

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 usi

Approach:

Requirements:

- ✓ CI/CD pipeline System
- ✓ Git - local version control system.
- ✓ GitHub - As Distributed version control system.
- ✓ Jenkins - Continuous Integration tool.
- ✓ Maven - As a Build Tool.
- ✓ docker -Containerization
- ✓ 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.

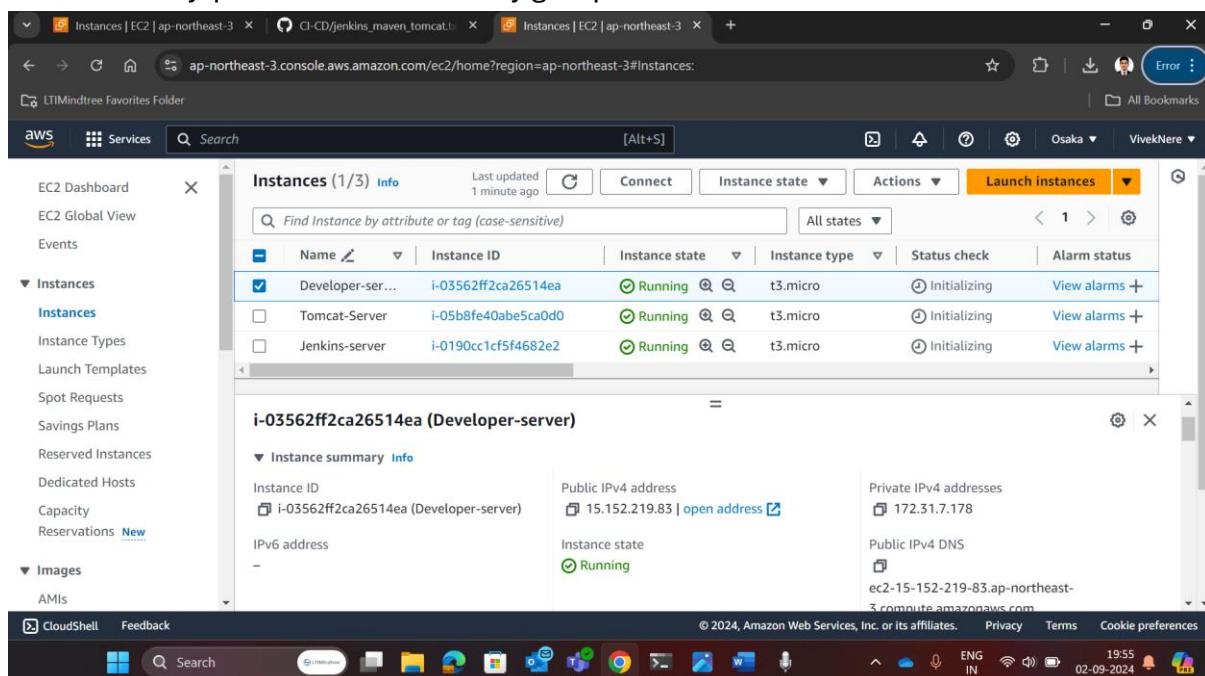
Step-4:

- 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:

#1: Git-Client:

Created Instances on AWS name as developer-server, Jenkins server and Tomcat server with same key-pair and same security group.



The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations (New), Images, and AMIs. The main content area displays a table of instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. There are three entries:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Developer-ser...	i-03562ff2ca26514ea	Running	t3.micro	Initializing	View alarms +
Tomcat-Server	i-05b8fe40abe5ca0d0	Running	t3.micro	Initializing	View alarms +
Jenkins-server	i-0190cc1cf5f4682e2	Running	t3.micro	Initializing	View alarms +

Below the table, a specific instance is selected: "i-03562ff2ca26514ea (Developer-server)". A detailed view panel shows the Instance summary, including Instance ID (i-03562ff2ca26514ea), Public IPv4 address (15.152.219.83), Private IPv4 addresses (172.31.7.178), and Public IPv4 DNS (ec2-15-152-219-83.ap-northeast-3.compute.amazonaws.com). The instance state is listed as "Running".

Connected developer server instance with local system through ssh and gave it a localhost name as developer-server

```

root@ip-172-31-7-178:~ + 
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\10748090>cd Downloads

C:\Users\10748090\Downloads>ssh -i "deployment-key-pair.pem" ec2-user@ec2-15-152-219-83.ap-northeast-3.compute.amazonaws.com
#_ _ _ _ _ Amazon Linux 2023
~~~ \####\ https://aws.amazon.com/linux/amazon-linux-2023
~~~ \###\ https://aws.amazon.com/linux/amazon-linux-2023
~~~ \#/ ___ V~-' '-->
~~~ .-' /-
~~~ .-' /-
~~~ .-' /-
Last login: Mon Sep  2 14:28:15 2024 from 136.226.233.19
[ec2-user@ip-172-31-7-178 ~]$ sudo su -
[root@ip-172-31-7-178 ~]# hostnamectl set-hostname developer-server.example.com
[root@ip-172-31-7-178 ~]# bash
[root@developer-server ~]#

```

Installed and initialized git in developer-server

```

root@developer-server ~]# yum install git -
Last metadata expiration check: 0:11:43 ago on Mon Sep  2 14:21:36 2024.
Dependencies resolved.
=====
Package           Architecture Version       Repository      Size
=====
Installing:
git              x86_64      2.40.1-1.amzn2023.0.3   amazonlinux    54 k
Installing dependencies:
git-core          x86_64      2.40.1-1.amzn2023.0.3   amazonlinux    4.3 M
git-core-doc     noarch      2.40.1-1.amzn2023.0.3   amazonlinux    2.6 M
perl-Error        noarch      1:0.17029-5.amzn2023.0.2   amazonlinux    41 k
perl-File-Find   noarch      1.37-477.amzn2023.0.6   amazonlinux    26 k
perl-Git          noarch      2.40.1-1.amzn2023.0.3   amazonlinux    42 k
perl-TermReadKey x86_64      2.38-9.amzn2023.0.2   amazonlinux    36 k
perl-lib          x86_64      0.65-477.amzn2023.0.6   amazonlinux    15 k
=====
Transaction Summary
=====
Install 8 Packages

Total download size: 7.1 M
Installed size: 34 M
Downloading Packages:
(1/8): git-2.40.1-1.amzn2023.0.3.x86_64.rpm          646 kB/s |  54 kB  00:00
(2/8): perl-Error-0.17029-5.amzn2023.0.2.noarch.rpm  1.7 MB/s |  41 kB  00:00
(3/8): perl-File-Find-1.37-477.amzn2023.0.6.noarch.rpm 1.6 MB/s |  26 kB  00:00
(4/8): git-core-doc-2.40.1-1.amzn2023.0.3.noarch.rpm  16 MB/s | 2.6 MB  00:00
(5/8): perl-Git-2.40.1-1.amzn2023.0.3.noarch.rpm    1.0 MB/s |  42 kB  00:00
(6/8): git-core-2.40.1-1.amzn2023.0.3.x86_64.rpm    19 MB/s | 4.3 MB  00:00
(7/8): perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64.rpm 567 kB/s |  36 kB  00:00
(8/8): perl-lib-0.65-477.amzn2023.0.6.x86_64.rpm    244 kB/s |  15 kB  00:00
=====
Complete!
[root@developer-server ~]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/.git/

```

Created and added public ssh-key to github-account for ssh-authentication

```
[root@developer-server ~]# ssh-keygen
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 fingerprint is:
SHA256:TtxFxdjWDhr6n236p2vhcbdrIZPqV506W81CZfPy8 root@developer-server.example.com
The key's randomart image is:
+---[RSA 3072]---+
|          . ++
|         . o o.o
|        . .
|       . o
|      So. o
|     o . . *+
|    ... . +o#=
|   . . . +E#
|  . . o==*XX|
+---[SHA256]---+
[root@developer-server ~]# ll
total 0
[root@developer-server ~]# ls -a
. .... .bash_logout .bash_profile .bashrc .cshrc .git .ssh .tcshrc
[root@developer-server ~]# cd .ssh/
[root@developer-server .ssh]# ll
total 12
-rw-----. 1 root root 567 Sep 2 14:21 authorized_keys
-rw-----. 1 root root 2622 Sep 2 14:35 id_rsa
-rw-r--r--. 1 root root 587 Sep 2 14:35 id_rsa.pub
[root@developer-server .ssh]# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCR1f+JL7wWQzrqJva2LBHotkq621Se1uQF+vPmSSqv2h/BUL65fjiaw+oCbXesFI+xiXce5YBRsJuqrpAabSB1moe7lWQedsZj55jzlyLW3rXAYKqav8
due5uWV18VkcqjLrRjA/xycfRBim+2almoYWFZaritaloWQzq5aK6Yt1Zsb7vAkAiCdGufrtlpvq0Cgjw9o+vt+BtzswHGeVhBTUN+SIP+9S6VeTbxEcdkrTwB89xG5PPKAkqKwvFzljqZAsvLY
tiwTA/BqvZkul.6P0Hpl3jRnBARJ7Djfut0spXawEANjQrSwTx/KDS19NyxxSssWpkXK8cVE/uAsmpyNorAoYdXmKrMzyVm/XyhAjSEb2Q0jxitMvgCoiBwqTj29W5YEkpA80e1Yswe2n8vu2ngglfUIE
quvqf4AVtomFqvOppsV1hNoNcoofpBmHSzs5xTgLvBAmCF11lQPvFO3GAGuwl7SsevUrgPLteQ/VilNm78= root@developer-server.example.com
[root@developer-server .ssh]# ^C
[root@developer-server .ssh]# |
```

Add new SSH Key

Title: Deployment-ssh-key

Key type: Authentication Key

Key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQCR1f+JL7wWQzrqJva2LBHotkq621Se1uQF+vPmSSqv2h/BUL65fjiaw+oCbXesFI+xiXce5YBRsJuqrpAabSB1moe7lWQedsZj55jzlyLW3rXAYKqav8
due5uWV18VkcqjLrRjA/xycfRBim+2almoYWFZaritaloWQzq5aK6Yt1Zsb7vAkAiCdGufrtlpvq0Cgjw9o+vt+BtzswHGeVhBTUN+SIP+9S6VeTbxEcdkrTwB89xG5PPKAkqKwvFzljqZAsvLY
tiwTA/BqvZkul.6P0Hpl3jRnBARJ7Djfut0spXawEANjQrSwTx/KDS19NyxxSssWpkXK8cVE/uAsmpyNorAoYdXmKrMzyVm/XyhAjSEb2Q0jxitMvgCoiBwqTj29W5YEkpA80e1Yswe2n8vu2ngglfUIE
quvqf4AVtomFqvOppsV1hNoNcoofpBmHSzs5xTgLvBAmCF11lQPvFO3GAGuwl7SsevUrgPLteQ/VilNm78= root@developer-server.example.com
```

Add SSH key

CLONE JAVA PROJECT FROM GIT REPOSITORY WHERE IT IS PRESENT

```
[root@developer-server .ssh]# cd
[root@developer-server ~]# git clone https://github.com/VivekNere1/JENKINS.git
Cloning into 'JENKINS'...
remote: Enumerating objects: 155, done.
remote: Counting objects: 100% (155/155), done.
remote: Compressing objects: 100% (69/69), done.
remote: Total 155 (delta 39), reused 139 (delta 35), pack-reused 0 (from 0)
Receiving objects: 100% (155/155), 12.41 MiB | 17.50 MiB/s, done.
Resolving deltas: 100% (39/39), done.
[root@developer-server ~]# cd jenkins
bash: cd: jenkins: No such file or directory
[root@developer-server ~]# ll
total 0
drwxr-xr-x. 5 root root 179 Sep 2 14:39 JENKINS
[root@developer-server ~]# cd JENKINS/
[root@developer-server JENKINS]# |
```

Created deployment name repository in my account to push the code in that repository which we have cloned just now

The screenshot shows a GitHub repository page for 'VivekNere1/Deployment'. The repository is public and has 0 stars. It features a 'Deployment' section. A 'Quick setup' box provides instructions for cloning the repository via HTTPS or SSH, and it suggests including a README, LICENSE, and .gitignore file.

Now we have successfully push the code on my own branch

```
[root@developer-server JENKINS]# git init
Reinitialized existing Git repository in /root/JENKINS/.git/
[root@developer-server JENKINS]# git add .
[root@developer-server JENKINS]# git commit -m "first commit"
On branch main
nothing to commit, working tree clean
[root@developer-server JENKINS]# git remote remove origin
[root@developer-server JENKINS]# git remote add origin git@github.com:VivekNere1/Deployment.git
[root@developer-server JENKINS]# git branch -M main
[root@developer-server JENKINS]# git push origin main
The authenticity of host 'github.com (20.27.177.113)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbZisF/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: 143, done.
Counting objects: 100% (143/143), done.
Delta compression using up to 2 threads
Compressing objects: 100% (60/60), done.
Writing objects: 100% (143/143), 12.41 MiB | 3.37 MiB/s, done.
Total 143 (delta 39), reused 143 (delta 39), pack-reused 0
remote: Resolving deltas: 100% (39/39), done.
To github.com:VivekNere1/Deployment.git
 * [new branch]      main -> main
[root@developer-server JENKINS]# ]
```

The GitHub repository page now shows the pushed code. The commit history includes:

- VivekNere1 Update index.jsp (4b78e64, 2 days ago, 28 commits)
- server first commit (last year)
- webapp Update index.jsp (2 days ago)
- Dockerfile first commit (last year)
- README.md Changed README File (last year)
- Srijay_Devops_projectc.pdf Add files via upload (2 months ago)
- pom.xml first commit (last year)
- regapp-deploy.yml Update regapp-deploy.yml (last year)
- regapp-service.yml Update regapp-service.yml (last year)

The repository has 0 stars and 0 forks. It also lists 'About', 'Releases', and 'Packages' sections.

Now we have to open Jenkins-server instance in our machine using ssh and I have gave the hostname as Jenkins-server

```
root@ip-172-31-7-178:~/JENKI root@ip-172-31-0-183:~ + 
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\10748090>cd Downloads

C:\Users\10748090\Downloads>ssh -i "deployment-key-pair.pem" ec2-user@ec2-15-152-106-53.ap-northeast-3.compute.amazonaws.com
The authenticity of host 'ec2-15-152-106-53.ap-northeast-3.compute.amazonaws.com (15.152.106.53)' can't be established.
ED25519 key fingerprint is SHA256:+rlHQNon09IRtiMbARTn0XyydN8QkTsRpAvSQ0/fs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-15-152-106-53.ap-northeast-3.compute.amazonaws.com' (ED25519) to the list of known hosts.

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

[ec2-user@ip-172-31-0-183 ~]$ sudo su -
[root@ip-172-31-0-183 ~]# hostnamectl set-hostname jenkins-server.example.com
[root@ip-172-31-0-183 ~]# bash
[root@jenkins-server ~]#
```

Now we have to add one extra inbound rule in security group for Jenkins server instance with port number 8080

The screenshot shows the AWS CloudWatch Metrics interface. A single metric named 'AWS/CloudWatch Metrics' is displayed with a value of 1. The metric has a timestamp of 2023-07-17T10:45:00Z.

Now we have to install and initialized git in Jenkins-server

```

root@jenkins-server ~]# yum install git -y
Last metadata expiration check: 0:32:05 ago on Mon Sep  2 14:23:16 2024.
Dependencies resolved.
=====
Package           Architecture      Version       Repository      Size
=====
Installing:
git              x86_64          2.40.1-1.amzn2023.0.3   amazonlinux    54 k
Installing dependencies:
git-core          x86_64          2.40.1-1.amzn2023.0.3   amazonlinux    4.3 M
git-core-doc      noarch          2.40.1-1.amzn2023.0.3   amazonlinux    2.6 M
perl-Error        noarch          1:0.17029-5.amzn2023.0.2   amazonlinux    41 k
perl-File-Find    noarch          1.37-477.amzn2023.0.6    amazonlinux    26 k
perl-Git          noarch          2.40.1-1.amzn2023.0.3   amazonlinux    42 k
perl-TermReadKey  x86_64          2.38-9.amzn2023.0.2     amazonlinux    36 k
perl-lib          x86_64          0.65-477.amzn2023.0.6    amazonlinux    15 k

Transaction Summary
=====
Install 8 Packages

Total download size: 7.1 M
Installed size: 34 M
Downloading Packages:
(1/8): git-2.40.1-1.amzn2023.0.3.x86_64.rpm           877 kB/s | 54 kB  00:00
(2/8): perl-Error-0.17029-5.amzn2023.0.2.noarch.rpm     1.7 MB/s | 41 kB  00:00
(3/8): perl-File-Find-1.37-477.amzn2023.0.6.noarch.rpm   977 kB/s | 26 kB  00:00
(4/8): git-core-doc-2.40.1-1.amzn2023.0.3.noarch.rpm    17 MB/s | 2.6 MB  00:00
(5/8): git-core-2.40.1-1.amzn2023.0.3.x86_64.rpm        23 MB/s | 4.3 MB  00:00
(6/8): perl-Git-2.40.1-1.amzn2023.0.3.noarch.rpm       549 kB/s | 42 kB  00:00
(7/8): perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64.rpm  1.0 MB/s | 36 kB  00:00
(8/8): perl-lib-0.65-477.amzn2023.0.6.x86_64.rpm       315 kB/s | 15 kB  00:00

Complete!
[root@jenkins-server ~]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/.git/

```

#2: Jenkin & Maven :

Now we have to install and configure java and Jenkins on Jenkins-server

```

root@jenkins-server ~]# java -version
openjdk version "17.0.12" 2024-07-16 LTS
OpenJDK Runtime Environment Corretto-17.0.12.7.1 (build 17.0.12+7-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.12.7.1 (build 17.0.12+7-LTS, mixed mode, sharing)
[root@jenkins-server ~]# wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-09-02 15:01:37 -- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io) ... 151.101.90.133, 2a00:4e42:15::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.90.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

/etc/yum.repos.d/jenkins.repo 100%[=====] 85 --.kB/s in 0s
2024-09-02 15:01:28 (5.55 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[root@jenkins-server ~]# rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[root@jenkins-server ~]# dnf install jenkins -y
Jenkins-stable
Dependencies resolved.
=====
Package           Architecture      Version       Repository      Size
=====
Installing:
jenkins          noarch          2.462.1-1.1      jenkins       89 M

Transaction Summary
=====
Install 1 Package

Total download size: 89 M
Installed size: 89 M
Downloading Packages:
jenkins-2.462.1-1.1.noarch.rpm           9.9 MB/s | 89 MB  00:08

Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Running scriptlet: jenkins-2.462.1-1.1.noarch
Installing  : jenkins-2.462.1-1.1.noarch 1/1
Running scriptlet: jenkins-2.462.1-1.1.noarch 1/1
Verifying   : jenkins-2.462.1-1.1.noarch 1/1

Installed:
jenkins-2.462.1-1.1.noarch

```

Enable and started Jenkins

```
[root@jenkins-server ~]# sudo systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[root@jenkins-server ~]# sudo systemctl start jenkins
[root@jenkins-server ~]# ^C
[root@jenkins-server ~]# |
```

Accessed Jenkins dashboard via, Jenkins-server machine's public ip address and 8080 port, and completed login process.

The screenshot shows a browser window with the URL `15.152.106.53:8080/login?from=%2F`. The page title is "Getting Started" and the main heading is "Unlock Jenkins". The text on the page says: "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`". Below this, it says "Please copy the password from either location and paste it below." There is a text input field labeled "Administrator password" with a placeholder "Paste your password here". At the bottom right is a blue "Continue" button.

The screenshot shows a browser window with the URL `ec2-13-208-222-196.ap-northeast-3.compute.amazonaws.com:8080`. The page title is "Getting Started". The main heading is "Getting Started". Below it is a table with several rows and columns of Jenkins features. A tooltip for the "Workspace Cleanup" feature is open, showing its dependencies: "Commons-text API", "Pipeline: Supporting APIs", "Plugin Utilities API", "Font Awesome API", "Bootstrap 5 API", "jQuery3 API", "ECharts API", "Display URL API", "Checks API", "JUnit", "Matrix Project", "Resource Disposer", and "Workspace Cleanup". At the bottom of the page, it says "Jenkins 2.462.1".

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

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Create a job +

Add description

Dashboard > Vivek Kailas Nere > Configure

Plain text Preview

API Token

Current token(s) ?

Token created on 2024-09-03T03:37:17.0 1143893bf3eb932e7ccb9ba759b0cc4355

⚠ Copy this token now, because it cannot be recovered in the future.

Add new Token

github.com/VivekNere1/Deployment/settings/hooks/new

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Webhooks / Add webhook

We'll send a `POST` request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, etc). More information can be found in our developer documentation.

Payload URL *

`http://13.57.202.161:8080/github-webhook/`

Content type *

`application/x-www-form-urlencoded`

Secret

`1143893bf3eb932e7ccb9ba759b0cc4355`

SSL verification

By default, we verify SSL certificates when delivering payloads.

Enable SSL verification Disable (not recommended)

Dashboard > Deployment-CI-CD > Configuration

Configure

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

Repositories

Repository URL: `https://github.com/VivekNere1/Deployment.git`

Credentials: - none -

+ Add

Advanced

Add Repository

Branches to build

+ Add

Advanced

Add Repository

Branch Specifier (blank for 'any')

*/main

Add Branch

Configure

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

Build Triggers

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?
- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

Not secure ec2-13-208-222-196.ap-northeast-3.compute.amazonaws.com:8080/job/Deployment-CI-CD/2/console

LTMindtree Favorites Folder

Jenkins

Search (CTRL+K)

Vivek Kailas Nere log out

Dashboard > Deployment-CI-CD > #2 > Console Output

Status

Changes

Console Output

Edit Build Information

Delete build '#2'

Timings

Git Build Data

Previous Build

Console Output

Download Copy View as plain text

```

Started by user Vivek Kailas Nere
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Deployment-CI-CD
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Deployment-CI-CD/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/VivekNere1/Deployment.git # timeout=10
Fetching upstream changes from https://github.com/VivekNere1/Deployment.git
> git --version # timeout=10
> git --version # 'git version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/VivekNere1/Deployment.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 4b78e644e172573f7b2e409ab0504d961e0a5b47 (refs/remotes/origin/main)

```

```

Fetching upstream changes from https://github.com/VivekNere1/Deployment.git
  > git --version # timeout=10
  > git --version # 'git version 2.40.1'
  > git fetch --tags --force --progress -- https://github.com/VivekNere1/Deployment.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
  > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 4b78e644e172573f7b2e409ab0504d961e0a5b47 (refs/remotes/origin/main)
  > git config core.sparsecheckout # timeout=10
  > git checkout -f 4b78e644e172573f7b2e409ab0504d961e0a5b47 # timeout=10
Commit message: "Update index.jsp"
  > git rev-list --no-walk 4b78e644e172573f7b2e409ab0504d961e0a5b47 # timeout=10
Finished: SUCCESS

```

#Installation and Configuration of maven:

```

[root@jenkins-server ~]# cd /opt
[root@jenkins-server opt]# wget https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz
--2024-09-03 03:59:46-- https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9102945 (8.7M) [application/x-gzip]
Saving to: 'apache-maven-3.9.9-bin.tar.gz'

apache-maven-3.9.9-bin.tar.gz    100%[=====] 8.68M --.-KB/s   in 0.02s

2024-09-03 03:59:46 (445 MB/s) - 'apache-maven-3.9.9-bin.tar.gz' saved [9102945/9102945]

[root@jenkins-server opt]# tar -xzf apache-maven-3.9.9-bin.tar.gz
apache-maven-3.9.9/README.txt
apache-maven-3.9.9/LICENSE
apache-maven-3.9.9/NOTICE
apache-maven-3.9.9/lib/
apache-maven-3.9.9/lib/aopalliance.license
apache-maven-3.9.9/lib/commons-cli.license
apache-maven-3.9.9/lib/commons-codec.license
apache-maven-3.9.9/lib/failureaccess.license
apache-maven-3.9.9/lib/guava.license
apache-maven-3.9.9/lib/guice.license
apache-maven-3.9.9/lib/httpclient.license
apache-maven-3.9.9/lib/httpcore.license
apache-maven-3.9.9/lib/jansi.license
apache-maven-3.9.9/lib/javax.annotation-api.license
apache-maven-3.9.9/lib/javax.inject.license
apache-maven-3.9.9/lib/jcl-over-slf4j.license
apache-maven-3.9.9/lib/org.eclipse.sisu.inject.license

[root@jenkins-server opt]# cd maven/
[root@jenkins-server maven]# cd bin/
[root@jenkins-server bin]# ./mvn -v
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcdc97d260186937)
Maven home: /opt/maven
Java version: 17.0.12, 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: "6.1.102-111.182.amzn2023.x86_64", arch: "amd64", family: "unix"
[root@jenkins-server bin]# cd ..
[root@jenkins-server maven]# cd ~
[root@jenkins-server ~]# pwd
/root
[root@jenkins-server ~]# ll -a
total 24
dr-xr-x---. 4 root root 133 Sep  3 03:41 .
dr-xr-xr-x. 18 root root 237 Aug 15 17:15 ..
-rw-r--r--. 1 root root 18 Feb  2 2023 .bash_logout
-rw-r--r--. 1 root root 141 Feb  2 2023 .bash_profile
-rw-r--r--. 1 root root 429 Feb  2 2023 .bashrc
-rw-r--r--. 1 root root 100 Feb  2 2023 .cshrc
drwxr-xr-x. 7 root root 119 Sep  3 03:41 .git
drwx-----. 2 root root 29 Sep  3 03:21 .ssh
-rw-r--r--. 1 root root 129 Feb  2 2023 .tcsSRC
-rw-r--r--. 1 root root 164 Sep  3 03:24 .wget-hsts

```

#Install maven plugins and configure Jenkins for maven

in Jenkins -> manage Jenkins -> plugins -> available plugins -> install maven integration

Available plugins - Plugins [Jenkins]

Not secure ec2-13-208-222-196.ap-northeast-3.compute.amazonaws.com:8080/manage/pluginManager/available

LTIMindtree Favorites Folder

Jenkins

Search (CTRL+K)

Vivek Kailas Nere log out

Dashboard > Manage Jenkins > Plugins

Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Search: maven integration

Install Name Released

Maven Integration 3.23 Build Tools
This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.
1 yr 1 mo ago

Pipeline Maven Integration 1421:v610fa_b_e2d60e Pipeline Maven
This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path.
2 mo 28 days ago

To find out the home directory of maven and java

```
[root@jenkins-server ~]# cd /opt/
[root@jenkins-server opt]# cd maven/
[root@jenkins-server maven]# cd bin/
[root@jenkins-server bin]# mvn -v
Apache Maven 3.8.4 (Red Hat 3.8.4-3.amzn2023.0.5)
Maven home: /usr/share/maven
Java version: 17.0.12, 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: "6.1.102-111.182.amzn2023.x86_64", arch: "amd64", family: "unix"
[root@jenkins-server bin]# |
```

- now we need to configure this plugins for that go to manage Jenkins tools-> add JDK -> name=java11 JAVA_HOME=/usr/lib/jvm/jre-17-openjdk then add maven ==> name=maven , MAVEN_HOME= /usr/share/maven

Tools [Jenkins]

Not secure ec2-13-208-222-196.ap-northeast-3.compute.amazonaws.com:8080/manage/configureTools/

LTIMindtree Favorites Folder

Dashboard > Manage Jenkins > Tools

JDK installations

Add JDK

Name: java11

JAVA_HOME: /usr/lib/jvm/java-17-amazon-corretto.x86_64

Install automatically

Save Apply

Maven installations

Add Maven

Maven

Name: maven

MAVEN_HOME: /usr/share/maven

Install automatically

Save Apply

- now manage Jenkins -> plugins -> Installed plugins -> Disable the GITHUB branch source plugin & enable GitHub plugin and restart.

Installed plugins - Plugins [Jenk]

Dashboard > Manage Jenkins > Plugins

Plugins

Search: github branch source

Updates Available plugins Installed plugins Advanced settings Download progress

GitHub Branch Source Plugin 1797.v86fdb_4d57d43
Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.
Report an issue with this plugin

Enabled

GitHub API Plugin 1.321-468.v6a_9f5f2d5a_7e
This plugin provides GitHub API for other plugins.
Report an issue with this plugin

Enabled

GitHub Branch Source Plugin 1797.v86fdb_4d57d43
Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.
Report an issue with this plugin

Enabled

GitHub plugin 1.40.0
This plugin integrates GitHub to Jenkins.
Report an issue with this plugin

Enabled



Please wait while Jenkins is restarting ...

Your browser will reload automatically when Jenkins is ready.

Safe Restart

Builds on agents can usually continue.

#3: Tomcat

#Installation and configuration of Tomcat:

Open the tomcat-server instance into local machine by using ssh and set a hostname as tomcat-server

```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\10748090\Downloads>clear
'clear' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\10748090\Downloads>ssh -i "deployment-key-pair.pem" ec2-user@ec2-15-152-159-145.ap-northeast-3.compute.amazonaws.com
The authenticity of host 'ec2-15-152-159-145.ap-northeast-3.compute.amazonaws.com (15.152.159.145)' can't be established.
ED25519 key fingerprint is SHA256:q76TWMMmxuqrPKTRu6YRuN0iWFk/P3FpFQ2Wfn908t/8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-15-152-159-145.ap-northeast-3.compute.amazonaws.com' (ED25519) to the list of known hosts.

      _#
     /_###_
    /_####\
   /##|
  /#/ _-->
  /`'`->
  /`_`_`/
  /`_`_`/
  /`_`_`/
  /`_`_`/
[ec2-user@ip-172-31-9-125 ~]$ sudo su -
[root@ip-172-31-9-125 ~]# hostnamectl set-hostname tomcat-server.example.com
[root@ip-172-31-9-125 ~]# bash
[root@tomcat-server ~]# |
```

Installed java in tomcat-server

```
[root@tomcat-server ~]# dnf install java-17-amazon-corretto -y
Last metadata expiration check: 14:33:58 ago on Mon Sep  2 14:24:17 2024.
Dependencies resolved.
=====
Package           Architecture Version      Repository  Size
=====
java-17-amazon-corretto x86_64    1:17.0.12+7-1.amzn2023.1 amazonlinux 187 k
Installing dependencies:
alsa-lib          x86_64    1.2.7.2-1.amzn2023.0.2 amazonlinux 504 k
cairo             x86_64    1.17.6-2.amzn2023.0.1 amazonlinux 684 k
dejavu-sans-fonts noarch    2.37-16.amzn2023.0.2 amazonlinux 1.3 M
dejavu-sans-mono-fonts noarch    2.37-16.amzn2023.0.2 amazonlinux 467 k
dejavu-serif-fonts noarch    2.37-16.amzn2023.0.2 amazonlinux 1.0 M
fontconfig         x86_64    2.13.94-2.amzn2023.0.2 amazonlinux 273 k
fonts-filesystem  noarch    1:2.0.5-12.amzn2023.0.1 amazonlinux 9.5 k
freetype           x86_64    2.13.2-5.amzn2023.0.1 amazonlinux 423 k
giflib             x86_64    5.2.1-9.amzn2023.0.1 amazonlinux 49 k
google-noto-fonts-common noarch    20201206-2.amzn2023.0.2 amazonlinux 15 k
google-noto-sans-vf-fonts noarch    20201206-2.amzn2023.0.2 amazonlinux 492 k
graphite2          x86_64    1.3.14-7.amzn2023.0.2 amazonlinux 97 k
harfbuzz           x86_64    7.0.0-2.amzn2023.0.1 amazonlinux 868 k
java-17-amazon-corretto-headless x86_64    1:17.0.12+7-1.amzn2023.1 amazonlinux 91 M
javapackages-filesystem noarch    6.0.0-7.amzn2023.0.6 amazonlinux 12 k
langpacks-core-font-en noarch    3.0-21.amzn2023.0.4 amazonlinux 10 k
libICE             x86_64    1.0.10-6.amzn2023.0.2 amazonlinux 71 k
libSM              x86_64    1.2.3-8.amzn2023.0.2 amazonlinux 42 k
libX11             x86_64    1.7.2-3.amzn2023.0.4 amazonlinux 657 k
libX11-common      noarch    1.7.2-3.amzn2023.0.4 amazonlinux 152 k
libXau             x86_64    1.0.9-6.amzn2023.0.2 amazonlinux 31 k
libXext             x86_64    1.3.4-6.amzn2023.0.2 amazonlinux 41 k
libXi              x86_64    1.7.10-6.amzn2023.0.2 amazonlinux 40 k
libXinerama        x86_64    1.1.4-8.amzn2023.0.2 amazonlinux 15 k
```

Download tar file of tomcat from tomcat official website in /opt folder

```
[root@tomcat-server ~]# cd /opt/
[root@tomcat-server opt]# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.tar.gz
--2024-09-03 05:06:13-- https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12122732 (12M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.93.tar.gz'

apache-tomcat-9.0.93.tar.gz      100%[=====] 11.56M --.-KB/s   in 0.03s

2024-09-03 05:06:14 (396 MB/s) - 'apache-tomcat-9.0.93.tar.gz' saved [12122732/12122732]
[root@tomcat-server opt]# |
```

Extract tar file:

```
[root@tomcat-server opt]# ls
apache-tomcat-9.0.93.tar.gz  aws
[root@tomcat-server opt]# tar xvzf apache-tomcat-9.0.93.tar.gz
apache-tomcat-9.0.93/conf/
apache-tomcat-9.0.93/conf/catalina.policy
apache-tomcat-9.0.93/conf/catalina.properties
apache-tomcat-9.0.93/conf/context.xml
apache-tomcat-9.0.93/conf/jaspic-providers.xml
apache-tomcat-9.0.93/conf/jaspic-providers.xsd
```

Move in tomcat folder

Now we have to start the tomcat server

```
[root@tomcat-server opt]# mv apache-tomcat-9.0.93 tomcat
[root@tomcat-server opt]# cd tomcat
[root@tomcat-server tomcat]# cd bin
[root@tomcat-server bin]# ./startup.sh
Using CATALINA_BASE:  /opt/tomcat
Using CATALINA_HOME:  /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:       /usr
Using CLASSPATH:      /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@tomcat-server bin]# |
```

in google open [http://\(ip of tomcat server instance\):8080](http://(ip of tomcat server instance):8080)

now we have make some changes in context.xml file for that we need to find that file in tomcat

```
[root@tomcat-server bin]# cd ..
[root@tomcat-server tomcat]# find / -name context.xml
/opt/tomcat/conf/context.xml
/opt/tomcat/webapps/docs/META-INF/context.xml
/opt/tomcat/webapps/examples/META-INF/context.xml
/opt/tomcat/webapps/host-manager/META-INF/context.xml
/opt/tomcat/webapps/manager/META-INF/context.xml
[root@tomcat-server tomcat]#
```

```
#comment value tag sections in below all files
[root@tomcat-server bin]# vim /opt/tomcat/webapps/examples/META-INF/context.xml
[root@tomcat-server bin]# vim /opt/tomcat/webapps/host-manager/META-INF/context.xml
[root@tomcat-server bin]# vim /opt/tomcat/webapps/manager/META-INF/context.xml
<!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
      allow="127.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:1" /> -->
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Licensed to the Apache Software Foundation (ASF) under one or more
    contributor license agreements. See the NOTICE file distributed with
    this work for additional information regarding copyright ownership.
    The ASF licenses this file to You under the Apache License, Version 2.0
    (the "License"); you may not use this file except in compliance with
    the License. You may obtain a copy of the License at
        http://www.apache.org/licenses/LICENSE-2.0
    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
-->
<Context>
    <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
        sameSiteCookies="strict" />
    <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:1" /> -->
</Context>
~
```

Now we need to change tomcat user configuration info:

The screenshot shows three terminal windows side-by-side:

- Terminal 1 (Left):** Shows the command `cd ..` followed by navigating to the `conf` directory and opening `tomcat-users.xml` with `vim`.
- Terminal 2 (Middle):** Shows the contents of the `tomcat-users.xml` file. It includes XML declarations, schema information, and a note about built-in manager roles (manager-gui, manager-script, manager-jmx, manager-status). It also contains comments about sample user entries and their intended use.
- Terminal 3 (Right):** Shows the XML configuration for the users, including entries for `admin` and `robot` with their respective roles and passwords.

now we need to reboot the tomcat server

The screenshot shows a terminal window with the following commands and output:

```
[root@tomcat-server conf]# cd ..
[root@tomcat-server bin]# cd bin/
[root@tomcat-server bin]# ./shutdown.sh
Using CATALINA_BASE: /opt/tomcat
Using CATALINA_HOME: /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME: /usr
Using CLASSPATH: /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
[root@tomcat-server bin]# ./startup.sh
Using CATALINA_BASE: /opt/tomcat
Using CATALINA_HOME: /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME: /usr
Using CLASSPATH: /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@tomcat-server bin]#
```

Below the terminal is a screenshot of a web browser showing the Jenkins dashboard at <http://15.152.159.145:8080/>. The URL bar also shows http://325b057e.isolation.zscaler.com/profile/b804a4b1-b2bc-43f9-8a10-833fbfb6660d/zia-session/?controls_id=62c05f8e-22cb-4be4-af4f-....

Integrating tomcat server with Jenkins:

there is no predefined tomcat plugin in Jenkins so we need to install "deploy to container plugin" then configure the tomcat server with credentials

manage genkins -> plugins -> available plugins -> deploy to container

The screenshot shows the Jenkins 'Available plugins' page. A search bar at the top right contains the text 'deploy to container'. Below it, a card for the 'Deploy to container' plugin is displayed, which is version 1.16 and was released 3 years and 10 months ago. The plugin is described as allowing deployment to a container after a successful build, specifically for Glassfish 3.x remote deployment. There is an 'Install' button at the top right of the card.

now we need to add credentials of tomcat to the Jenkins
manage Jenkins -> credentials -> system -> global credentials -> add
credentials(<user username="deployer" password="deployer" roles="manager-script"/>) id and description=tomcat-credentials

The screenshot shows the 'New credentials' page under 'Manage Jenkins > Credentials > System > Global credentials (unrestricted)'. The 'Kind' dropdown is set to 'Username with password'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc.)'. The 'Username' field contains 'deployer'. A checkbox for 'Treat username as secret' is unchecked. At the bottom is a 'Create' button.

The screenshot shows the 'Global credentials (unrestricted)' page. It lists a single credential: 'manager-script' (Type: Username with password, Name: deployer/***** (manager-script), Description: manager-script). A blue '+ Add Credentials' button is at the top right. Below the table are icons for sorting by ID, Name, Kind, and Description, and a link to change the page size (Icon: S M L).

then go to dashboard and create new item name as BuildAndDeployToTomcat as a maven project

add git repository http url

make branch main

and at the last click on "Add post-build action" and select "Deploy war/ear to a container"

in war/ear file add "*/*.war"

add container Tomcat 8.x

add tomcat-server instance url

save and apply

The screenshot shows two consecutive screenshots of the Jenkins configuration interface for a job named 'BuildAndDeployToTomcat'.

Source Code Management (Top Screenshot):

- The 'Source Code Management' tab is selected.
- The 'Git' option is chosen.
- The 'Repository URL' field contains the value `https://github.com/VivekNere1/Deployment.git`.
- The 'Credentials' dropdown is set to '- none -'.
- Buttons at the bottom: 'Save' (blue) and 'Apply'.

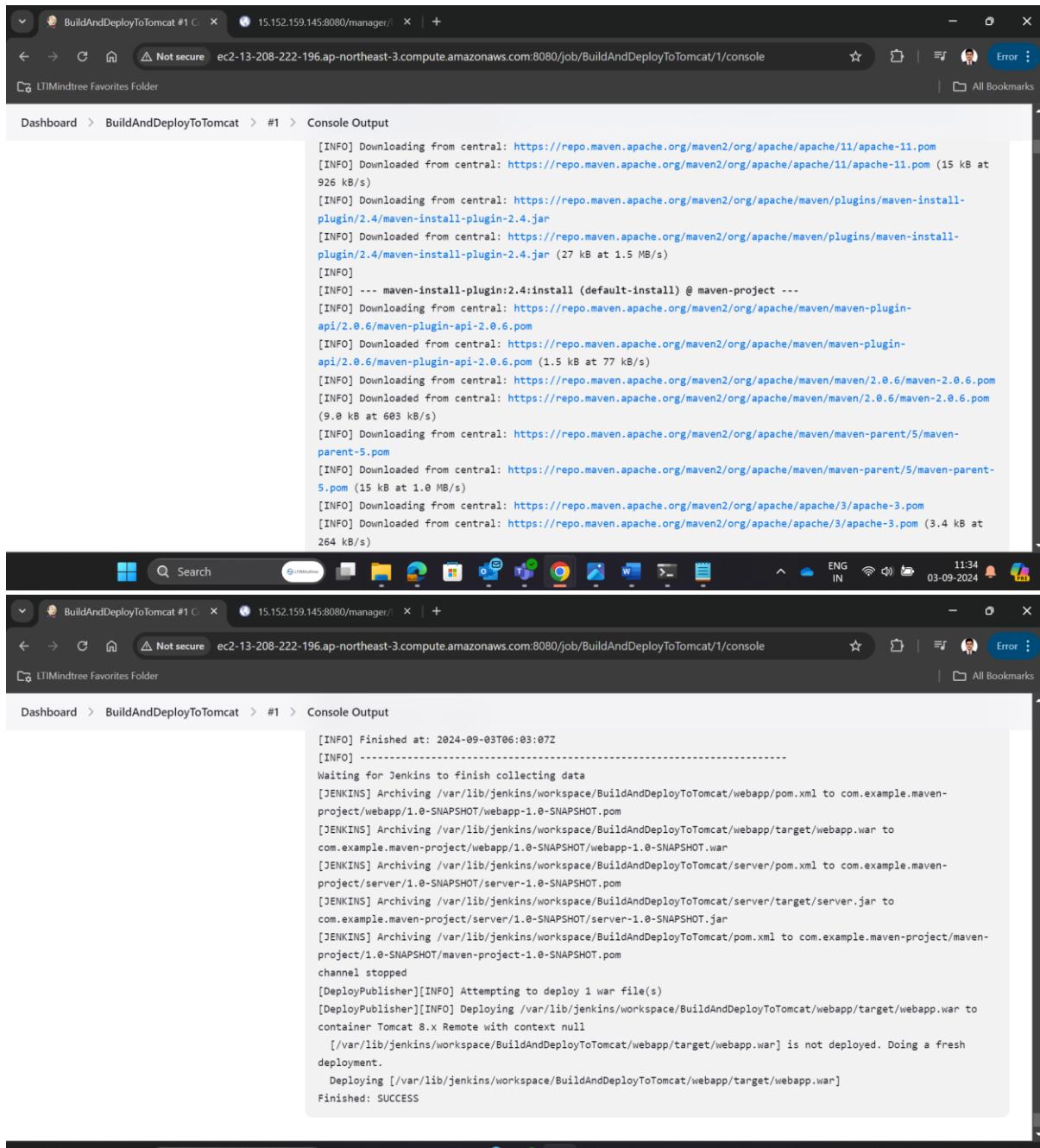
Post-build Actions (Bottom Screenshot):

- The 'Post-build Actions' tab is selected.
- A single action is defined: 'Deploy war/ear to a container'.
- The 'WAR/EAR files' field contains `*/*.war`.
- The 'Context path' field is empty.
- The 'Containers' section shows 'Tomcat 8.x Remote'.
- The 'Credentials' dropdown is set to '- none -'.
- Buttons at the bottom: 'Save' (blue) and 'Apply'.

The screenshot shows the Jenkins configuration interface for a job named 'BuildAndDeployToTomcat'. On the left, a sidebar lists various configuration sections: General, Source Code Management, Build Triggers, Build Environment, Pre Steps, Build, Post Steps, and Build Settings. The main panel is titled 'Containers' and contains a section for 'Tomcat 8.x Remote'. It includes a dropdown for 'Credentials' set to 'deployer/******** (manager-script)', a 'Tomcat URL' input field containing 'http://15.152.159.145:8080', and an 'Advanced' dropdown. A button labeled 'Add Container' is visible at the bottom.

We have successfully deployed project on tomcat server

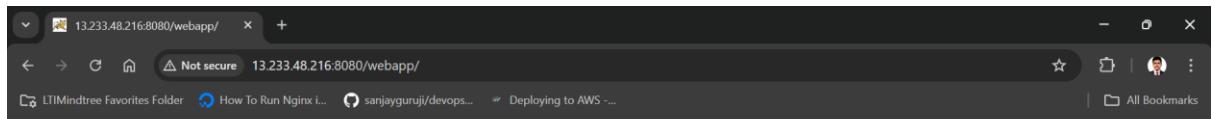
The screenshot shows the Jenkins console output for build #1 of the 'BuildAndDeployToTomcat' job. The left sidebar shows links for Status, Changes, Console Output (which is selected), Edit Build Information, Delete build '#1', Timings, Git Build Data, Redeploy Artifacts, Test Result, and See Fingerprints. The right pane displays the console logs, which include the command to clone a GitHub repository: `git clone https://github.com/VivekNere1/Deployment.git`. The logs also show the execution of Jenkins scripts and the deployment of artifacts to a Tomcat server.



The screenshot shows two stacked windows of a web browser displaying Jenkins console output. Both windows have the URL `ec2-13-208-222-196.ap-northeast-3.compute.amazonaws.com:8080/job/BuildAndDeployToTomcat/1/console`. The top window shows the initial Maven dependency download phase, while the bottom window shows the final deployment phase where the war file is archived and deployed to Tomcat.

```
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/11/apache-11.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/11/apache-11.pom (15 kB at 926 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-install-plugin-2.4.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-install-plugin-2.4.jar (27 kB at 1.5 MB/s)
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ maven-project ---
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.pom (1.5 kB at 77 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven/2.0.6/maven-2.0.6.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven/2.0.6/maven-2.0.6.pom (9.0 kB at 603 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/5/maven-parent-5.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/5/maven-parent-5.pom (15 kB at 1.0 MB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/3/apache-3.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/3/apache-3.pom (3.4 kB at 264 kB/s)

[INFO] Finished at: 2024-09-03T06:03:07Z
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/BuildAndDeployToTomcat/webapp/pom.xml to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/BuildAndDeployToTomcat/webapp/target/webapp.war to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/BuildAndDeployToTomcat/server/pom.xml to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/BuildAndDeployToTomcat/server/target/server.jar to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/BuildAndDeployToTomcat/pom.xml to com.example.maven-project/1.0-SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
channel stopped
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/BuildAndDeployToTomcat/webapp/target/webapp.war to container Tomcat 8.x Remote with context null
[/var/lib/jenkins/workspace/BuildAndDeployToTomcat/webapp/target/webapp.war] is not deployed. Doing a fresh deployment.
Deploying [/var/lib/jenkins/workspace/BuildAndDeployToTomcat/webapp/target/webapp.war]
Finished: SUCCESS
```



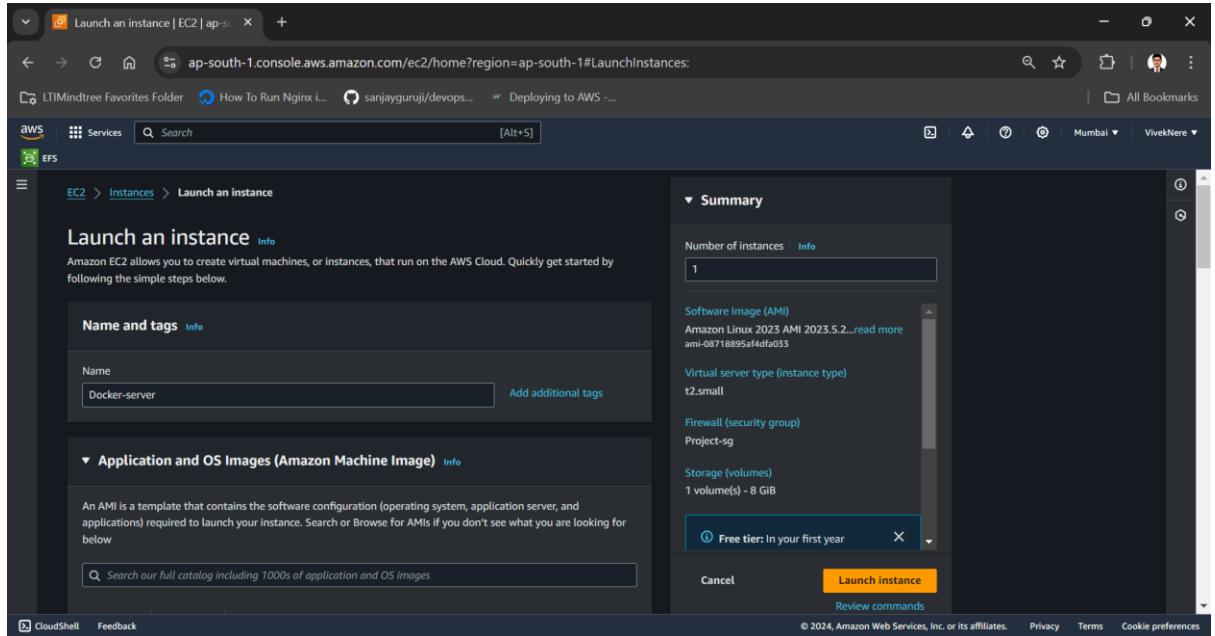
Thank You, Happy Learning!

See You Again

#4: Docker

Docker Installation and Configuration with Jenkins :

First create an instance for docker using amazon linux as OS with instance type t2.micro and same key-pair and security group which we have used in previous instances.



The screenshot shows the AWS EC2 Launch Instance wizard. The first step, "Configure instance details", is completed. The second step, "Configure storage", is currently being configured. The user has selected an 8 GiB gp3 root volume. A tooltip indicates that free-tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. The third step, "Review and launch", is visible at the bottom.

Now connect this docker ec2 in terminal and give hostname as docker-server to it.

```

Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\10748090>cd Downloads

C:\Users\10748090\Downloads>ssh -i "Project-Key-Pair.pem" ec2-user@ec2-65-0-104-175.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-65-0-104-175.ap-south-1.compute.amazonaws.com (65.0.104.175)' can't be established.
ED25519 key fingerprint is SHA256:dhRh9NqzEw9rnSyjqkt3fT3k85cMHfJbf2LICpfssH0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-65-0-104-175.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

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

[ec2-user@ip-172-31-33-238 ~]$ sudo su -
[root@ip-172-31-33-238 ~]# hostnamectl set-hostname docker-server.example.com
[root@ip-172-31-33-238 ~]# bash
[root@docker-server ~]# |

```

Now we have to install docker in docker-instance.

```

[root@docker-server ~]# yum install -y yum-utils
Last metadata expiration check: 0:09:04 ago on Sat Sep 21 03:43:47 2024.
Package dnf-utils=4.3.0-13.amzn2023.0.4.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@docker-server ~]# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
[root@docker-server ~]# yum install docker -y
Docker CE Stable - x86_64                                         343 B/s | 397 B    00:01
Errors during downloading metadata for repository 'docker-ce-stable':
 - Status code: 404 for https://download.docker.com/linux/centos/2023.5.20240916/x86_64/stable/repo/repodata/repomd.xml (IP: 18.239.111.1
02)
Error: Failed to download metadata for repo 'docker-ce-stable': Cannot download repomd.xml: Cannot download repodata/repomd.xml: All
mirrors were tried
Ignoring repositories: docker-ce-stable
Last metadata expiration check: 0:09:33 ago on Sat Sep 21 03:43:47 2024.
Dependencies resolved.
=====
Package           Architecture      Version       Repository      Size
=====
Installing:
docker            x86_64          25.0.6-1.amzn2023.0.2      amazonlinux   44 M
Installing dependencies:
containerd        x86_64          1.7.20-1.amzn2023.0.1      amazonlinux   35 M
iptables         x86_64          1.8.8-3.amzn2023.0.2      amazonlinux   401 k
iptables-nft    x86_64          1.8.8-3.amzn2023.0.2      amazonlinux   183 k
libcgroup         x86_64          3.0-1.amzn2023.0.1      amazonlinux   75 k
libnetfilter_conntrack x86_64        1.0.8-2.amzn2023.0.2      amazonlinux   58 k
libnftnl          x86_64          1.0.1-19.amzn2023.0.2     amazonlinux   30 k
libnftnl          x86_64          1.2.2-2.amzn2023.0.2     amazonlinux   84 k
pigz              x86_64          2.5-1.amzn2023.0.3      amazonlinux   83 k
runc              x86_64          1.1.13-1.amzn2023.0.1     amazonlinux   3.2 M

```

```

root@tomcat-server:~ root@jenkins-server:~ root@ip-172-31-33-238:~ Command Prompt
Installing : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 4/10
Installing : libnftnlink-1.0.1-19.amzn2023.0.2.x86_64 5/10
Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Installing : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 7/10
Installing : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 8/10
Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 8/10
Installing : libcgroup-3.0-1.amzn2023.0.1.x86_64 9/10
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Installing : docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

Verifying : containerd-1.7.20-1.amzn2023.0.1.x86_64 1/10
Verifying : docker-25.0.6-1.amzn2023.0.2.x86_64 2/10
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 3/10
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 4/10
Verifying : libcgroup-3.0-1.amzn2023.0.1.x86_64 5/10
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying : libnftnl-1.0.1-19.amzn2023.0.2.x86_64 7/10
Verifying : libnftnlink-1.2.2-2.amzn2023.0.2.x86_64 8/10
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64 9/10
Verifying : runc-1.1.13-1.amzn2023.0.1.x86_64 10/10

Installed:
containerd-1.7.20-1.amzn2023.0.1.x86_64 docker-25.0.6-1.amzn2023.0.2.x86_64 iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 libcgroup-3.0-1.amzn2023.0.1.x86_64 libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
libnftnlink-1.0.1-19.amzn2023.0.2.x86_64 libnftnl-1.2.2-2.amzn2023.0.2.x86_64 pigz-2.5-1.amzn2023.0.3.x86_64
runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
[root@docker-server ~]# docker --version
Docker version 25.0.5, build 5dc9bcc
[root@docker-server ~]#

```

Now aws configure in docker-server

```

root@tomcat-server:~ root@jenkins-server:~ root@docker-server:~ 
[root@docker-server ~]# aws configure
AWS Access Key ID [None]: AKIAQXUIX2SQABC62THV
AWS Secret Access Key [None]: 7BrQ/GICMlG7W3xg7/oRKByzpNaRHaWbyJhijwk
Default region name [None]: ap-south-1
Default output format [None]: table
[root@docker-server ~]# systemctl restart sshd
[root@docker-server ~]# systemctl enable sshd
[root@docker-server ~]#

```

Now we have to set password to the docker-instance and give password related permissions in `sshd_config` file.

```

root@tomcat-server:~ root@jenkins-server:~ root@docker-server:~ 
[root@docker-server ~]# 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@docker-server ~]# vim /etc/ssh/sshd_config
[root@docker-server ~]# 

# 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 yes

# Change to no to disable s/key passwords

```

42,15 35%

Now we have to created ssh key in docker-instance.

```
[root@docker-server ~]# ssh-keygen
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 fingerprint is:
SHA256:gQRoZI0b6IGZbHe0BF6FRTMnFHQJ2J5GQdoTFNg7GpU root@docker-server.example.com
The key's randomart image is:
+---[RSA 3072]---+
|o=o+=+=/*+o.
|==*.*++EB.
|o.+o=o+*
| .. . *..
| + S
|
|
|
+---[SHA256]---+
[root@docker-server ~]# cd .ssh
[root@docker-server .ssh]# ll
total 12
-rw----- 1 root root 564 Sep 21 03:43 authorized_keys
-rw----- 1 root root 2622 Sep 21 04:00 id_rsa
-rw-r--r-- 1 root root 584 Sep 21 04:00 id_rsa.pub
[root@docker-server .ssh]# |
```

Now we need to transfer this docker ssh-key in jenkins-server.

```
[root@docker-server .ssh]# systemctl restart sshd
[root@docker-server .ssh]# ssh-copy-id root@172.31.41.115
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.41.115 (172.31.41.115)' can't be established.
ED25519 key fingerprint is SHA256:F31F2026wQ8ZUs+HtPINK0auCQgLFwvtGkxCxsZkSQE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.41.115's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.41.115'"
and check to make sure that only the key(s) you wanted were added.

[root@docker-server .ssh]# |
```



```
[root@tomcat-server:~] x [root@jenkins-server:~] x [root@docker-server:~/ssh] x + - 
[root@jenkins-server ~]# cd .ssh
[root@jenkins-server .ssh]# ll
total 20
-rw----- 1 root root 1169 Sep 21 04:04 authorized_keys
-rw----- 1 root root 2622 Sep 20 03:42 id_rsa
-rw-r--r-- 1 root root 585 Sep 20 03:42 id_rsa.pub
-rw----- 1 root root 810 Sep 20 07:24 known_hosts
-rw----- 1 root root 635 Sep 20 07:22 known_hosts.old
[root@jenkins-server .ssh]# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQBg9Ch9LyLzd0Tq3EIwpKm7gQ04nb5TbCyqQa16y7aYYh03vv21s0nmXtQR8doqTMSeI89o5fB+1CPj/WM1eHFFJ2MVFoOR/zTbsjCrnvZgZQ0mhojFPRoU/rL4KhG2fKtEdax1XUvX0FnwONB/TfWVcLjBvVwzluCvavYbjomovQLlgv7VnZl3ANUpjG056zNB35hxOxMPoWLfuBScveqgaXgZDAAo0ux+Fzrp078NGnjyjcgLOjih+XeoItLSR0axMxSYWcKQ0PhpxbP5Kd11aFRUkFTMuURzc/yjYoMkTCc+7EvYkaPMPPyZnT0pLzD4Ef9IM0u3FULBrt4LamSVeG+QMRaEd5wYItwlzIOXoxQDLl2IujulVfq3akWNiTMxNCT5ckUq8w4VFkMgTkUQgn1875gSQqtu3Ad8pfEB7UzqRjyVq1/JxDv+nle0Et+6nTBqj5tkwqjY/Mtax9YDLKM/sk2XQGLTxzS04ak2Cw/09TFYdWdStPiEtE= root@jenkins-server.example.com
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQBg9ClbRXZP5ySi3y32e25EDZHfgqCjRWQj0X6UCK7YOKNLpnJ04LK/5NyQ6ttIwgZFMMN0+EoURMaRkZZk595zBSH3uJe2ECShh521JpqawX16k07p28hf1/xMvCh/hb3/kf6yaQ0JLkX/o6Kwf+B8IPtW1Deis5B5o/16coEfg27sB016VLPG8uv12kgqshj2LrM819RKXhsK1ZcdDxbh0ZzCTVvYp/hty/wk3pGtp2+Bdv>cGNTkMXReG/H4hPhdnr98PBxD0+GASmrC7L0YRq5k9JZvgVhMs2c/5a23jbyo/h3+Vci19E50tbt0+0ig4xy+/R0Kdf0ijXACM8ho7Rypejg/rH5kmwca7+HhpGVylqNtw1NDLEDVhPetkh0F/E2n02zU155maBybNckkLihkQsFuipI0Kwc3qprm6v930bwB3MBA6Ag0aSzrmypybdCw7CR5rvrC8jBr0xWjQA3BQ0I7y69+wPPhAyPLxQ811IGofm66DHskQffLTgvadk= root@docker-server.example.com
[root@jenkins-server .ssh]# |
```

pass the ssh-key of jenkins-server in docker-server.

```
[root@jenkins-server .ssh]# ssh-copy-id root@172.31.33.238
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.33.238 (172.31.33.238)' can't be established.
ED25519 key fingerprint is SHA256:dhRh9NqzEw9rnSyjqkt3fT3k85cMHfJbf2LICpfssH0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.33.238's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.33.238'"
and check to make sure that only the key(s) you wanted were added.

[root@jenkins-server .ssh]# |
```

```

root@tomcat-server:~          root@jenkins-server:~          root@docker-server:~/ssh
[root@docker-server .ssh]# cat authorized_keys
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\"';echo;sleep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAQABAAQDnD7pxRKId26Xfc18dMCvEQ8GfrxB4z31bZwF1ltG8PFL2FYNapPrHzGSvzEFS
hL9uB2BkEwwENUVON3HF1TS13kV3yZ6bIooIhPnf0uoF1rpZogBnyooSClyPVEiipmhZ60IsmWz3S4rNzqD2uDJE+bNljrdkp7VrEW5zyS1Rkoh+eT5keZQH0WatMsEhk
Tua824g+qI6Opj6JCBCpipqMcky00tSD7L4X/idCQGG1qPQXgNrU+fTzR+/rn24UN6BjCFeciPV0ql6GTMKbEQaKCnWBDJ2pd0vEWjGrFyCTKaLvmIEAOePi9WVHrlPjtlo14R
uxIbHLA/fPIfbT Project-Key-Pair
ssh-rsa AAAAB3NzaC1yc2EAAQABAAQDnD7pxRKId26Xfc18dMCvEQ8GfrxB4z31bZwF1ltG8PFL2FYNapPrHzGSvzEFS
bsjCrvnZgZQ0mh0jFPRoU/x14khG2fkTeDaXLXUvX0Fmw0NB/TfWYclJbvVwzluCavvYBvjomovQlgv17gVNz13ANUqpjG056zNB35ghx0XMp0lLfUBScceYqgXgZDA0aou
x+Fzpd70NGNjyjcGLQjh+XEoItLSR0aMxSYlcKQPphbxP5Kd1iAFRUtFTMnURzc/yjYoMkTCc+7EvYKaPMPyZmT0pLzD4Ef9IMOUo3fULBRt4lAmSveG+QMRaEd5wYItwLz
IOxoxQDL12IujuLFq3akWMiTmNCT5ckUq8w4VFKMgTKUQgn1875gSQqQtu3Ad8pfEB7UzqRjyVq1/JxDv+nleOEt+6nTBqi5tkqwjY/Mtax9YDLKM/sgK2XQGLTXizS04ak
2Cw/09TFYdWdstPiEte= root@jenkins-server.example.com
[root@docker-server .ssh]#

```

Now we have to create one ECR container to store our docker image.

The screenshot shows the AWS ECR console interface. At the top, there's a navigation bar with tabs for 'Services' and 'Search'. Below the navigation bar, the main content area has a heading 'Amazon Elastic Container Registry' with the subtext 'Share and deploy container software, publicly or privately'. On the right side, there's a large orange 'Create' button. To the left of the 'Create' button, there's a 'Create a repository' link. Below the main heading, there's a section titled 'How it works' with a diagram illustrating the workflow. Further down, there's a 'Pricing (US)' section with a note about pay-for-use pricing and a link to 'ECR pricing'. At the bottom of the main content area, there's a breadcrumb navigation: 'Amazon ECR > Private registry > Repositories > Create repository'. The bottom-most part of the screenshot shows the 'Create private repository' form. This form has two sections: 'General settings' and 'Encryption settings'. In the 'General settings' section, there's a 'Repository name' field containing '05072640160.dkr.ecr.ap-south-1.amazonaws.com/registration-app-ecr'. Below this, there's a 'Image tag immutability' section with two options: 'Mutable' (selected) and 'Immutable'. The 'Mutable' option is described as allowing image tags to be overwritten, while the 'Immutable' option prevents them from being overwritten. The 'Encryption settings' section is partially visible at the bottom.

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see [Getting Started with Amazon ECR](#).

1. Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS CLI:

```
aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin
050752640160.dkr.ecr.ap-south-1.amazonaws.com
```
2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:

```
docker build -t registration-app-ecr .
```
3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag registration-app-ecr:latest 050752640160.dkr.ecr.ap-south-1.amazonaws.com/registration-app-ecr:latest
```
4. Run the following command to push this image to your newly created AWS repository:

```
docker push 050752640160.dkr.ecr.ap-south-1.amazonaws.com/registration-app-ecr:latest
```

Now we have to configure our docker in jenkins for that we have make some changes in jenkins system.

Dashboard > Manage jenkins > System

In keys we have to paste the private key of jenkins

Path to key ?

Key ?

```
-----BEGIN OPENSSH PRIVATE KEY-----  
b3BlnNzC1rZxtdEAaaaABg5vbmUAAAEBm9uZQAAAAAAABAABlwAAAAdzc2gtn  
NHAAAAAwEAQAAAAYAfocf3c3Q06NxCMKSp4EE0i2+U3QspGKtes2mGlt779bJUSI  
7UEHak2zEnIPaOxwftQj4/JNWhxRsdfRaokf8027wq752/GUNj0toIx0afP65eCoR  
tn5E3g2j5V1L198ZsNDQf31mhCY71tMSbrgr2tAb46lqjLCSYL9e4FtcsdwDVkqYxq0e  
szQseaoCv5TDf6f37gu+nImKmh4GQqdmttLsfa6x6e9DRjYb3B0l0/xkjZUkdGjMU  
mfNnCKED6fW8t+SndygbUVLRUzJfE3Pb02KDfEwnPux8imjz08mZk9S8w+Bh/SDDIKN3  
1CwJbeQuikhnKEWHeCGLCc8NF6MUAySdlo71Rat2pRjkaMTQk+XfKwM0FRSj  
IESEII4o+YEKKKLbwtfXXaaEtM6kclafyC7/p9XhLupowauzbKslp2LwsfTwA  
yyP7IC0bgU14s0t0GtgP9PxWhVhUr4HLRAAFmBPrWCazfRAAAAB3NzC1y2  
EAAAAGBAK9H31N3NOoTcqCkpbuBBdvhN0Lpbpxrltph17e-/BWSeze1Bhx2ipMx  
4p2j0H7UJ-P9YzV4UcUnYwWg5H/NNyMu-kdnBfDsa6GMl9Gh+UxgEqDz+RN4NpeVd  
59IQWbDQ0H9N92wsluXDOW4K9g9G-Oia9auWxuBU3OKx15qmMjnRM0EnmqH7Y  
Ew+hYt+4Fx5ipoZe8kMAS57H4WvO13Q0Y2PKNwTCM5cSW2VfHroZjhZwpBA+nFv  
E/kp3WIAVSFSVMyZRHnZKnigjRM1z7s/pooW/lnzPSKwMPoB/0gw5sd9sFG3UczJ  
V4b5AxFr3nBg3AMjReFAmu604tWwdRv/MzE0P1yRsrzDhuuyBORRCelzv  
mbJpc27cB3j8QHTOpEnWx3nEO/6fV44S37qdMGqlm2SrCNj8y1rH1gMs0z+yArZda  
aVNeLNThorYLD/T1Mvh1Z1k0+ISQAAAAAMBAAEAAAGANjgjZtReuJxeOqeBPWoxesR  
-----END OPENSSH PRIVATE KEY-----
```

Save Apply

Add jenkins ssh server with hostname as ip and root username ans path as /root

The screenshot shows the Jenkins System configuration page. Under the 'SSH Servers' section, a new server named 'jenkins' is being configured. The fields filled are:

- Name: jenkins
- Hostname: 172.31.41.115
- Username: root
- Remote Directory: /root

A checkbox for 'Avoid sending files that have not changed' is unchecked. A 'Test Configuration' button is visible on the right.

Add docker ssh server with hostname as ip and root username ans path as /root

The screenshot shows the Jenkins System configuration page. Under the 'SSH Servers' section, a new server named 'docker' is being configured. The fields filled are:

- Name: docker
- Hostname: 172.31.33.238
- Username: root
- Remote Directory: /root

A checkbox for 'Avoid sending files that have not changed' is unchecked. A 'Test Configuration' button is visible on the right.

Apply and save.

Create job name as Registration-application-job.

Now select our job in jenkins and in 'Add post-build action' add 'Send build artifacts over SSH'

The screenshot shows the Jenkins job configuration page for 'Registration-application-Job'. The left sidebar lists configuration sections: General, Source Code Management, Build Triggers, Build Environment, Pre Steps, Build, Post Steps, Build Settings, and Post-build Actions. The 'General' section is selected.

General

- Description: Build and Deploy
- Enabled:
- Source Code Management: Git
- Repository URL: https://github.com/Vivekhere1/Project-Registration-application.git
- Credentials: none
- Advanced options: Discard old builds, GitHub project, This project is parameterized, Throttle builds, Execute concurrent builds if necessary.

Source Code Management

- Repository URL: https://github.com/Vivekhere1/Project-Registration-application.git
- Credentials: none
- Advanced options: Add Repository, Branches to build: 'main', Add Branch, Repository browser: (Auto), Additional Behaviours: Add.

Build Triggers

- Build whenever a SNAPSHOT dependency is built:

Save and Apply buttons are at the bottom.

The screenshot shows two stacked Jenkins job configuration pages. Both pages have a header bar with tabs, a search bar, and a bookmarks section.

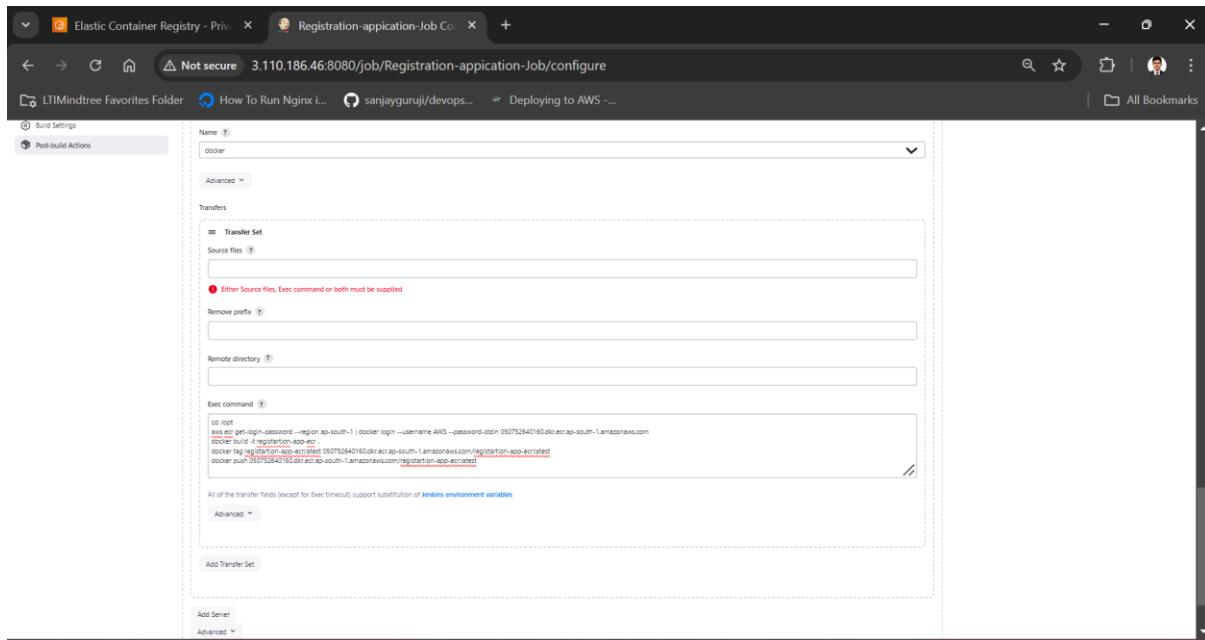
Top Configuration (Post-build Actions):

- Deploy war/ear to a container:**
 - WAR/EAR files: `**/*.war`
 - Context path: `/`
- Containers:**
 - Tomcat 8.x Remote:**
 - Credentials: `developer*****(manager-script)`
 - Add Container:
 - Tomcat URL: `http://13.235.48.216:8080`
 - Advanced:
- Send build artifacts over SSH:**
 - SSH Publishers: `Jenkins`

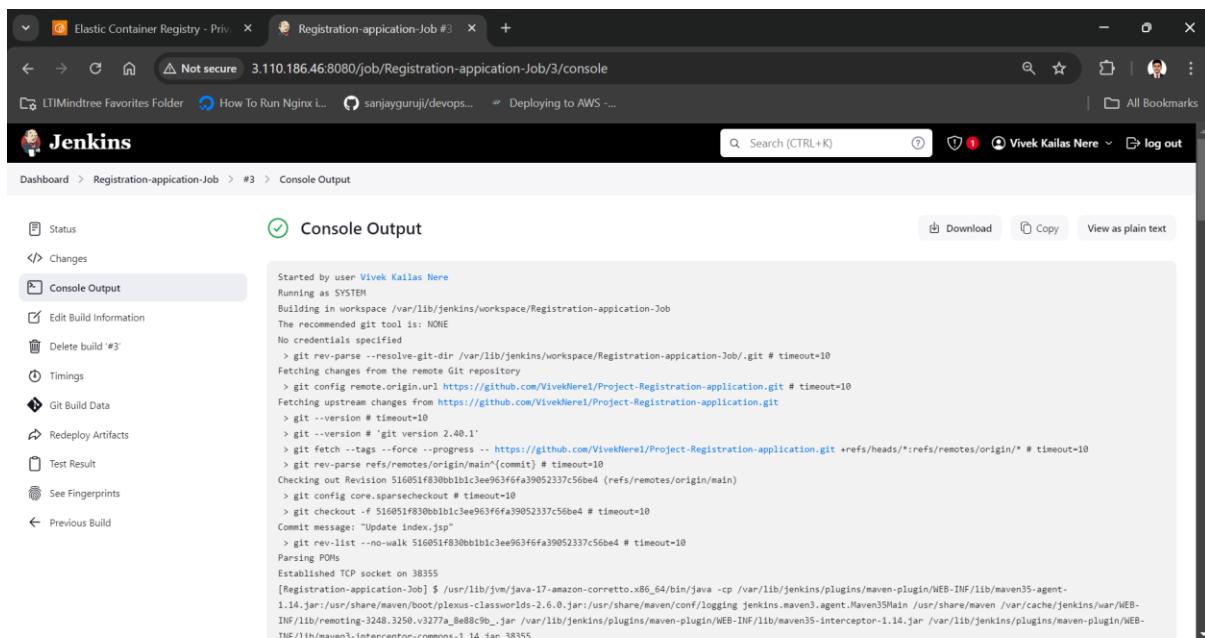
Bottom Configuration (Send build artifacts over SSH):

- SSH Server:**
 - Name: `Jenkins`
- Transfers:**
 - Transfer Set:**
 - Source files: `**/*` (with a red error message: **Either Source files, Exec command or both must be supplied**)
 - Remove prefix: `/var/lib/jenkins/workspace/Registration-application-Job/*`
 - Remote directory: `/opt`
 - Exec command: `ls -l /opt`
 - All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables
 - Advanced:

Action Buttons: Save, Apply



Appy and save.



```

[INFO] -----
[JENKINS] Archiving /var/lib/jenkins/workspace/Registration-application-Job/webapp/pom.xml to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Registration-application-Job/webapp/target/webapp.war to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/Registration-application-Job/server/pom.xml to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Registration-application-Job/server/target/server.jar to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/Registration-application-Job/pom.xml to com.example.maven-project/maven-project/1.0-SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
channel stopped
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/Registration-application-Job/webapp/target/webapp.war to container Tomcat 8.x Remote with context null
Redeploying [/var/lib/jenkins/workspace/Registration-application-Job/webapp/target/webapp.war]
Undeploying [/var/lib/jenkins/workspace/Registration-application-Job/webapp/target/webapp.war]
Deploying [/var/lib/jenkins/workspace/Registration-application-Job/webapp/target/webapp.war]
SSH: Connecting from host [jenkins-server.example.com]
SSH: Connecting with configuration [jenkins] ...
SSH: EXEC: completed after 3,203 ms
SSH: Disconnecting configuration [jenkins] ...
SSH: Transferred 0 file(s)
SSH: Connecting from host [jenkins-server.example.com]
SSH: Connecting with configuration [docker] ...
SSH: EXEC: completed after 21,015 ms
SSH: Disconnecting configuration [docker] ...
SSH: Transferred 0 file(s)
Finished: SUCCESS

```

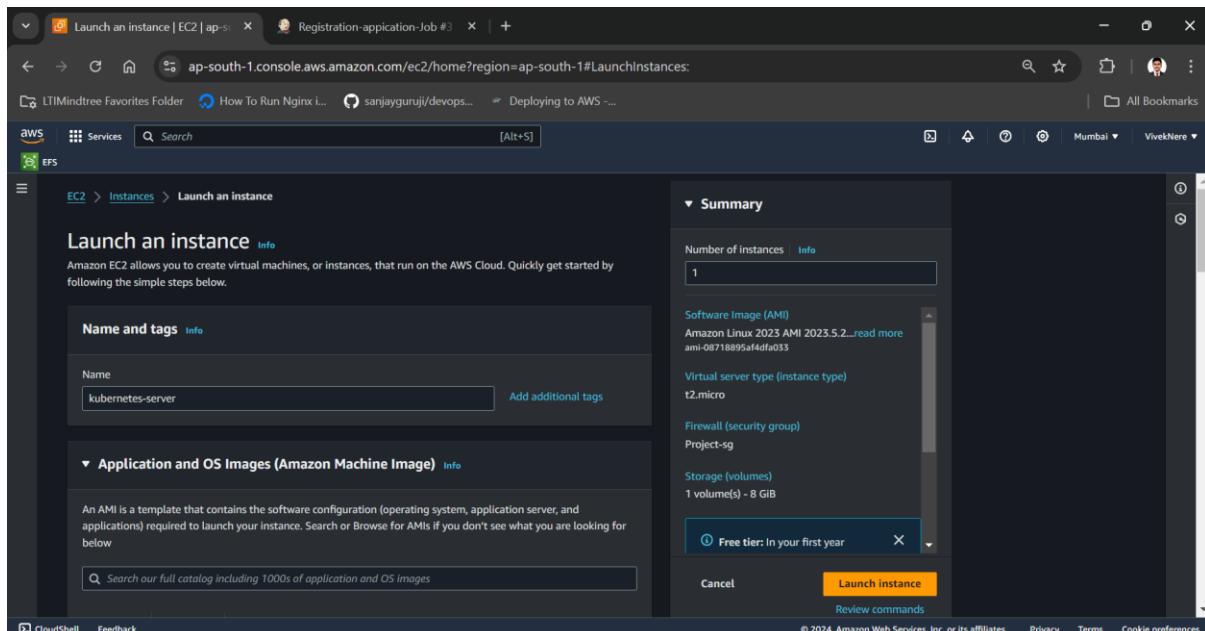
REST API Jenkins 2.462.2

From above we can see our docker build get successful!

#5: Kubernetes EKS

kubernetes Installation and Configuration with jenkins:

First create an instance for k8s using amazon linux as OS with instance type t2.micro and same key-pair and security group which we have use in previous instances.



```

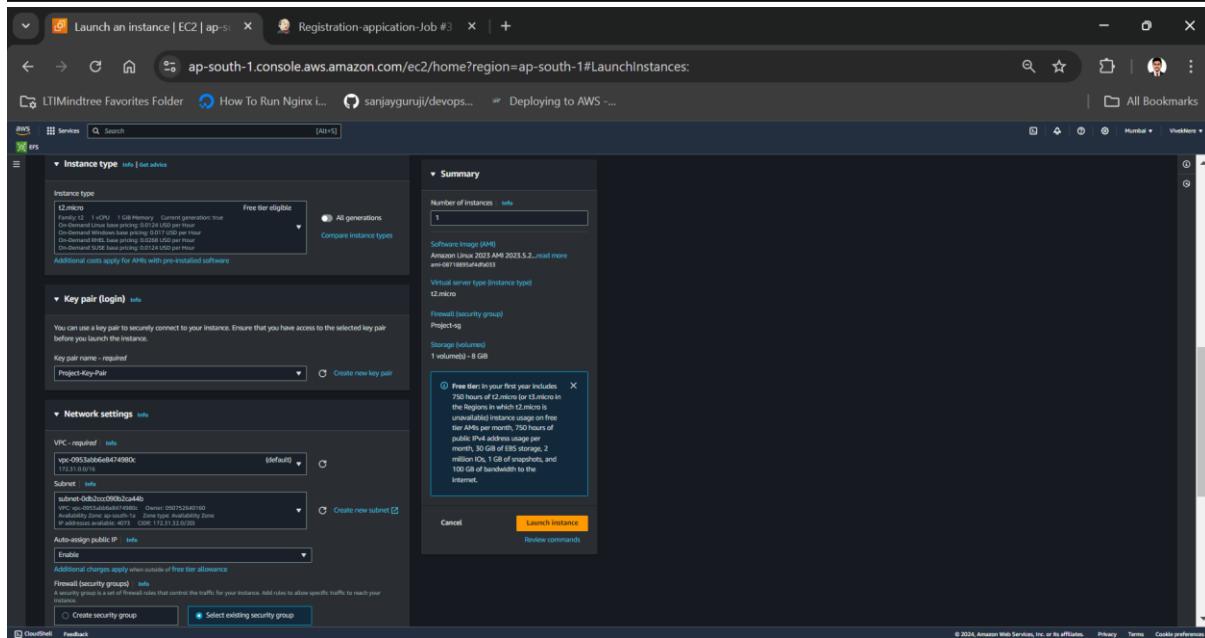
root@tomcat-server:~ root@jenkins-server:~ root@docker-server:~ root@ip-172-31-41-52:~ + - X
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

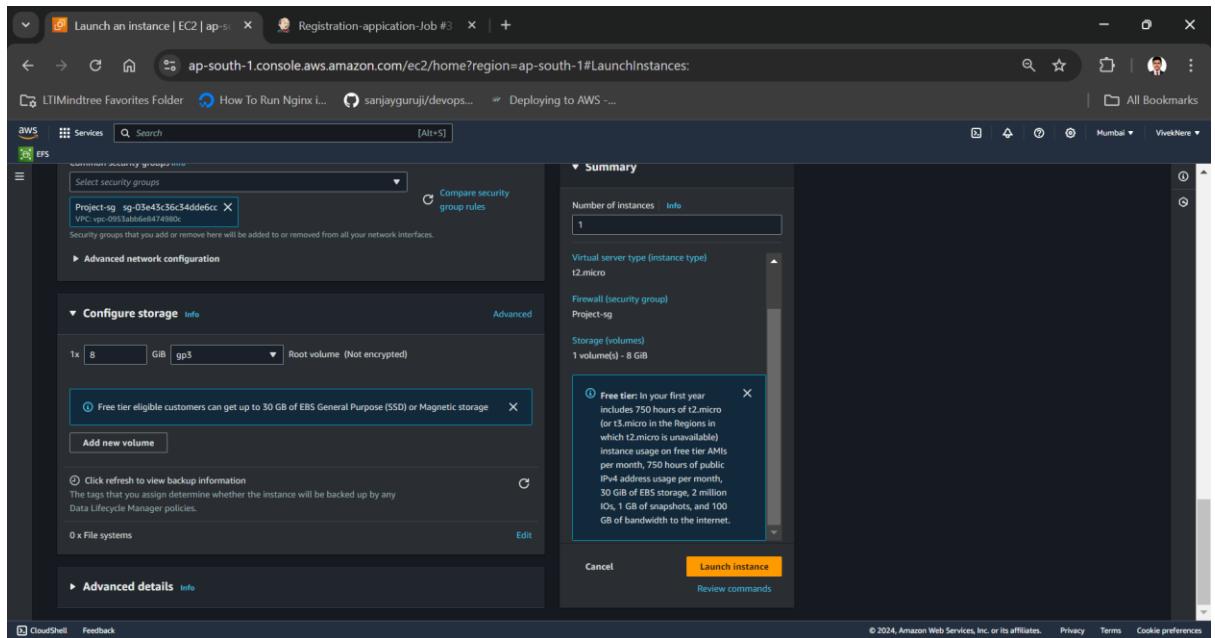
C:\Users\10748090>cd Downloads

C:\Users\10748090\Downloads>ssh -i "Project-Key-Pair.pem" ec2-user@ec2-13-201-25-170.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-201-25-170.ap-south-1.compute.amazonaws.com (13.201.25.170)' can't be established.
ED25519 key fingerprint is SHA256:cifAz1IDh2pMgpGVCLC+8WAF8IOoyyDP7Vu1gjUIuMg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-201-25-170.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

      _#
     /_###_          Amazon Linux 2023
    /_####\_
   \##|_
  /#/ __- https://aws.amazon.com/linux/amazon-linux-2023
  \/_\ \_/\_/
   \/_\ \_/
  /_m/_/ [ec2-user@ip-172-31-41-52 ~]$ sudo su -
[root@ip-172-31-41-52 ~]# hostnamectl set-hostname k8s-server.example.com
[root@ip-172-31-41-52 ~]# bash
[root@k8s-server ~]# 

```





Now connect this kubernetes ec2 in terminal and give hostname as docker-server to it.

```

root@tomcat-server:~ x root@jenkins-server:~ x root@docker-server:~ x root@ip-172-31-41-52:~ x + -
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\10748090>cd Downloads

C:\Users\10748090\Downloads>ssh -i "Project-Key-Pair.pem" ec2-user@ec2-13-201-25-170.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-201-25-170.ap-south-1.compute.amazonaws.com (13.201.25.170)' can't be established.
ED25519 key fingerprint is SHA256:cifAziIDh2pMgpGVCLC+8WAF8IOyyDP7Vu1gjUIuMg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-201-25-170.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

#_
\_ _###_      Amazon Linux 2023
\_ _##_##\_
\##|_
#/ _-- https://aws.amazon.com/linux/amazon-linux-2023
\/_/ \_>
\_/_ \_/
\_/_ \_/
\_/_ \_/
[ec2-user@ip-172-31-41-52 ~]$ sudo su -
[root@ip-172-31-41-52 ~]# hostnamectl set-hostname k8s-server.example.com
[root@ip-172-31-41-52 ~]# bash
[root@k8s-server ~]#

```

Create EKS cluster in kubernetes-server

Create one IAM role with ECR fullaccess , EKSpolicy and IAM full access policy and attach that role to this kubernetes instance.

The screenshot shows two tabs in a browser window. The top tab is titled 'Instances | EC2 | ap-south-1' and displays a list of EC2 instances. It shows three instances: 'kubernetes-server' (running, t2.medium, 2/2 checks passed), 'Jenkins-Server' (running, t2.micro, 2/2 checks passed), and 'Developer-server' (stopped, t2.micro). The bottom tab is titled 'Modify IAM role | EC2 | ap-south-1' and shows the 'Modify IAM role' configuration page for the 'kubernetes-server' instance. It lists the attached IAM role 'eks-roles' and provides options to 'Create new IAM role' or 'Update IAM role'.

Install aws cli in k8s ec2

```
[root@k8s-server ~]# yum update -y
Last metadata expiration check: 0:15:56 ago on Sat Sep 21 05:02:51 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@k8s-server ~]# yum install unzip -y
Last metadata expiration check: 0:16:18 ago on Sat Sep 21 05:02:51 2024.
Package unzip-6.0-57.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@k8s-server ~]# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total Spent   Left Speed
100  62.9M  100  62.9M    0      0  82.3M    0:--:-- --:--:--:--:-- 82.2M
[root@k8s-server ~]# unzip awscliv2.zip
Archive: awscliv2.zip
  creating: aws/
  creating: aws/dist/
  inflating: aws/install
  inflating: aws/README.md
  inflating: aws/THIRD_PARTY_LICENSES
  creating: aws/dist/awscli/
  creating: aws/dist/cryptography/
  creating: aws/dist/docutils/
  creating: aws/dist/lib-dynload/
  inflating: aws/dist/aws
  inflating: aws/dist/aws_completer
  inflating: aws/dist/libpython3.12.so.1.0
  inflating: aws/dist/_awsCRT.abi3.so
  inflating: aws/dist/_cffi_backend.cpython-312-x86_64-linux-gnu.so
  inflating: aws/dist/_ruamel_yaml.cpython-312-x86_64-linux-gnu.so
  inflating: aws/dist/libz.so.1

[root@k8s-server ~]# sudo ./aws/install
You can now run: /usr/local/bin/aws --version
[root@k8s-server ~]# aws configure
AWS Access Key ID [None]: AKIAQXUIX2SQABC62THV
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: table
[root@k8s-server ~]#
```

Install EKS tool:

```
[root@k8s-server ~]# curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
[root@k8s-server ~]# sudo mv /tmp/eksctl /usr/local/bin
[root@k8s-server ~]# eksctl version
0.190.0
[root@k8s-server ~]#
```

Install Kubectl:

```
[root@k8s-server ~]# curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kube-rnernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
% Total    % Received % Xferd  Average Speed   Time     Time   Current
          Dload  Upload Total   Spent    Left Speed
100 53.7M  100 53.7M    0     0  10.9M      0  0:00:04  0:00:04  --:--:-- 11.0M
[root@k8s-server ~]# sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
[root@k8s-server ~]# kubectl version --client
Client Version: v1.31.0
Kustomize Version: v5.4.2
[root@k8s-server ~]#
```

Now create cluster on EKS:

```
[root@k8s-server ~]# eksctl create cluster --name my-cluster --region ap-south-1 --version 1.29 --vpc-public-subnets subnet-0db2ccc090b2ca44b,subnet-02acccf4a1afdf91e6 --without-nodegroup
2024-09-21 05:39:51 [ℹ] eksctl version 0.190.0
2024-09-21 05:39:51 [ℹ] using region ap-south-1
2024-09-21 05:39:51 [ℹ] using existing VPC (vpc-0953abb6e84740980c) and subnets (private:map[] public:map[ap-south-1a:[subnet-0db2ccc090b2ca44b ap-south-1a:172.31.32.0/20 0 } ap-south-1b:[subnet-02acccf4a1afdf91e6 ap-south-1b:172.31.33.0/20 0 }])
2024-09-21 05:39:51 [ℹ] note: existing VPC/subnets will be used; if resulting cluster doesn't function as expected, make sure to review the configuration of VPC/subnets
2024-09-21 05:39:51 [ℹ] using Kubernetes version 1.29
2024-09-21 05:39:51 [ℹ] creating EKS cluster "my-cluster" in "ap-south-1" region with
2024-09-21 05:39:51 [ℹ] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=my-cluster'
2024-09-21 05:39:51 [ℹ] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "my-cluster" in "ap-south-1"
2024-09-21 05:39:51 [ℹ] CloudWatch logging will not be enabled for cluster "my-cluster" in "ap-south-1"
2024-09-21 05:39:51 [ℹ] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=my-cluster'
2024-09-21 05:39:51 [ℹ] default addons coredns, vpc-cni, kube-proxy were not specified, will install them as EKS addons
2024-09-21 05:39:51 [ℹ]

2 sequential tasks: [ create cluster control plane "my-cluster",
  2 sequential sub-tasks: [
    1 task: [ create addons ],
    wait for control plane to become ready,
  ]
]
2024-09-21 05:39:51 [ℹ] building cluster stack "eksctl-my-cluster-cluster"
2024-09-21 05:39:51 [ℹ] deploying stack "eksctl-my-cluster-cluster"
2024-09-21 05:40:21 [ℹ] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-21 05:40:51 [ℹ] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-21 05:41:21 [ℹ] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-21 05:41:52 [ℹ] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-21 05:43:52 [ℹ] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-21 05:44:52 [ℹ] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-21 05:45:52 [ℹ] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-21 05:46:52 [ℹ] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-21 05:47:52 [ℹ] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2024-09-21 05:47:52 [ℹ] creating addon
2024-09-21 05:47:52 [ℹ] successfully created addon
2024-09-21 05:47:53 [ℹ] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot 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'
2024-09-21 05:47:53 [ℹ] creating addon
2024-09-21 05:47:53 [ℹ] successfully created addon
2024-09-21 05:47:53 [ℹ] successfully created addon
2024-09-21 05:47:53 [ℹ] successfully created addon
2024-09-21 05:47:54 [ℹ] waiting for the control plane to become ready
2024-09-21 05:49:55 [ℹ] saved kubeconfig as "/root/.kube/config"
2024-09-21 05:49:55 [ℹ] no tasks
2024-09-21 05:49:55 [ℹ] all EKS cluster resources for "my-cluster" have been created
2024-09-21 05:49:55 [ℹ] created nodegroup(s) in cluster "my-cluster"
2024-09-21 05:49:55 [ℹ] created 0 managed nodegroups in cluster "my-cluster"
2024-09-21 05:49:56 [ℹ] kubelet command should work with "/root/.kube/config", try 'kubectl get nodes'
2024-09-21 05:49:56 [ℹ] EKS cluster "my-cluster" in "ap-south-1" region is ready
[root@k8s-server ~]# client_loop: send disconnect: Connection reset
```



```
[root@k8s-server ~]# kubectl config get-clusters
NAME
my-cluster.ap-south-1.eksctl.io
[root@k8s-server ~]#
```

Create ssh-keygen in k8s ec2:

```

[root@k8s-server ~]# ssh-keygen
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 fingerprint is:
SHA256:BLrGLWq2lPcFu8CWEpAk+e8kHH2M6PauE8Ld012g+MI root@k8s-server.example.com
The key's randomart image is:
+---[RSA 3072]----+
| . .
| +. .
| o. o.o .
| .+.ooooo
| .oooB.+ S
| ..*Oo= o .
| oO*B + o
| +.Eo* +
| o+o +
+---[SHA256]----+
[root@k8s-server ~]#

```

Now we have to create nodes in our cluster:

```

[root@k8s-server ~]# eksctl create nodegroup \
--cluster my-cluster \
--region ap-south-1 \
--name my-node-group \
--node-ami-family AmazonLinux2023 \
--node-type t2.small \
--nodes-min 2 \
--nodes-max 4 \
--ssh-access \
--ssh-public-key /root/.ssh/id_rsa.pub
2024-09-21 06:27:41 [ ] will use version 1.29 for new nodegroup(s) based on control plane version
2024-09-21 06:27:41 [ ] nodegroup "my-node-group" will use "" [AmazonLinux2023/1.29]
2024-09-21 06:27:41 [ ] using SSH public key "/root/.ssh/id_rsa.pub" as "eksctl-my-cluster-nodegroup-my-node-group-98:b4:0a:cf:f9:94:12:f8:44:70:c8:8c:3f:5e:82:d0"
2024-09-21 06:27:42 [ ] 1 nodegroup (my-node-group) was included (based on the include/exclude rules)
2024-09-21 06:27:42 [ ] will create a CloudFormation stack for each of 1 managed nodegroups in cluster "my-cluster"
2024-09-21 06:27:42 [ ] 2 sequential tasks: { fix cluster compatibility, 1 task: { create managed nodegroup "my-node-group" } }
2024-09-21 06:27:42 [ ] checking cluster stack for missing resources
2024-09-21 06:27:42 [ ] cluster stack has all required resources
2024-09-21 06:27:42 [ ] building managed nodegroup stack "eksctl-my-cluster-nodegroup-my-node-group"
2024-09-21 06:27:42 [ ] deploying stack "eksctl-my-cluster-nodegroup-my-node-group"
2024-09-21 06:27:42 [ ] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-node-group"
2024-09-21 06:28:12 [ ] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-node-group"
2024-09-21 06:28:44 [ ] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-node-group"
2024-09-21 06:30:32 [ ] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-node-group"
2024-09-21 06:30:33 [ ] no tasks
2024-09-21 06:30:33 [ ] created 0 nodegroup(s) in cluster "my-cluster"
2024-09-21 06:30:33 [ ] nodegroup "my-node-group" has 3 node(s)
2024-09-21 06:30:33 [ ] node "ip-172-31-12-236.ap-south-1.compute.internal" is ready
2024-09-21 06:30:33 [ ] node "ip-172-31-32-115.ap-south-1.compute.internal" is ready
2024-09-21 06:30:33 [ ] node "ip-172-31-5-174.ap-south-1.compute.internal" is ready
2024-09-21 06:30:33 [ ] waiting for at least 2 node(s) to become ready in "my-node-group"
2024-09-21 06:30:33 [ ] nodegroup "my-node-group" has 3 node(s)
2024-09-21 06:30:33 [ ] node "ip-172-31-12-236.ap-south-1.compute.internal" is ready
2024-09-21 06:30:33 [ ] node "ip-172-31-32-115.ap-south-1.compute.internal" is ready
2024-09-21 06:30:33 [ ] node "ip-172-31-5-174.ap-south-1.compute.internal" is ready
2024-09-21 06:30:33 [ ] created 1 managed nodegroup(s) in cluster "my-cluster"
2024-09-21 06:30:33 [ ] checking security group configuration for all nodegroups
2024-09-21 06:30:33 [ ] all nodegroups have up-to-date cloudformation templates
[root@k8s-server ~]#

```

```

[root@k8s-server ~]# kubectl get all
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes   ClusterIP  10.100.0.1 <none>        443/TCP   47m
[root@k8s-server ~]# kubectl get nodes
NAME                               STATUS   ROLES      AGE   VERSION
ip-172-31-12-236.ap-south-1.compute.internal   Ready   <none>   3m    v1.29.8-eks-a737599
ip-172-31-32-115.ap-south-1.compute.internal   Ready   <none>   2m59s  v1.29.8-eks-a737599
ip-172-31-5-174.ap-south-1.compute.internal   Ready   <none>   2m59s  v1.29.8-eks-a737599
[root@k8s-server ~]#

```

Now we have to create one deployment.yml file ans service.yml file

Give ssh-key and password related permission in sshd_config file of kubernetes-server.

```

Command Prompt      X  root@jenkins-server:~      X  root@docker-server:~      X  root@k8s-server:~      X  +  -  X
# 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 yes

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes

# Kerberos options
-- INSERT --

```

66,25 36%

Now we need to transfer this kubernetes ssh-key in jenkins-server And jenkins ssh-key in kubernetes-server

```

Command Prompt      X  root@jenkins-server:~      X  root@docker-server:~      X  root@k8s-server:~/ssh      X  +  -  X
[root@k8s-server ~]# cd .ssh
[root@k8s-server .ssh]# systemctl restart sshd
[root@k8s-server .ssh]# systemctl enable sshd
[root@k8s-server .ssh]# ssh-copy-id root@172.31.41.115
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.41.115' ('172.31.41.115') can't be established.
ED25519 key fingerprint is SHA256:F31F2026wQ8ZUs+HtPINK0auCQgLFwvtGKxCxsZkSQE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.41.115's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.41.115'"
and check to make sure that only the key(s) you wanted were added.

[root@k8s-server .ssh]#

```

```

Command Prompt      X  root@jenkins-server:~      X  root@docker-server:~      X  root@k8s-server:~/ssh      X  +  -  X
[root@jenkins-server .ssh]# systemctl restart sshd
[root@jenkins-server .ssh]# 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: enX0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 02:5e:63:15:d3:brd ff:ff:ff:ff:ff:ff
    altname eni-0fea0b51d24a37a7e
    altname device-number-0
    inet 172.31.41.115/20 metric 512 brd 172.31.47.255 scope global dynamic enX0
        valid_lft 3590sec preferred_lft 3590sec
    inet6 fe80::5e:63ff:fe15:51d3/64 scope link
        valid_lft forever preferred_lft forever
[root@jenkins-server .ssh]# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAADQABAAABgQCh9yLzd2dTqE3EIwPkM7gQ04nb5TdCyqQa16y7aYYh03v21s0nmXtQR8doqTMSeI89o5fB+1CPj/WM1eHFFJ2MVFoOR/zT
bsjCrnvZgZQ0mh0jFPRoU/l4khG2fkTedaXLXuvX0Fmw0NB/TfwLcJbvVwzlucCavvYBvjomovQLlgv17gVNz13ANUqpjg056zNBj5qhwx0xMPoWlfuBSceYqaGxgZDA0a0u
x+Fzrp70NGNjyjcGLQjh+XeoltlsR0amxSYwckQQphbxP5k1d1AFRuTfTmmlRzc/yjYomKTCc+7EvYkaPMpYzmtOpLzD4Ef9IMOUo3fULBr4LamsVeG+QMrAd5wYItwlz
IOXoqQDL2IujulVFq3akWMiTMxNCT5cklq8w4VFKMgTKU0gn1875gS0qqTu3Ad8pFEB7UzqRjyVq1/JxDv+n1eOEt+6nTBqj5tkqwjY/Mtax9YDLKM/sgK2xQGLTxzS04ak
2Cw/09TFvdWdstPiEtE= root@jenkins-server.example.com
ssh-rsa AAAAB3NzaC1yc2EAAADQABAAABgQClbRXZP5ySiY32e25ED2HFqgCjRNQJ0X6UCK7YOKNLpnJ04lk/5NyQ6ttIwgZFMN0+EoURMaRkZZL595zBSH3uJe2ECSH5
21JpqawX16kQ7Pz8Hfi/xMvCh/Hb3/kf6yaQ0JLK/xG6Kwf+B8IPtW1Dei5B5o/16coEFg27sB016VLPG8uV12kqShLj2LrM8ITi9RKXhsK1ZCdXhb0Z2CTVvYp/hty/wk3
pGtp2+BdvCGNTkVMXreG/H4hPhdr9BPBxDo+GAsmxCK7LOYRq5K9JzVmGsc2/Sa23Gbyo/h3+CiIE95Ortb0+0ig4xy+/R0kdFoijxACM8h0/7Ypejg/rH5kmca7+HpmG
YilgNtw1NDL8EDvhPtekh0F/E2No2zU15SmaByNckkLihkOsFuip10KwcS3qprm6v930wbw3MBA6Ag0aSzrmptybdCw7CR5rvrC8jBx0xVjqQA3BQ017y69+wPPhAyPlxQ8II1G
oFm66DHSkqOfflTgvadk= root@docker-server.example.com
ssh-rsa AAAAB3NzaC1yc2EAAAADQABAAABgQCvM5SS1XVqbhIIwEprJgHCL0mZgR/UEkGU3io+PRS3uo05YpU0T0yYekt30NiiloJHEM8VCnSw0TAMkRM4avpQtWsu1XuixP
ck2HWsA8gzb3do6aUZBQnlQczGvo3Eej22m1wEp9EagmAxinqbl0WHWxVsJnw2rN97clahzcflz/+k8A4kr2j9JYaqlfPBf1LokMKkIG5K3XLChnLe2AAiwp6gVFltd//xyC
nDV4lTcyx+/8o2aRvUwAoM59tP50vtPu2AKFJOZzGwPLcPfrVlm/lq5MKVzxqwpMjtcApbYDC7Am6KnjMoR4ayXSEw+6bRVM4fpfjFyufFsDg205IMDa38KXqcSXQeE35U2
TTHCYiu2WtS2b2+/UW11YmNN1laVyrx6sGkzPwAUQBb+liIPCZLjJ4SdGhVok/HL1bzQF1dXbjBDrb4a0z6FpgdDckvlfxTTGhAieLk5p3EvRtn4vAs8rxzsJA+yg8mf0DCo
fyy6r2m+EPWUz+he25E= root@k8s-server.example.com
[root@jenkins-server .ssh]#

```

```

[root@jenkins-server .ssh]# ssh-copy-id root@172.31.41.52
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.41.52's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.41.52'"
and check to make sure that only the key(s) you wanted were added.

[root@jenkins-server .ssh]# |
```



```

[root@k8s-server .ssh]# cat authorized_keys
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than th
e user \"root\".';echo;sleep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDnD7pxRKiD26Fc18dMCvE08GFRxB4z3TbwF11tG8Pf
L2FYNaPprHzGSvzEFShl9uB2BKEwwENUVON3Hf1TSL3kKY3yZ6bIooIlhPNF0uoIfIpZogBnyooSClyPVeiipmhZ601smwlz354rNzqD2uODJE+bNLjrdKp7VrEw5
zyS1RkOh+eT5keZQHWat0MsEhkTua824g+qI6OpJ6JCBCipqMCky0tSD7L4X/idCQGG1qPQXgNrU+fTzR+/rn24UN6BjCFeciPV0ql6GTMKbEQaKCNwBDJ2pd0
vEWjGrFyCTKaLvmIEAOgPi9WVhLPjtlo14RuxIbHLA/FPIfbT Project-Key-Pair
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCh9yLdzdDTqE3EIwpKm7gQ04nb5TdCyqQa16y7aYYh03vv21s0nmXtQR8doqTMSeI89o5fB+1CPj/wM1eHFFJ2
MFoOR/zTbsjCrvnZgZ00mh0jFPRoU/r4khG2fkTeDa1XUvX0Fmw0NB/TfWYcLJbvVwzlCuavYBvjomovQLglv17gVNz13ANUpjGo56zNBj5qxhxlOxMPowLf
uBSceYqaGxgZDA0a0ux+Frzpd70NGNjyjcGLQjH+XeoltLSR0aMxSYwckQOPphbxP5Kd1iaFRUTFTMuURzc/yjYoMkTcc+7EvyKaMPyZmT0plzD4Ef9IMOUo3fU
LBRT4lAmSVeG+QMRAEd5wYitwlzIOxoxQDLl2iujuLFq3akWmITMxNCT5ckUq8w4VFKMgTkUQgni875gSqqtu3Ad8pEB7UzqRJyVq1/JxDv+n1eEt+6nTBqi
5tkqwjY/Mtax9VDLKMs/gk2XQGLTXizS04ak2Cw/09TFYdWdStPiEtE= root@jenkins-server.example.com
[root@k8s-server .ssh]# |

```

Now we have to configure our kubernetes in jenkins for that we have make some changes in jenkins system.

Dashboard > Manage Jenkins > System

The screenshot shows the Jenkins 'System' configuration page under 'Manage Jenkins > System'. In the 'SSH Server' section, the 'Name' field is set to 'kubernetes', 'Hostname' is '172.31.41.52', 'Username' is 'root', and 'Remote Directory' is '/root'. A checkbox for 'Avoid sending files that have not changed' is checked. At the bottom, there are 'Save' and 'Apply' buttons, and a 'Test Configuration' button.

Now in our job **Registration-application-Job** we have to add ssh-server with exec commands of kubernetes to deploy and create service by Jenkins build.

The screenshot shows the Jenkins configuration interface for an EC2 instance. The left sidebar lists various build steps: General, Source Code Management, Build Triggers, Build Environment, Pre Steps, Build, Post Steps, Build Settings, and Post-build Actions. The 'Post-build Actions' section is currently selected. On the right, under 'SSH Server', the 'Name' field is set to 'kubernetes'. Under 'Transfers', a 'Transfer Set' is defined with 'Source files' and 'Remote directory'. The 'Exec command' field contains the Jenkinsfile snippet:

```
kubectl apply -f deployment.yaml  
kubectl apply -f service.yaml
```

. A note at the bottom states: "All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables". At the bottom of the configuration page are 'Save' and 'Apply' buttons.

Apply save and build now.

The screenshot shows the Jenkins console output for build #4. The left sidebar includes links for Status, Changes, Console Output (which is selected), Edit Build Information, Delete build '#4', Timings, Git Build Data, Redeploy Artifacts, Test Result, See Fingerprints, and Previous Build. The main area displays the build log, which starts with: "Started by user Vivek Kailas Nere", "Running as SYSTEM", and "Building in workspace /var/lib/jenkins/workspace/Registration-application-Job". It then details the git fetch and checkout process, followed by Maven commands like "mvn clean install -DskipTests" and finally "mvn -e". The log concludes with "Build step 'Invoke bat script' marked build as failure".

```

channel stopped
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/Registration-application-Job/webapp/target/webapp.war to container Tomcat 8.x Remote with context null
Redeploying [/var/lib/jenkins/workspace/Registration-application-Job/webapp/target/webapp.war]
Undeploying [/var/lib/jenkins/workspace/Registration-application-Job/webapp/target/webapp.war]
Deploying [/var/lib/jenkins/workspace/Registration-application-Job/webapp/target/webapp.war]
SSH: Connecting from host [jenkins-server.example.com]
SSH: Connecting with configuration [jenkins] ...
SSH: EXEC: completed after 3,203 ms
SSH: Disconnecting configuration [jenkins] ...
SSH: Transferred 0 file(s)
SSH: Connecting from host [jenkins-server.example.com]
SSH: Connecting with configuration [docker] ...
SSH: EXEC: completed after 3,203 ms
SSH: Disconnecting configuration [docker] ...
SSH: Transferred 0 file(s)
SSH: Connecting from host [jenkins-server.example.com]
SSH: Connecting with configuration [kubernetes] ...
SSH: EXEC: completed after 2,002 ms
SSH: Disconnecting configuration [kubernetes] ...
SSH: Transferred 0 file(s)
Finished: SUCCESS

```

REST API Jenkins 2.46.2

Our Kubernetes configuration with Jenkins build get successful!

Now we have to check in our Kubernetes-server whether service get created or not.

```

[root@k8s-server ~]# kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP
kubernetes   ClusterIP   10.100.0.1    <none>
443/TCP   67m
service-reg-app   LoadBalancer   10.100.15.151   a1bc97069991b4d72933cc5fc5737b04-2035678712.ap-south-1.elb.amazonaws.com
8080:32536/TCP  2m7s
[root@k8s-server ~]#

```

Our service get deployed successfully.

Now we have to check on chrome.



New user Register for DevOps Learning made by Vivek Nere for Milestone-2 project

Please fill in this form to create an account.

Enter Name	<input type="text"/>
Enter mobile	<input type="text"/>
Enter Email	<input type="text"/>
Password	<input type="password"/>
Repeat Password	<input type="password"/>

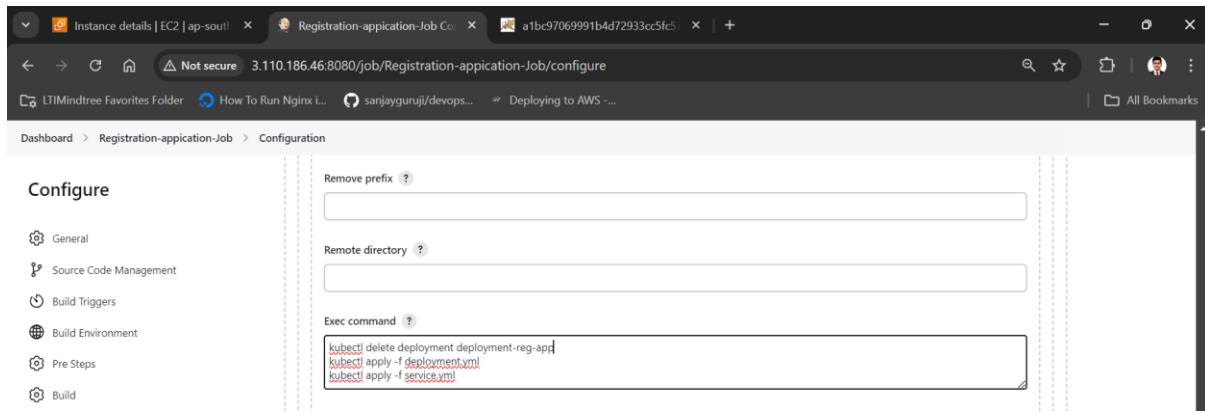
By creating an account you agree to our [Terms & Privacy](#).

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

Thank You, Happy Learning!

See You Again

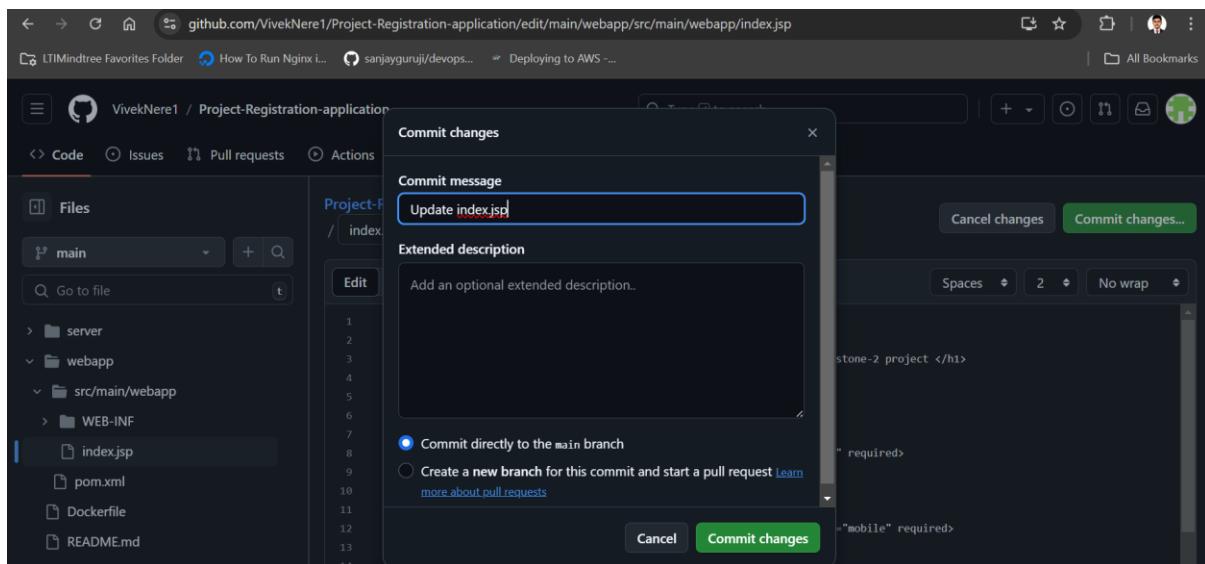
On every change in git we need to delete our old deployment and add new one so for that we have to add one more command in Kubernetes ssh-server exec commands



Now we have to made change in code

```
<h1> Thank You, Happy Learning for devops!</h1>
<h1> See You Again </h1>
```

After committing it will automatically reflect in service deployment.



The screenshot shows the Jenkins job details page for 'Registration-application-Job'. The top navigation bar includes tabs for 'Instance details | EC2 | ap-south' and 'Registration-application-Job [Job]'. The URL is '3.110.186.46:8080/job/Registration-application-Job/'. The main content area shows the build history with the last build (#7) being successful. It also displays a 'Latest Test Result (no failures)' badge and a 'Permalinks' section.

Build History:

- Last build (#7), 1.9 sec ago
- Last stable build (#6), 3 min 26 sec ago
- Last successful build (#6), 3 min 26 sec ago
- Last unstable build (#2), 2 hr 20 min ago
- Last unsuccessful build (#2), 2 hr 20 min ago
- Last completed build (#6), 3 min 26 sec ago

Our build is successful

The screenshot shows the Jenkins console output for build #7. The top navigation bar includes tabs for 'Instance details | EC2 | ap-south' and 'Registration-application-Job #7'. The URL is '3.110.186.46:8080/job/Registration-application-Job/7/console'. The Jenkins logo is visible in the header. The main content area shows the console output for build #7, which includes a successful git fetch and checkout from the GitHub repository.

```
Started by GitHub push by VivekNere1
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Registration-application-Job
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Registration-application-Job/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/VivekNere1/Project-Registration-application.git # timeout=10
Fetching upstream changes from https://github.com/VivekNere1/Project-Registration-application.git
> git -version # timeout=10
> git -version # 'git version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/VivekNere1/Project-Registration-application.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 1beafeed3e4f89026b241b6343175bb67be7b9318 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 1beafeed3e4f89026b241b6343175bb67be7b9318 # timeout=10
Commit message: "Update index.jsp"
> git rev-list --no-walk dd01b089ff98df7f830136e73c2ca1657f7271470c # timeout=10
Parsing POMs
```



New user Register for DevOps Learning made by Vivek Nere for Milestone-2 project

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, Happy Learning for devops!

See You Again

Changes get automatically reflected on server with the help of docker and Kubernetes integration with Jenkins hence we have successfully Created an end to end CI/CD pipeline in AWS platform using Jenkins as the orchestration tool, GitHub as the SCM, Maven as the Build tool, Deploy in a docker instance and create a Docker image, Store the docker image in ECR, Achieve Kubernetes deployment using the ECR image. Build a sample java web app using maven.