



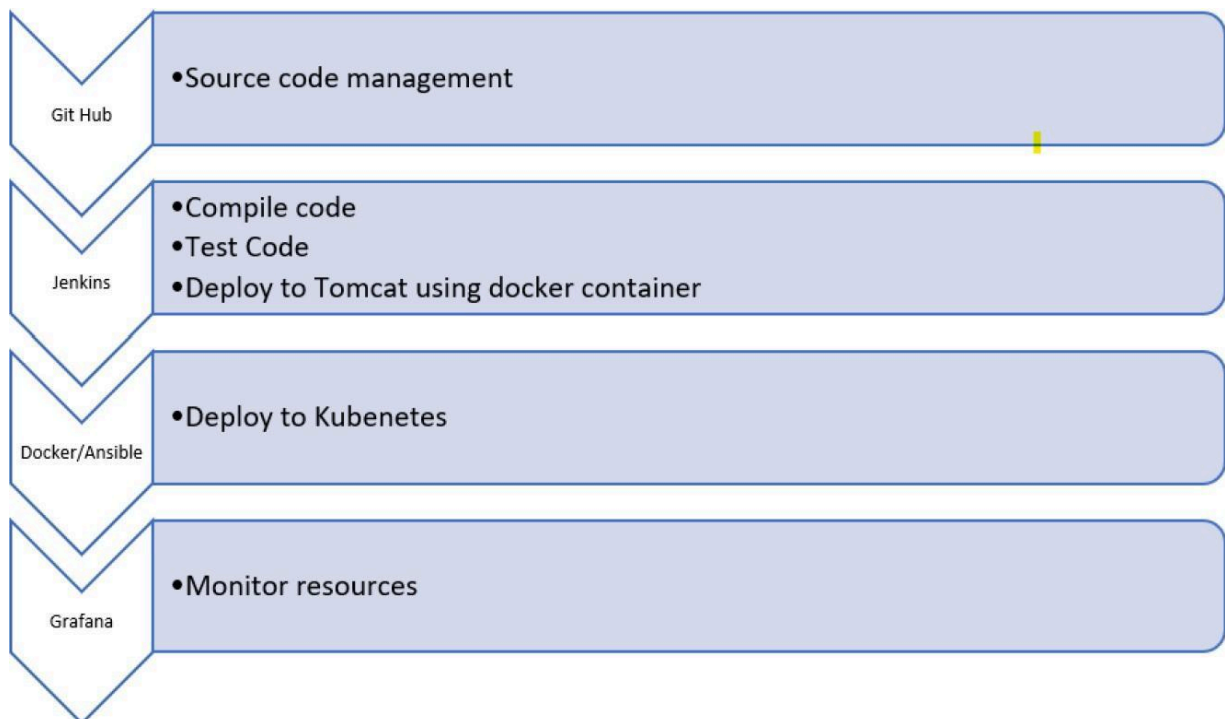
Industry Grade Project

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 | 1. AWS Account |
| 2. Maven | 2. Putty/MobaXTerm |
| 3. Git | 3. Github account |
| 4. Jenkins | 4. Docker-hub account |
| 5. Docker | |
| 6. Ansible | |
| 7. Kubernetes | |
| 8. Grafana | |
| 9. Prometheus | |

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 Technologies (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 Technologies (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 touches 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 Technologies (master)
$ git commit -m "First commit"
On branch master
nothing to commit, working tree clean

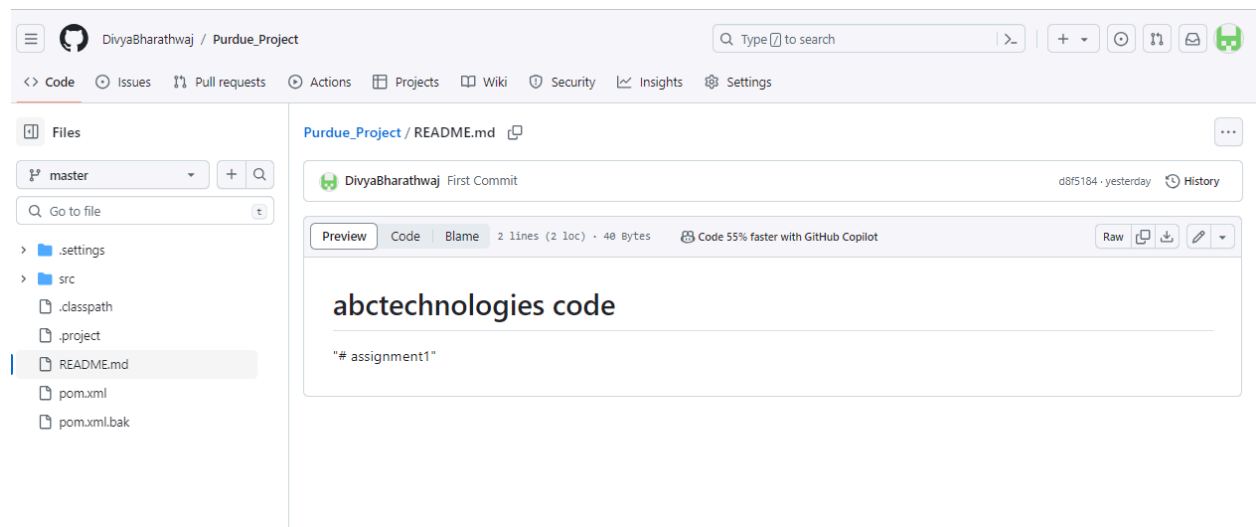
Divya Venkataraman@DESKTOP-ACLEQ51 MINGW64 ~/Desktop/Edureka Project/Industry Grade Project I - Java Project/ABC Technologies (master)
$ git remote add origin "https://github.com/DivyaBharathwaj/Purdue_Project.git"
error: remote origin already exists.

Divya Venkataraman@DESKTOP-ACLEQ51 MINGW64 ~/Desktop/Edureka Project/Industry Grade Project I - Java Project/ABC Technologies (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'.

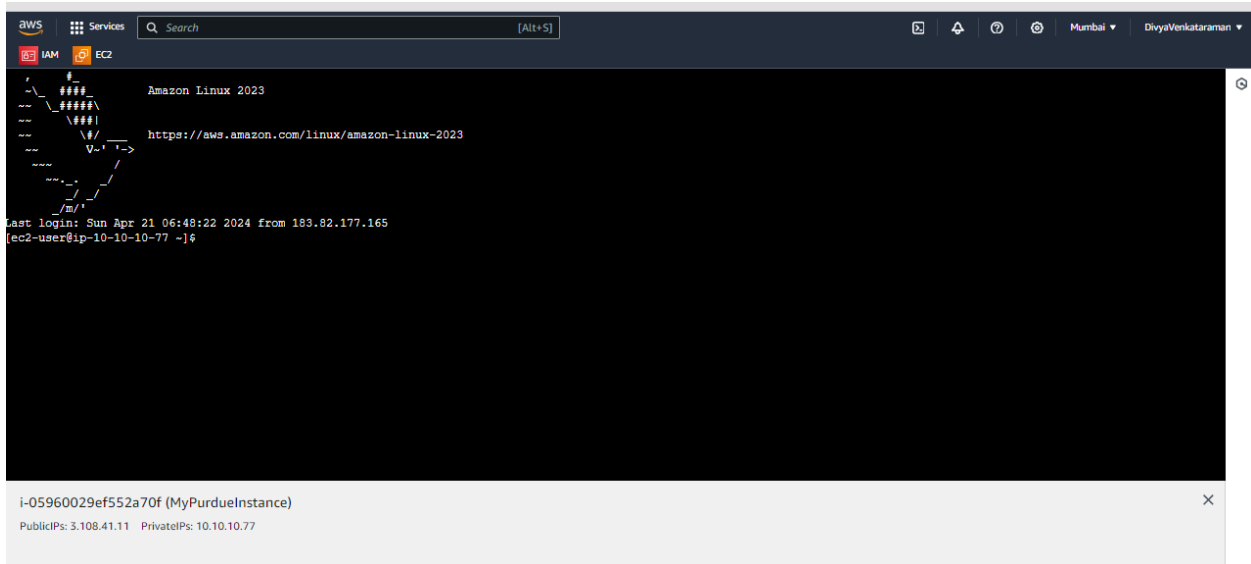
Divya Venkataraman@DESKTOP-ACLEQ51 MINGW64 ~/Desktop/Edureka Project/Industry Grade Project I - Java Project/ABC Technologies (master)

```

Verified the Github whether commit and push was successful.



Launch an EC2 instance:



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`

```
Running transaction test
Transaction test succeeded.
Running transaction
Preparing      :
Installing     : git-core-2.40.1-1.amzn2023.0.1.x86_64 1/1
Installing     : git-core-doc-2.40.1-1.amzn2023.0.1.noarch 1/8
Installing     : perl-lib-0.65-477.amzn2023.0.6.x86_64 2/8
Installing     : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 3/8
Installing     : perl-File-Find-1.37-477.amzn2023.0.6.noarch 4/8
Installing     : perl-Error-1:0.17029-5.amzn2023.0.2.noarch 5/8
Installing     : perl-Git-2.40.1-1.amzn2023.0.1.noarch 6/8
Installing     : git-2.40.1-1.amzn2023.0.1.x86_64 7/8
Installing     : git-2.40.1-1.amzn2023.0.1.x86_64 8/8
Running scriptlet: git-2.40.1-1.amzn2023.0.1.x86_64 8/8
Verifying      : git-2.40.1-1.amzn2023.0.1.x86_64 1/8
Verifying      : git-core-2.40.1-1.amzn2023.0.1.noarch 2/8
Verifying      : git-core-doc-2.40.1-1.amzn2023.0.1.noarch 3/8
Verifying      : perl-Error-1:0.17029-5.amzn2023.0.2.noarch 4/8
Verifying      : perl-File-Find-1.37-477.amzn2023.0.6.noarch 5/8
Verifying      : perl-Git-2.40.1-1.amzn2023.0.1.noarch 6/8
Verifying      : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 7/8
Verifying      : perl-lib-0.65-477.amzn2023.0.6.x86_64 8/8

Installed:
git-2.40.1-1.amzn2023.0.1.x86_64      git-core-2.40.1-1.amzn2023.0.1.x86_64  git-core-doc-2.40.1-1.amzn2023.0.1.noarch  perl-Error-1:0.17029-5.amzn2023.0.2.noarch
perl-File-Find-1.37-477.amzn2023.0.6.noarch  perl-Git-2.40.1-1.amzn2023.0.1.noarch  perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64  perl-lib-0.65-477.amzn2023.0.6.x86_64

Complete!
```

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

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
cairo-1.17.6-2.amzn2023.0.1.x86_64
dejavu-sans-mono-fonts-2.37-16.amzn2023.0.2.noarch
fontconfig-2.13.94-2.amzn2023.0.2.x86_64
freetype-2.13.0-2.amzn2023.0.1.x86_64
google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
graphite2-1.3.14-7.amzn2023.0.2.x86_64
java-17-amazon-corretto-headless-1:17.0.10+8-1.amzn2023.1.x86_64
java-22-amazon-corretto-devel-1:22.0.0+37-1.amzn2023.1.x86_64
javapackages-filestystem-6.0.0-7.amzn2023.0.6.noarch
langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
libSM-1.2.3-8.amzn2023.0.2.x86_64
libX11-common-1.7.2-3.amzn2023.0.4.noarch
libXext-1.3.4-6.amzn2023.0.2.x86_64
libXinerama-1.1.4-8.amzn2023.0.2.x86_64
libXrender-0.9.10-14.amzn2023.0.2.x86_64
libXtst-1.2.3-14.amzn2023.0.2.x86_64
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
libxcb-1.13.1-7.amzn2023.0.2.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.

```
aws IAM EC2
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-compiler-manager/2.2/plexus-compiler-manager-2.2.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-incremental/1.1/maven-shared-incremental-1.1.jar (14 KB at 388.8 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-compiler-javac/2.2/plexus-compiler-javac-2.2.jar
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-compiler-api/2.2/plexus-compiler-api-2.2.jar (25 KB at 976.2 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/0.1/maven-shared-utils-0.1.jar (151 KB at 3772.4 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/apache/xbean/xbean-reflect/3.4/xbean-reflect-3.4.jar
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/1.5.1/plexus-utils-1.5.1.jar (206 KB at 4472.7 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven2/commons-logging/commons-logging-api/1.1/commons-logging-api-1.1.jar
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-compiler-manager/2.2/plexus-compiler-manager-2.2.jar (5 KB at 178.2 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/com/google/collections/google-collections/1.0/google-collections-1.0.jar
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-compiler-javac/2.2/plexus-compiler-javac-2.2.jar (19 KB at 475.0 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/commons-logging/commons-logging-api/1.1/commons-logging-api-1.1.jar (44 KB at 1037.0 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/junit/junit/3.8.2/junit-3.8.2.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/xbean/xbean-reflect/3.4/xbean-reflect-3.4.jar (131 KB at 2376.2 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/junit/junit/3.8.2/junit-3.8.2.jar (118 KB at 1840.8 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven2/log4j/log4j/1.2.12/log4j-1.2.12.jar (350 KB at 4371.2 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/com/google/collections/google-collections/1.0/google-collections-1.0.jar (625 KB at 6184.2 KB/sec)
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 3 source files to /home/ec2-user/Purdue_Project/target/classes
[INFO] BUILD SUCCESS
[INFO] Total time: 10.340 s
[INFO] Finished at: 2024-04-21T07:03:00+00:00
[INFO] Final Memory: 20M/49M
[ec2-user@ip-10-10-10-77 Purdue_Project]$
```

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

```
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/maven-surefire-common/2.12.4/maven-surefire-common-2.12.4.jar (257 KB at 1046.8 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/apache/commons/commons-lang3/3.1/commons-lang3-3.1.jar (309 KB at 1172.6 KB/sec)
[INFO] Surefire report directory: /home/ec2-user/Purdue_Project/target/surefire-reports
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.12.4/surefire-junit4-2.12.4.pom (3 KB at 235.4 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-providers/2.12.4/surefire-providers-2.12.4.pom (3 KB at 191.1 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.12.4/surefire-junit4-2.12.4.jar (37 KB at 2253.5 KB/sec)

T E S T S

Running com.abc.dataAccessObject.ProductImpTest
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.108 sec

Results :

Tests run: 4, Failures: 0, Errors: 0, Skipped: 0

[INFO] BUILD SUCCESS
[INFO] Total time: 7.357 s
[INFO] Finished at: 2024-04-21T07:04:34+00:00
[INFO] Final Memory: 20M/49M
[ec2-user@ip-10-10-10-77 Purdue_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.

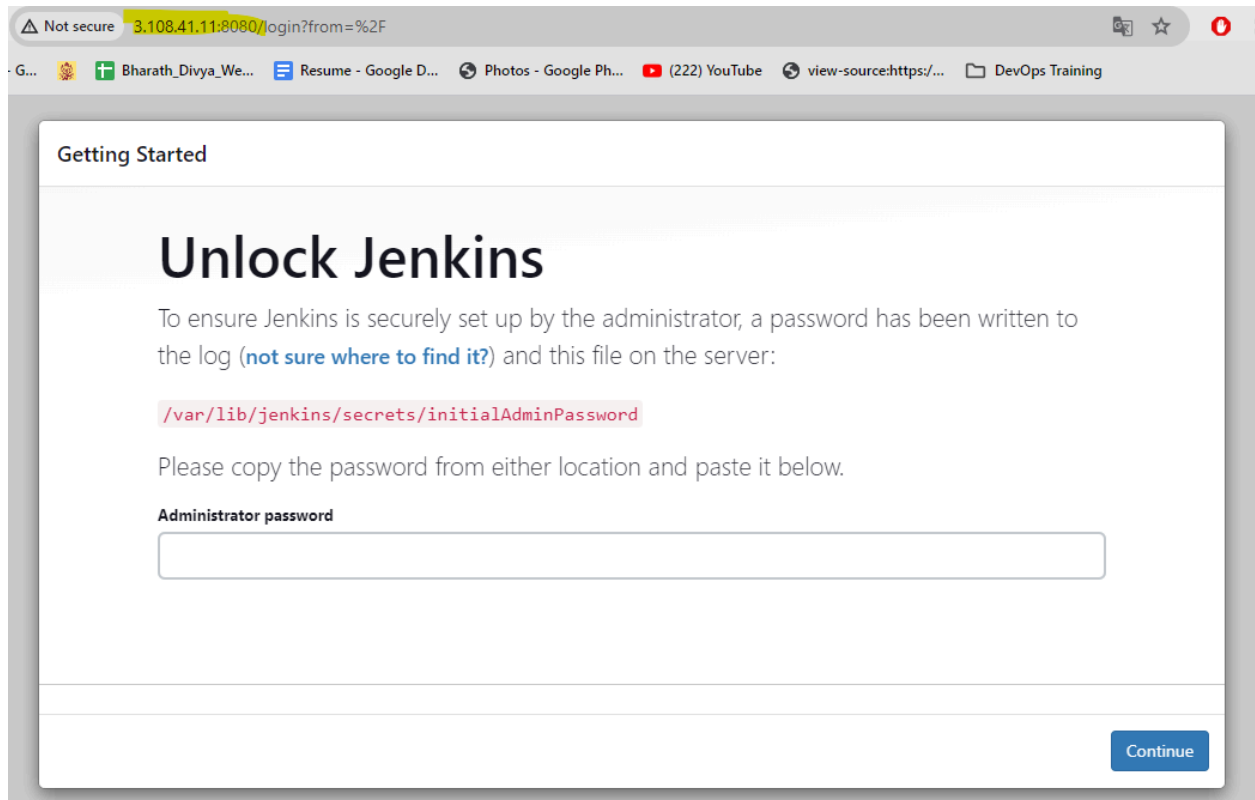
The screenshot displays the AWS Management Console interface for the EC2 service. The left sidebar shows the navigation menu with categories like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main content area is titled 'Instances (1/1)' and shows a table with one instance, 'MyPurdueInstance', which is in a 'Running' state. Below the table, the 'Details' tab for the instance 'i-05960029ef552a70f (MyPurdueInstance)' is selected, showing the 'Instance summary' section. This section lists the instance ID, public IPv4 address (52.108.41.111), private IPv4 addresses (10.10.10.77), and the instance state (Running).

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
MyPurdueInstance	i-05960029ef552a70f	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	-

i-05960029ef552a70f (MyPurdueInstance)

Instance summary

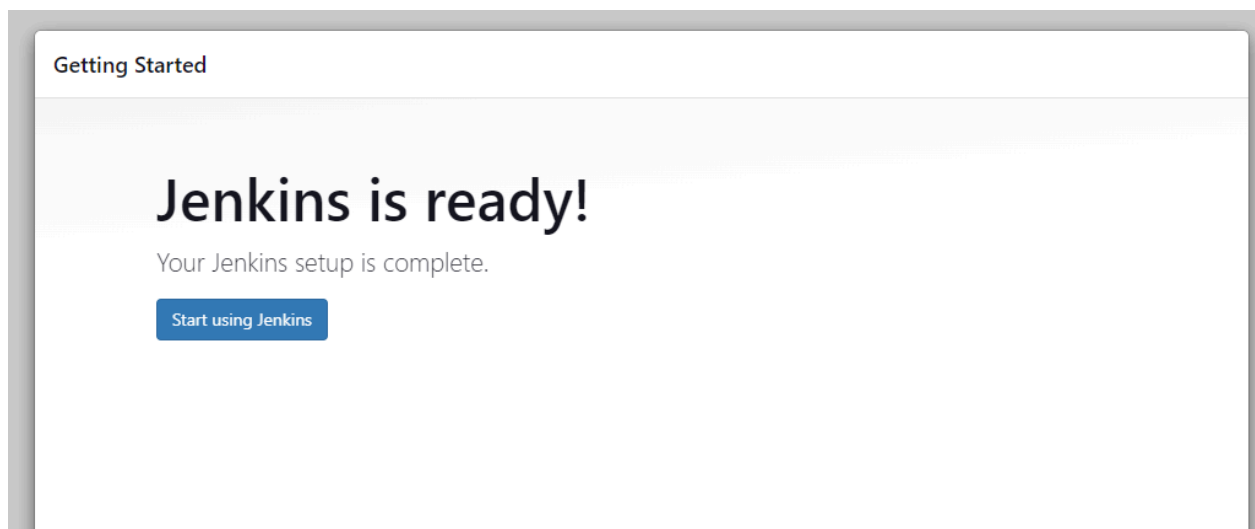
Instance ID	Public IPv4 address	Private IPv4 addresses
i-05960029ef552a70f (MyPurdueInstance)	52.108.41.111 open address	10.10.10.77
IPv6 address	Instance state	Public IPv4 DNS
-	Running	-



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.

Dashboard > Compile Source Code > Configuration

Configure

- General
- Source Code Management**
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Git ?

Repositories ?

Repository URL ?

`https://github.com/DivyaBharathwaj/Purdue_Project.git`

Credentials ?

- none -

+ Add

Advanced

Add Repository

Save Apply

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

Dashboard > Compile Source Code > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment**
- Build Steps
- Post-build Actions

☐ Use secret text(s) or file(s) ?
☐ Add timestamps to the Console Output
☐ Inspect build log for published build scans
☐ Terminate a build if it's stuck
☐ With Ant ?

Build Steps

Invoke top-level Maven targets ?

Goals

Advanced ▾

Save

Apply

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

Jenkins

Search (CTRL+K)

admin log out

Dashboard > Compile Source Code > Workspace

Status

</> Changes

Workspace

Wipe Out Current Workspace

Build Now

Configure

Delete Project

Rename

Workspace of Compile Source Code on Built-In Node

Compile Source Code /

.git			
.settings			
src			
target			
.classpath	Apr 27, 2024, 1:24:09 PM	1.01 KiB	
.project	Apr 27, 2024, 1:24:09 PM	545 B	
pom.xml	Apr 27, 2024, 1:24:09 PM	1.97 KiB	
pom.xml.bak	Apr 27, 2024, 1:24:09 PM	772 B	
README.md	Apr 27, 2024, 1:24:09 PM	40 B	

(all files in zip)

Build History

trend ▾

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

Dashboard > Verify and Test Code > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

Invoke top-level Maven targets ?

Goals

Advanced ▾

Add build step ▾

Post-build Actions

Add post-build action ▾

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

← → ↻ Not secure 3.108.41.11:8080/manage/pluginManager/installed

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 > Manage Jenkins > Plugins

Plugins

Q Jaco /

Name ↓	Enabled
JaCoCo plugin 3.3.6 This plugin integrates JaCoCo code coverage reports to Jenkins. Report an issue with this plugin	<input checked="" type="checkbox"/>

← → ↻ ⚠ 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

git

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 .

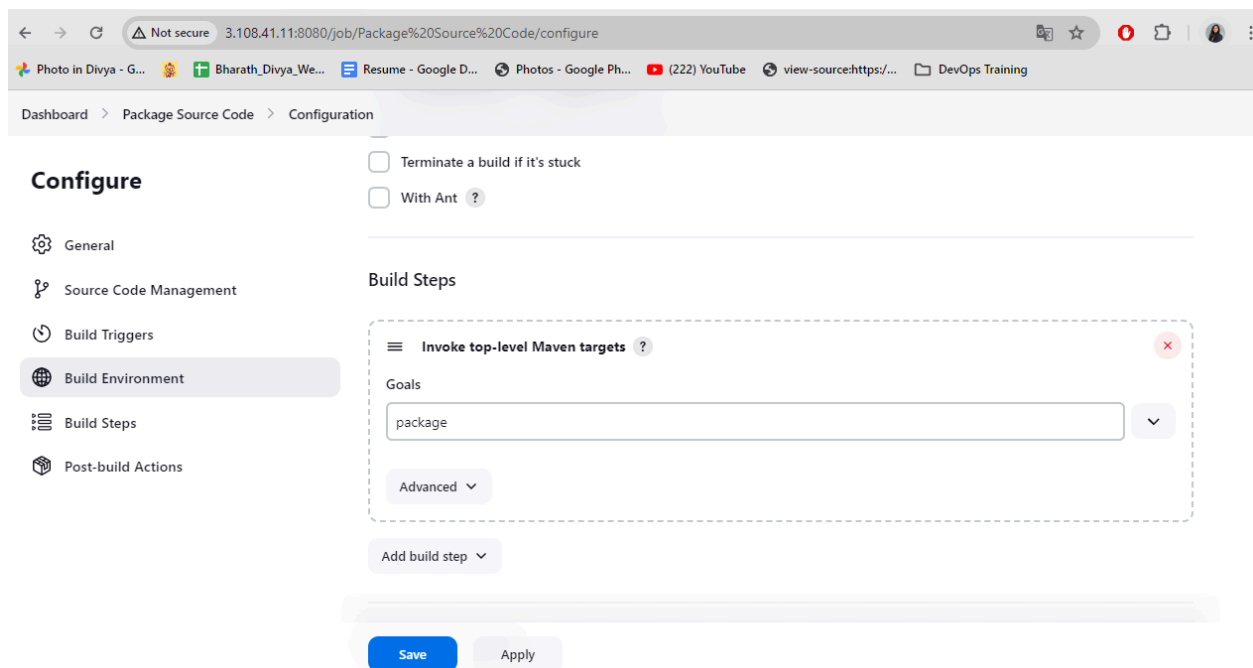
The following dependency should be added in the pom.xml if not there already

```
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.

```
Dashboard > Package Source Code > #1 > Console Output

[INFO] Packaging webapp
[INFO] Assembling webapp [ABCtechnologies] in [/var/lib/jenkins/workspace/Package Source Code/target/ABCtechnologies-1.0]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/Package Source Code/src/main/webapp]
[INFO] Webapp assembled in [139 msecs]
[INFO] Building war: /var/lib/jenkins/workspace/Package Source Code/target/ABCtechnologies-1.0.war
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.6:report (jacoco-site) @ ABCtechnologies ---
[INFO] Loading execution data file /var/lib/jenkins/workspace/Package Source Code/target/jacoco.exec
[INFO] Analyzed bundle 'RetailModule' with 2 classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 8.993 s
[INFO] Finished at: 2024-04-27T13:42:00+00:00
[INFO] Final Memory: 23M/83M
[INFO] -----
Finished: SUCCESS
```

All 3 created jobs in a glance:

The screenshot shows the Jenkins dashboard with the following components:

- Header:** Jenkins logo, search bar, and user 'admin' with a 'log out' button.
- Left Sidebar:** Navigation menu with 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'My Views'.
- Build Queue:** A dropdown menu showing 'No builds in the queue.'
- Build Executor Status:** A dropdown menu.
- Job List:** A table with columns: S, W, Name, Last Success, Last Failure, Last Duration, and a play button icon.
- Footer:** 'Icon: S M L', 'Icon legend', and three Atom feed links.

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☁	Compile Source Code	20 min #2	22 min #1	13 sec	▶
✓	☀	Package Source Code	5 min 5 sec #1	N/A	12 sec	▶
✓	☀	Verify and Test Code	11 min #1	N/A	14 sec	▶

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

Dashboard > CI CD Pipeline > Configuration

Configure

- General
- Advanced Project Options
- Pipeline**

Pipeline

Definition

Pipeline script

```

1 pipeline{
2
3   tools{
4     // what tool version to use for build stages
5     maven 'mymaven'
6   }
7
8   agent any
9
10  stages{
11
12    stage ('CloneRepo')
13
14    {
15      steps{
16        echo 'This is stage 1'
17        git 'https://github.com/Sonal0409/DevOpsCodeDemo.git'
18      }
19    }
20  }
21 }

```

Save Apply

Dashboard > CI CD Pipeline >

CI CD Pipeline

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

Add description

Disable Project

Stage View

Average stage times:
(Average full run time: ~30s)

	Declarative: Tool Install	CloneRepo	Compile	Test	package
#4 Apr 27 20:35 No Changes	95ms	985ms	6s	9s	11s
#3 Apr 27 20:32 No Changes	197ms	1s	7s	9s	11s

Build History

Filter...

#4
Apr 27 20:35
No Changes

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

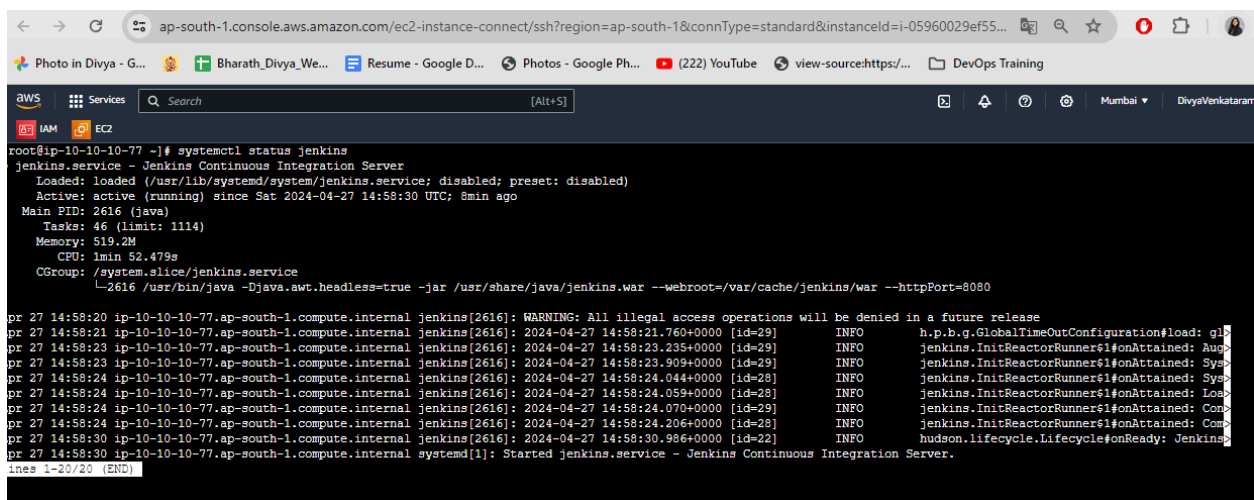
Created a user and ssh keys on

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::

A screenshot of the AWS Management Console showing the Jenkins service status on an EC2 instance. The top navigation bar includes the AWS logo, 'Services' dropdown, a search bar, and user information for 'DivyaVenkataram'. The main content area shows the 'Jenkins.service' status as 'loaded' and 'active (running)'. Below this, the command used to start the service is displayed: 'systemctl status jenkins'. The output shows the service is running on PID 2616. The bottom section shows logs for the Jenkins service, including a warning about illegal access operations and several INFO messages from the Jenkins reactor runners.

```
root@ip-10-10-10-77 ~# systemctl status jenkins
jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; preset: disabled)
   Active: active (running) since Sat 2024-04-27 14:58:30 UTC; 8min ago
     Main PID: 2616 (java)
       Task: 46 (limit: 1114)
      Memory: 519.2M
         CPU: 1min 52.479s
    CGroup: /system.slice/jenkins.service
            └─2616 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

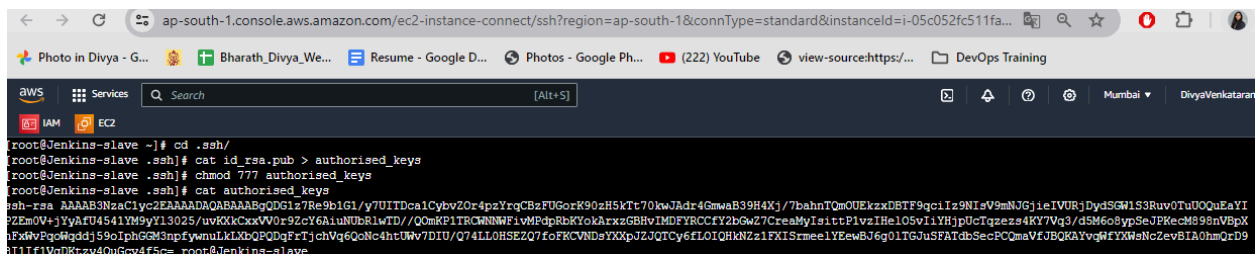
pr 27 14:58:20 ip-10-10-10-77.ap-south-1.compute.internal jenkins[2616]: WARNING: All illegal access operations will be denied in a future release
pr 27 14:58:21 ip-10-10-10-77.ap-south-1.compute.internal jenkins[2616]: 2024-04-27 14:58:21.760+0000 [id=29] INFO h.p.b.g.GlobalTimeOutConfiguration#load: gbl
pr 27 14:58:23 ip-10-10-10-77.ap-south-1.compute.internal jenkins[2616]: 2024-04-27 14:58:23.235+0000 [id=29] INFO jenkins.InitReactorRunner$1onAttained: Aug
pr 27 14:58:23 ip-10-10-10-77.ap-south-1.compute.internal jenkins[2616]: 2024-04-27 14:58:23.909+0000 [id=29] INFO jenkins.InitReactorRunner$1onAttained: Sys
pr 27 14:58:24 ip-10-10-10-77.ap-south-1.compute.internal jenkins[2616]: 2024-04-27 14:58:24.044+0000 [id=28] INFO jenkins.InitReactorRunner$1onAttained: Sys
pr 27 14:58:24 ip-10-10-10-77.ap-south-1.compute.internal jenkins[2616]: 2024-04-27 14:58:24.059+0000 [id=28] INFO jenkins.InitReactorRunner$1onAttained: Lob
pr 27 14:58:24 ip-10-10-10-77.ap-south-1.compute.internal jenkins[2616]: 2024-04-27 14:58:24.070+0000 [id=29] INFO jenkins.InitReactorRunner$1onAttained: Com
pr 27 14:58:24 ip-10-10-10-77.ap-south-1.compute.internal jenkins[2616]: 2024-04-27 14:58:24.206+0000 [id=28] INFO jenkins.InitReactorRunner$1onAttained: Com
pr 27 14:58:30 ip-10-10-10-77.ap-south-1.compute.internal jenkins[2616]: 2024-04-27 14:58:30.986+0000 [id=22] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins
pr 27 14:58:30 ip-10-10-10-77.ap-south-1.compute.internal systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
```

Java installed in slave and created ssh key and user:

```
[root@Jenkins-slave tmp]# useradd js1
[root@Jenkins-slave tmp]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key's fingerprint is:
SHA256:aq48AGU7DdzUo0q4a7XC5dxUgpVA7paGcveFtoC6l7s root@Jenkins-slave
The key's randomart image is:
+---[RSA 3072]-----+
| ..=O.. |
| * .oo |
| + =O. . |
| O *O+. O |
| |.=oBo = S |
| oo=+ = + |
| O.=O+ = |
| |.=O=.+ |
| O.Eooo. |
+---[SHA256]-----+
```

```
[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
AAAAB3NzaC1yc2EAAAADAQABAAQGDG1z7Re9b1G1/y7UITDca1CybvZOr4pzYrq
CBzFUGorK90zH5kTt70kwJAdr4GmwaB39H4Xj/7bahnTQmOUEkzxDBTF9qciIz9NIsV9
mNJGjieIVURjDydSGW1S3Ruv0TuUOQuEaYIPZEm0V+jYyAfU4541YM9yYl3025/uvKX
kCxxVV0r9ZcY6AiuNUbRlwTD//QOmKP1TRCWNWFiwMPdpRbKYokArxzGBHvIMDF
YRCCfY2bGwZ7CreaMyIsittP1vzIHelO5vliYHjpUcTqzezs4KY7Vq3/d5M6o8ypSeJPKec
M898nVBpXnFxWvPqoWqddj59oIphGGM3npfywnuLkLXbQPQDqFrTjchVq6QoNc4htU
Wv7DIU/Q74LL0HSEZQ7foFKCVNDsYXXpJZJQTCy6fLOIQHkNZz1FXISrmeelYEewBJ
6g0ITGJuSFATdbSecPCQmaVfJBQKAYvqWfYXWsNcZevBIA0hmQrD98I1If1VgDKtzy4
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
AAAAB3NzaC1yc2EAAAADAQABAAQGDG1z7Re9b1G1/y7UITDca1CybvZOr4pzYrqCBzFUGorK90zH5kTt70kwJAdr4GmwaB
39H4Xj/7bahnTQmOUEkzxDBTF9qciIz9NIsV9mNJGjieIVURjDydSGW1S3Ruv0TuUOQuEaYIPZEmoV+jYyAfU4541YM9yYl30
25/uvKXkCxxVV0r9ZcY6AiuNUbRlwTD//QOmKP1TRCWNWFiwMPdpRbKYokArxzGBHvIMDFYRCCfY2bGwZ7CreaMyIsittP1
vzIHelO5vliYHjpUcTqzezs4KY7Vq3/d5M6o8ypSeJPKecM898nVBpXnFxWvPqoWqddj59oIphGGM3npfywnuLkLXbQPQDqFrTj
chVq6QoNc4htUWv7DIU/Q74LL0HSEZQ7foFKCVNDsYXXpJZJQTCy6fLOIQHkNZz1FXISrmeelYEewBJ6golTGJuSFATdbSecP
CQmaVfJBQKAYvqWfYXWsNcZevBIAohmQrD98I1If1VgDKtzy4QuGcv4f5c= root@Jenkins-slave"
>/var/lib/jenkins/.ssh/known_hosts
[root@jenkinsmaster ~]# cat /var/lib/jenkins/.ssh/known_hosts
```

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGDG1z7Re9b1G1/y7UITDca1CybvZOr4pzYrqCBzFUGorK90zH5kTt70kwJAdr4GmwaB
39H4Xj/7bahnTQmOUEkzxDBTF9qciIz9NIsV9mNJGjieIVURjDydSGW1S3Ruv0TuUOQuEaYIPZEmoV+jYyAfU4541YM9yYl30
25/uvKXkCxxVV0r9ZcY6AiuNUbRlwTD//QOmKP1TRCWNWFiwMPdpRbKYokArxzGBHvIMDFYRCCfY2bGwZ7CreaMyIsittP1
vzIHelO5vliYHjpUcTqzezs4KY7Vq3/d5M6o8ypSeJPKecM898nVBpXnFxWvPqoWqddj59oIphGGM3npfywnuLkLXbQPQDqFrTj
chVq6QoNc4htUWv7DIU/Q74LL0HSEZQ7foFKCVNDsYXXpJZJQTCy6fLOIQHkNZz1FXISrmeelYEewBJ6golTGJuSFATdbSecP
CQmaVfJBQKAYvqWfYXWsNcZevBIAohmQrD98I1If1VgDKtzy4QuGcv4f5c= 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 it 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) 🛡️ 1 👤 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) 🔔 1 🛡️ 1 👤 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

Linux_Agent_Slave

Monitoring Data ▾

Labels
jenkins_slave

Projects tied to Linux_Agent_Slave
None

Mark this node temporarily offline

🔍

Edit description

Dashboard > Pipeline CI CD Maven App >

Status ✓ Pipeline CI CD Maven App

</> Changes

▶ Build Now

⚙️ Configure

🗑️ Delete Pipeline

🔍 Full Stage View

✎ Rename

❓ Pipeline Syntax

Build History trend ▾

Filter...

✓ #6

Apr 28 08:30 No Changes

Apr 28 08:29 No Changes

Stage View

	checkout	Execute Maven clean	Test code	Package Code
Average stage times: (Average full run time: ~30s)	2s	2s	7s	4s
#6	729ms	3s	12s	8s
#5	807ms	4s	18s	9s

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

Dashboard > Pipeline CI CD Maven App > #6

Status ✓ Console Output

</> Changes

📄 Console Output

📄 View as plain text

📄 Edit Build Information

🗑️ Delete build '#6'

📄 Git Build Data

🔄 Restart from Stage

🔄 Replay

📄 Pipeline Steps

📄 Workspaces

```

Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Linux Agent Slave in /tmp/jenkinsdir/workspace/Pipeline CI CD Maven App
[Pipeline] {
[Pipeline] stage
[Pipeline] { (checkout)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Fetching changes from the remote Git repository
Checking out Revision d8f5184ac9346cf51a3105963e01cabd198f02cd (refs/remotes/origin/master)
Commit message: "First Commit"
> git rev-parse --resolve-git-dir /tmp/jenkinsdir/workspace/Pipeline CI CD Maven App/.git # timeout=10
> git config remote.origin.url https://github.com/DivyaBharathwaj/Purdue_Project.git # timeout=10
Fetching upstream changes from https://github.com/DivyaBharathwaj/Purdue_Project.git
> git --version # timeout=10
> git --version # 'git version 2.40.1'

```

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'
  }
}
}
}
}
}

```

Committed 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

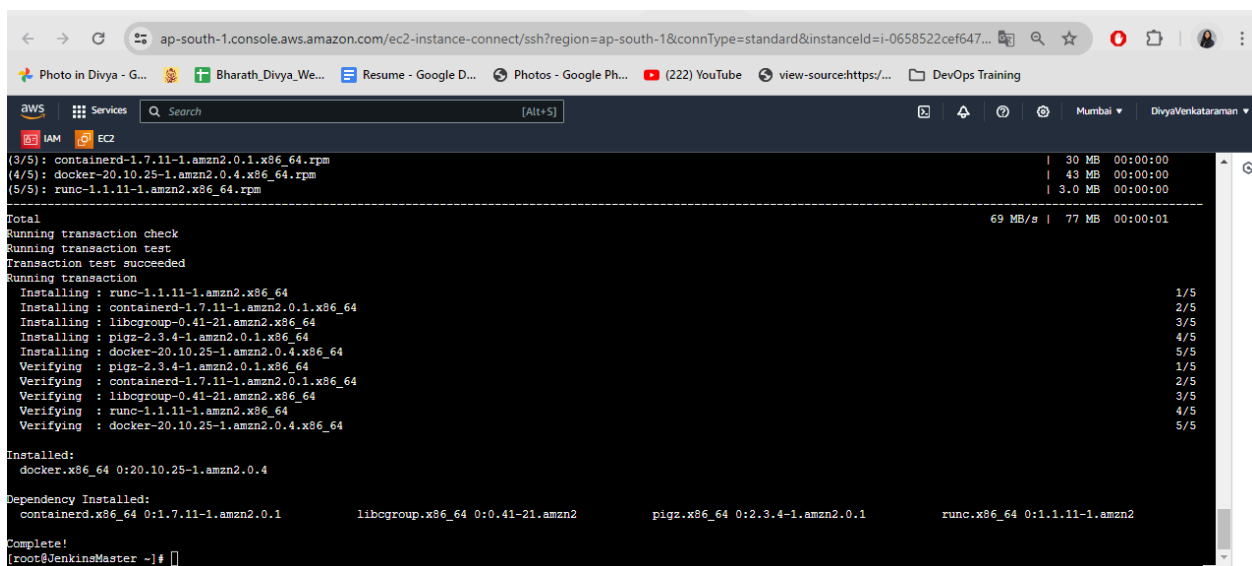
Create a Docker-hub Account

Install Docker in VM Machine

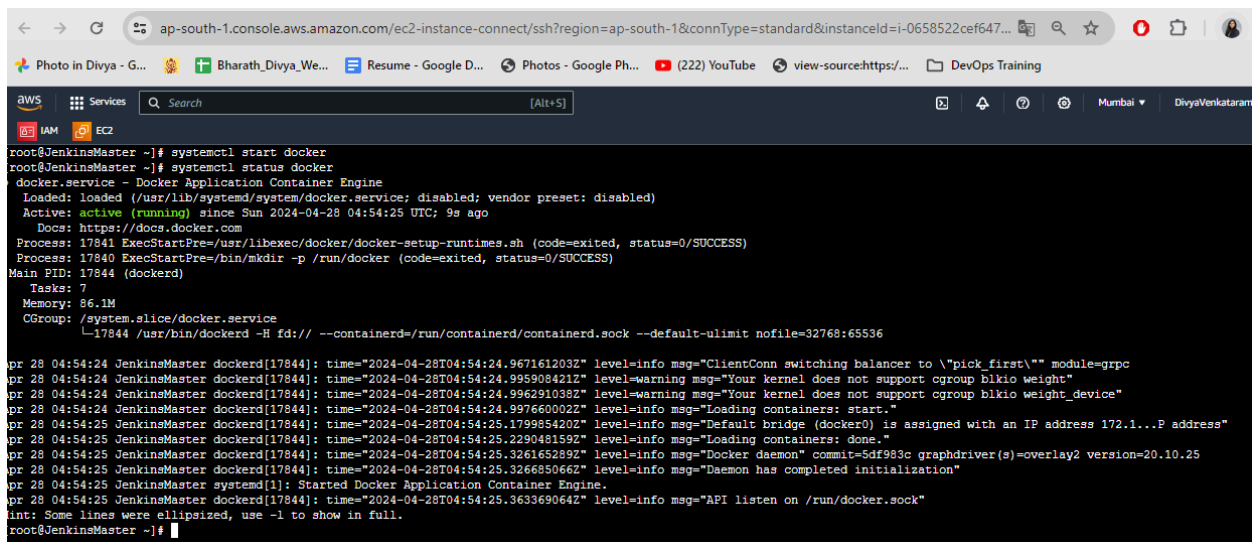
The script for the same is committed with the file name :: installdocker.sh

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

Successful Installation of Docker



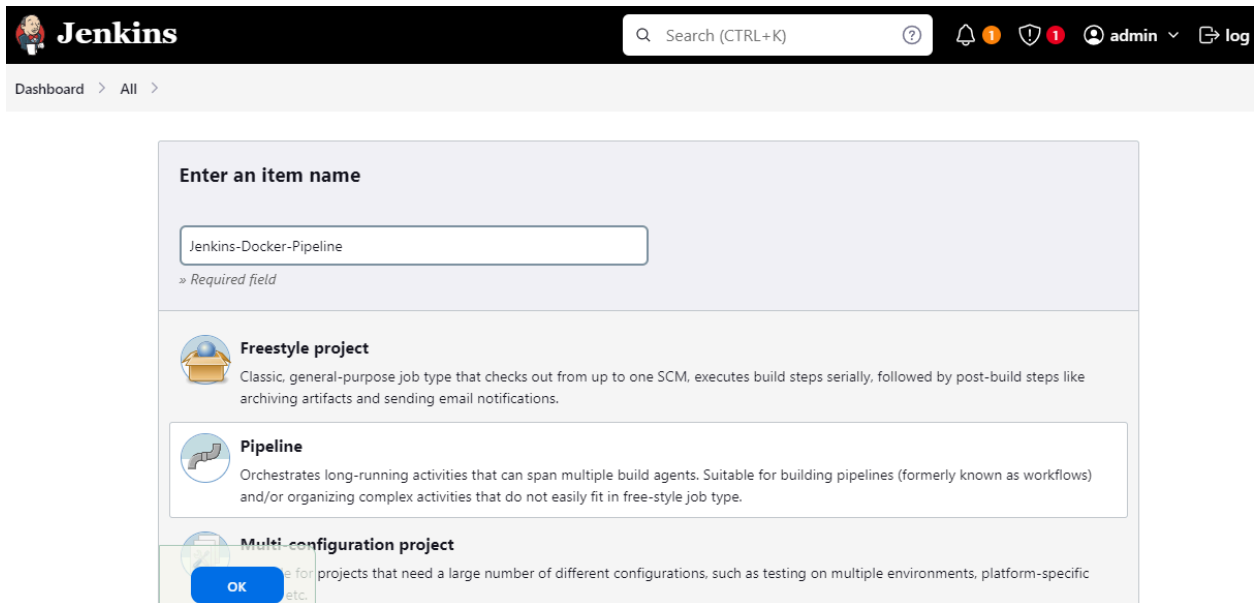
```
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-0658522cef647...
Photo in Divya - G... Bharath_Divya_We... Resume - Google D... Photos - Google Ph... (222) YouTube view-source:https:// DevOps Training
AWS IAM EC2
(3/5): containerd-1.7.11-1.amzn2.0.1.x86_64.rpm | 30 MB 00:00:00
(4/5): docker-20.10.25-1.amzn2.0.4.x86_64.rpm | 43 MB 00:00:00
(5/5): runc-1.1.11-1.amzn2.x86_64.rpm | 3.0 MB 00:00:00
Total 69 MB/s | 77 MB 00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : runc-1.1.11-1.amzn2.x86_64 1/5
Installing : containerd-1.7.11-1.amzn2.0.1.x86_64 2/5
Installing : libcgrou-0.41-21.amzn2.x86_64 3/5
Installing : pigz-2.3.4-1.amzn2.0.1.x86_64 4/5
Installing : docker-20.10.25-1.amzn2.0.4.x86_64 5/5
Verifying : pigz-2.3.4-1.amzn2.0.1.x86_64 1/5
Verifying : containerd-1.7.11-1.amzn2.0.1.x86_64 2/5
Verifying : libcgrou-0.41-21.amzn2.x86_64 3/5
Verifying : runc-1.1.11-1.amzn2.x86_64 4/5
Verifying : docker-20.10.25-1.amzn2.0.4.x86_64 5/5
Installed:
docker.x86_64 0:20.10.25-1.amzn2.0.4
Dependency Installed:
containerd.x86_64 0:1.7.11-1.amzn2.0.1 libcgrou.x86_64 0:0.41-21.amzn2 pigz.x86_64 0:2.3.4-1.amzn2.0.1 runc.x86_64 0:1.1.11-1.amzn2
Complete!
[root@JenkinsMaster ~]#
```



```
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-0658522cef647...
Photo in Divya - G... Bharath_Divya_We... Resume - Google D... Photos - Google Ph... (222) YouTube view-source:https:// DevOps Training
AWS IAM EC2
root@JenkinsMaster ~]# systemctl start docker
root@JenkinsMaster ~]# systemctl status docker
docker.service - Docker Application Container Engine
Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
Active: active (running) since Sun 2024-04-28 04:54:25 UTC; 9s ago
Docs: https://docs.docker.com
Process: 17841 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
Process: 17840 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
Main PID: 17844 (dockerd)
Tasks: 7
Memory: 86.1M
CGroup: /system.slice/docker.service
└─17844 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536
apr 28 04:54:24 JenkinsMaster dockerd[17844]: time="2024-04-28T04:54:24.967161203Z" level=info msg="ClientConn switching balancer to \\"pick_first\\" module=grpc
apr 28 04:54:24 JenkinsMaster dockerd[17844]: time="2024-04-28T04:54:24.995908421Z" level=warning msg="Your kernel does not support cgroup blkio weight"
apr 28 04:54:24 JenkinsMaster dockerd[17844]: time="2024-04-28T04:54:24.996291038Z" level=warning msg="Your kernel does not support cgroup blkio weight_device"
apr 28 04:54:24 JenkinsMaster dockerd[17844]: time="2024-04-28T04:54:24.997660002Z" level=info msg="Loading containers: start."
apr 28 04:54:25 JenkinsMaster dockerd[17844]: time="2024-04-28T04:54:25.179985420Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.1...P address"
apr 28 04:54:25 JenkinsMaster dockerd[17844]: time="2024-04-28T04:54:25.229048159Z" level=info msg="Loading containers: done."
apr 28 04:54:25 JenkinsMaster dockerd[17844]: time="2024-04-28T04:54:25.326165289Z" level=info msg="Docker daemon" commit=5df983c graphdriver(s)=overlay2 version=20.10.25
apr 28 04:54:25 JenkinsMaster dockerd[17844]: time="2024-04-28T04:54:25.326685066Z" level=info msg="Daemon has completed initialization"
apr 28 04:54:25 JenkinsMaster systemd[1]: Started Docker Application Container Engine.
apr 28 04:54:25 JenkinsMaster dockerd[17844]: time="2024-04-28T04:54:25.363369064Z" level=info msg="API listen on /run/docker.sock"
hint: Some lines were ellipsized, use -l to show in full.
root@JenkinsMaster ~]#
```

CI-CD pipeline creation.

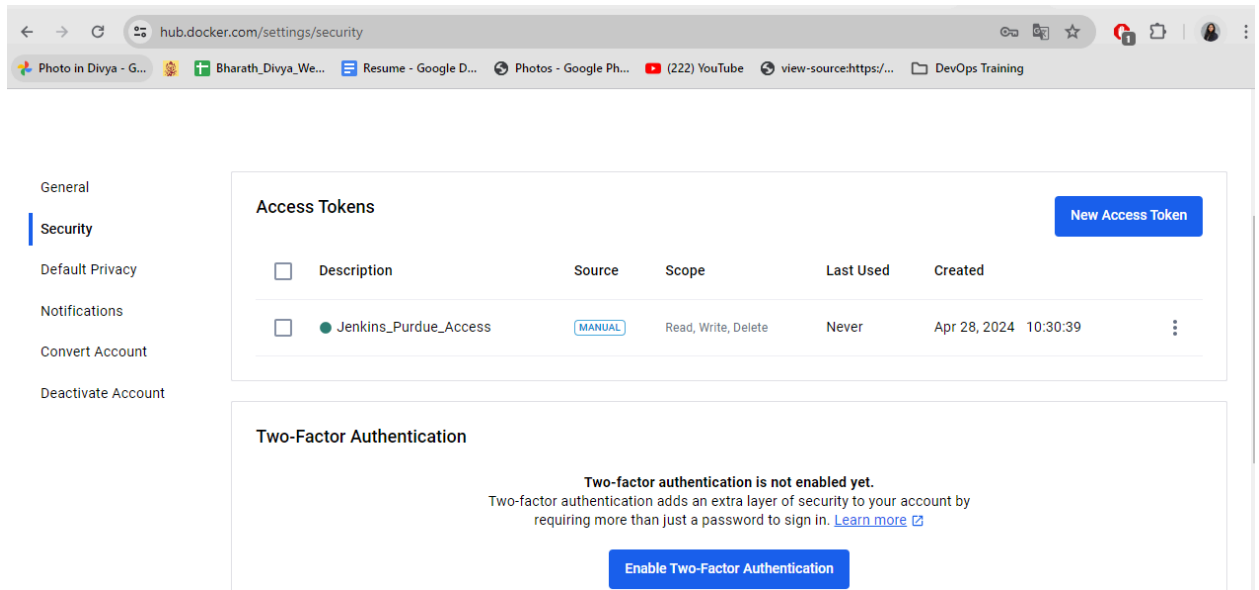
1. Create new item ☐ Pipeline ☐ provide a name .



The screenshot shows the Jenkins dashboard. At the top, there's a header with the Jenkins logo, a search bar, and user information (admin). Below the header, there's a breadcrumb trail: Dashboard > All >. The main content area is titled "Enter an item name" and contains a text input field with "Jenkins-Docker-Pipeline" entered. Below the input field, there's a note: » Required field. Underneath, there are three project type options: "Freestyle project" (Classic, general-purpose job type), "Pipeline" (Orchestrates long-running activities), and "Multi-configuration project" (For projects that need a large number of different configurations). Each option has a brief description and an icon. At the bottom left of the options, there's a blue "OK" button.

2. Add Docker Hub access token into Jenkins

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











The screenshot shows the Docker Hub security settings page. The browser address bar shows "hub.docker.com/settings/security". The left sidebar has a menu with "General", "Security" (selected), "Default Privacy", "Notifications", "Convert Account", and "Deactivate Account". The main content area is titled "Access Tokens" and has a "New Access Token" button. Below this, there's a table with columns: "Description", "Source", "Scope", "Last Used", and "Created". The table contains one entry: "Jenkins_Purdue_Access" with a "MANUAL" source, "Read, Write, Delete" scope, "Never" last used, and "Apr 28, 2024 10:30:39" created. Below the table, there's a section titled "Two-Factor Authentication" with a message: "Two-factor authentication is not enabled yet. Two-factor authentication adds an extra layer of security to your account by requiring more than just a password to sign in. [Learn more](#)". At the bottom of this section is a blue "Enable Two-Factor Authentication" button.

3. Open Manage Jenkins → Manage Credentials and Add the System Credentials into Jenkins

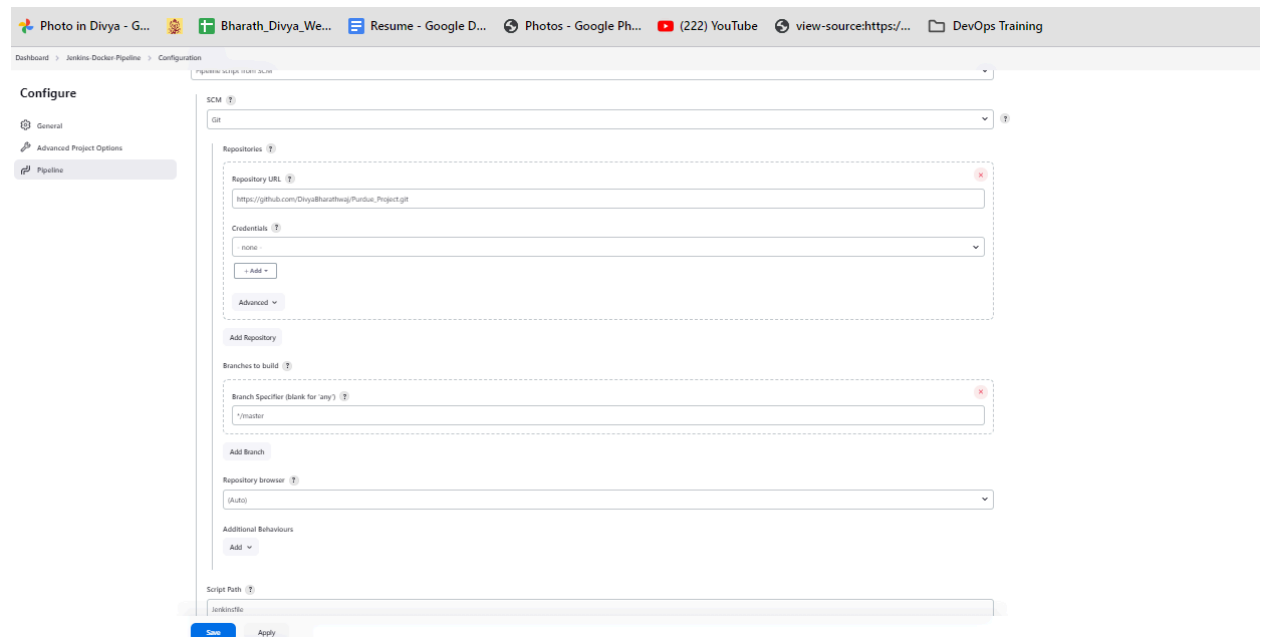
Global credentials (unrestricted)

[+ Add Credentials](#)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description 1	
 jenkins	divyabharathwaj/***** (jenkins)	Username with password	jenkins	
 agentid	ec2-user (agentid)	SSH Username with private key	agentid	
 jenkinsagentid	ec2-user (agentcredentials)	SSH Username with private key	agentcredentials	
 agentcredentials	ec2-user (agentcredentials)	SSH Username with private key	agentcredentials	

- 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.



The screenshot shows the Jenkins Pipeline Configuration page. The left sidebar has a 'Configure' section with 'General', 'Advanced Project Options', and 'Pipeline' (selected). The main area is titled 'Pipeline' and contains the following fields:

- SCM:** Set to 'Git'.
- Repository URL:** `https://github.com/Divyabharathwaj/Pondus_Project.git`
- Credentials:** Set to 'none'.
- Advanced:** A dropdown menu.
- Add Repository:** A button.
- Branches to build:** A section with a 'Branch Specifier (blank for 'any')' field set to `*/master`.
- Add Branch:** A button.
- Repository browser:** Set to 'Auto'.
- Additional Behaviours:** A section with an 'Add' button.
- Script Path:** Set to `jenkinsfile`.

At the bottom, there are 'Save' and 'Apply' buttons.

What is the Jenkins pipeline stages that we have written?

- We will checkout our git repo.
- Run maven clean install and package it to war.
- Run docker build to create a image and tag.
- Login to Docker hub and push the image.
- 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/](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"
    }
}
}
}

```

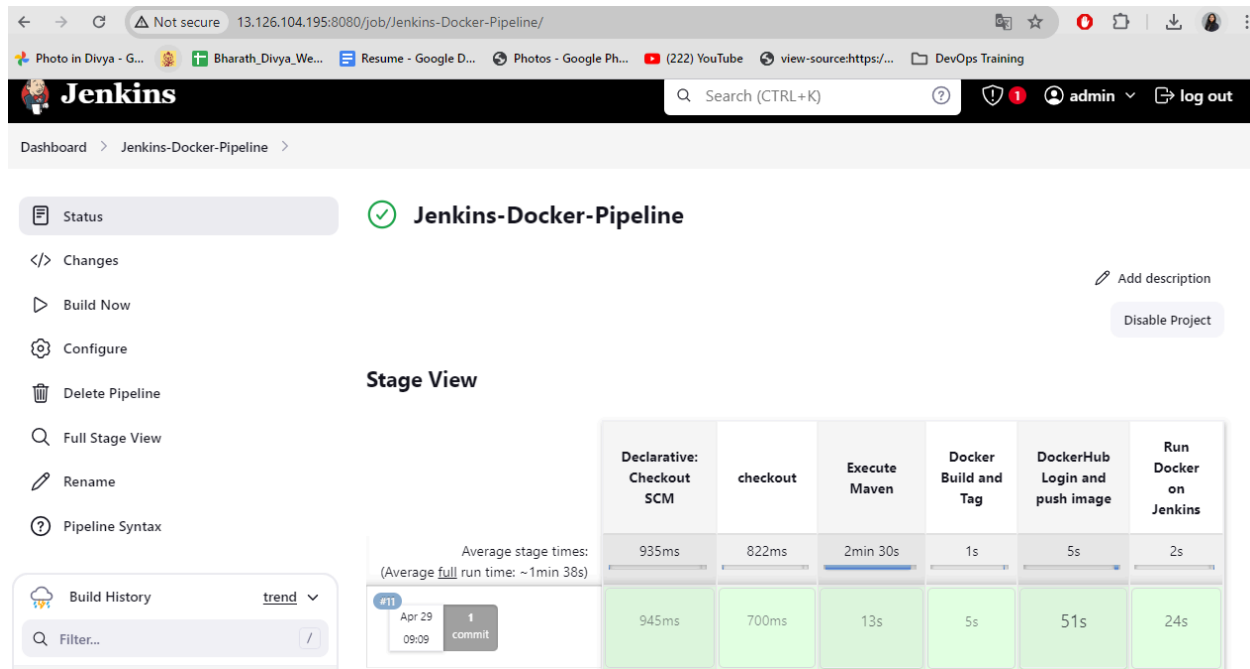
3.Add DockerFile with following commands.

```

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

```

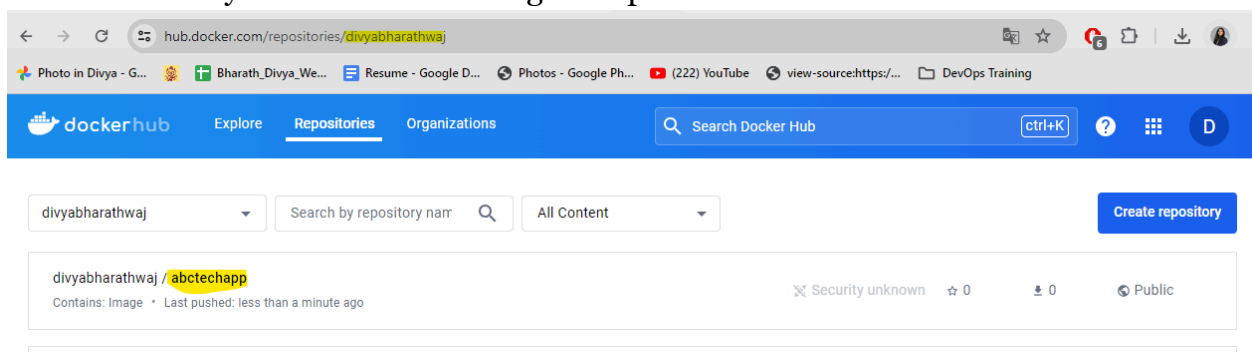
4.Screenshot of the various stages in build.



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 container. 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 radically simple IT automation system. It handles configuration-management, application deployment, cloud provisioning, ad-hoc task-execution, and multinode orchestration - including trivializing things like zero-downtime rolling updates with load balancers.

Set up on Ansible Controller Machine:

```
root@AnsibleController ~# ansible --version
ansible 2.9.27
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /bin/ansible
  python version = 2.7.18 (default, Dec 18 2023, 22:08:43) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]
root@AnsibleController ~#
```

Successful connection from Ansible Controller to Ansible Host:

```
[ec2-user@AnsibleController ~]$ ssh 'ec2-user@172.31.32.111'
Last login: Mon Apr 29 16:05:51 2024 from ec2-13-233-177-4.ap-south-1.compute.amazonaws.com

      _   _          _   _
     / \   \        / \   \
    /   \_\/        /   \_\/
   /____/ \_      /____/ \_
  /_____/   \    /_____/   \
 /         _\  /         _\
/_        /  _/_        /  _/
/_       /_/_/_       /_/_/_
/_      /___/_/_     /___/_/_
/_     /___/_/_/_   /___/_/_/_
/_    /___/_/_/_/_  /___/_/_/_/_
/_   /___/_/_/_/_/_/___/_/_/_/_/

      Amazon Linux 2

      AL2 End of Life is 2025-06-30.

      A newer version of Amazon Linux is available!

      Amazon Linux 2023, GA and supported until 2028-03-15.
      https://aws.amazon.com/linux/amazon-linux-2023/

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

The ansible set up is now successfully complete.

```
[ec2-user@ansiblecontroller ansible]$ ansible 172.31.32.111 -m ping
[WARNING]: Platform linux on host 172.31.32.111 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could
change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
172.31.32.111 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

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 from 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 return 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-container"], "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 16:32:25.836608", "stderr": "Error: No such image: jayjodev/devops-image", "stderr_lines": ["Error: No such image: jayjodev/devops-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

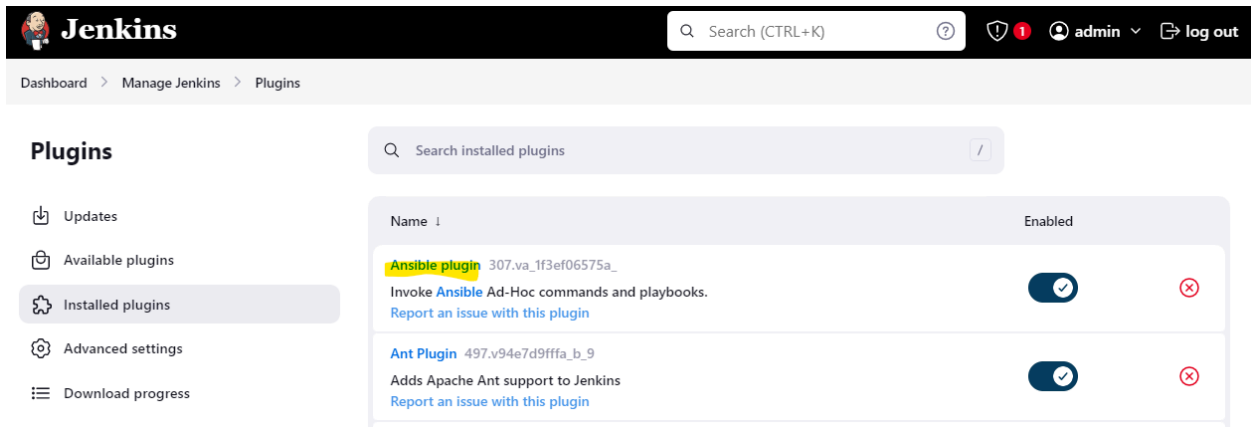
[ec2-user@JenkinsMaster ~]$
```

Verify Docker Container:

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

Ansible installations ^ Edited

Add Ansible

Ansible

Name

Ansible

Path to ansible executables directory

/var/lib/jenkins

☐ Install automatically ?

Add Ansible



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:

Cloud Shell Editor

Use the Legacy Editor

(aerial-gadget-412311) X +

Terminal tabs have been recovered from an existing session. Dismiss

Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to **aerial-gadget-412311**.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
divyaraman993@cloudshell:~ (aerial-gadget-412311)\$ gcloud container clusters get-credentials edureka-project-cluster --zone us-central1-c --project aerial-gadget-412311
Fetching cluster endpoint and auth data.
kubeconfig entry generated for edureka-project-cluster.
divyaraman993@cloudshell:~ (aerial-gadget-412311)\$ kubectl get nodes -o wide

NAME	KERNEL-VERSION	CONTAINER-RUNTIME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE
gke-edureka-project-clus-default-pool-08119c66-9x4m	6.1.58+	containerd://1.7.10	Ready	<none>	3m1s	v1.28.7-gke.1026000	10.128.0.3	34.41.118.229	Container-Optimized OS from G
oogle 6.1.58+	6.1.58+	containerd://1.7.10	Ready	<none>	2m59s	v1.28.7-gke.1026000	10.128.0.5	35.223.11.106	Container-Optimized OS from G
gke-edureka-project-clus-default-pool-08119c66-pnld	6.1.58+	containerd://1.7.10	Ready	<none>	3m	v1.28.7-gke.1026000	10.128.0.4	34.31.29.157	Container-Optimized OS from G
oogle 6.1.58+	6.1.58+	containerd://1.7.10	Ready	<none>					

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           4ms   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
pod/abctech-deployment-65f95cdfbc-8ppjt	1/1	Running	0	3m32s	10.76.1.6	gke-edureka-project-clus-default-pool-08119c66-qxc9	<none>
pod/abctech-deployment-65f95cdfbc-cxf7l	1/1	Running	0	3m32s	10.76.2.5	gke-edureka-project-clus-default-pool-08119c66-pnld	<none>
pod/abctech-deployment-65f95cdfbc-frmhl	1/1	Running	0	3m32s	10.76.0.13	gke-edureka-project-clus-default-pool-08119c66-9x4m	<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

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



Ansible Playbook for K8 deployment

```
---
- hosts: all
  become: true
  tasks:
    - name: Create Production namespace
      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. Install 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].metadata.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].metadata.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-dqhgq 1/1 Running 0 2m18s
pod/prometheus-prometheus-node-exporter-6js67 1/1 Running 0 2m19s
pod/prometheus-prometheus-node-exporter-sjmw1 1/1 Running 0 2m19s
pod/prometheus-prometheus-node-exporter-wf9ax 1/1 Running 0 2m19s
pod/prometheus-prometheus-pushgateway-568fbf799-bpr6w 1/1 Running 0 2m18s
pod/prometheus-server-579dc9cfd9-sb5sx 2/2 Running 0 2m18s

```

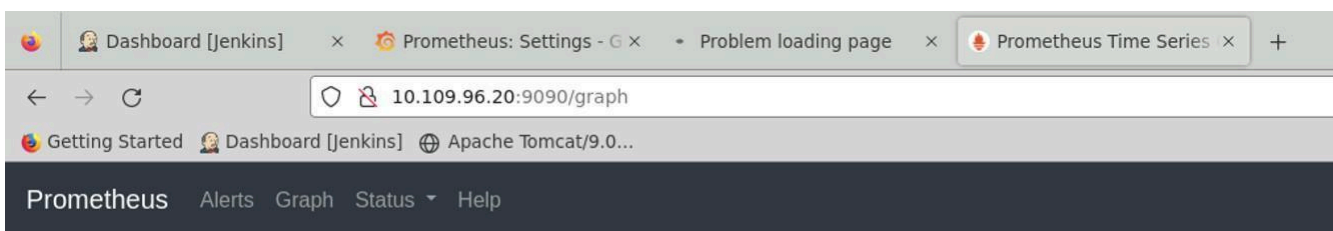
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/abctech-service	LoadBalancer	10.91.10.80	34.170.48.146	8082:30722/TCP	12m
service/kubernetes	ClusterIP	10.91.0.1	<none>	443/TCP	29m
service/prometheus-alertmanager	ClusterIP	10.91.1.12	<none>	9093/TCP	2m20s
service/prometheus-alertmanager-headless	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-579dc9cfd9	1	1	1	2m20s

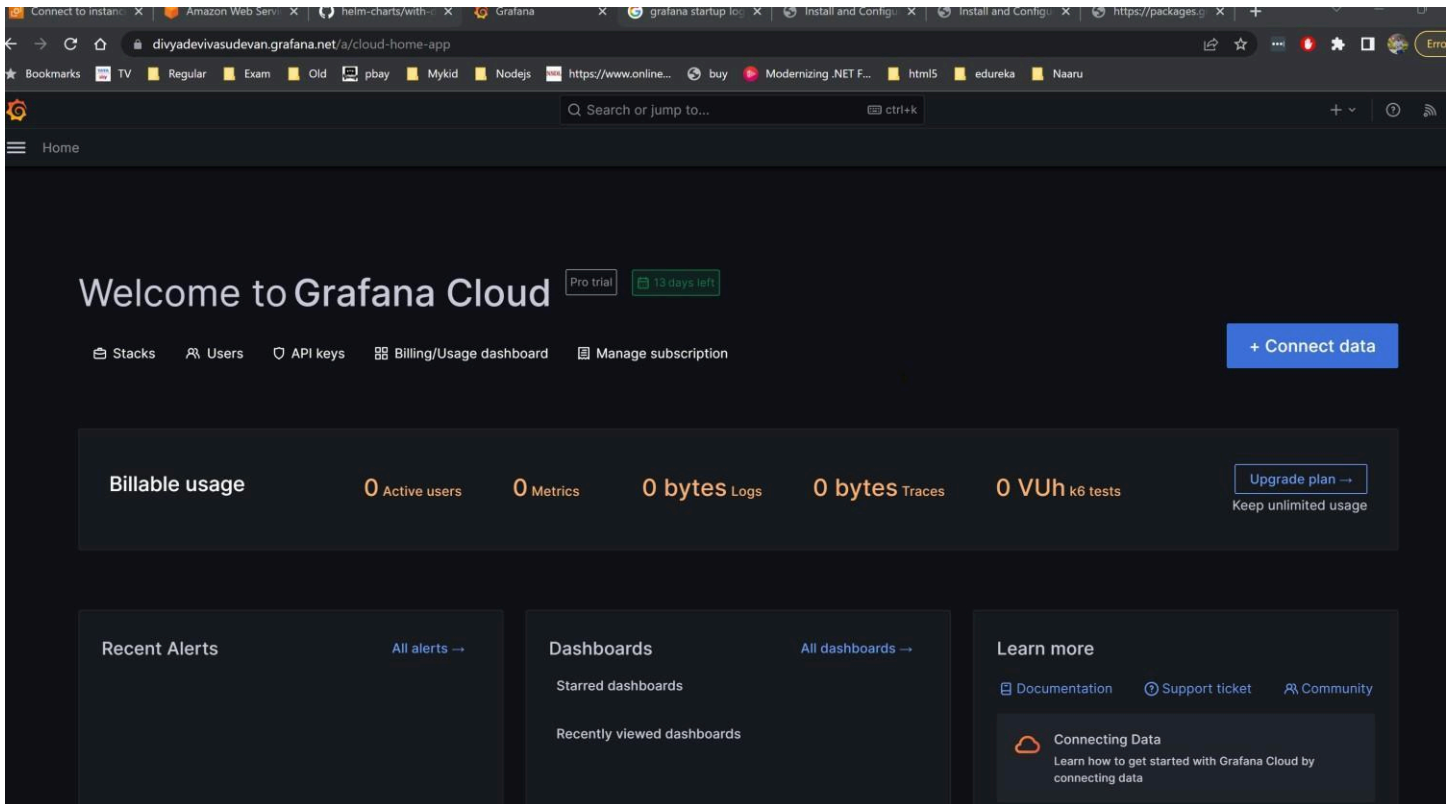
NAME	READY	AGE
statefulset.apps/prometheus-alertmanager	1/1	2m20s



Install Grafana

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

I used Grafana cloud instance.



Created a role for integration AWS cloud watch

Stack name

Stack name

GrafanaLabs-CloudWatch-Metrics

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

GrafanaLabs Account

AccountID

This is Grafana Lab's AWS account ID and is used to allow Grafana Cloud access to your AWS metrics.

arn:aws:iam::008923505280:root

ExternalID

This is your Grafana Cloud Identifier and is used for security purposes.

1009139

New Role

Role name

Customize the name of the IAM role used by GrafanaLabs for the CloudWatch Integration.

GrafanaLabsCloudWatchIntegration

Capabilities

The following resource(s) require capabilities: [AWS::IAM::Role]

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

☒ I acknowledge that AWS CloudFormation might create IAM resources with custom names.

CloudShell

Feedback

Language

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

2. Connect to AWS account

Once you have successfully created a new AWS IAM role, you can proceed with the installation by entering your account info below.

ARN

Paste the ARN from your AWS IAM role.

arn:aws:iam::869405457603:role/GrafanaLabsCloudWatchIntegration

AWS Regions

ap-south-1 ×

Test Connection

✓ All good! The account is working properly.

3. Create scrape job

Select which services you want scraped. Dashboards will be automatically installed for all supported services.

 New AWS services available to scrape!

Scrape job name

Monitoring

Services to scrape

AWS/ApplicationELB × ec2 ×

☒ Include your AWS resource tags on an `aws_<service-name>_info` metric (ex, `aws_ec2_info`)



Tags will appear as labels on the exported metric with a `tag_` prefix. Choosing to include tags will increase the total number of active series which can have an impact on your Grafana Cloud Costs. Additionally, please ensure your tags adhere to AWS best practices in that they do not contain personally identifiable information (PII) or other confidential or sensitive information.






Create scrape job

✓ All good! Scrape job created.

Import dashboard from Grafana.com

Integration - CloudWatch Metrics

Manage folder dashboards and permissions

 Dashboards  Panels  Alert rules  Permissions  Settings

Search for dashboards

New ▾

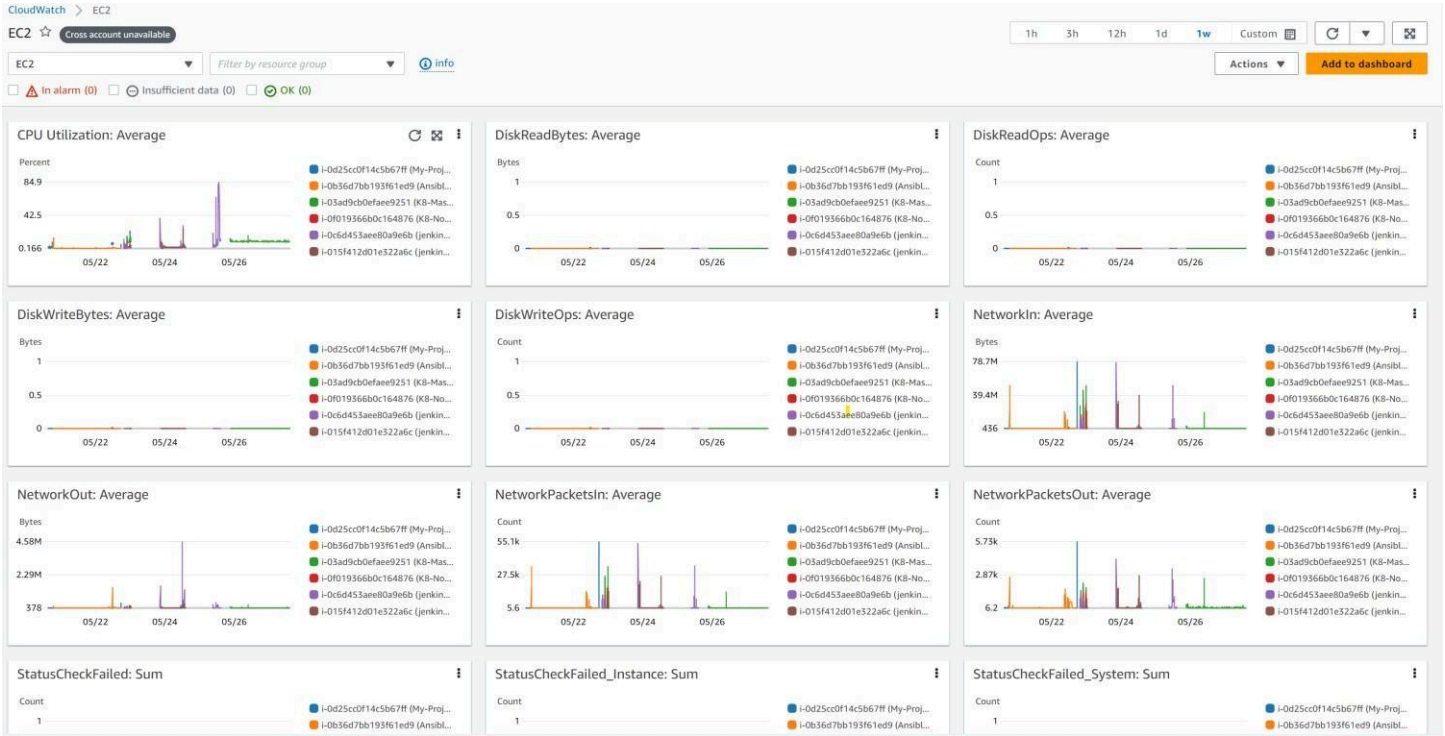
Filter by tag ▾

☐ Starred



Sort ▾

<input type="checkbox"/>	AWS EBS Integration - CloudWatch Metrics	cloudwatch-integration
<input type="checkbox"/>	AWS EC2 Integration - CloudWatch Metrics	cloudwatch-integration
<input type="checkbox"/>	AWS ECS Integration - CloudWatch Metrics	cloudwatch-integration
<input type="checkbox"/>	AWS Lambda Integration - CloudWatch Metrics	cloudwatch-integration
<input type="checkbox"/>	AWS RDS Integration - CloudWatch Metrics	cloudwatch-integration
<input type="checkbox"/>	AWS S3 Integration - CloudWatch Metrics	cloudwatch-integration



END of Project