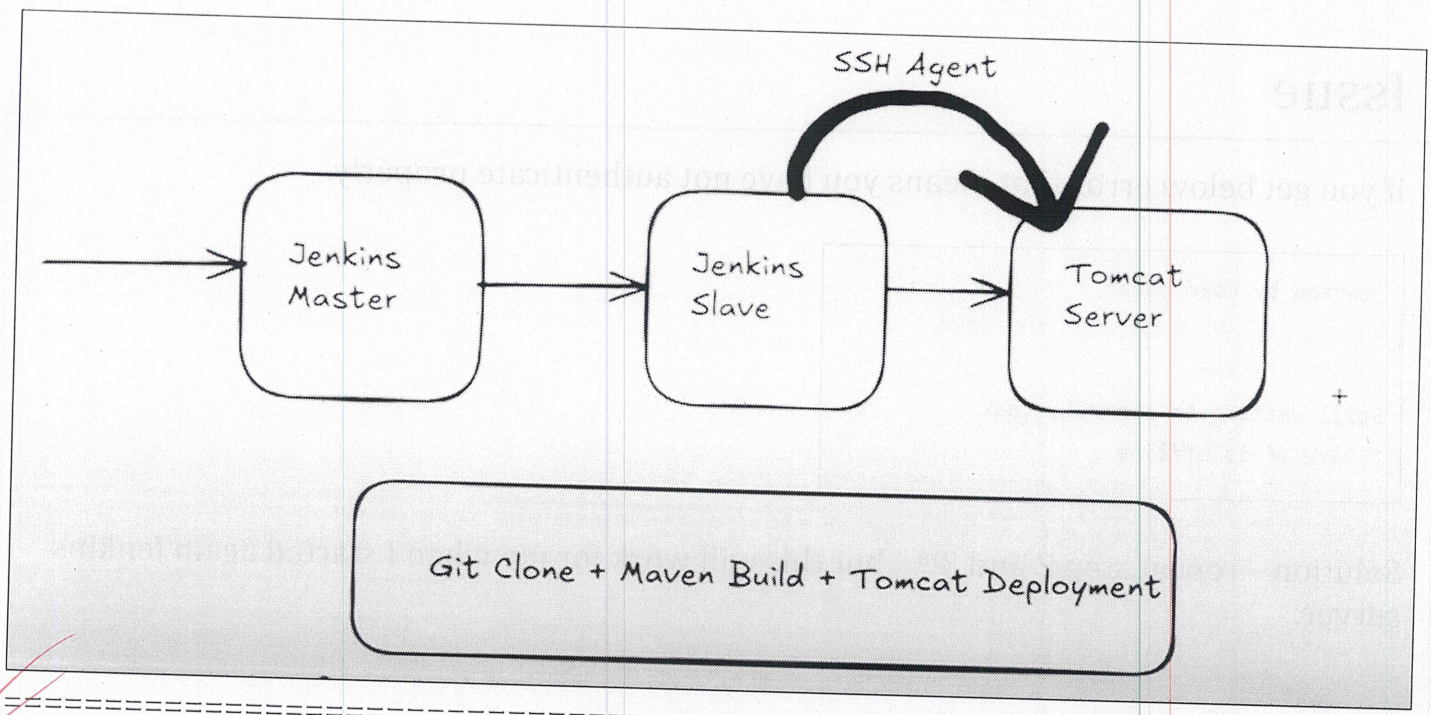
Solution – repeat step 2 and 3rd , but this will work for me when I started again Jenkins server.

- If your slave getting offline again & again

Solution – copy keypair from Starting to Ending without Containing Space.

Note - for me only Java 17 is Helpful
- Resolving issue.

git cloning
↳ Maven build
↳ Deploy war file to tomcat Server through pipeline



==========

I want to create a job that job should contains (git clone+maven build + tomcat deployment)

- job you need to create on the master machine using pipeline
- after job is executing maven build will happen and war file will be generated
- after generating war file we need to deploy that war file to the tomcat server.

- Jenkins master contains a job
- Job execution will happen on slave
- And war file will be deploy on the tomcat server.
- We need to connect Jenkins slave to tomcat server

- Earlier we used deploy to war container for deploying war file
- Now we are going to used SSH Agent to copy war file from Jenkins slave to the tomcat server.

SSH Agent Plugin Installation

- The purpose of SSH Agent plugin is to copy the data from one machine to another machine
- Install SSH Agent plugin
- Manage Jenkins → Plugins → Available plugins

18-DevOps -07-NOV-24

==========

Assignments

==========

1) Jenkins CI CD pipeline with below 3 stages

Stage-1 : Git Clone

Stage-2 : Maven Build

Stage-3 : Tomcat Deployment using SSH Agent

Solution-

```
pipeline {
    agent {
        label 'Slave'
```

1. Create fresh pipeline and add above script.
2. Use github repo reference https://github.com/sachin2460

2) How to take jenkins backup ?

@@ Reference Video : https://www.youtube.com/watch?v=5Tb-AOUFuKQ

Solution

-install Thinbackup plugin
- go to manage Jenkins → setting →global tool configuration (manage/configure)

==========

SSH Agent Configuration ⟶ Installation

==========

=> SSH Agent is used to establish remote ssh connection from one linux vm to another linux vm

Ex: jenkins server should connect with tomcat server to copy war file

=> Install SSH Agent plugin

(Manage Jenkins -> Plugins -> Available -> Search for SSH Agent -> Install)

=> Use pipeline syntax and create ssh-agent for tomcat server vm.

-> Snippet Generator
-> Sample Step -> Select SSh Agent
-> Add -> SSH Username with private key
-> Configure Tomcat server username and pem file content
-> Click on generate pipeline script to get ssh-agent

=> Configure SSH Agent details in Deployment stage steps like below

```
sshagent(['tomcat-server-credentials']) {
// some block
}
```

=> With the help of ssh-agent we will copy war file to tomcat server using scp command

```
sh 'scp -o StrictHostKeyChecking=no target/01-maven-web-app.war ec2-user@public-ip:/home/ec2-user/apache-tomcat-9.0.91/webapps'
```

```
=========================================
Git + Maven + Tomcat + Jenkins Pipeline
=========================================

pipeline {
  agent any

  tools{
    maven "maven-3.9.8"
  }

  stages {
    stage('Git Clone') {
      steps {
        git branch: 'develop',
          url: 'https://github.com/ashokitschool/maven-web-app.git'
      }
    }
    stage('Maven Build'){
      steps{
       sh 'mvn clean package'
      }
    }
    stage('Deployment'){
      steps{
        sshagent(['tomcat-server-credentials']) {
          sh 'scp -o StrictHostKeyChecking=no
            target/01-maven-web-app.war
              ec2-user@public-ip:/home/ec2-user/apache-tomcat-9.0.91/webapps'
                copy public ip address from aws
        }
      }
    }
  }
}
```

==========

Integrate Email Notifications in Jenkins

18-DevOps -07-NOV-24

==========

Step-1

-> We can configure Email notifications in Jenkins

-> With this option we can send email notification to team members after jenkins job execution completed.

-> We need to configure SMTP properties to send emails.

-> Go To Manage Jenkins
-> Go To System
-> Go to "Extended E-mail Notification"
-> We will add company provided SMTP server details to send emails.

Note: For practise we can use GMAIL SMTP Properties

SMTP Server : smtp.gmail.com
SMTP Port : 587

Note: Under Advanced section add your gmail account credential for authentication purpose.

Note: Instead of gmail password we need to add gmail app password

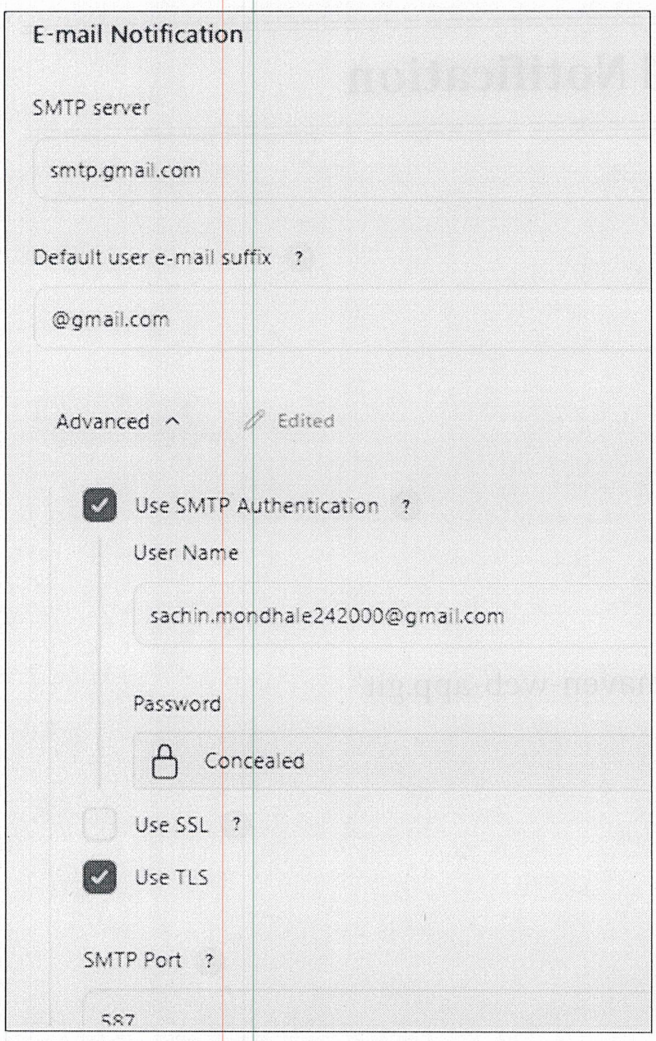
URL To generate gmail app pwd : https://myaccount.google.com/apppasswords

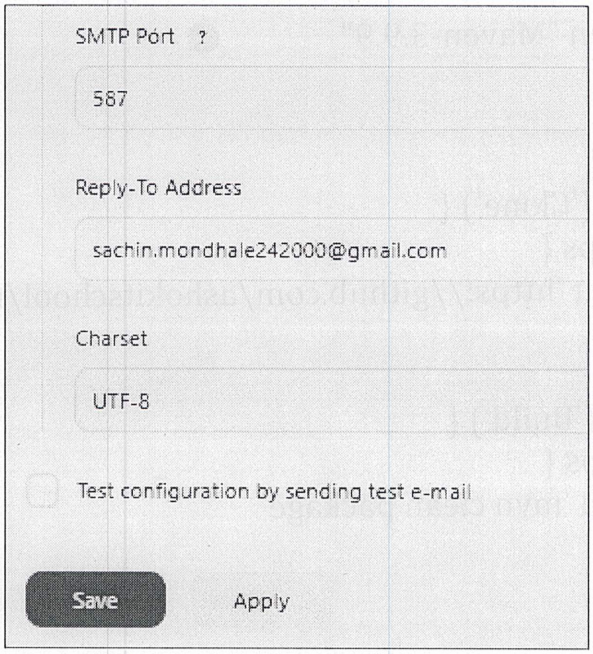
=> Select use TLS checkbox

Step-2

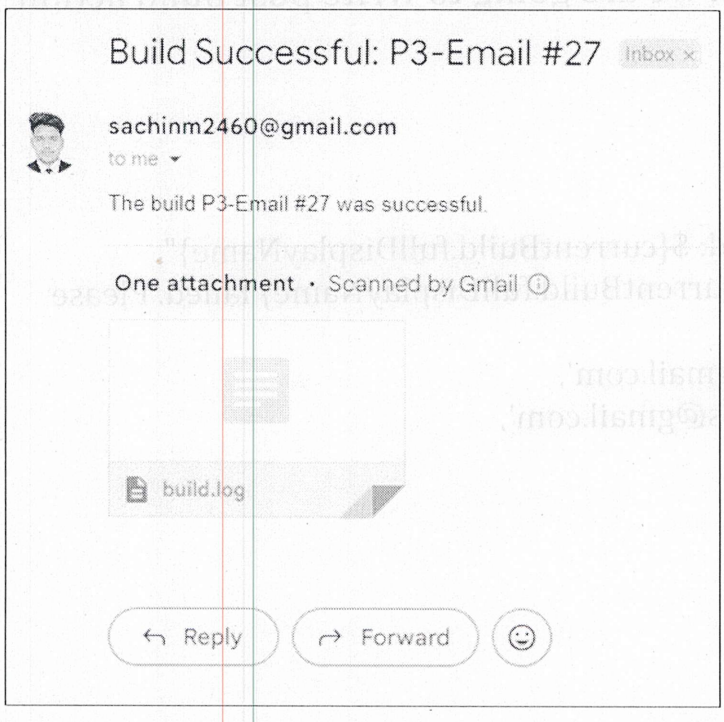
=> For testing purpose we can use "Email Notification option which is available at the bottom of the page"

- Add 2nd gmail account credentials in Email Notification.


E-mail Notification
SMTP server
smtp.gmail.com
Default user e-mail suffix ?
@gmail.com
Advanced
Edited
Use SMTP Authentication ?
User Name
sachin.mondhale242000@gmail.com
Password
Concealed
Use SSL ?
Use TLS
SMTP Port ?
587


SMTP Port ?
587
Reply-To Address
sachin.mondhale242000@gmail.com
Charset
UTF-8
Test configuration by sending test e-mail
Save
Apply

Output-


Build Successful: P3-Email #27
Inbox ×
sachinm2460@gmail.com
to me
The build P3-Email #27 was successful.
One attachment • Scanned by Gmail
build.log
Reply
Forward

==========

Declarative Pipeline with Email Notification

==========

```
pipeline {
  agent any

  tools{
    maven "Maven-3.9.9"
  }

  stages {
    stage('Clone') {
      steps {
        git 'https://github.com/ashokitschool/maven-web-app.git'
      }
    }
    stage('Build') {
      steps {
        sh 'mvn clean package'
      }
    }
  }
```

Note – After all the stages completion we are going to write post build action

```
post {
        failure {
            emailext(
                subject: "Build Failed: ${currentBuild.fullDisplayName}",
                body: "The build ${currentBuild.fullDisplayName} failed. Please
check the console output for more details.",
                to: 'ashokitschool@gmail.com',
                from: 'ashokit.classes@gmail.com',
                attachLog: true
            )
        }
```

```
success {
                emailext(
                        subject: "Build Successful: ${currentBuild.fullDisplayName}",
                        body: "The build ${currentBuild.fullDisplayName} was
successful.",
                        to: 'ashokitschool@gmail.com',
                        from: 'ashokit.classes@gmail.com',
                        attachLog: true
                )
            }
        }
}
```

==========

DL= Distribution list
- Distribution list is email group in a team
- Ex. All@c1india.com

— Interview Question

Que - How you will Send email Notification using Jenkins pipeline?

Ans - using post build Action

— Before that Configure Smtp properties.

Del
— Deploy war file using tomcat
OR
— Create Pipeline -

==========

Jenkins Pipeline with Parallel Stages

==========

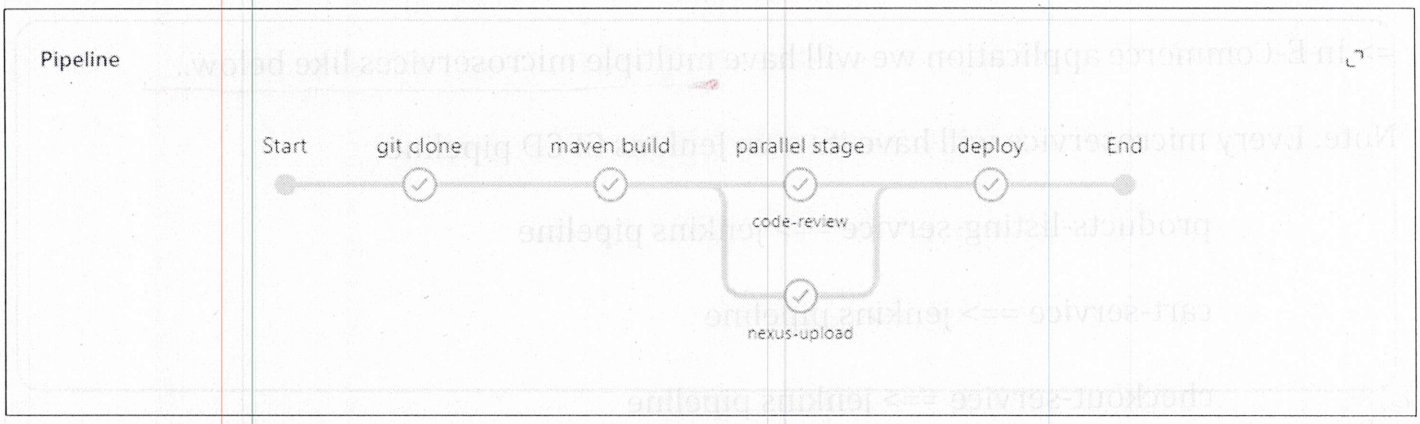
=> In general jenkins job stages will execute in sequential manner (one after other)

=> Jenkins support parallel stages execution also for reduce the time

```
pipeline {

    agent any

    stages{
        stage('git clone'){
            steps{
                echo 'git clone....'
            }
        }
        stage('maven build'){
            steps{
                echo 'maven build...'
            }
        }

        stage('parallel stage'){
          parallel{
            stage('code-review'){
              steps{
                echo 'code review....'
              }
            }
            stage('nexus-upload'){
              steps{
                echo 'nexus upload...'
              }
            }
          }
        }

        stage('deploy'){
            steps{
                echo 'deployment...'
```

```
            }
        }
    }
}
```

Output –


Pipeline
Start
git clone
maven build
parallel stage
deploy
End
code-review
nexus-upload

groovy basically Means Calling Defined functions for Code reusability.

Working with Shared Libraries in Jenkins

Main advantage to use shared libraries is Logic Reuseability.

=> Lets understand "ECommerce Application" usecase in real-time

=> In E-Commerce application we will have multiple microservices like below..

Note: Every microservice will have its own Jenkins CI CD pipeline.

products-listing-service ==> jenkins pipeline

cart-service ==> jenkins pipeline

checkout-service ==> jenkins pipeline

tracking-service ==> jenkins pipeline

cancel-service ==> jenkins pipeline

admin-service ==> jenkins pipeline

reports-service ==> jenkins pipeline

=> When we are writing multiple pipelines, we can see some common logics in all pipelines.

Ex: Maven build + code review + nexus upload...

=> Instead of writing common logics in all pipelines we can write the logic at one place and we can re-use it at multiple places.

=> To achieve pipeline logic re-usability we can use 'shared libraries' concept in jenkins.

=> To create shared libraries we will use 'groovy scripting'.

Note: Shared Libraries we can store in our git repo.

Shared Libraries = groovy Scripting

19-DevOps -08-NOV-24

==========

Jenkins Pipeline with shared library

==========

How to Configure Shared Libraries ?

Ans →

Step-1 : Create shared library and store in git repo

@@ Git Repo : https://github.com/ashokitschool/my_shared_libraries.git

Step-2 : Configure shared library in jenkins

=> Go to Jenkins Dashboard > Manage Jenkins > Configure System.

=> Scroll to the "Global Pipeline Libraries" section.

=> Click Add to create a new library.

=> Enter the following details:

=> Name: A unique identifier for the library (e.g., ashokit_lib).

Default Version: The branch or tag you want to use by default, such as main or master.

=> Retrieval Method: Choose Modern SCM.

=> Source Code Management: Select Git (or your preferred SCM).

=> Project Repository: Enter the Git repository URL of your shared library.
- Here no need of credentials beacause library is open .

=> Click Save to apply the changes

Syntax

LibraryName

- @Library(' library_name ')

Step-3 : Create Pipeline and use shared library in pipeline like below

---

```
@Library('ashokit_lib') _

pipeline {
    agent any

    tools{
        maven "maven-3.9.9"
    }

    stages {
        stage('Hello') {
            steps {
                welcome()
            }
        }

        stage('git clone'){
            steps{
                    gitClone();
            }
        }

        stage('maven build'){
            steps{
                    mavenBuild()
            }
        }
    }
}
```

What is Jenkinsfile ?

=> It is used to store pipeline code.

=> We will maintain Jenkins file in project git repo only. (for Multibranch Pipeline)

=>Pipeline code is available in Jenkins file

—) for Normal pipeline No Need to keep Jenkins file.

Multi Branch Pipeline

— To Support Parallel Developement in Project

=> In one git repo we can have multiple branches

a) main
b) develop
c) feature
d) release

=> Creating seperate jenkins pipeline for every branch is difficult.

=> We can use "Multi Branch Pipeline" to build the code available in multiple branches at a time using single pipeline.

=> When we create "Multi Branch Pipeline" it will scan all the branches in given git repo and it will execute pipeline for all branches.

⇨ Two or more branches cannot merge in multi branch pipeline.

Note :—

We will not use multi-branch in production.

Note:

When we run multi branch pipeline for second time it will verify code changes happend in which branch and it will execute pipeline for only those branches.

Note – Earlier default branch is master but now default branch is main.

==========

User Management

==========

=> In our project multiple team members will be available

a) developers

b) testers

c) devops engineers

=> For every team member we need to provide jenkins login access.

=> Ops team is responsible to create user accounts in jenkins for team members.

Ops Team : Should have all priviliges in Jenkins

Ex : Create / edit / update / delete / run

Dev Team & Testing Team : Only Job Execution Priviliges

Ex : Run

==========

How to create users and manage user permissions

==========

-> Go to Jenkins Dashboard

-> Manage Jenkins -> Manage Users

-> Create Users

-> Go to Configure Global Security

-> Manage Roles & Assign Roles

->manage Jenkins → security→Authorization→change to matrix based.-->add user

Note: By default admin role will be available and we can create custom role based on requirement

-> In Role we can configure what that Role assigned user can do in jenkins

-> In Assign Roles we can add users to particular role

==========

Working with User Roles in Jenkins

==========

Step-1 : Install Required Plugins

=> Install "Role-based Authorization Strategy" Plugin

=> This plugin allows you to define roles and assign them to users or groups.

Step-2 : Configure Security

=> Go to "Manage Jenkins" > "Configure Security."

=> Select Authorization as "Role-Based Strategy"

=> Click "Save" to apply the changes

Step-3 : Create User Roles

=> Go to "Manage Jenkins" > "Manage and Assign Roles."

=> Click "Manage Roles" and define new roles based on your requirements (e.g., admin, developer, tester).

=> Click "Add" to create a new role, and specify the permissions for that role.

Step-4 : Assign Users to Roles

=> After creating roles, go to "Manage Jenkins" > "Manage Users & Roles."

=> Select a user and click "Assign Roles" to add them to one or more roles.

Step-5 : Test the user login functionality