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, Achieve Kubernetes deployment using ECR image. Build sample java web app using maven.

Required tools:

- 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

Process:

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

- 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.

#Workflow for CI/CD Pipeline

1. Source Code Management (SCM)

- **Tool:** GitHub
- **Action:** Store the source code of the sample Java web application in a GitHub repository.

2. Continuous Integration (CI)

- **Tool:** Jenkins
- **Action:**
    - **Setup Jenkins:** Install Jenkins on an EC2 instance or use Jenkins on AWS.
    - **Integrate GitHub with Jenkins:** Use the GitHub plugin to connect Jenkins with your GitHub repository.
    - **Create Jenkins Pipeline:** Define a Jenkins pipeline using a Jenkins file stored in the GitHub repository.
    - **Build Trigger:** Configure Jenkins to trigger builds on code commits or pull requests.

3. Build Automation

- **Tool:** Maven
- **Action:**
  - **Define Build Steps:** In the Jenkins file, use Maven to compile the code, run tests, and package the application into a JAR/WAR file.

4. Containerization

- **Tool:** Docker
- **Action:**
  - **Create Docker file:** Write a Docker file to containerize the Java application.
  - **Build Docker Image:** Use Jenkins to build the Docker image from the Docker file.

5. Docker Image Storage

- **Tool:** Amazon Elastic Container Registry (ECR)
- **Action:**
  - **Create ECR Repository:** Set up a repository in ECR to store Docker images.
  - **Push Docker Image to ECR:** Use Jenkins to push the Docker image to the ECR repository.

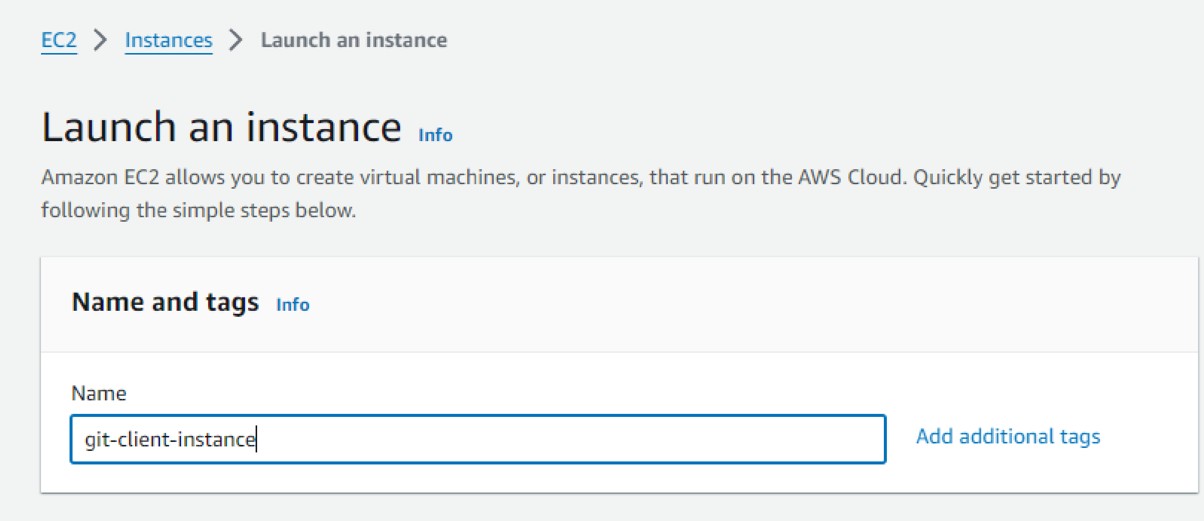
6. Continuous Deployment (CD)

- **Tool:** Kubernetes
- **Action:**
  - **Setup Kubernetes Cluster:** Use Amazon EKS (Elastic Kubernetes Service) to create a Kubernetes cluster.

- **Deploy Application:** Use Kubernetes manifests (YAML files) to deploy the application using the Docker image stored in ECR.

#Start with Git Client instance.

- Launch instance.



EC2 > Instances > Launch an instance

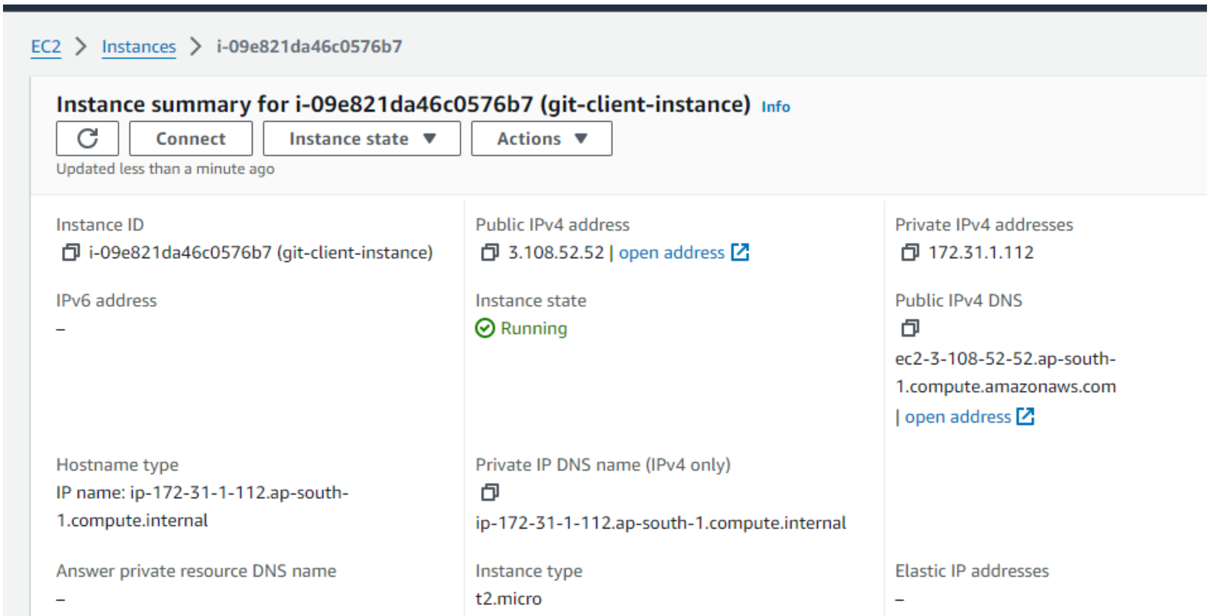
Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: git-client-instance

- Instance created.



EC2 > Instances > i-09e821da46c0576b7

Instance summary for i-09e821da46c0576b7 (git-client-instance) Info

Updated less than a minute ago

Instance ID: i-09e821da46c0576b7 (git-client-instance)

Public IPv4 address: 3.108.52.52 | open address

Private IPv4 addresses: 172.31.1.112

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-3-108-52-52.ap-south-1.compute.amazonaws.com | open address

Hostname type: IP name: ip-172-31-1-112.ap-south-1.compute.internal

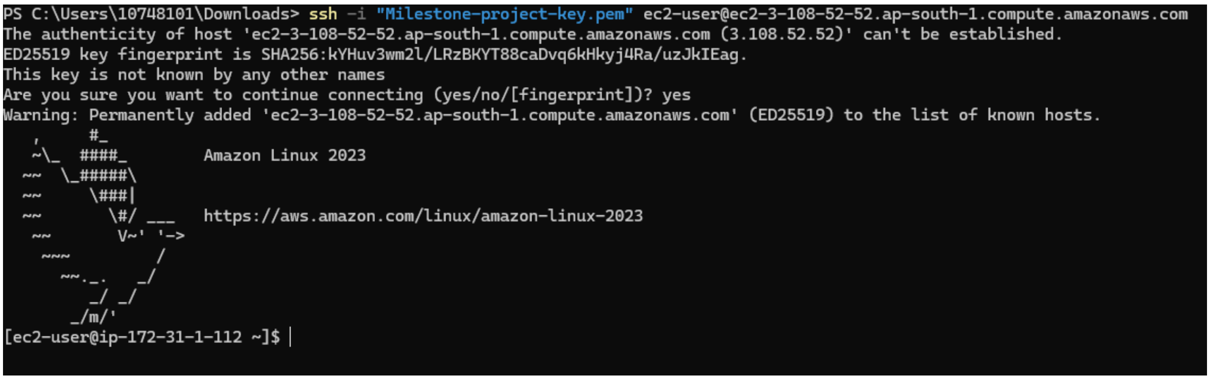
Private IP DNS name (IPv4 only): ip-172-31-1-112.ap-south-1.compute.internal

Answer private resource DNS name: –

Instance type: t2.micro

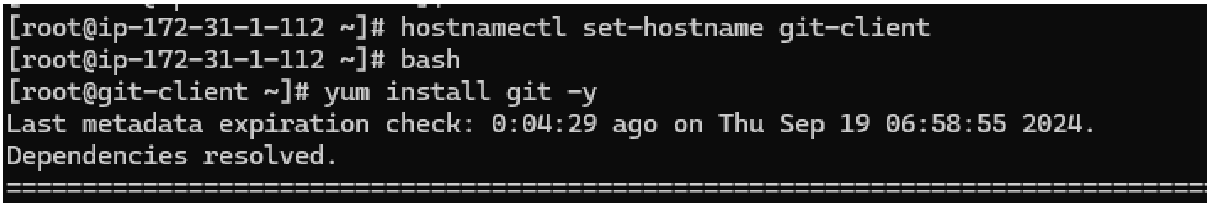
Elastic IP addresses: –

- Connect with terminal.



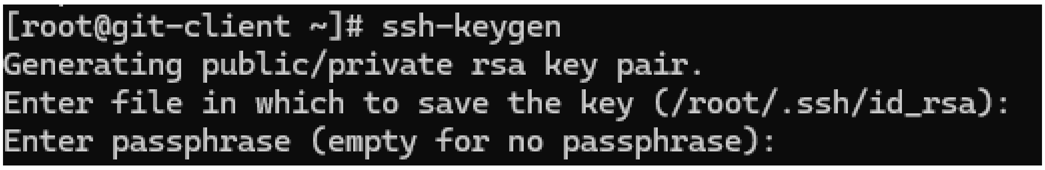
```
PS C:\Users\10748101\Downloads> ssh -i "Milestone-project-key.pem" ec2-user@ec2-3-108-52-52.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-3-108-52-52.ap-south-1.compute.amazonaws.com (3.108.52.52)' can't be established.
ED25519 key fingerprint is SHA256:kYHuv3wm2l/LRzBKYT88caDvq6kHkyj4Ra/uzJkIEag.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-108-52-52.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-1-112 ~]$ |
```

- Set Host name and install git in client-machine.



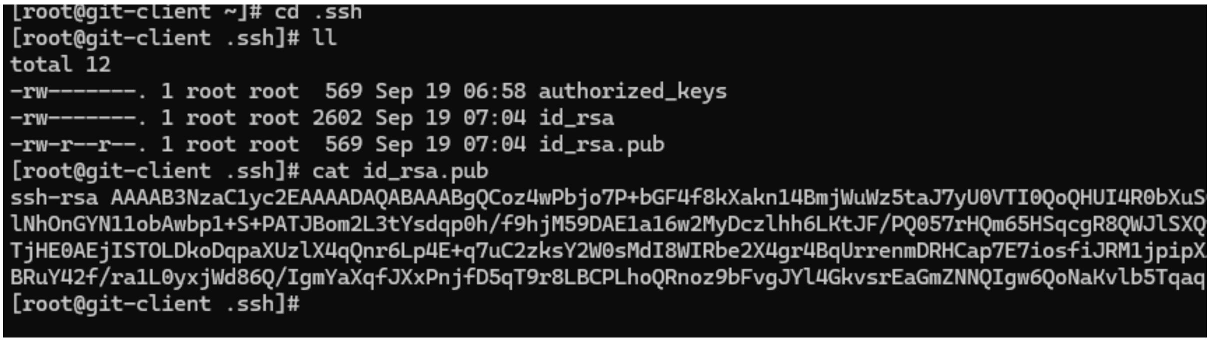
```
[root@ip-172-31-1-112 ~]# hostnamectl set-hostname git-client
[root@ip-172-31-1-112 ~]# bash
[root@git-client ~]# yum install git -y
Last metadata expiration check: 0:04:29 ago on Thu Sep 19 06:58:55 2024.
Dependencies resolved.
==========================================================================================
```

- Generate a ssh-key for connecting the client machine with GitHub.



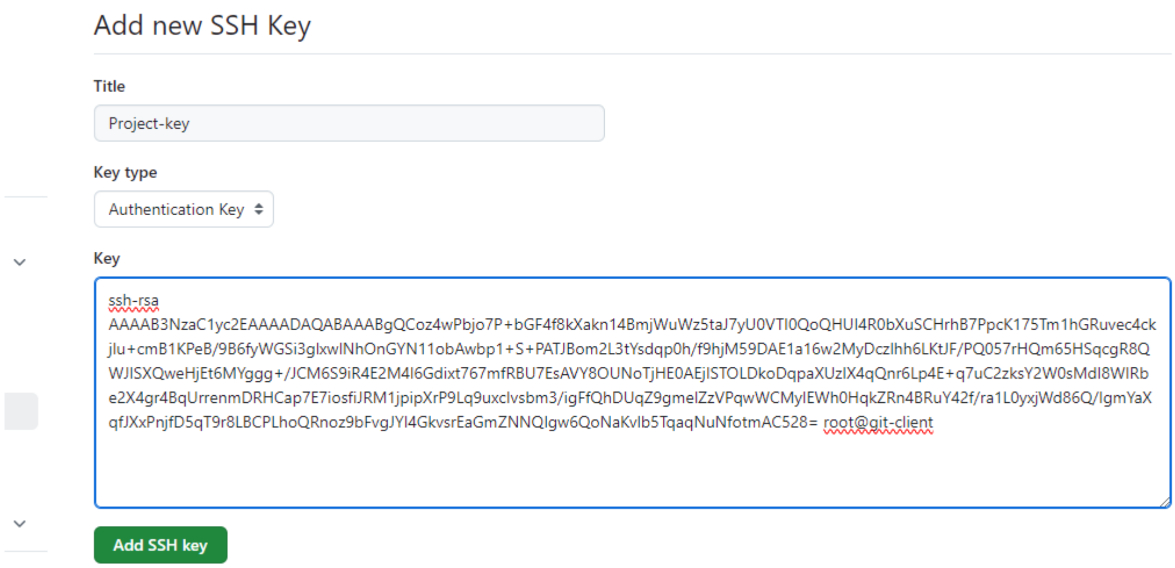
```
[root@git-client ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
```

- Copy the ssh key.



```
[root@git-client ~]# cd .ssh
[root@git-client .ssh]# ll
total 12
-rw-------. 1 root root  569 Sep 19 06:58 authorized_keys
-rw-------. 1 root root 2602 Sep 19 07:04 id_rsa
-rw-r--r--. 1 root root  569 Sep 19 07:04 id_rsa.pub
[root@git-client .ssh]# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCoz4wPbjo7P+bGF4f8kXakn14BmjWuWz5taJ7yU0VTI0QoQHUI4R0bXuS
lNhOnGYN11obAwbp1+S+PATJBom2L3tYsdqp0h/f9hjM59DAE1a16w2MyDczlhh6LKtJF/PQ057rHQm65HSqcgR8QWJlSXQ
TjHE0AEjISTOLDkoDqpaXUzlX4qQnr6Lp4E+q7uC2zksY2W0sMdI8WIRbe2X4gr4BqUrrenmDRHCap7E7iosfiJRM1jpipX
BRuY42f/ra1L0yxjWd86Q/IgmYaXqfJXxPnjfD5qT9r8LBCPLhoQRnoz9bFvgJYl4GkvsrEaGmZNNQIgw6QoNaKvlb5Tqaq
[root@git-client .ssh]#
```

- Go on GitHub and add new SSH key.



Add new SSH Key

Title

Project-key

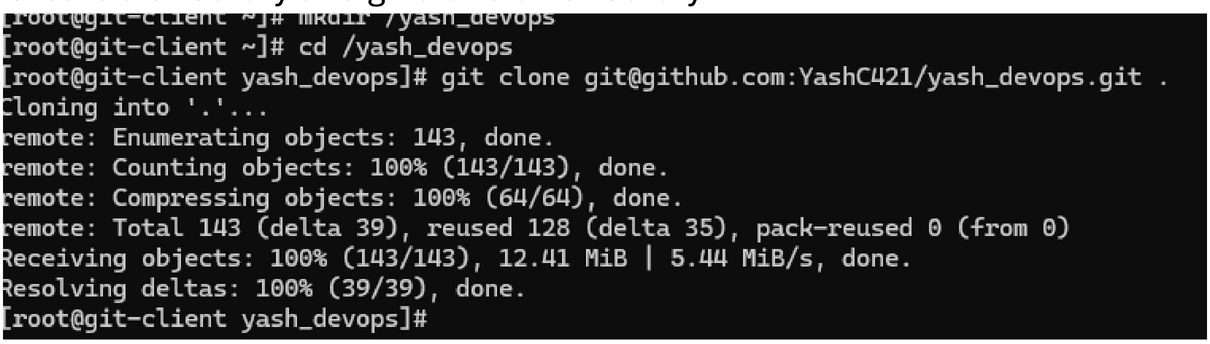
Key type

Authentication Key

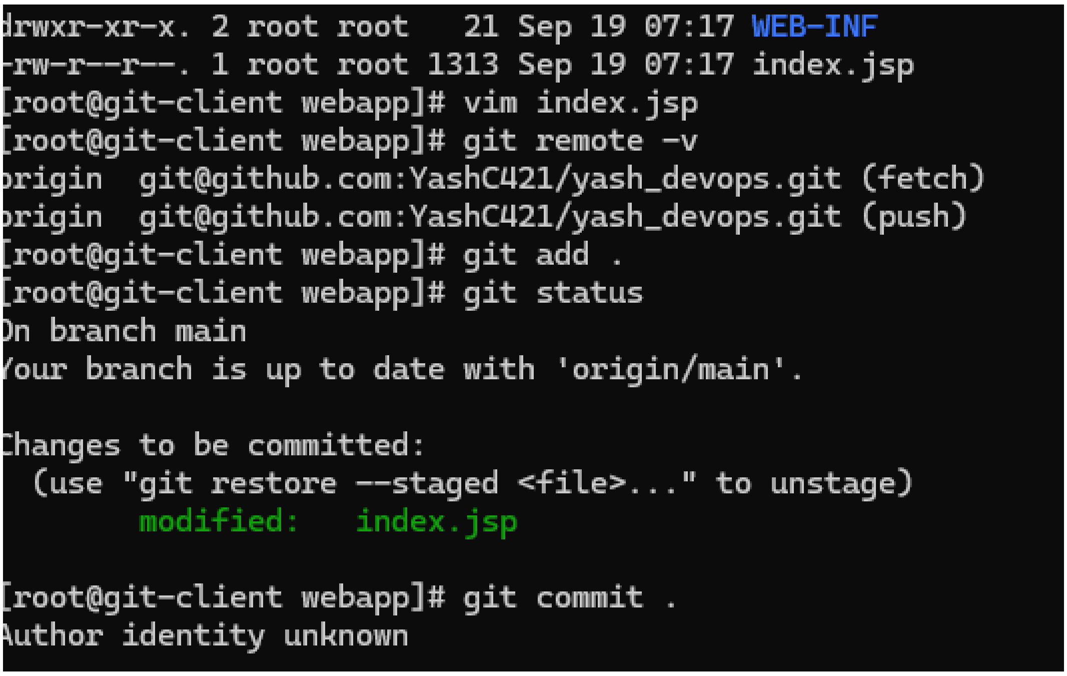
Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCoz4wPbjo7P+bGF4f8kXakn14BmjWuWz5taJ7yU0VTI0QoQHUI4R0bXuSCHrhB7PpcK175Tm1hGRuvec4ck
jlu+cmB1KPeB/9B6fyWGSi3glxwlNhOnGYN11obAwbp1+S+PATJBom2L3tYsdqp0h/f9hjM59DAE1a16w2MyDczlhh6LKtJF/PQ057rHQm65HSqcgR8Q
WJISXQweHjEt6MYggg+/JCM6S9iR4E2M4l6Gdixt767mfRBU7EsAVY8OUNoTjHE0AEjISTOLDkoDqpaXUzlX4qQnr6Lp4E+q7uC2zksY2W0sMdI8WIRb
e2X4gr4BqUrrenmDRHCap7E7iosfiJRM1jpipXrP9Lq9uxclvsbm3/igFfQhDUqZ9gmelZzVPqwWCMylEWh0HqkZRn4BRuY42f/ra1L0yxjWd86Q/IgmYaX
qfJXxPnjfD5qT9r8LBCPLhoQRnoz9bFvgJYI4GkvsrEaGmZNNQIgw6QoNaKvIb5TqaqNuNfotmAC528= root@git-client
```

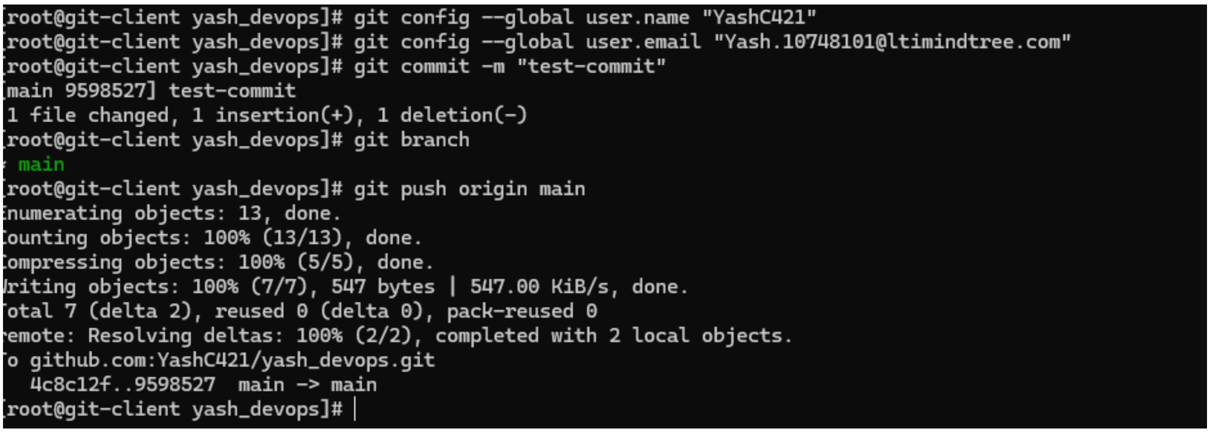
- Create a directory and git clone on directory.



```
[root@git-client ~]# mkdir /yash_devops
[root@git-client ~]# cd /yash_devops
[root@git-client yash_devops]# git clone git@github.com:YashC421/yash_devops.git .
Cloning into '.'...
remote: Enumerating objects: 143, done.
remote: Counting objects: 100% (143/143), done.
remote: Compressing objects: 100% (64/64), done.
remote: Total 143 (delta 39), reused 128 (delta 35), pack-reused 0 (from 0)
Receiving objects: 100% (143/143), 12.41 MiB | 5.44 MiB/s, done.
Resolving deltas: 100% (39/39), done.
[root@git-client yash_devops]#
```

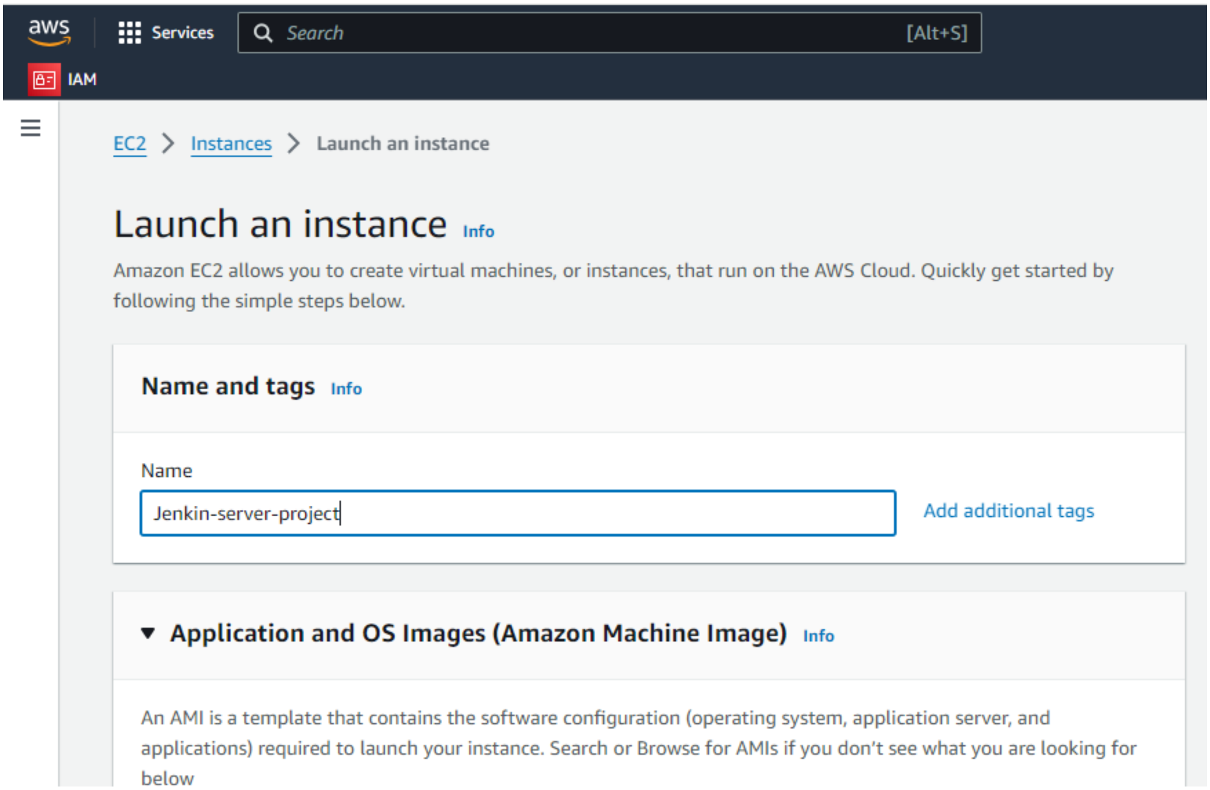


```
drwxr-xr-x. 2 root root   21 Sep 19 07:17 WEB-INF
-rw-r--r--. 1 root root 1313 Sep 19 07:17 index.jsp
[root@git-client webapp]# vim index.jsp
[root@git-client webapp]# git remote -v
origin  git@github.com:YashC421/yash_devops.git (fetch)
origin  git@github.com:YashC421/yash_devops.git (push)
[root@git-client webapp]# git add .
[root@git-client webapp]# git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.jsp

[root@git-client webapp]# git commit .
Author identity unknown
```



```
[root@git-client yash_devops]# git config --global user.name "YashC421"
[root@git-client yash_devops]# git config --global user.email "Yash.10748101@ltimindtree.com"
[root@git-client yash_devops]# git commit -m "test-commit"
[main 9598527] test-commit
 1 file changed, 1 insertion(+), 1 deletion(-)
[root@git-client yash_devops]# git branch
* main
[root@git-client yash_devops]# git push origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 547 bytes | 547.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:YashC421/yash_devops.git
   4c8c12f..9598527  main -> main
[root@git-client yash_devops]# |
```

# Launch an instance for Jenkins Machine.



EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

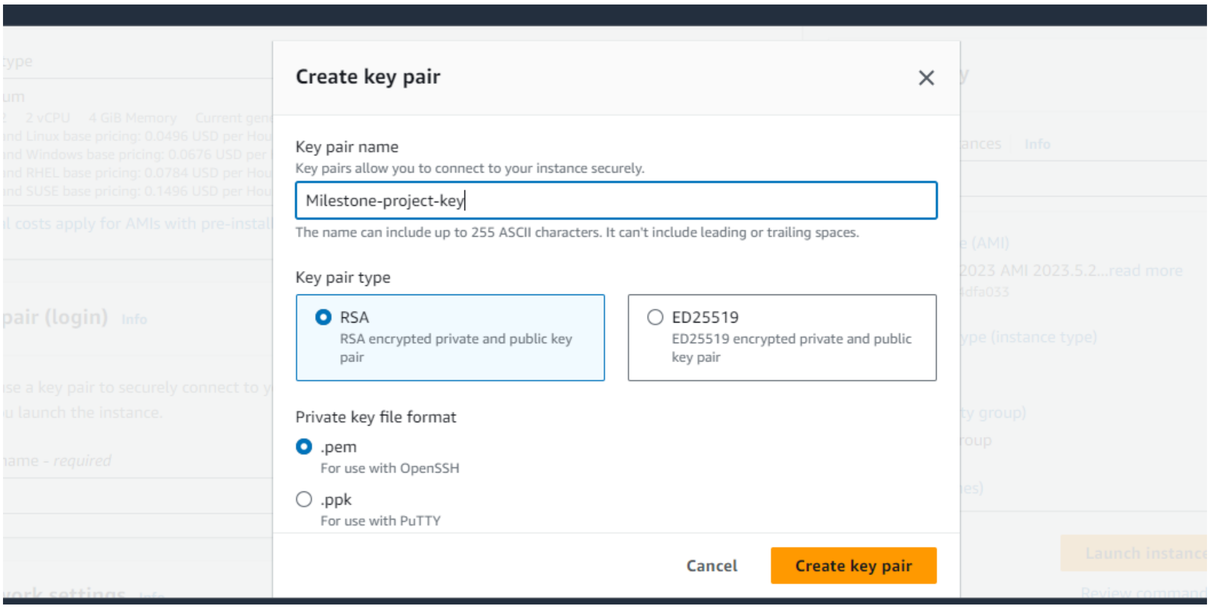
Name: Jenkin-server-project

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

# Create a key pair to attach with the instance.



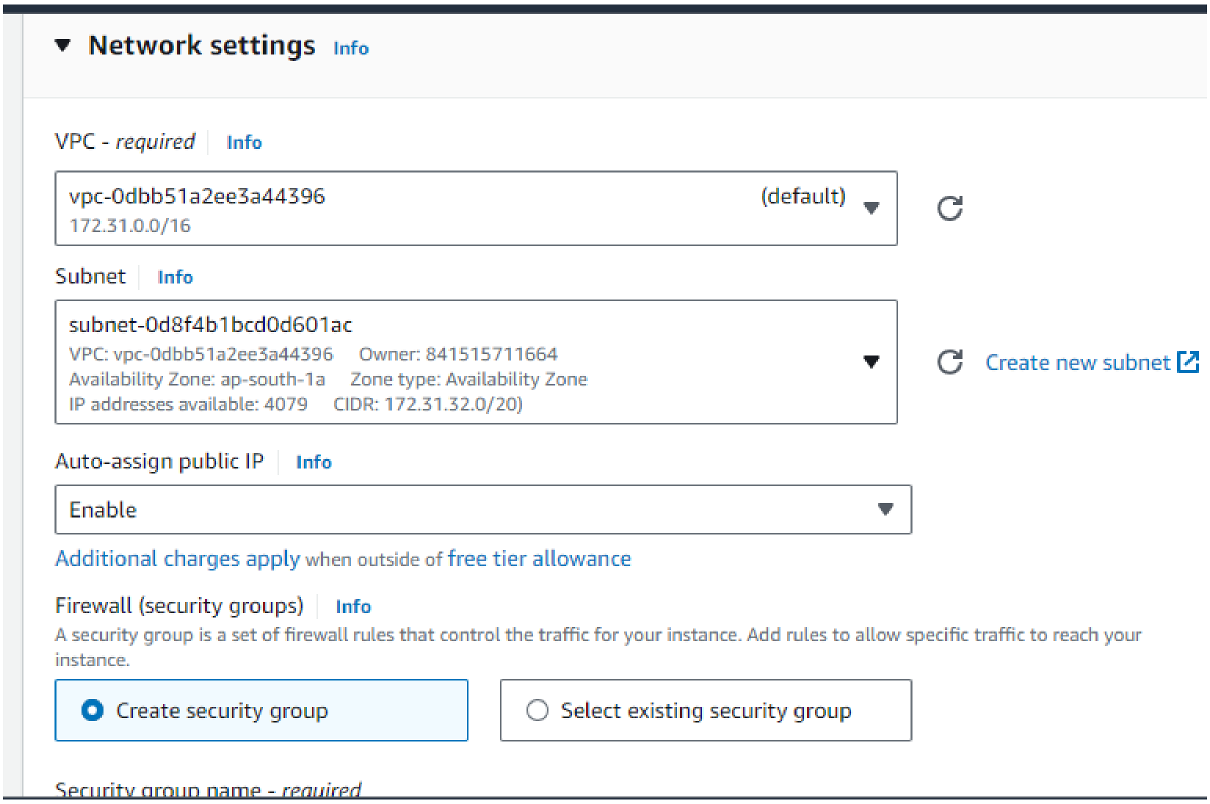
Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.
Milestone-project-key
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type
● RSA
RSA encrypted private and public key pair
○ ED25519
ED25519 encrypted private and public key pair

Private key file format
● .pem
For use with OpenSSH
○ .ppk
For use with PuTTY

# In Network Setting select VPC, Subnet and Create security Group and attach with instance.



▼ Network settings Info

VPC - *required* Info

vpc-0dbb51a2ee3a44396 (default)
172.31.0.0/16

Subnet Info

subnet-0d8f4b1bcd0d601ac
VPC: vpc-0dbb51a2ee3a44396 Owner: 841515711664
Availability Zone: ap-south-1a Zone type: Availability Zone
IP addresses available: 4079 CIDR: 172.31.32.0/20)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

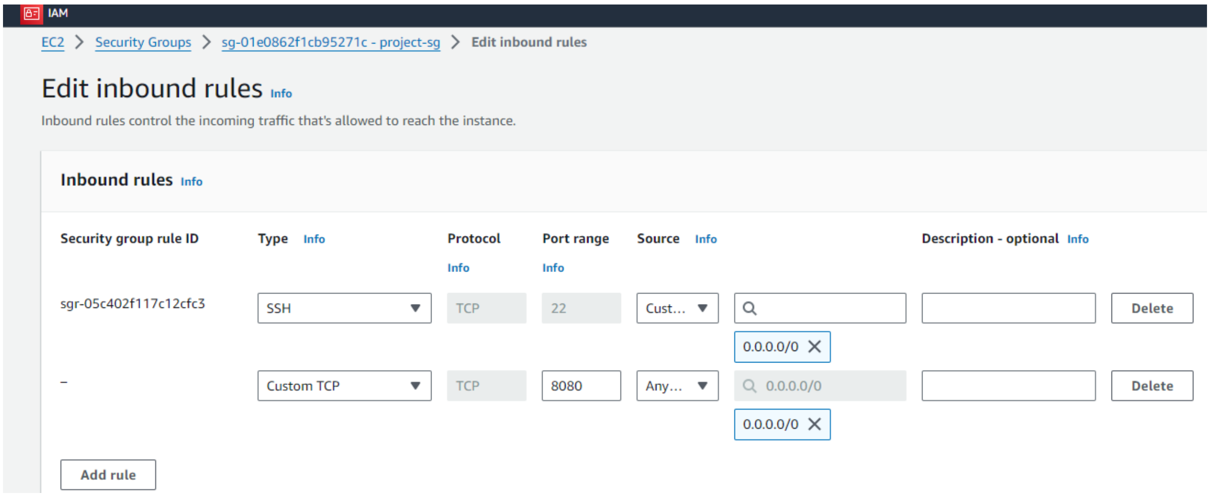
Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

Security group name - *required*

#Now add inbound rules where we add port 8080 enable for Jenkins-server and access it anywhere.



EC2 > Security Groups > sg-01e0862f1cb95271c - project-sg > Edit inbound rules

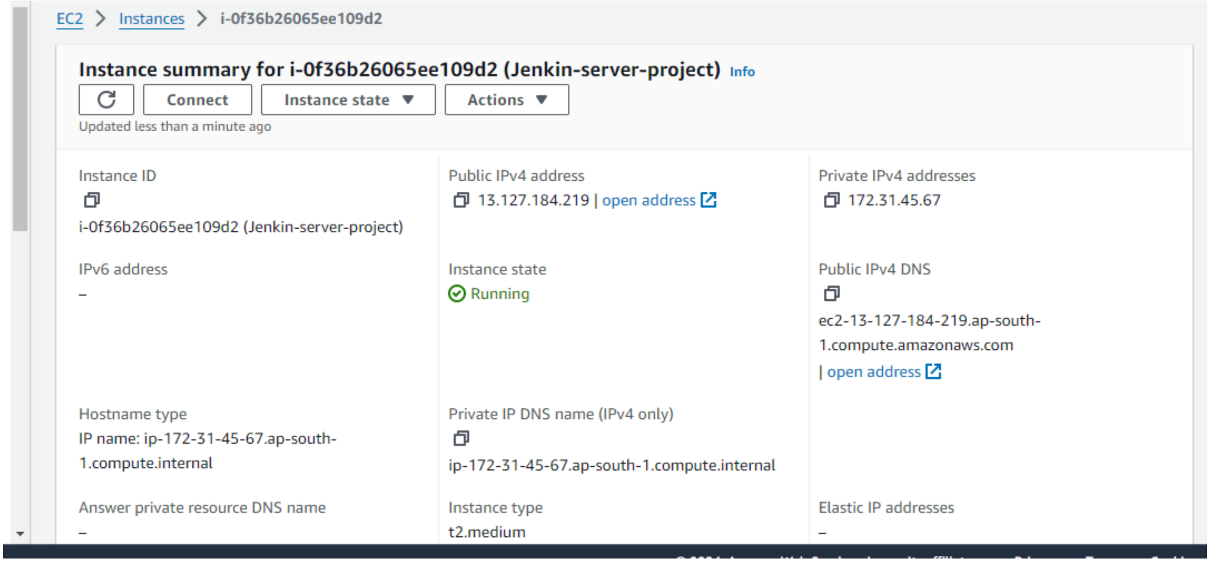
Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-05c402f117c12cfc3	SSH	TCP	22	Cust...	0.0.0.0/0		Delete
–	Custom TCP	TCP	8080	Any...	0.0.0.0/0		Delete

# Our instance is created with all configuration.



Instance summary for i-0f36b26065ee109d2 (Jenkin-server-project) Info

Updated less than a minute ago

Instance ID
i-0f36b26065ee109d2 (Jenkin-server-project)

Public IPv4 address
13.127.184.219 | open address

Private IPv4 addresses
172.31.45.67

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-13-127-184-219.ap-south-1.compute.amazonaws.com
| open address

Hostname type
IP name: ip-172-31-45-67.ap-south-1.compute.internal

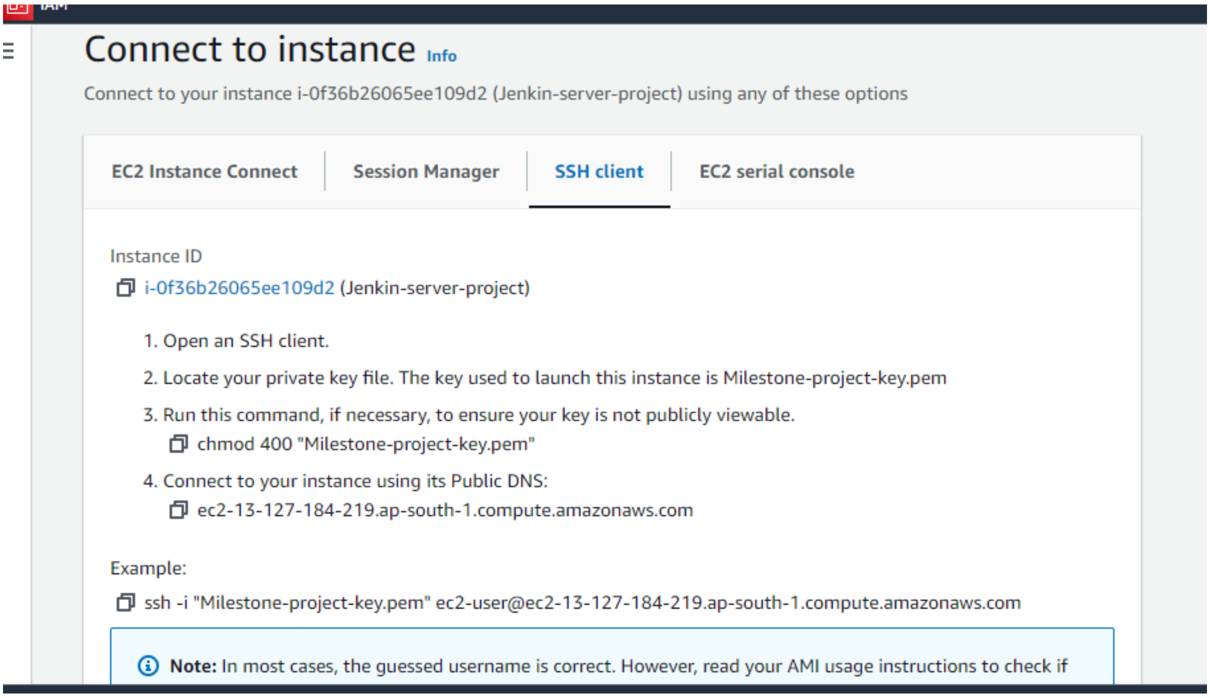
Private IP DNS name (IPv4 only)
ip-172-31-45-67.ap-south-1.compute.internal

Answer private resource DNS name
–

Instance type
t2.medium

Elastic IP addresses
–

# Now connect instance in terminal to start with Jenkins.



Connect to instance Info

Connect to your instance i-0f36b26065ee109d2 (Jenkin-server-project) using any of these options

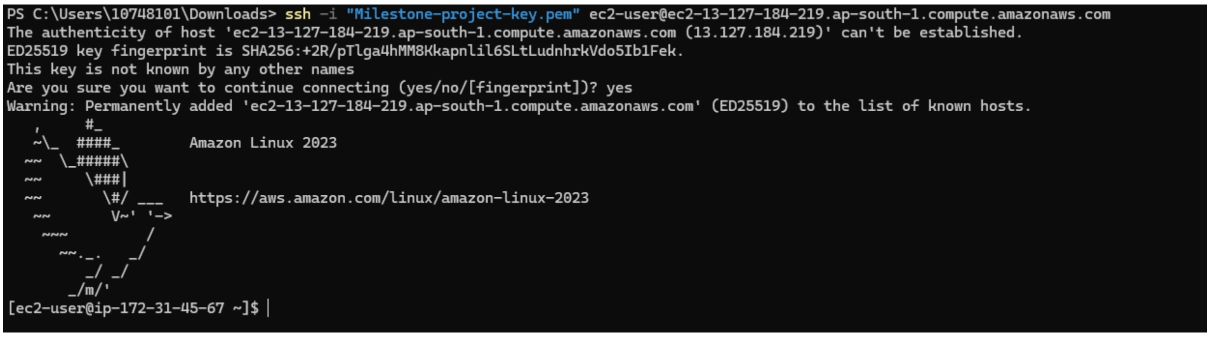
Instance ID
i-0f36b26065ee109d2 (Jenkin-server-project)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Milestone-project-key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   ```
   chmod 400 "Milestone-project-key.pem"
   ```
4. Connect to your instance using its Public DNS:
   ec2-13-127-184-219.ap-south-1.compute.amazonaws.com

Example:

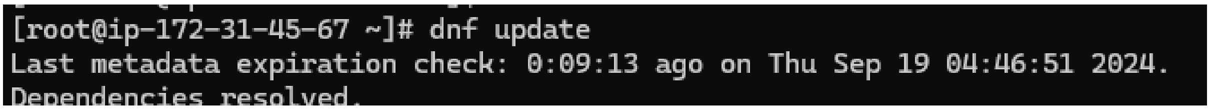
```
ssh -i "Milestone-project-key.pem" ec2-user@ec2-13-127-184-219.ap-south-1.compute.amazonaws.com
```

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if

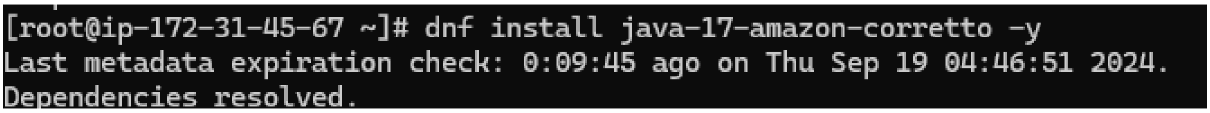


```
PS C:\Users\10748101\Downloads> ssh -i "Milestone-project-key.pem" ec2-user@ec2-13-127-184-219.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-127-184-219.ap-south-1.compute.amazonaws.com (13.127.184.219)' can't be established.
ED25519 key fingerprint is SHA256:+2R/pTlga4hMM8Kkapnlil6SLtLudnhrkVdo5Ib1Fek.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-127-184-219.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-45-67 ~]$ |
```

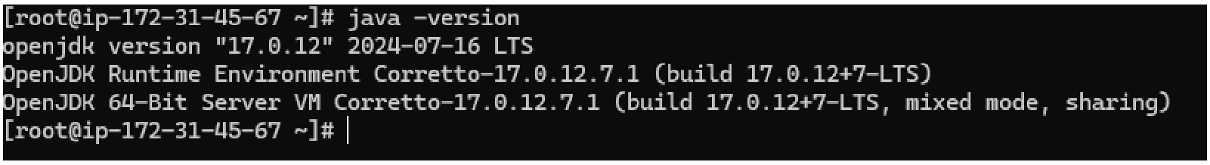
# Installing Jenkins in this Machine.



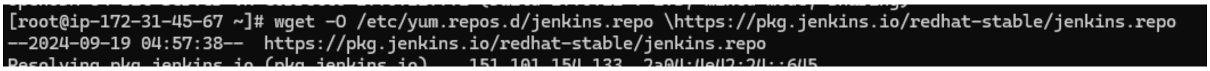
```
[root@ip-172-31-45-67 ~]# dnf update
Last metadata expiration check: 0:09:13 ago on Thu Sep 19 04:46:51 2024.
Dependencies resolved.
```



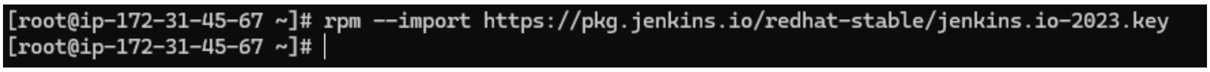
```
[root@ip-172-31-45-67 ~]# dnf install java-17-amazon-corretto -y
Last metadata expiration check: 0:09:45 ago on Thu Sep 19 04:46:51 2024.
Dependencies resolved.
```



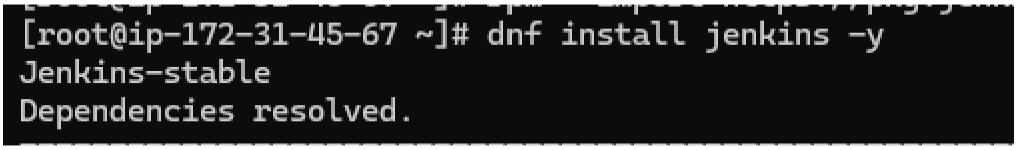
```
[root@ip-172-31-45-67 ~]# java -version
openjdk version "17.0.12" 2024-07-16 LTS
OpenJDK Runtime Environment Corretto-17.0.12.7.1 (build 17.0.12+7-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.12.7.1 (build 17.0.12+7-LTS, mixed mode, sharing)
[root@ip-172-31-45-67 ~]# |
```



```
[root@ip-172-31-45-67 ~]# wget -O /etc/yum.repos.d/jenkins.repo \https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-09-19 04:57:38--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.154.133, 2a04:4e42:24::645
```

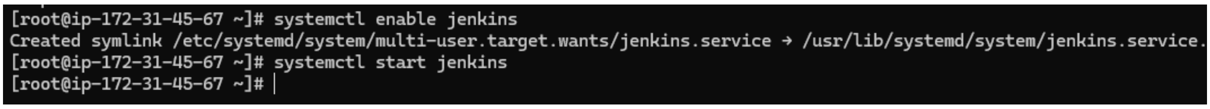


```
[root@ip-172-31-45-67 ~]# rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[root@ip-172-31-45-67 ~]# |
```



```
[root@ip-172-31-45-67 ~]# dnf install jenkins -y
Jenkins-stable
Dependencies resolved.
```

# Enable and Start Jenkins in this Machine.



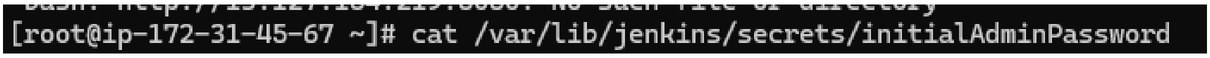
```
[root@ip-172-31-45-67 ~]# systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[root@ip-172-31-45-67 ~]# systemctl start jenkins
[root@ip-172-31-45-67 ~]# |
```

# Now with your instance public IP go on browser and search the url. Our Jenkins – server is running on this IP.

http://13.127.184.219:8080

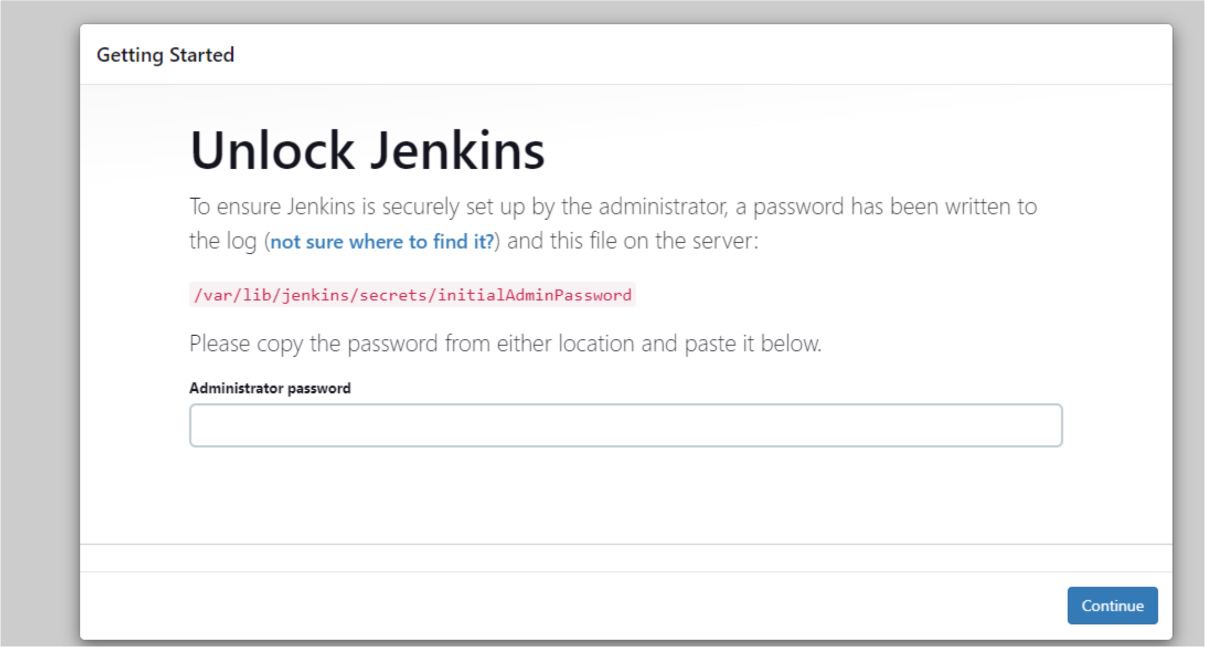
# Now we have to unlock the Jenkins.

- Copy the path which is shown in with unlock tab.
- Paste it with cat command in terminal to see the password.



```
bash: http://13.127.184.219:8080: No such file or directory
[root@ip-172-31-45-67 ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
```

- Copy password and paste it as administrator password and continue.



Getting Started

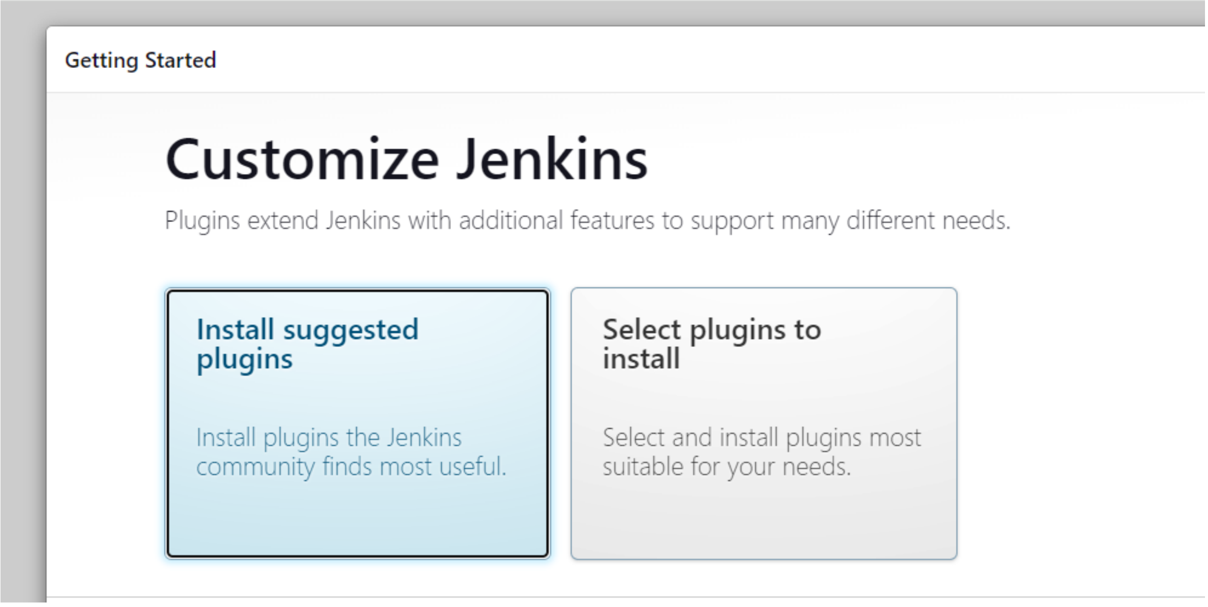
Unlock Jenkins

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

/var/lib/jenkins/secrets/initialAdminPassword

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

Administrator password



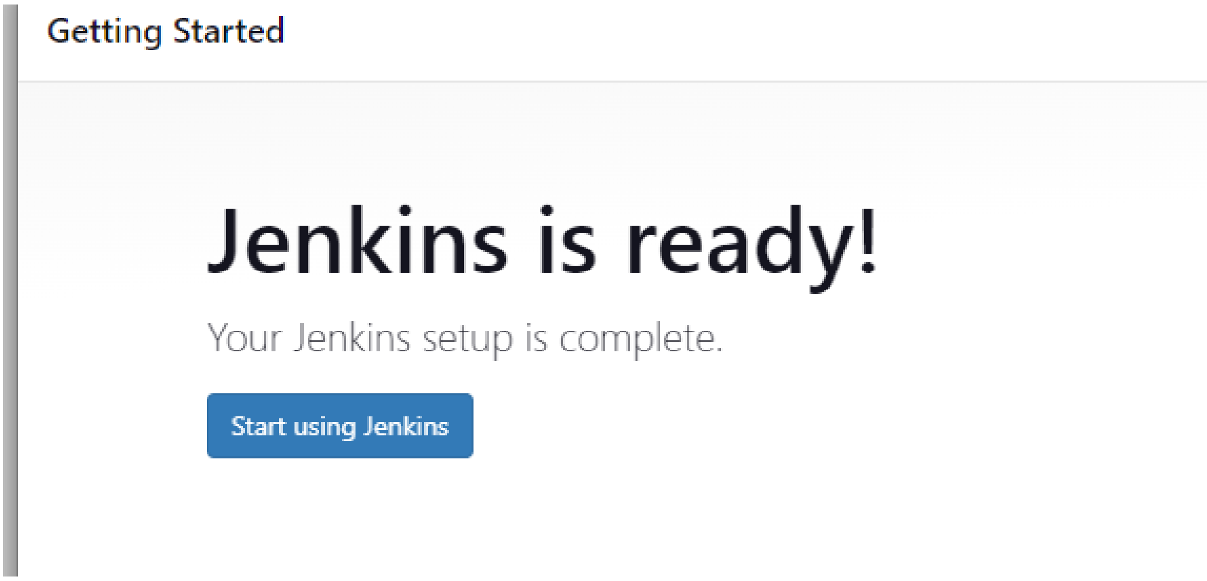
Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.

# Jenkins setup is completed now.



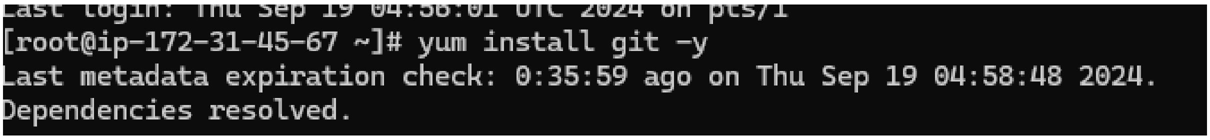
Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

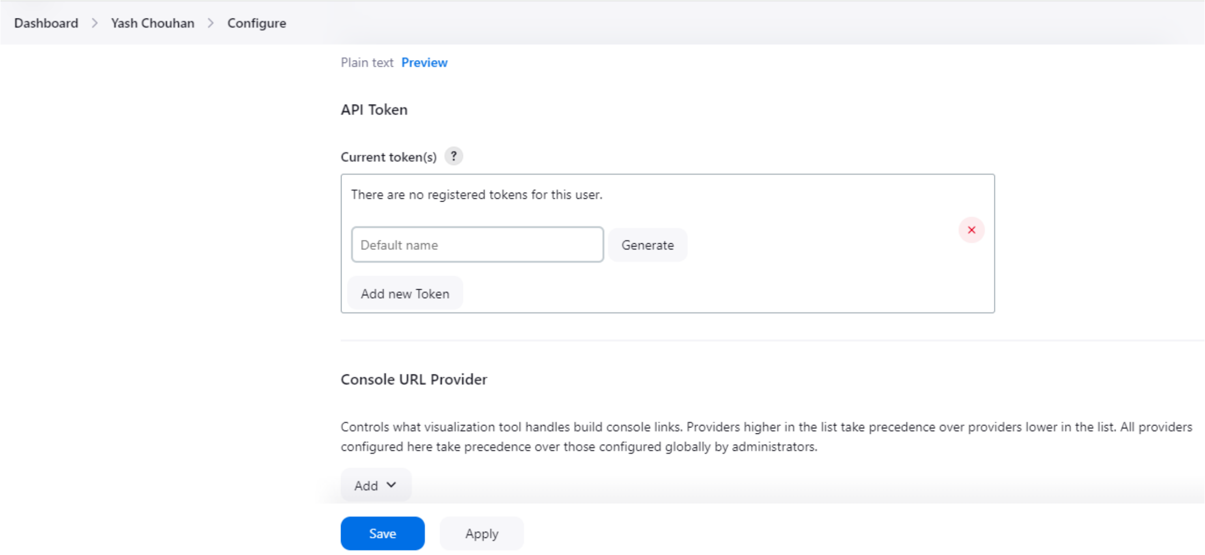
#Now we are going to configure Jenkins with GitHub.

- Install git on Jenkins server to start with GitHub.



```
Last login: Thu Sep 19 04:58:01 UTC 2024 on pts/1
[root@ip-172-31-45-67 ~]# yum install git -y
Last metadata expiration check: 0:35:59 ago on Thu Sep 19 04:58:48 2024.
Dependencies resolved.
```

- Generate a token in Jenkins configuration.



API Token

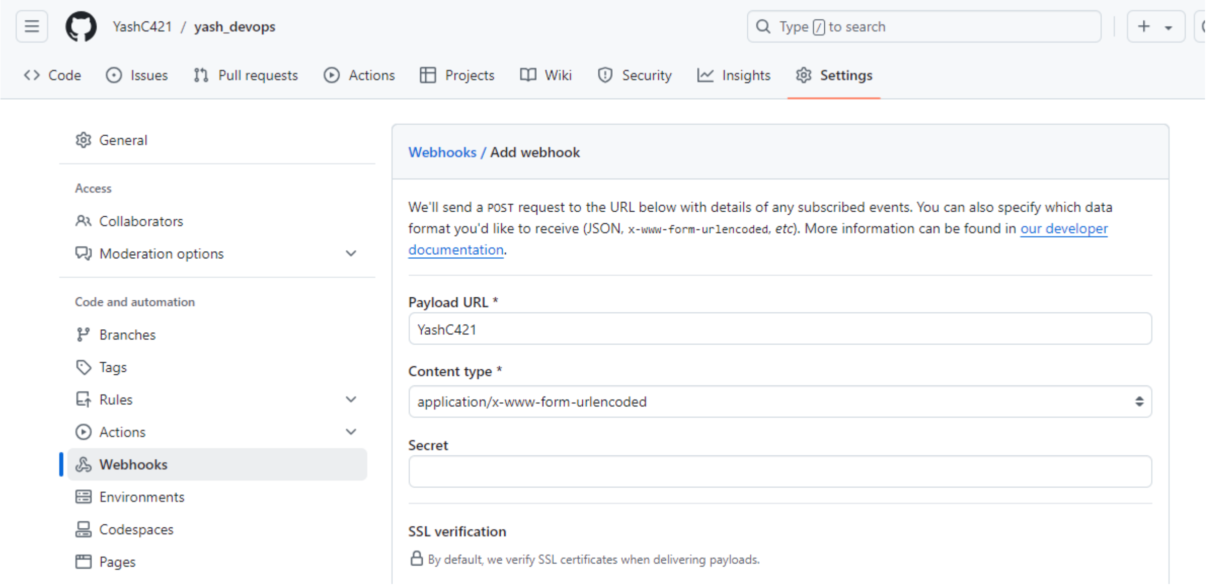
Current token(s) ?

There are no registered tokens for this user.

Console URL Provider

Controls what visualization tool handles build console links. Providers higher in the list take precedence over providers lower in the list. All providers configured here take precedence over those configured globally by administrators.

# Open GitHub Create a repository where your project is stored. After that Add Webhook to configure with Jenkins.



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 *: YashC421

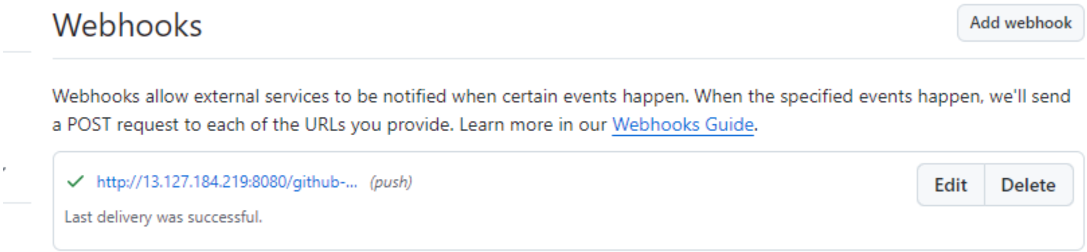
Content type *: application/x-www-form-urlencoded

Secret:

SSL verification

By default, we verify SSL certificates when delivering payloads.

- Webhook is created successfully.



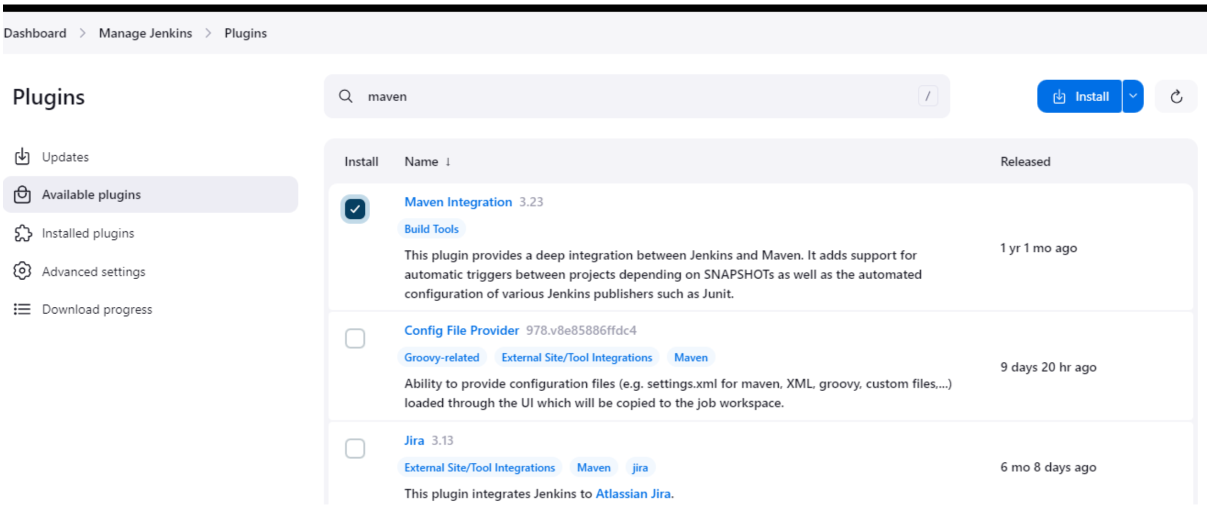
Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

✓ http://13.127.184.219:8080/github-... *(push)*

Last delivery was successful.

# Now we have to Configure maven with Jenkins.

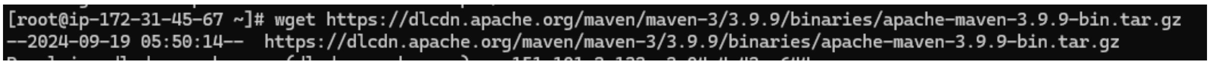
- Add maven plugin and install it.



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
☐	Config File Provider 978.v8e85886ffdc4 Groovy-related External Site/Tool Integrations Maven Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace.	9 days 20 hr ago
☐	Jira 3.13 External Site/Tool Integrations Maven jira This plugin integrates Jenkins to Atlassian Jira.	6 mo 8 days ago

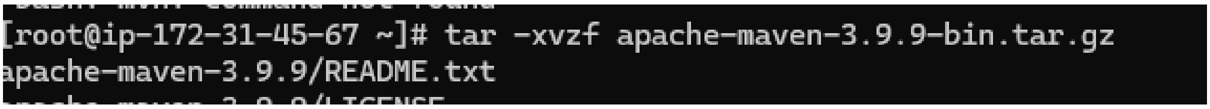
#Now install maven in Jenkins -server.

- Get maven.



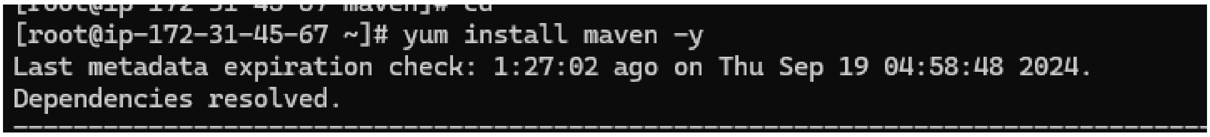
```
[root@ip-172-31-45-67 ~]# wget https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz
--2024-09-19 05:50:14--  https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz
```

- Extract the tar file.



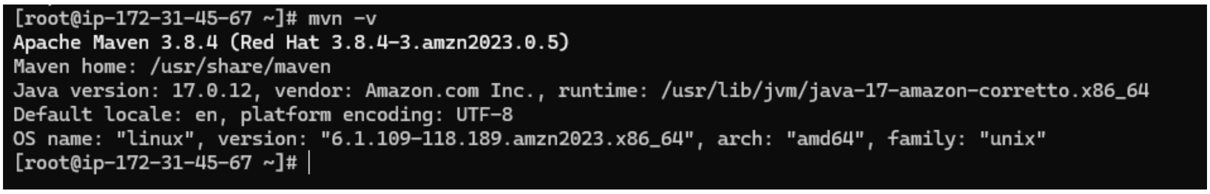
```
bash: mvn: command not found
[root@ip-172-31-45-67 ~]# tar -xvzf apache-maven-3.9.9-bin.tar.gz
apache-maven-3.9.9/README.txt
apache-maven-3.9.9/LICENSE
```

- Install maven globally.



```
[root@ip-172-31-45-67 maven]# cd
[root@ip-172-31-45-67 ~]# yum install maven -y
Last metadata expiration check: 1:27:02 ago on Thu Sep 19 04:58:48 2024.
Dependencies resolved.
================================================================================
```

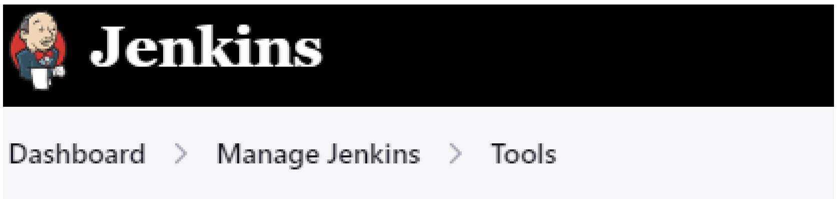
- Maven is successfully installed in Jenkins Machine.

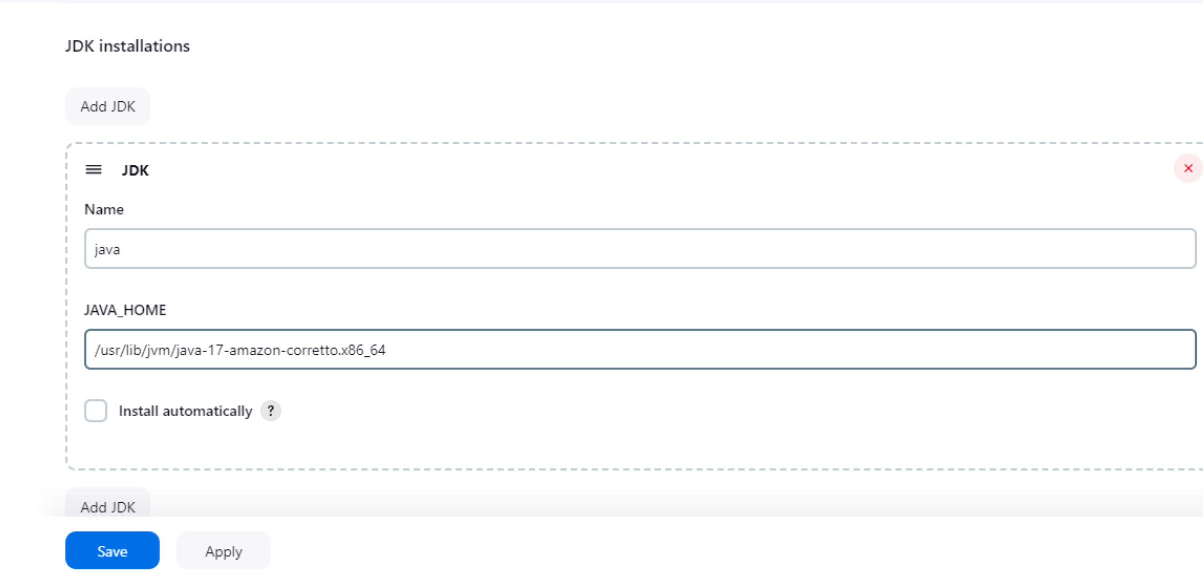


```
[root@ip-172-31-45-67 ~]# mvn -v
Apache Maven 3.8.4 (Red Hat 3.8.4-3.amzn2023.0.5)
Maven home: /usr/share/maven
Java version: 17.0.12, vendor: Amazon.com Inc., runtime: /usr/lib/jvm/java-17-amazon-corretto.x86_64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.1.109-118.189.amzn2023.x86_64", arch: "amd64", family: "unix"
[root@ip-172-31-45-67 ~]# |
```

#Configuring/Adding JDK And Maven in Jenkins.

- Go to tools and add JDK and Maven installation path -→ Save and apply.





JDK installations

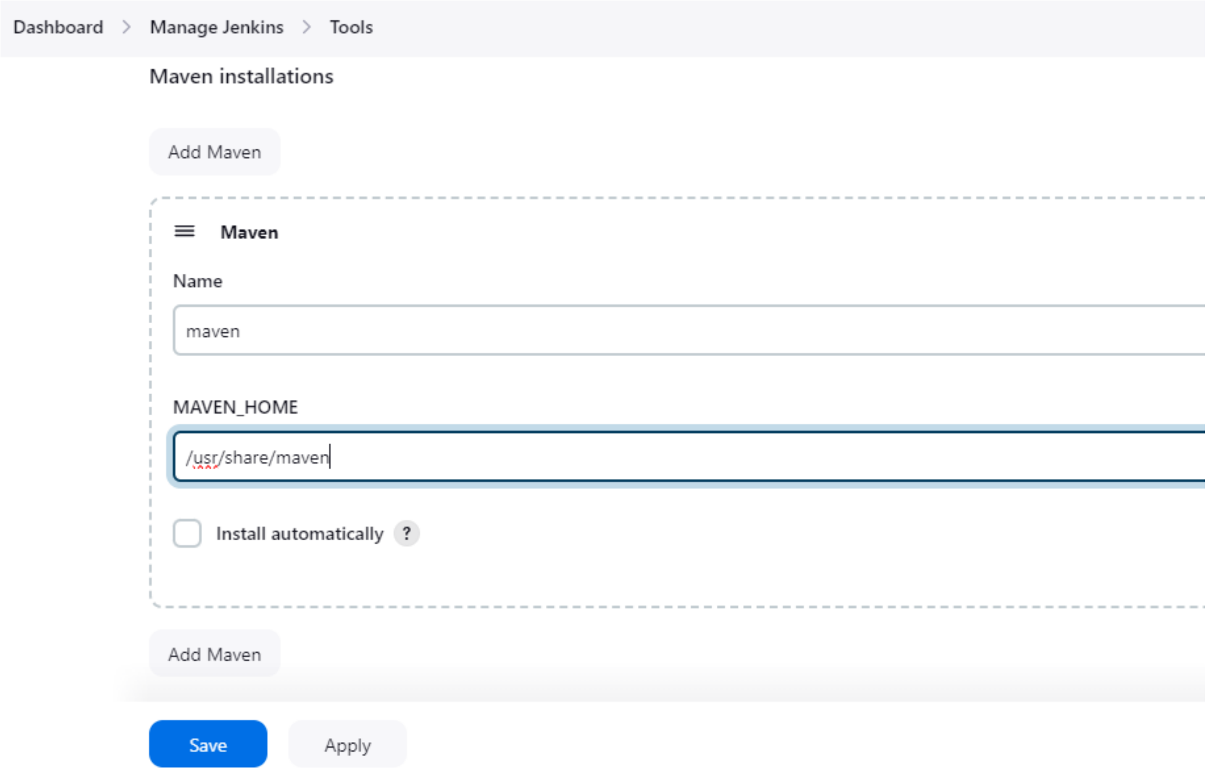
Add JDK

JDK

Name: java

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

□ Install automatically ?



Maven installations

Maven

Name

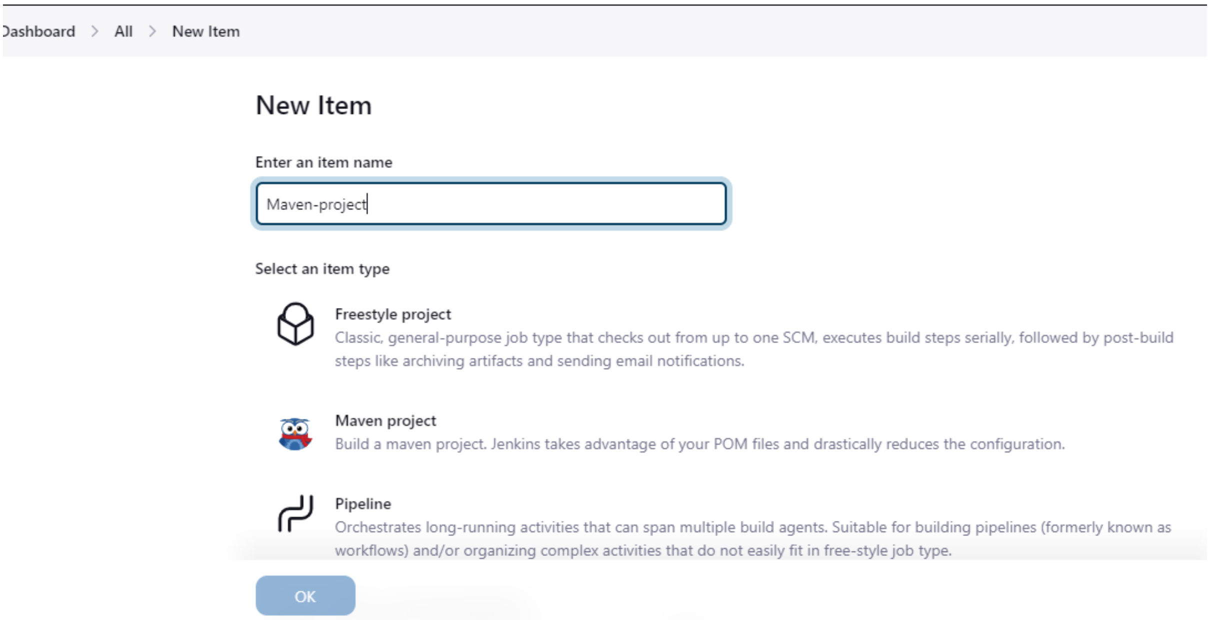
maven

MAVEN_HOME

/usr/share/maven

□ Install automatically ?

- Add new item on Jenkins for maven project.



New Item

Enter an item name

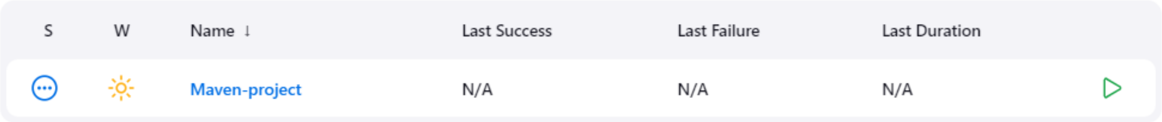
Maven-project

Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

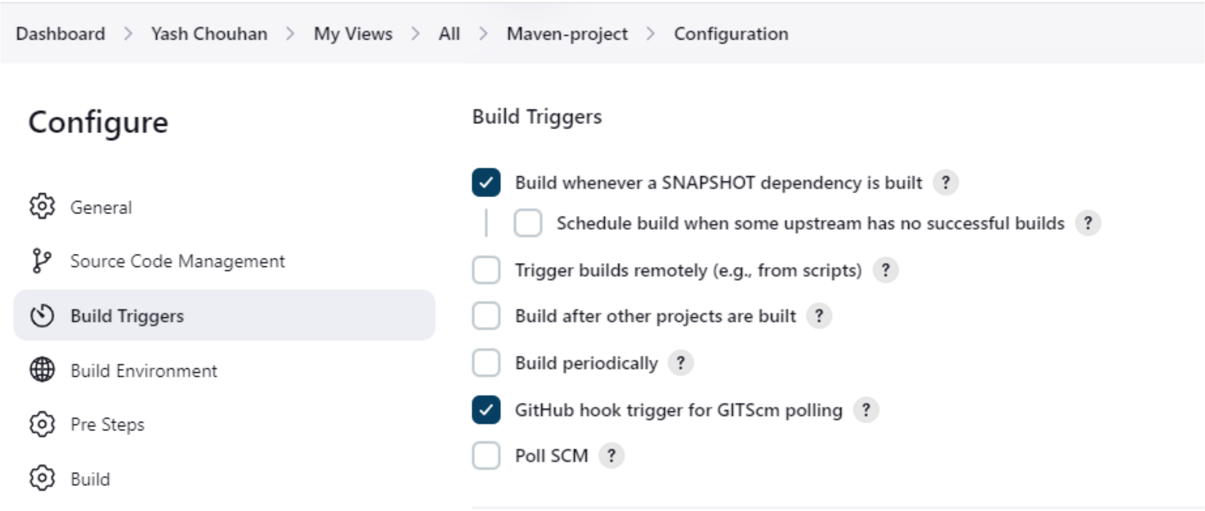
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



S	W	Name ↓	Last Success	Last Failure	Last Duration	
⋯	☀	Maven-project	N/A	N/A	N/A	▷

- Save configuration to connect with GitHub.

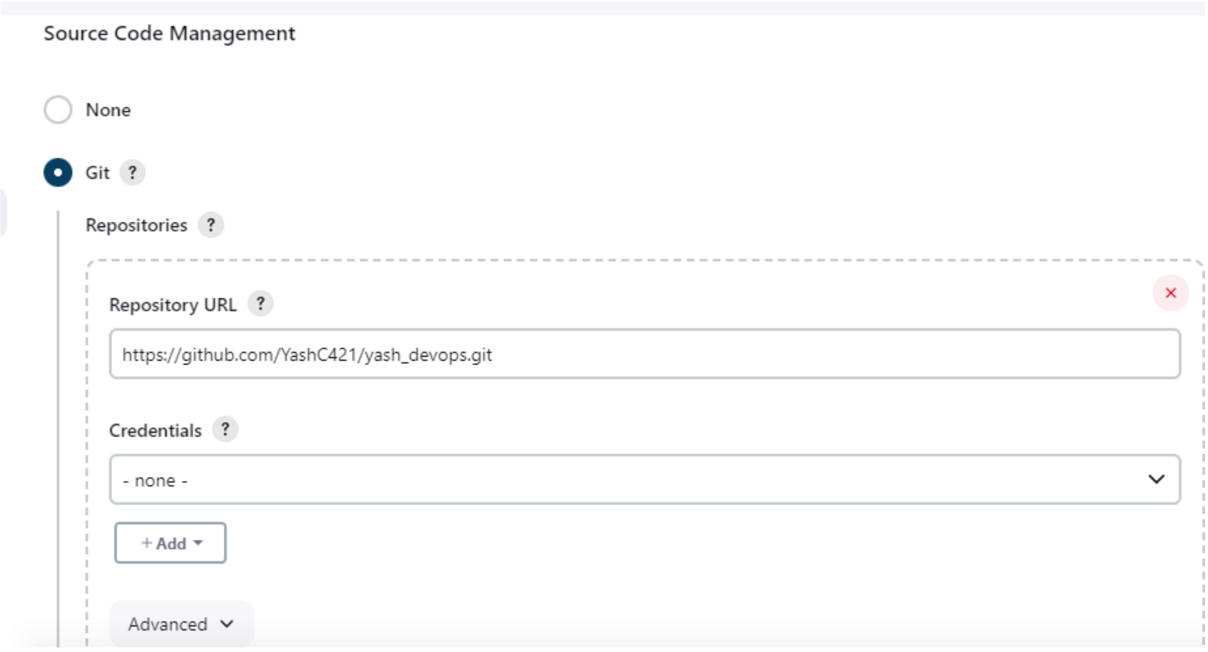


Dashboard > Yash Chouhan > My Views > All > Maven-project > Configuration

Configure

Build Triggers

- ☑ Build whenever a SNAPSHOT dependency is built ?
  - □ Schedule build when some upstream has no successful builds ?
- □ Trigger builds remotely (e.g., from scripts) ?
- □ Build after other projects are built ?
- □ Build periodically ?
- ☑ GitHub hook trigger for GITScm polling ?
- □ Poll SCM ?



Source Code Management

○ None

● Git ?

Repositories ?

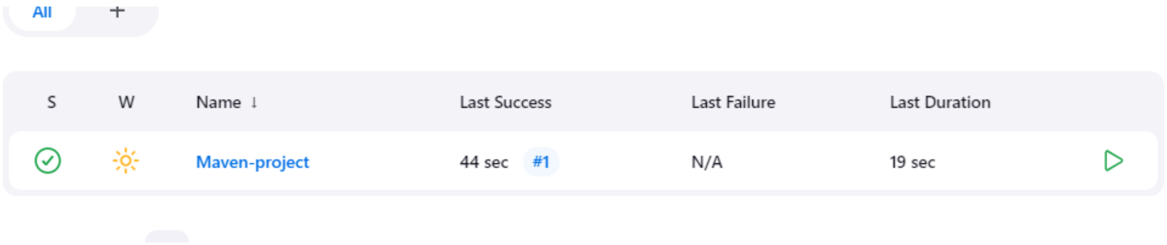
Repository URL ?

https://github.com/YashC421/yash_devops.git

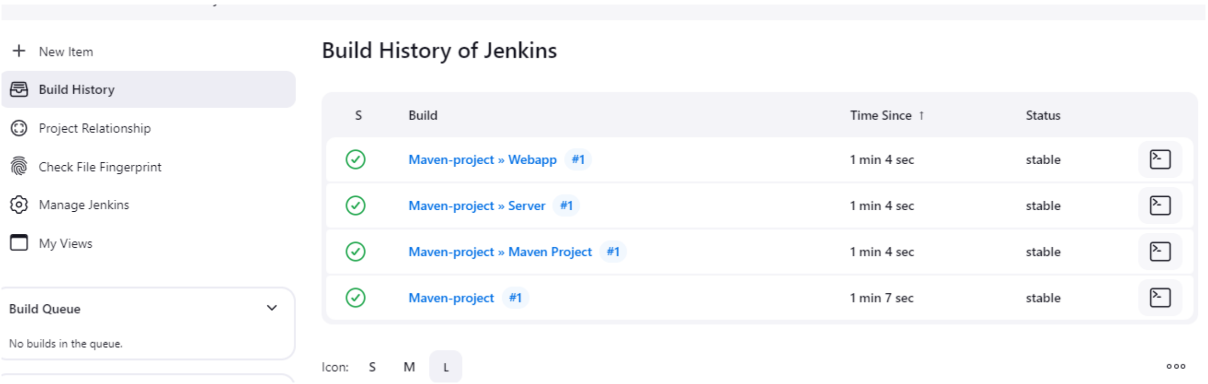
Credentials ?

- none -

# Now build the job.



S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	Maven-project	44 sec #1	N/A	19 sec	▷

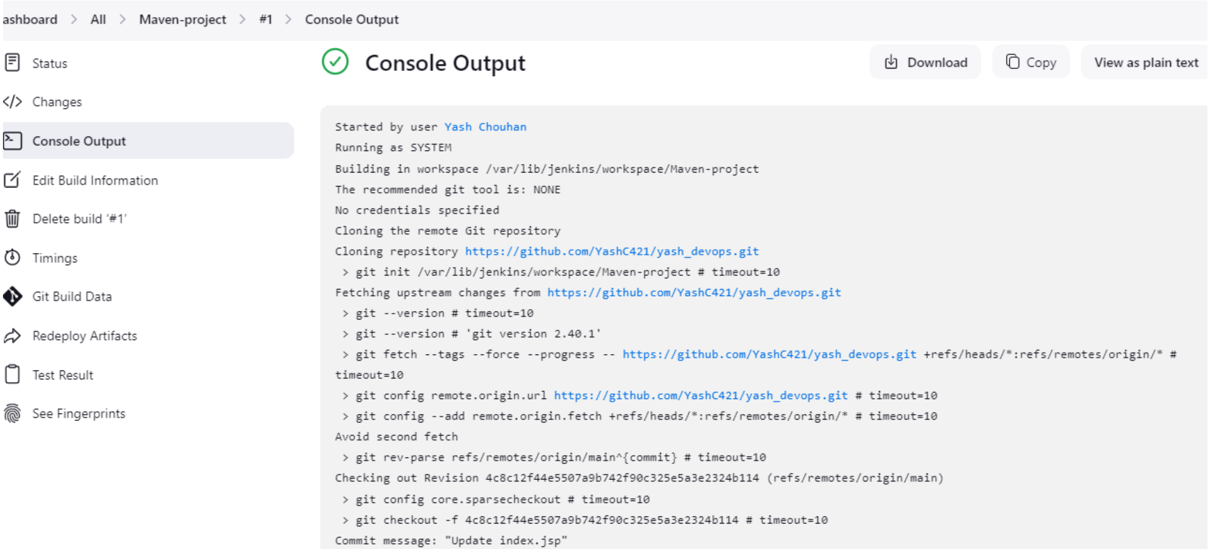


No builds in the queue.

Build History of Jenkins

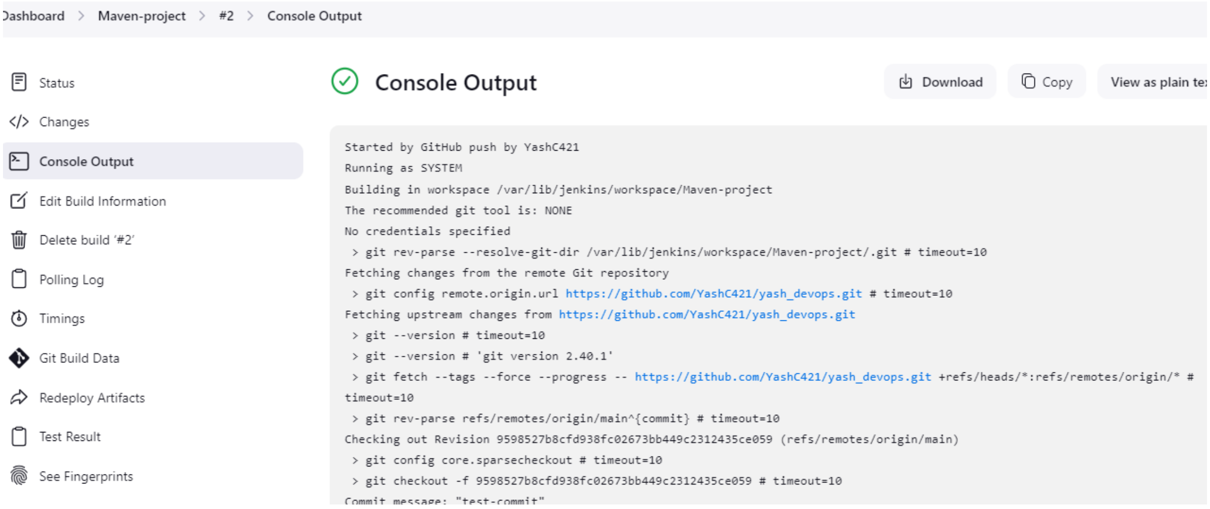
S	Build	Time Since ↑	Status
✓	Maven-project » Webapp #1	1 min 4 sec	stable
✓	Maven-project » Server #1	1 min 4 sec	stable
✓	Maven-project » Maven Project #1	1 min 4 sec	stable
✓	Maven-project #1	1 min 7 sec	stable

# job build successfully.



Console Output

```
Started by user Yash Chouhan
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Maven-project
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/YashC421/yash_devops.git
 > git init /var/lib/jenkins/workspace/Maven-project # timeout=10
Fetching upstream changes from https://github.com/YashC421/yash_devops.git
 > git --version # timeout=10
 > git --version # 'git version 2.40.1'
 > git fetch --tags --force --progress -- https://github.com/YashC421/yash_devops.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
 > git config remote.origin.url https://github.com/YashC421/yash_devops.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 4c8c12f44e5507a9b742f90c325e5a3e2324b114 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 4c8c12f44e5507a9b742f90c325e5a3e2324b114 # timeout=10
Commit message: "Update index.jsp"
```

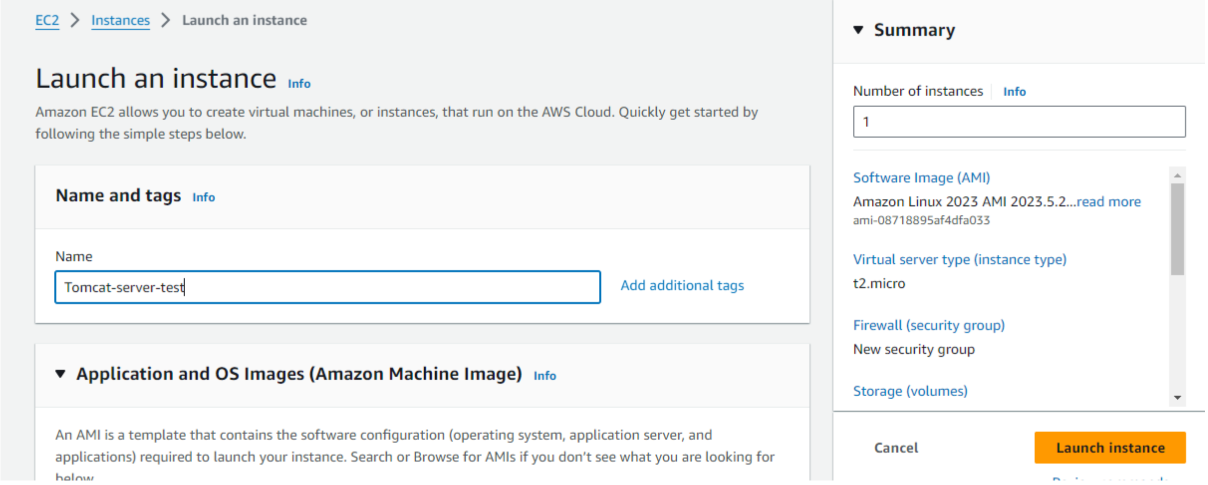


Console Output

```
Started by GitHub push by YashC421
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Maven-project
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Maven-project/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/YashC421/yash_devops.git # timeout=10
Fetching upstream changes from https://github.com/YashC421/yash_devops.git
 > git --version # timeout=10
 > git --version # 'git version 2.40.1'
 > git fetch --tags --force --progress -- https://github.com/YashC421/yash_devops.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 9598527b8cfd938fc02673bb449c2312435ce059 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 9598527b8cfd938fc02673bb449c2312435ce059 # timeout=10
Commit message: "test-commit"
```

# Configure Tomcat for Testing purpose.

- Create a instance.



Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: Tomcat-server-test

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Summary

Number of instances: 1

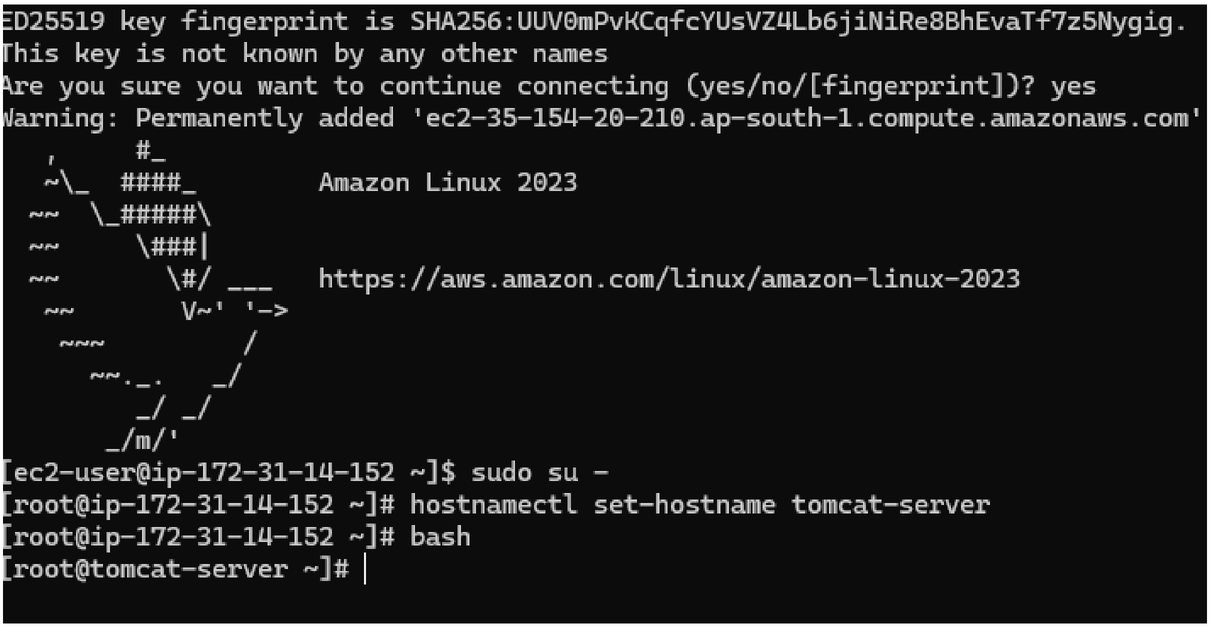
Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-08718895af4dfa033

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

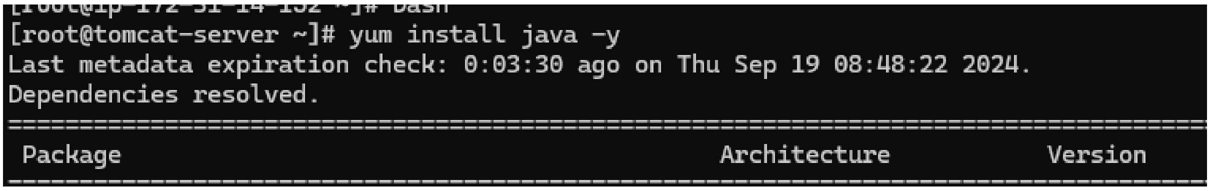
Storage (volumes)

- Connect instance

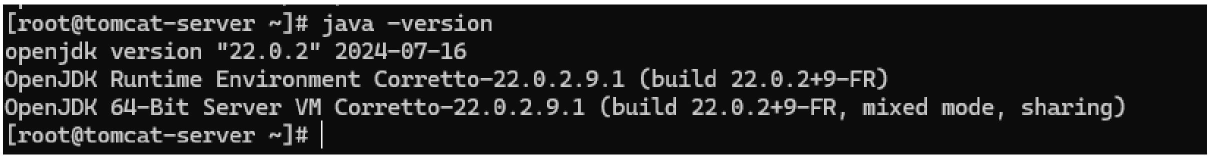


```
ED25519 key fingerprint is SHA256:UUV0mPvKCqfcYUsVZ4Lb6jiNiRe8BhEvaTf7z5Nygig.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-35-154-20-210.ap-south-1.compute.amazonaws.com'
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-14-152 ~]$ sudo su -
[root@ip-172-31-14-152 ~]# hostnamectl set-hostname tomcat-server
[root@ip-172-31-14-152 ~]# bash
[root@tomcat-server ~]# |
```

- Install java in tomcat machine

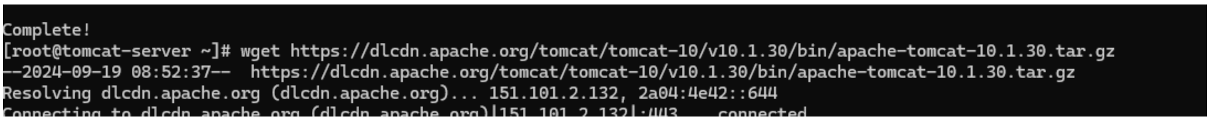


```
[root@ip-172-31-14-152 ~]# bash
[root@tomcat-server ~]# yum install java -y
Last metadata expiration check: 0:03:30 ago on Thu Sep 19 08:48:22 2024.
Dependencies resolved.
==========================================================================================
 Package                                          Architecture          Version
------------------------------------------------------------------------------------------
```



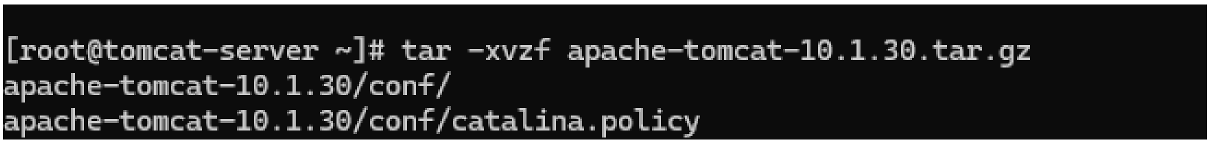
```
[root@tomcat-server ~]# java -version
openjdk version "22.0.2" 2024-07-16
OpenJDK Runtime Environment Corretto-22.0.2.9.1 (build 22.0.2+9-FR)
OpenJDK 64-Bit Server VM Corretto-22.0.2.9.1 (build 22.0.2+9-FR, mixed mode, sharing)
[root@tomcat-server ~]# |
```

#Install Apache tomcat in this machine.



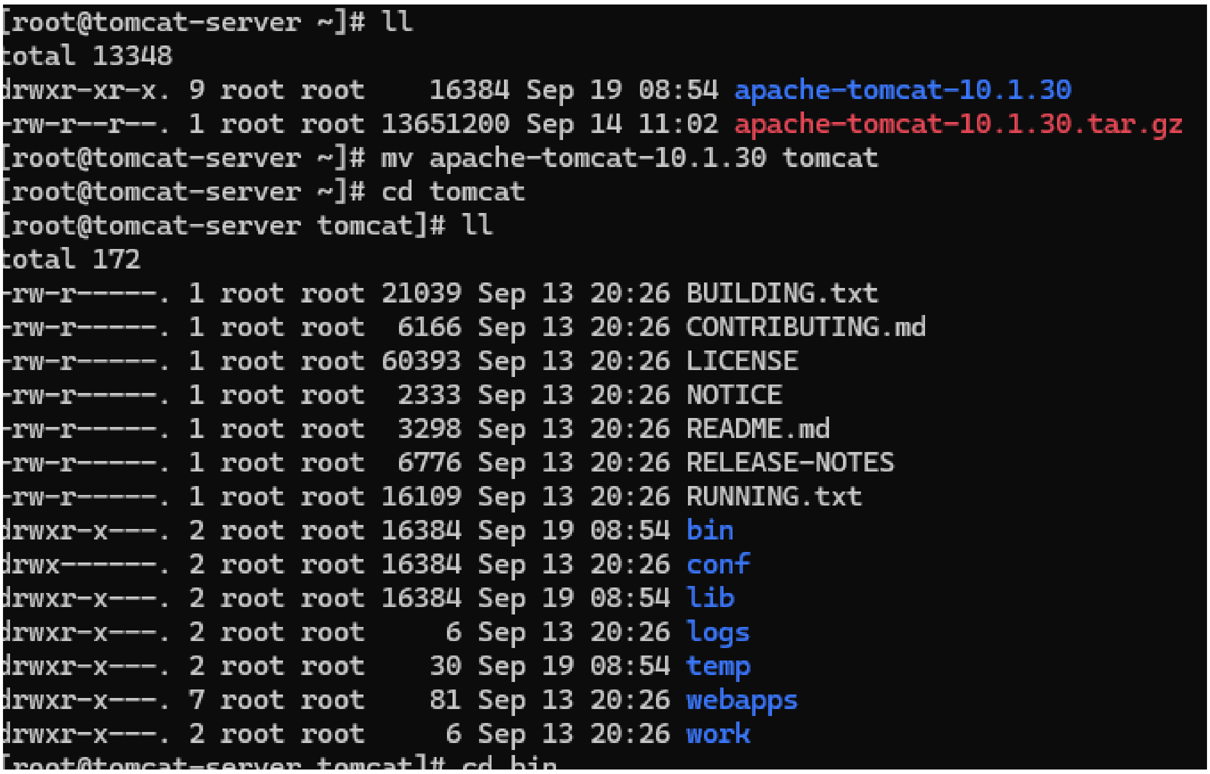
```
Complete!
[root@tomcat-server ~]# wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.30/bin/apache-tomcat-10.1.30.tar.gz
--2024-09-19 08:52:37--  https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.30/bin/apache-tomcat-10.1.30.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
```

- Extract tar file.

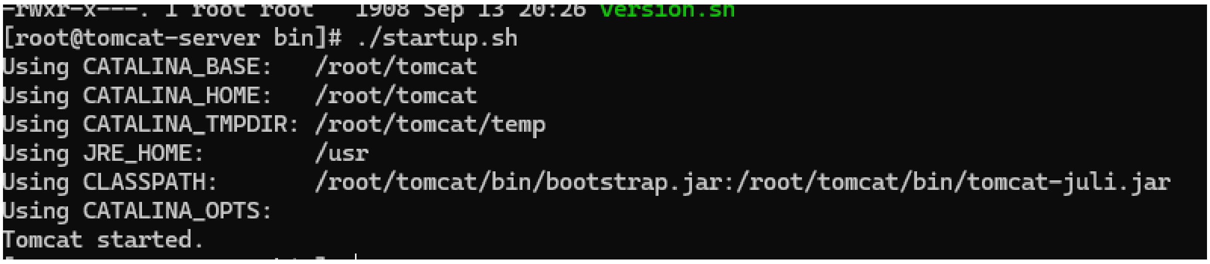


```
[root@tomcat-server ~]# tar -xvzf apache-tomcat-10.1.30.tar.gz
apache-tomcat-10.1.30/conf/
apache-tomcat-10.1.30/conf/catalina.policy
```

- Start Services.

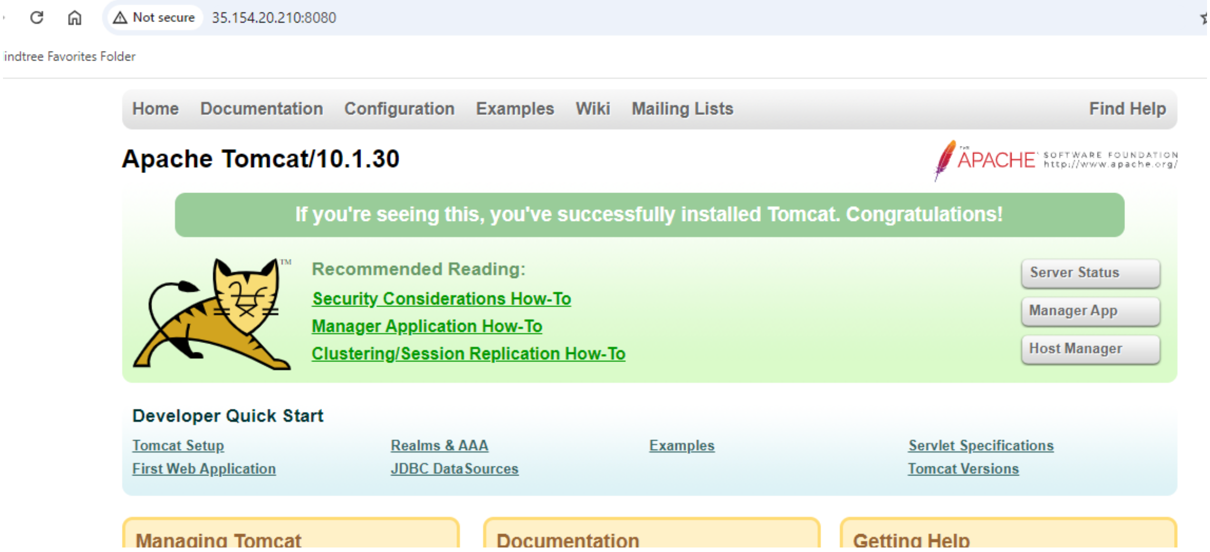


```
[root@tomcat-server ~]# ll
total 13348
drwxr-xr-x. 9 root root    16384 Sep 19 08:54 apache-tomcat-10.1.30
-rw-r--r--. 1 root root 13651200 Sep 14 11:02 apache-tomcat-10.1.30.tar.gz
[root@tomcat-server ~]# mv apache-tomcat-10.1.30 tomcat
[root@tomcat-server ~]# cd tomcat
[root@tomcat-server tomcat]# ll
total 172
-rw-r-----. 1 root root 21039 Sep 13 20:26 BUILDING.txt
-rw-r-----. 1 root root  6166 Sep 13 20:26 CONTRIBUTING.md
-rw-r-----. 1 root root 60393 Sep 13 20:26 LICENSE
-rw-r-----. 1 root root  2333 Sep 13 20:26 NOTICE
-rw-r-----. 1 root root  3298 Sep 13 20:26 README.md
-rw-r-----. 1 root root  6776 Sep 13 20:26 RELEASE-NOTES
-rw-r-----. 1 root root 16109 Sep 13 20:26 RUNNING.txt
drwxr-x---. 2 root root 16384 Sep 19 08:54 bin
drwx------. 2 root root 16384 Sep 13 20:26 conf
drwxr-x---. 2 root root 16384 Sep 19 08:54 lib
drwxr-x---. 2 root root     6 Sep 13 20:26 logs
drwxr-x---. 2 root root    30 Sep 19 08:54 temp
drwxr-x---. 7 root root    81 Sep 13 20:26 webapps
drwxr-x---. 2 root root     6 Sep 13 20:26 work
[root@tomcat-server tomcat]# cd bin
```



```
-rwxr-x---. 1 root root   1908 Sep 13 20:26 version.sh
[root@tomcat-server bin]# ./startup.sh
Using CATALINA_BASE:   /root/tomcat
Using CATALINA_HOME:   /root/tomcat
Using CATALINA_TMPDIR: /root/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /root/tomcat/bin/bootstrap.jar:/root/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

- Tomcat has started using public Ip.



Apache Tomcat/10.1.30

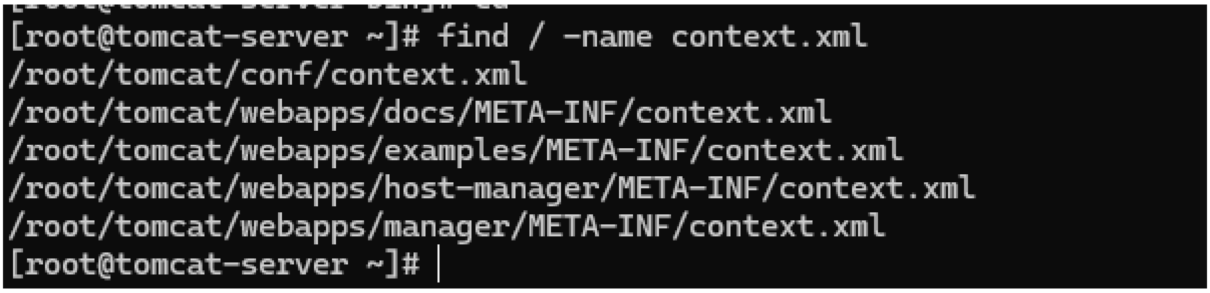
If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

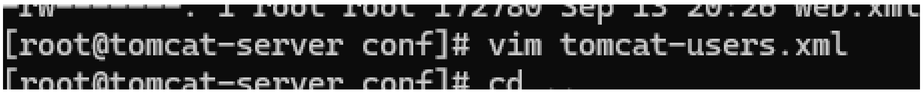
Managing Tomcat
Documentation
Getting Help

- Configure this file.

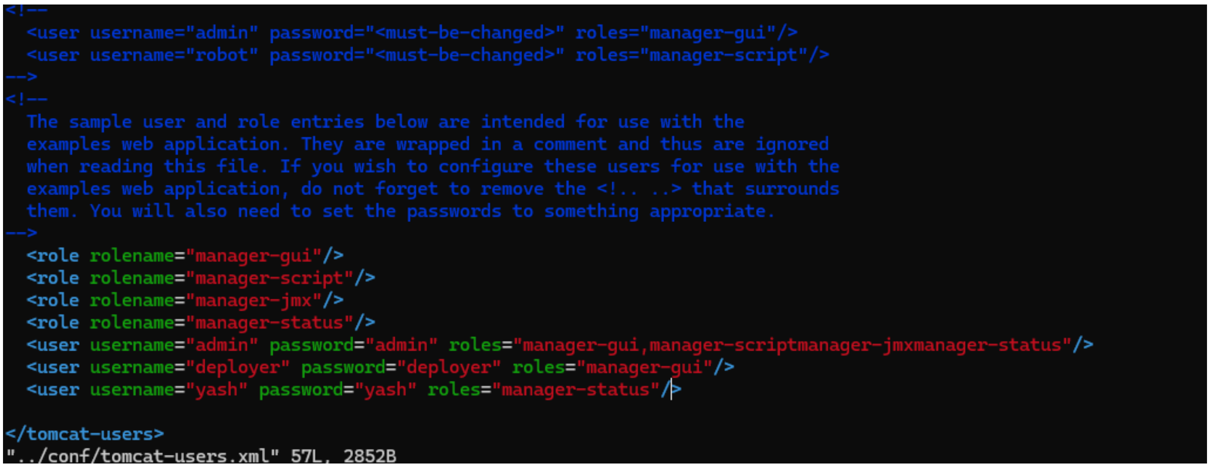


```
[root@tomcat-server bin]# cd
[root@tomcat-server ~]# find / -name context.xml
/root/tomcat/conf/context.xml
/root/tomcat/webapps/docs/META-INF/context.xml
/root/tomcat/webapps/examples/META-INF/context.xml
/root/tomcat/webapps/host-manager/META-INF/context.xml
/root/tomcat/webapps/manager/META-INF/context.xml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

- Add users in tomcat-users.xml file.

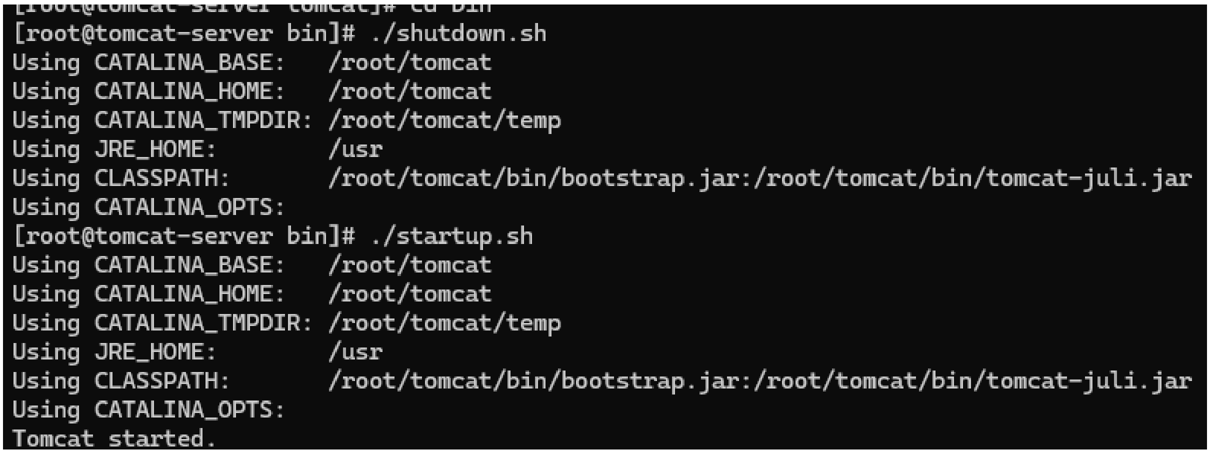


```
-rw-------. 1 root root 172780 Sep 13 20:28 web.xml
[root@tomcat-server conf]# vim tomcat-users.xml
[root@tomcat-server conf]# cd ..
```

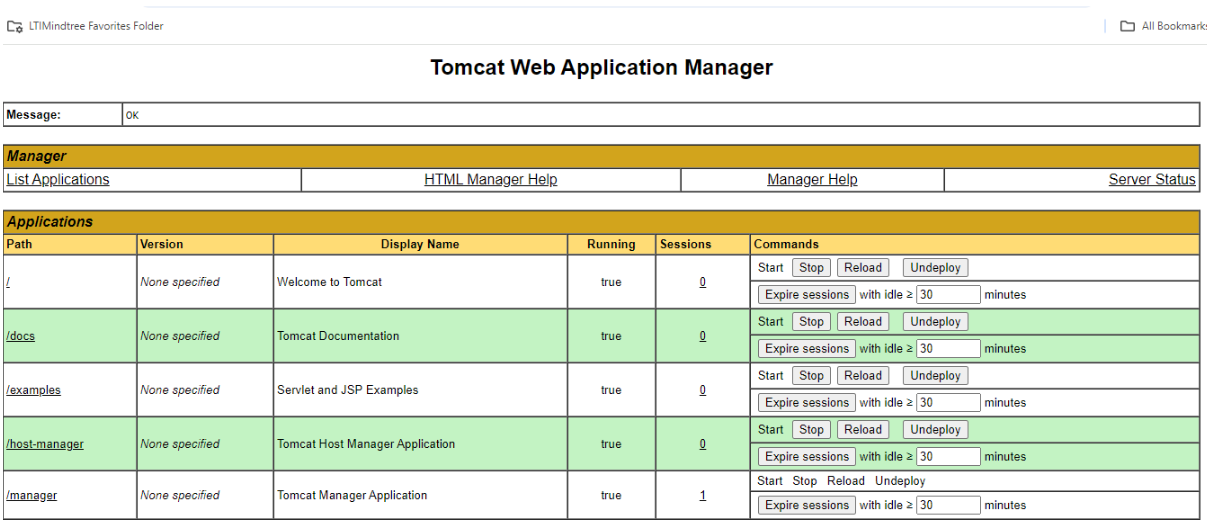


```
<!--
  <user username="admin" password="<must-be-changed>" roles="manager-gui"/>
  <user username="robot" password="<must-be-changed>" roles="manager-script"/>
-->
<!--
  The sample user and role entries below are intended for use with the
  examples web application. They are wrapped in a comment and thus are ignored
  when reading this file. If you wish to configure these users for use with the
  examples web application, do not forget to remove the <!.. ..> that surrounds
  them. You will also need to set the passwords to something appropriate.
-->
  <role rolename="manager-gui"/>
  <role rolename="manager-script"/>
  <role rolename="manager-jmx"/>
  <role rolename="manager-status"/>
  <user username="admin" password="admin" roles="manager-gui,manager-scriptmanager-jmxmanager-status"/>
  <user username="deployer" password="deployer" roles="manager-gui"/>
  <user username="yash" password="yash" roles="manager-status"/>

</tomcat-users>
"../conf/tomcat-users.xml" 57L, 2852B
```

- Restart the tomcat server.



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



LTIMindtree Favorites Folder

All Bookmarks

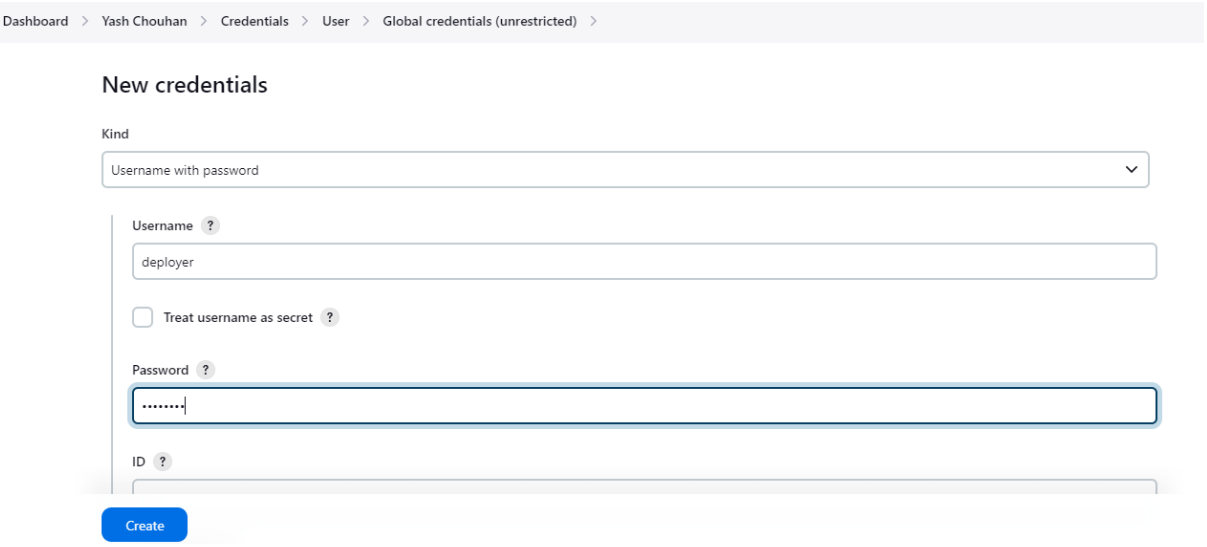
Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

#Now go on Jenkins and create credentials.



Dashboard > Yash Chouhan > Credentials > User > Global credentials (unrestricted) >

New credentials

Kind

Username with password

Username ?

deployer

□ Treat username as secret ?

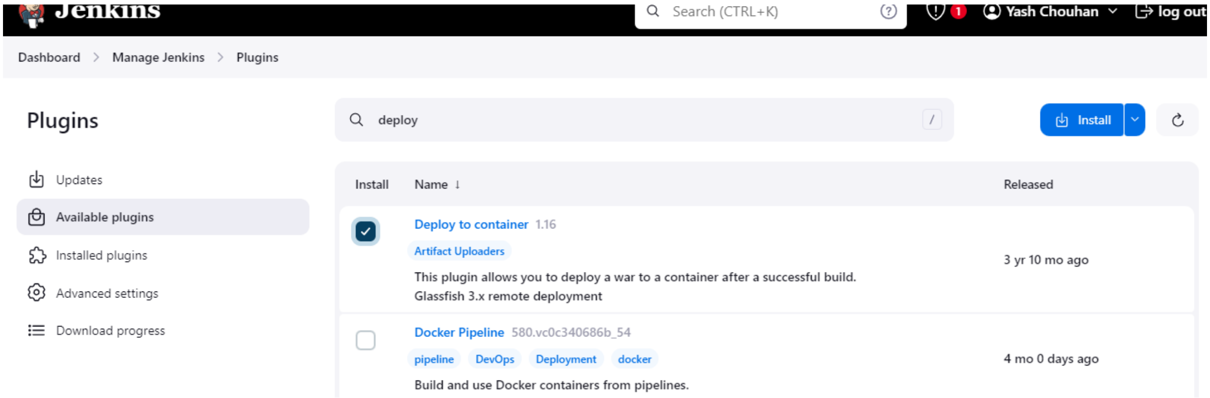
Password ?

••••••••

ID ?

Create

- Install deploy to container plugins for deployment.



Jenkins

Search (CTRL+K)

Yash Chouhan

log out

Dashboard > Manage Jenkins > Plugins

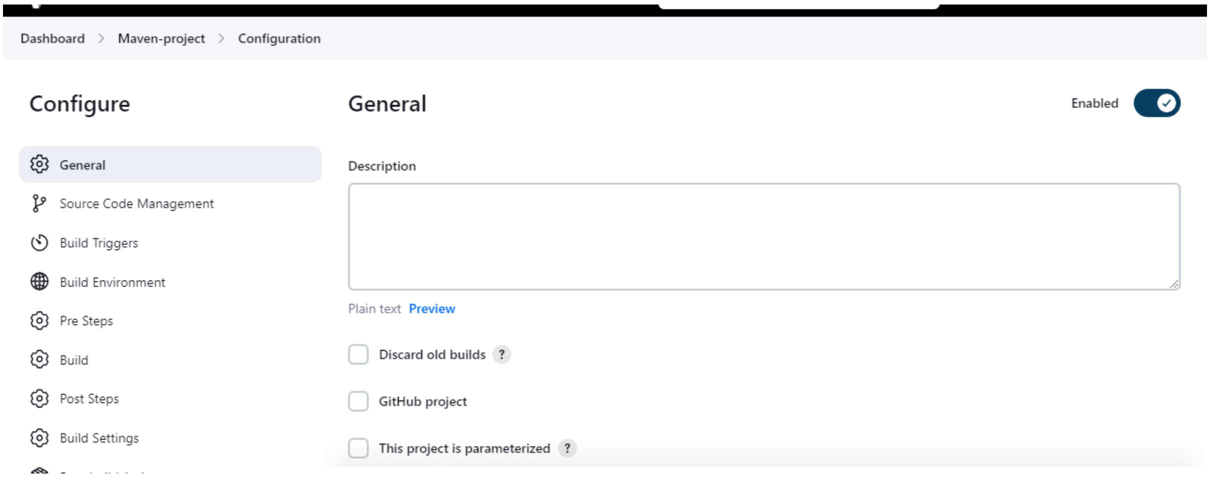
Plugins

deploy

Install

Install	Name ↓	Released
☑	Deploy to container 1.16 Artifact Uploaders This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment	3 yr 10 mo ago
□	Docker Pipeline 580.vc0c340686b_54 pipeline DevOps Deployment docker Build and use Docker containers from pipelines.	4 mo 0 days ago

#Create a job and configure and add credentials in it.



Dashboard > Maven-project > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Pre Steps
- Build
- Post Steps
- Build Settings

General

Enabled

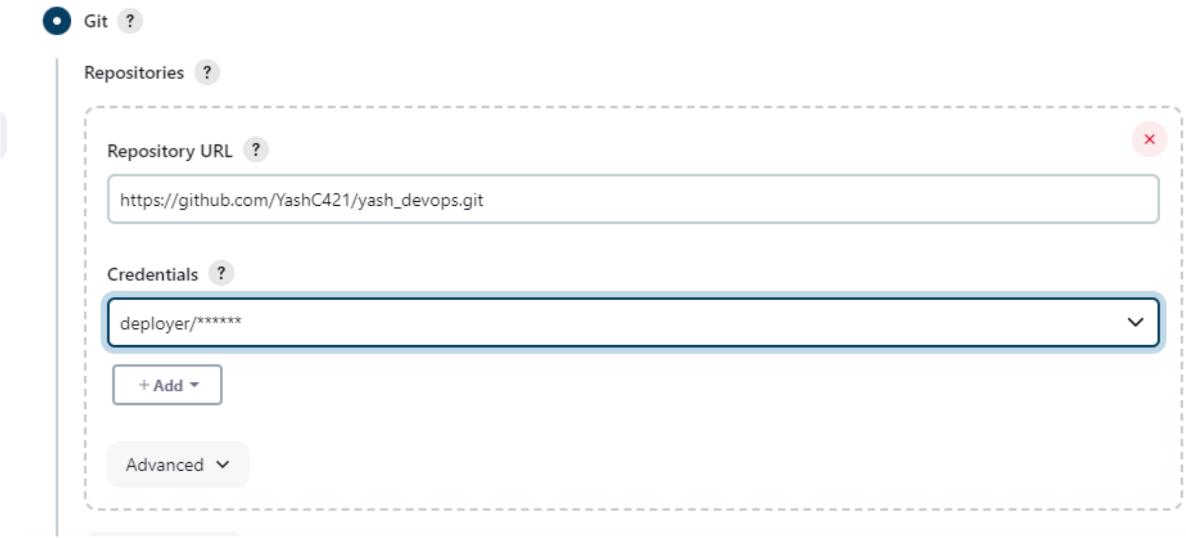
Description

Plain text Preview

□ Discard old builds ?

□ GitHub project

□ This project is parameterized ?



● Git ?

Repositories ?

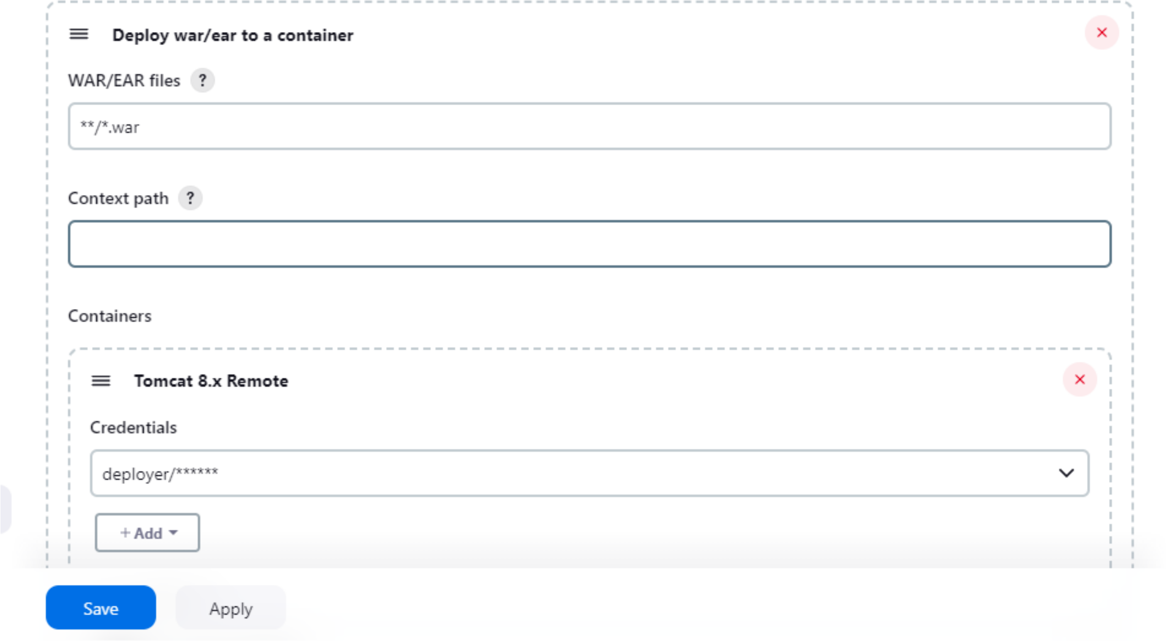
Repository URL ?

https://github.com/YashC421/yash_devops.git

Credentials ?

deployer/******

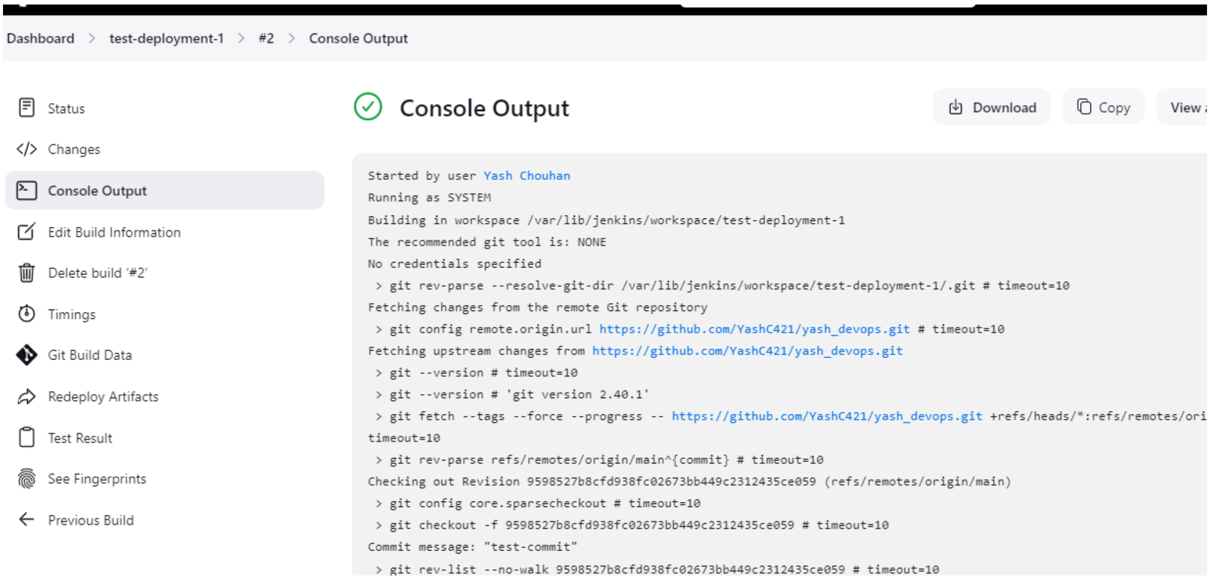
- Add war file configuration.



**/*.war

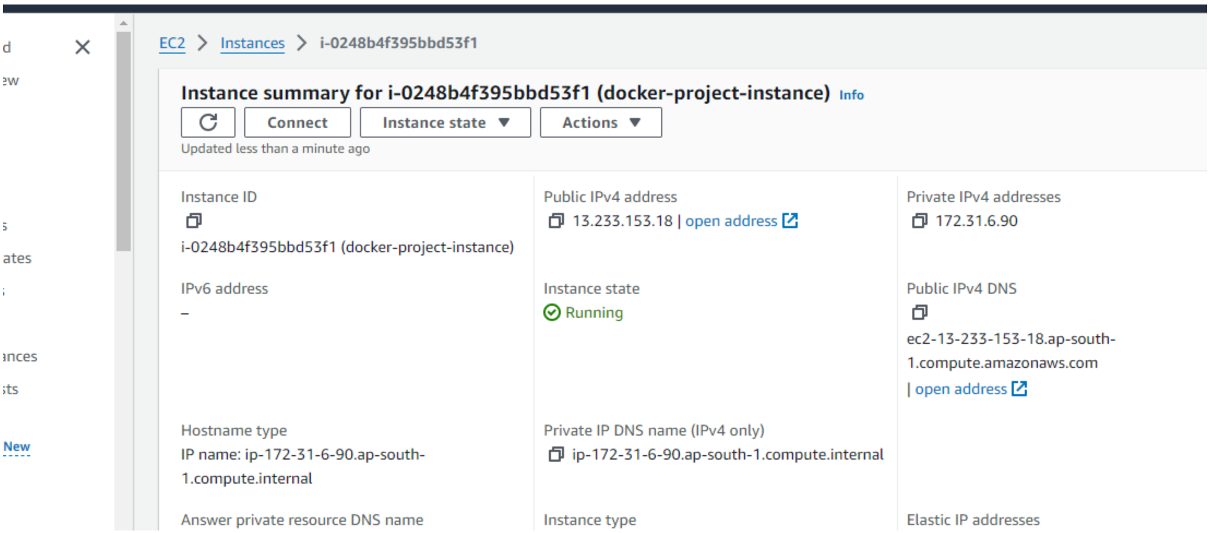
deployer/******

- Apply and save.
- Now build the job and check the console output.



```
Started by user Yash Chouhan
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/test-deployment-1
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/test-deployment-1/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/YashC421/yash_devops.git # timeout=10
Fetching upstream changes from https://github.com/YashC421/yash_devops.git
 > git --version # timeout=10
 > git --version # 'git version 2.40.1'
 > git fetch --tags --force --progress -- https://github.com/YashC421/yash_devops.git +refs/heads/*:refs/remotes/ori
timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 9598527b8cfd938fc02673bb449c2312435ce059 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 9598527b8cfd938fc02673bb449c2312435ce059 # timeout=10
Commit message: "test-commit"
 > git rev-list --no-walk 9598527b8cfd938fc02673bb449c2312435ce059 # timeout=10
```

# Create new Instance to start Docker-server



Instance summary for i-0248b4f395bbd53f1 (docker-project-instance)

Updated less than a minute ago

Instance ID: i-0248b4f395bbd53f1 (docker-project-instance)

Public IPv4 address: 13.233.153.18 | open address

Private IPv4 addresses: 172.31.6.90

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-13-233-153-18.ap-south-1.compute.amazonaws.com | open address

Hostname type: IP name: ip-172-31-6-90.ap-south-1.compute.internal

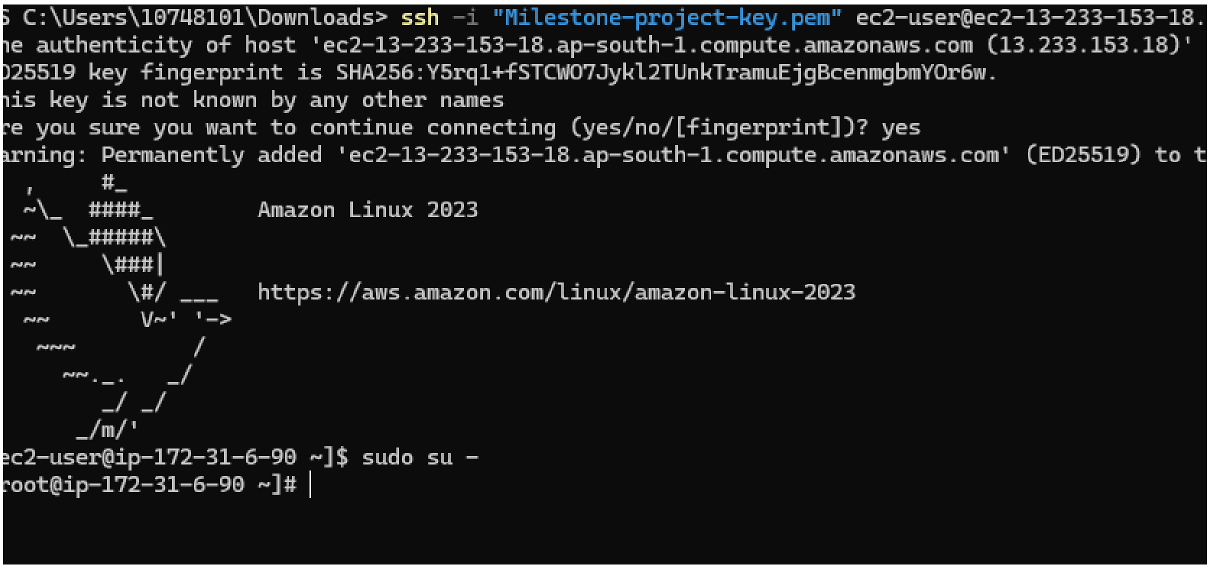
Private IP DNS name (IPv4 only): ip-172-31-6-90.ap-south-1.compute.internal

Answer private resource DNS name

Instance type

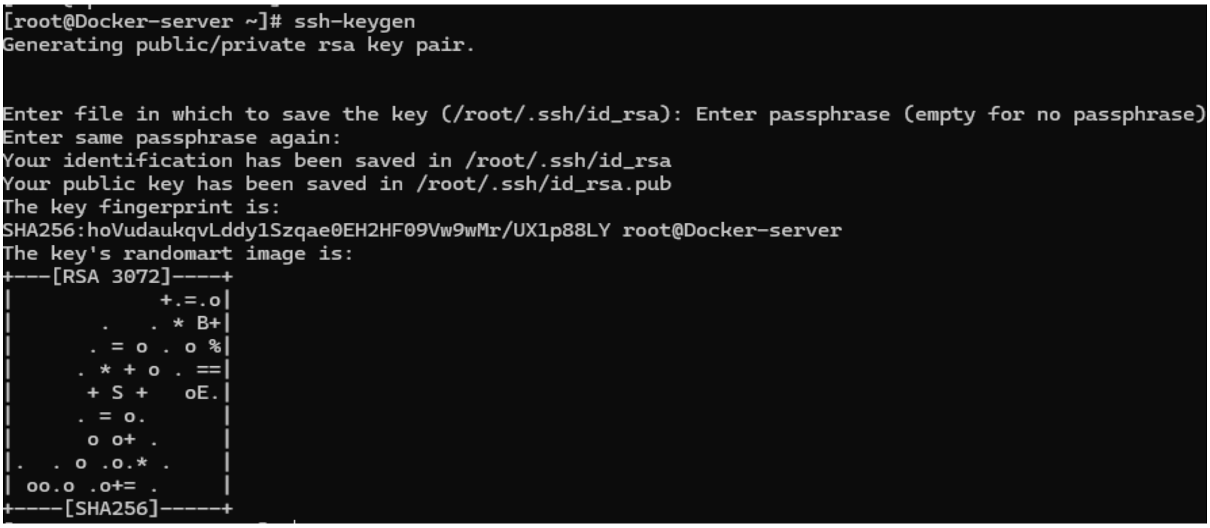
Elastic IP addresses

- Connect with docker-instance.

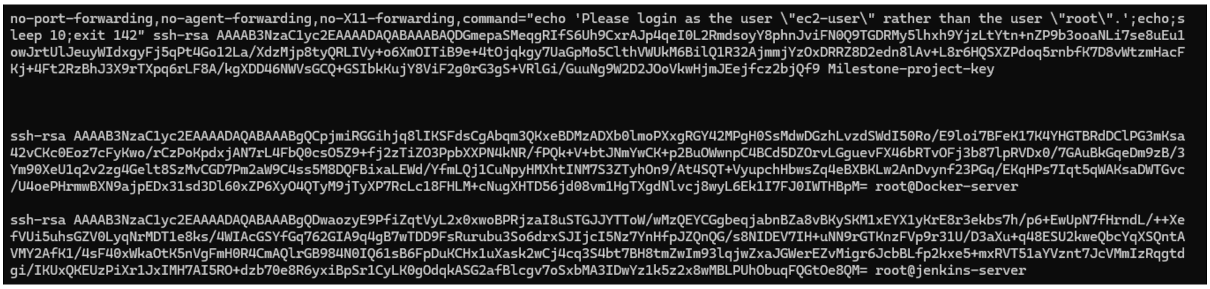


```
S C:\Users\10748101\Downloads> ssh -i "Milestone-project-key.pem" ec2-user@ec2-13-233-153-18.
he authenticity of host 'ec2-13-233-153-18.ap-south-1.compute.amazonaws.com (13.233.153.18)'
D25519 key fingerprint is SHA256:Y5rq1+fSTCWO7Jykl2TUnkTramuEjgBcenmgbmYOr6w.
his key is not known by any other names
re you sure you want to continue connecting (yes/no/[fingerprint])? yes
arning: Permanently added 'ec2-13-233-153-18.ap-south-1.compute.amazonaws.com' (ED25519) to t
  ,     #_
  ~\_  ####_        Amazon Linux 2023
 ~~  \_#####\
 ~~     \###|
 ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
  ~~       V~' '->
   ~~~         /
     ~~._.   _/
        _/ _/
      _/m/'
ec2-user@ip-172-31-6-90 ~]$ sudo su -
root@ip-172-31-6-90 ~]# |
```

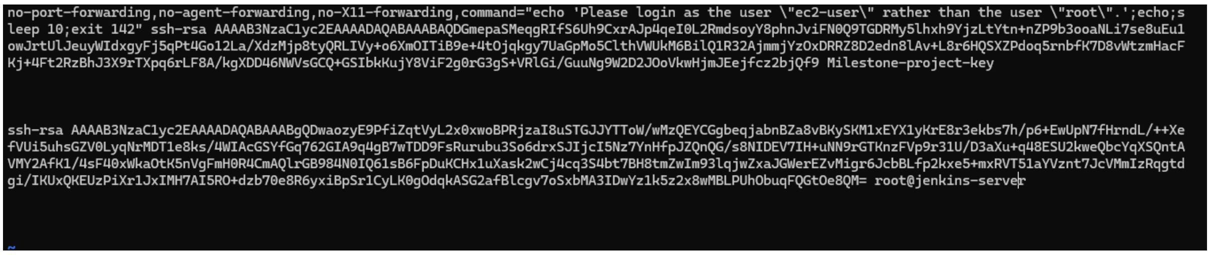
#Generate ssh keys in Jenkins machine and docker machine. Exchange the ssh keys with each other and copy also in their own authorized keys.



```
[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:hoVudaukqvLddy1Szqae0EH2HF09Vw9wMr/UX1p88LY root@Docker-server
The key's randomart image is:
+---[RSA 3072]----+
|            +.=.o|
|       .    . * B+|
|      . = o . o %|
|     . * + o . ==|
|      + S +   oE.|
|     . = o.      |
|      o o+ .     |
|.   . o .o.* .   |
| oo.o .o+= .     |
+----[SHA256]-----+
```

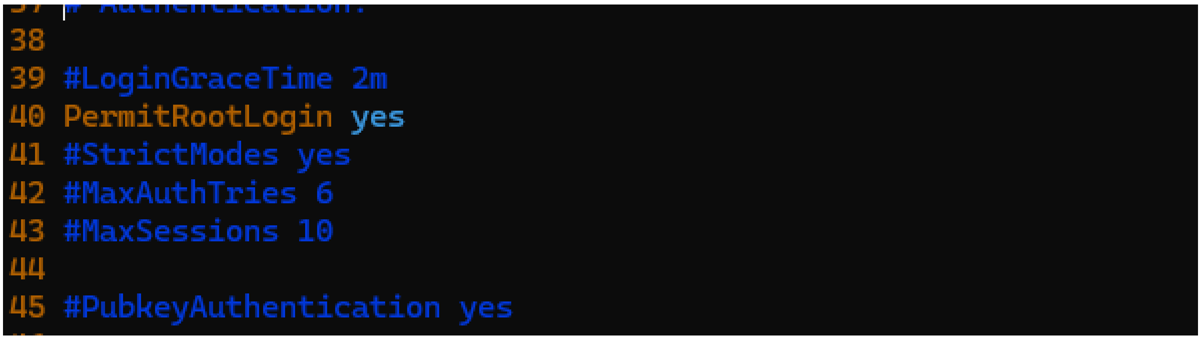


```
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\".';echo;s
leep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDGmepaSMeqgRIfS6Uh9CxrAJp4qeI0L2RmdsoyY8phnJviFN0Q9TGDRMy5lhxh9YjzLtYtn+nZP9b3ooaNLi7se8uEu1
owJrtUlJeuyWIdxgyFj5qPt4Go12La/XdzMjp8tyQRLIVy+o6XmOITiB9e+4tOjqkgy7UaGpMo5ClthVWUkM6BilQ1R32AjmmjYzOxDRRZ8D2edn8lAv+L8r6HQSXZPdoq5rnbfK7D8vWtzmHacF
Kj+4Ft2RzBhJ3X9rTXpq6rLF8A/kgXDD46NWVsGCQ+GSIbkKujY8ViF2g0rG3gS+VRlGi/GuuNg9W2D2JOoVkwHjmJEejfcz2bjQf9 Milestone-project-key



ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCpjmiRGGihjq8lIKSFdsCgAbqm3QKxeBDMzADXb0lmoPXxgRGY42MPgH0SsMdwDGzhLvzdSWdI50Ro/E9loi7BFeK17K4YHGTBRdDClPG3mKsa
42vCKc0Eoz7cFyKwo/rCzPoKpdxjAN7rL4FbQ0csO5Z9+fj2zTiZO3PpbXXPN4kNR/fPQk+V+btJNmYwCK+p2BuOWwnpC4BCd5DZOrvLGguevFX46bRTvOFj3b87lpRVDx0/7GAuBkGqeDm9zB/3
Ym90XeU1q2v2zg4Gelt8SzMvCGD7Pm2aW9C4ss5M8DQFBixaLEWd/YfmLQj1CuNpyHMXhtINM7S3ZTyhOn9/At4SQT+VyupchHbwsZq4eBXBKLw2AnDvynf23PGq/EKqHPs7Iqt5qWAKsaDWTGvc
/U4oePHrmwBXN9ajpEDx31sd3Dl60xZP6XyO4QTyM9jTyXP7RcLc18FHLM+cNugXHTD56jd08vm1HgTXgdNlvcj8wyL6Ek1I7FJ0IWTHBpM= root@Docker-server

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDwaozyE9PfiZqtVyL2x0xwoBPRjzaI8uSTGJJYTToW/wMzQEYCGgbeqjabnBZa8vBKySKM1xEYX1yKrE8r3ekbs7h/p6+EwUpN7fHrndL/++Xe
fVUi5uhsGZV0LyqNrMDT1e8ks/4WIAcGSYfGq762GIA9q4gB7wTDD9FsRurubu3So6drxSJIjcI5Nz7YnHfpJZQnQG/s8NIDEV7IH+uNN9rGTKnzFVp9r31U/D3aXu+q48ESU2kweQbcYqXSQntA
VMY2AfK1/4sF40xWkaOtK5nVgFmH0R4CmAQlrGB984N0IQ61sB6FpDuKCHx1uXask2wCj4cq3S4bt7BH8tmZwIm93lqjwZxaJGWerEZvMigr6JcbBLfp2kxe5+mxRVT51aYVznt7JcVMmIzRqgtd
gi/IKUxQKEUzPiXr1JxIMH7AI5RO+dzb70e8R6yxiBpSr1CyLK0gOdqkASG2afBlcgv7oSxbMA3IDwYz1k5z2x8wMBLPUhObuqFQGtOe8QM= root@jenkins-server
```

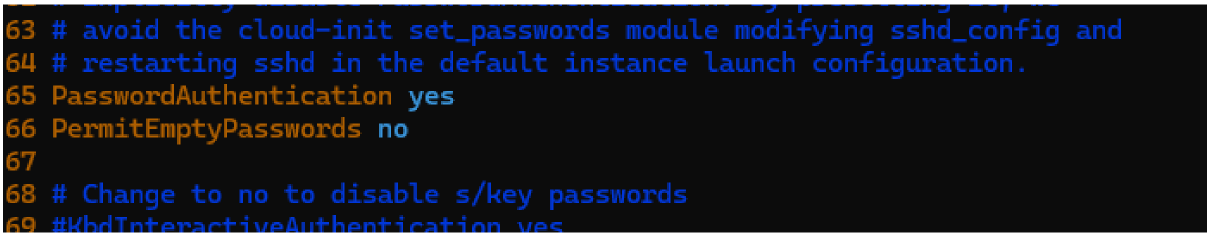


```
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\".';echo;s
leep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDGmepaSMeqgRIfS6Uh9CxrAJp4qeI0L2RmdsoyY8phnJviFN0Q9TGDRMy5lhxh9YjzLtYtn+nZP9b3ooaNLi7se8uEu1
owJrtUlJeuyWIdxgyFj5qPt4Go12La/XdzMjp8tyQRLIVy+o6XmOITiB9e+4tOjqkgy7UaGpMo5ClthVWUkM6BilQ1R32AjmmjYzOxDRRZ8D2edn8lAv+L8r6HQSXZPdoq5rnbfK7D8vWtzmHacF
Kj+4Ft2RzBhJ3X9rTXpq6rLF8A/kgXDD46NWVsGCQ+GSIbkKujY8ViF2g0rG3gS+VRlGi/GuuNg9W2D2JOoVkwHjmJEejfcz2bjQf9 Milestone-project-key



ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDwaozyE9PfiZqtVyL2x0xwoBPRjzaI8uSTGJJYTToW/wMzQEYCGgbeqjabnBZa8vBKySKM1xEYX1yKrE8r3ekbs7h/p6+EwUpN7fHrndL/++Xe
fVUi5uhsGZV0LyqNrMDT1e8ks/4WIAcGSYfGq762GIA9q4gB7wTDD9FsRurubu3So6drxSJIjcI5Nz7YnHfpJZQnQG/s8NIDEV7IH+uNN9rGTKnzFVp9r31U/D3aXu+q48ESU2kweQbcYqXSQntA
VMY2AfK1/4sF40xWkaOtK5nVgFmH0R4CmAQlrGB984N0IQ61sB6FpDuKCHx1uXask2wCj4cq3S4bt7BH8tmZwIm93lqjwZxaJGWerEZvMigr6JcbBLfp2kxe5+mxRVT51aYVznt7JcVMmIzRqgtd
gi/IKUxQKEUzPiXr1JxIMH7AI5RO+dzb70e8R6yxiBpSr1CyLK0gOdqkASG2afBlcgv7oSxbMA3IDwYz1k5z2x8wMBLPUhObuqFQGtOe8QM= root@jenkins-server


~
```

- Permit root login and password authentication yes in Jenkins machine and docker machine.

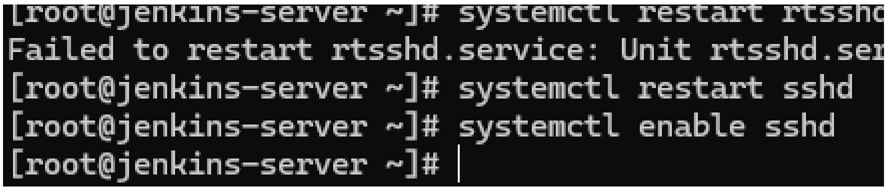


```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes
```



```
# avoid the cloud-init set_passwords module modifying sshd_config and
# restarting sshd in the default instance launch configuration.
PasswordAuthentication yes
PermitEmptyPasswords no

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

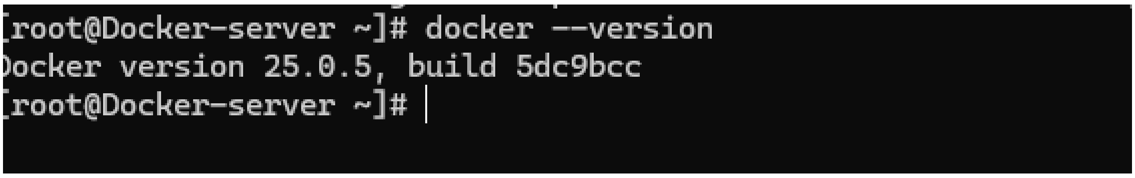
- Restart services of sshd in Jenkins and docker machines.



```
[root@jenkins-server ~]# systemctl restart rtsshd
Failed to restart rtsshd.service: Unit rtsshd.ser
[root@jenkins-server ~]# systemctl restart sshd
[root@jenkins-server ~]# systemctl enable sshd
[root@jenkins-server ~]#
```

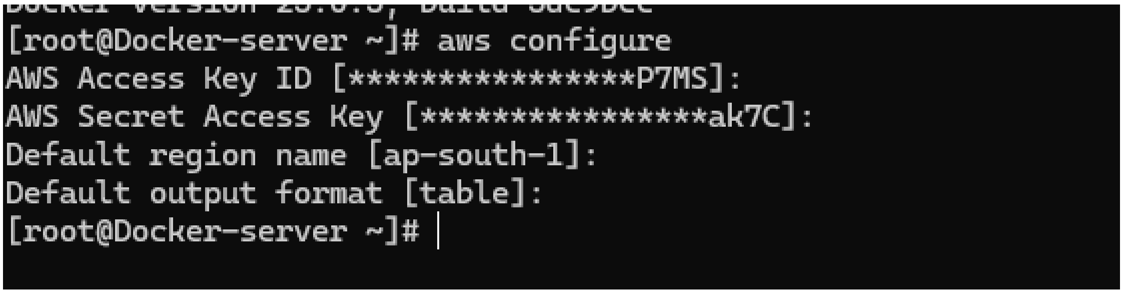
#Install Docker .

- Yum install docker -y



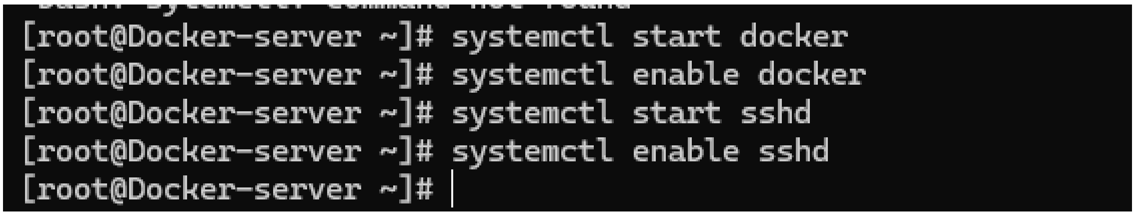
```
[root@Docker-server ~]# docker --version
Docker version 25.0.5, build 5dc9bcc
[root@Docker-server ~]# |
```

- Configure AWSCLI in docker machine.



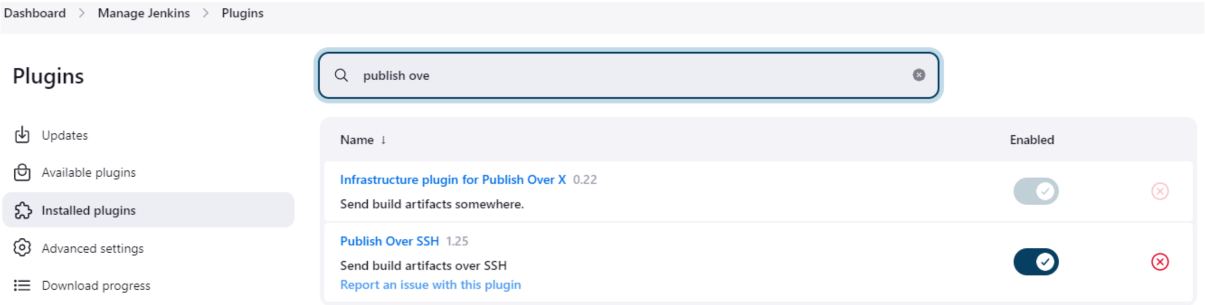
```
[root@Docker-server ~]# aws configure
AWS Access Key ID [****************P7MS]:
AWS Secret Access Key [****************ak7C]:
Default region name [ap-south-1]:
Default output format [table]:
[root@Docker-server ~]# |
```

- Start docker and restart sshd services.



```
[root@Docker-server ~]# systemctl start docker
[root@Docker-server ~]# systemctl enable docker
[root@Docker-server ~]# systemctl start sshd
[root@Docker-server ~]# systemctl enable sshd
[root@Docker-server ~]# |
```

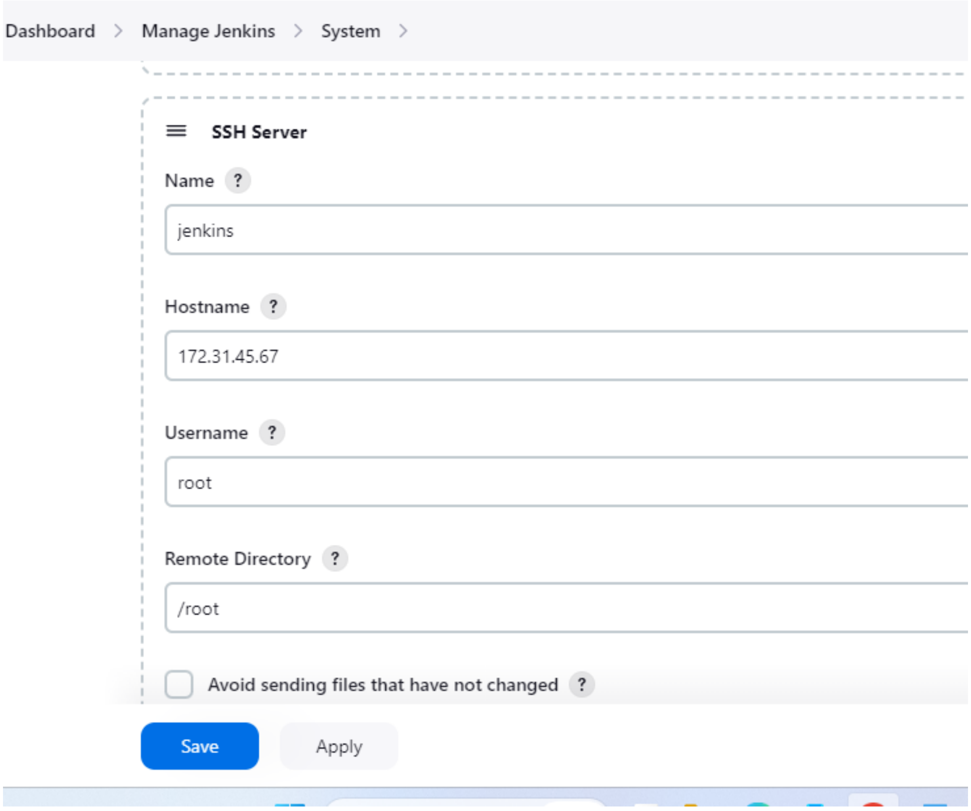
#Now go on Jenkins and install publish over ssh.



publish ove

Name ↓	Enabled
Infrastructure plugin for Publish Over X 0.22 Send build artifacts somewhere.	✓
Publish Over SSH 1.25 Send build artifacts over SSH Report an issue with this plugin	✓

- Add the private key of Jenkins in ssh server and add Jenkins and docker configuration.



SSH Server

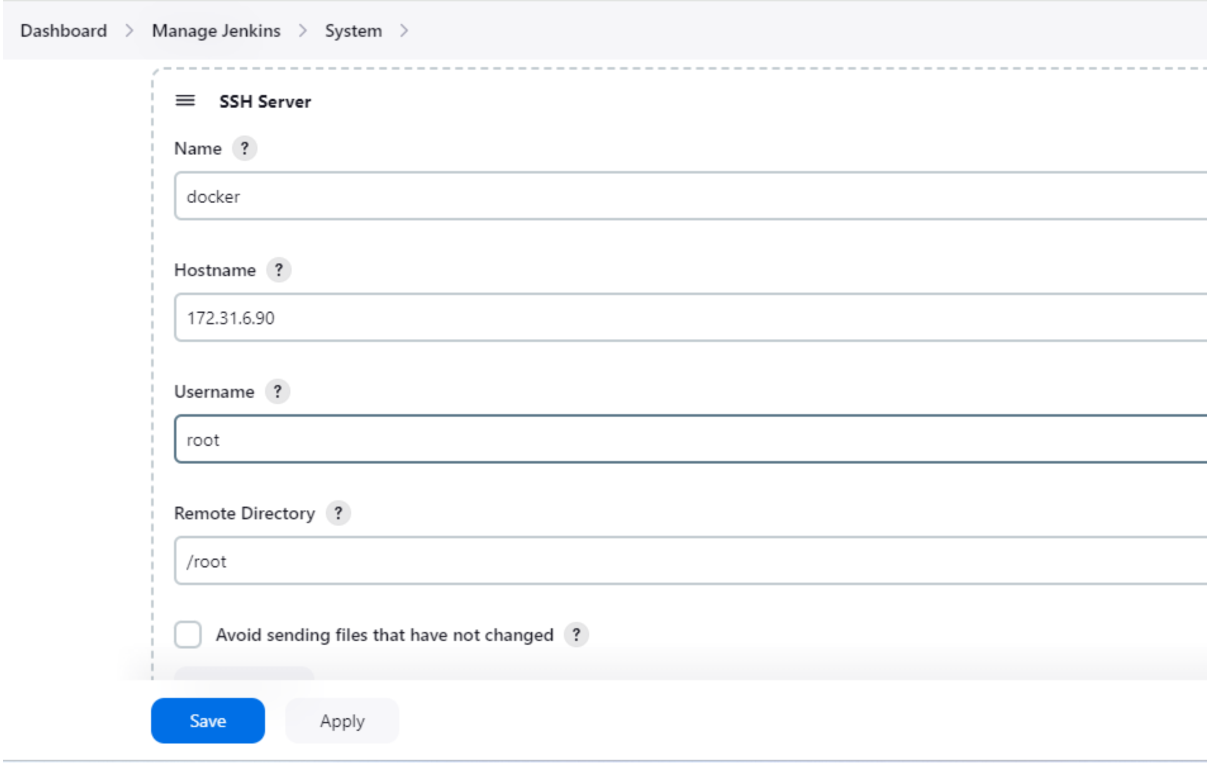
Name: jenkins

Hostname: 172.31.45.67

Username: root

Remote Directory: /root

□ Avoid sending files that have not changed



SSH Server

Name: docker

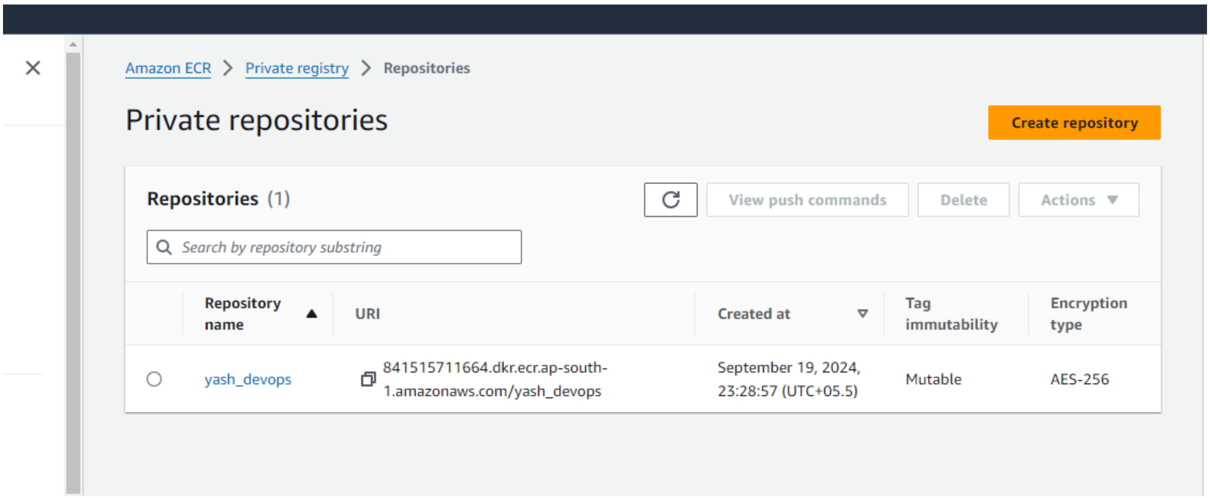
Hostname: 172.31.6.90

Username: root

Remote Directory: /root

□ Avoid sending files that have not changed

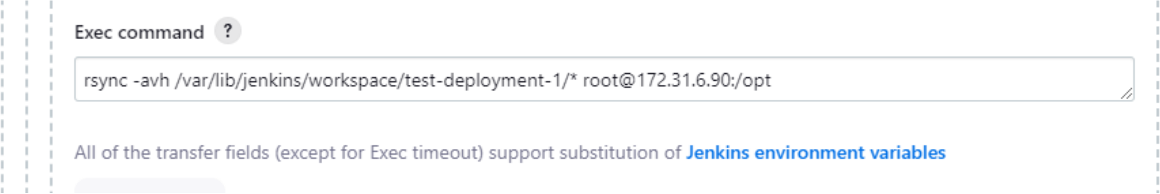
Go on AWS and create an ECR corresponding to a user. Here we get our all the latest images of our project which are pushed by docker. After this Kubernetes pull the image from ECR.



	Repository name ▲	URI	Created at ▽	Tag immutability	Encryption type
○	yash_devops	841515711664.dkr.ecr.ap-south-1.amazonaws.com/yash_devops	September 19, 2024, 23:28:57 (UTC+05.5)	Mutable	AES-256

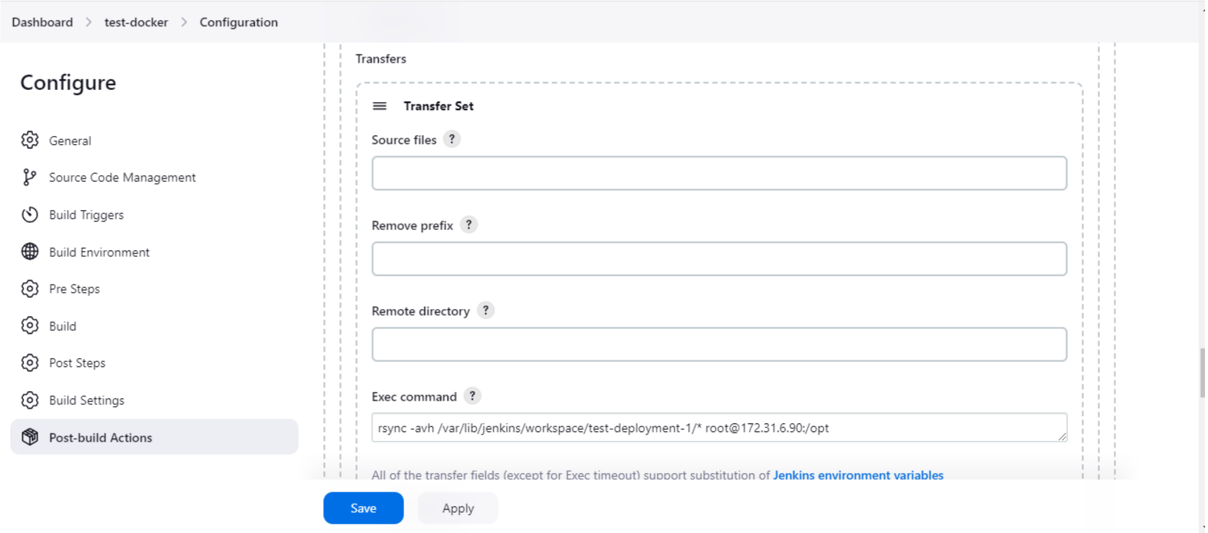
#Configure Job with post build actions.

- Add Exec Command which are automatic execute in Jenkins server.


Exec command ?

```
rsync -avh /var/lib/jenkins/workspace/test-deployment-1/* root@172.31.6.90:/opt
```

All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables



Source files ?

Remove prefix ?

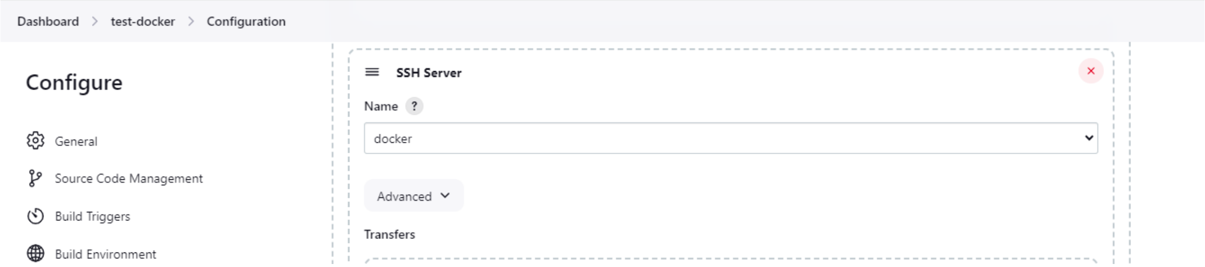
Remote directory ?

Exec command ?

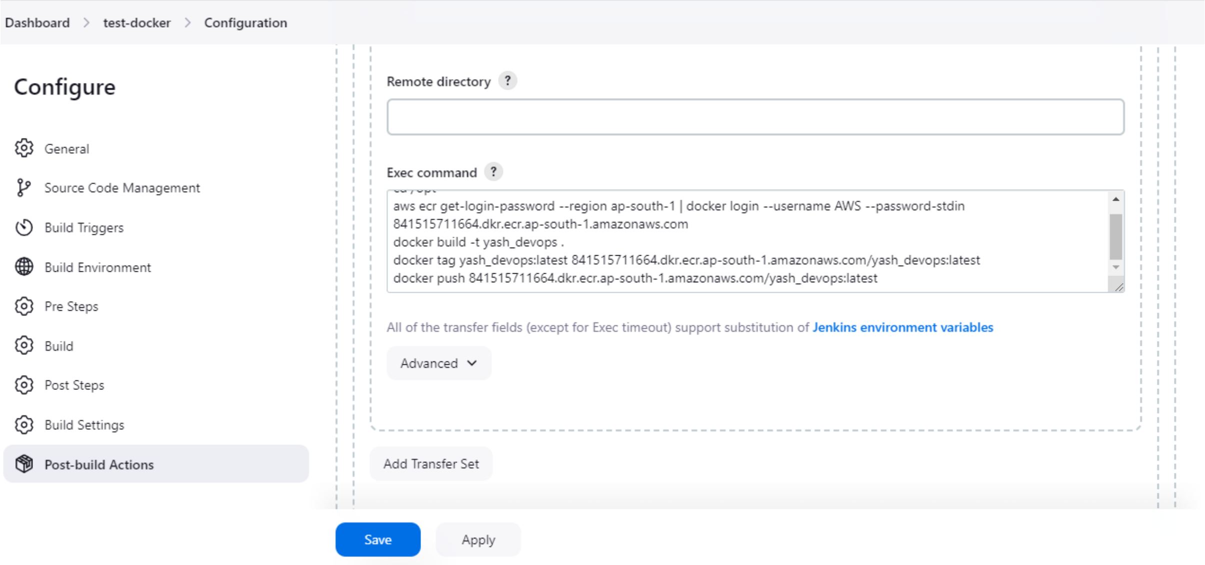
```
rsync -avh /var/lib/jenkins/workspace/test-deployment-1/* root@172.31.6.90:/opt
```

All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables

- Add Exec Command to automate docker. This command is going to connect ECR to docker to push images from docker.



Name: docker



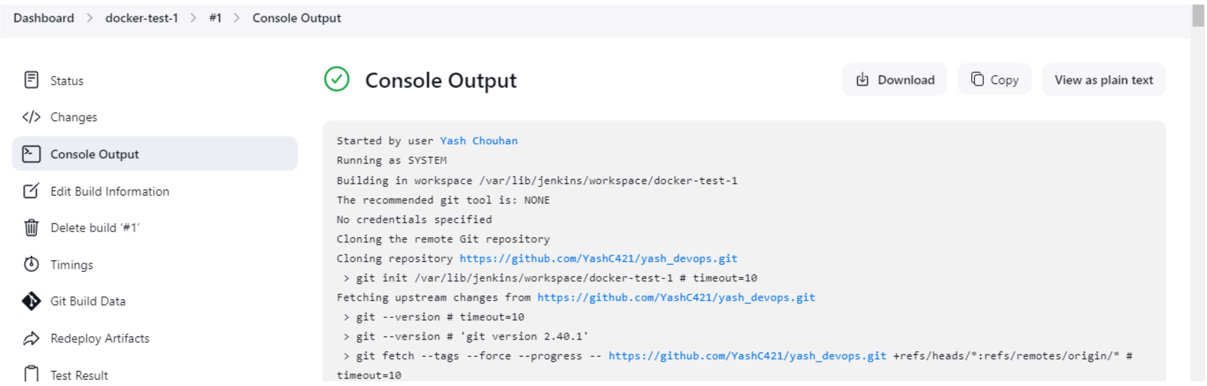
Remote directory:

Exec command:

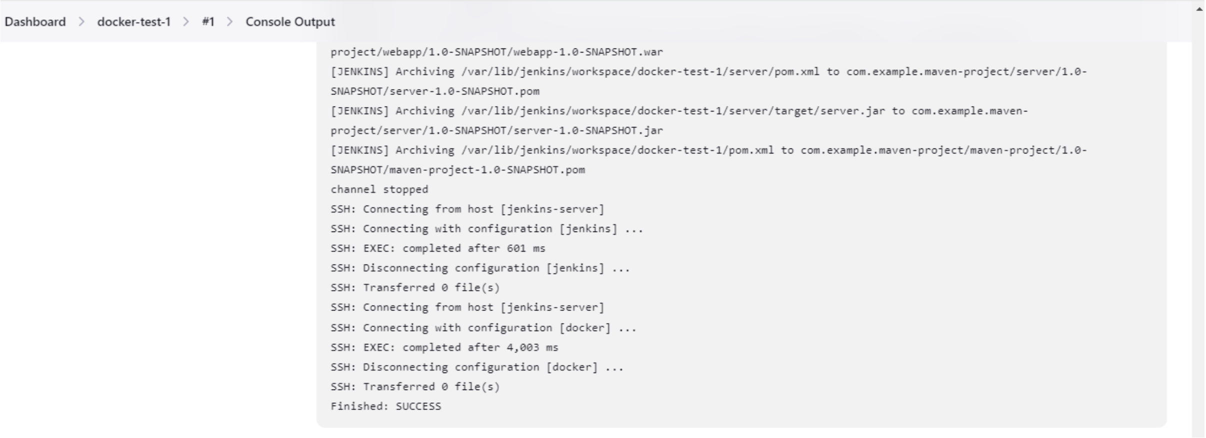
```
aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin
841515711664.dkr.ecr.ap-south-1.amazonaws.com
docker build -t yash_devops .
docker tag yash_devops:latest 841515711664.dkr.ecr.ap-south-1.amazonaws.com/yash_devops:latest
docker push 841515711664.dkr.ecr.ap-south-1.amazonaws.com/yash_devops:latest
```

All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables

- Now apply and save all the changes and build the job.



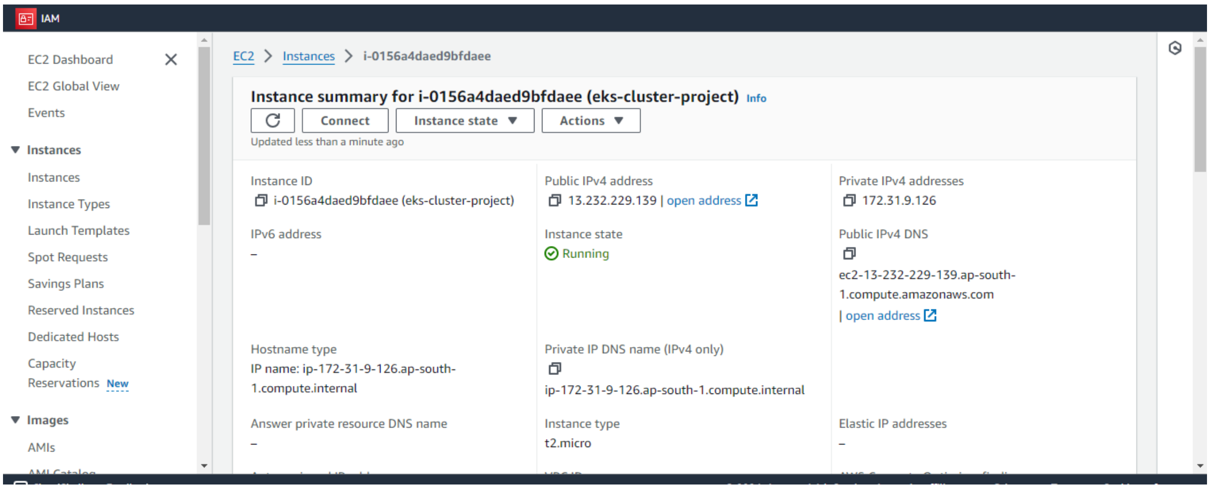
```
Started by user Yash Chouhan
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/docker-test-1
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/YashC421/yash_devops.git
 > git init /var/lib/jenkins/workspace/docker-test-1 # timeout=10
Fetching upstream changes from https://github.com/YashC421/yash_devops.git
 > git --version # timeout=10
 > git --version # 'git version 2.40.1'
 > git fetch --tags --force --progress -- https://github.com/YashC421/yash_devops.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
```



```
project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/docker-test-1/server/pom.xml to com.example.maven-project/server/1.0-
SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/docker-test-1/server/target/server.jar to com.example.maven-
project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/docker-test-1/pom.xml to com.example.maven-project/maven-project/1.0-
SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
channel stopped
SSH: Connecting from host [jenkins-server]
SSH: Connecting with configuration [jenkins] ...
SSH: EXEC: completed after 601 ms
SSH: Disconnecting configuration [jenkins] ...
SSH: Transferred 0 file(s)
SSH: Connecting from host [jenkins-server]
SSH: Connecting with configuration [docker] ...
SSH: EXEC: completed after 4,003 ms
SSH: Disconnecting configuration [docker] ...
SSH: Transferred 0 file(s)
Finished: SUCCESS
```

#Start with Kubernetes.

- Create an instance to create EKS cluster.



Instance summary for i-0156a4daed9bfdaee (eks-cluster-project)

Updated less than a minute ago

Instance ID: i-0156a4daed9bfdaee (eks-cluster-project)

Public IPv4 address: 13.232.229.139 | open address

Private IPv4 addresses: 172.31.9.126

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-13-232-229-139.ap-south-1.compute.amazonaws.com | open address

Hostname type: IP name: ip-172-31-9-126.ap-south-1.compute.internal

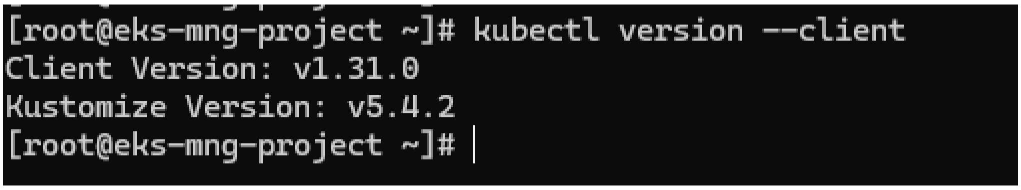
Private IP DNS name (IPv4 only): ip-172-31-9-126.ap-south-1.compute.internal

Answer private resource DNS name: –

Instance type: t2.micro

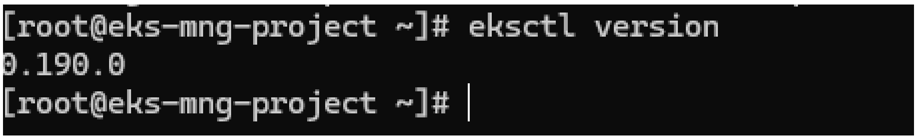
Elastic IP addresses: –

- create IAM role and attach policy ECRfull access,EKSpolicy and IAM full access.
- Configure AWS with credentials.
- Install kubectl.



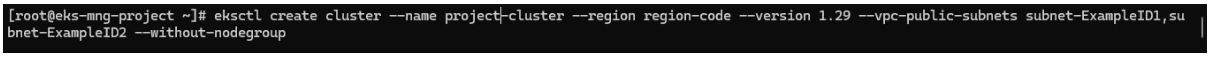
```
[root@eks-mng-project ~]# kubectl version --client
Client Version: v1.31.0
Kustomize Version: v5.4.2
[root@eks-mng-project ~]#
```

- Install eksctl.



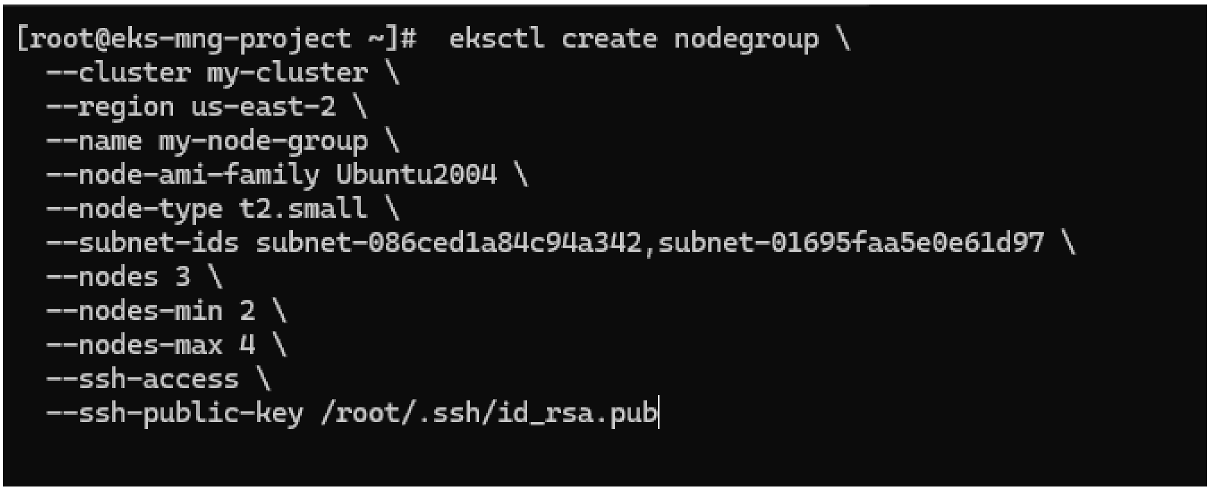
```
[root@eks-mng-project ~]# eksctl version
0.190.0
[root@eks-mng-project ~]# 
```

- Create EKS cluster.



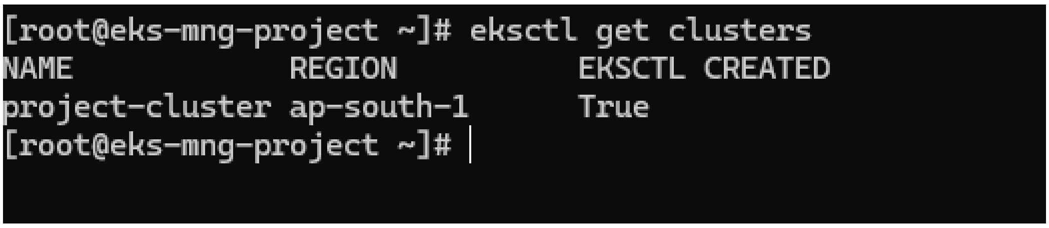
```
[root@eks-mng-project ~]# eksctl create cluster --name project-cluster --region region-code --version 1.29 --vpc-public-subnets subnet-ExampleID1,su
bnet-ExampleID2 --without-nodegroup
```

- Create a Node Group.



```
[root@eks-mng-project ~]#  eksctl create nodegroup \
  --cluster my-cluster \
  --region us-east-2 \
  --name my-node-group \
  --node-ami-family Ubuntu2004 \
  --node-type t2.small \
  --subnet-ids subnet-086ced1a84c94a342,subnet-01695faa5e0e61d97 \
  --nodes 3 \
  --nodes-min 2 \
  --nodes-max 4 \
  --ssh-access \
  --ssh-public-key /root/.ssh/id_rsa.pub
```

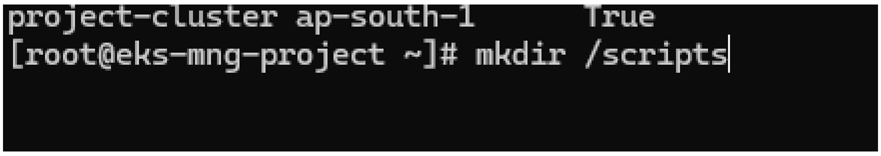
- EKS Cluster is created. for fetch clusters.



```
[root@eks-mng-project ~]# eksctl get clusters
NAME            REGION          EKSCTL CREATED
project-cluster ap-south-1      True
[root@eks-mng-project ~]# 
```

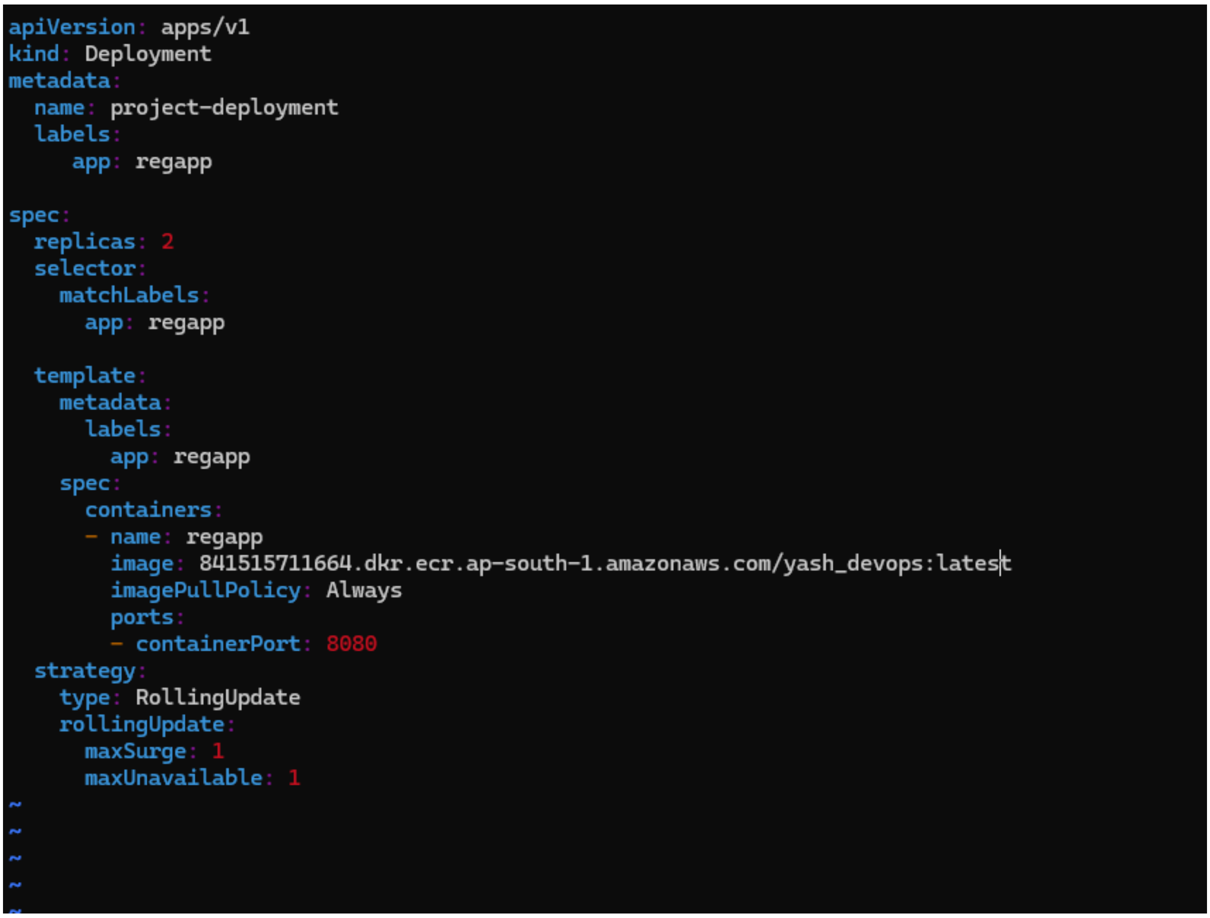
#Now we have to create deployment and services to deploy our ECR images in Kubernetes cluster.

- Create a working directory.



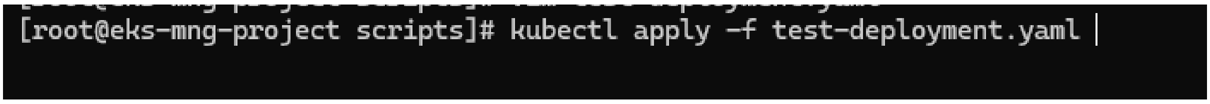
```
project-cluster ap-south-1      True
[root@eks-mng-project ~]# mkdir /scripts
```

- Create deployment file with. yaml extension.

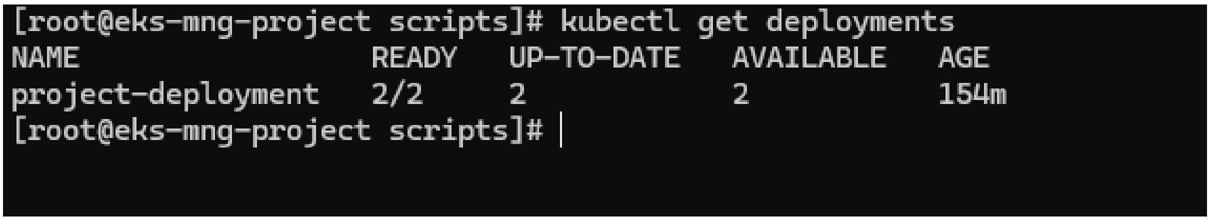


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: project-deployment
  labels:
     app: regapp

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

  template:
    metadata:
      labels:
        app: regapp
    spec:
      containers:
      - name: regapp
        image: 841515711664.dkr.ecr.ap-south-1.amazonaws.com/yash_devops:latest
        imagePullPolicy: Always
        ports:
        - containerPort: 8080
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
~
~
~
~
```

- Execute a command to create a deployment.

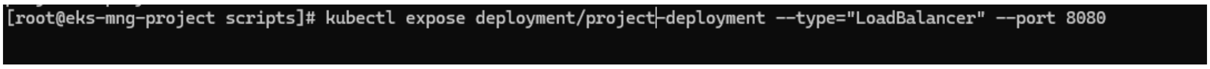


```
[root@eks-mng-project scripts]# kubectl apply -f test-deployment.yaml
```

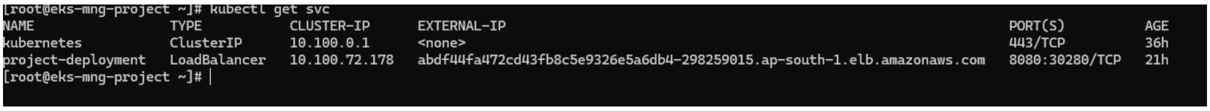


```
[root@eks-mng-project scripts]# kubectl get deployments
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
project-deployment   2/2     2            2           154m
[root@eks-mng-project scripts]#
```

- Expose a service for deployment.

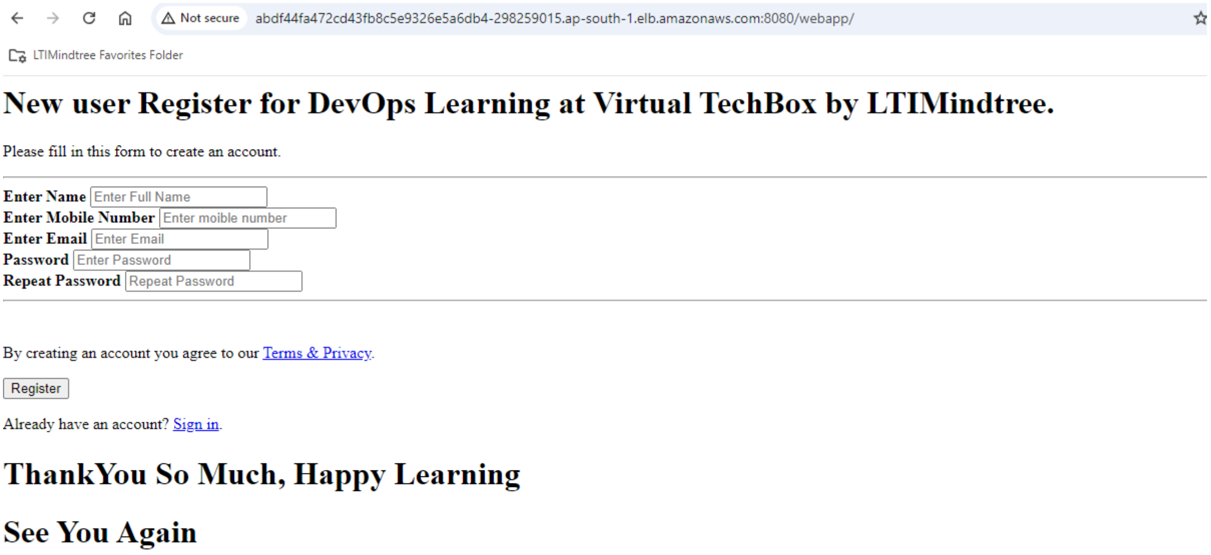


```
[root@eks-mng-project scripts]# kubectl expose deployment/project-deployment --type="LoadBalancer" --port 8080
```



```
[root@eks-mng-project ~]# kubectl get svc
NAME                 TYPE           CLUSTER-IP       EXTERNAL-IP                                                                PORT(S)          AGE
kubernetes           ClusterIP      10.100.0.1       <none>                                                                     443/TCP          36h
project-deployment   LoadBalancer   10.100.72.178    abdf44fa472cd43fb8c5e9326e5a6db4-298259015.ap-south-1.elb.amazonaws.com   8080:30280/TCP   21h
[root@eks-mng-project ~]#
```

- Copy the External IP and paste it in browser to see that your project is working.



Not secure abdf44fa472cd43fb8c5e9326e5a6db4-298259015.ap-south-1.elb.amazonaws.com:8080/webapp/

LTIMindtree Favorites Folder

New user Register for DevOps Learning at Virtual TechBox by LTIMindtree.

Please fill in this form to create an account.

Enter Name	Enter Full Name
Enter Mobile Number	Enter moible number
Enter Email	Enter Email
Password	Enter Password
Repeat Password	Repeat Password

By creating an account you agree to our Terms & Privacy.

Register

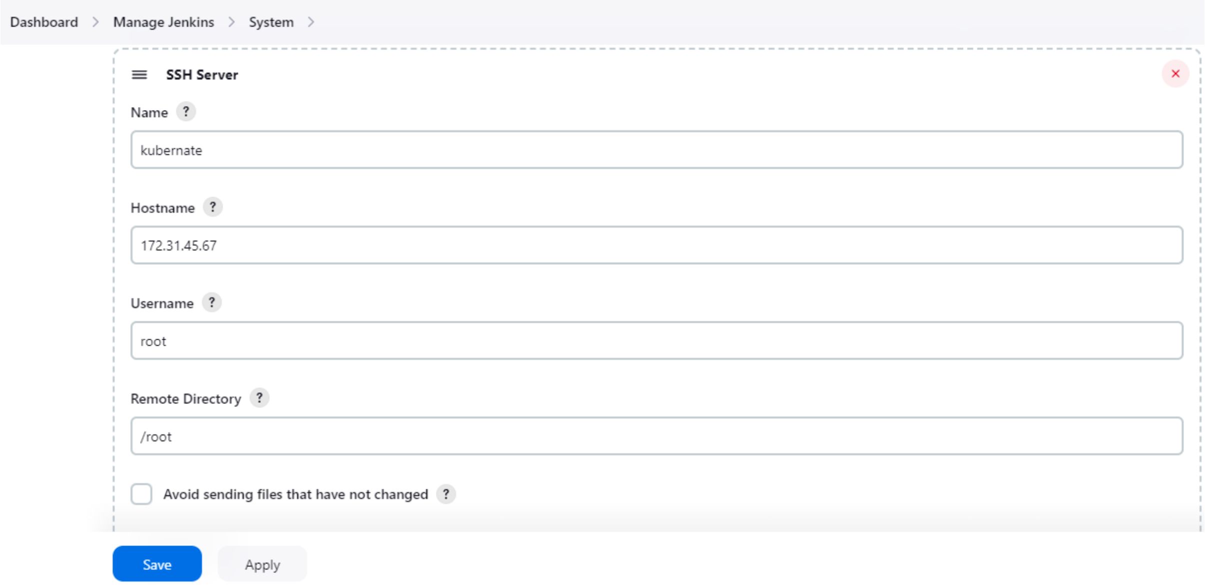
Already have an account? Sign in.

ThankYou So Much, Happy Learning

See You Again

#Automate Kubernetes cluster deployment using Jenkins.

- Go on system and add Kubernetes configuration.



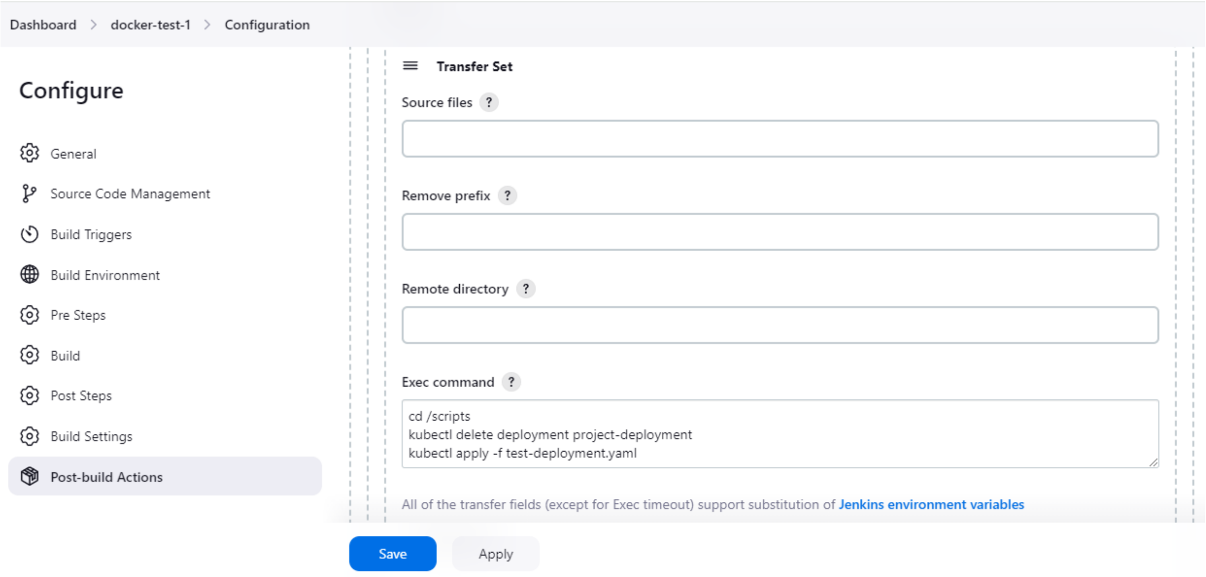
Dashboard > Manage Jenkins > System >

SSH Server

Name	kubernate
Hostname	172.31.45.67
Username	root
Remote Directory	/root

□ Avoid sending files that have not changed

- Configure the job and add Jenkins.



Transfer Set

Source files ?

Remove prefix ?

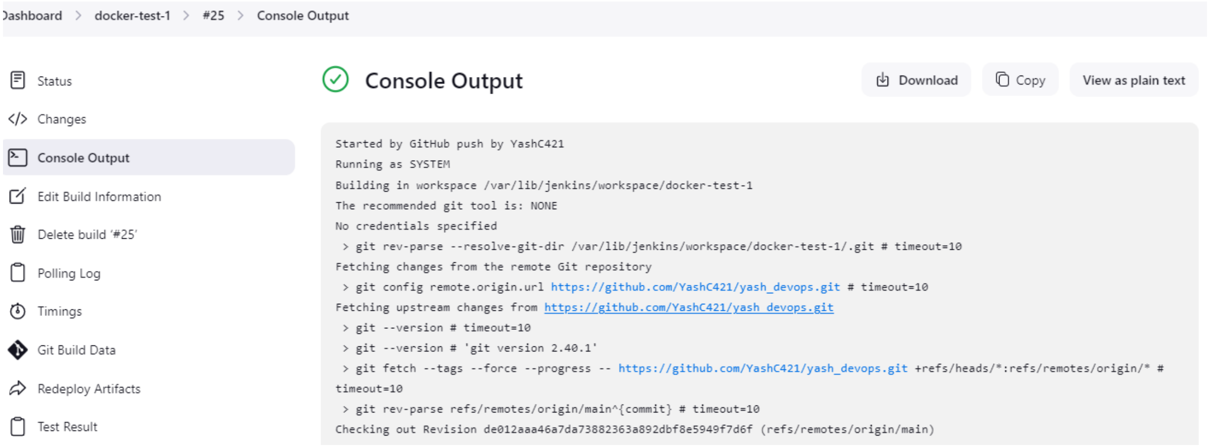
Remote directory ?

Exec command ?

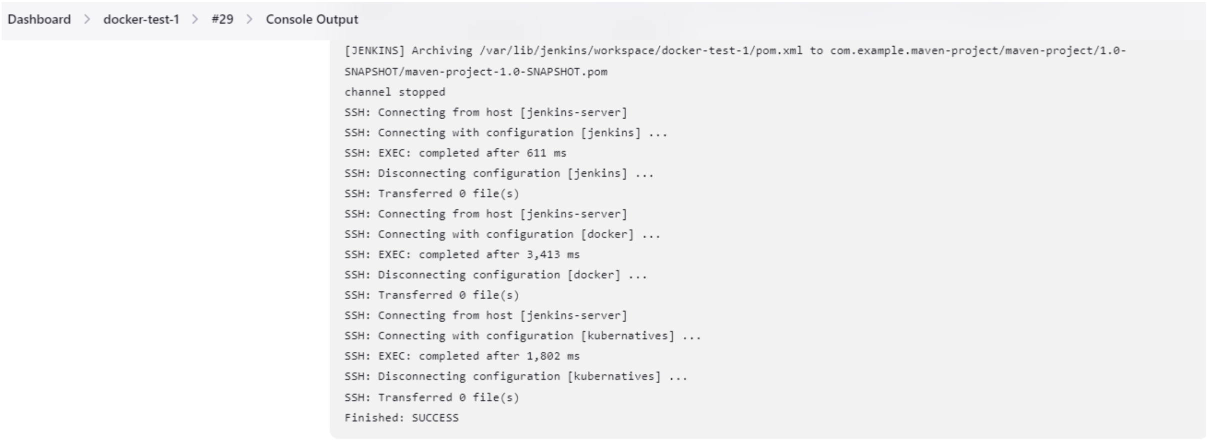
```
cd /scripts
kubectl delete deployment project-deployment
kubectl apply -f test-deployment.yaml
```

All of the transfer fields (except for Exec timeout) support substitution of **Jenkins environment variables**

- Build the job it will be automated Kubernetes.



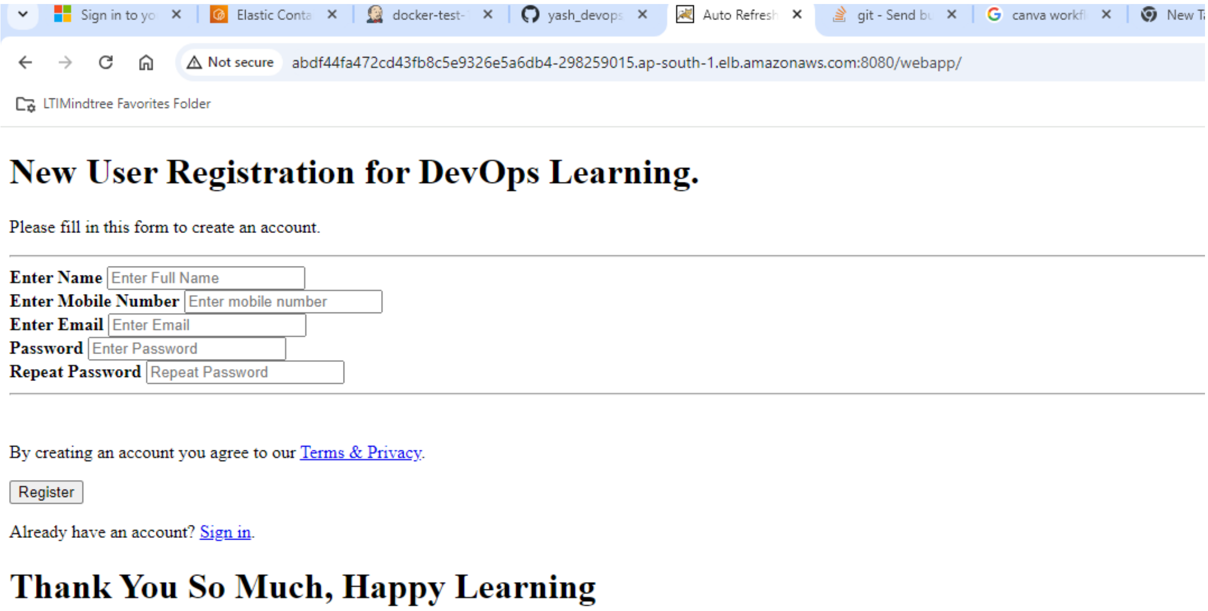
```
Started by GitHub push by YashC421
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/docker-test-1
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/docker-test-1/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/YashC421/yash_devops.git # timeout=10
Fetching upstream changes from https://github.com/YashC421/yash_devops.git
 > git --version # timeout=10
 > git --version # 'git version 2.40.1'
 > git fetch --tags --force --progress -- https://github.com/YashC421/yash_devops.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision de012aaa46a7da73882363a892dbf8e5949f7d6f (refs/remotes/origin/main)
```



```
[JENKINS] Archiving /var/lib/jenkins/workspace/docker-test-1/pom.xml to com.example.maven-project/maven-project/1.0-
SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
channel stopped
SSH: Connecting from host [jenkins-server]
SSH: Connecting with configuration [jenkins] ...
SSH: EXEC: completed after 611 ms
SSH: Disconnecting configuration [jenkins] ...
SSH: Transferred 0 file(s)
SSH: Connecting from host [jenkins-server]
SSH: Connecting with configuration [docker] ...
SSH: EXEC: completed after 3,413 ms
SSH: Disconnecting configuration [docker] ...
SSH: Transferred 0 file(s)
SSH: Connecting from host [jenkins-server]
SSH: Connecting with configuration [kubernatives] ...
SSH: EXEC: completed after 1,802 ms
SSH: Disconnecting configuration [kubernatives] ...
SSH: Transferred 0 file(s)
Finished: SUCCESS
```

#Now go on browser and do changes in git repository it automatically reflects on server we can see in browser.

#Now the Website is live.



abdf44fa472cd43fb8c5e9326e5a6db4-298259015.ap-south-1.elb.amazonaws.com:8080/webapp/

New User Registration for DevOps Learning.

Please fill in this form to create an account.

Enter Name Enter Full Name
Enter Mobile Number 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 So Much, Happy Learning

By: Yash Chouhan