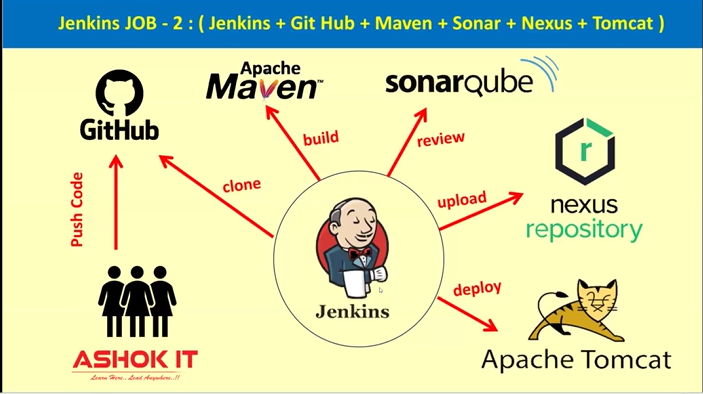
13-DevOps -28-OCT-24

=====================================================================-

**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**

=====================================================================

=> 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).

=====================================================================

**Jenkins Setup**

=====================================================================

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

=====================================================================

**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

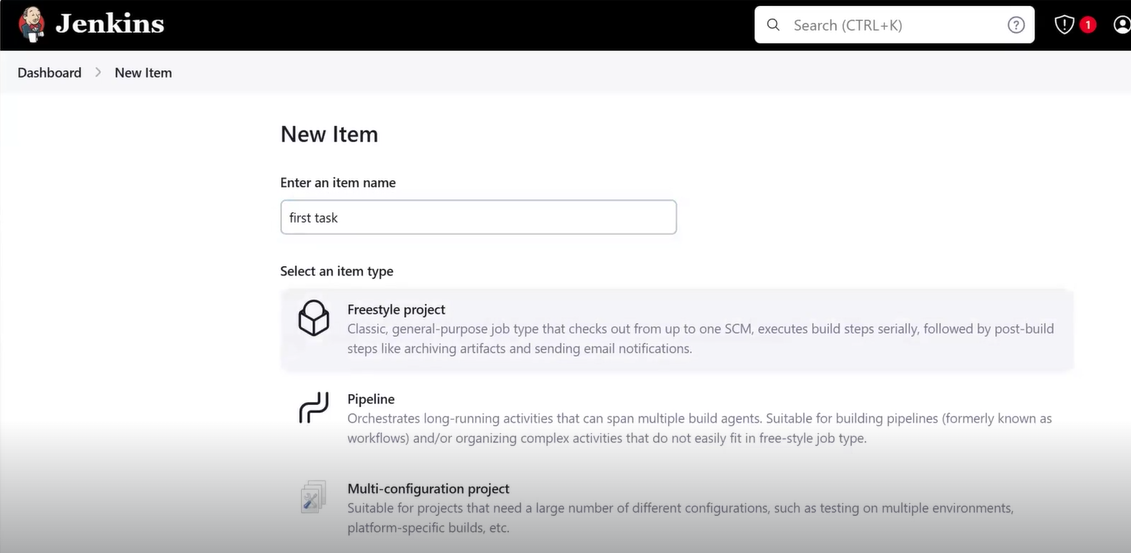
## Step-3 : Run Jenkins Job with "Build Now" option

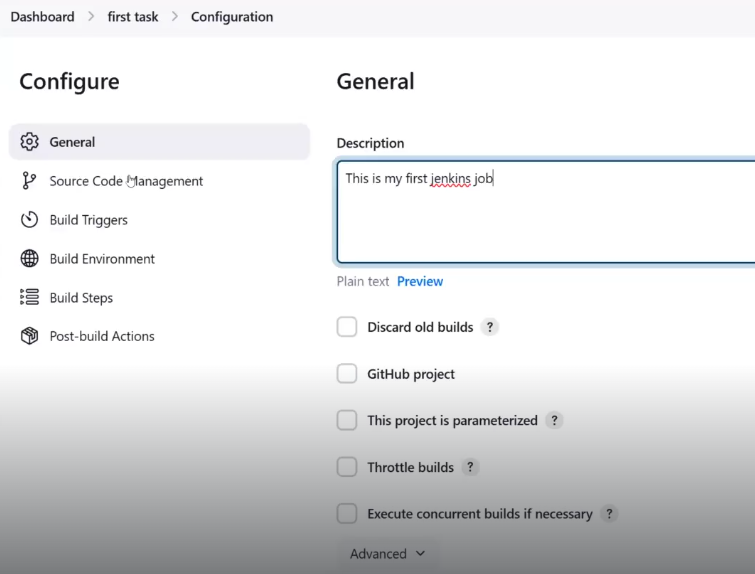
## Step-4 : Click on 'Build Number' and then click on 'Console Ouput' to see job execution details.

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

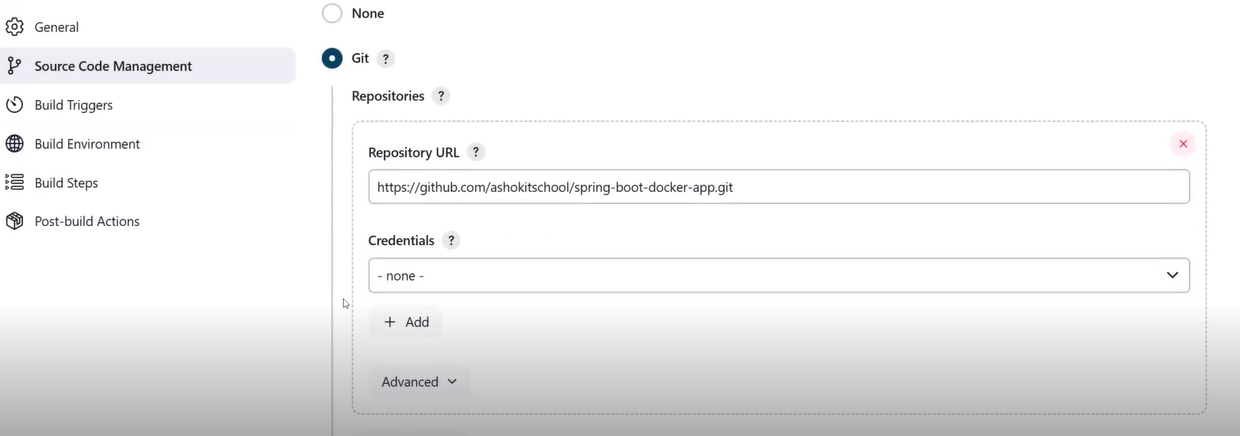
=> Go to jenkins workspace and then go to job folder then go to target folder there we see jar/war file created.

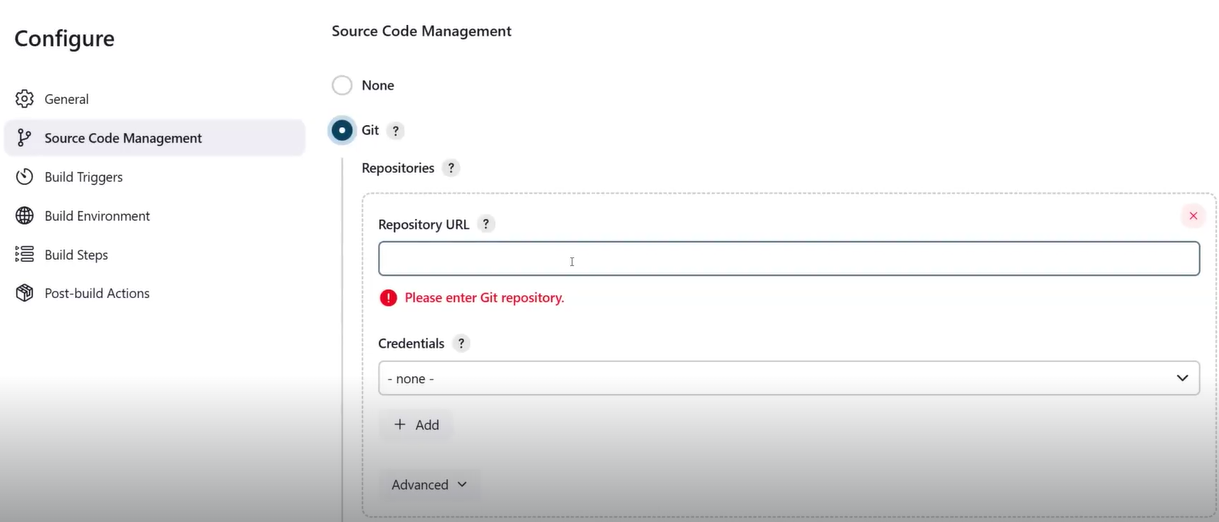
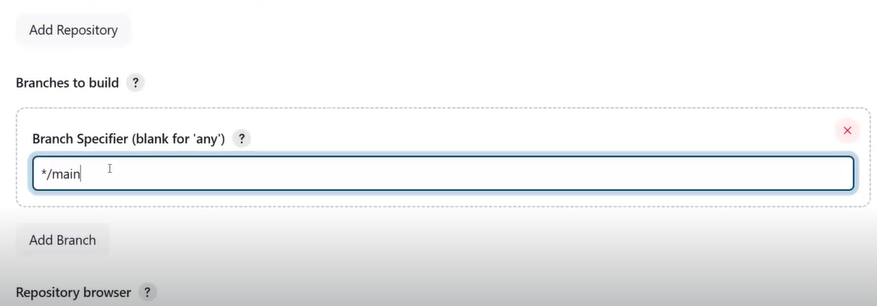
=====================================================================

**Creating job in Jenkins**

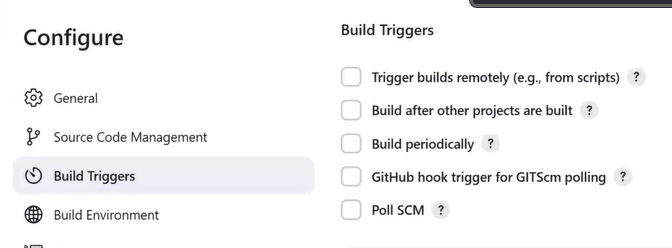
**Step 1: General**

**Step 2 : Source code management**

****

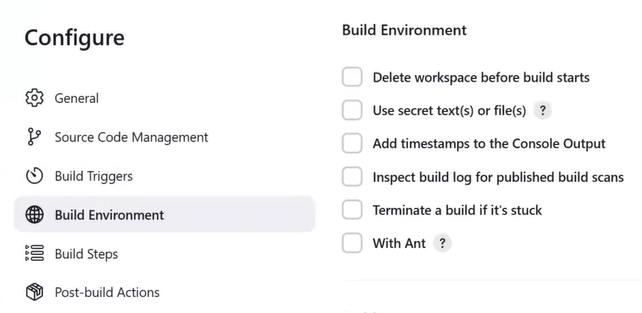
* we can clone git-hub repository , if github repository is private repo then we have to configure account credentials also .

**3 – Build Triggers**



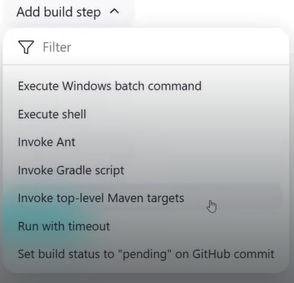
* When we want to run our job , automatically / manually (build and deployment)
* Do you want to run our on specific time
* Do you want to run our job whenever there is code change in github that is done by using build triggers.
* Build triggers is basically when our job is executed.

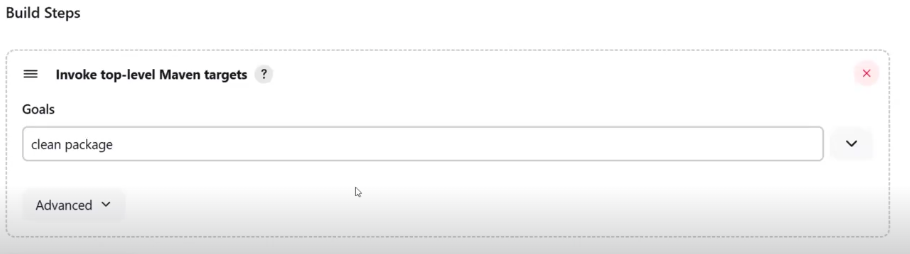
**Step 4 : Build Environment**

****

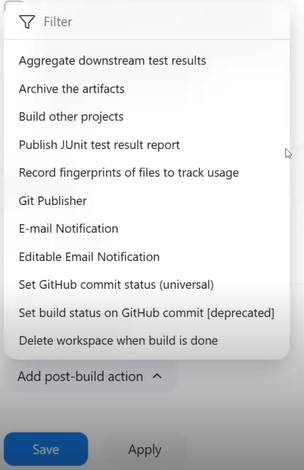
**-**read above properties that we need to configure according to our needs.

**Step 5 : Build Step**

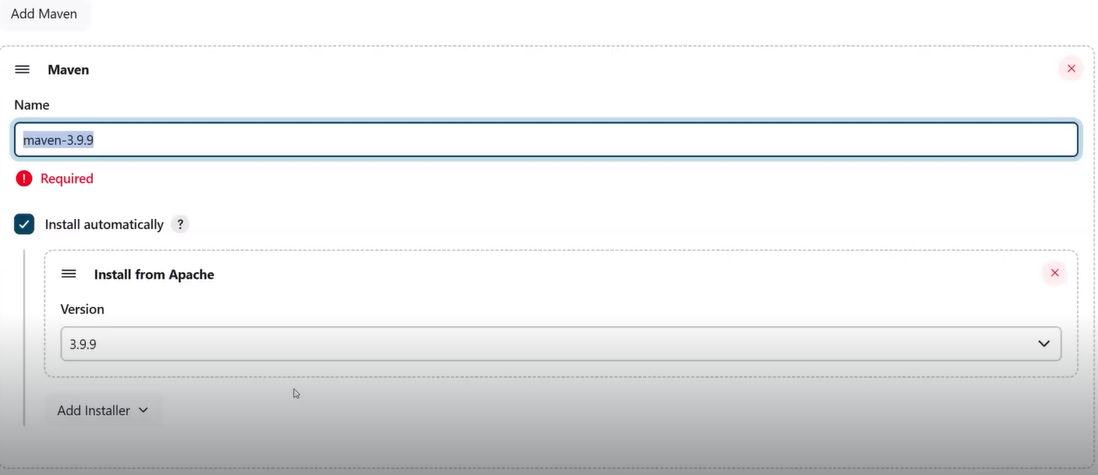
****

****

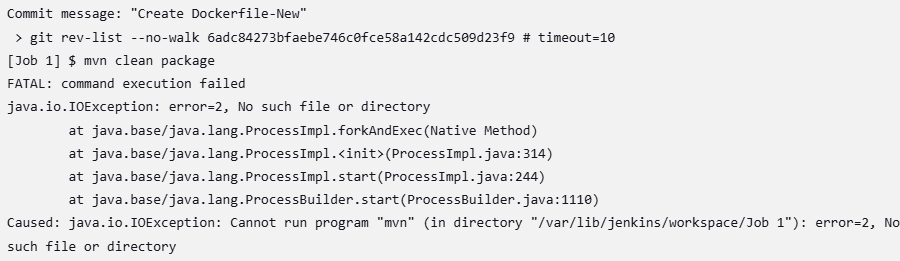
**Step 6 : Post-build Actions**

****

* If build is failed and maven is not available then click on 🡪manage Jenkins 🡪tools
* ****

****

**Issue-If You got an issue like below**



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

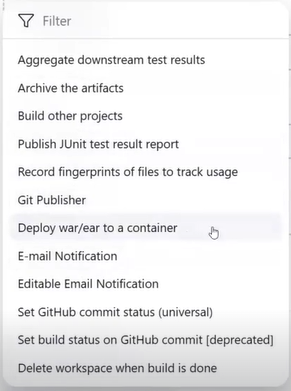
Go to dashboard 🡪 job 1 🡪configure 🡪 go to build step 🡪change maven version🡪change default to the latest maven version 🡪save & apply

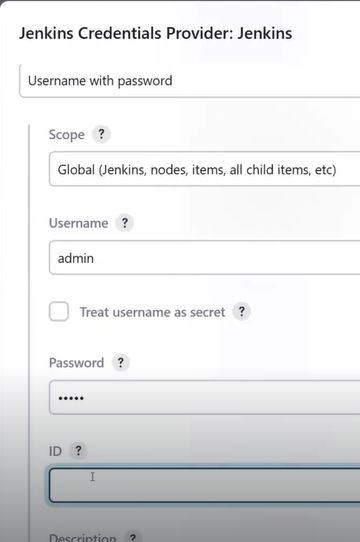
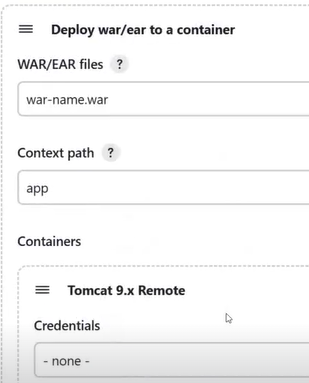
* Our build get success.



* 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

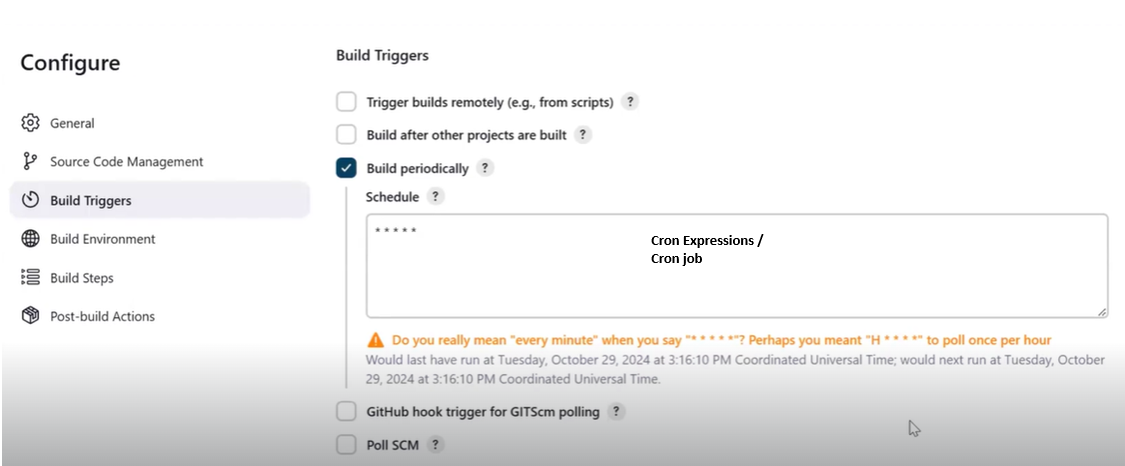
What jenkins do is copying war file from Jenkins to the tomcat webapps folder

* 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**
* ****

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.**

=====================================================================-

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.

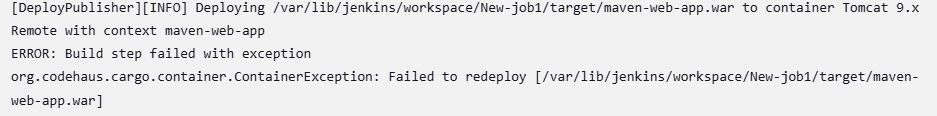
Go to Jenkins Dashboard -> Manage Jenkins --> Manage Plugins -> Goto Available Tab -> Search For "Deploy To Container" Plugin -> Install without restart.

=====================================================================

**Issue – Redeploy failed**

=====================================================================

If you got below error



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

=====================================================================-

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'

}

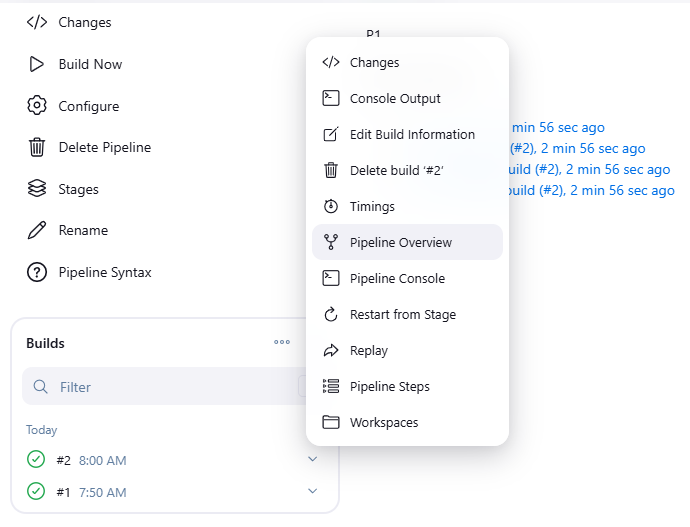
}

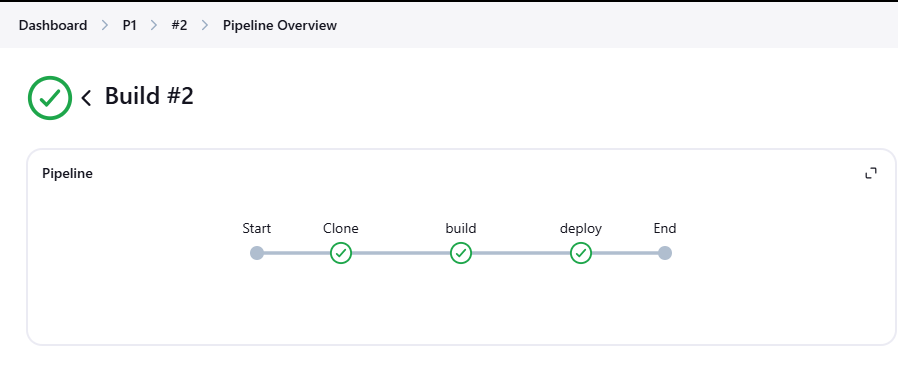
}

}

How to create simple pipeline in Jenkins ?

* Create new job
* go to pipeline
* Select pipeline script
* Save & apply





These are the dummy stages

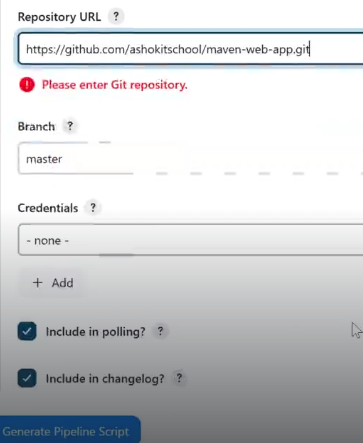
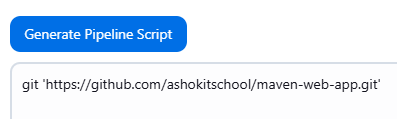
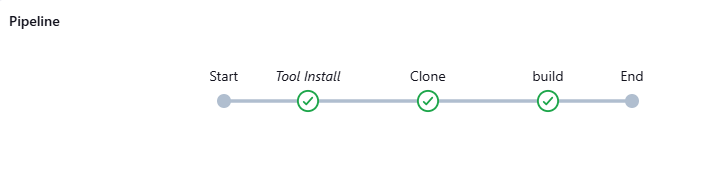
Additional Script

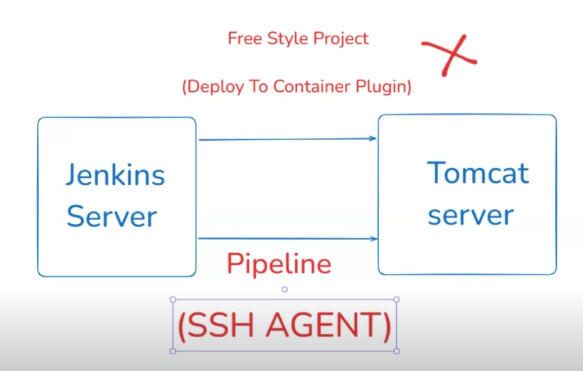


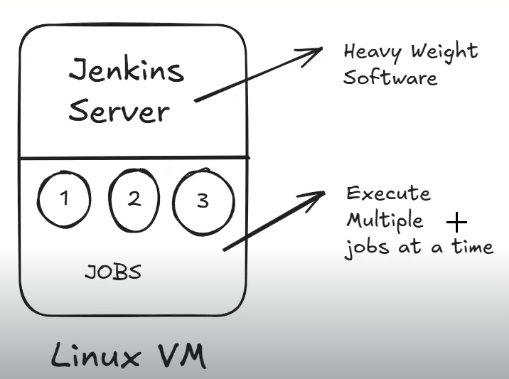
If you want to integrate git and maven in Jenkins pipeline?

* 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
* 
* Check pipeline overview
* 





**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.

=====================================================================-

Jenkins Master

=====================================================================-

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

=> It is used to create the jobs

=> 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.

=====================================================================-

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.

=====================================================================-

Step-1 : Create Jenkins Master vm

=====================================================================-

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)

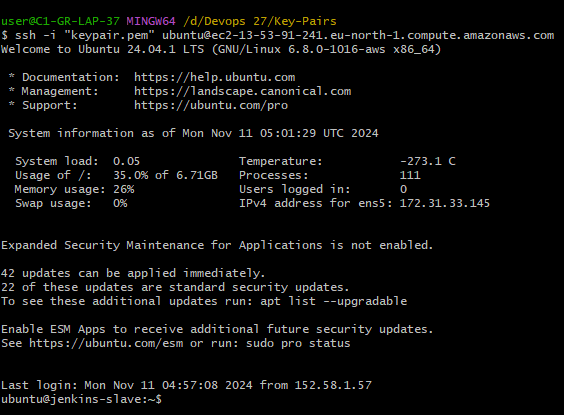
2) Connect to EC2 using ssh client

3) Change hostname for readability

$ sudo hostname jenkins-slave

$ exit and connect back

* After connecting back you will get below output

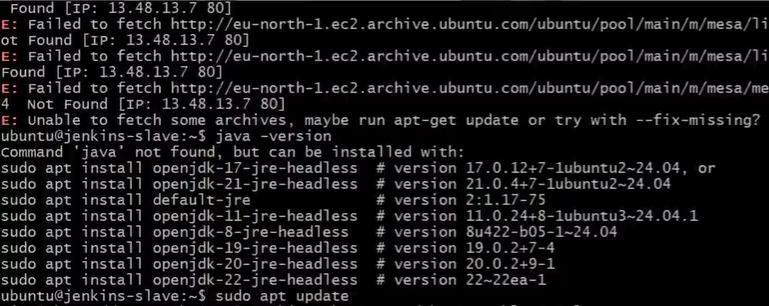


3) Install Java Software

$ sudo apt install default-jre

If you encounter below error use command- sudo apt update

* Apt is a package manager for Ubuntu machine.



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

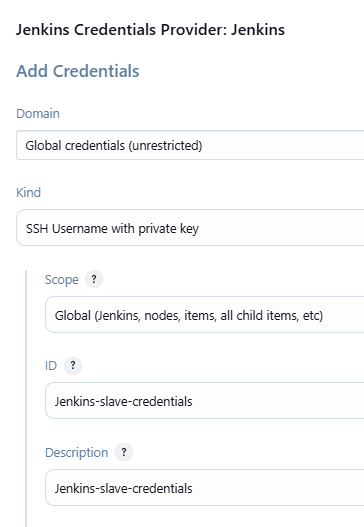
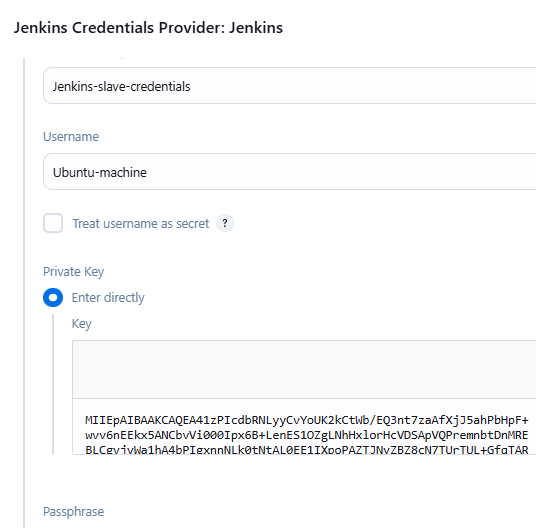
5) Enter Remote Root Directory ( /home/ubuntu/slavenode )

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)

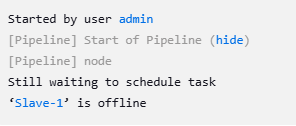
=====================================================================-

Now we need to run our job on slave node

* Go to the job ­­and edit pipeline script
* 
* My job should be run on slave machine not on the master machine, we have assigned the task to the Slave-1 machine.

Issue

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



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

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