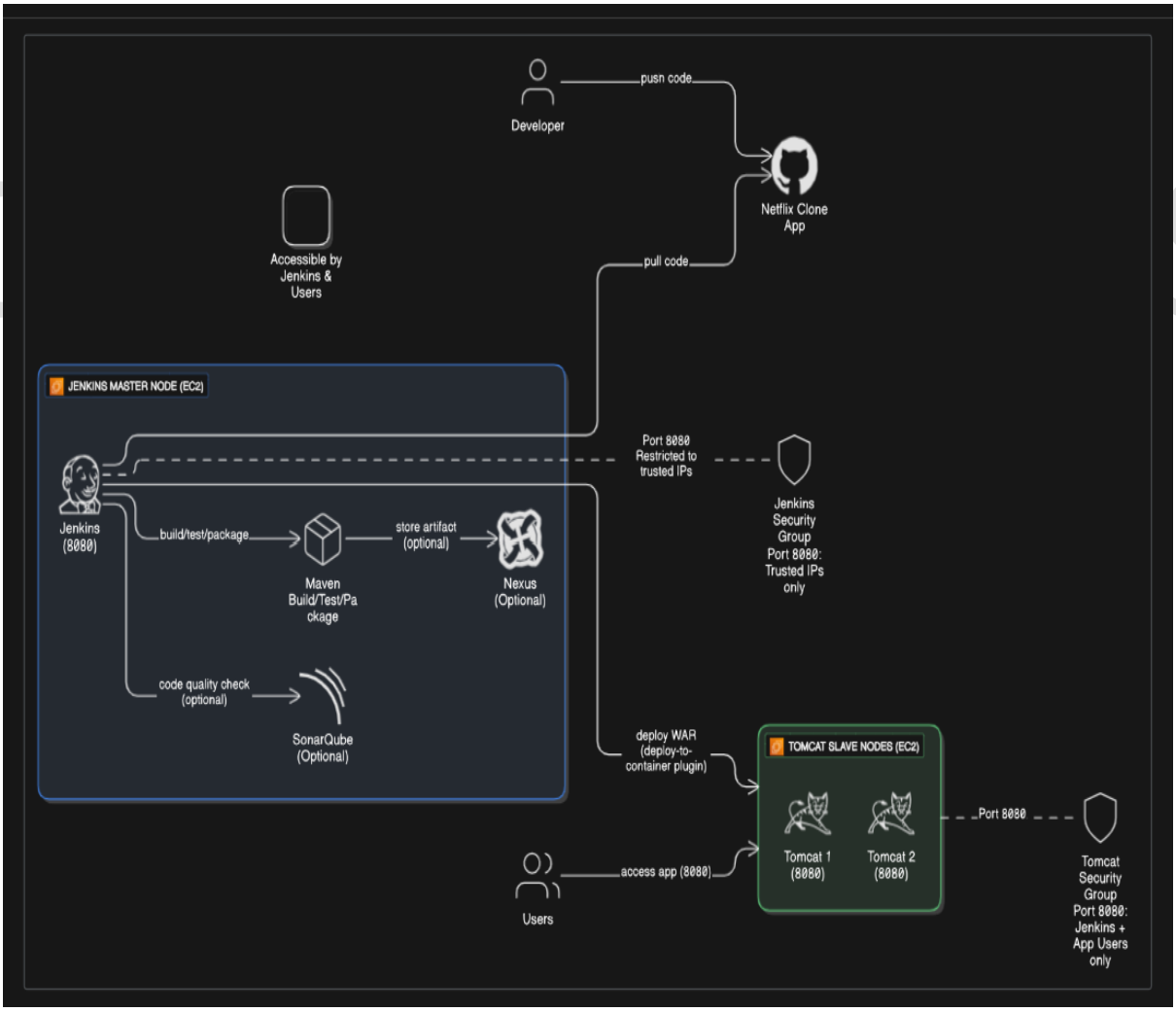
Netflix Jenkins CI/CD Pipeline Setup with Tomcat Server:

This task involves setting up a complete CI/CD pipeline using Jenkins to automate the deployment of a Netflix clone application. The solution is designed to follow industry-standard practices by utilizing Jenkins as the automation server. Jenkins is installed on a main node, while deployment targets are configured with Apache Tomcat to serve the application.

The pipeline fetches source code from a Git repository, compiles and tests the application using Maven, and packages it as a WAR file. The final stage automatically deploys the WAR file to Tomcat servers on the target nodes. This setup enables seamless, automated, and scalable deployment of applications in a DevOps environment, promoting faster release cycles, consistency, and reduced human error.



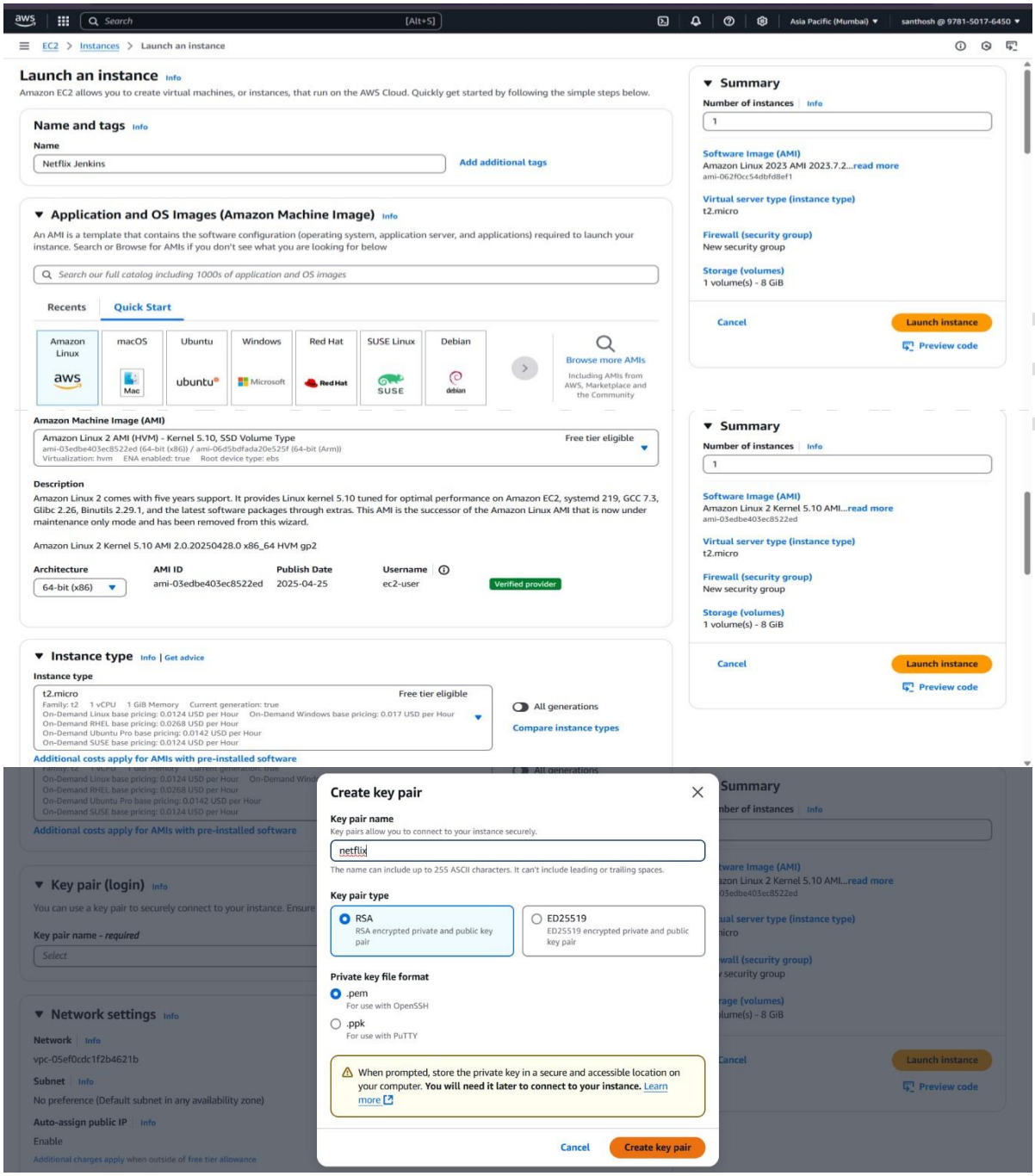
Step 1: Sign into the AWS Console.

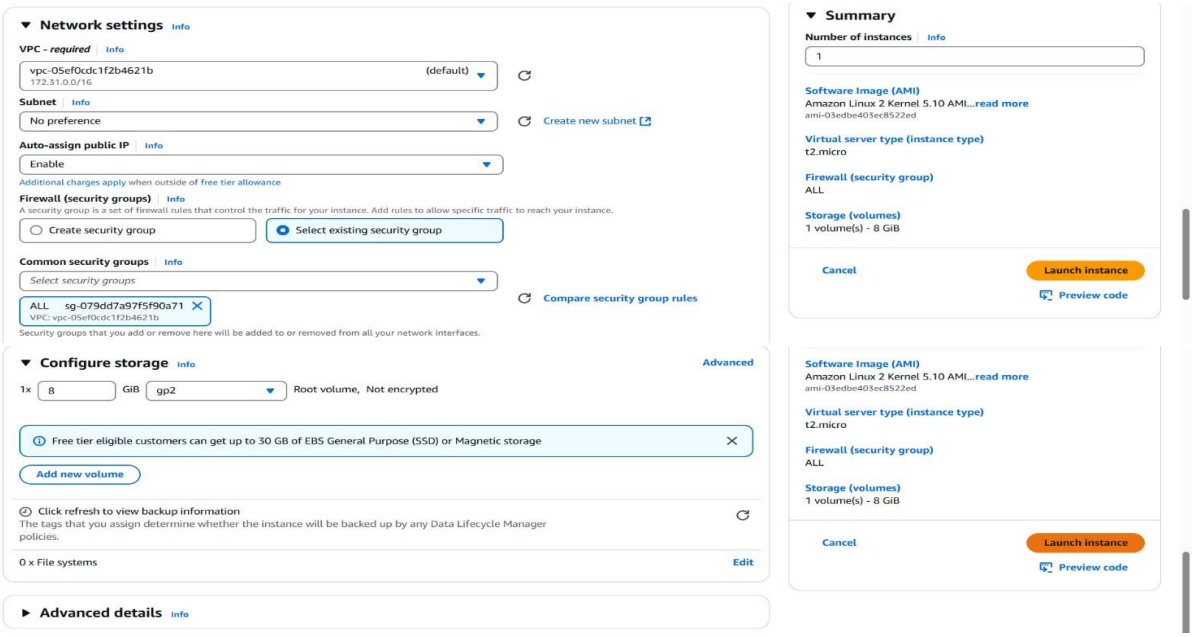
Step 2: Navigate to EC2.

Step 3: Create 1 instance – for Installation of Jenkins.

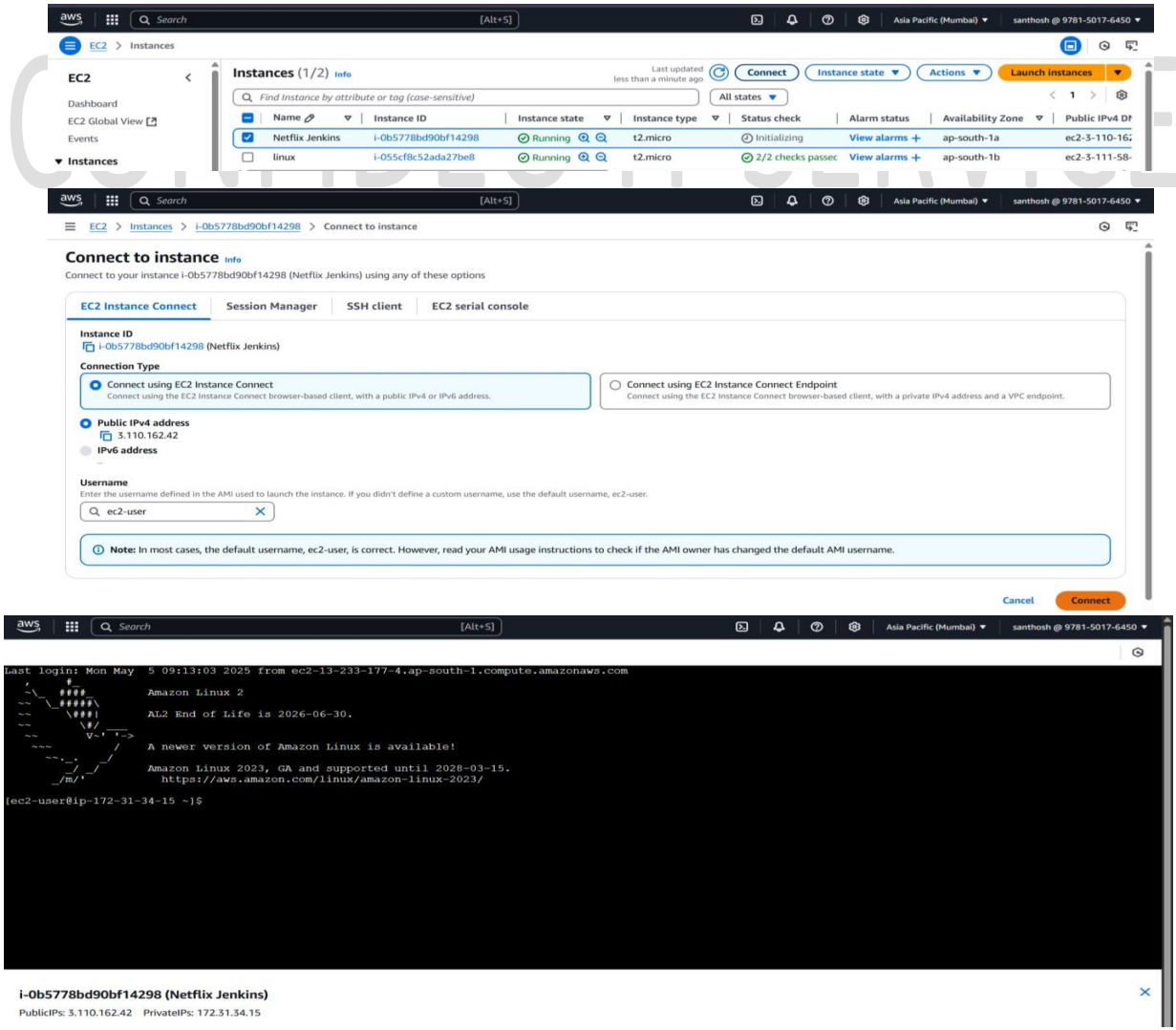
Step 4: Name → AMI- Amazon Linux (choose Amazon Linux 2 AMI) → Instance Type :t2.micro → Create a key pair (.pem /.ppk) [if .ppk choosen then later convert it to .pem file using putty Gen] → Security Group – Allow all security/ Allow All TCP [Allow these ports- 22, 8080, 80, 50000) → Click on Launch Instance.

[**NOTE: Allowing All Security & All TCP -: It's strongly discouraged in production environments due to serious security risks. It is always a best practise to allow the ports which is required.**]



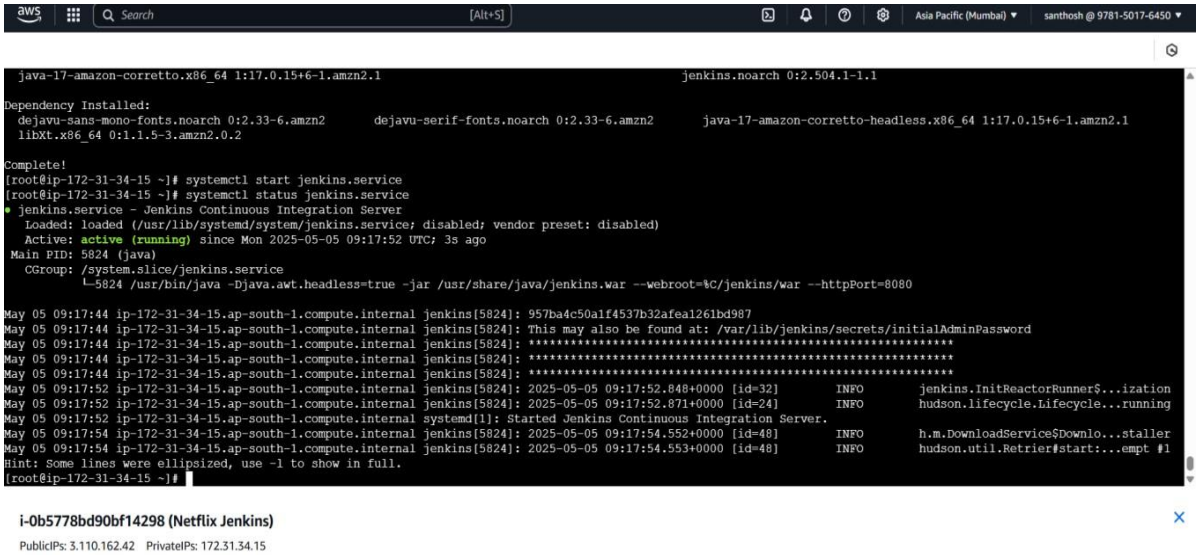


Step 5: Select the jenkins instance → click on Connect.



Step 6: Commands:

- sudo -i
- use the below link to install Jenkins
  https://github.com/confideoit/Confideo-All_Setup/blob/main/jenkins.sh
- Copy the entire script and paste in the terminal.
- Don't forget to start the service



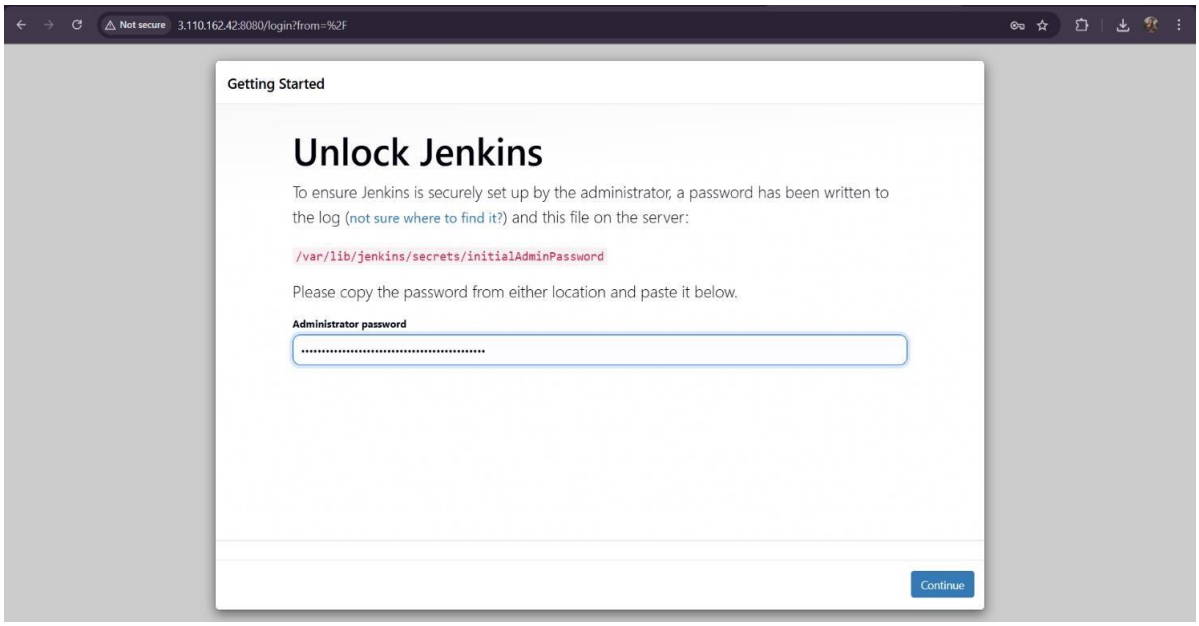
CONFIDEO IT SERVICES

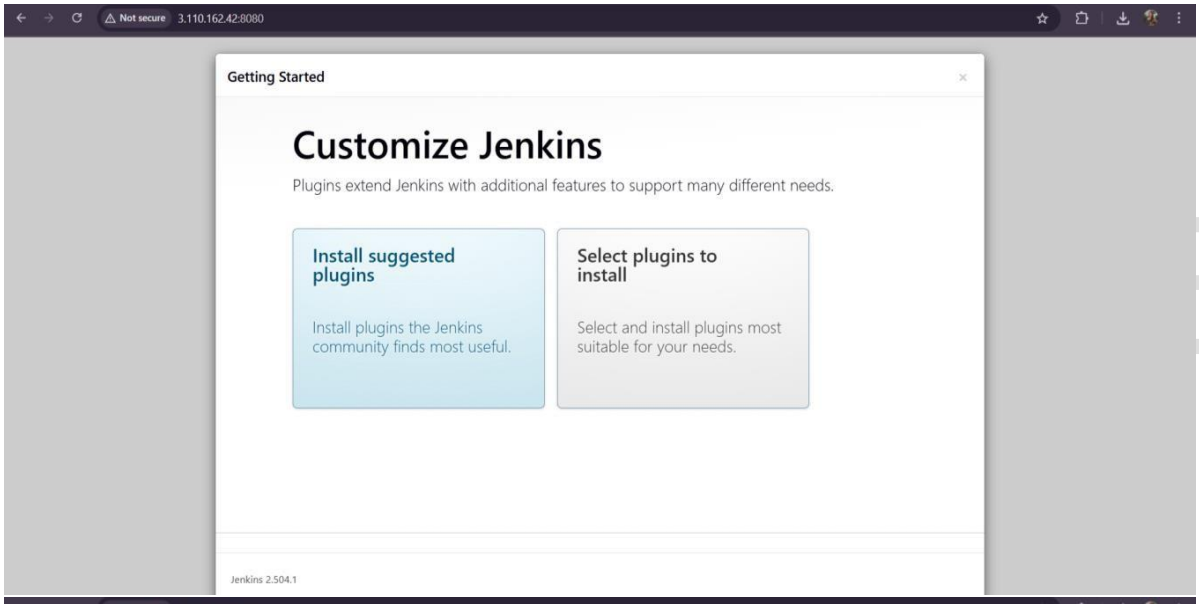
Step 7: After, the service get started copy the IP of your Jenkins instance and paste in the browser with port-8080 (e.g: 3.110.162.42:8080).

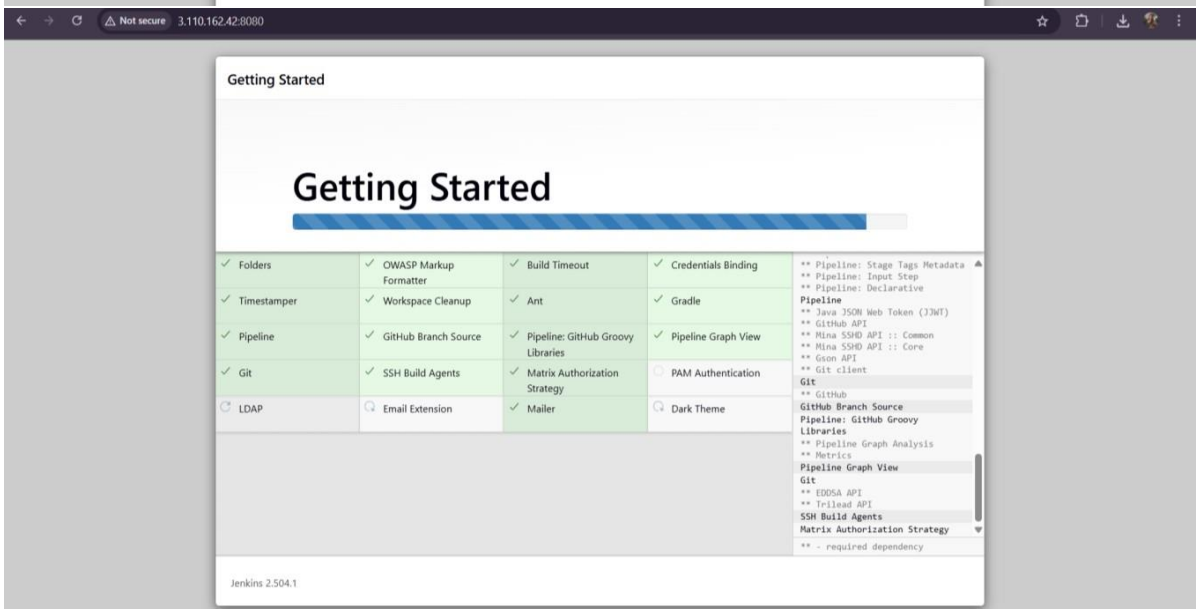
Step 8: Paste the IP – Jenkins Sign-in page will appear.

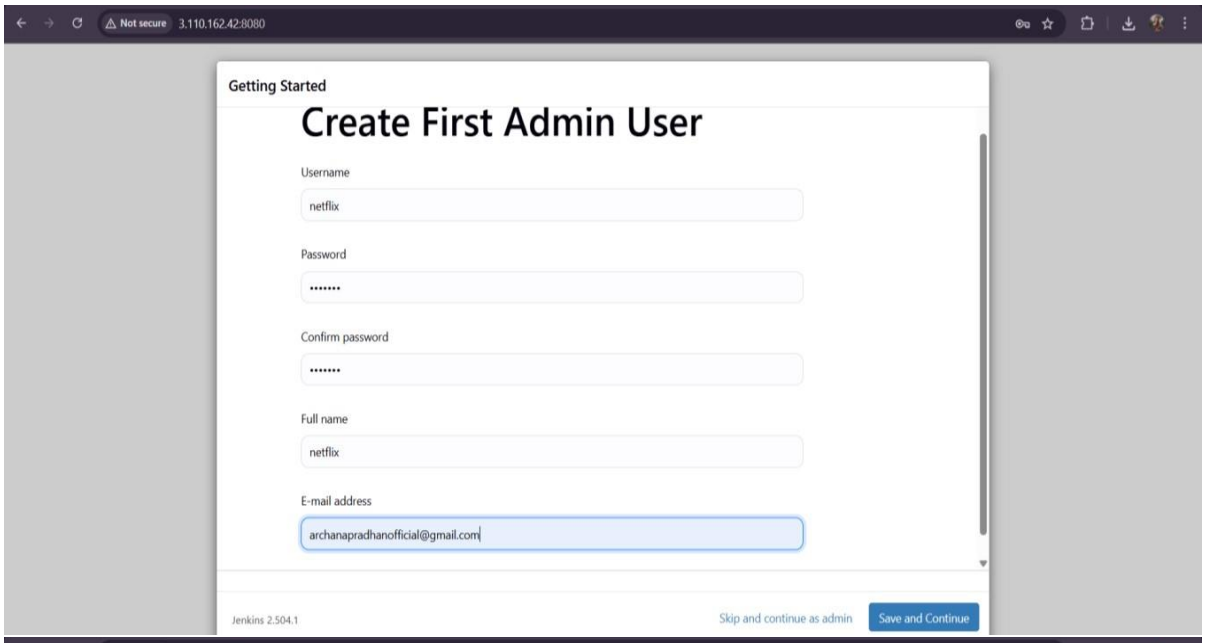
Step 9: Copy the /var/lib/jenkins/secrets/initialAdminPassword which is present in the jenkins sign-in page

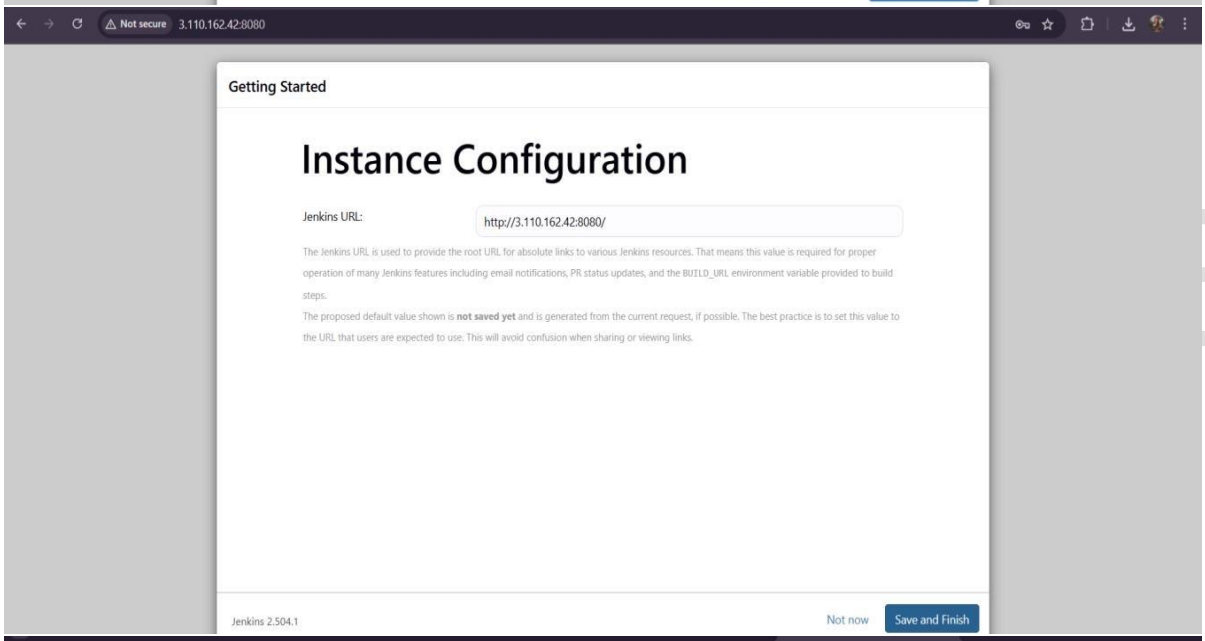
- Command:
  - cat /var/lib/jenkins/secrets/initialAdminPassword
- You will get a code copy and paste it in the Administrator password slot.

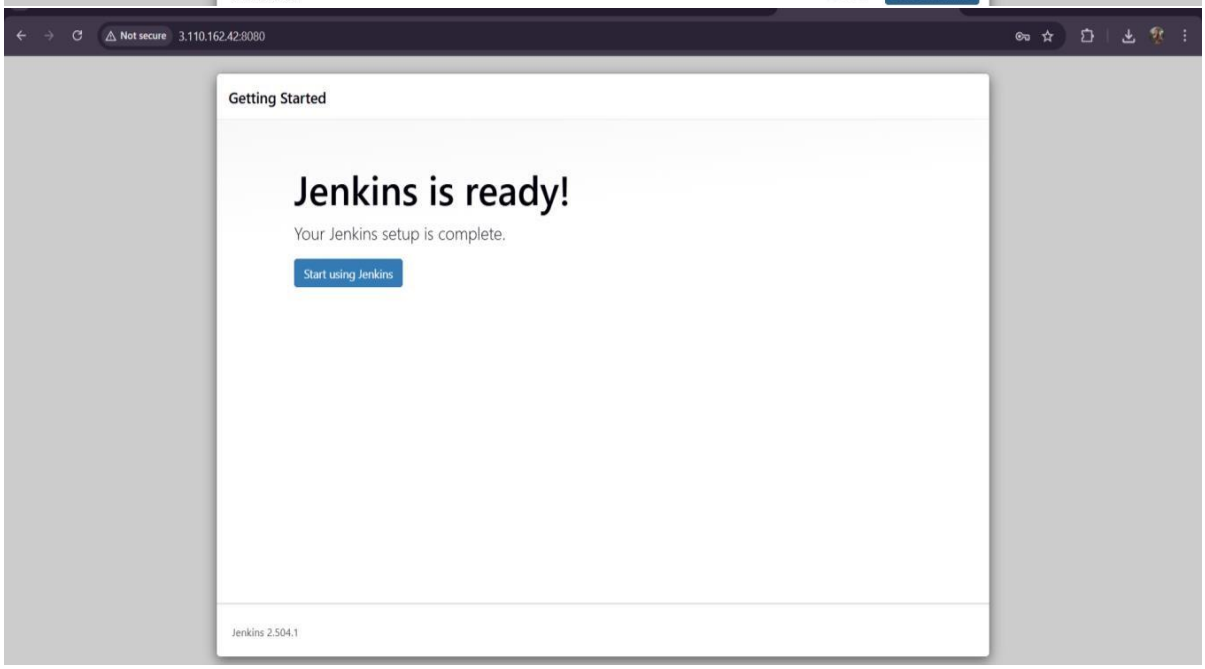

Not secure 3.110.162.42:8080/login?from=%2F
Getting Started
Unlock Jenkins
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:
/var/lib/jenkins/secrets/initialAdminPassword
Please copy the password from either location and paste it below.
Administrator password
Continue

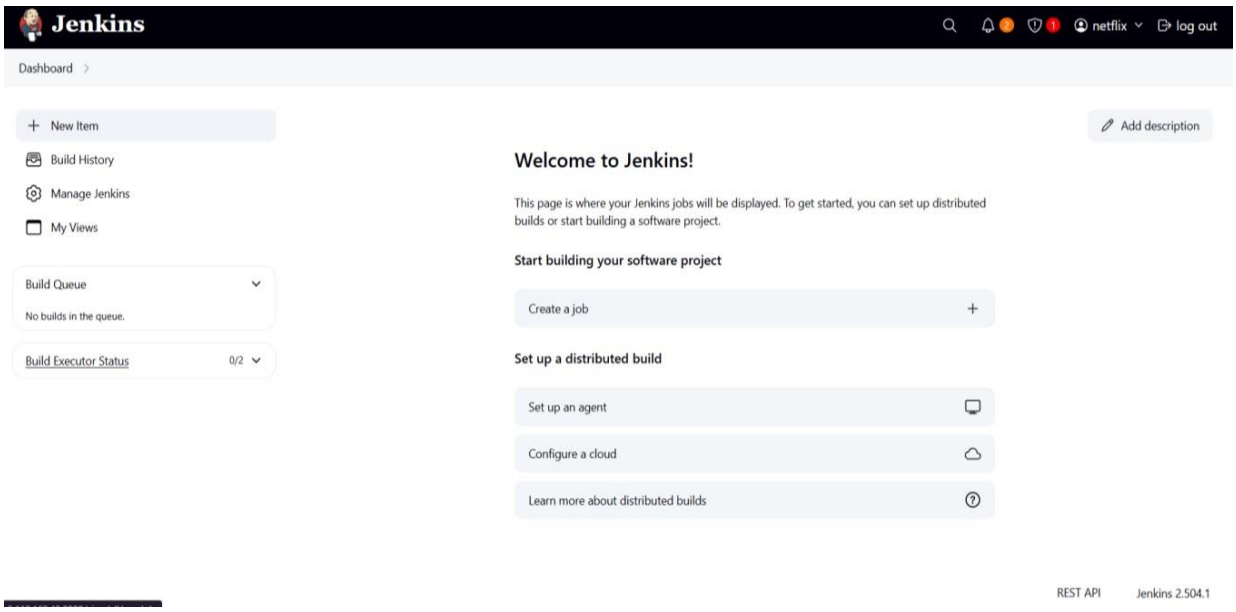

Not secure 3.110.162.42:8080
Getting Started
Customize Jenkins
Plugins extend Jenkins with additional features to support many different needs.
Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.
Jenkins 2.504.1


Not secure 3.110.162.42:8080
Getting Started
Getting Started
Folders
OWASP Markup Formatter
Build Timeout
Credentials Binding
Timestamper
Workspace Cleanup
Ant
Gradle
Pipeline
GitHub Branch Source
Pipeline: GitHub Groovy Libraries
Pipeline Graph View
Git
SSH Build Agents
Matrix Authorization Strategy
PAM Authentication
LDAP
Email Extension
Mailer
Dark Theme
Jenkins 2.504.1


Not secure 3.110.162.42:8080
Getting Started
Create First Admin User
Username
netflix
Password
Confirm password
Full name
netflix
E-mail address
archanapradhanofficial@gmail.com
Jenkins 2.504.1
Skip and continue as admin
Save and Continue

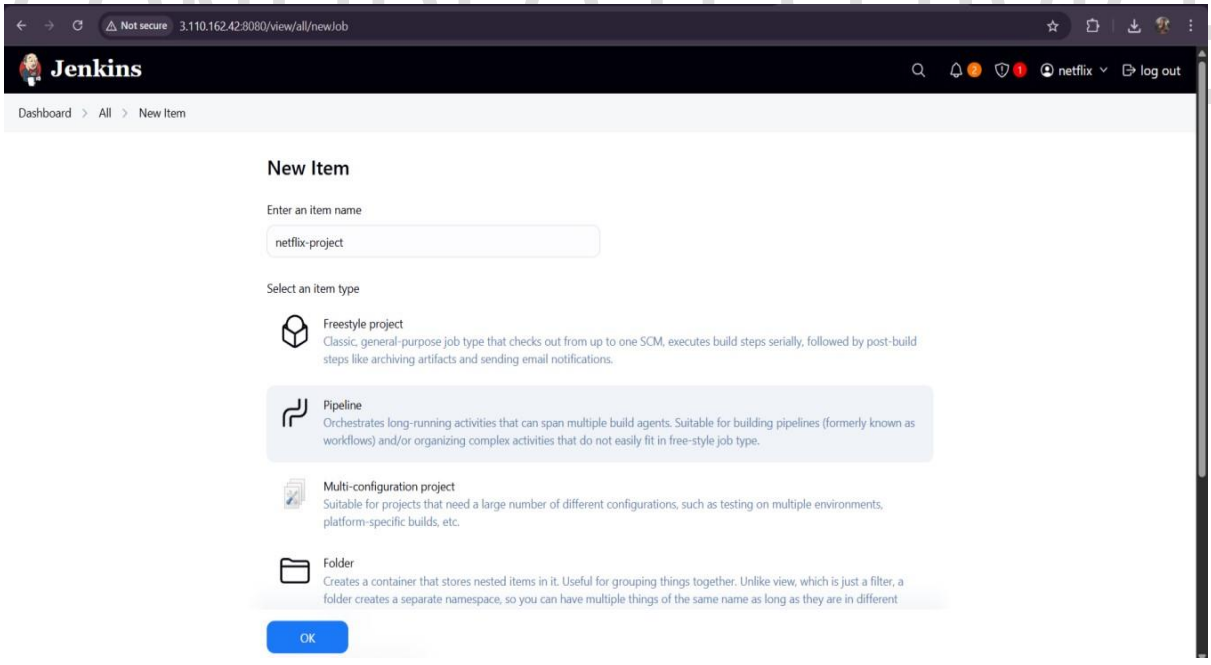

Not secure 3.110.162.42:8080
Getting Started
Instance Configuration
Jenkins URL:
http://3.110.162.42:8080/
The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.
The proposed default value shown is not saved yet and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.
Jenkins 2.504.1
Not now
Save and Finish


Not secure 3.110.162.42:8080
Getting Started
Jenkins is ready!
Your Jenkins setup is complete.
Start using Jenkins
Jenkins 2.504.1



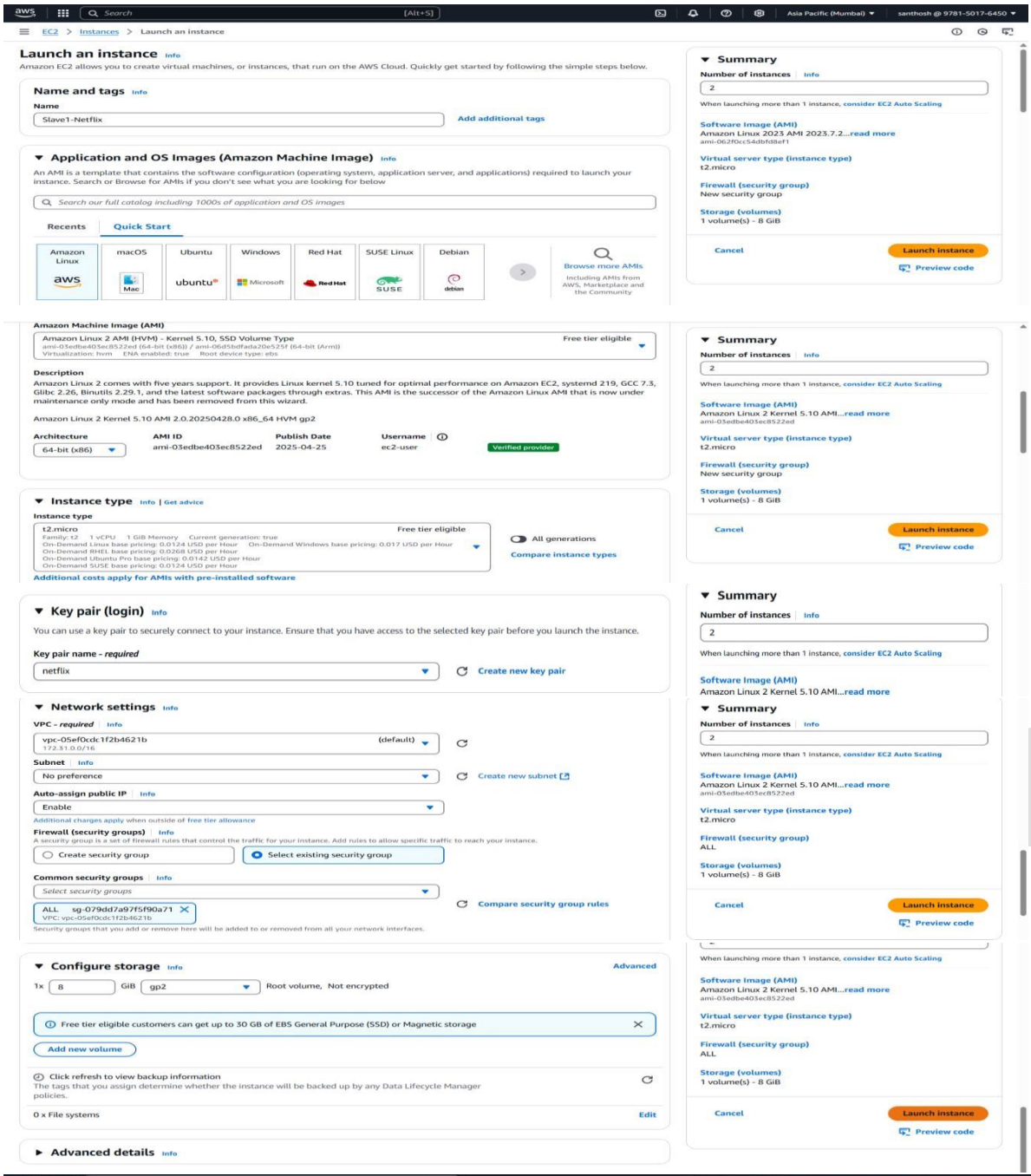
Step 10: Click on New Item → Enter the name of the item → choose Pipeline → Click on OK.

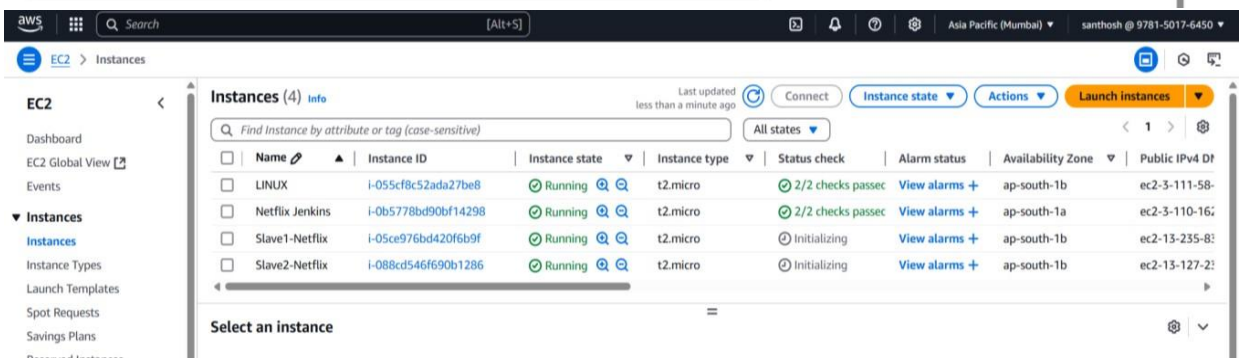
CONFIDEO IT SERVICES



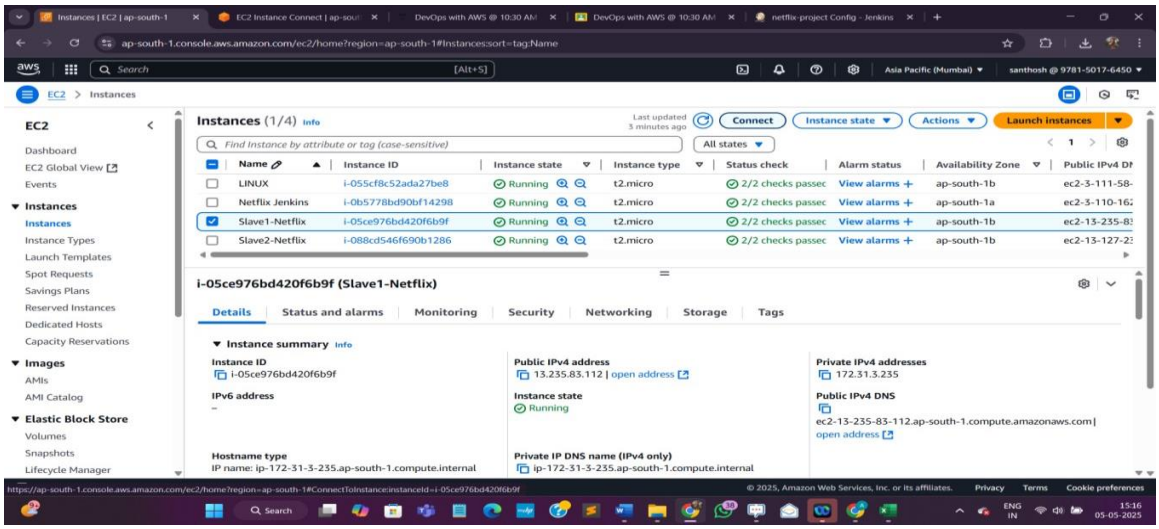
Step 11: Now Create 2 instances – for Installation of Apache Tomcat.

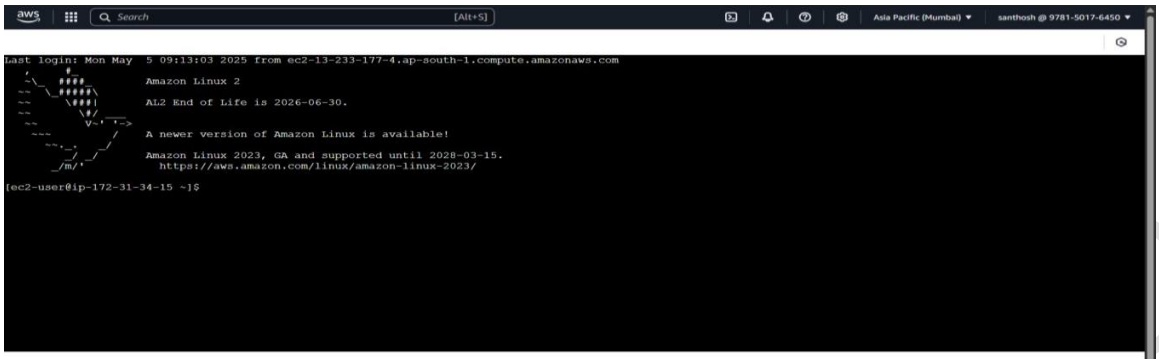
Step 12: Give 2 in the Number of Instances → Name → AMI- Amazon Linux (choose Amazon Linux 2 AMI) → Instance Type – t2.micro → use the existing key pair → Use the Existing Security Group → Click on Launch Instance.


Launch an instance
Name and tags
Slave1-Netflix
Application and OS Images (Amazon Machine Image)
Instance type
t2.micro
Key pair (login)
netflix
Network settings
Configure storage
Advanced details
Summary
Number of instances
2
Launch instance
Preview code


EC2 > Instances
Instances (4)
Launch instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
LINUX
Netflix Jenkins
Slave1-Netflix
Slave2-Netflix
Select an instance

Step 13: Select the 1st Apache Instance → click on connect.





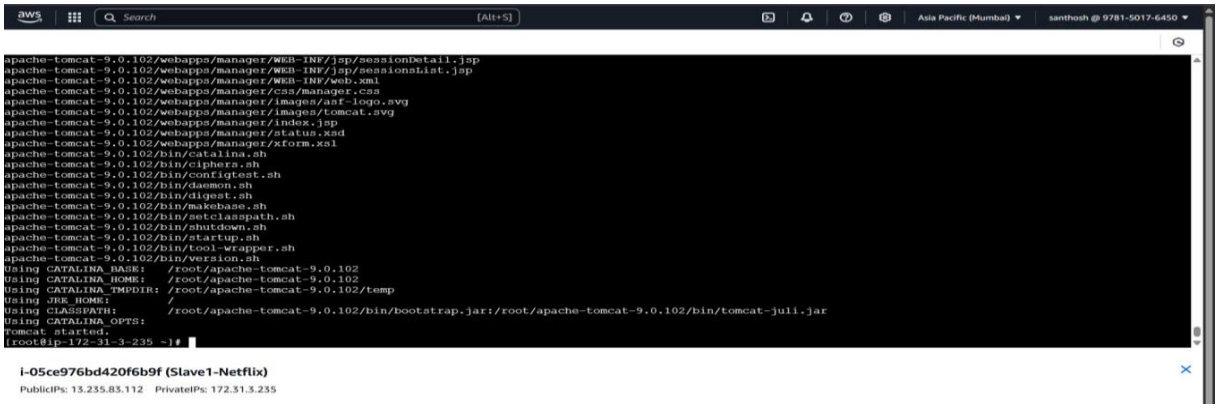
Step 14: Commands:

- sudo -i
- yum install git java-1.8.0-openjdk maven -y
- sudo yum install java-17-amazon-corretto -y
- vim tomcat.sh

Inside vim tomcat.sh paste the script:

https://github.com/confideoit/Confideo-All_Setup/blob/main/tomcat.sh

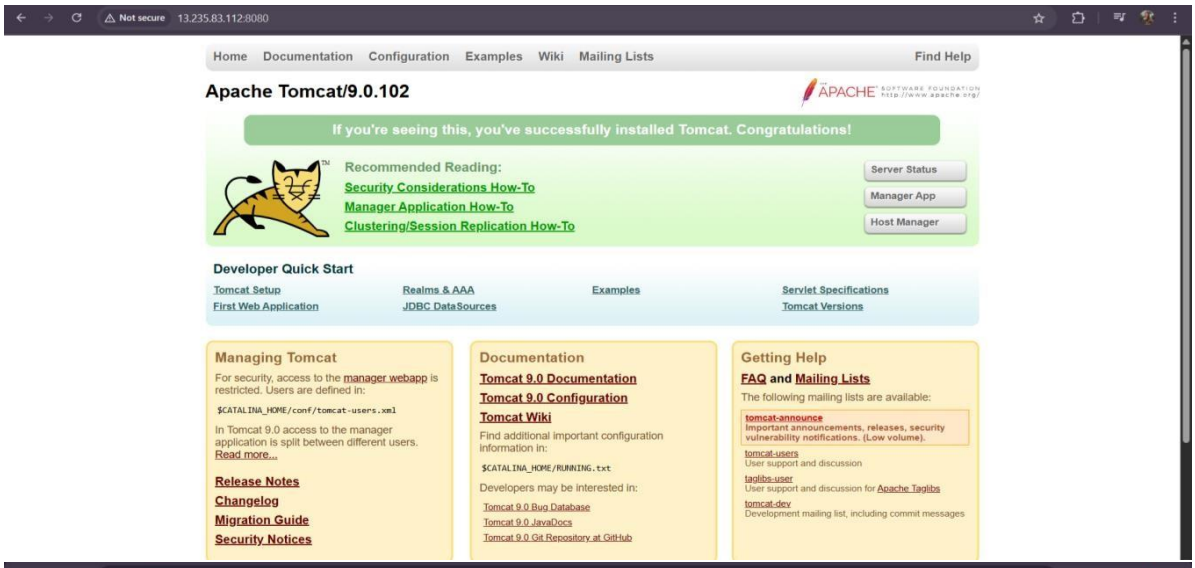
- sh tomcat.sh

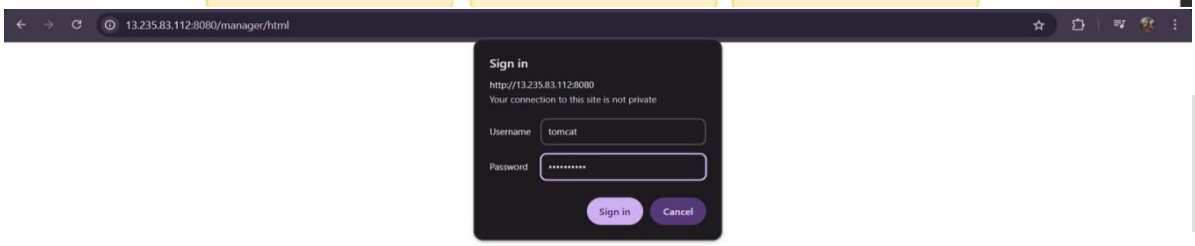


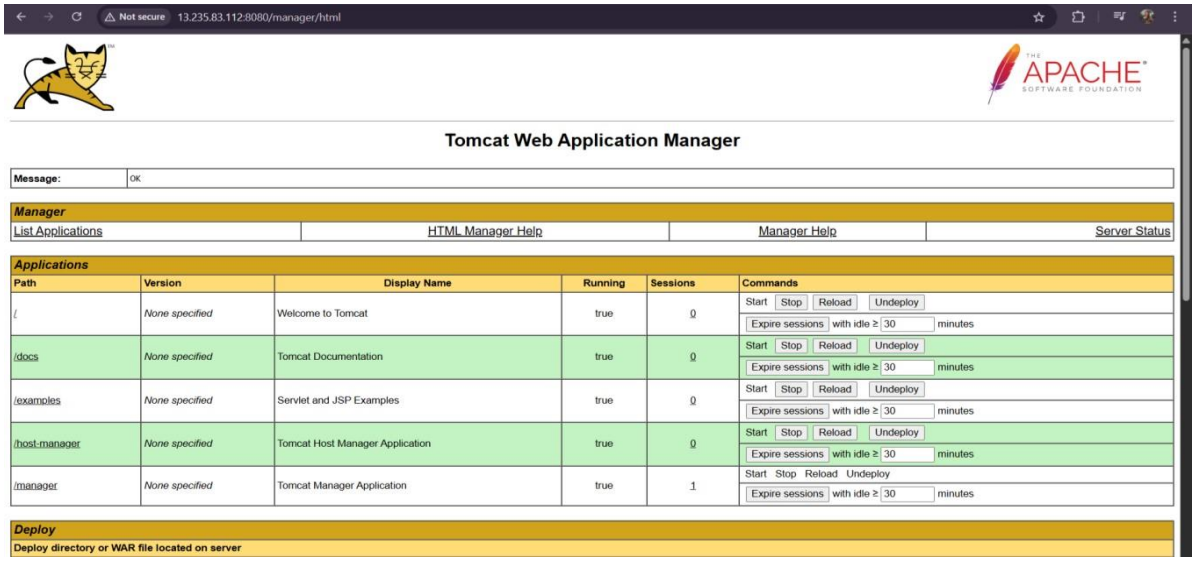
Step 15: Now, check that the tomcat is installed properly or not.

Step 16: Copy the instance public ip and paste it.

Step 17: Once the page appears → click on manager app → user=tomcat & pw=confideo123.





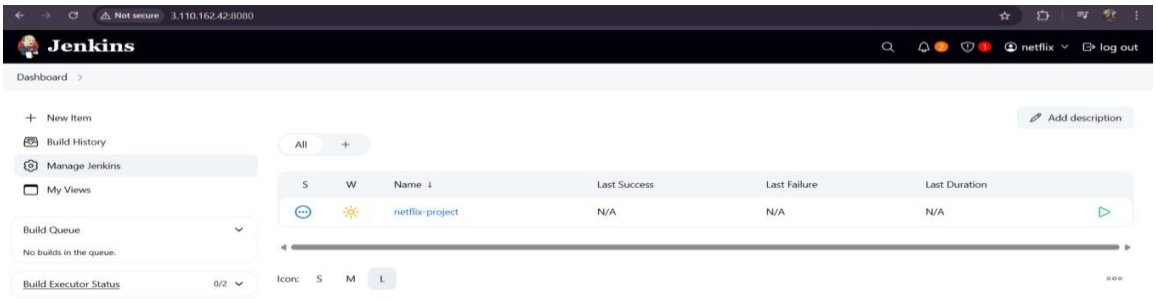


Step 18: Repeat from step-14 to step-17 for another Instance.

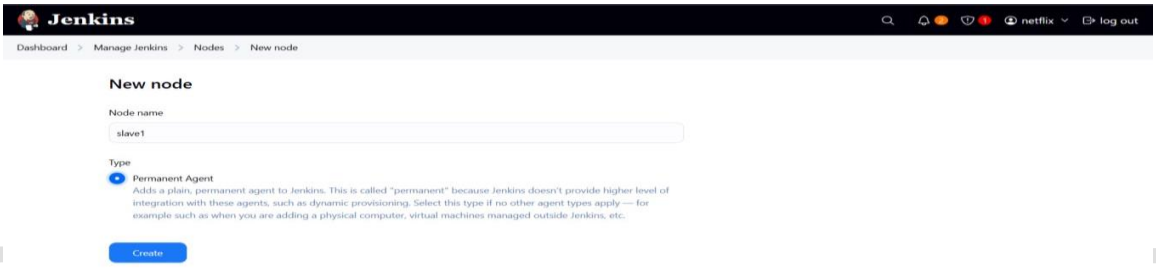
Step 19: Now login to Jenkins Console.

Step 20: From Dashboard → Click on manage Jenkins →nodes →click on new node→ give node name → select permanent agent → Save

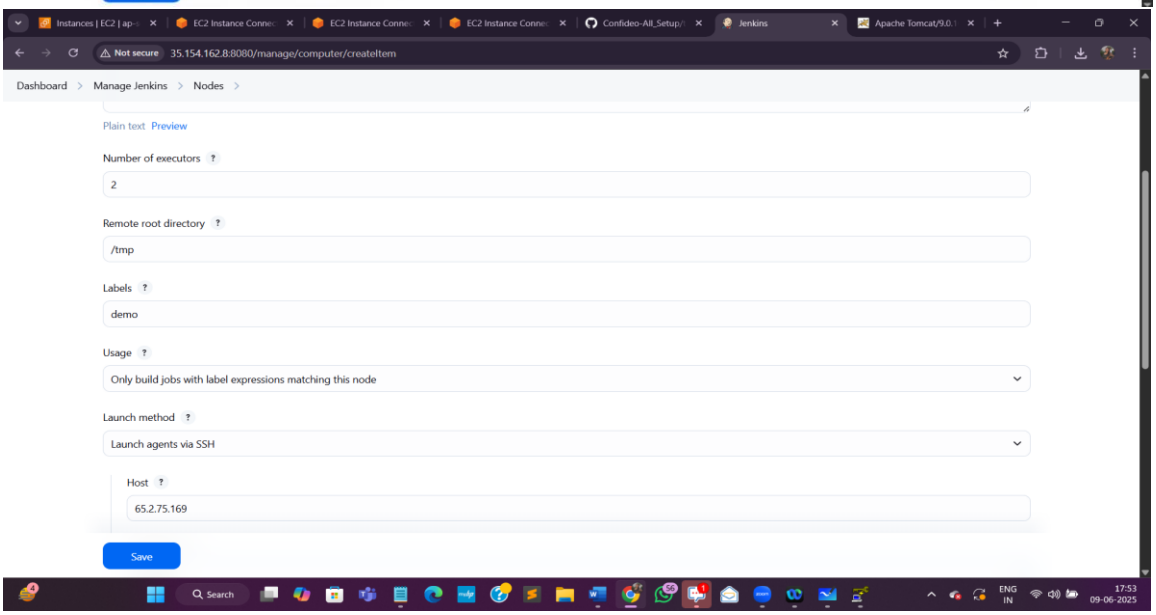
- In the Host add the private-ip.

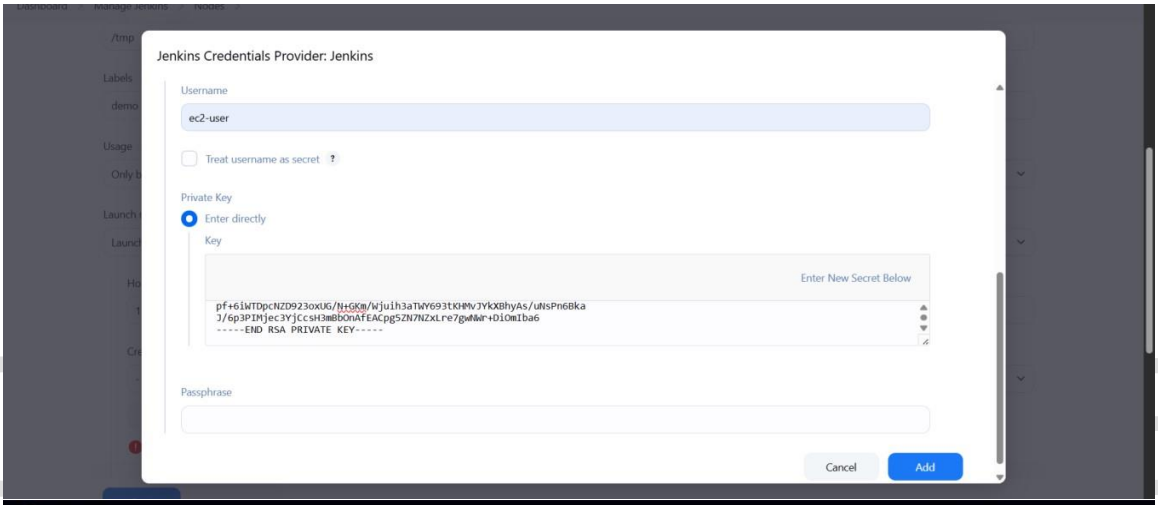

Not secure 3.110.162.42:8080
Jenkins
netflix
log out
Dashboard
New Item
Build History
Manage Jenkins
My Views
Add description
All
S
W
Name
Last Success
Last Failure
Last Duration
netflix-project
N/A
N/A
N/A
Build Queue
No builds in the queue.
Build Executor Status
0/2
Icon: S M L

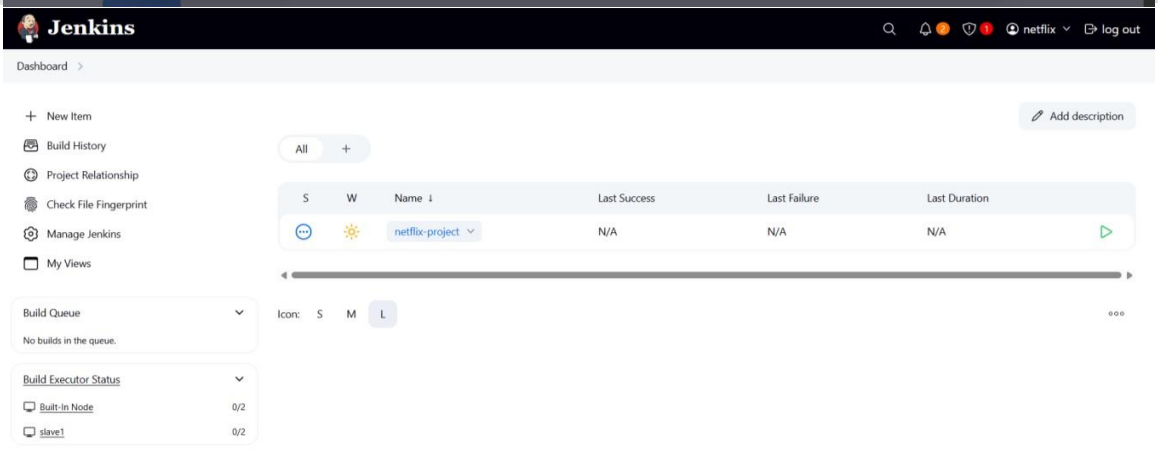

Jenkins
netflix
log out
Dashboard
Manage Jenkins
Nodes
Nodes
New Node
Configure Monitors
Clouds
Name
Architecture
Clock Difference
Free Disk Space
Free Swap Space
Free Temp Space
Response Time
Built-In Node
Linux (amd64)
In sync
5.25 GiB
0 B
5.25 GiB
0ms
Data obtained
37 min
Build Queue
No builds in the queue.
Build Executor Status
0/2
Icon: S M L
Legend


Jenkins
netflix
log out
Dashboard
Manage Jenkins
Nodes
New node
New node
Node name
slave1
Type
Permanent Agent
Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.
Create


Not secure 3.110.162.42:8080/manage/computer/createItem
Dashboard
Manage Jenkins
Nodes
Name
slave1
Description
Plain text Preview
Number of executors
2
Remote root directory
/tmp
Labels
demo
Save

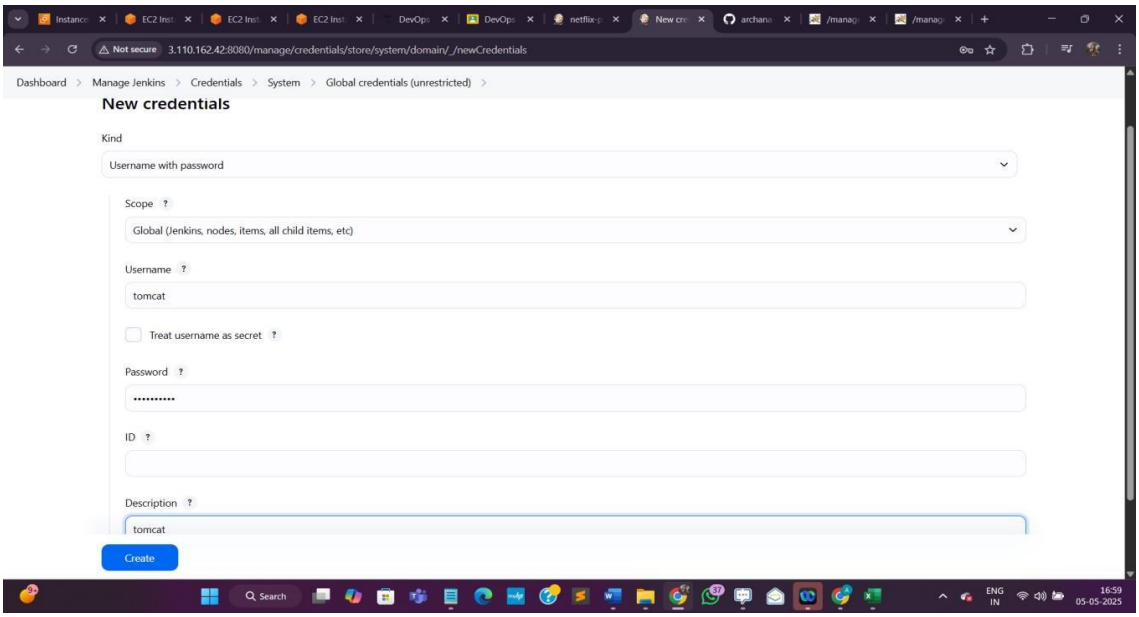

Not secure 35.154.162.8:8080/manage/computer/createItem
Dashboard
Manage Jenkins
Nodes
Plain text Preview
Number of executors
2
Remote root directory
/tmp
Labels
demo
Usage
Only build jobs with label expressions matching this node
Launch method
Launch agents via SSH
Host
65.2.75.169
Save





Step 21: Now, Add the Tomcat Credentials

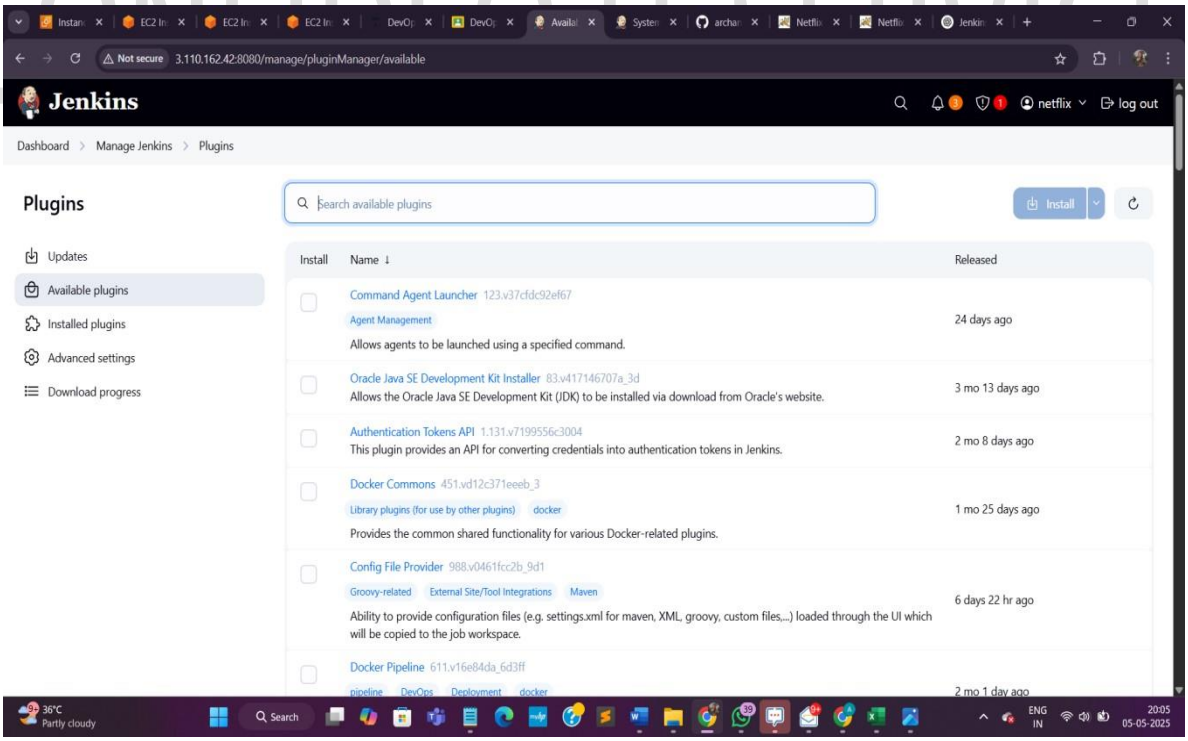
Step 22: Click on manage Jenkins → Credentials → Global → Add credentials → give the tomcat username & password → create .



Step 23: Now add the pulg-in

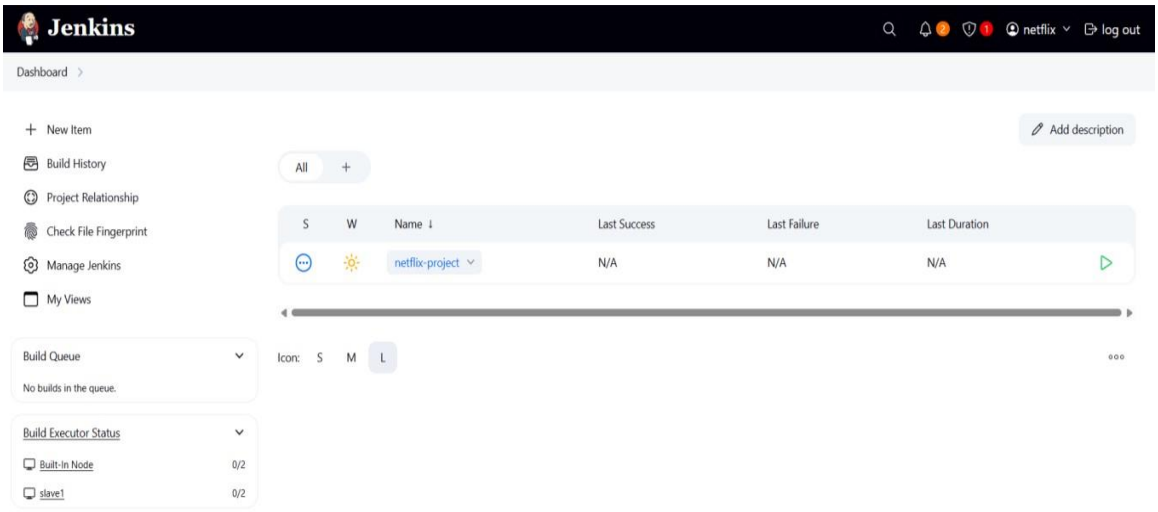
Step 24: Click on Manage Jenkins → Pulgins → select Available Pulgins → select deploy to container & pipeline-stage view.

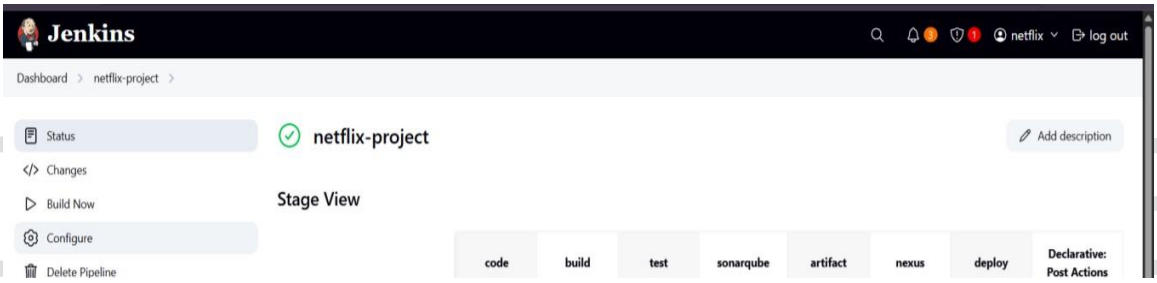
CONFIDEO IT SERVICES

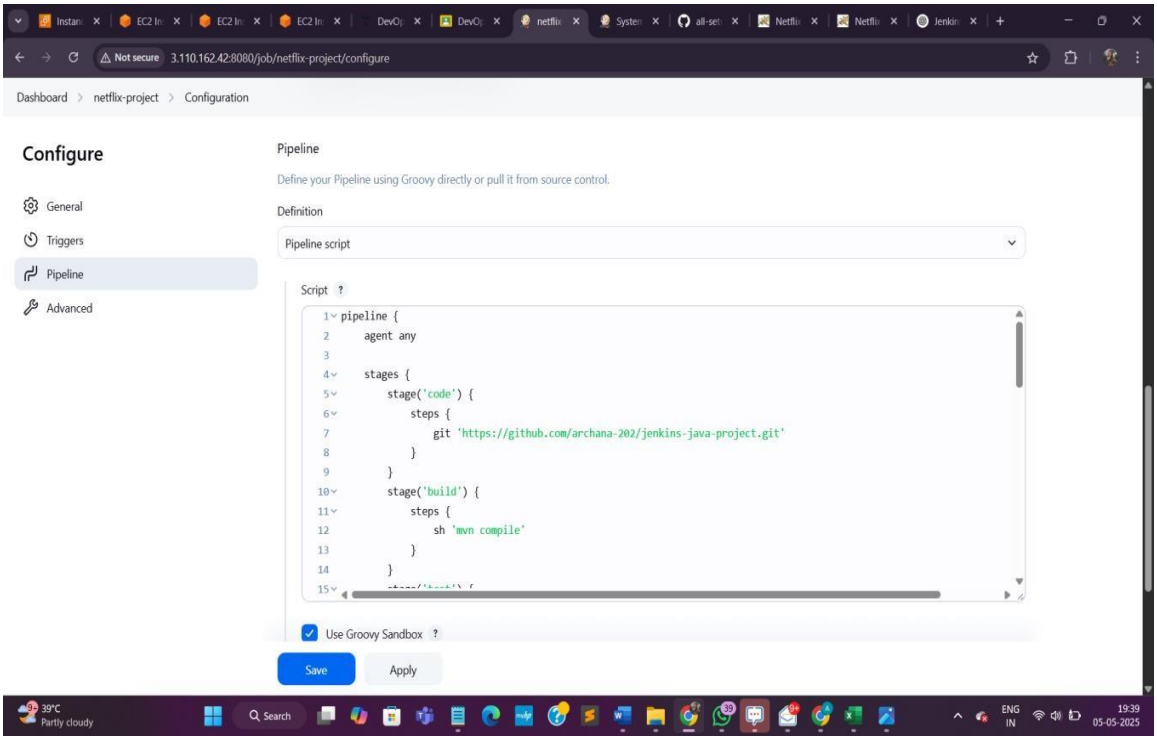


Step 25: Now, Its time to Add the pipeline.

Step 26: Click on the project →Click on Configure →Scroll down to Pipeline → and paste the script.







SCRIPT:

```
pipeline {
agent any

stages {
stage('code') {
steps {
git branch: 'main', url: 'https://github.com/confideoit/Confideo-Netflix-Jenkins-ec2.git'
}
}
stage('build') {
steps {
sh 'mvn compile'
}
}
stage('test') {
steps {
sh 'mvn test'
}
}
stage('sonarqube') {
steps {
echo "my code is tested"
}
}
stage('artifact') {
steps {
sh 'mvn clean package'
}
}
stage('nexus') {
steps {
echo "my artifact is stored on nexus"
}
}
stage('deploy') {
steps {
deploy adapters: [
        tomcat9(credentialsId: '540b11a8-4021-4b9f-a99b-23aceb2217e4', path: '', url:
        'http://3.110.117.33:8080/')
],
contextPath: 'netflix',
war: 'target/NETFLIX-1.2.2.war'
}
}
}
post {
```

CONFIDEO IT SERVICES

always {
echo "my code deployed"
}
}
}

Changes requird at your end in the script is:

Replace the github url.

Step 27: To generate GITHUB URL → Click on pipeline syntax → select GIT → add the git hub url → click on generate pipeline script → copy and paste that in the script

stage('code') {

steps {

git 'https://github.com/archana-202/jenkins-java-project.git'

}

}

change the credential id to your generated credential id and the URL of the slave instance

CONFIDEO IT SERVICES

stage('deploy') {

steps {

deploy

adapters: [

tomcat9(credentialsId: 'ed4edddd-04a5-4027-9a0e-f5f59b6bcd86', path: '', url: 'http://13.235.83.112:8080/')

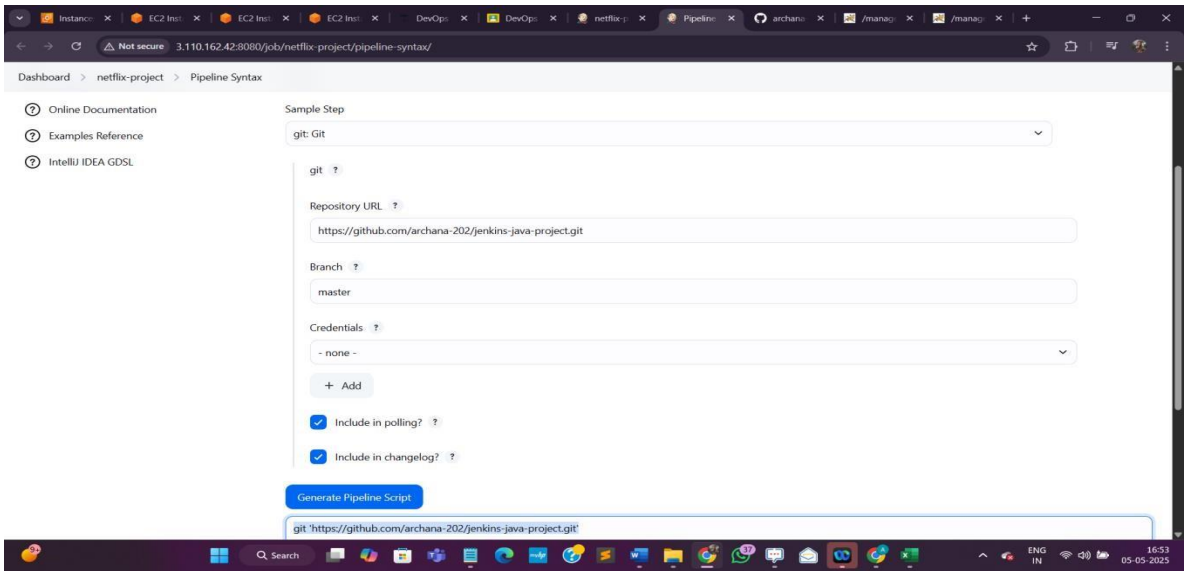
],

contextPath: 'netflix',

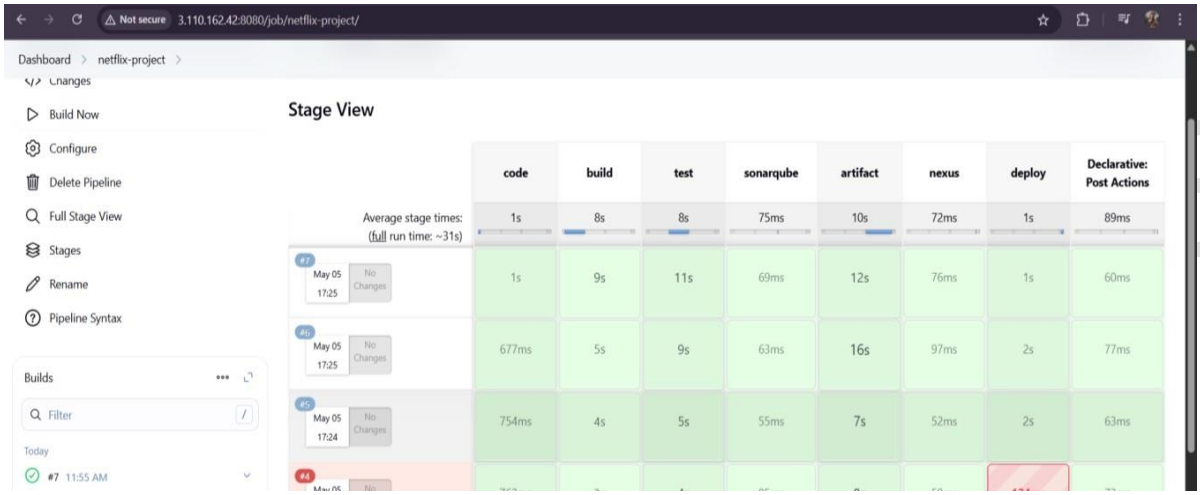
war: 'target/NETFLIX-

1.2.2.war'

Step 29: From github also frok this repo.
https://github.com/confideoit/Confideo-Netflix-Jenkins-ec2.git



Step 28: In the End, Click on Bulid Now.



----------X----------