

DevOps Project

Problem Statement:

Create an end-to-end CI/CD pipeline in AWS platform using Jenkins as the orchestration tool, GitHub as scm, maven as the build tool, deploy in a docker instance and create a docker image, store the docker image in ECR, Kubernetes deployment using ECR image. Build sample java web app using maven.

Approach:

Requirements:

- CI/CD declarative pipeline System
- Git - local version control system.
- GitHub - As Distributed version control system.
- Jenkins - Continuous Integration tool.
- Maven - As a Build Tool.
- docker -Containerization build tool using ECR
- Kubernetes - As Container Management Tool

Step-1:

- Setup CI/CD with GitHub, Jenkins, Maven & Tomcat.
- Setup Jenkins
- Setup & Configure Maven, Git.
- Setup Tomcat Server.
- Integrating GitHub, Maven, Tomcat Server with Jenkins
- Create a CI and CD Job.
- Test the Deployment

Step-2:

- Setup CI/CD with GitHub, Jenkins, Maven & Docker.
- Setting up the docker Environment.
- Create an Image and Container on Docker Host.
- Integrate Docker Host with Jenkins.
- Create CI/CD Job on Jenkins to build and deploy on container.

Step-3:

- Build and Deploy on Container.
- CI/CD with GitHub, Jenkins, Maven & Kubernetes.
- Setting up the Kubernetes (EKS).
- Write pod service and deployment manifest file.
- CI/CD Job to build code on Jenkins & Deploy it on Kubernetes.

Step-4:

- Deploy artifacts on the Kubernetes
- Write codes in the artifacts of docker and Kubernetes which we want to run.
- Now build the code in Jenkins.
- Check in Kubernetes the pods are getting created or not.
- Now copy the service IP and paste it in the browser and check the output.

Solution:

Creating an EC2 instance for Git operation.

```
Total  
Running transaction check  
Transaction check succeeded.  
Running transaction test  
Transaction test succeeded.  
Running transaction  
Preparing :  
Installing : git-core-2.50.1-1.amzn2023.0.1.x86_64 1/1  
Installing : git-core-doc-2.50.1-1.amzn2023.0.1.noarch 1/1  
Installing : perl-lib-0.65-477.amzn2023.0.7.x86_64 2/8  
Installing : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 3/8  
Installing : perl-File-Find-1.37-477.amzn2023.0.7.noarch 4/8  
Installing : perl-Error-1.0.17029-5.amzn2023.0.2.noarch 5/8  
Installing : perl-Git-2.50.1-1.amzn2023.0.1.noarch 6/8  
Installing : git-2.50.1-1.amzn2023.0.1.x86_64 7/8  
Running scriptlet: git-2.50.1-1.amzn2023.0.1.x86_64 8/8  
Verifying : git-2.50.1-1.amzn2023.0.1.x86_64 1/8  
Verifying : git-core-2.50.1-1.amzn2023.0.1.noarch 2/8  
Verifying : git-core-doc-2.50.1-1.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.amzn2023.0.7.noarch 5/8  
Verifying : perl-Git-2.50.1-1.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.amzn2023.0.7.x86_64 8/8  
Installed:  
git-2.50.1-1.amzn2023.0.1.x86_64 git-core-2.50.1-1.amzn2023.0.1.x86_64 git-core-doc-2.50.1-1.amzn2023.0.1.noarch  
perl-Error-1.0.17029-5.amzn2023.0.2.noarch perl-File-Find-1.37-477.amzn2023.0.7.noarch perl-Git-2.50.1-1.amzn2023.0.1.noarch  
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 perl-lib-0.65-477.amzn2023.0.7.x86_64  
Complete!  
[root@dev-server ~]# |
```

The screenshot shows a terminal window with a dark background and white text. It displays the output of a package transaction process. The terminal window has a title bar with the text "root@ip-172-31-4-217:~". At the top right, there are status indicators for network speed (49 MB/s), memory usage (7.9 MB), and time (00:00). Below the terminal area, there is a taskbar with various icons for file management, system tools, and communication. The date and time (17-10-2025) are also visible at the bottom right of the screen.

Code files are being cloned.

```
Installing : git-core-2.50.1-1.amzn2023.0.1.x86_64 1/8  
Installing : git-core-doc-2.50.1-1.amzn2023.0.1.noarch 2/8  
Installing : perl-lib-0.65-477.amzn2023.0.7.x86_64 3/8  
Installing : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 4/8  
Installing : perl-File-Find-1.37-477.amzn2023.0.7.noarch 5/8  
Installing : perl-Error-1.0.17029-5.amzn2023.0.2.noarch 6/8  
Installing : perl-Git-2.50.1-1.amzn2023.0.1.noarch 7/8  
Installing : git-2.50.1-1.amzn2023.0.1.x86_64 8/8  
Running scriptlet: git-2.50.1-1.amzn2023.0.1.x86_64 8/8  
Verifying : git-2.50.1-1.amzn2023.0.1.x86_64 1/8  
Verifying : git-core-2.50.1-1.amzn2023.0.1.noarch 2/8  
Verifying : git-core-doc-2.50.1-1.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.amzn2023.0.7.noarch 5/8  
Verifying : perl-Git-2.50.1-1.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.amzn2023.0.7.x86_64 8/8  
Installed:  
git-2.50.1-1.amzn2023.0.1.x86_64 git-core-2.50.1-1.amzn2023.0.1.x86_64 git-core-doc-2.50.1-1.amzn2023.0.1.noarch  
perl-Error-1.0.17029-5.amzn2023.0.2.noarch perl-File-Find-1.37-477.amzn2023.0.7.noarch perl-Git-2.50.1-1.amzn2023.0.1.noarch  
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 perl-lib-0.65-477.amzn2023.0.7.x86_64  
Complete!  
[root@dev-server ~]# git clone https://github.com/sanjayguruji/shourjo-10743365.git  
Cloning into 'shourjo-10743365'...  
remote: Enumerating objects: 84, done.  
remote: Counting objects: 100% (84/84), done.  
remote: Compressing objects: 100% (52/52), done.  
remote: Total 84 (delta 17), reused 29 (delta 1), pack-reused 0 (from 0)  
Receiving objects: 100% (84/84), 15.96 KiB | 3.99 MiB/s, done.  
Resolving deltas: 100% (17/17), done.  
[root@dev-server ~]# |
```

This screenshot shows a terminal window with a dark background and white text. It displays the output of cloning a GitHub repository named "shourjo-10743365". The terminal window has a title bar with the text "root@ip-172-31-4-217:~". At the top right, there are status indicators for network speed (16.10 MB/s), memory usage (ENG IN), and time (17-10-2025). Below the terminal area, there is a taskbar with various icons for file management, system tools, and communication.

We can see that the files are being cloned.

The terminal window shows the following command sequence:

```
[root@dev-server .ssh]# cd
[root@dev-server ~]# ll
total 0
drwxr-xr-x. 5 root root 115 Oct 17 10:40 shourjo-10743365
[root@dev-server ~]# cd shourjo-10743365
[root@dev-server shourjo-10743365]# ll
total 20
-rw-r--r--. 1 root root 143 Oct 17 10:40 Dockerfile
-rw-r--r--. 1 root root 3533 Oct 17 10:40 Jenkinsfile
-rw-r--r--. 1 root root 33 Oct 17 10:40 README.md
-rw-r--r--. 1 root root 6333 Oct 17 10:40 pom.xml
drwxr-xr-x. 3 root root 32 Oct 17 10:40 server
drwxr-xr-x. 3 root root 32 Oct 17 10:40 webapp
[root@dev-server shourjo-10743365]# git add .
[root@dev-server shourjo-10743365]# git status
On branch main
nothing to commit, working tree clean
[root@dev-server shourjo-10743365]# git push origin main --force
The authenticity of host 'github.com (140.82.112.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3vvvV6TuJJhbZisF/zLDA0zPMsvHdkr4UvC0qu.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enumerating objects: 84, done.
Counting objects: 100% (84/84), done.
Delta compression using up to 2 threads
Compressing objects: 100% (36/36), done.
Writing objects: 100% (84/84) 15.96 KiB | 7.98 MiB/s, done.
Total 84 (delta 17), reused 84 (delta 17), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (17/17), done.
To github.com:arpitkumar777/project.git
 * [new branch]      main    -> main
[root@dev-server shourjo-10743365]#
```

The browser window shows the GitHub project page for `arpitkumar777/project`. The code tab displays the following file structure and recent commits:

- `main` branch
- 1 Branch
- 0 Tags
- Go to file
- Code
- Issues
- Pull requests
- Actions
- Projects
- Wiki
- Security
- Insights
- Settings

Recent commits:

Author	File	Commit Message	Date
sanjayguruji	index.jsp	Update index.jsp	fd3ed44 · 2 months ago
server		fin-com	last year
webapp		Update index.jsp	2 months ago
Dockerfile		fin-com	last year
Jenkinsfile		fin-com	last year
README.md		Update README.md	last year
pom.xml		fin-com	last year

About:

No description, website, or topics provided.

Readme

Activity

0 stars

0 watching

0 forks

Releases:

No releases published

Create a new release

Creating a new EC2 instance for Jenkins setup.

```
root@ip-172-31-91-18:~ x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10844621> cd downloads
PS C:\Users\10844621\downloads> ssh -i "project.pem" ec2-user@ec2-44-211-78-216.compute-1.amazonaws.com
The authenticity of host 'ec2-44-211-78-216.compute-1.amazonaws.com (44.211.78.216)' can't be established.
ED25519 key fingerprint is SHA256:u3u/UJu05WcHtaKLr/9YJrS+3An1MzxgEMhYxsp0BJ4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-211-78-216.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

'   #
` \_ #####_      Amazon Linux 2023
` \_ #####`_
` ` \###]
` ` \#/     https://aws.amazon.com/linux/amazon-linux-2023
` ` \#/_`_>
` ` \/_`_`/
` ` \/_`_`/`/
` ` \/_`_`/`/`/
` ` \/_`_`/`/`/`/
` ` \/_`_`/`/`/`/`/
` ` \/_`_`/`/`/`/`/`/
` ` \/_`_`/`/`/`/`/`/`/
[ec2-user@ip-172-31-91-18 ~]$ sudo su -
[root@ip-172-31-91-18 ~]# hostnamectl set-hostname jenkins.example.com
[root@ip-172-31-91-18 ~]# bash
[root@jenkins ~]# |
```

The screenshot shows a Windows PowerShell window titled 'root@ip-172-31-91-18:~'. It displays the process of connecting via SSH to an Amazon Linux 2023 instance. The user runs 'ssh -i "project.pem" ec2-user@ec2-44-211-78-216.compute-1.amazonaws.com'. After confirming the host fingerprint, the session enters a root shell. The user then runs 'hostnamectl set-hostname jenkins.example.com' and 'bash' to switch to a standard user shell. The taskbar at the bottom shows various application icons, and the system tray indicates 'ENG IN' and the date '17-10-2025'.

Jenkins script file for Jenkins setup.

```
root@jenkins:~ x + v
#!/bin/bash
sudo yum update -y
sudo wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
sudo yum upgrade
sudo yum install java-21-amazon-corretto -y
sudo yum install maven -y
sudo yum install git -y
sudo yum install jenkins -y
sudo systemctl start jenkins
sudo systemctl enable jenkins
sudo systemctl status jenkins|
```

The screenshot shows a Windows terminal window titled 'root@jenkins:~'. It displays a Jenkins setup script being run. The script installs Java, Maven, and Git, and starts the Jenkins service. The command 'systemctl status jenkins' is shown at the end. The taskbar at the bottom shows various application icons, and the system tray indicates 'ENG IN' and the date '18-10-2025'.

Jenkins setup is being done.

The image shows two screenshots of a web browser window. The top screenshot displays the 'Unlock Jenkins' setup page, which includes instructions for finding the initial admin password from a log file and a text input field for entering it. The bottom screenshot shows the Jenkins dashboard, featuring sections for building software projects and setting up distributed builds.

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

Continue

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job +

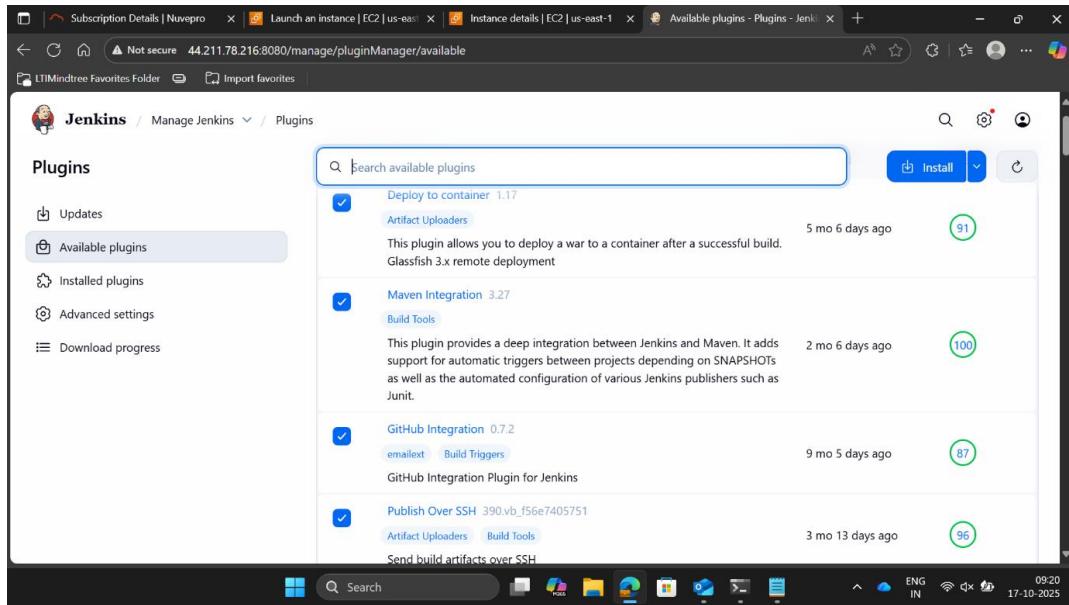
Set up a distributed build

Set up an agent

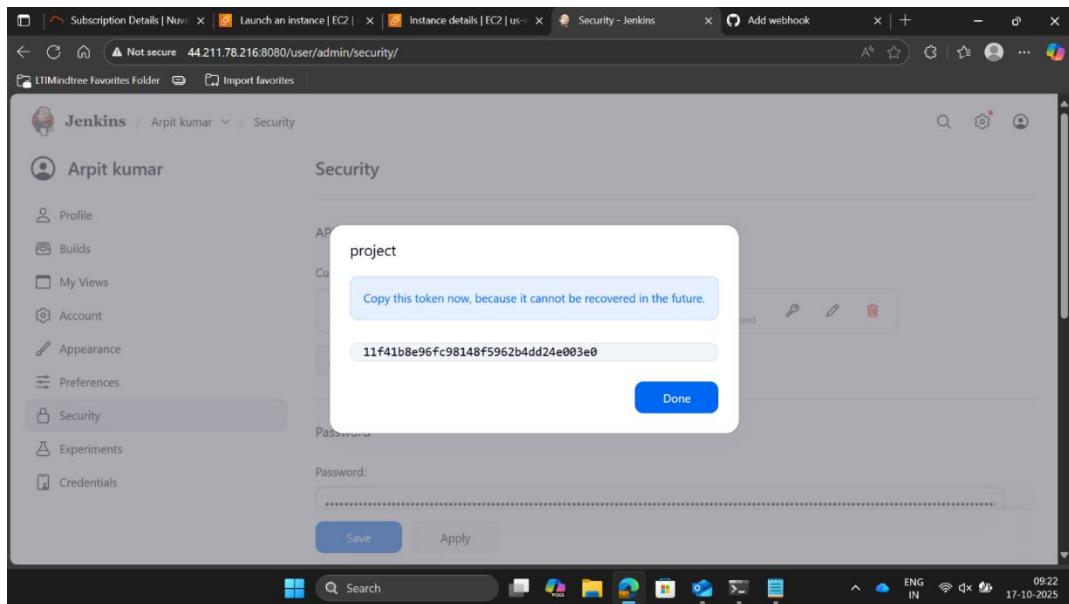
Configure a cloud

Learn more about distributed builds ?

Installing the required plugins in Jenkins.



Creating security key in Jenkins



Adding webhook in GitHub.

The screenshot shows the 'Add webhook' page in GitHub. The URL in the address bar is <https://github.com/arpitkumar777/project/settings/hooks/new>. The left sidebar shows the 'Webhooks' section is selected. The main form fields are:

- Payload URL ***: http://44.211.78.216:8080/github-webhook/
- Content type ***: application/json
- Secret**: 11f41b8e96fc98148f5962b4dd24e003e0

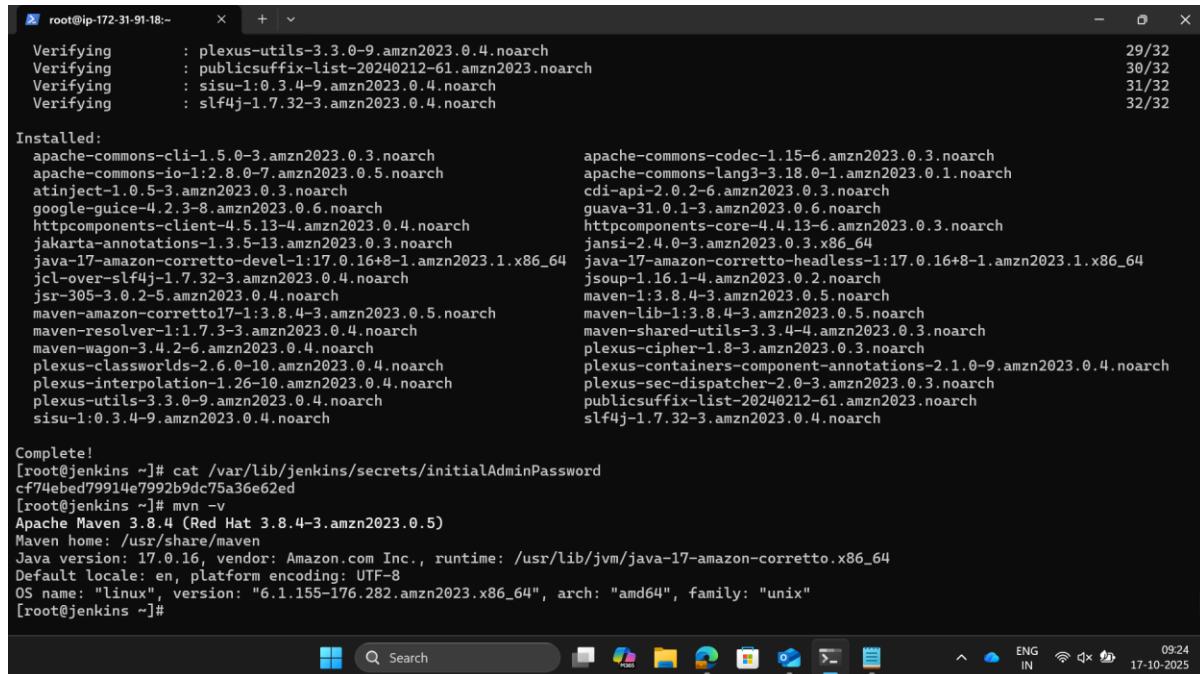
At the bottom, there is a note: "By default, we verify SSL certificates when delivering payloads."

Webhook is successfully being added.

The screenshot shows the 'Webhooks' settings page in GitHub. The URL in the address bar is <https://github.com/arpitkumar777/project/settings/hooks>. The left sidebar shows the 'Webhooks' section is selected. The main page displays the added webhook:

- URL**: http://44.211.78.216:8080/github-w... (push)
- Status**: Last delivery was successful.
- Actions**: Edit | Delete

Getting java and maven path in java Jenkins terminal.

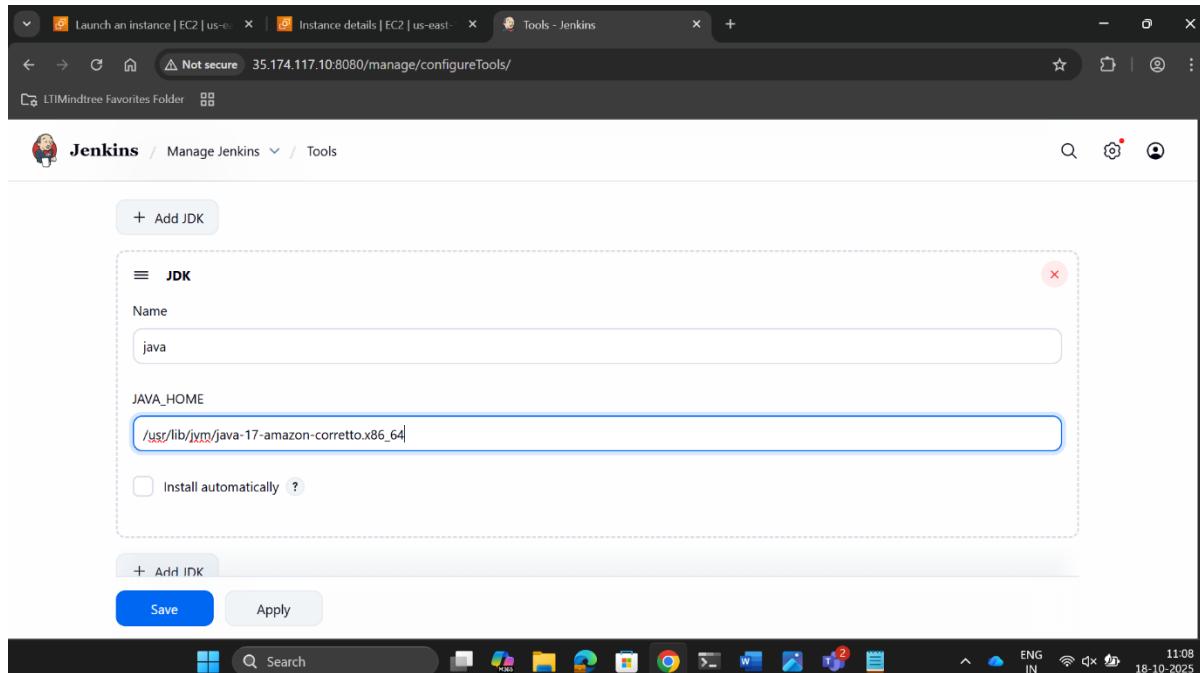


```
root@ip-172-31-91-18:~ Verifying : plexus-utils-3.3.0-9.amzn2023.0.4.noarch 29/32
root@ip-172-31-91-18:~ Verifying : publicsuffix-list-20240212-61.amzn2023.noarch 30/32
root@ip-172-31-91-18:~ Verifying : sisu-1.0.3.4-9.amzn2023.0.4.noarch 31/32
root@ip-172-31-91-18:~ Verifying : slf4j-1.7.32-3.amzn2023.0.4.noarch 32/32

Installed:
  apache-commons-cli-1.5.0-3.amzn2023.0.3.noarch
  apache-commons-io-1.2.8.0-7.amzn2023.0.5.noarch
  atinject-1.0.5-3.amzn2023.0.3.noarch
  google-guice-4.2.3-8.amzn2023.0.6.noarch
  httpcomponents-client-4.5.13-4.amzn2023.0.4.noarch
  jakarta-annotations-1.3.5-13.amzn2023.0.3.noarch
  java-17-amazon-corretto-devel-1:17.0.16+8-1.amzn2023.1.x86_64
  jcl-over-slf4j-1.7.32-3.amzn2023.0.4.noarch
  jsr-305-3.0.2-5.amzn2023.0.4.noarch
  maven-amazon-corretto17-1:3.8.4-3.amzn2023.0.5.noarch
  maven-resolver-1:1.7.3-3.amzn2023.0.4.noarch
  maven-wagon-3.4.2-6.amzn2023.0.4.noarch
  plexus-classworlds-2.6.0-10.amzn2023.0.4.noarch
  plexus-interpolation-1.26-10.amzn2023.0.4.noarch
  plexus-utils-3.3.0-9.amzn2023.0.4.noarch
  sisu-1.0.3.4-9.amzn2023.0.4.noarch

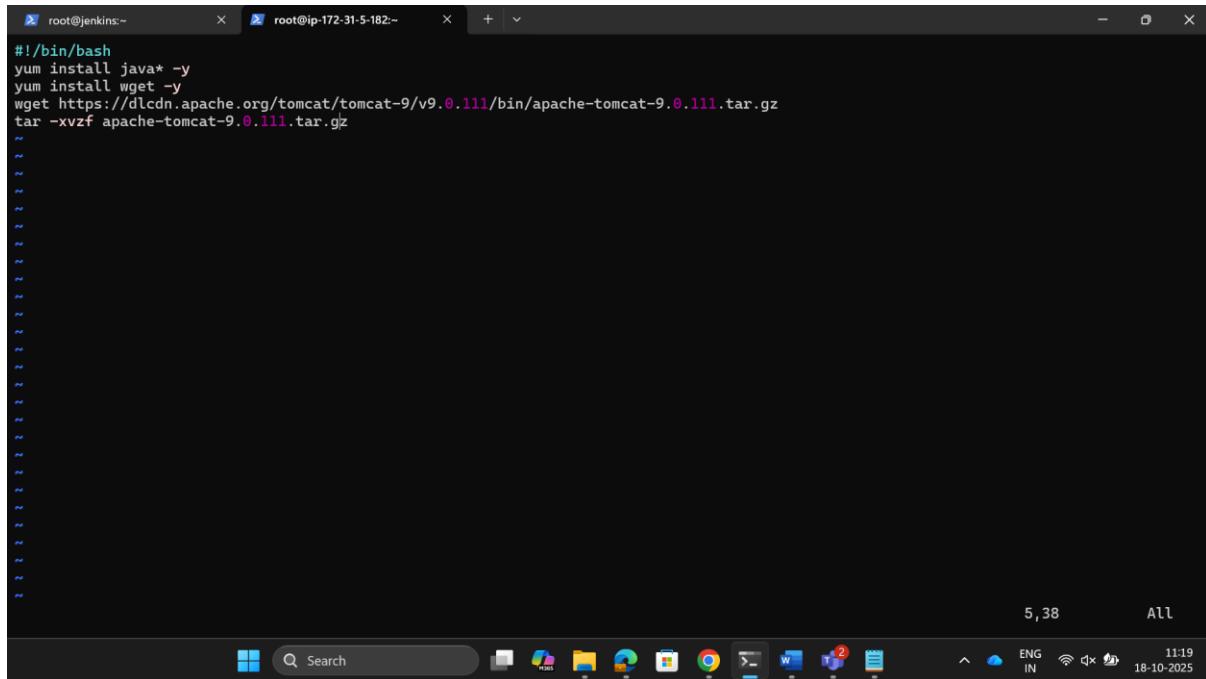
Complete!
[root@jenkins ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
c7f4ebcd79914e7992b9dc75a36e62ed
[root@jenkins ~]# mvn -v
Apache Maven 3.8.4 (Red Hat 3.8.4-3.amzn2023.0.5)
Maven home: /usr/share/maven
Java version: 17.0.16, vendor: Amazon.com Inc., runtime: /usr/lib/jvm/java-17-amazon-corretto.x86_64
Default locale: en, platform encoding: UTF-8
OS name: "Linux", version: "6.1.155-176.282.amzn2023.x86_64", arch: "amd64", family: "unix"
[root@jenkins ~]#
```

Adding java and maven path to Jenkins tool.



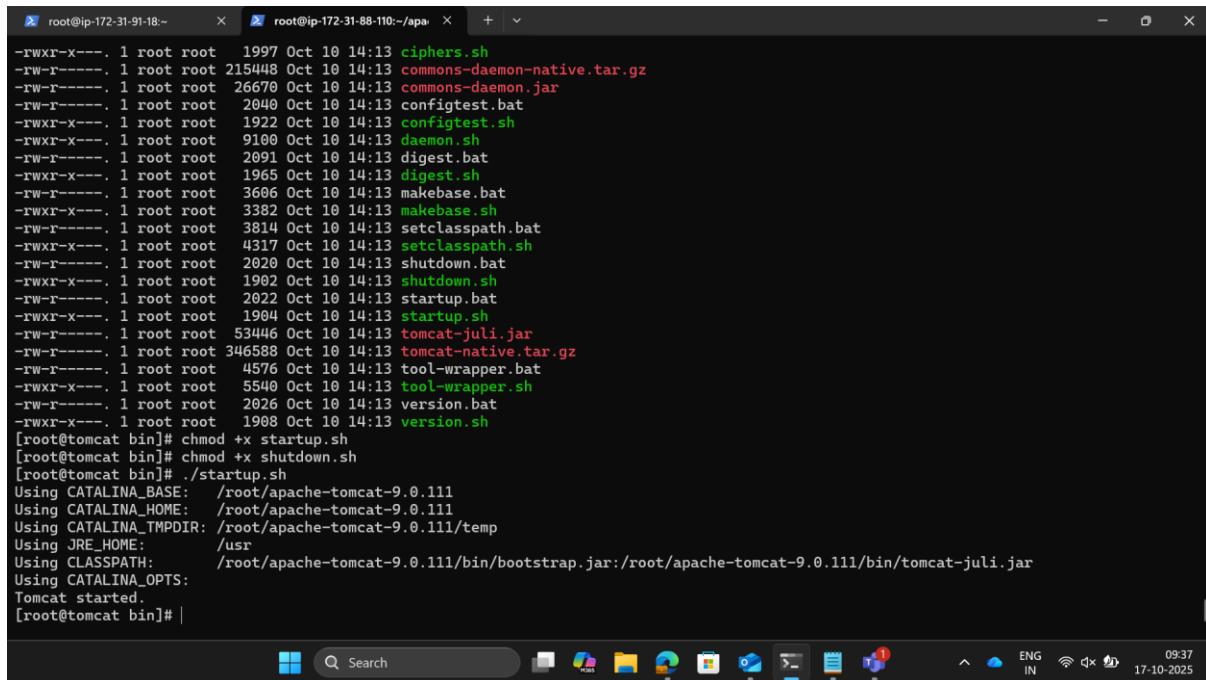
The screenshot shows the Jenkins configuration interface for adding a new JDK. A new entry for 'java' is being added, with the JAVA_HOME path set to '/usr/lib/jvm/java-17-amazon-corretto.x86_64'. The 'Install automatically' checkbox is unchecked. At the bottom, there are 'Save' and 'Apply' buttons.

Creating new EC2 instance and setting up the tomcat server.



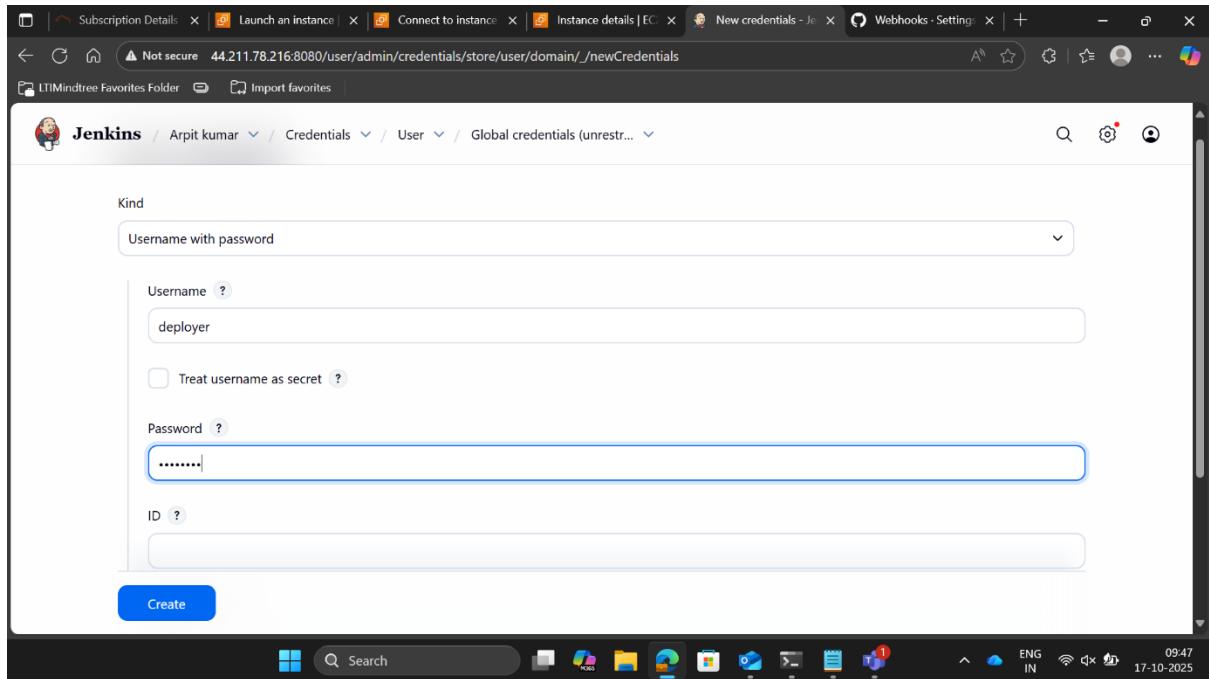
```
#!/bin/bash
yum install java* -y
yum install wget -y
wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.111/bin/apache-tomcat-9.0.111.tar.gz
tar -xvzf apache-tomcat-9.0.111.tar.gz
```

Giving permissions for starting up and shutdown of tomcat server.

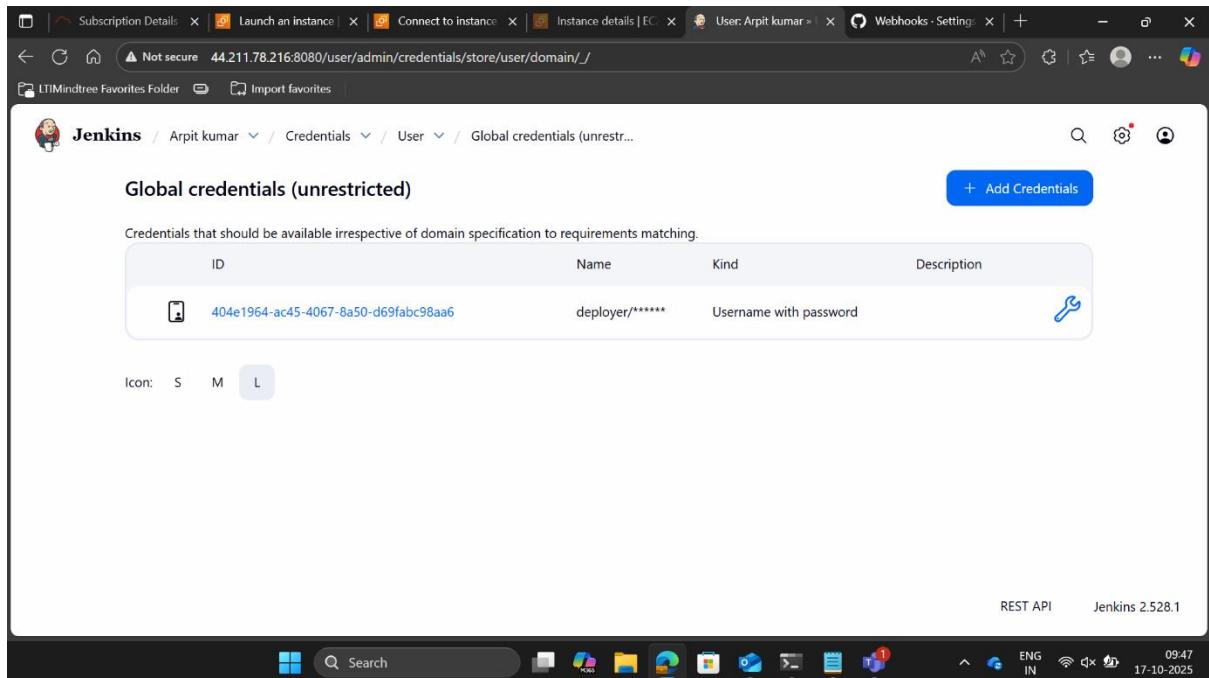


```
-rwxr-x---. 1 root root 1997 Oct 10 14:13 ciphers.sh
-rw-r-----. 1 root root 215448 Oct 10 14:13 commons-daemon-native.tar.gz
-rw-r-----. 1 root root 26670 Oct 10 14:13 commons-daemon.jar
-rw-r-----. 1 root root 2040 Oct 10 14:13 configtest.bat
-rwxr-x---. 1 root root 1922 Oct 10 14:13 configtest.sh
-rwxr-x---. 1 root root 9100 Oct 10 14:13 daemon.sh
-rw-r-----. 1 root root 2091 Oct 10 14:13 digest.bat
-rwxr-x---. 1 root root 1965 Oct 10 14:13 digest.sh
-rw-r-----. 1 root root 3606 Oct 10 14:13 makebase.bat
-rwxr-x---. 1 root root 3382 Oct 10 14:13 makebase.sh
-rw-r-----. 1 root root 3814 Oct 10 14:13 setclasspath.bat
-rwxr-x---. 1 root root 4317 Oct 10 14:13 setclasspath.sh
-rw-r-----. 1 root root 2020 Oct 10 14:13 shutdown.bat
-rwxr-x---. 1 root root 1902 Oct 10 14:13 shutdown.sh
-rw-r-----. 1 root root 2022 Oct 10 14:13 startup.bat
-rwxr-x---. 1 root root 1904 Oct 10 14:13 startup.sh
-rw-r-----. 1 root root 53446 Oct 10 14:13 tomcat-juli.jar
-rw-r-----. 1 root root 346588 Oct 10 14:13 tomcat-native.tar.gz
-rw-r-----. 1 root root 4576 Oct 10 14:13 tool-wrapper.bat
-rwxr-x---. 1 root root 5540 Oct 10 14:13 tool-wrapper.sh
-rw-r-----. 1 root root 2026 Oct 10 14:13 version.bat
-rwxr-x---. 1 root root 1908 Oct 10 14:13 version.sh
[root@tomcat bin]# chmod +x startup.sh
[root@tomcat bin]# chmod +x shutdown.sh
[root@tomcat bin]# ./startup.sh
Using CATALINA_BASE: /root/apache-tomcat-9.0.111
Using CATALINA_HOME: /root/apache-tomcat-9.0.111
Using CATALINA_TMPDIR: /root/apache-tomcat-9.0.111/temp
Using JRE_HOME: /usr
Using CLASSPATH: /root/apache-tomcat-9.0.111/bin/bootstrap.jar:/root/apache-tomcat-9.0.111/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@tomcat bin]# |
```

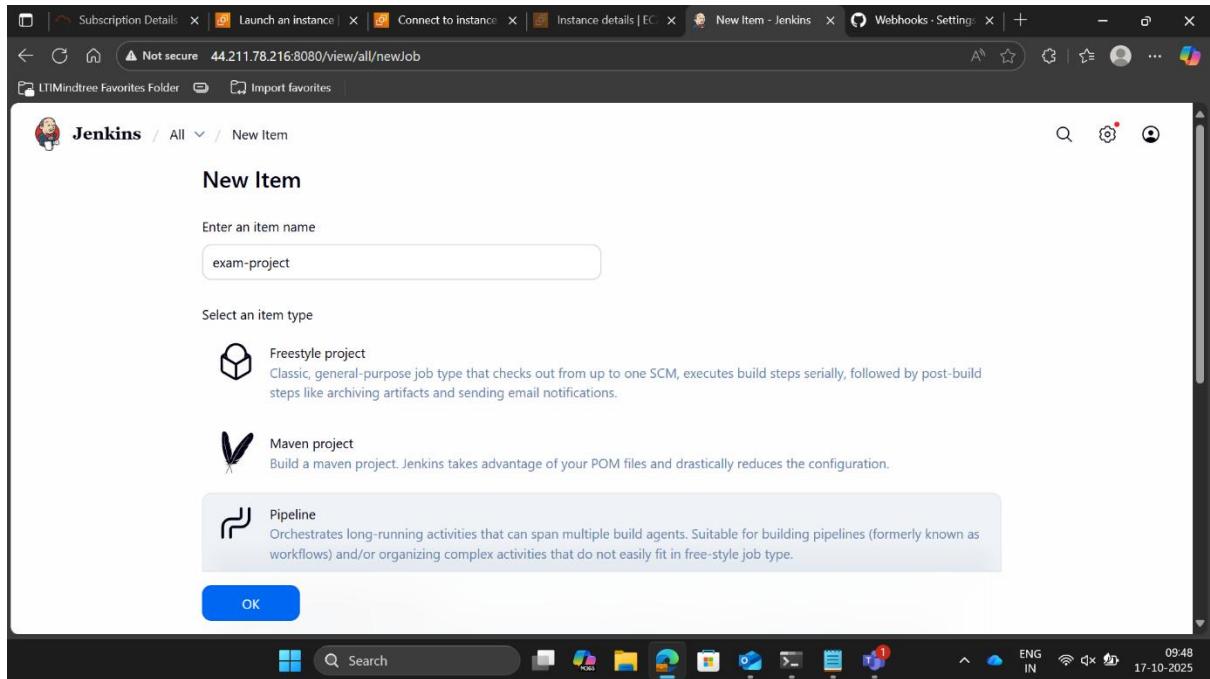
Setting up the global credentials in Jenkins.



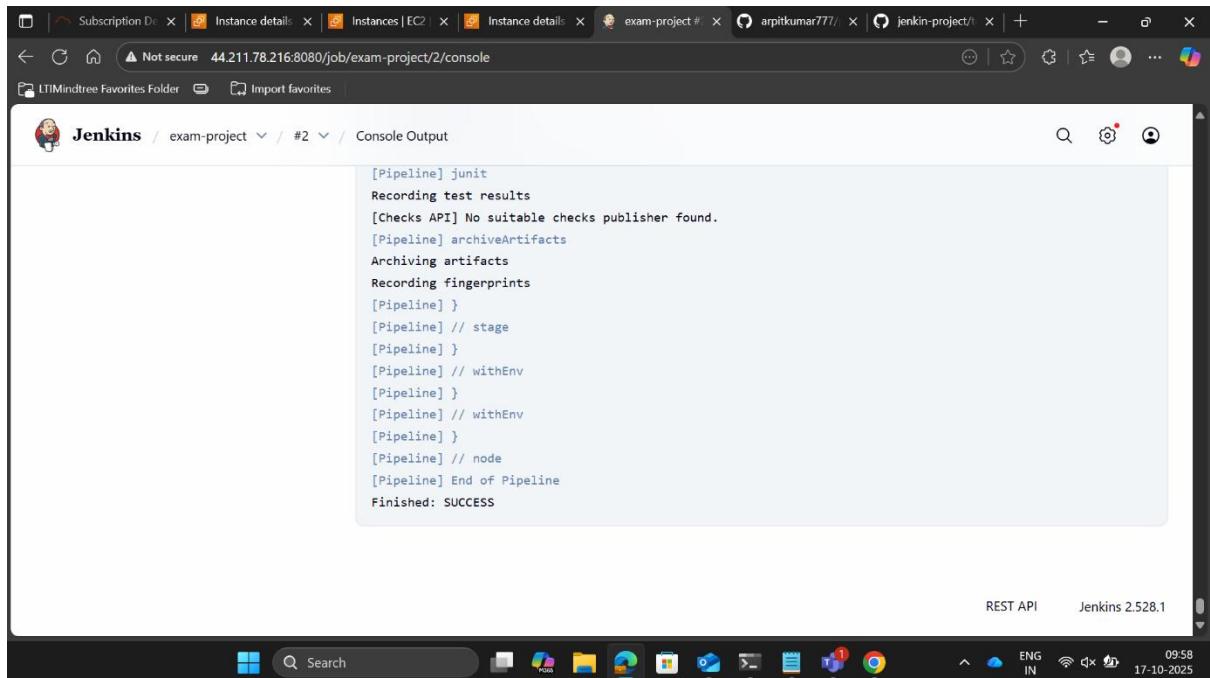
Credentials are being set up in Jenkins.



Now we are creating a pipeline project.



We can see that our build is being successfully finished till here.



Creating EC2 instance for docker installation.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10844621> cd downloads
PS C:\Users\10844621\downloads> ssh -i "project.pem" ec2-user@ec2-18-206-162-127.compute-1.amazonaws.com
The authenticity of host 'ec2-18-206-162-127.compute-1.amazonaws.com (18.206.162.127)' can't be established.
ED25519 key fingerprint is SHA256:NCq4nux9pmKV5AhTG4pMSPpSfxkEKguWBHD8QynEU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-206-162-127.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

          _#_
         /###\
        /####\
       /##|
      /#/ __ https://aws.amazon.com/linux/amazon-linux-2023
     /# \_-->
    /# / /
   /# / /
  /# / /
 [ec2-user@ip-172-31-81-114 ~]$ sudo su -
[root@ip-172-31-81-114 ~]# hostnamectl set-hostname docker.example.com
[root@ip-172-31-81-114 ~]# bash
[root@docker ~]# yum install docker
```

Giving password authentication permissions.

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

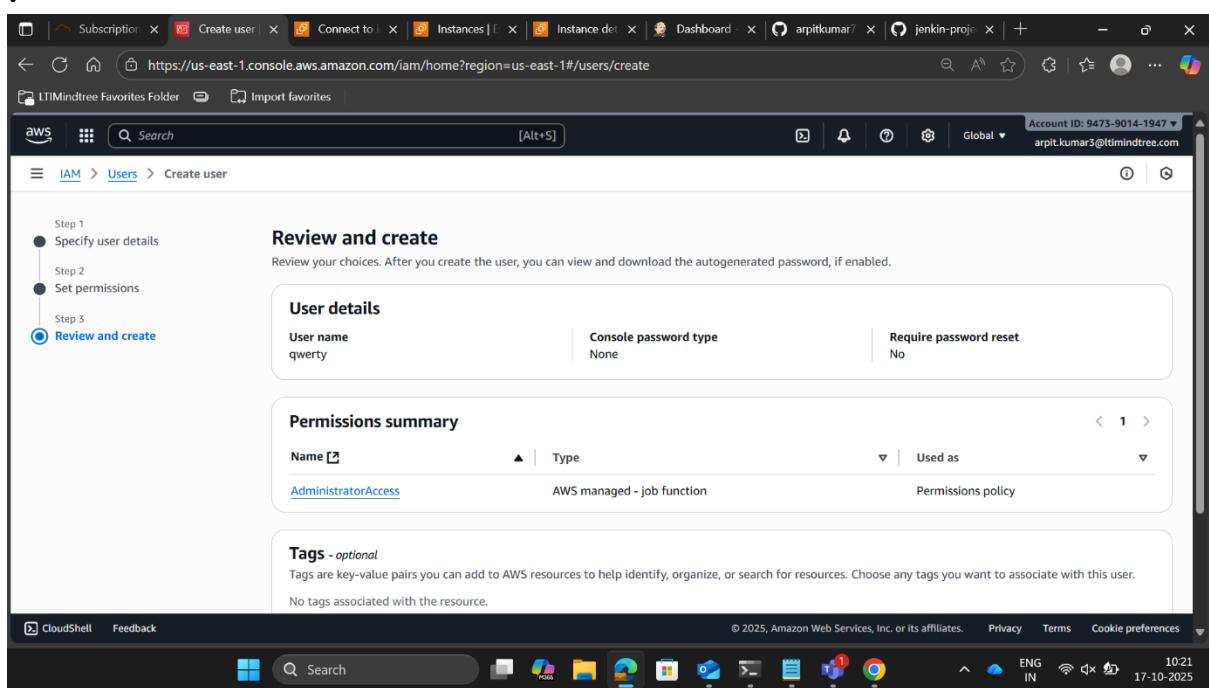
#AuthorizedPrincipalsFile none

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

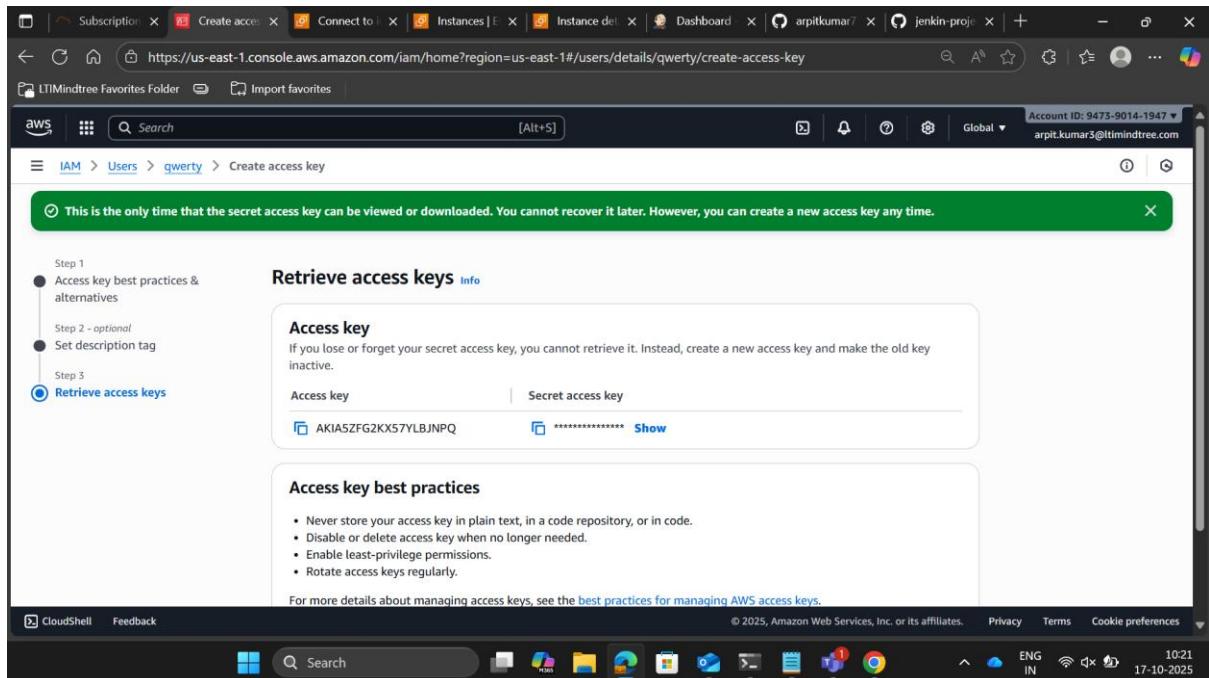
# Explicitly disable PasswordAuthentication. By presetting it, we
# avoid the cloud-init set_passwords module modifying sshd_config and
# restarting sshd in the default instance launch configuration.
PasswordAuthentication yes
PermitEmptyPasswords no

# Change to no to disable s/key passwords
-- INSERT --
```

Creating the user and giving administrator full access permission



Generating access key for aws configuration.



Creating AWS ECR

```
root@ip-172-31-91-18:~ root@ip-172-31-88-110:~/apac root@ip-172-31-81-114:~ + - X
valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
link/ether 02:42:e4:4a:82:d4 brd ff:ff:ff:ff:ff:ff
inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
    valid_lft forever preferred_lft forever
[root@docker ~]# vim /etc/hosts
[root@docker ~]# aws configure
AWS Access Key ID [None]: AKIA5ZFG2KX57YLBJNPQ
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: table
[root@docker ~]# aws ecr create-repository --repository-name tomcat-webapp
|-----+
| CreateRepository
|-----+
|-----+-----+
| repository
|-----+
|-----+-----+
| createdAt | 2025-10-17T04:53:26.779000+00:00
| imageTagMutability | MUTABLE
| registryId | 947390141947
| repositoryArn | arn:aws:ecr:us-east-1:947390141947:repository/tomcat-webapp
| repositoryName | tomcat-webapp
| repositoryUri | 947390141947.dkr.ecr.us-east-1.amazonaws.com/tomcat-webapp
|-----+-----+
|-----+-----+
| encryptionConfiguration
|-----+
| encryptionType | AES256
|-----+
|-----+-----+
| imageScanningConfiguration
|-----+
| scanOnPush | False
|-----+
[root@docker ~]#
```

Creating image inside docker.

The screenshot shows the Jenkins Pipeline configuration for the 'exam-project' job. The 'Pipeline' section is selected in the left sidebar. The main area contains a Groovy script editor with the following code:

```
42 configName: 'docker', // Must match SSH config name in Jenkins
43 transfers: [
44     sshTransfer(
45         sourceFiles: '**/*',
46         removePrefix: '',
47         remoteDirectory: 'lti',
48         execCommand: ''
49             cd lti
50             aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-
51             docker build -t tomcat-webapp .
52             docker tag tomcat-webapp:latest 947390141947.dkr.ecr.us-east-1.amazonaws.com/tomcat-
53             docker push 947390141947.dkr.ecr.us-east-1.amazonaws.com/tomcat-webapp:lates
54
55     )
56 ]
```

Below the script editor, there is a checkbox labeled "Use Groovy Sandbox". At the bottom are "Save" and "Apply" buttons. The status bar at the bottom right shows the date and time: 17-10-2025 10:30.

Our project is successful build for docker image .

The screenshot displays two windows. The top window is a Jenkins Pipeline view for the 'exam-project' job. It shows a 'Stage View' with six stages: 'Declarative: Tool Install' (190ms), 'Clone Repository' (730ms), 'Build with Maven' (4s), 'Deploy to Tomcat' (473ms), 'Deploy to Docker Host' (27s), and 'Declarative: Post Actions' (223ms). The 'Build with Maven' stage is highlighted. Below the stage view is a 'Latest Test Result (no failures)' section. The bottom window is a terminal session on a Windows 10 desktop. The user is navigating through a directory structure within a Docker container, specifically moving between 'image' and 'image' directories and listing files. The terminal output shows standard Linux file listing commands like 'cd', 'll', and 'ls'.

Stage	Declarative: Tool Install	Clone Repository	Build with Maven	Deploy to Tomcat	Deploy to Docker Host	Declarative: Post Actions
Oct 17 10:31 No Changes	190ms	730ms	4s	473ms	27s	223ms

```
bash: cd: image: No such file or directory
[root@docker image]# cd lti
bash: cd: lti: No such file or directory
[root@docker image]# ll
total 0
[root@docker image]# cd
[root@docker ~]# cd /root/image
bash: cd: /root/image: No such file or directory
[root@docker ~]# cd /image
[root@docker image]# ll
total 0
[root@docker image]# cd image
bash: cd: image: No such file or directory
[root@docker image]# cd\
> '^'
[root@docker image]# cd
[root@docker ~]# cd image
bash: cd: image: No such file or directory
[root@docker ~]# ll
total 0
drwxr-xr-x. 4 root root 103 Oct 17 05:08 image
drwxr-xr-x. 4 root root 103 Oct 17 05:01 lti
drwxr-xr-x. 3 root root 19 Oct 17 05:05 root
[root@docker ~]# cd image
[root@docker image]# ll
total 20
-rw-r--r--. 1 root root 143 Oct 17 05:08 Dockerfile
-rw-r--r--. 1 root root 3533 Oct 17 05:08 Jenkinsfile
-rw-r--r--. 1 root root 33 Oct 17 05:08 README.md
-rw-r--r--. 1 root root 6333 Oct 17 05:08 pom.xml
drwxr-xr-x. 4 root root 46 Oct 17 05:08 server
drwxr-xr-x. 4 root root 46 Oct 17 05:08 webapp
```

We can see the image in the amazon ECR.

The screenshot shows the AWS Lambda console with the following details:

- Function Name:** HelloWorld
- Region:** us-east-1
- Last Triggered:** 17 minutes ago
- Execution Duration:** 10ms
- Memory Size:** 128 MB
- Log Stream:** /aws/lambda/HelloWorld-1d1f33c3
- Logs:** View logs (1 log entry)

The log entry is:

```
START RequestId: 1d1f33c3-0000-4000-8000-000000000000 Version: $LATEST
```

Creating EC2 instance for eks cluster

```
PS C:\Users\10844621> cd downloads
PS C:\Users\10844621\downloads> ssh -i "project.pem" ec2-user@ec2-3-86-195-228.compute-1.amazonaws.com
The authenticity of host 'ec2-3-86-195-228.compute-1.amazonaws.com (3.86.195.228)' can't be established.
ED25519 key fingerprint is SHA256:UUXUYcsmysgZD67U89Xhttp87CZUSdi707A7SAvy17P8.
This key is not known by any other name.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-86-195-228.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
~\_ #####      Amazon Linux 2023
~~ \#####\_
~~ \###|_
~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~` '->
~~ .-' /
~~ /_/
~/m/'
```

[ec2-user@ip-172-31-80-80 ~]\$ sudo su -
[root@ip-172-31-80-80 ~]# hostnamectl set-hostname eks-cluster.example.com
[root@ip-172-31-80-80 ~]# bash
[root@eks-cluster ~]#

Assigning role to the instance.

The screenshot shows the AWS EC2 Instances page. In the main table, there is a row for an instance named 'eks-cluster' with the ID 'i-04e7628295e8efc9f'. A context menu is open over this row, with the option 'Modify IAM role' highlighted. The table columns include Name, Instance ID, Instance state, Instance type, and Status check. The instance state for 'eks-cluster' is 'Running' and the instance type is 't3.small'. The status check shows 'Initializing'. To the right of the table, there is a vertical sidebar with various EC2-related options like 'Instance diagnostics', 'Instance settings', 'Networking', 'Security', etc. Below the table, a detailed view for the 'eks-cluster' instance is shown, including tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Details' tab is selected. At the bottom of the page, there is a standard Windows taskbar with icons for File Explorer, Task View, and other system tools.

AWS configuration of EKS instance.

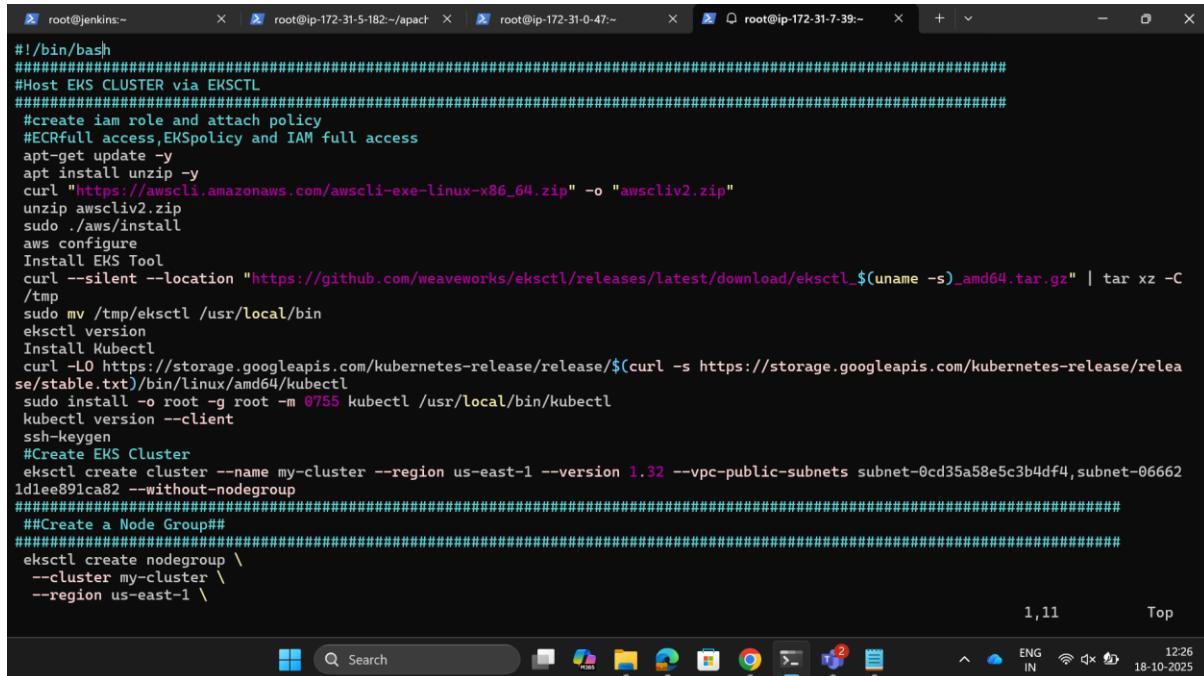
The screenshot shows a Windows PowerShell session running on an EKS instance. The session starts with the command 'cd downloads' followed by 'ssh -i "project.pem" ec2-user@ec2-3-86-195-228.compute-1.amazonaws.com'. It then prompts for a key fingerprint and adds it to the known hosts. The user then runs 'aws configure' to set up AWS credentials. The session ends with the command 'sudo su -'. The PowerShell window has a dark theme and includes a taskbar at the bottom with icons for File Explorer, Task View, and other system tools. The date and time at the bottom right of the screen are 17-10-2025 10:44.

```
PS C:\Users\10844621> cd downloads
PS C:\Users\10844621\downloads> ssh -i "project.pem" ec2-user@ec2-3-86-195-228.compute-1.amazonaws.com
The authenticity of host 'ec2-3-86-195-228.compute-1.amazonaws.com (3.86.195.228)' can't be established.
ED25519 key fingerprint is SHA256:UUXUYcsmysgZD67U89Xhtp87CZUSdi707A7SAvy17P8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-86-195-228.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_          Amazon Linux 2023
\_###_
\###|
 \#/ _--> https://aws.amazon.com/linux/amazon-linux-2023
 /_
 /_/
 /m/'

[ec2-user@ip-172-31-80-80 ~]$ sudo su -
[root@ip-172-31-80-80 ~]# hostnamectl set-hostname eks-cluster.example.com
[root@ip-172-31-80-80 ~]# bash
[root@eks-cluster ~]# aws configure
AWS Access Key ID [None]: AKIA5ZFG2KX57YLBJNPQ
AWS Secret Access Key [None]: NE3ZjSNqNT8gp0Xu4azwnqaKfd8Hera+L9j5LyYD
Default region name [None]: us-east-1
Default output format [None]: table
[root@eks-cluster ~]# |
```

Script for setting up the EKS cluster EC2 instance.

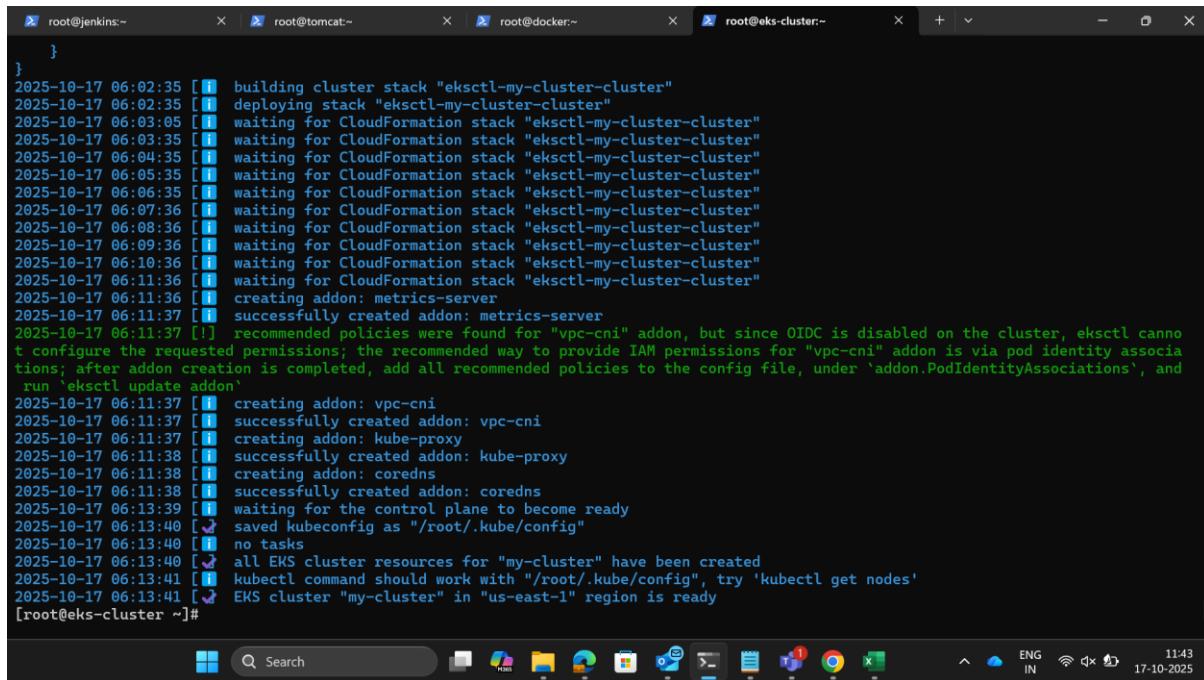


```
#!/bin/bash
#####
#Host EKS CLUSTER via EKSCTL
#####
#create iam role and attach policy
#ECRfull access,EKSpolicy and IAM full access
apt-get update -y
apt install unzip -y
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
aws configure
Install EKS Tool
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
sudo mv /tmp/eksctl /usr/local/bin
eksctl version
Install Kubectl
curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/releas
se/stable.txt)/bin/linux/amd64/kubectl
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
kubectl version --client
ssh-keygen
#Create EKS Cluster
eksctl create cluster --name my-cluster --region us-east-1 --version 1.32 --vpc-public-subnets subnet-0cd35a58e5c3b4df4,subnet-06662
1d1ee891ca82 --without-nodegroup
#####
#Create a Node Group#
#####
eksctl create nodegroup \
--cluster my-cluster \
--region us-east-1 \

```

1,11 Top

Cluster is being created.



```
}
```

```
2025-10-17 06:02:35 [i] building cluster stack "eksctl-my-cluster-cluster"
2025-10-17 06:02:35 [i] deploying stack "eksctl-my-cluster-cluster"
2025-10-17 06:03:05 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-10-17 06:03:35 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-10-17 06:04:35 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-10-17 06:05:35 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-10-17 06:06:35 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-10-17 06:07:36 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-10-17 06:08:36 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-10-17 06:09:36 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-10-17 06:10:36 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-10-17 06:11:36 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-10-17 06:11:36 [i] creating addon: metrics-server
2025-10-17 06:11:37 [i] successfully created addon: metrics-server
2025-10-17 06:11:37 [!] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl can't
configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and
run 'eksctl update addon'.
2025-10-17 06:11:37 [i] creating addon: vpc-cni
2025-10-17 06:11:37 [i] successfully created addon: vpc-cni
2025-10-17 06:11:37 [i] creating addon: kube-proxy
2025-10-17 06:11:38 [i] successfully created addon: kube-proxy
2025-10-17 06:11:38 [i] creating addon: coredns
2025-10-17 06:11:38 [i] successfully created addon: coredns
2025-10-17 06:13:39 [i] waiting for the control plane to become ready
2025-10-17 06:13:40 [i] saved kubeconfig as "/root/.kube/config"
2025-10-17 06:13:40 [i] no tasks
2025-10-17 06:13:40 [i] all EKS cluster resources for "my-cluster" have been created
2025-10-17 06:13:41 [i] kubectl command should work with "/root/.kube/config", try 'kubectl get nodes'
2025-10-17 06:13:41 [i] EKS cluster "my-cluster" in "us-east-1" region is ready
[root@eks-cluster ~]#
```

11:43 ENG IN 17-10-2025

Creating deployment file for Autoscaling purpose.

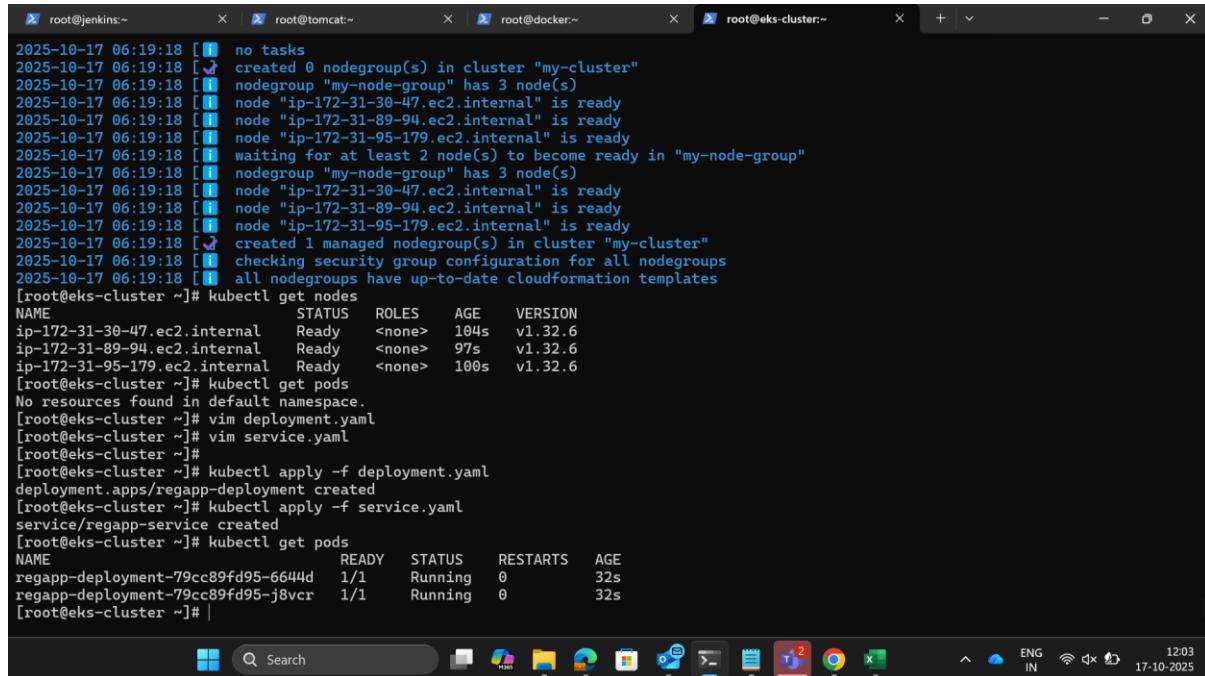
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: regapp-deployment
  labels:
    app: regapp

spec:
  replicas: 2
  selector:
    matchLabels:
      app: regapp

  template:
    metadata:
      labels:
        app: regapp
    spec:
      containers:
        - name: regapp
          image: 947390141947.dkr.ecr.us-east-1.amazonaws.com/tomcat-webapp:latest
          imagePullPolicy: Always
          ports:
            - containerPort: 8080
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
~
~
~
-- INSERT --
```

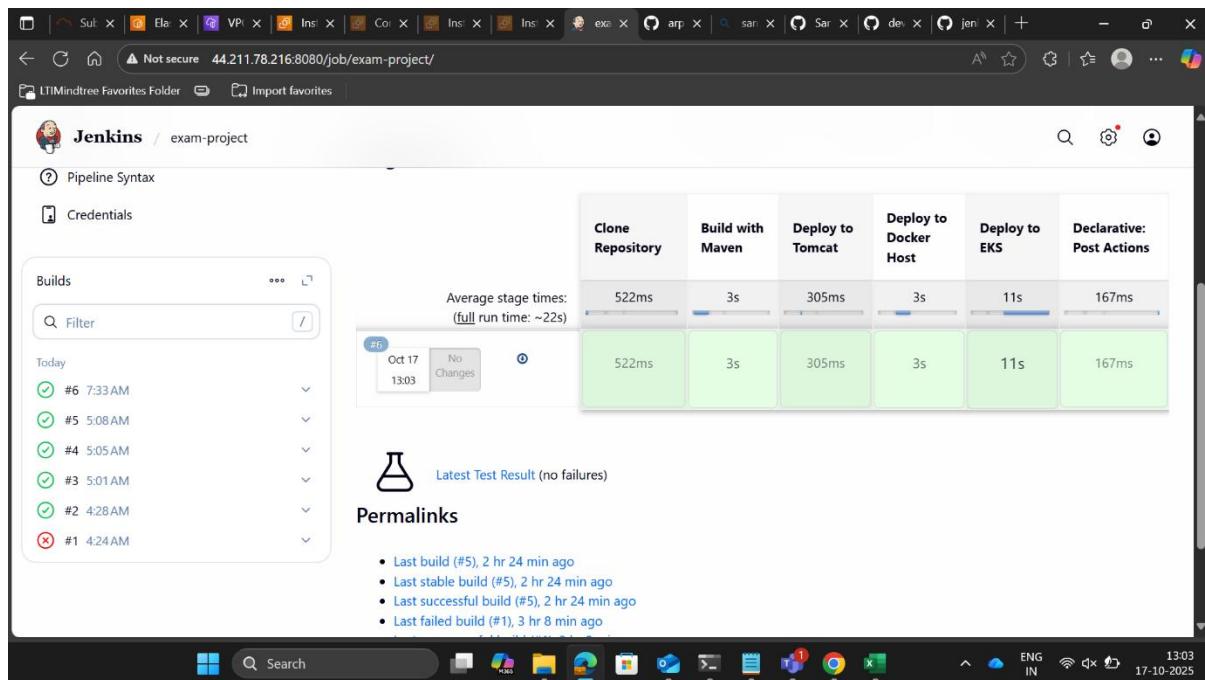
Creating service file for assigning i/p address for autoscaling group.

Running those two files deployment and service manifest files.



```
2025-10-17 06:19:18 [root@eks-cluster ~]# no tasks
2025-10-17 06:19:18 [root@eks-cluster ~]# created 0 nodegroup(s) in cluster "my-cluster"
2025-10-17 06:19:18 [root@eks-cluster ~]# nodegroup "my-node-group" has 3 node(s)
2025-10-17 06:19:18 [root@eks-cluster ~]# node "ip-172-31-30-47.ec2.internal" is ready
2025-10-17 06:19:18 [root@eks-cluster ~]# node "ip-172-31-89-94.ec2.internal" is ready
2025-10-17 06:19:18 [root@eks-cluster ~]# node "ip-172-31-95-179.ec2.internal" is ready
2025-10-17 06:19:18 [root@eks-cluster ~]# waiting for at least 2 node(s) to become ready in "my-node-group"
2025-10-17 06:19:18 [root@eks-cluster ~]# nodegroup "my-node-group" has 3 node(s)
2025-10-17 06:19:18 [root@eks-cluster ~]# node "ip-172-31-30-47.ec2.internal" is ready
2025-10-17 06:19:18 [root@eks-cluster ~]# node "ip-172-31-89-94.ec2.internal" is ready
2025-10-17 06:19:18 [root@eks-cluster ~]# node "ip-172-31-95-179.ec2.internal" is ready
2025-10-17 06:19:18 [root@eks-cluster ~]# created 1 managed nodegroup(s) in cluster "my-cluster"
2025-10-17 06:19:18 [root@eks-cluster ~]# checking security group configuration for all nodegroups
2025-10-17 06:19:18 [root@eks-cluster ~]# all nodegroups have up-to-date cloudformation templates
[root@eks-cluster ~]# kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-30-47.ec2.internal Ready <none> 104s v1.32.6
ip-172-31-89-94.ec2.internal Ready <none> 97s v1.32.6
ip-172-31-95-179.ec2.internal Ready <none> 100s v1.32.6
[root@eks-cluster ~]# kubectl get pods
No resources found in default namespace.
[root@eks-cluster ~]# vim deployment.yaml
[root@eks-cluster ~]# vim service.yaml
[root@eks-cluster ~]# kubectl apply -f deployment.yaml
deployment.apps/regapp-deployment created
[root@eks-cluster ~]# kubectl apply -f service.yaml
service/regapp-service created
[root@eks-cluster ~]# kubectl get pods
NAME READY STATUS RESTARTS AGE
regapp-deployment-79cc89fd95-6644d 1/1 Running 0 32s
regapp-deployment-79cc89fd95-j8vcr 1/1 Running 0 32s
[root@eks-cluster ~]#
```

Our build is successful.



The screenshot shows the Jenkins pipeline interface for a project named 'exam-project'. On the left, there's a sidebar with a 'Builds' section listing recent builds: #6 (7:33 AM), #5 (5:08 AM), #4 (5:05 AM), #3 (5:01 AM), #2 (4:28 AM), and #1 (4:24 AM). The main area displays a summary of pipeline stages: Clone Repository (522ms), Build with Maven (3s), Deploy to Tomcat (305ms), Deploy to Docker Host (3s), Deploy to EKS (11s), and Declarative: Post Actions (167ms). Below this, a 'Latest Test Result' indicates 'no failures'. At the bottom, a 'Permalinks' section provides links to specific build logs.

Clone Repository	Build with Maven	Deploy to Tomcat	Deploy to Docker Host	Deploy to EKS	Declarative: Post Actions
522ms	3s	305ms	3s	11s	167ms

Average stage times: (full run time: ~22s)

Oct 17 13:03 No Changes

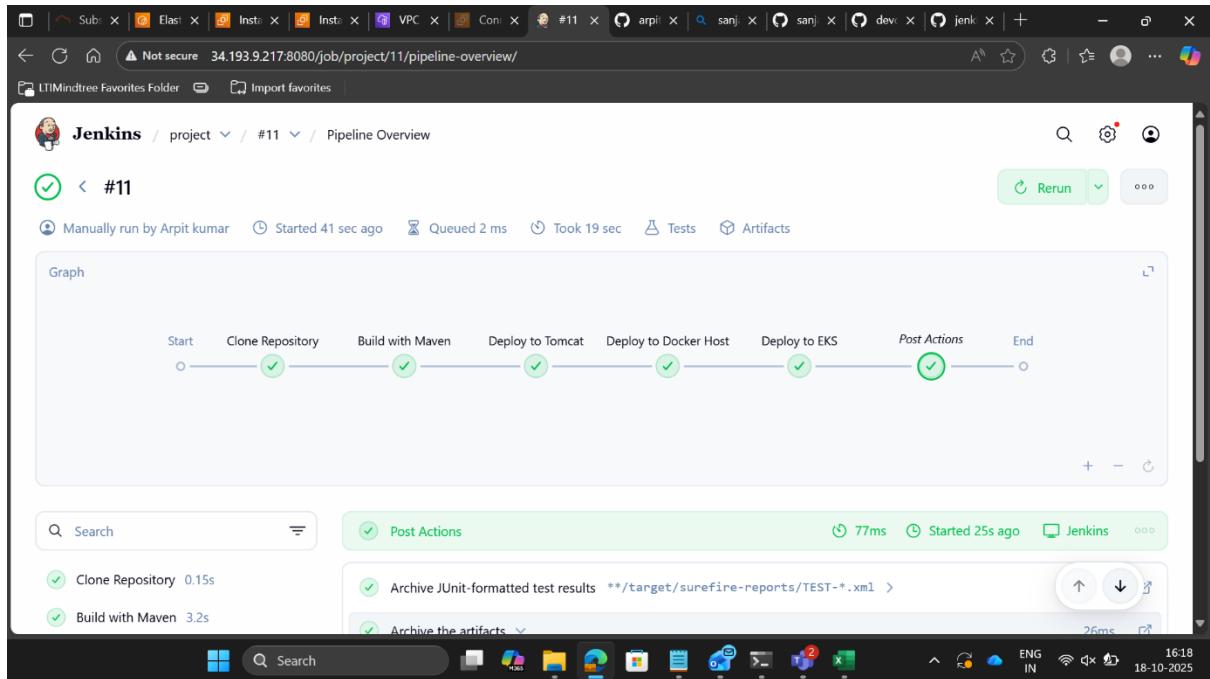
522ms 3s 305ms 3s 11s 167ms

Latest Test Result (no failures)

Permalinks

- Last build (#5), 2 hr 24 min ago
- Last stable build (#5), 2 hr 24 min ago
- Last successful build (#5), 2 hr 24 min ago
- Last failed build (#1), 3 hr 8 min ago

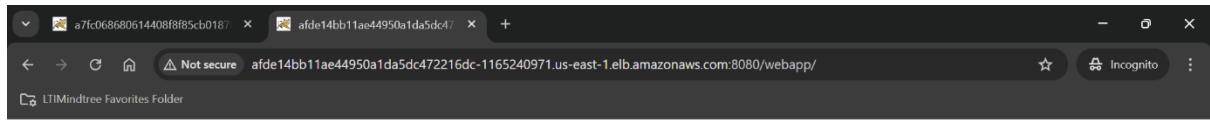
We can see the different stages of our build



Nodes, Pods and service i/p is being shown.

```
[root@eks-cluster ~]# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: ens5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 12:f5:c8:f7:3b:35 brd ff:ff:ff:ff:ff:ff
        altname enp0s5
        altname eni-04e8c731a241bf286
        altname device-number-0.0
        inet 172.31.80.93/20 metric 512 brd 172.31.95.255 scope global dynamic ens5
            valid_lft 3006sec preferred_lft 3006sec
            inet6 fe80::10f5:c8ff:fe7f:3b35/64 scope link proto kernel_ll
                valid_lft forever preferred_lft forever
[root@eks-cluster ~]# vim /etc/ssh/sshd_config
[root@eks-cluster ~]# systemctl restart sshd
[root@eks-cluster ~]# systemctl enable sshd
[root@eks-cluster ~]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@eks-cluster ~]# kubectl get svc
NAME      TYPE      CLUSTER-IP     EXTERNAL-IP          PORT(S)
AGE
kubernetes   ClusterIP   10.100.0.1   <none>           443/TCP
24m
regapp-service   LoadBalancer  10.100.22.89  afde14bb11ae44950a1da5dc472216dc-1165240971.us-east-1.elb.amazonaws.com  8080:31511/T
CP 37s
[root@eks-cluster ~]# |
```

Final output is displayed by taking the service ip:8080/webapp/ which is being hosted on the Load balancer.



I love learn devops teaching with sanjay sir loveeeee

Please fill in this form to create an account.

Enter Name Enter Full Name
Enter mobile Enter mobile number
Enter Email Enter Email
Password Enter Password
Repeat Password Repeat Password

By creating an account you agree to our [Terms & Privacy](#).

[Register](#)

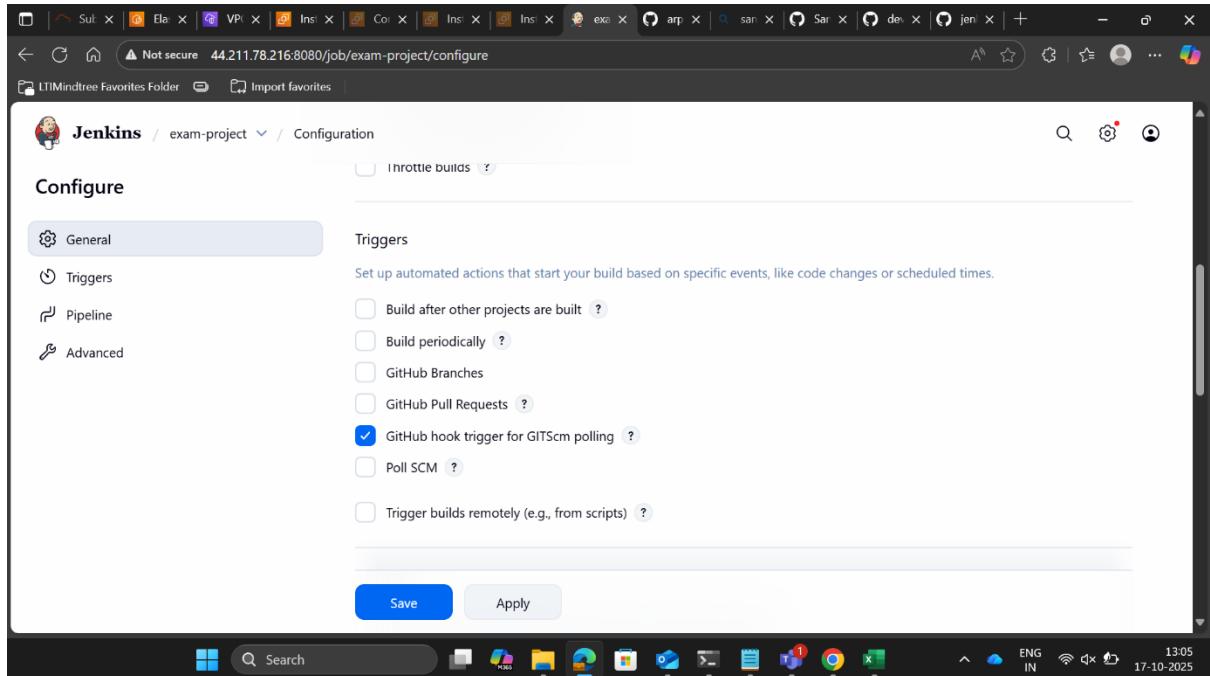
Already have an account? [Sign in](#).

Thank You

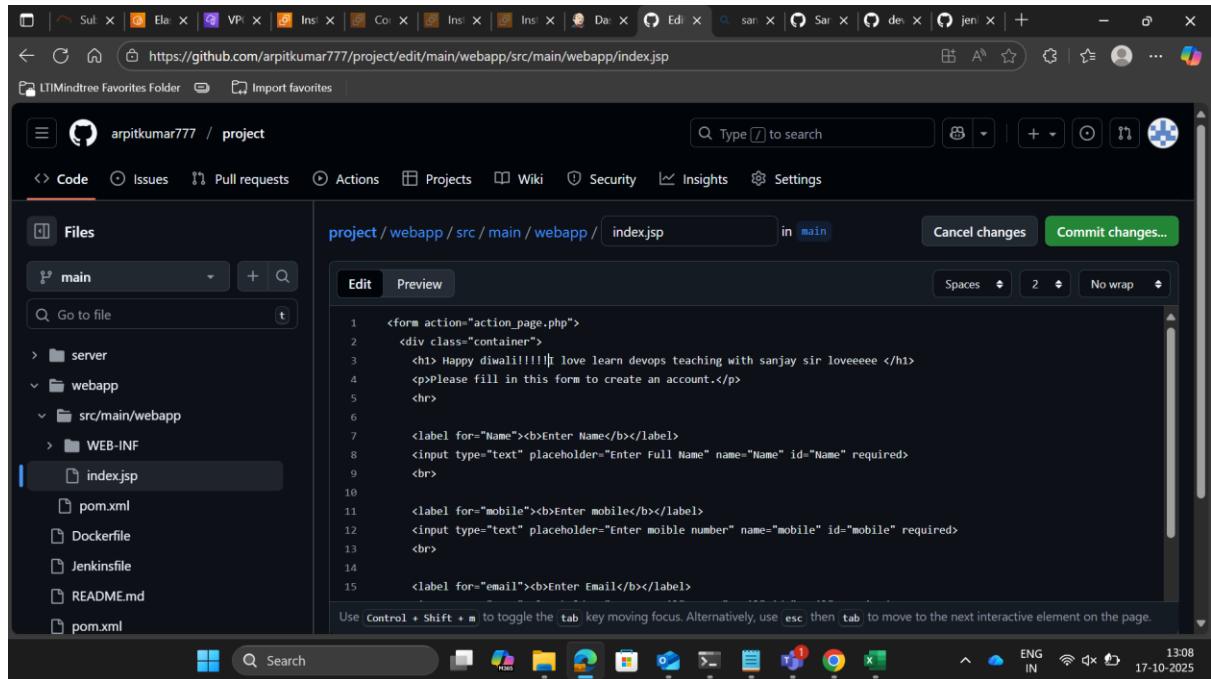
bye



Now we are allowing GitHub hook trigger for auto build purpose if any changes are being made then there will be automatic build.

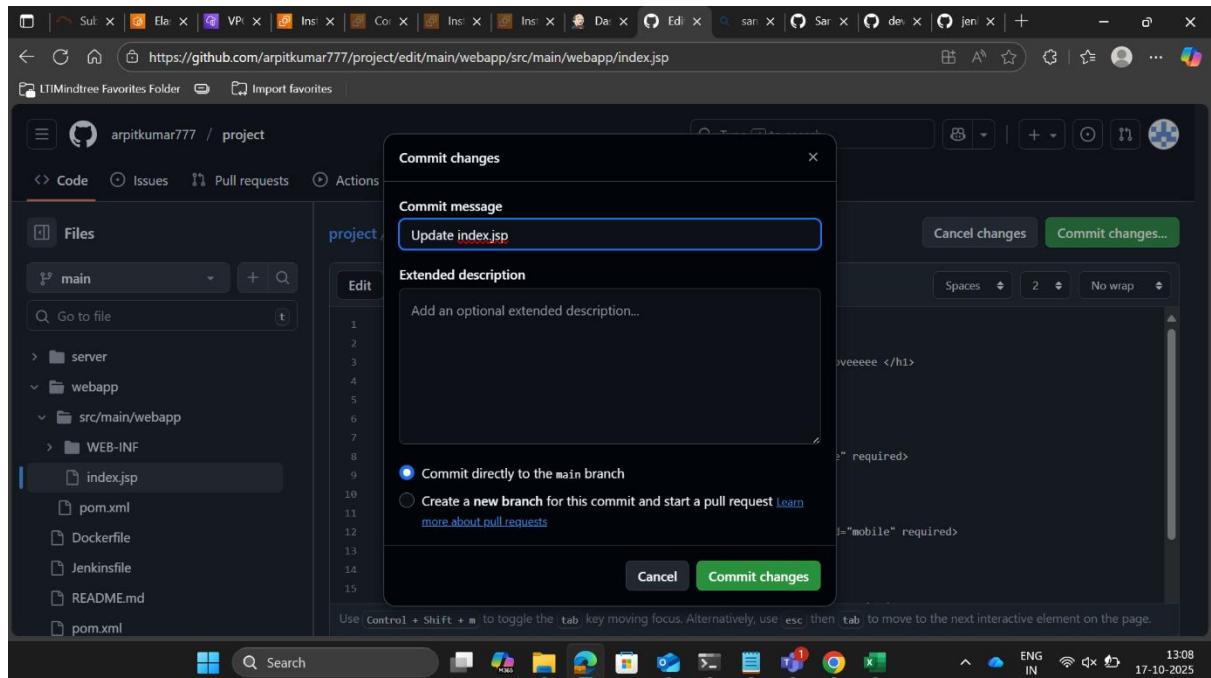


Now we are doing some changes in file from GitHub.



```
1 <form action="action_page.php">
2   <div class="container">
3     <h1> Happy diwali!!!! I love learn devops teaching with sanjay sir loveeee </h1>
4     <p>Please fill in this form to create an account.</p>
5     <hr>
6
7     <label for="Name"><b>Enter Name</b></label>
8     <input type="text" placeholder="Enter Full Name" name="Name" id="Name" required>
9     <br>
10
11    <label for="mobile"><b>Enter mobile</b></label>
12    <input type="text" placeholder="Enter moible number" name="mobile" id="mobile" required>
13    <br>
14
15    <label for="email"><b>Enter Email</b></label>
```

Now we are committing the changes.



We can see that there is automatic build after the changes are being committed.

Clone Repository	Build with Maven	Deploy to Tomcat	Deploy to Docker Host	Deploy to EKS	Declarative: Post Actions
324ms	3s	314ms	3s	7s	124ms
257ms	3s	315ms	4s	5s	102ms
194ms	3s	324ms	3s	5s	104ms
522ms	3s	305ms	3s	11s	167ms

We can see the changes

Please fill in this form to create an account.

Enter Name Enter Full Name
Enter mobile Enter mobile number
Enter Email Enter Email
Password Enter Password
Repeat Password Repeat Password

By creating an account you agree to our [Terms & Privacy](#).

[Register](#)

Already have an account? [Sign in](#).

Thank You

bye



Pipeline script:

```
pipeline {
    agent any
    environment {
        TOMCAT_USER = 'deployer'
        TOMCAT_PASS = 'deployer'
        TOMCAT_HOST = '3.87.231.9'
        TOMCAT_PORT = '8080'
    }
    stages {
        stage('Clone Repository') {
            steps {
                git branch: 'main', url: 'https://github.com/arpitkumar777/project.git'
            }
        }
        stage('Build with Maven') {
            steps {
                sh 'mvn clean package'
            }
        }
        stage('Deploy to Tomcat') {
            steps {
                sh """
                    curl -u $TOMCAT_USER:$TOMCAT_PASS \
                    --upload-file webapp/target/webapp.war \
                    "http://$TOMCAT_HOST:$TOMCAT_PORT/manager/text/deploy?path=/\
                    webapp&update=true"
                    """
            }
        }
        stage('Deploy to Docker Host') {
            steps {
                sshPublisher(publishers: [
                    sshPublisherDesc(
                        configName: 'docker',
                        transfers: [

```

```

        sshTransfer(
            sourceFiles: '**/*',
            removePrefix: '',
            remoteDirectory: 'image',
            execCommand: ""
                cd image
                aws ecr get-login-password --region us-east-1 | docker login --
username AWS --password-stdin 947390141947.dkr.ecr.us-east-
1.amazonaws.com
                docker build -t tomcat-webapp .
                docker tag tomcat-webapp:latest 947390141947.dkr.ecr.us-
east-1.amazonaws.com/tomcat-webapp:latest
                docker push 947390141947.dkr.ecr.us-east-
1.amazonaws.com/tomcat-webapp:latest
            """
        )
    ]
)
])
}
}
stage('Deploy to EKS'){
steps {
sshPublisher(publishers: [
sshPublisherDesc(
configName: 'eks',
transfers: [
sshTransfer(
sourceFiles: 'deployment.yaml,service.yaml',
remoteDirectory: "", // adjust as needed
removePrefix: '',
execCommand: ""
set -ex
aws eks update-kubeconfig --region us-east-1 --name my-cluster
kubectl delete -f deployment.yaml
kubectl apply -f deployment.yaml
kubectl apply -f service.yaml
kubectl rollout status deployment/regapp-deployment
"""
)
],

```

```
        usePromotionTimestamp: false,
        verbose: true
    )
])
}
}
}
post{
    success{
        junit '**/target/surefire-reports/TEST-*.xml'
        archiveArtifacts artifacts: '**/target/*.war', fingerprint: true
    }
}
}
```

NAME : ARPIT KUMAR
PS ID: 10844621