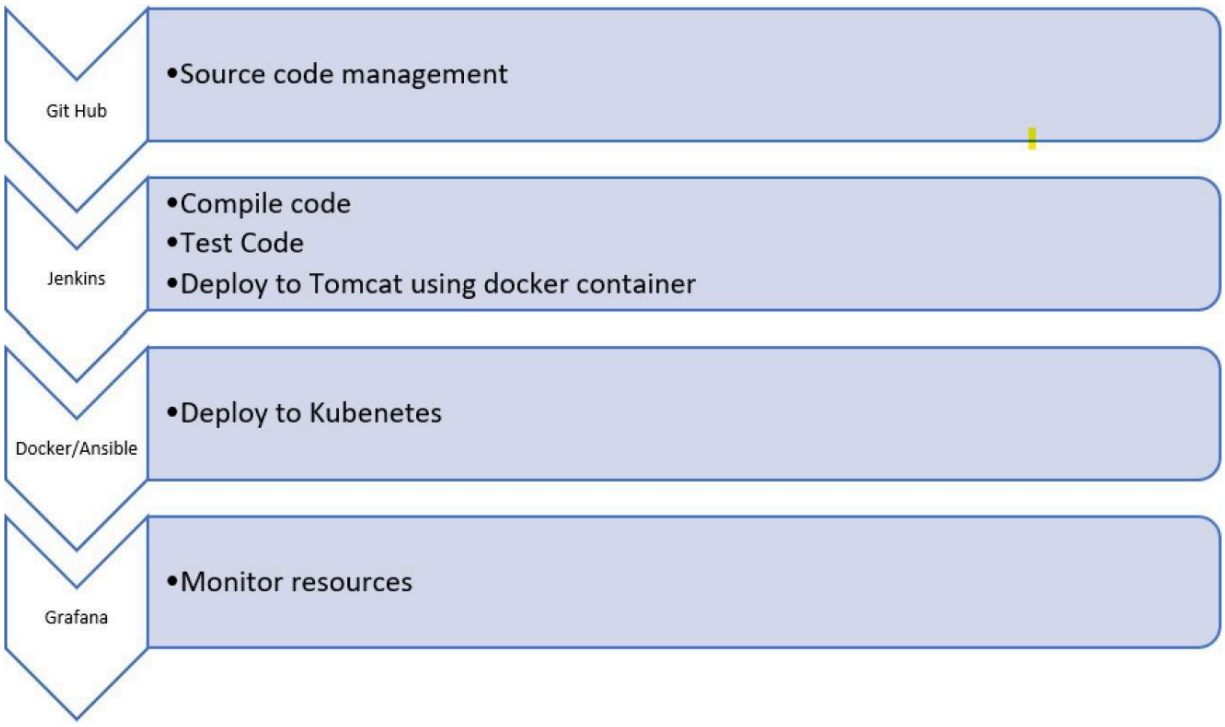
Industry Grade Project

Post Graduate Program in DevOps

Divya Venkataraman

Building a CI-CD pipeline

The Process Flow to be followed to achieve this.



Tasks and their Explanation that will be done.

- Push the code to our GitHub repository.
- Create a continuous integration pipeline using Jenkins to compile, test, and package the code
- Write Dockerfile to push the war file to the Tomcat server.
- Integrate Docker with Ansible and write the playbook.
- Deploy artifacts to the Kubernetes cluster
- Monitor resources using Grafana.

What Softwares will be installed? Resources Required

1. Java
2. Maven
3. Git
4. Jenkins
5. Docker
6. Ansible
7. Kubernetes
8. Grafana
9. Prometheus

1. AWS Account
2. Putty/MobaXTerm
3. Github account
4. Docker-hub account

We will also add details on the same document on how these installations and resource setups were done.

Task 1:

Clone the project from the GitHub link shared in resources to your local machine. Build the code using Maven commands.

Steps done to complete the task.

1. Creating a git hub account.
2. Clone the project from Edureka resources to your IDE.
3. Installing git bash client and push the code to your github account
4. Verify the status of the push.
5. Installing maven
6. Build the code with maven

Creating a GIT hub account

Go to https://github.com/join to create a GitHub account.

You'll need to verify your email during the signup process.

Select the free plan account and create one.

Successfully created github account.

Also created a initial repository with only a readme file.

https://github.com/DivyaBharathwaj/Purdue_Project/blob/master/README.md

Installed Git bash on my PC. Git BASH:: Git for Windows provides a BASH emulation used to run Git from the command line which is same as the linux env.

The git client is then used from our IDE -Intellij in my case. Copied the project from Edureka to my local

Committed and pushed the repo to my github remote repo using the git commands. Git commands executed are:

git init for initializing a local repository

git add . to add all your files that the local repository

git commit -m 'commit message' to save the changes you made to those files

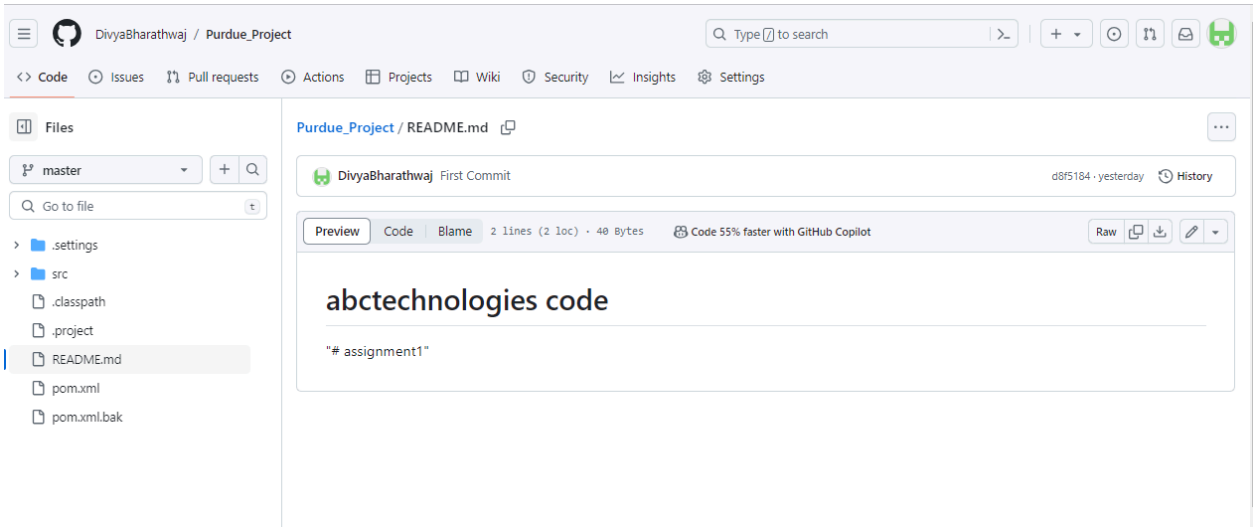
git remote add origin "https://github.com/DivyaBharathwaj/Purdue_Project.git"

git push -u origin main – This pushes my code in the local to the main branch in the remote.

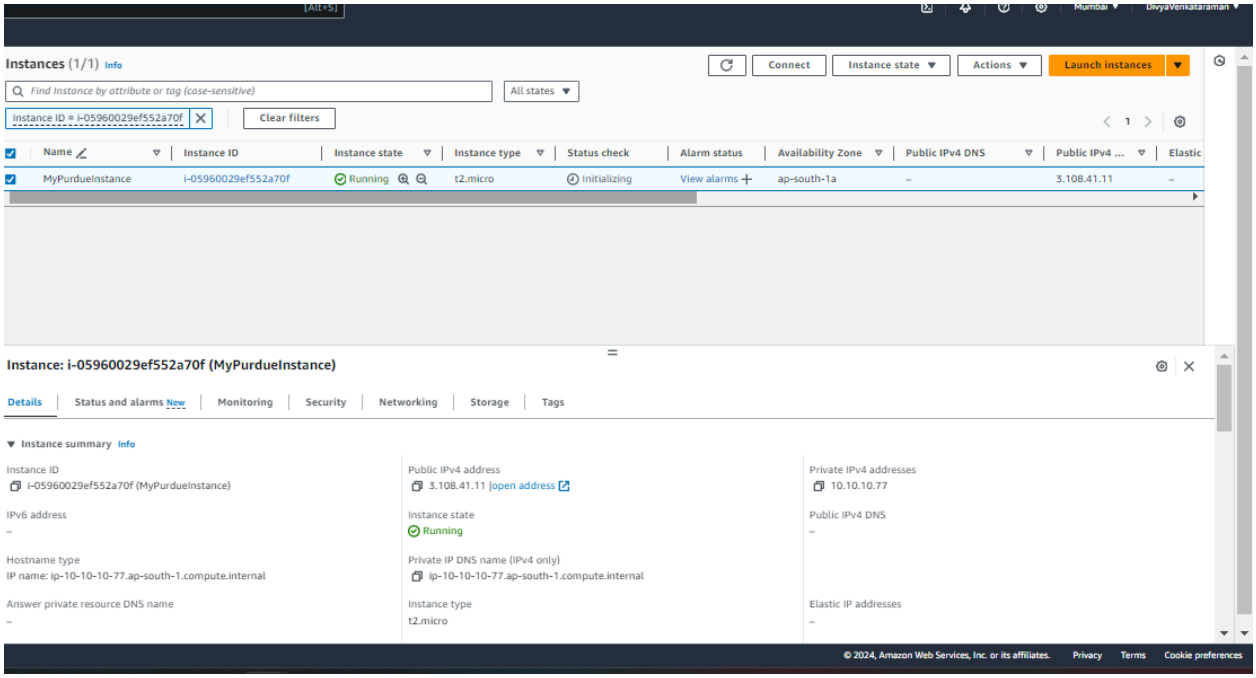
The Git logs can be seen below in Git bash:

```
Divya Venkataraman@DESKTOP-ACLEQ51 MINGW64 ~/Desktop/Edureka Project/Industry Grade Project I - Java Project/ABC Te
chnologies (master)
$ git init
Reinitialized existing Git repository in C:/Users/Divya Venkataraman/Desktop/Edureka Project/Industry Grade Project
 I - Java Project/ABC Technologies/.git/

Divya Venkataraman@DESKTOP-ACLEQ51 MINGW64 ~/Desktop/Edureka Project/Industry Grade Project I - Java Project/ABC Te
chnologies (master)
$ git add .
warning: in the working copy of 'src/main/webapp/WEB-INF/web.xml', LF will be replaced by CRLF the next time Git to
uches it
warning: in the working copy of 'src/main/webapp/index.jsp', LF will be replaced by CRLF the next time Git touches
it

Divya Venkataraman@DESKTOP-ACLEQ51 MINGW64 ~/Desktop/Edureka Project/Industry Grade Project I - Java Project/ABC Te
chnologies (master)
$ git commit -m "First commit"
On branch master
nothing to commit, working tree clean

Divya Venkataraman@DESKTOP-ACLEQ51 MINGW64 ~/Desktop/Edureka Project/Industry Grade Project I - Java Project/ABC Te
chnologies (master)
$ git remote add origin "https://github.com/DivyaBharathwaj/Purdue_Project.git"
error: remote origin already exists.

Divya Venkataraman@DESKTOP-ACLEQ51 MINGW64 ~/Desktop/Edureka Project/Industry Grade Project I - Java Project/ABC Te
chnologies (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 8 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (29/29), 4.44 KiB | 505.00 KiB/s, done.
Total 29 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DivyaBharathwaj/Purdue_Project.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

Divya Venkataraman@DESKTOP-ACLEQ51 MINGW64 ~/Desktop/Edureka Project/Industry Grade Project I - Java Project/ABC Te
chnologies (master)
```

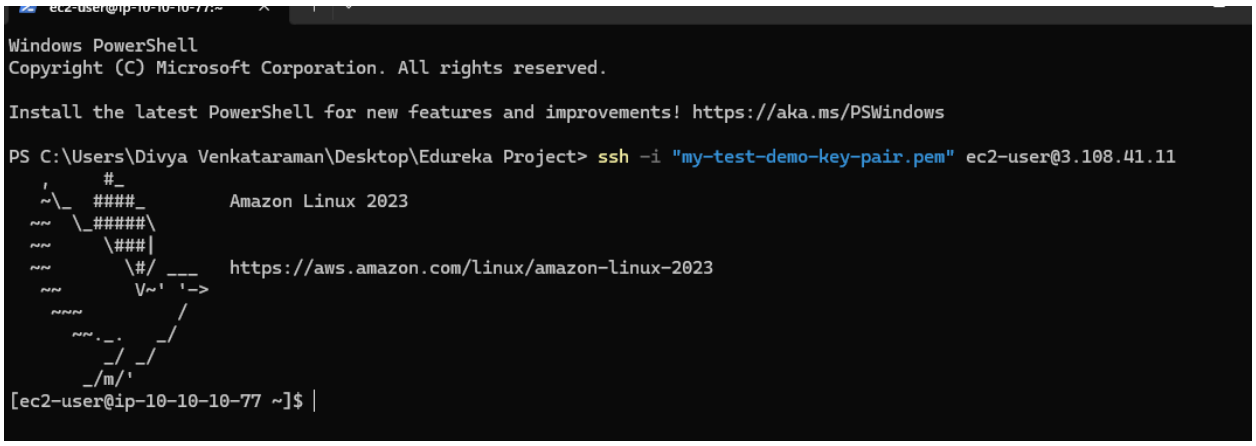
Verified the Github whether commit and push was successful.



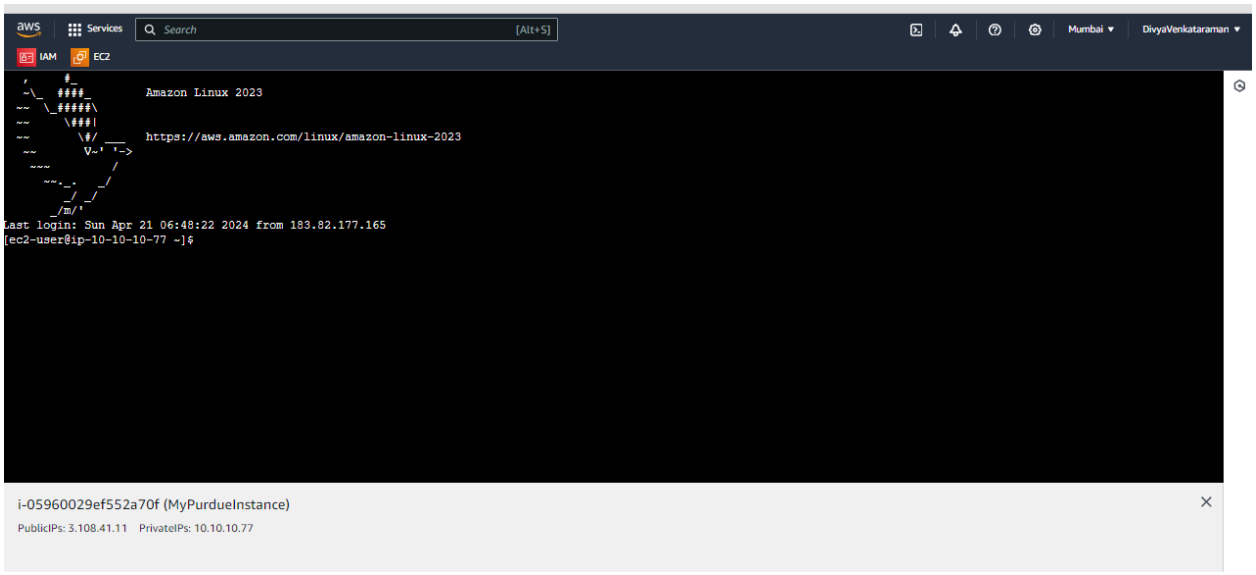
Launch an EC2 instance:



Connecting to the instance via Windows PowerShell:



Can also use the connect ssh client facility available in the EC2 portal which is much simpler in process.

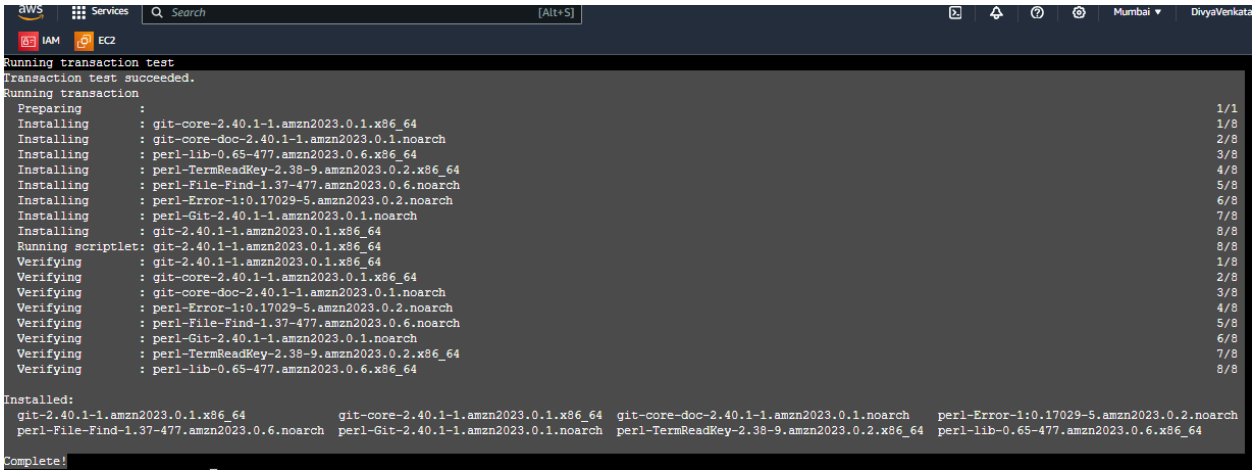


Command to verify the OS version : cat /etc/os-release.

```
Last login: Sun Apr 21 06:48:22 2024 from 183.82.177.165
[ec2-user@ip-10-10-10-77 ~]$ cat /etc/os-release
NAME="Amazon Linux"
VERSION="2023"
ID="amzn"
ID_LIKE="fedora"
VERSION_ID="2023"
PLATFORM_ID="platform:al2023"
PRETTY_NAME="Amazon Linux 2023.4.20240416"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2023"
HOME_URL="https://aws.amazon.com/linux/amazon-linux-2023/"
DOCUMENTATION_URL="https://docs.aws.amazon.com/linux/"
SUPPORT_URL="https://aws.amazon.com/premiumsupport/"
BUG_REPORT_URL="https://github.com/amazonlinux/amazon-linux-2023"
VENDOR_NAME="AWS"
VENDOR_URL="https://aws.amazon.com/"
SUPPORT_END="2028-03-15"
[ec2-user@ip-10-10-10-77 ~]$
```

Installing GIT and MAVEN and Run MVN commands

Step 1:: Install Git -> Command : sudo yum install git -y



Step 2::clone the repo in ec2 ->Command –
git clone https://github.com/DivyaBharathwaj/Purdue_Project.git

[ec2-user@ip-10-10-10-77 ~]$ git clone
https://github.com/DivyaBharathwaj/Purdue_Project.git

Cloning into 'Purdue_Project'...

remote: Enumerating objects: 29, done.

remote: Counting objects: 100% (29/29), done.

remote: Compressing objects: 100% (20/20), done.

remote: Total 29 (delta 0), reused 29 (delta 0), pack-reused 0

Receiving objects: 100% (29/29), 4.44 KiB | 2.22 MiB/s, done.

Committed the installation script to git repo – file name git/installgit.sh

Maven What is Maven?

Maven is a build automation tool used primarily for Java projects.
Maven can also be used to build and manage projects written in C#,
Ruby, Scala, and other languages.

#Install maven in the server

sudo wget http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo

sudo sed -i s/\$releasever/6/g

/etc/yum.repos.d/epel-apache-maven.repo sudo yum

install -y apache-maven

Committed the installation script to git repo – file name maven/installmaven.sh

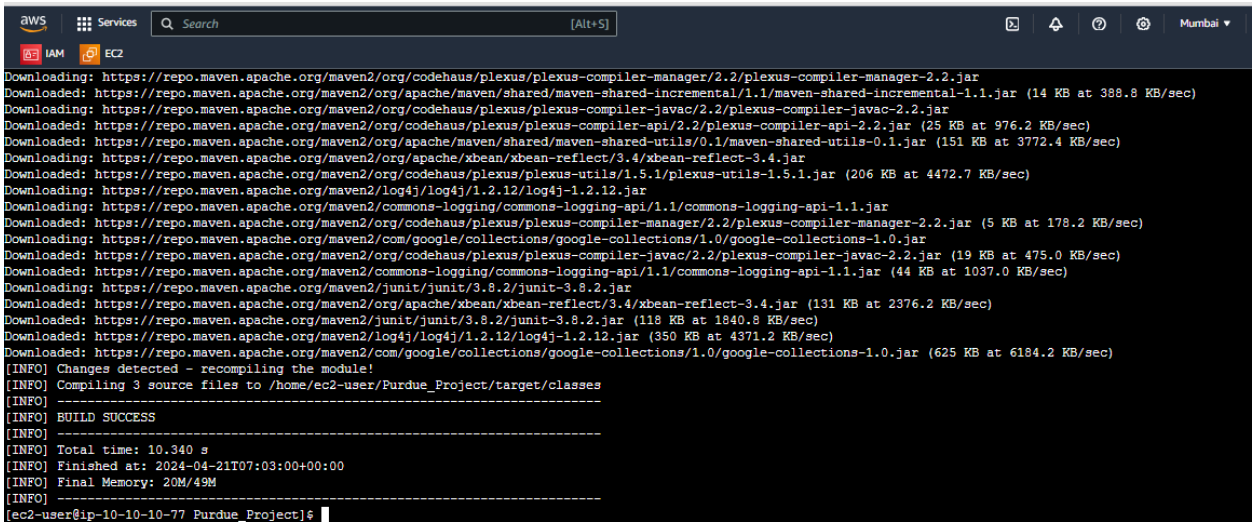
```
Installed:
  alsa-lib-1.2.7.2-1.amzn2023.0.2.x86_64                apache-maven-3.2.5-1.el6.noarch
  cairo-1.17.6-2.amzn2023.0.1.x86_64                    dejavu-sans-fonts-2.37-16.amzn2023.0.2.noarch
  dejavu-sans-mono-fonts-2.37-16.amzn2023.0.2.noarch    dejavu-serif-fonts-2.37-16.amzn2023.0.2.noarch
  fontconfig-2.13.94-2.amzn2023.0.2.x86_64              fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noarch
  freetype-2.13.0-2.amzn2023.0.1.x86_64                 giflib-5.2.1-9.amzn2023.0.1.x86_64
  google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
  graphite2-1.3.14-7.amzn2023.0.2.x86_64                harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
  java-17-amazon-corretto-headless-1:17.0.10+8-1.amzn2023.1.x86_64 java-22-amazon-corretto-1:22.0.0+37-1.amzn2023.1.x86_64
  java-22-amazon-corretto-devel-1:22.0.0+37-1.amzn2023.1.x86_64 java-22-amazon-corretto-headless-1:22.0.0+37-1.amzn2023.1.x86_64
  javapackages-filesystem-6.0.0-7.amzn2023.0.6.noarch   javapackages-tools-6.0.0-7.amzn2023.0.6.noarch
  langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch     libICE-1.0.10-6.amzn2023.0.2.x86_64
  libSM-1.2.3-8.amzn2023.0.2.x86_64                     libX11-1.7.2-3.amzn2023.0.4.x86_64
  libX11-common-1.7.2-3.amzn2023.0.4.noarch             libXau-1.0.9-6.amzn2023.0.2.x86_64
  libXext-1.3.4-6.amzn2023.0.2.x86_64                   libXi-1.7.10-6.amzn2023.0.2.x86_64
  libXinerama-1.1.4-8.amzn2023.0.2.x86_64               libXrandr-1.5.2-6.amzn2023.0.2.x86_64
  libXrender-0.9.10-14.amzn2023.0.2.x86_64              libXt-1.2.0-4.amzn2023.0.2.x86_64
  libXtst-1.2.3-14.amzn2023.0.2.x86_64                  libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64             libpng-2:1.6.37-10.amzn2023.0.6.x86_64
  libxcb-1.13.1-7.amzn2023.0.2.x86_64                   pixman-0.40.0-3.amzn2023.0.3.x86_64
  xml-common-0.6.3-56.amzn2023.0.2.noarch
Skipped:
  apache-maven-3.5.2-1.el6.noarch

Complete!
```

Basic Maven Commands :

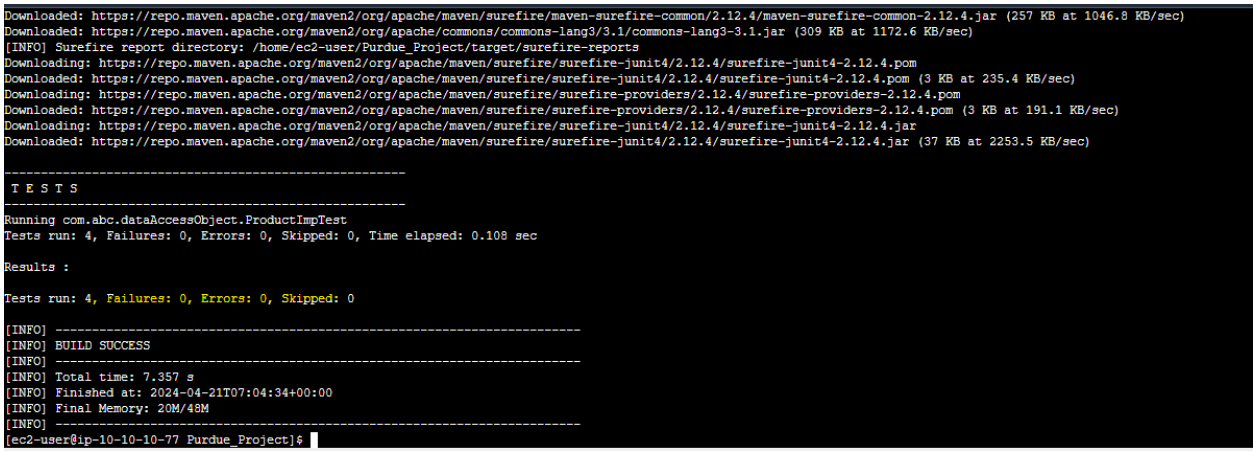
1) mvn --version to find the version of the maven install

```
[ec2-user@ip-10-10-10-77 ~]$ mvn --version
Apache Maven 3.2.5 (12a6b3acb947671f09b81f49094c53f426d8cea1; 2014-12-14T17:29:23+00:00)
Maven home: /usr/share/apache-maven
Java version: 22, vendor: Amazon.com Inc.
Java home: /usr/lib/jvm/java-22-amazon-corretto.x86_64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.1.84-99.169.amzn2023.x86_64", arch: "amd64", family: "unix"
```

2) mvn compile – to compile the source code.



3) mvn test - to run the test of the project



4) mvn package – to build and package the artifacts of the project to jar or war

mvn package from ec2 :: added war plugin in the pom.xml and verified the /target directory

```
[INFO] Loading execution data file /home/ec2-user/Purdue_Project/target/jacoco.exec
[INFO] Analyzed bundle 'RetailModule' with 2 classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 8.727 s
[INFO] Finished at: 2024-04-21T07:06:15+00:00
[INFO] Final Memory: 23M/98M
[INFO] ------------------------------------------------------------------------
[ec2-user@ip-10-10-10-77 Purdue_Project]$ ls -lrt /home/ec2-user/Purdue_Project/target/ABCtechnologies-1.0
total 4
-rw-r--r--. 1 ec2-user ec2-user 519 Apr 21 06:54 index.jsp
drwxr-xr-x. 2 ec2-user ec2-user   6 Apr 21 07:06 META-INF
drwxr-xr-x. 4 ec2-user ec2-user  47 Apr 21 07:06 WEB-INF
[ec2-user@ip-10-10-10-77 Purdue_Project]$ ls -lrt /home/ec2-user/Purdue_Project/target/
total 6980
drwxr-xr-x. 3 ec2-user ec2-user      35 Apr 21 07:02 maven-status
drwxr-xr-x. 3 ec2-user ec2-user      25 Apr 21 07:02 generated-sources
drwxr-xr-x. 3 ec2-user ec2-user      17 Apr 21 07:03 classes
drwxr-xr-x. 3 ec2-user ec2-user      30 Apr 21 07:04 generated-test-sources
drwxr-xr-x. 3 ec2-user ec2-user      17 Apr 21 07:04 test-classes
drwxr-xr-x. 2 ec2-user ec2-user     113 Apr 21 07:04 surefire-reports
-rw-r--r--. 1 ec2-user ec2-user    8646 Apr 21 07:06 jacoco.exec
drwxr-xr-x. 4 ec2-user ec2-user      54 Apr 21 07:06 ABCtechnologies-1.0
drwxr-xr-x. 2 ec2-user ec2-user      28 Apr 21 07:06 maven-archiver
-rw-r--r--. 1 ec2-user ec2-user 7132819 Apr 21 07:06 ABCtechnologies-1.0.war
drwxr-xr-x. 3 ec2-user ec2-user      20 Apr 21 07:06 site
```

-\

END of TASK 1

Task 2: Set up the Git repository and push the source code. Then, log in to Jenkins.

1. Create a build pipeline containing a job for each
   - One for compiling source code
   - Second for testing source code
   - Third for packing the code
2. Execute the CI/CD pipeline to execute the jobs created in step 1
3. Set up a master-slave node to distribute the tasks in the pipeline.

What is Jenkins?

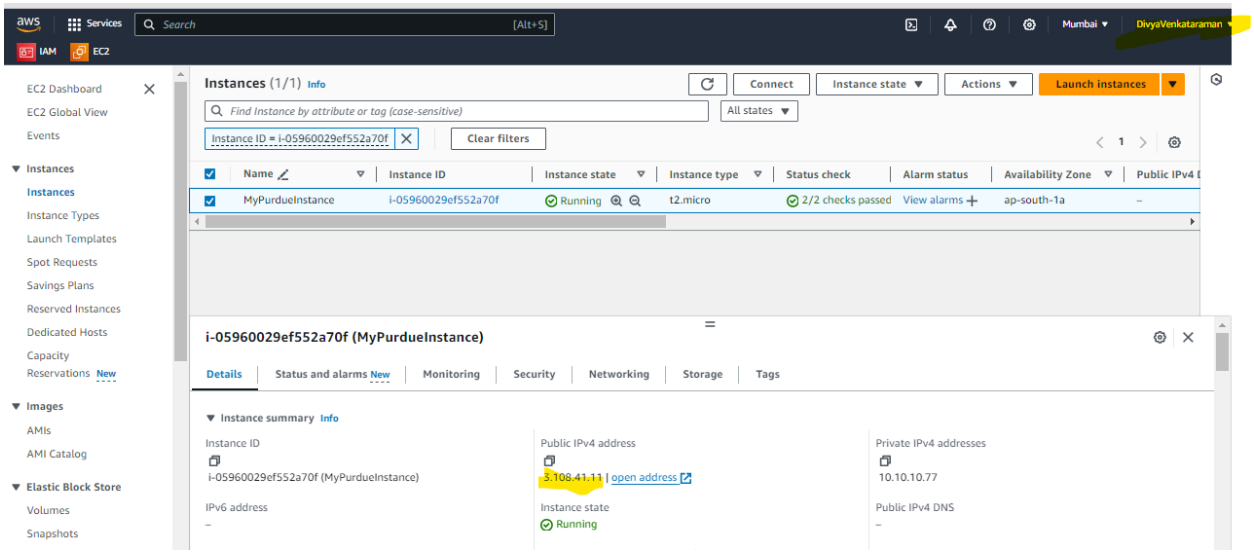
Jenkins is an open-source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery.

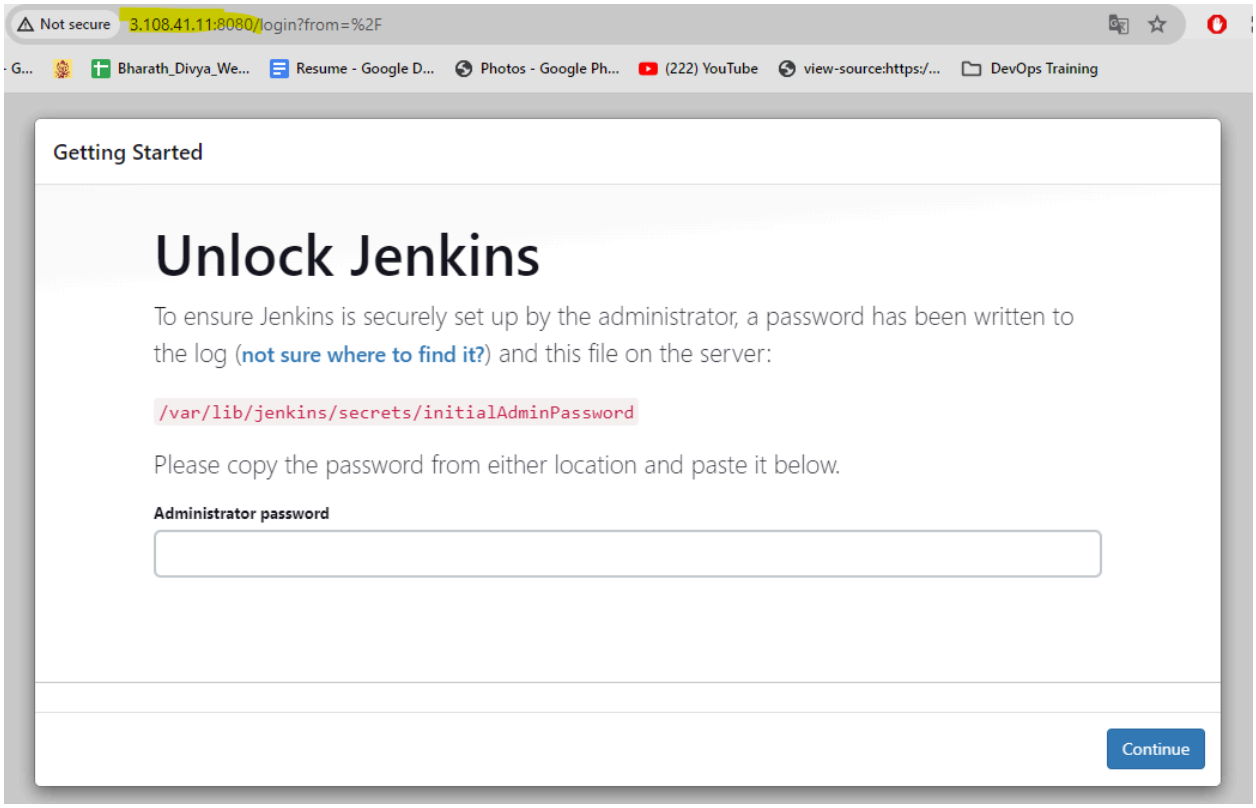
How to install Jenkins?

```
13  sudo dnf update
14  sudo dnf install java-11-amazon-corretto -y
15  sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
16  sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
17  yum install jenkins -y
```

Then the same can be accessed using the public IP : 8080 with admin password.

The screen will show a startup config, where you could choose to install the recommended plugins.

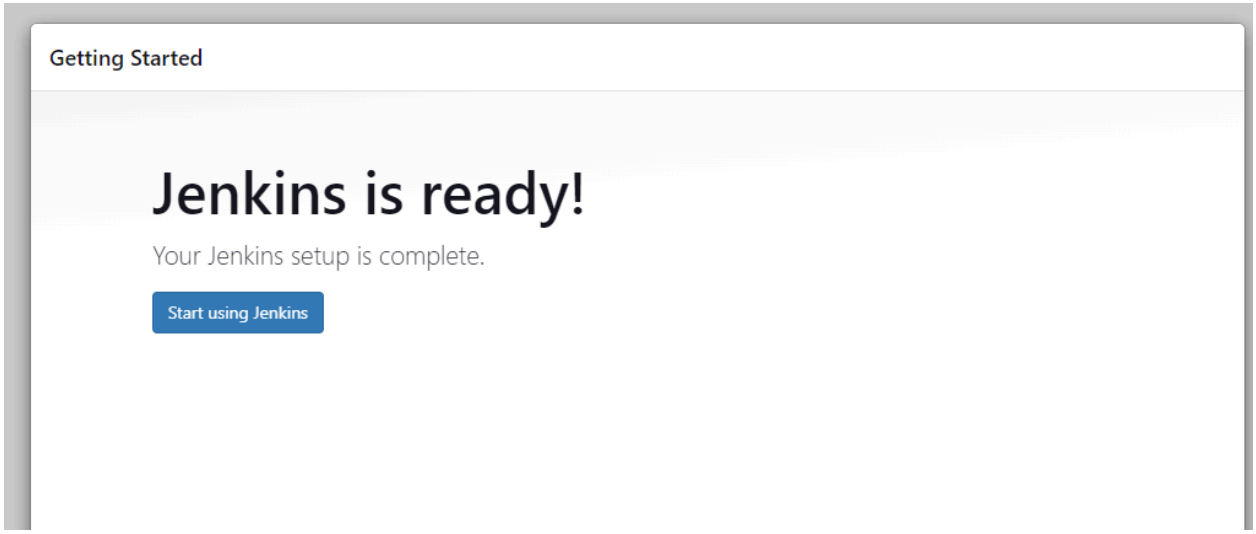




On the ec2 server, execute below command

$ cat /var/lib/jenkins/secrets/initialAdminPassword

Copy the password and paste in the browser (jenkins)



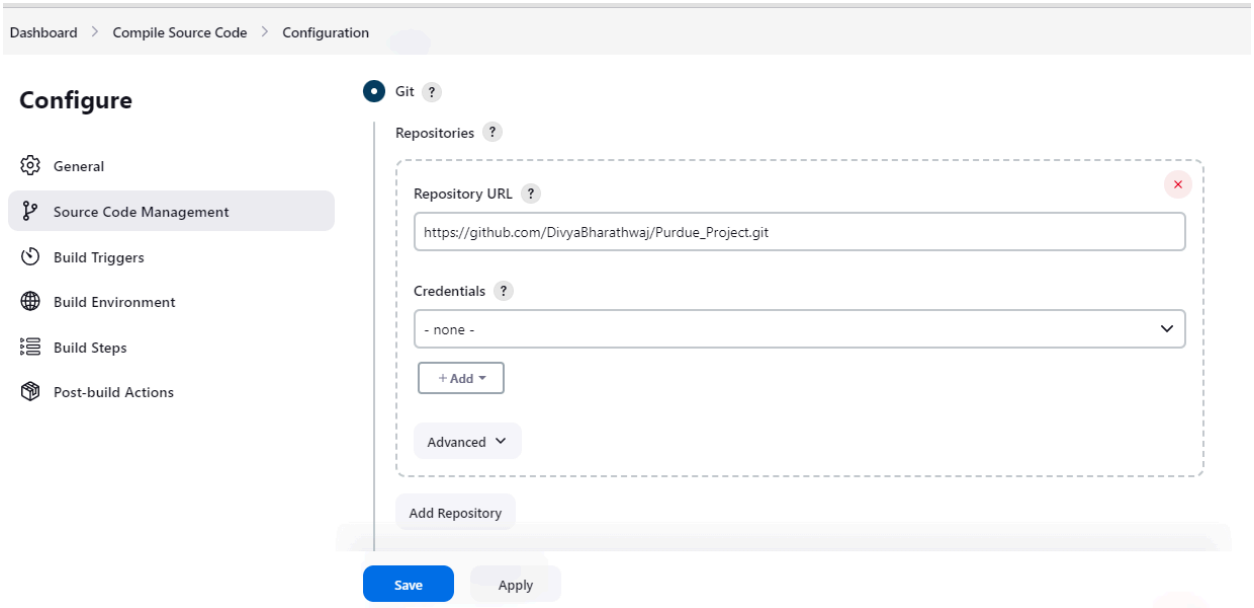
JOB Creation- Source Code compilation

1) compiling source code 2) Test code and 3) Package code

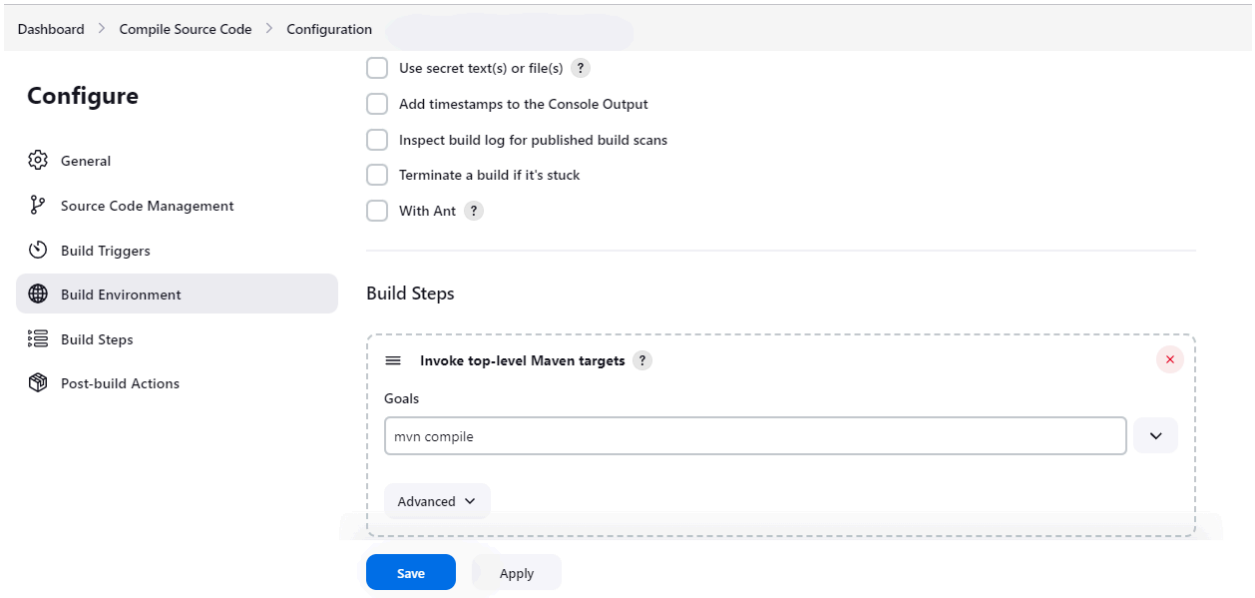
Steps :

1) Click New Item -> give a name to it Like -> compile Source code
2) Then click Freestyle project and save it.
3) We can then click on configure and add the github url from where the source code has to be pulled and then add the maven goals clean and compile

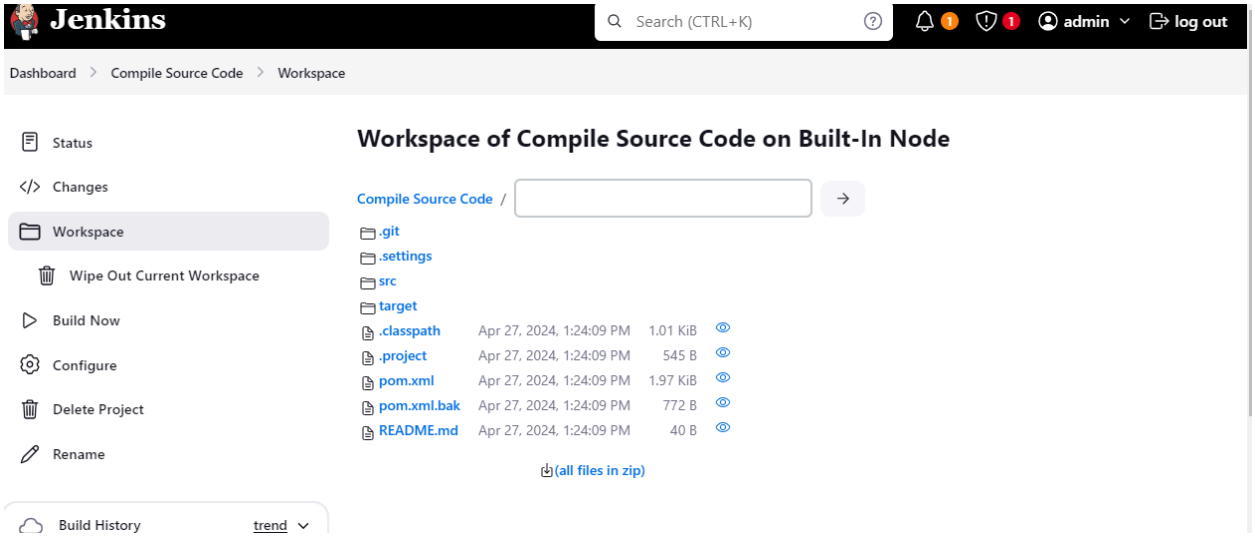
Adding Github details.



Adding Maven Goals – Compile goal is responsible to run mvn compile



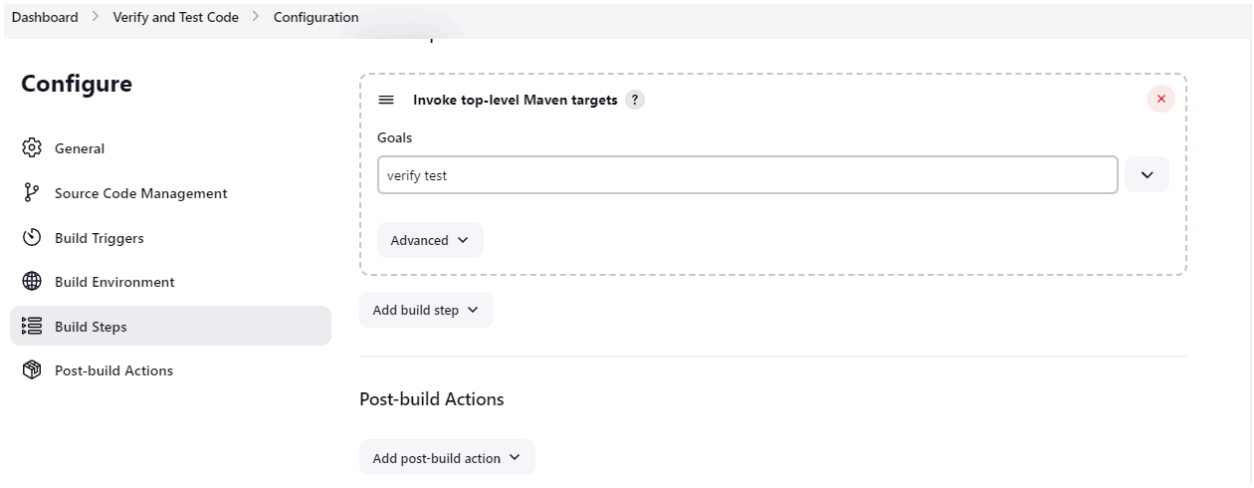
The Compiled source code can be viewed in the target directory of workspace.



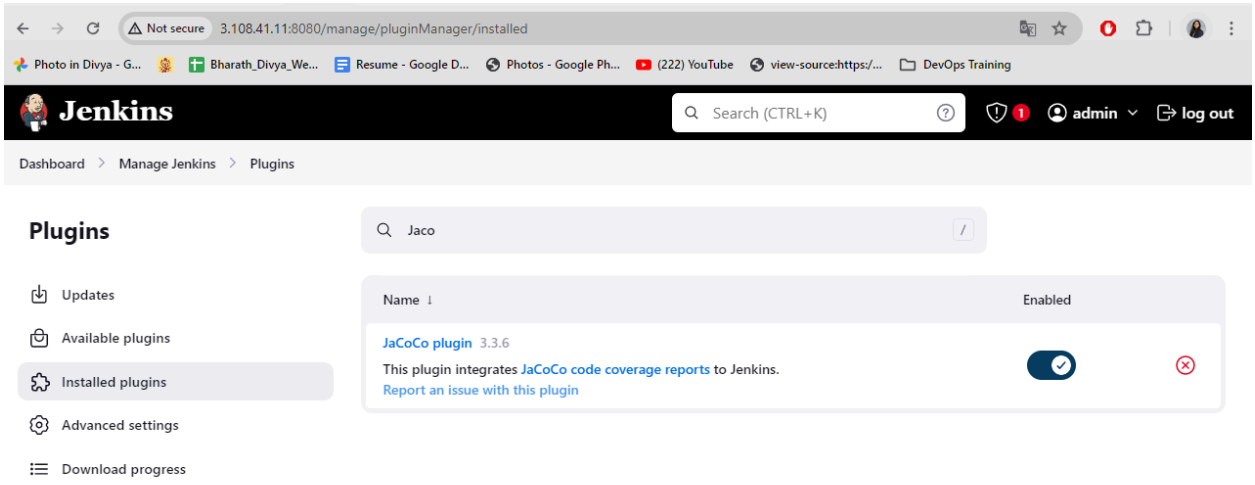
JOB 2 : Review and Test Code

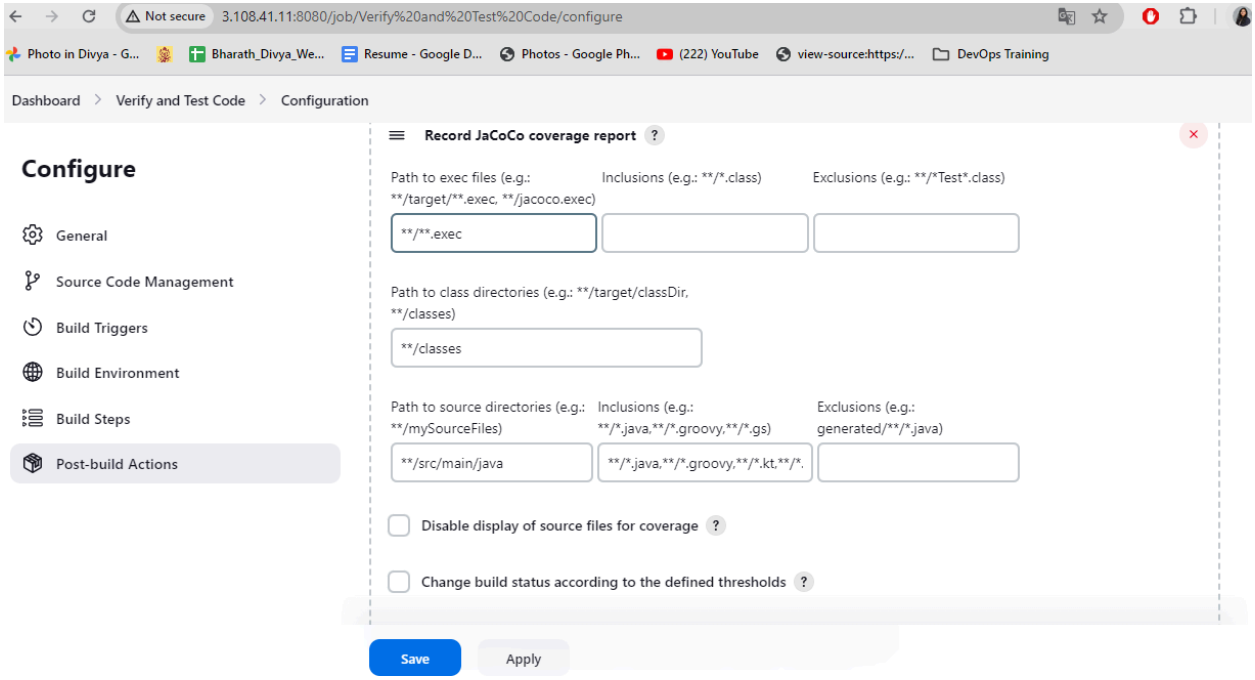
The Github setup is same as the job 1.

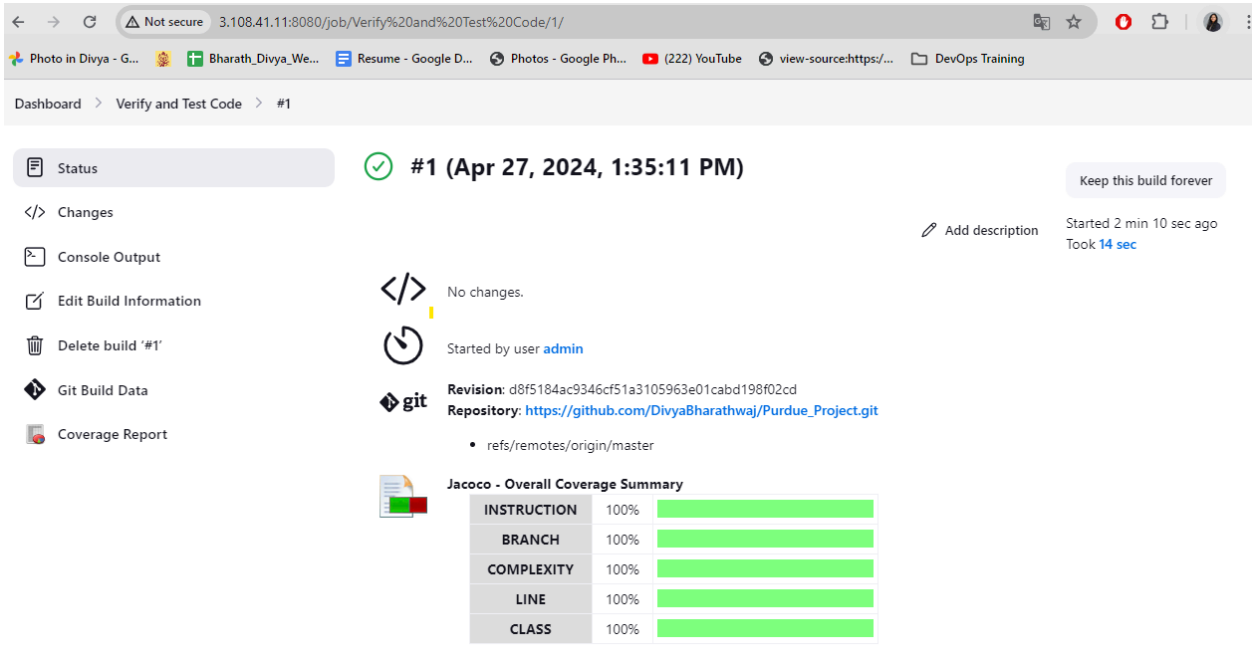
The maven goal is verify and test – which is same as our mvn verify and mvn test commands



Also configuring Jacoco reports in the post build step. We need the jacoco plugin




Not secure 3.108.41.11:8080/job/Verify%20and%20Test%20Code/configure
Photo in Divya - G...
Bharath_Divya_We...
Resume - Google D...
Photos - Google Ph...
(222) YouTube
view-source:https:/...
DevOps Training
Dashboard
Verify and Test Code
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Record JaCoCo coverage report
Path to exec files (e.g.: **/target/**.exec, **/jacoco.exec)
Inclusions (e.g.: **/*.class)
Exclusions (e.g.: **/*Test*.class)
/.exec
Path to class directories (e.g.: **/target/classDir, **/classes)
**/classes
Path to source directories (e.g.: **/mySourceFiles)
Inclusions (e.g.: **/*.java,**/*.groovy,**/*.gs)
Exclusions (e.g.: generated/**/*.java)
**/src/main/java
/*.java,/*.groovy,**/*.kt,**/*.
Disable display of source files for coverage
Change build status according to the defined thresholds
Save
Apply

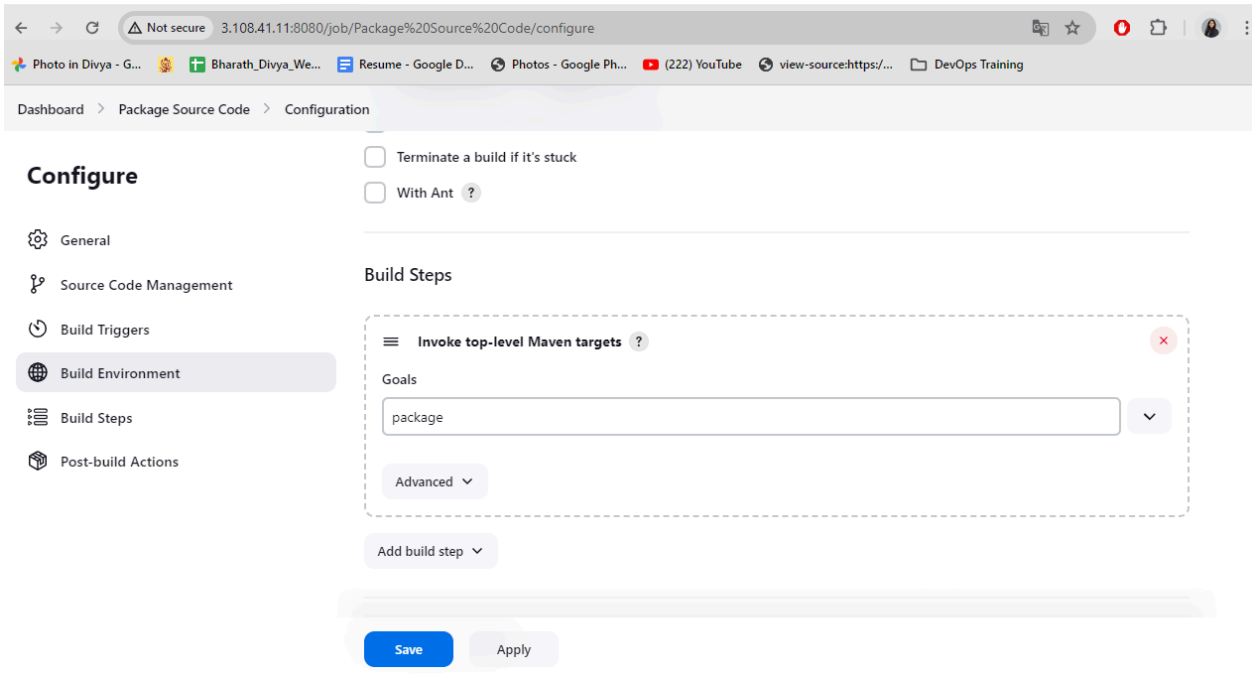

Not secure 3.108.41.11:8080/job/Verify%20and%20Test%20Code/1/
Photo in Divya - G...
Bharath_Divya_We...
Resume - Google D...
Photos - Google Ph...
(222) YouTube
view-source:https:/...
DevOps Training
Dashboard
Verify and Test Code
#1
Status
Changes
Console Output
Edit Build Information
Delete build '#1'
Git Build Data
Coverage Report
#1 (Apr 27, 2024, 1:35:11 PM)
Keep this build forever
Add description
Started 2 min 10 sec ago
Took 14 sec
No changes.
Started by user admin
Revision: d8f5184ac9346cf51a3105963e01cabd198f02cd
Repository: https://github.com/DivyaBharathwaj/Purdue_Project.git
refs/remotes/origin/master
Jacoco - Overall Coverage Summary
INSTRUCTION 100%
BRANCH 100%
COMPLEXITY 100%
LINE 100%
CLASS 100%

JOB 3 : Package Code

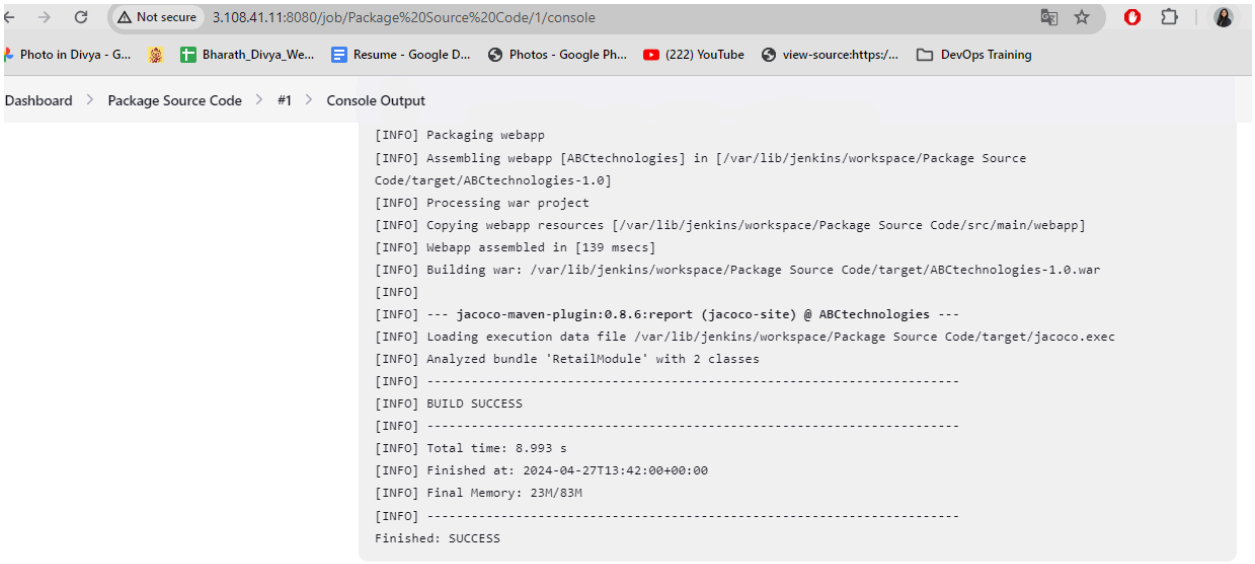
Now package the code into a war file .
The following dependency should be added in the pom.xml if not there already

```
35          <build>
36                  <plugins>
37                          <plugin>
38                  <groupId>org.apache.maven.plugins</groupId>
39                  <artifactId>maven-war-plugin</artifactId>
40                  <version>3.2.2</version>
41                </plugin>
```

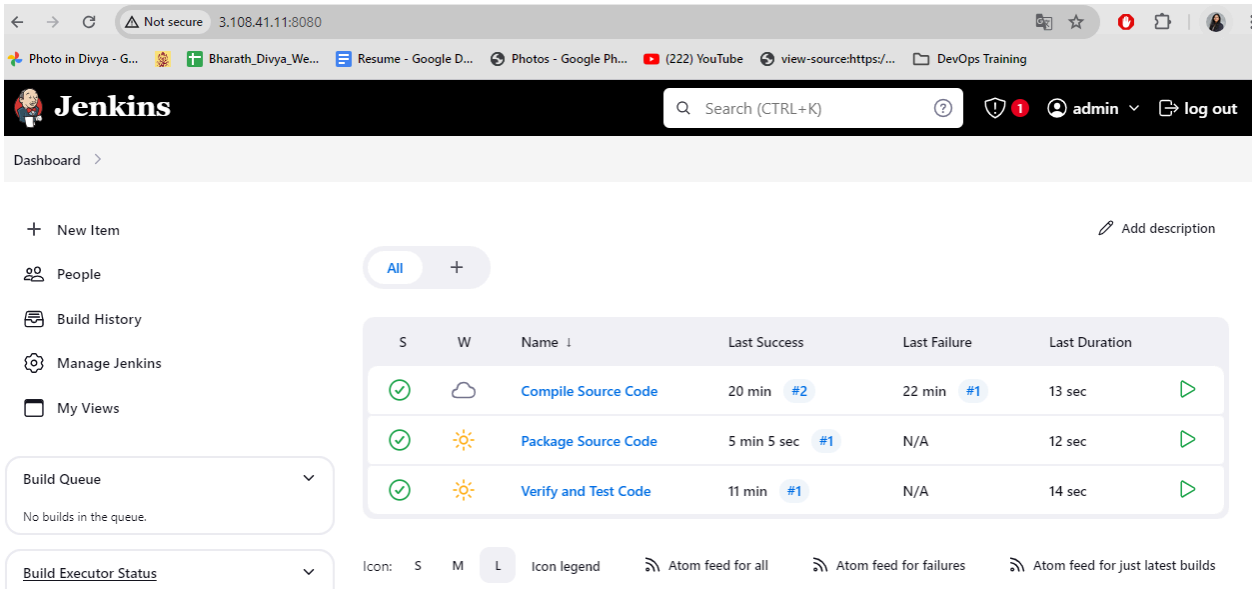
Then repeat the same procedure like above.
Create a new freestyle project and add Github connectivity.
Then in maven goals add package – this is equivalent to command mvn package



Running this will ensure that war is created in the target folder. This can be used to deploy to our tomcat server and launch the application.

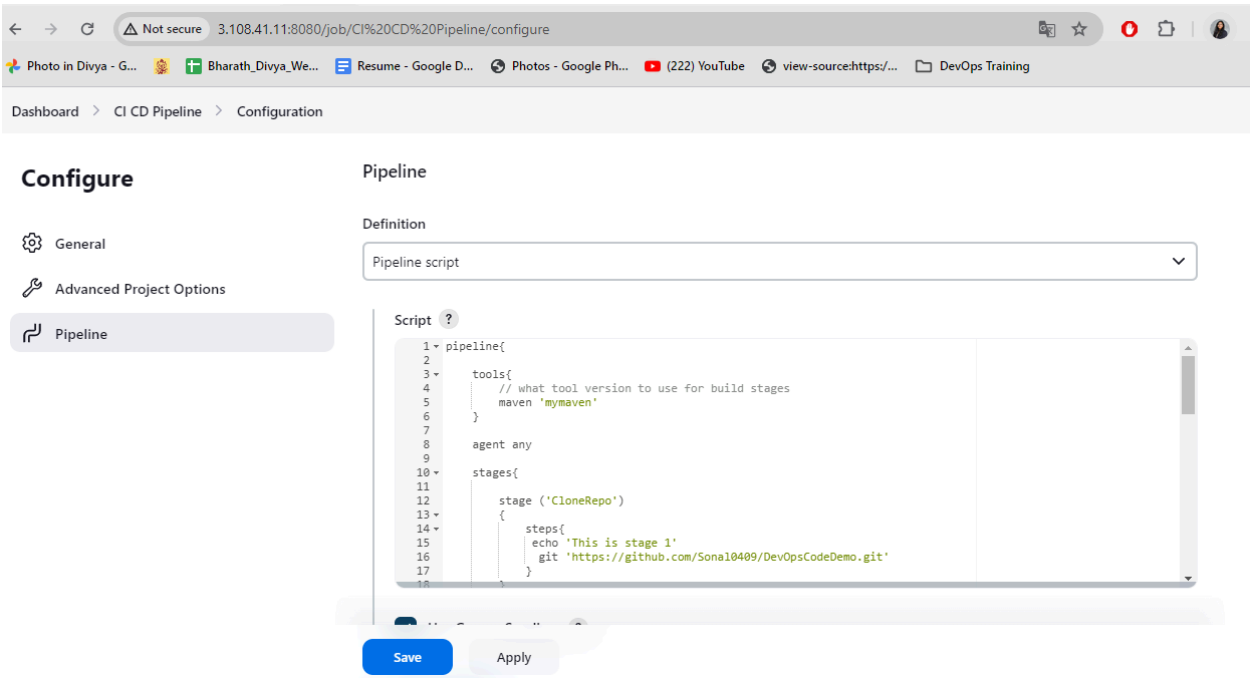


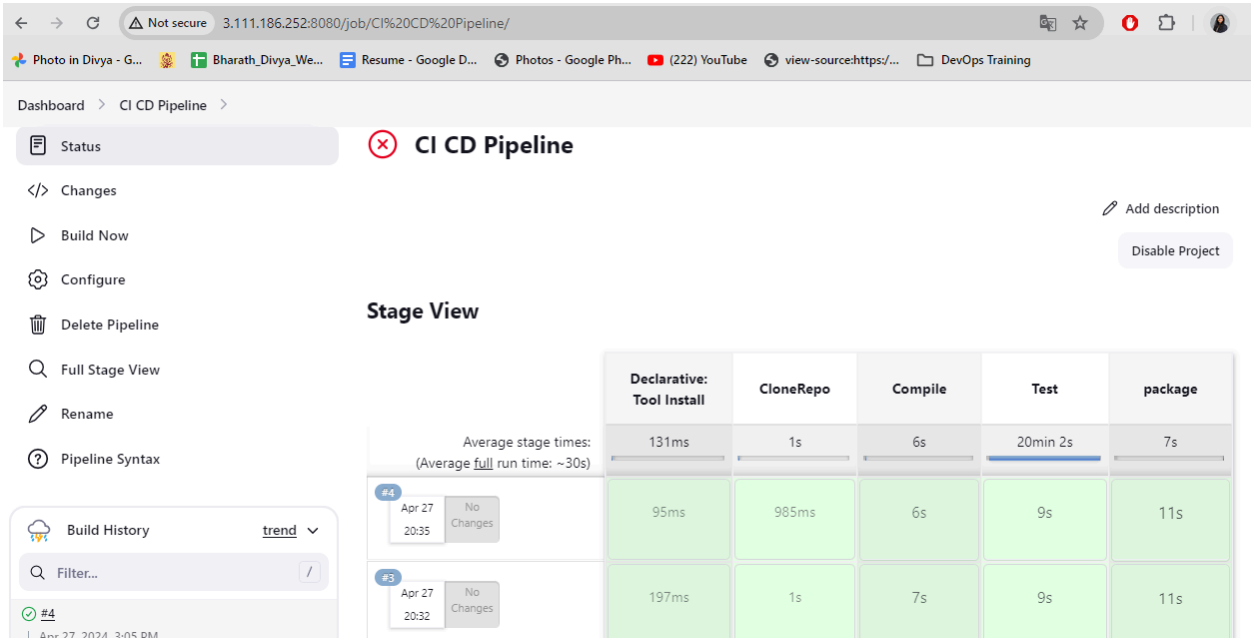
All 3 created jobs in a glance:



The above three jobs can be integrated into a pipeline.

Create new item Pipeline provide the suitable name [I have given here CI Cd pipeline] The commands to run the three tasks are coming from the pipeline script





Master-Slave setup for jenkins

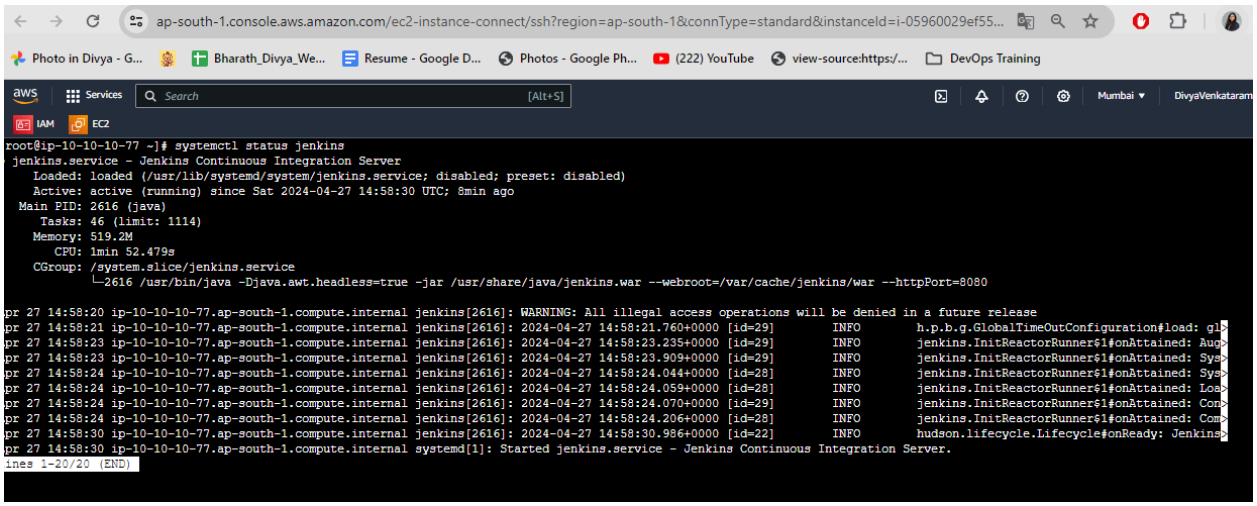
Steps -

Installed Java on master node

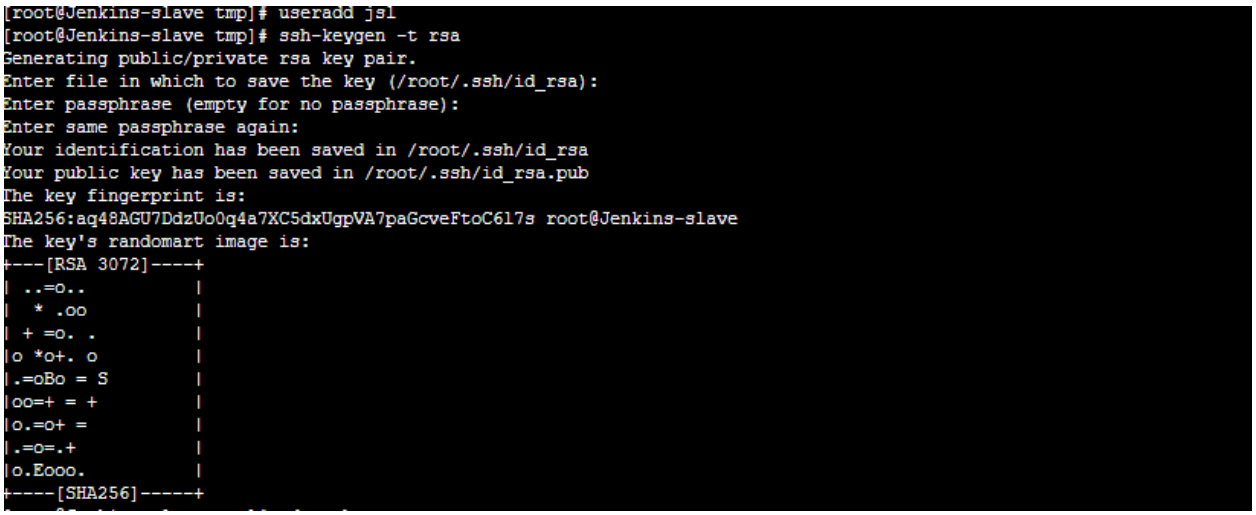
Installed

Jenkins,Java,Git,Maven on
master node Installed java,Git
and Maven on slave node
Created a user and ssh keys on
slave node
Copy keys on master node and join Slave with MAsster
Create Jenkins File – added slave labels to stages that should be run by slave.

Jenkins Running on master::

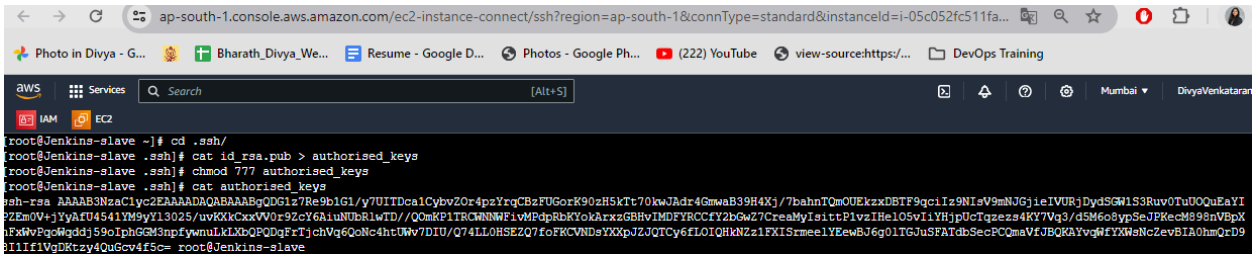


Java installed in slave and created ssh key and user:



[root@Jenkins-slave ~]# cd .ssh/

[root@Jenkins-slave .ssh]# cat id_rsa.pub > authorised_keys
[root@Jenkins-slave .ssh]# chmod 777 authorised_keys
[root@Jenkins-slave .ssh]# cat authorised_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDG1z7Re9b1G1/y7UITDca1CybvZOr4pzYrq
CBzFUGorK90zH5kTt70kwJAdr4GmwaB39H4Xj/7bahnTQmOUEkzxDBTF9qciIz9NIsV9
mNJGjieIVURjDydSGW1S3Ruv0TuUOQuEaYIPZEm0V+jYyAfU4541YM9yYl3025/uvKX
kCxxVV0r9ZcY6AiuNUbRlwTD//QOmKP1TRCWNNWFivMPdpRbKYokArxzGBHvIMDF
YRCCfY2bGwZ7CreaMyIsittP1vzIHelO5vIiYHjpUcTqzezs4KY7Vq3/d5M6o8ypSeJPKec
M898nVBpXnFxWvPqoWqddj59oIphGGM3npfywnuLkLXbQPQDqFrTjchVq6QoNc4htU
Wv7DIU/Q74LL0HSEZQ7foFKCVNDsYXXpJZJQTCy6fLOIQHkNZz1FXISrmeelYEewBJ
6g0lTGJuSFATdbSecPCQmaVfJBQKAYvqWfYXWsNcZevBIA0hmQrD98I1If1VgDKtzy4
QuGcv4f5c= root@Jenkins-slave



Copy keys in master

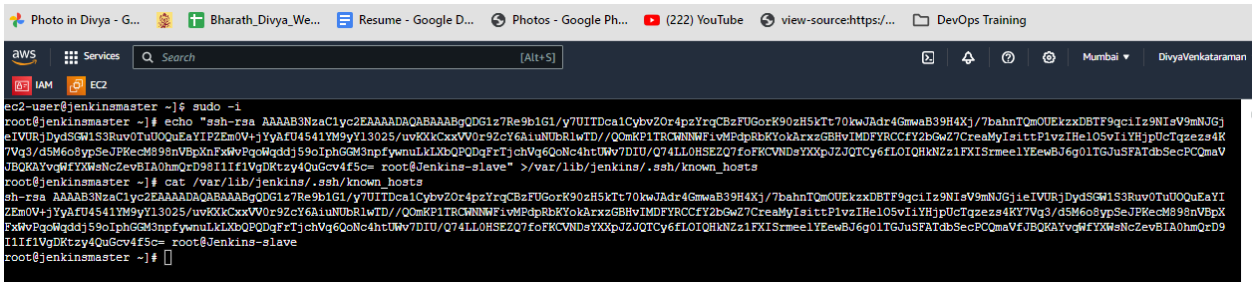
sudo mkdir /var/lib/jenkins/.ssh sudo -i

chown jenkins:jenkins
/var/lib/jenkins/.ssh/

[root@jenkinsmaster ~]# echo "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDG1z7Re9b1G1/y7UITDca1CybvZOr4pzYrqCBzFUGorK90zH5kTt70kwJAdr4GmwaB
39H4Xj/7bahnTQmOUEkzxDBTF9qciIz9NIsV9mNJGjieIVURjDydSGW1S3Ruv0TuUOQuEaYIPZEm0V+jYyAfU4541YM9yYl30
25/uvKXkCxxVV0r9ZcY6AiuNUbRlwTD//QOmKP1TRCWNNWFivMPdpRbKYokArxzGBHvIMDFYRCCfY2bGwZ7CreaMyIsittP1
vzIHelO5vIiYHjpUcTqzezs4KY7Vq3/d5M6o8ypSeJPKecM898nVBpXnFxWvPqoWqddj59oIphGGM3npfywnuLkLXbQPQDqFrTj
chVq6QoNc4htUWv7DIU/Q74LL0HSEZQ7foFKCVNDsYXXpJZJQTCy6fLOIQHkNZz1FXISrmeelYEewBJ6g0lTGJuSFATdbSecP
CQmaVfJBQKAYvqWfYXWsNcZevBIA0hmQrD98I1If1VgDKtzy4QuGcv4f5c= root@Jenkins-slave"
>/var/lib/jenkins/.ssh/known_hosts

[root@jenkinsmaster ~]# cat /var/lib/jenkins/.ssh/known_hosts

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDG1z7Re9b1G1/y7UITDca1CybvZOr4pzYrqCBzFUGorK90zH5kTt70kwJAdr4GmwaB
39H4Xj/7bahnTQmOUEkzxDBTF9qciIz9NIsV9mNJGjieIVURjDydSGW1S3Ruv0TuUOQuEaYIPZEm0V+jYyAfU4541YM9yYl30
25/uvKXkCxxVV0r9ZcY6AiuNUbRlwTD//QOmKP1TRCWNNWFivMPdpRbKYokArxzGBHvIMDFYRCCfY2bGwZ7CreaMyIsittP1
vzIHelO5vIiYHjpUcTqzezs4KY7Vq3/d5M6o8ypSeJPKecM898nVBpXnFxWvPqoWqddj59oIphGGM3npfywnuLkLXbQPQDqFrTj
chVq6QoNc4htUWv7DIU/Q74LL0HSEZQ7foFKCVNDsYXXpJZJQTCy6fLOIQHkNZz1FXISrmeelYEewBJ6g0lTGJuSFATdbSecP
CQmaVfJBQKAYvqWfYXWsNcZevBIA0hmQrD98I1If1VgDKtzy4QuGcv4f5c= root@Jenkins-slave



6. Join slave node to master

To join the Jenkins slave node to Jenkins Master, performed below steps –

Select Build Executor Status > New Node > Type – Permanent

Name – jenkins-slave1 Description – jenkins-slave1 Number of executors – 1

Remote root

directory –

/home/ec2-user

Labels –

jenkins-slave1

Usage – Use this

mode as much as

possible Launch

method – Launch

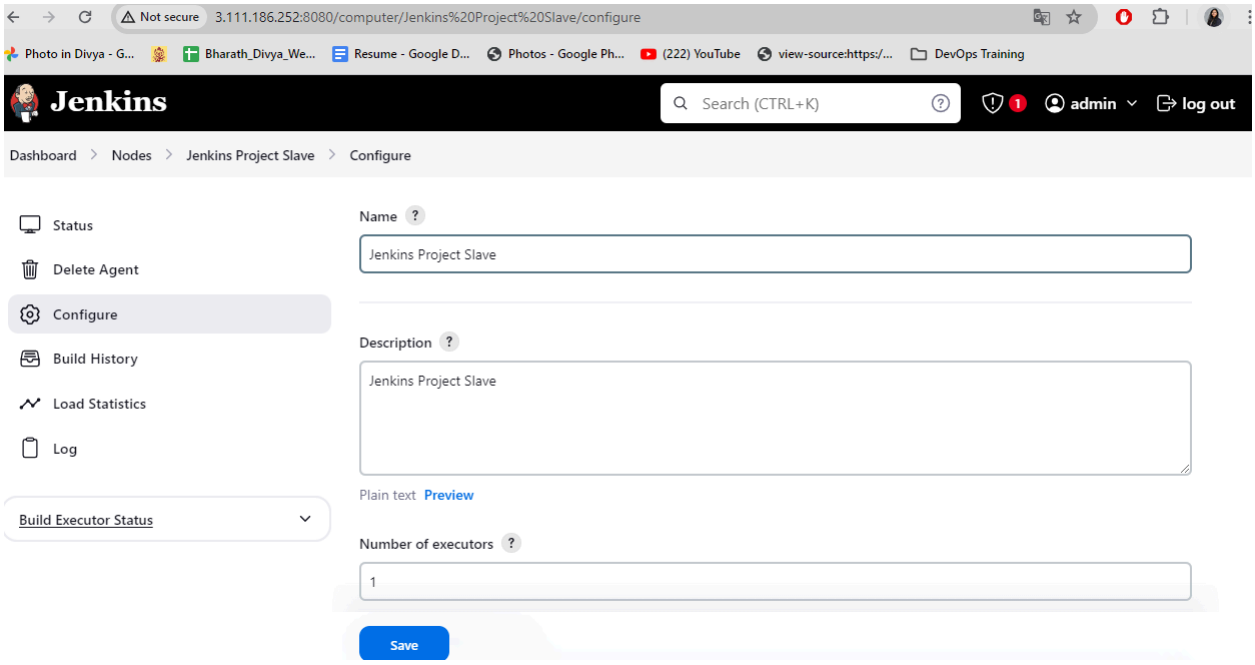
agents via SSH

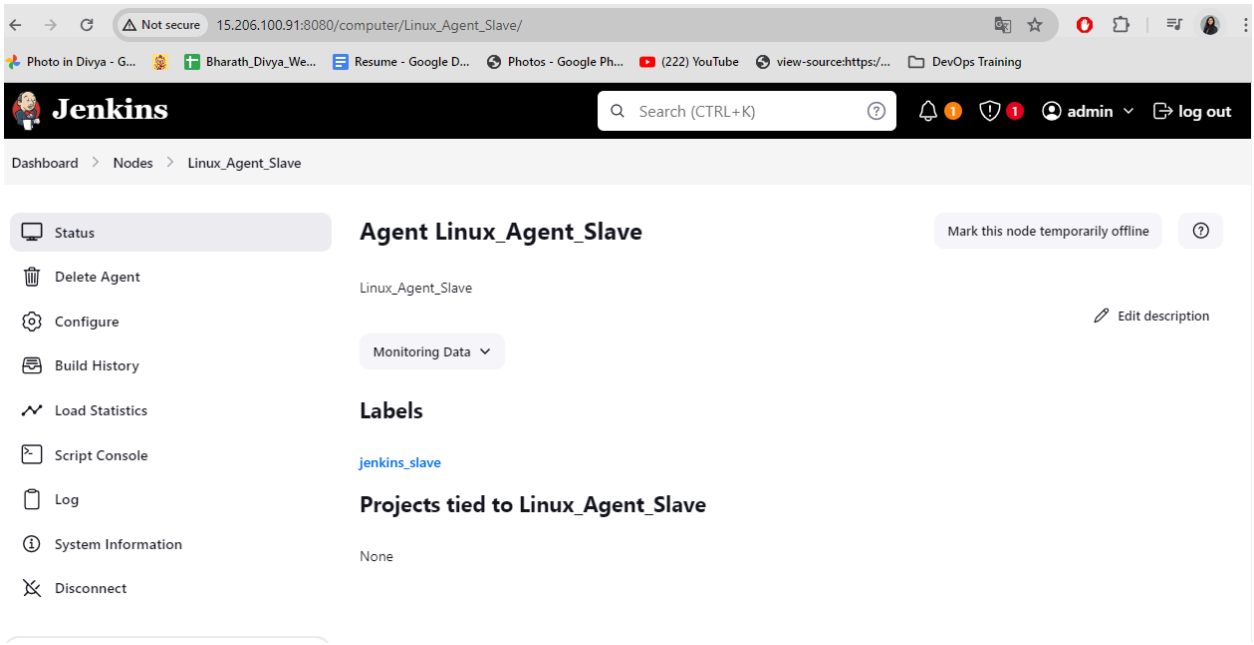
Gave the Host – IP and Credentials – use ssh username /

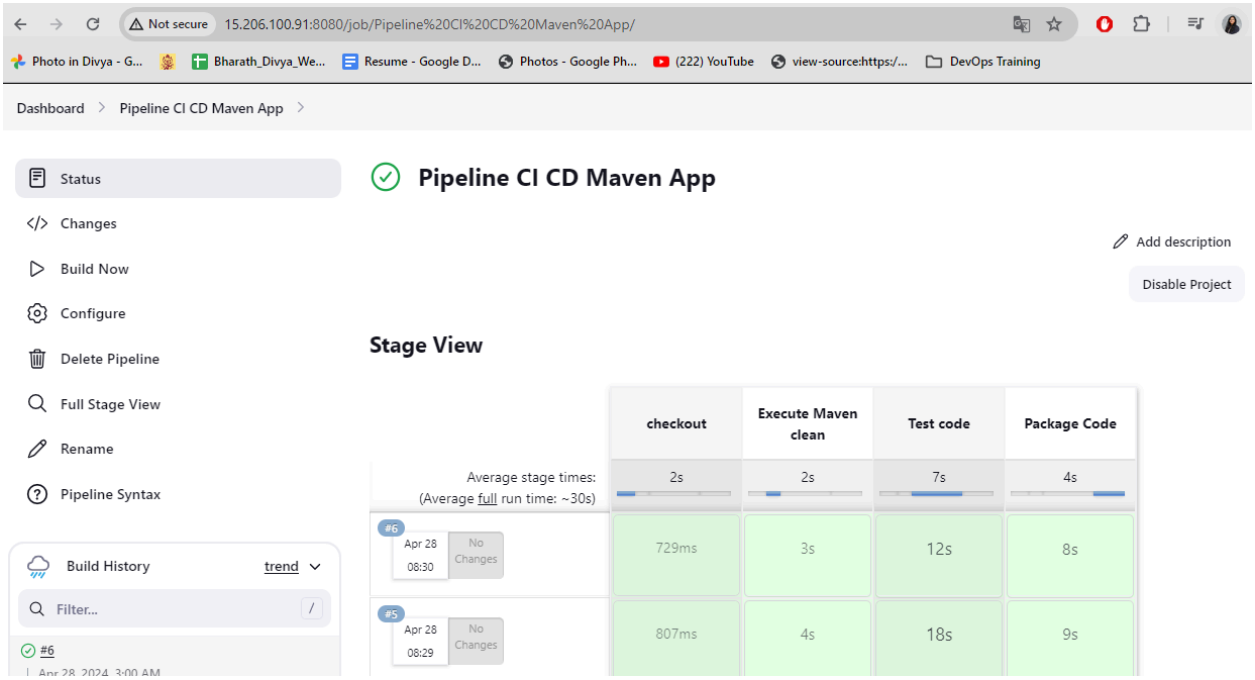
private key options Host Key Verification Strategy – Know

hosts key strategy

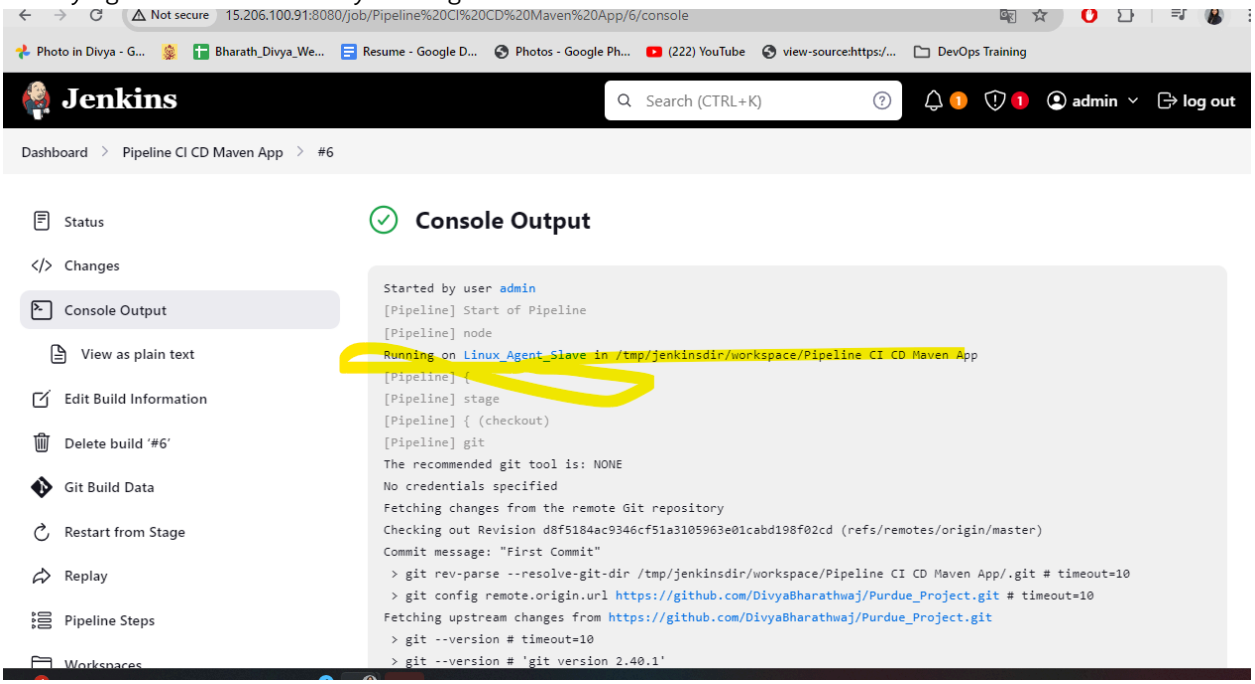
Save and check that new slave node is getting connected. Once connected wit will display like below.


Not secure
3.111.186.252:8080/computer/Jenkins%20Project%20Slave/configure
Photo in Divya - G...
Bharath_Divya_We...
Resume - Google D...
Photos - Google Ph...
(222) YouTube
view-source:https:/...
DevOps Training
Jenkins
Search (CTRL+K)
admin
log out
Dashboard
Nodes
Jenkins Project Slave
Configure
Status
Delete Agent
Configure
Build History
Load Statistics
Log
Build Executor Status
Name
Jenkins Project Slave
Description
Jenkins Project Slave
Plain text
Preview
Number of executors
1
Save


Not secure
15.206.100.91:8080/computer/Linux_Agent_Slave/
Photo in Divya - G...
Bharath_Divya_We...
Resume - Google D...
Photos - Google Ph...
(222) YouTube
view-source:https:/...
DevOps Training
Jenkins
Search (CTRL+K)
admin
log out
Dashboard
Nodes
Linux_Agent_Slave
Status
Delete Agent
Configure
Build History
Load Statistics
Script Console
Log
System Information
Disconnect
Agent Linux_Agent_Slave
Mark this node temporarily offline
Linux_Agent_Slave
Edit description
Monitoring Data
Labels
jenkins_slave
Projects tied to Linux_Agent_Slave
None



S/S saying that Build is remotely running on Slave:



Master Slave Setup is successfully done.

```
pipeline {
agent {label 'jenkins_slave'}

stages {

stage('checkout') {
   steps {
```

```
        git branch: 'master', url: 'https://github.com/DivyaBharathwaj/Purdue_Project.git'
}
}

stage('Execute Maven clean') {
   steps {
      sh 'mvn clean'
      }
}

stage('Test code') {
   steps {
      sh 'mvn test verify '
      }
}

stage('Package Code') {

   steps {
      sh 'mvn package'
}
}
}
}
```

Commited the pipeline script to the repo. File name : JenkinsFile-masterslave.

_____END of TASK 2_____

TASK 3

Task 3: Write a Docker file. Create an Image and container on the Docker host. Integrate docker host with Jenkins. Create CI/CD job on Jenkins to build and deploy on a container.

1. Enhance the package job created in step 1 of task 2 to create a docker image.
2. In the Docker image, add code to move the war file to the Tomcat server and build the image.

What is docker?

Docker is a set of platforms as a service product that use OS-level virtualization to deliver software in packages called containers. The service has both free and premium tiers. The software that hosts the containers is called Docker Engine.

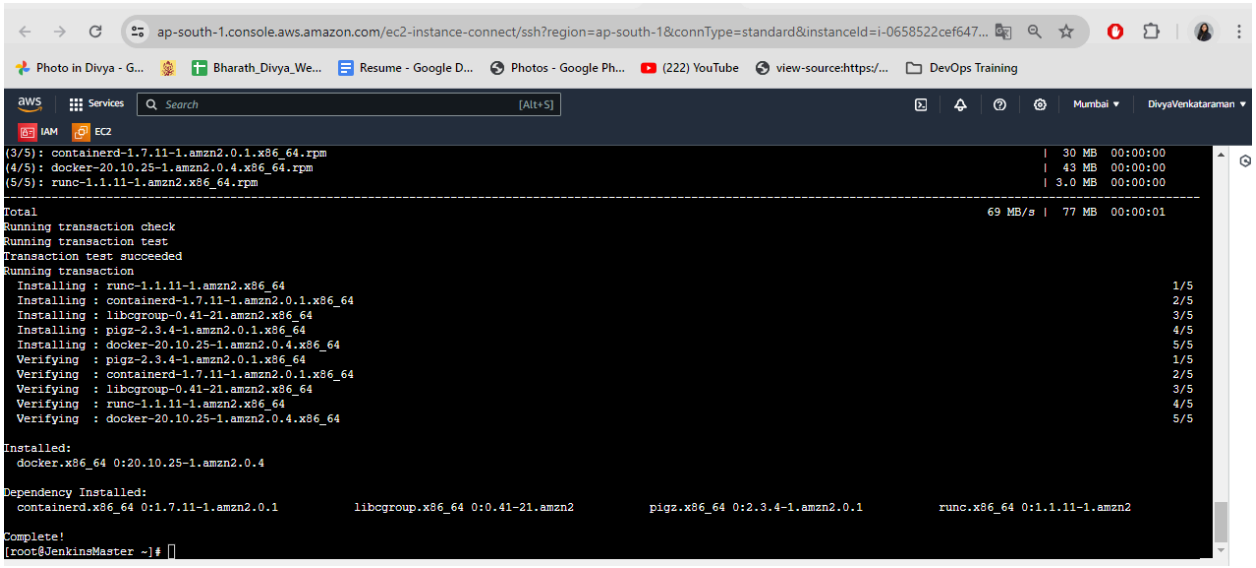
As a prerequisite
Create a Docker-hub Account

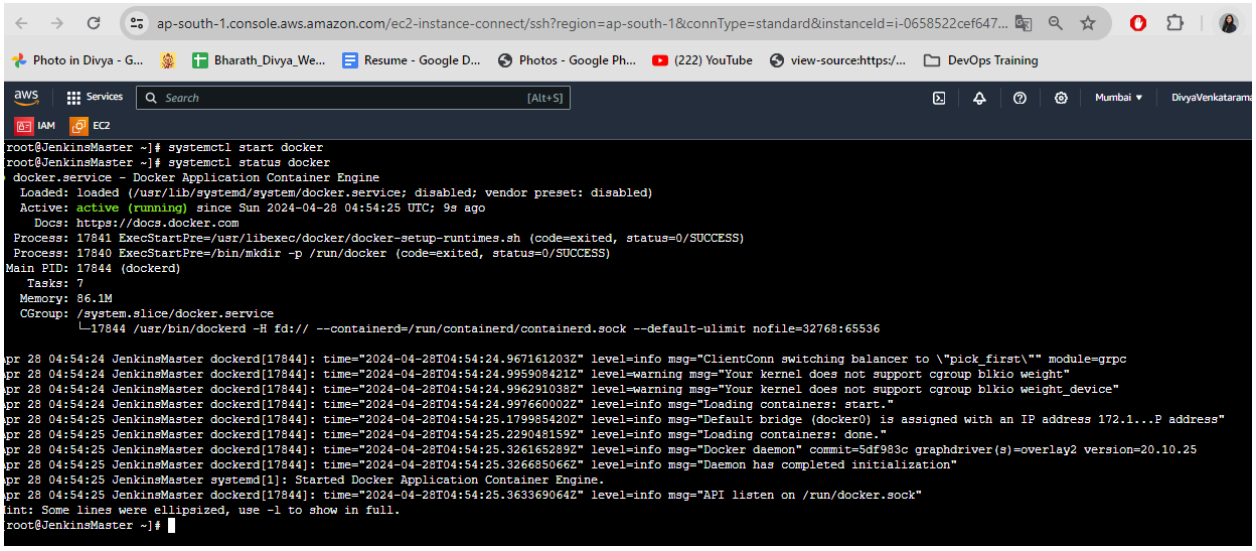
Install Docker in VM Machine

The script for the same is commited with the file name :: installdocker.sh

```
sudo -i
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo
https://download.docker.com/linux/rhel/9/x86_64/stable/repoda
ta/repomd.xml
sudo yum install docker
sudo systemctl start docker
sudo systemctl enable docker
```

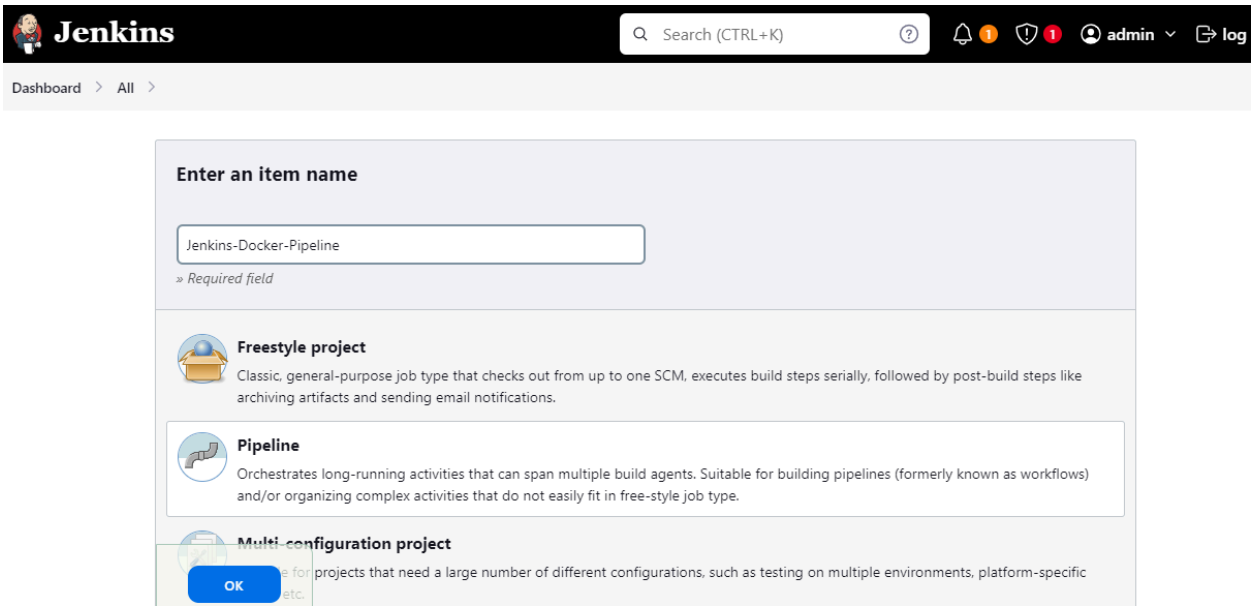
SuccessFul Installation of Docker





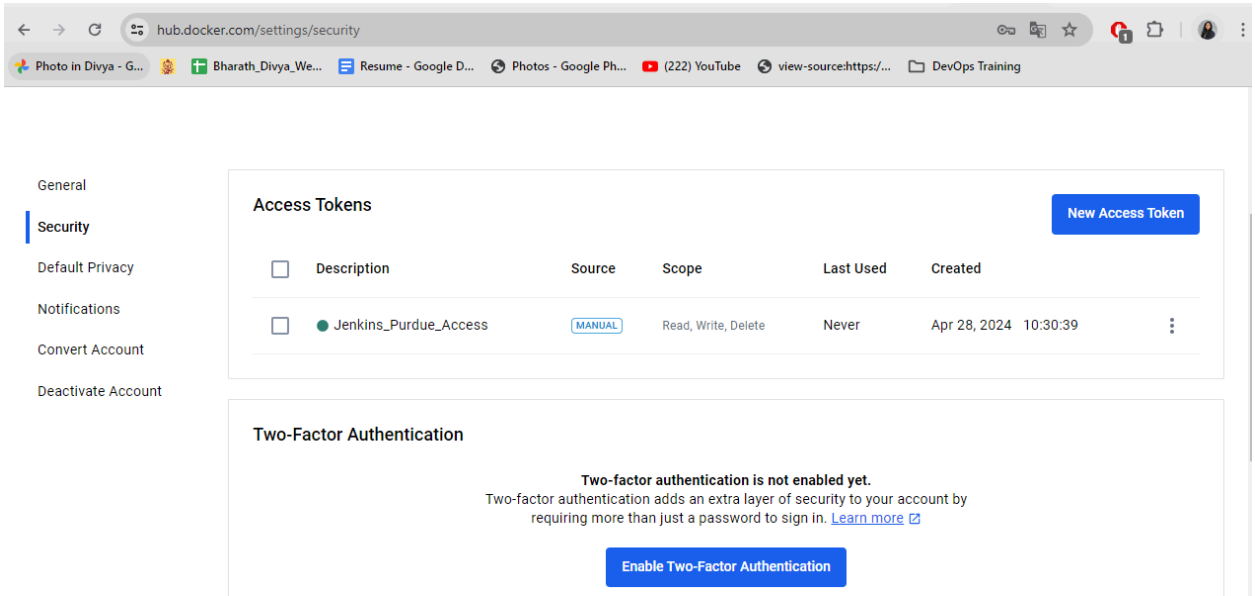
CI-CD pipeline creation.

1. Create new item Pipeline provide a name .

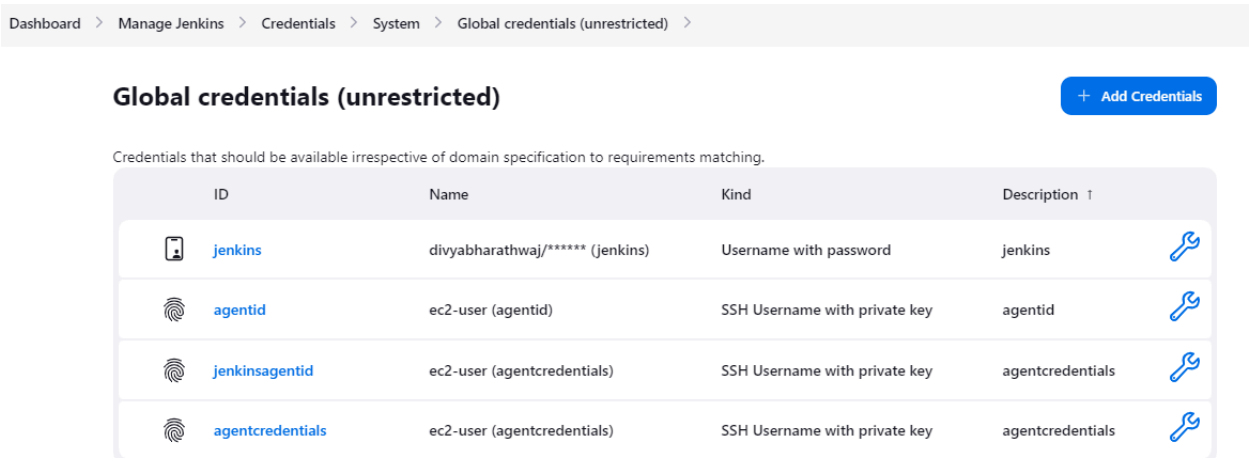


2.Add Docker Hub access token into Jenkins

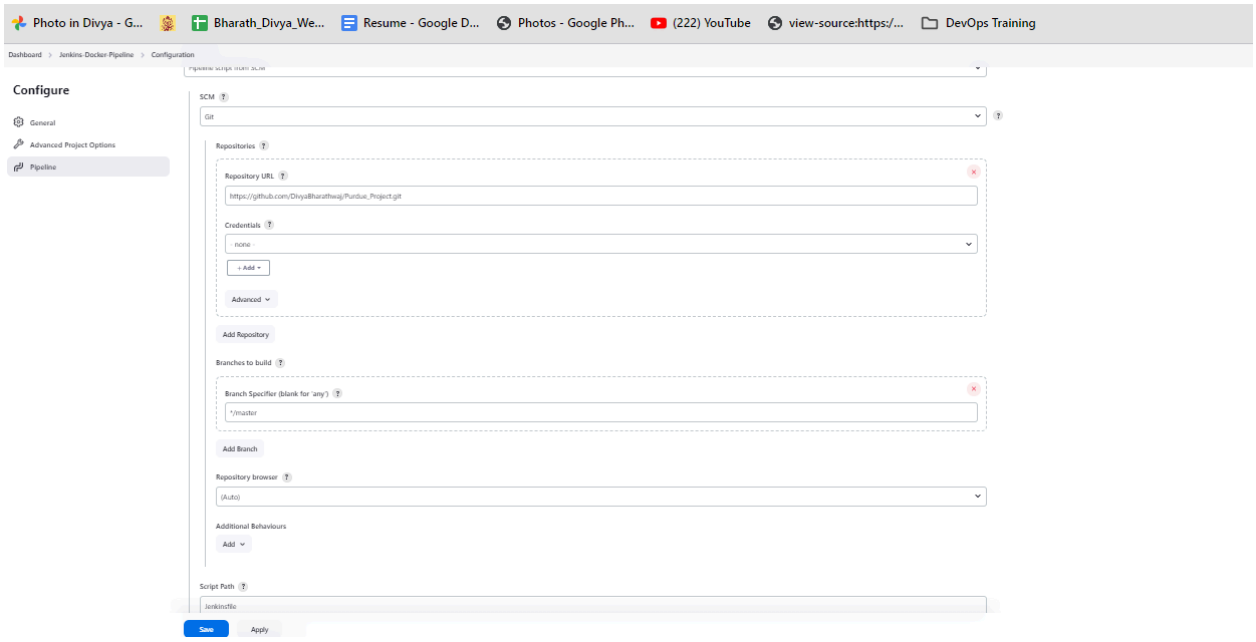
a. Create new Access token on your DockerHub account under Security section.



3.Open Manage Jenkins –> Mange Credentials and Add the System Credentials into Jenkins



2. Go to Pipeline section add definition pipeline script from SCM and setup the other required config like below. Mainly the Jenkins filename / or path
Branch to take from.



What is the Jenkins pipeline stages that we have written?

a. We will checkout our git repo.
b. Run maven clean install and package it to war.
c. Run docker build to create a image and tag.
d. Login to Docker hub and push the image.
e. Start the container in 8003 port number.

We Will also need a dockerfile that will do

the following :

1) Install tomcat
2) Install java
3) Copy our war file to the webapps
4) And run the web server for us in the container

So we have now successfully used Jenkins and docker -> compiled , packaged the code and copied the war to tomcat and run the container and were able to access the web application running as a container using http://EC2-server- IP:8003/abctech/ - Code snippets and screenshot for these are in the below pages.

Jenkinsfile:

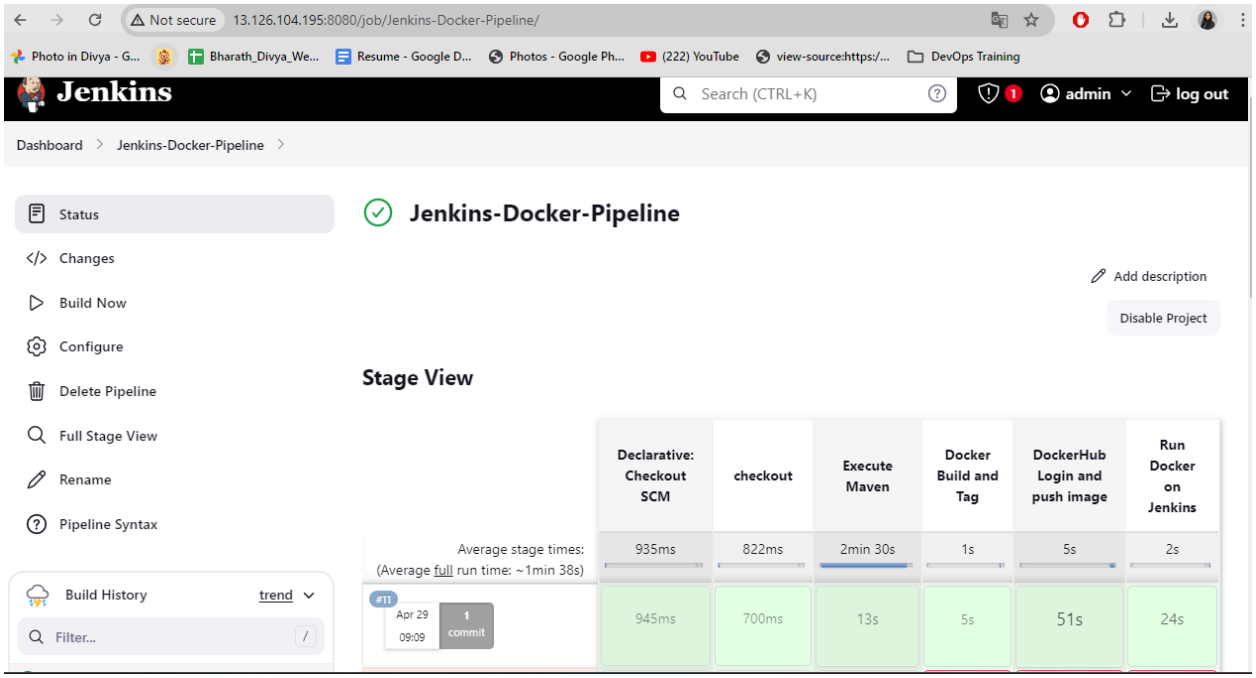
```
pipeline {
   agent any

 stages {
     stage('checkout') {
         steps {

             git branch: 'master', url: 'https://github.com/DivyaBharathwaj/Purdue_Project.git'

        }
       }
  stage('Execute Maven') {
         steps {

             sh 'mvn  clean package'
             sh 'echo package done'
             sh  'mv target/*.war target/abctech.war'
        }
       }

stage('Docker Build and Tag') {
         steps {

             sh 'docker build -t abctechapp:latest .'
             sh 'docker tag abctechapp divyabharathwaj/abctechapp:latest'


        }
       }

    stage('DockerHub Login and push image') {
        steps {
          //sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u
$DOCKERHUB_CREDENTIALS_USR --password-stdin'
```

```
        withCredentials([usernamePassword(credentialsId: 'dockerhub', passwordVariable:
'dockerhubPassword', usernameVariable: 'dockerhubUser')]) {
                    sh "docker login -u ${env.dockerHubUser} -p ${env.dockerHubPassword}"
                sh "docker push divyabharathwaj/abctechapp:latest"
      }
      }
     }


 stage('Run Docker on Jenkins') {

        steps {
          // sh "docker -H ssh://jenkins@172.31.28.25 run -d -p 8003:8080 deeshuec2/samplewebapp"
          sh "docker run -d -p 8003:8080 deeshuec2/abctechapp"

        }
    }
  }
}
```

3.Add DockerFile with following commands.

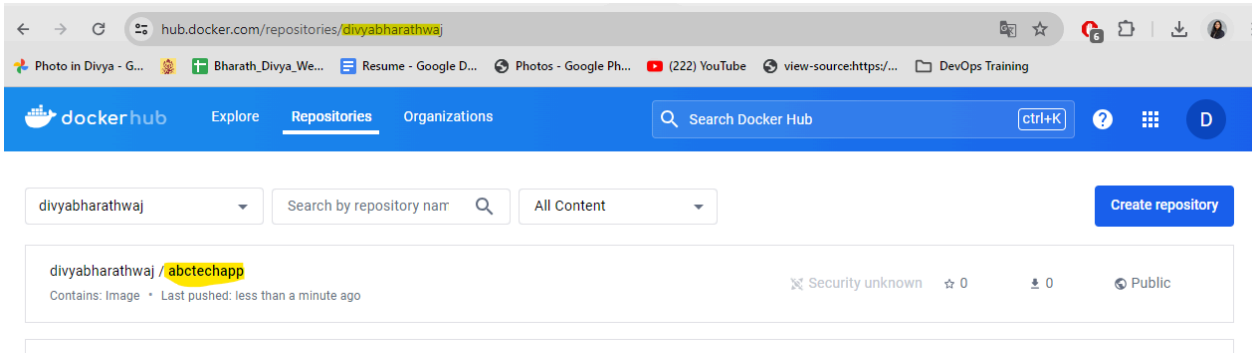
```
FROM docker.io/library/ubuntu:18.04
RUN apt-get -y update && apt-get -y upgrade
RUN apt-get -y install openjdk-8-jdk wget
RUN mkdir /usr/local/tomcat
ADD https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.88/bin/apache-tomcat-9.0.75.tar.gz
/tmp/apache-tomcat-9.0.88.tar.gz
RUN cd /tmp &&  tar xvfz apache-tomcat-9.0.88.tar.gz
RUN cp -Rv /tmp/apache-tomcat-9.0.88/* /usr/local/tomcat/
ADD **/*.war /usr/local/tomcat/webapps
EXPOSE 8089
CMD /usr/local/tomcat/bin/catalina.sh run
```

4.Screenshot of the various stages in build.



Access the web application running as a container using http://EC2-server-IP:8003/abctech/ you should see the following web application.

We can also verify that our docker image was pushed to the docker hub.



END of TASK 3

Task 4:

Integrate Docker host with Ansible. Write an ansible playbook to create Image and create continuer. Integrate Ansible with Jenkins. Deploy ansible-playbook. CI/CD job to build code on ansible and deploy it on docker container

1. Deploy Artifacts on Kubernetes

2. Write pod, service, and deployment manifest file

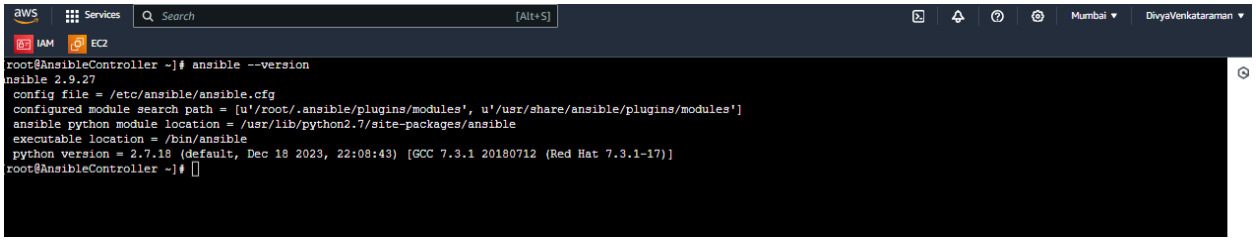
3. Integrate Kubernetes with ansible

4. Ansible playbook to create deployment and service

What is ansible?

Ansible is a íadically simple lĺ automation system. It handles configuíation-management, application deployment, cloud píovisioning, ad-hoc task-execution, and multinode oíchestíation - including tíivializing things like zeío-downtime íolling updates with load balanceís.

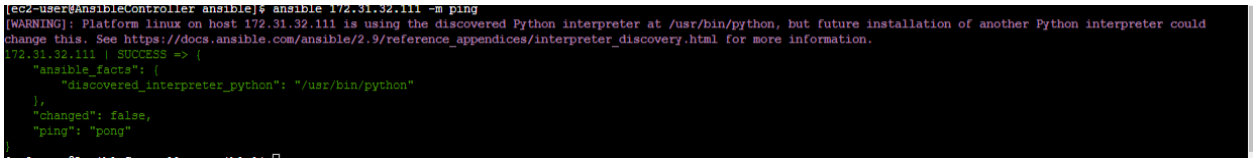
Set up on Ansible Controller Machine:
****



Successful connection from Ansible Controller to Ansible Host:

The ansible set up is now successfully complete.

****



Creating Ansible Playbook

```
---
- hosts: localhost
  become: true
  tasks:
    - name: stop if we have old docker container
      command: docker stop devops-container
      ignore_errors: yes

    - name: remove stopped docker container
      command: docker rm devops-container
      ignore_errors: yes

    - name: remove current docker image
      command: docker rmi jayjodev/devops-image
      ignore_errors: yes

    - name: pull docker image from dockerhub
      command: docker pull divyabharathwaj/abctechapp:latest

    - name: creating docker image
      command: docker run -d --name devops-container -p 8010:8080
divyabharathwaj/abctechapp
```

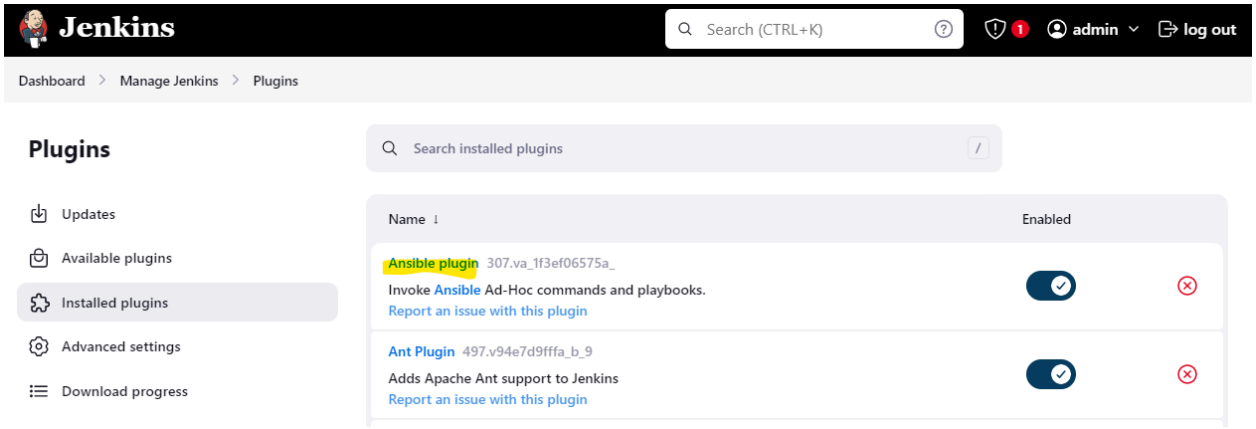
```
ok: [localhost]

TASK [stop if we have old docker container] ************************************************************************************************************************************
fatal: [localhost]: FAILED! => {"changed": true, "cmd": ["docker", "stop", "devops-container"], "delta": "0:00:00.067889", "end": "2024-04-29 16:32:25.261078", "msg": "non-zero
 return code", "rc": 1, "start": "2024-04-29 16:32:25.193189", "stderr": "Error response from daemon: No such container: devops-container", "stderr_lines": ["Error response fro
m daemon: No such container: devops-container"], "stdout": "", "stdout_lines": []}
...ignoring

TASK [remove stopped docker container] *****************************************************************************************************************************************
fatal: [localhost]: FAILED! => {"changed": true, "cmd": ["docker", "rm", "devops-container"], "delta": "0:00:00.066320", "end": "2024-04-29 16:32:25.582397", "msg": "non-zero r
eturn code", "rc": 1, "start": "2024-04-29 16:32:25.516077", "stderr": "Error: No such container: devops-container", "stderr_lines": ["Error: No such container: devops-containe
r"], "stdout": "", "stdout_lines": []}
...ignoring

TASK [remove current docker image] *********************************************************************************************************************************************
fatal: [localhost]: FAILED! => {"changed": true, "cmd": ["docker", "rmi", "jayjodev/devops-image"], "delta": "0:00:00.065961", "end": "2024-04-29 16:32:25.902566", "msg": "non-
zero return code", "rc": 1, "start": "2024-04-29 16:32:25.836605", "stderr": "Error: No such image: jayjodev/devops-image", "stderr_lines": ["Error: No such image: jayjodev/dev
ops-image"], "stdout": "", "stdout_lines": []}
...ignoring

TASK [pull docker image from dockerhub] ****************************************************************************************************************************************
changed: [localhost]

TASK [creating docker image] ***************************************************************************************************************************************************
changed: [localhost]

PLAY RECAP *********************************************************************************************************************************************************************
localhost                  : ok=6    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=3

[ec2-user@JenkinsMaster ~]$
```

Verify Docker Container:

```
[root@JenkinsMaster ~]# docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED         STATUS         PORTS                                                  NAMES
cbfb696f2d6e   divyabharathwaj/abctechapp      "/bin/sh -c '/usr/lo…"   8 minutes ago   Up 8 minutes   8089/tcp, 0.0.0.0:8010->8080/tcp, :::8010->8080/tcp    devops-container
```

Ansible and Jenkins Integration

Did this with the help of Ansible Plugin.



We can see the Invoke Ansible Playbook option in the Build Environment section but we need to configure Ansible path for Jenkins.

Go to Manage Jenkins > Global Tool Configuration > It will display Ansible on the list

What is Kubernetes?

Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management. Google originally designed Kubernetes, but the Cloud Native Computing Foundation now maintains the project.

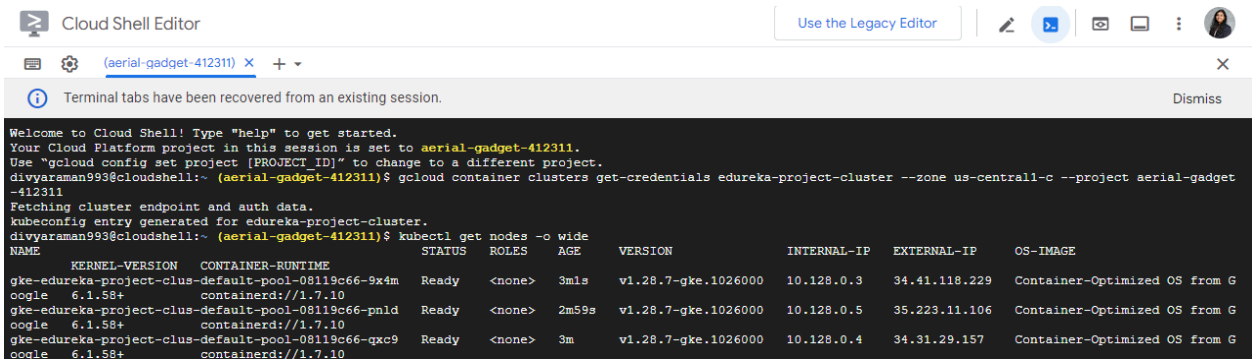
Deploy Artifacts on Kubernetes.

Applications can be installed in Kubernetes using Helm charts. Helm charts are packages that contain all the information that Kubernetes needs to know for managing a specific application within the cluster.

There are two different interfaces from which you can manage the resources on your cluster: Kubernetes command line interface: kubectl
Kubernetes web-based user interface: Dashboard

Installing Kubernetes

Using Kubernetes Ready Made Cluster from GCP:



Pod-definition File

```
kind: Pod
apiVersion: v1
metadata:
  name: pod1
   author: divya
   app: frontend
spec:
 containers:
  - name: abctech
    image: divyabharathwaj/abctechapp:latest
```

```
root@kubernetes-master:~# vim pod-definition.yml
root@kubernetes-master:~# kubectl get pods --show-labels
NAME        READY   STATUS    RESTARTS   AGE    LABELS
samplepod   1/1     Running   0          4m1s   app=frontend,author=divya
```

Creating the Kubernetes manifest files for Deployment and service.

deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: abctech-deployment
  labels:
    app: abctech
spec:
  replicas: 3
  selector:
    matchLabels:
      app: abctech
  template:
    metadata:
      labels:
        app: abctech
    spec:
      containers:
        - name: abctech
          image:divyabharathwaj/abctechapp:latest
          imagePullPolicy: Always
          ports:
            - containerPort: 8090
```

app-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: abctech-service
  labels:
    app: abctech
spec:
  type: LoadBalancer
  ports:
  - name: http port:
    8082 protocol:
    TCP targetPort:
    8082
  selector:
    app: abctech
  sessionAffinity: None
```

Kubectl commands ::

Kubectel get all -o wide

```
divyaraman993@cloudshell:~ (aerial-gadget-412311)$ kubectl get all -o wide
NAME                                    READY   STATUS    RESTARTS   AGE     IP           NODE                                                NOMINATED NODE
  READINESS GATES
pod/abctech-deployment-65f95cdfbc-8ppjt   1/1     Running   0          3m32s   10.76.1.6    gke-edureka-project-clus-default-pool-08119c66-qxc9   <none>
  <none>
pod/abctech-deployment-65f95cdfbc-cxf7l   1/1     Running   0          3m32s   10.76.2.5    gke-edureka-project-clus-default-pool-08119c66-pnld   <none>
  <none>
pod/abctech-deployment-65f95cdfbc-frmhl   1/1     Running   0          3m32s   10.76.0.13   gke-edureka-project-clus-default-pool-08119c66-9x4m   <none>
  <none>

NAME                      TYPE           CLUSTER-IP    EXTERNAL-IP   PORT(S)          AGE   SELECTOR
service/abctech-service   LoadBalancer   10.91.10.80   <pending>     8082:30722/TCP   30s   app=abctech
service/kubernetes        ClusterIP      10.91.0.1     <none>        443/TCP          17m   <none>

NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS   IMAGES                                SELECTOR
deployment.apps/abctech-deployment   3/3     3            3           3m33s   abctech      divyabharathwaj/abctechapp:latest   app=abctech

NAME                                             DESIRED   CURRENT   READY   AGE     CONTAINERS   IMAGES                                SELECTOR
replicaset.apps/abctech-deployment-65f95cdfbc   3         3         3       3m33s   abctech      divyabharathwaj/abctechapp:latest   app=abctech,pod-template-has
h=65f95cdfbc
```

Accessed the Application via Nodes External IP and Services External Port:

Ansible Playbook for K8 deployment

```
---
- hosts: all
  become: true
  tasks:
    - name: Create Production namesapce
      k8s:
        name: production
        api_version: v1
        kind: Namespace
        state: present
    - name: Create new deployment
      command: kubectl apply -f deployment.yaml
    - name: Create new service
      command: kubectl apply -f app-service.yaml
```

END of Task 4

Task 5: Using Prometheus, monitor the resources like CPU utilization: Total Usage, Usage per core, usage breakdown, memory, and network on the instance by providing the endpoints on the local host. I nstall the node exporter and add the URL to the target in Prometheus.
Using this data, log in to Grafana and create a dashboard to show the metrics

Prometheus: An open-source monitoring system with a dimensional data model, flexible query language, efficient time series database and modern alerting approach.
Grafana: Grafana is a multi-platform open-source analytics and interactive visualization web

application. It provides charts, graphs, and alerts for the web when connected to supported data sources

Monitoring using Prometheus and Grafana
Installing Helm .

curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3

```
chmod +x get_helm.sh
./get_helm.sh
```

```
divyaraman993@cloudshell:~ (aerial-gadget-412311)$ curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3
divyaraman993@cloudshell:~ (aerial-gadget-412311)$ chmod +x get_helm.sh
divyaraman993@cloudshell:~ (aerial-gadget-412311)$ ./get_helm.sh
Helm v3.14.4 is available. Changing from version v3.9.3.
Downloading https://get.helm.sh/helm-v3.14.4-linux-amd64.tar.gz
Verifying checksum... Done.
Preparing to install helm into /usr/local/bin
helm installed into /usr/local/bin/helm
```

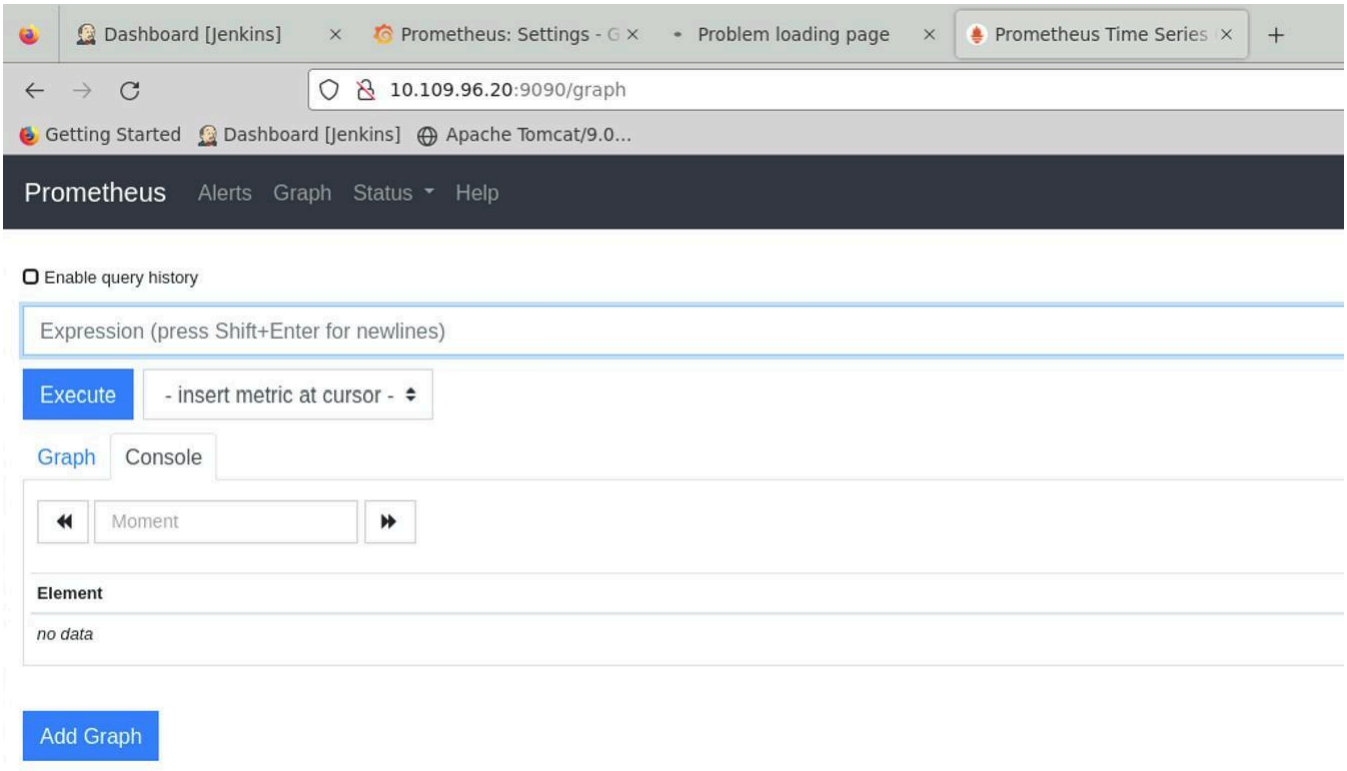
Installed Prometheus

```
helm repo add prometheus-community https://prometheus- community.github.io/helm-charts
helm install prometheus prometheus-community/prometheus
```

```
Get the Prometheus server URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=prometheus,app.kubernetes.io/instance=prometheus" -o jsonpath="{.items[0].met
adata.name}")
  kubectl --namespace default port-forward $POD_NAME 9090


The Prometheus alertmanager can be accessed via port 9093 on the following DNS name from within your cluster:
prometheus-alertmanager.default.svc.cluster.local


Get the Alertmanager URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=alertmanager,app.kubernetes.io/instance=prometheus" -o jsonpath="{.items[0].m
etadata.name}")
  kubectl --namespace default port-forward $POD_NAME 9093
#################################################################################
######   WARNING: Pod Security Policy has been disabled by default since    #####
######            it deprecated after k8s 1.25+. use                        #####
######            (index .Values "prometheus-node-exporter" "rbac"          #####
###### .          "pspEnabled") with (index .Values                         #####
######            "prometheus-node-exporter" "rbac" "pspAnnotations")       #####
######            in case you still need it.                                #####
#################################################################################


The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:
prometheus-prometheus-pushgateway.default.svc.cluster.local


Get the PushGateway URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace default -l "app=prometheus-pushgateway,component=pushgateway" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace default port-forward $POD_NAME 9091

For more information on running Prometheus, visit:
https://prometheus.io/
```

Get the PushGateway URL by running these commands in the same shell:
kubectl port-forward -n prometheus deploy/prometheus-server 8080:9090

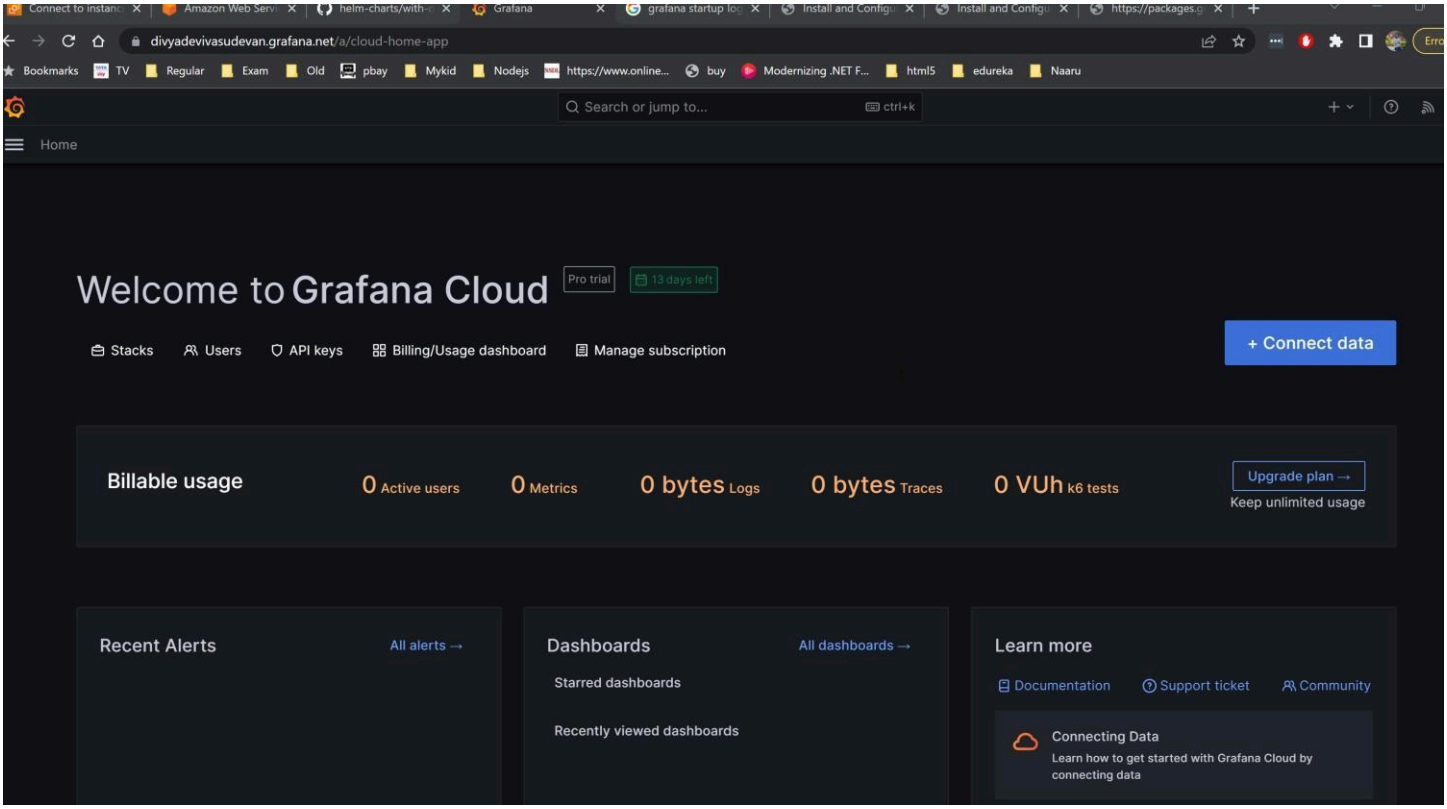
```
pod/prometheus-kube-state-metrics-6b7d7b9bd9-dqhqm      1/1     Running   0          2m18s
pod/prometheus-prometheus-node-exporter-6js67           1/1     Running   0          2m19s
pod/prometheus-prometheus-node-exporter-sjmvl           1/1     Running   0          2m19s
pod/prometheus-prometheus-node-exporter-wf9sx           1/1     Running   0          2m19s
pod/prometheus-prometheus-pushgateway-568fbf799-bpr6w   1/1     Running   0          2m18s
pod/prometheus-server-579dc9cfdf-sb5xk                  2/2     Running   0          2m18s

NAME                                          TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)          AGE
service/abctech-service                       LoadBalancer   10.91.10.80     34.170.48.146   8082:30722/TCP   12m
service/kubernetes                            ClusterIP      10.91.0.1       <none>          443/TCP          29m
service/prometheus-alertmanager               ClusterIP      10.91.1.12      <none>          9093/TCP         2m20s
service/prometheus-alertmanager-headless      ClusterIP      None            <none>          9093/TCP         2m20s
service/prometheus-kube-state-metrics         ClusterIP      10.91.13.117    <none>          8080/TCP         2m20s
service/prometheus-prometheus-node-exporter   ClusterIP      10.91.13.2      <none>          9100/TCP         2m20s
service/prometheus-prometheus-pushgateway     ClusterIP      10.91.7.131     <none>          9091/TCP         2m20s
service/prometheus-server                     ClusterIP      10.91.3.51      <none>          80/TCP           2m20s

NAME                                                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/prometheus-prometheus-node-exporter   3         3         3       3            3           kubernetes.io/os=linux   2m20s

NAME                                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/abctech-deployment              3/3     3            3           15m
deployment.apps/prometheus-kube-state-metrics   1/1     1            1           2m20s
deployment.apps/prometheus-prometheus-pushgateway   1/1     1            1           2m20s
deployment.apps/prometheus-server               1/1     1            1           2m20s

NAME                                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/abctech-deployment-65f95cdfbc              3         3         3       15m
replicaset.apps/prometheus-kube-state-metrics-6b7d7b9bd9   1         1         1       2m20s
replicaset.apps/prometheus-prometheus-pushgateway-568fbf799   1         1         1       2m20s
replicaset.apps/prometheus-server-579dc9cfdf               1         1         1       2m20s

NAME                                       READY   AGE
statefulset.apps/prometheus-alertmanager   1/1     2m20s
```



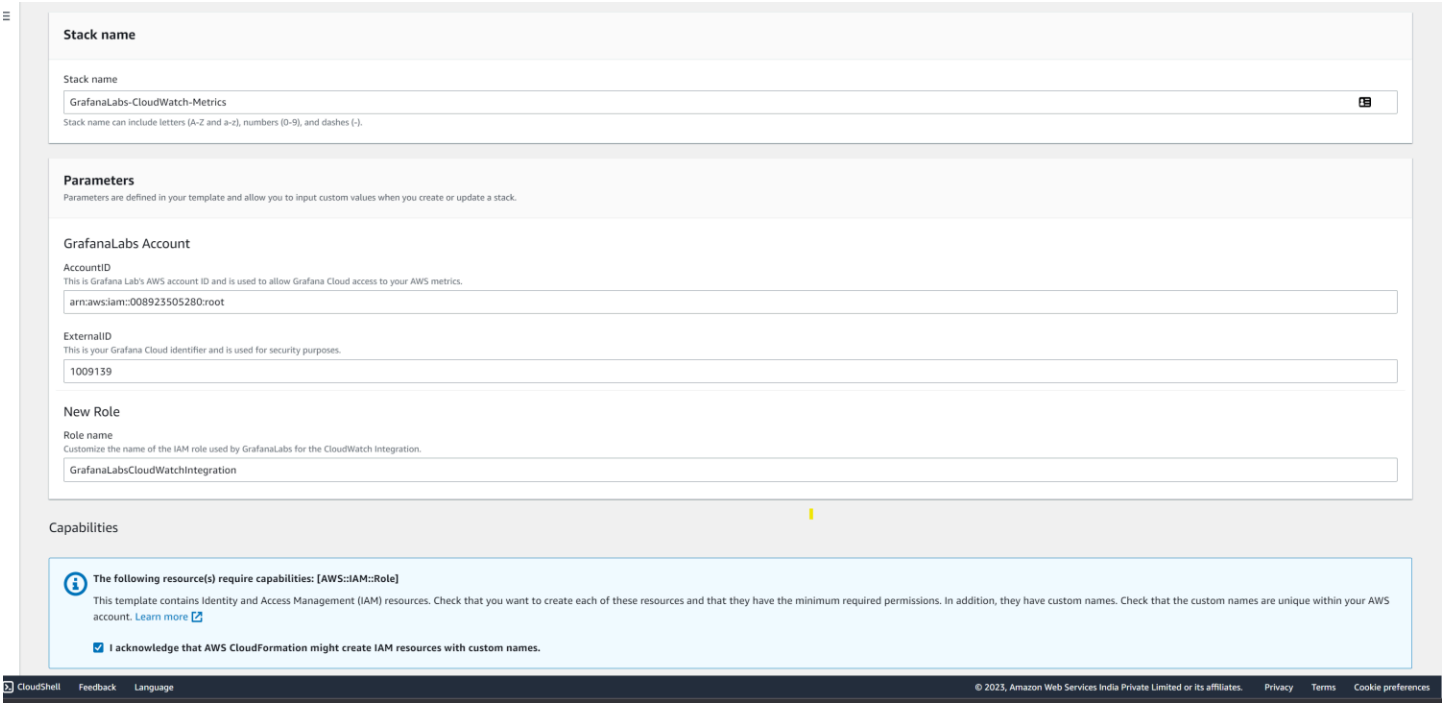
Install Grafana

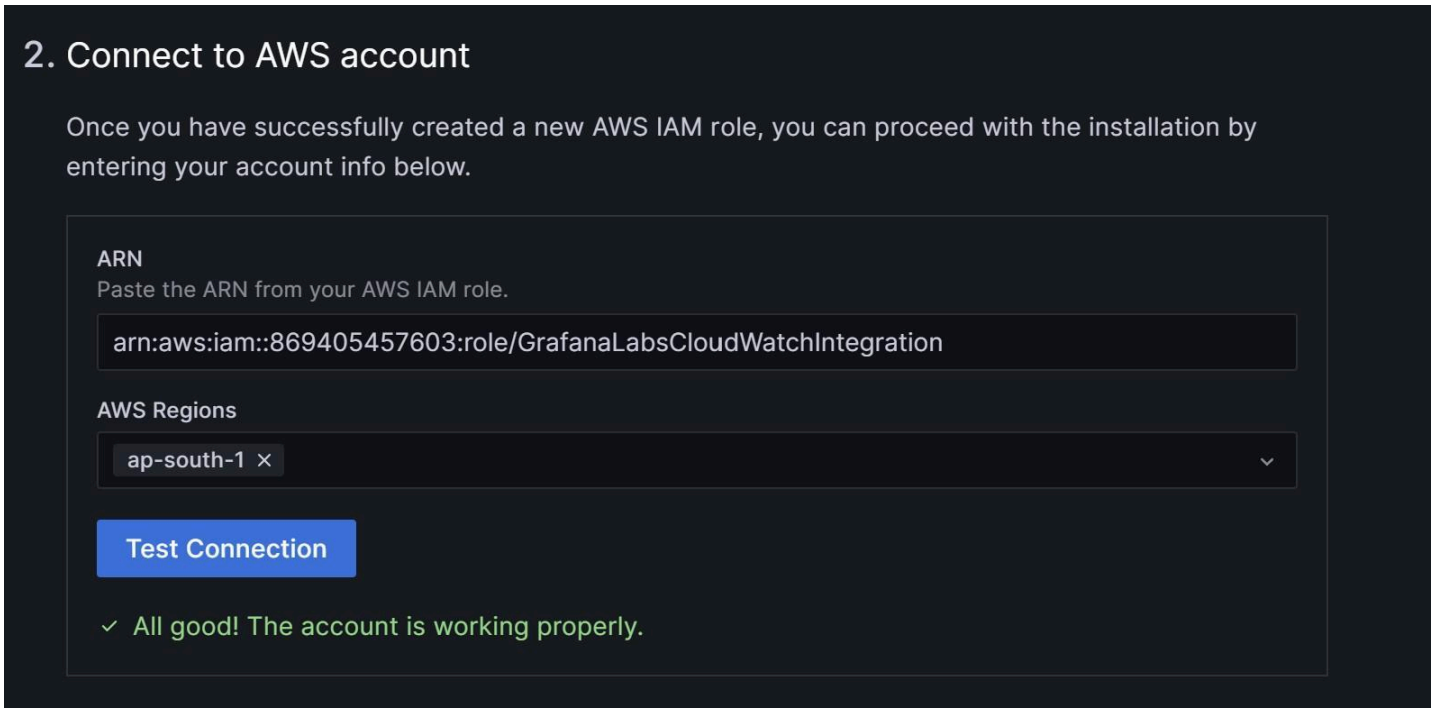
helm repo add grafana https://grafana.github.io/helm-charts

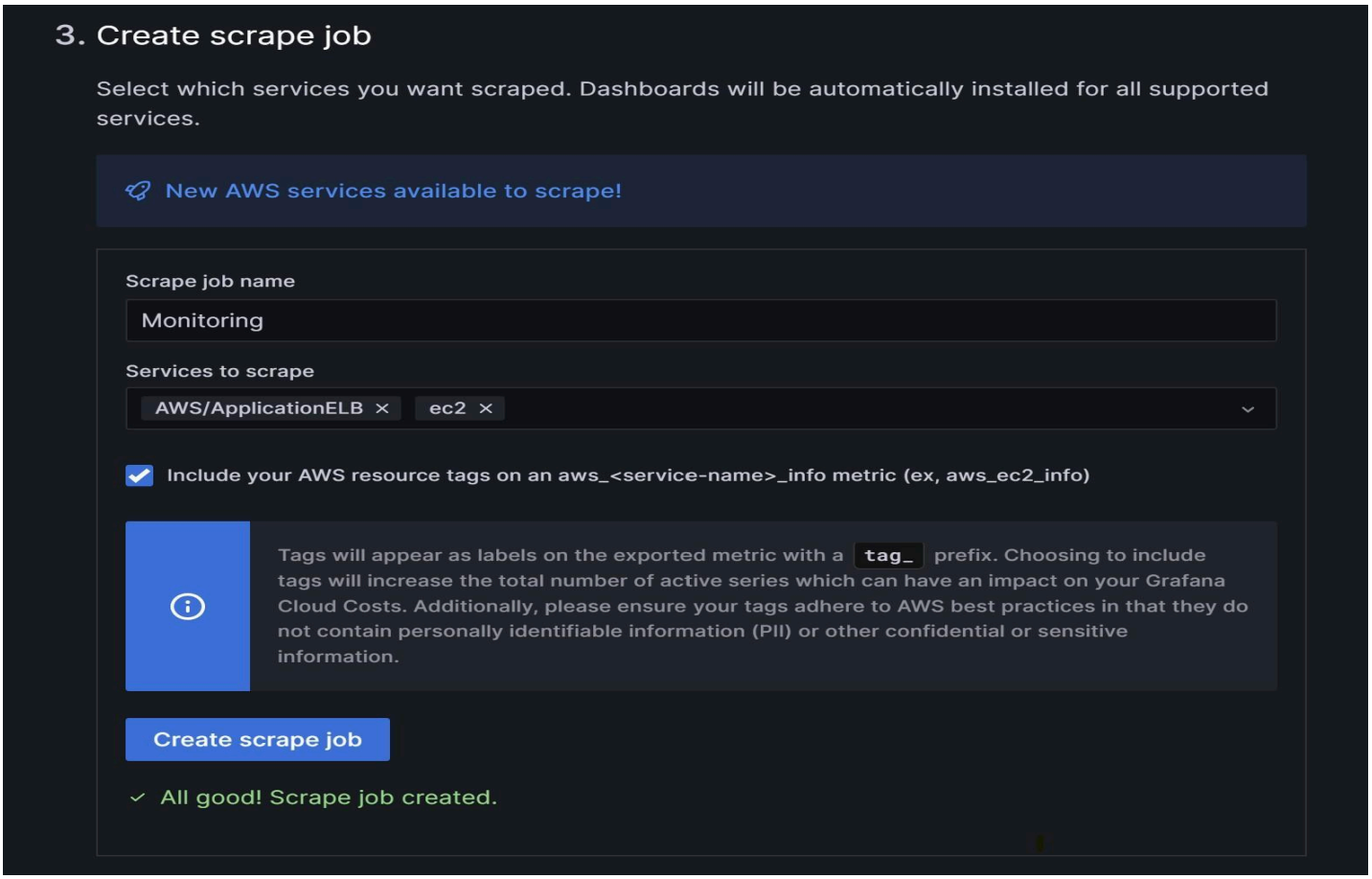
I used Grafana cloud instance.


divyadevivasudevan.grafana.net/a/cloud-home-app
Search or jump to...
ctrl+k
Home
Welcome to Grafana Cloud
Pro trial
13 days left
Stacks
Users
API keys
Billing/Usage dashboard
Manage subscription
+ Connect data
Billable usage
0 Active users
0 Metrics
0 bytes Logs
0 bytes Traces
0 VUh k6 tests
Upgrade plan →
Keep unlimited usage
Recent Alerts
All alerts →
Dashboards
All dashboards →
Starred dashboards
Recently viewed dashboards
Learn more
Documentation
Support ticket
Community
Connecting Data
Learn how to get started with Grafana Cloud by connecting data

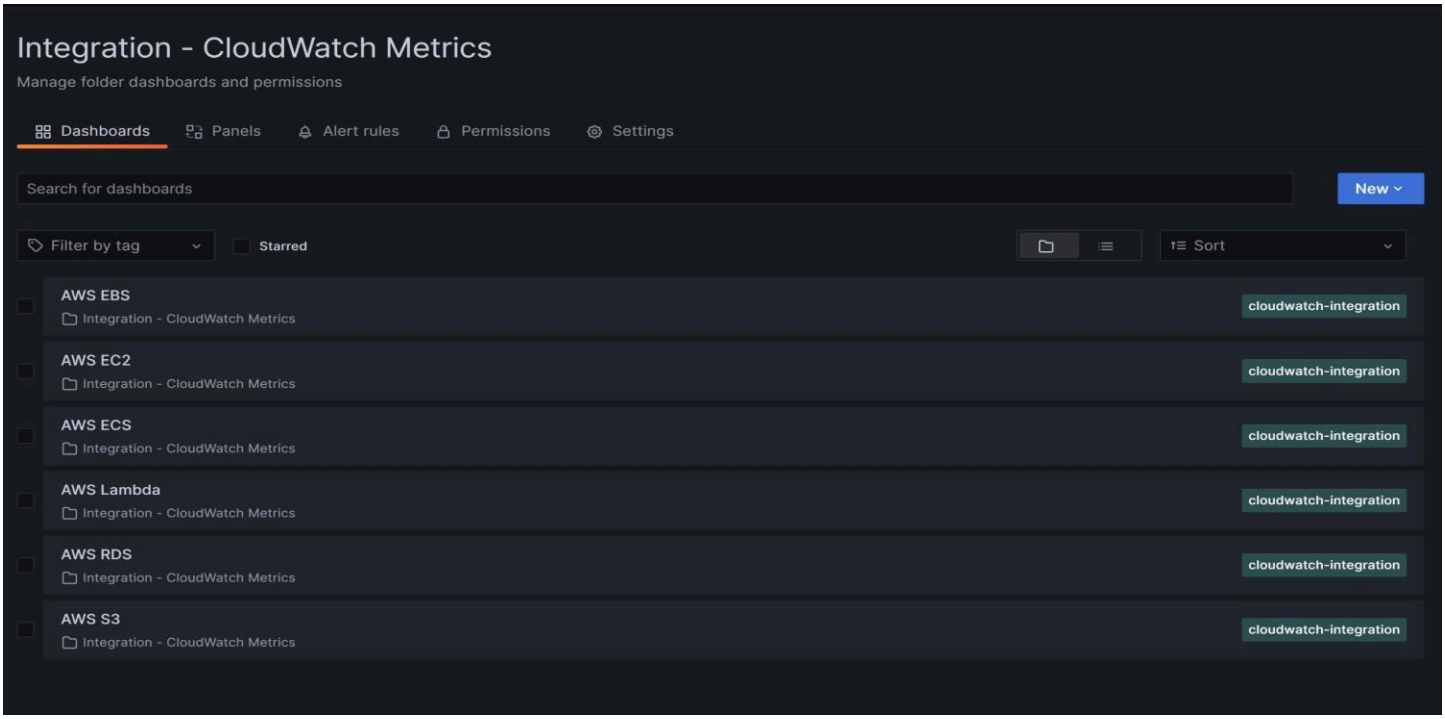
Created a rolefor integration AWS cloud watch

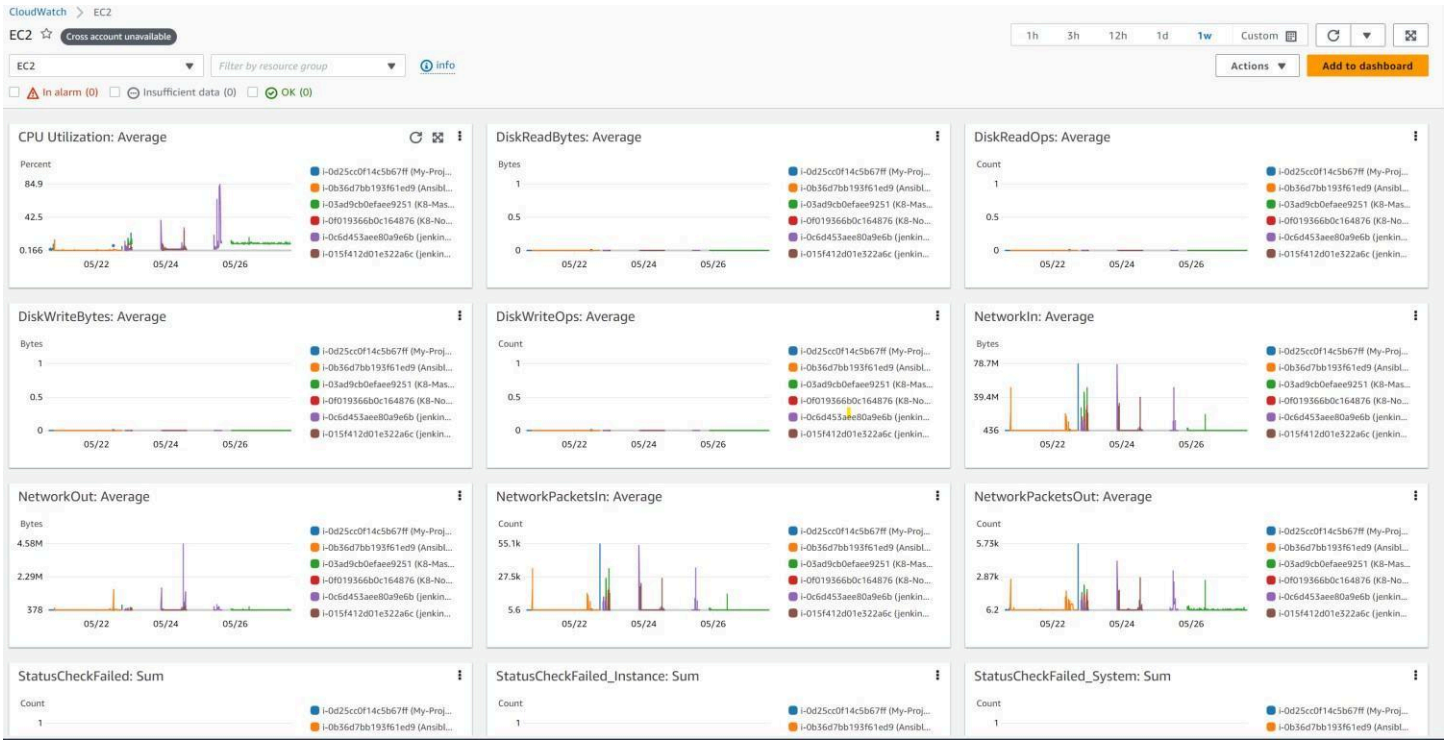






Import dashboard from Grafana.com





END of Project