

Module 4: Continuous Integration Using Jenkins

Abhijit Lalasaheb Zende

1. L1 – Create Jenkins Master-Slave Configurations and Create a workspace in Jenkins Slave Node using Jenkins Free-style project

Ans.

(*** Note: Screen shots attached to end of each question ***)

Step 1. Setup Jenkins Master node on a EC2 instance:

1. Launch an EC2 instance:

AMI=22.04, t2.medium with Security group having inbound rule for SSH i.e. port = 22, Jenkins web app port(Java applications port) i.e. port = 8080. All traffic should be allowed to these 2 ports(22,8080)

2. Install Jenkins:

a. SSH into EC2 instance

```
'ssh -i "your-key.pem" ubuntu@ec2-instance-public-ip-address'
```

b. Update the system

```
'sudo apt update && sudo apt upgrade -y'
```

c. Install Java (Jenkins requires Java 11 or higher):

```
'sudo apt install fontconfig openjdk-17-jdk'
```

Verify Java installation

```
'java -version'
```

d. Add Jenkins Repository:

i. First, add the Jenkins Debian repository key:

```
''
```

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
'''
```

- ii. Then, add the Jenkins repository to the system:

```

```
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

```

- e. Install Jenkins:

```

```
sudo apt-get update
sudo apt-get install jenkins
```
```

3. Start Jenkins:

- a. Start and enable Jenkins to run on system boot:

```

```
sudo systemctl start jenkins
sudo systemctl enable jenkins
```
```

- b. Check Jenkins status:

```

```
sudo systemctl status jenkins
```
```

4. Access Jenkins Web UI

- a. Open your browser and access Jenkins using:

'<http://your-ec2-public-ip:8080>'

- b. Unlock Jenkins:

i) You'll be prompted to enter an initial admin password.

ii) Run the following command to get the password:

'`sudo cat /var/lib/jenkins/secrets/initialAdminPassword`'

iii) Copy and paste the password into the Jenkins setup screen.

- c. Install Suggested Plugins when prompted.

- d. Create First Admin User and finish the setup.

Step 2: Setup Jenkins slave instance on different EC2 instance:

1. Launch another EC2 instance:

AMI=22.04, t2.micro with Security group having inbound rule for SSH i.e. port = 22, Jenkins web app port(Java applications port) i.e. port = 8080. All traffic should be allowed to these 2 ports(22,8080)

2. Install required build tool:

- a. Jenkins Master node is used to manage overall jenkins ops., Scheduling jobs, distributing work to slaves, etc. So Jenkins will only be installed on Jenkins Master node host instance. On the slave machines only required build tools are installed. Slave node instance is to perform tasks assigned by the master instance. This architecture results in parallel executions of multiple jobs and improving scalability and performance

- b. For our use case we will installing ‘JDK’

...

```
sudo -i  
apt update -y  
apt update -y  
apt install openjdk-17-jre -y  
java -version  
apt install git -y  
...
```

3. Create a Linux user for slave agent:

Create a dedicated Linux user to perform assigned builds/tasks. Create User in Jenkins Slave Machine & Create SSH Keys

...

```
useradd devopsadmin -s /bin/bash -m -d /home/devopsadmin  
su - devopsadmin  
ssh-keygen -t ecdsa -b 521  
ls ~/.ssh  
cd .ssh  
cat id_ecdsa.pub > authorized_keys  
chmod 600 /home/devopsadmin/.ssh/*
```

...

Step 3: Attach Jenkins Slave node EC2 instance to Jenkins Master node EC2 instance:

1. Login to Jenkins (Master node)
2. Click on 'Manage Jenkins > New Node'
3. Give appropriate Node name and set as permanent agent:
 - a. Give appropriate Node name like 'SlaveNode1' and select as permanent agent and click next to continue
4. Fill necessary details in Node configuration page
 - a. Fill description as 'Slave node 1 for making maven builds'
 - b. Fill number of executors as per need. I filled it to '2'
 - c. Remote root directory: '/home/devopsadmin'
 - d. Labels: 'slave1'
 - e. Launch method: 'Launch Agent via SSH'
 - f. Host: Private IP address of slave node EC2 instance
 - g. Credentials > Add
 - i. Domain: 'Global credentials unrestricted'
 - ii. Kind: 'SSH username with private key'
 - iii. Scope: 'Global(Jenkins,nodes,items,all child items,etc)'
 - iv. ID: Anything e.g. 'slavenode1cred'
 - v. Description: Anything e.g. 'slavenode1cred'
 - vi. Username: User name of the user in slave node EC2 linux instance that we created for build activities. In our case 'devopsadmin'
 - vii. Private key: Check 'Enter Directly'
 - viii. Add private key
 - ix. Copy the 'private key' from the desired user from the slave node EC2 instance by SSH into instance and then login as root using `sudo -i` then login as that user `su - devopsadmin` then changing directory to .ssh by `cd .ssh` and copying private key after entering cmd `cat id_ecdsa`
 - x. Paste the copied private of dedicated user

- xi. Enter passphrase if any
- xii. Click 'Save' to continue

- h. Select the newly created credentials: 'devopsadmin(slavenode1cred)'
- i. Host Key Verification Strategy: 'Manually trusted key Verification Strategy'
- j. Require manual verification of initial connection: Enable this checkbox
- k. Click on 'save' to continue

Step 4: Complete manual verification of initial connection to slave node:

1. Navigate to 'Dashboard > Manage Jenkins > Nodes'
2. Click on the newly created Jenkins node
3. On the left pane select 'Trust SSH host key'
4. Click 'Yes' to verify the host key

Step 5: Create a freestyle project:

1. Go to New Item > Freestyle Project.
 - a. Enter the project name (e.g., DemoFreestyleProject1) and select Freestyle Project.
 - b. Click 'Ok'

2. Configure the Project:
 - a. Under "General" select the checkbox "Restrict where this project can be run"
 - b. Enter the label you gave to the Slave node (e.g., SlaveNode1).

3. Define a Simple Build Step:
 - a. Scroll down to the "Build" section.
 - b. Click "Add Build Step > Execute Shell"

 - c. Enter a simple command like
`echo "Running on Slave Node 1" `

4. Save and Build the Project:
 - a. Click "Save"
 - b. Click "Build Now"

5. Check the Console Output:

- a. Go to the “Build History” of the project.
- b. Click on the build number and select “Console Output”
- c. You should see the message “Running on Slave Node 1” and confirm that the job ran on the slave node.

Step 6: Verify Workspace Creation on Slave Node:

1. SSH into the Slave EC2 Instance:

```
` ssh -i "your-slave-key.pem" ubuntu@<slave-public-ip>`
```

2. Navigate to the Workspace Directory:

```
```
```

```
sudo -i
```

```
su – devopsadmin
```

```
cd /home/devopsadmin/workspace/
```

```
ls
```

```
```
```

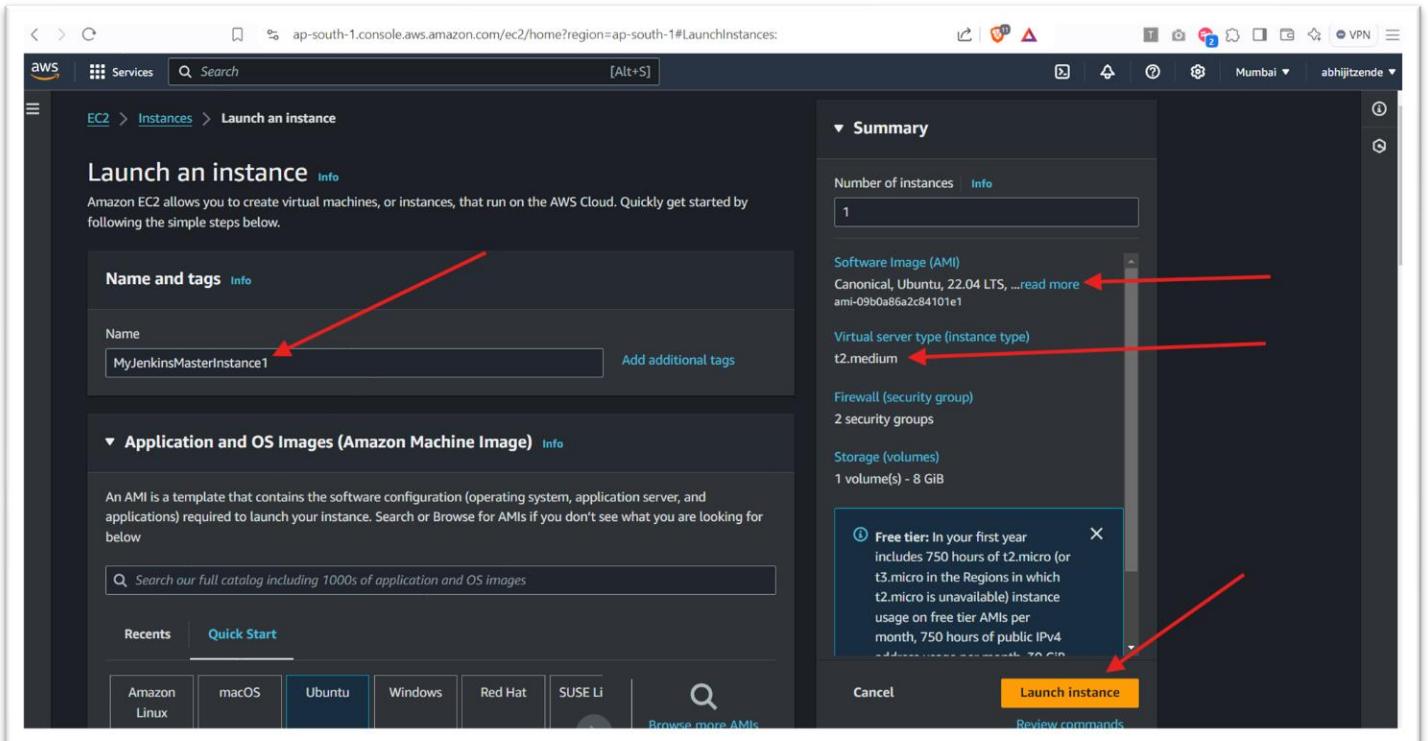


Fig. 1.01: Launch AWS EC2 instance for Jenkins Master node

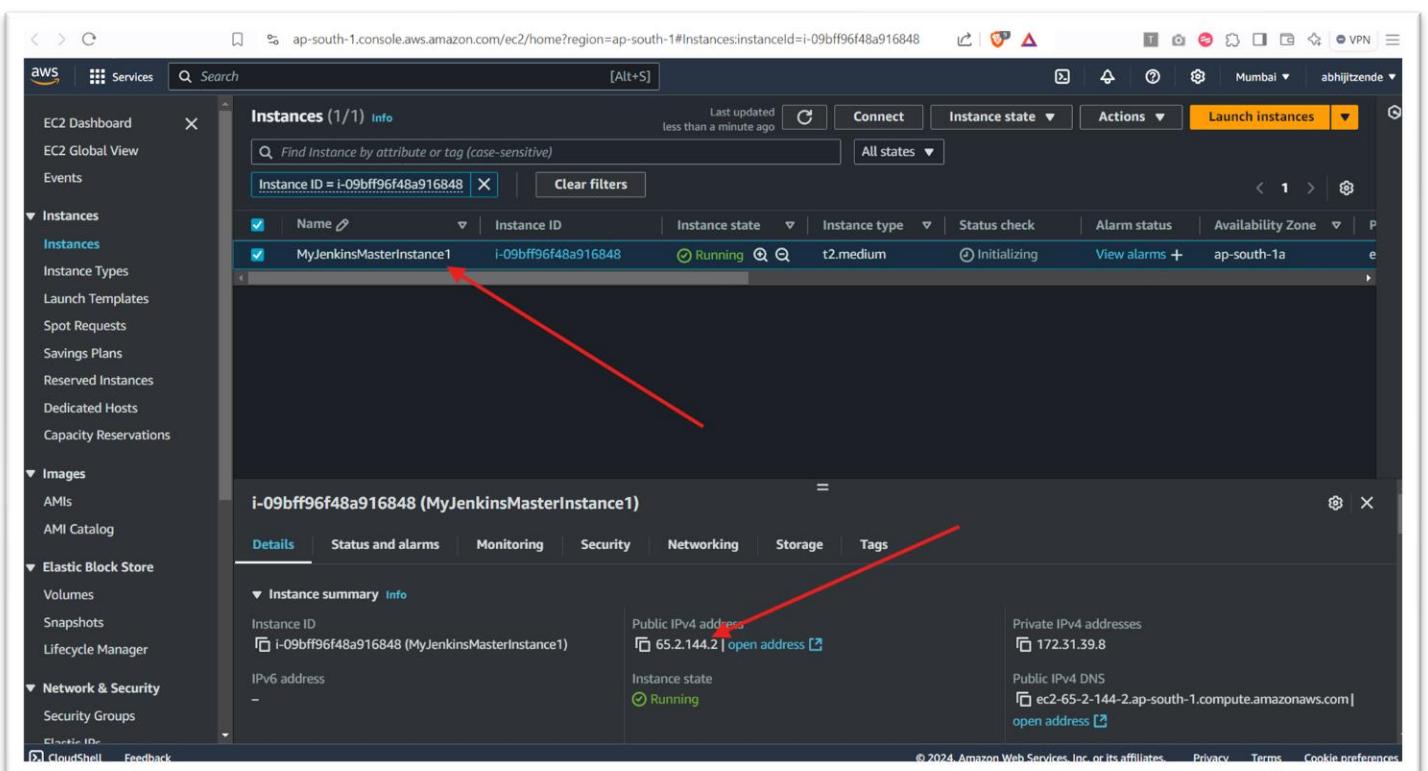


Fig. 1.02: Copy public IPv4 address of Jenkins Master AWS EC2 instance

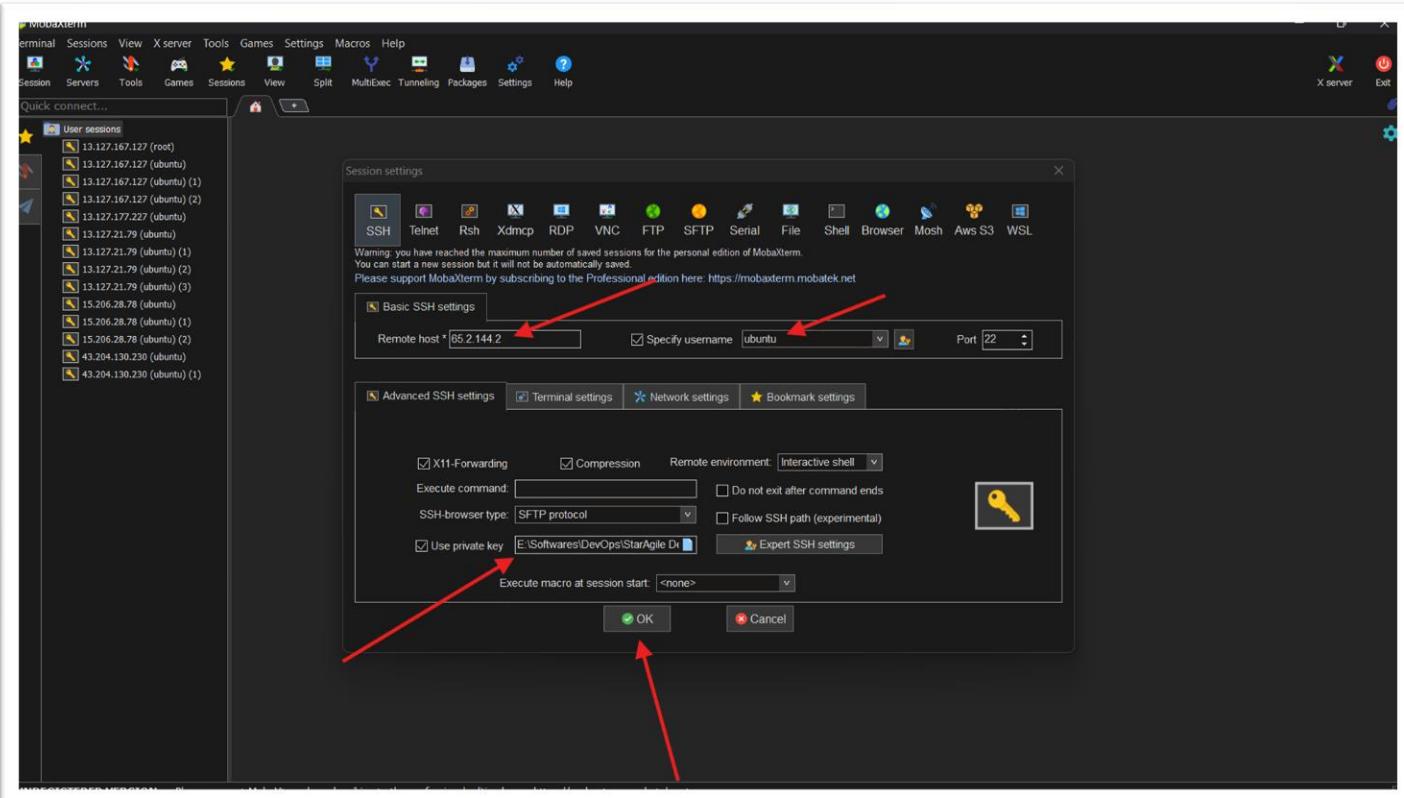


Fig. 1.03: SSH into the Jenkins Master AWS EC2 instance using MobaXterm

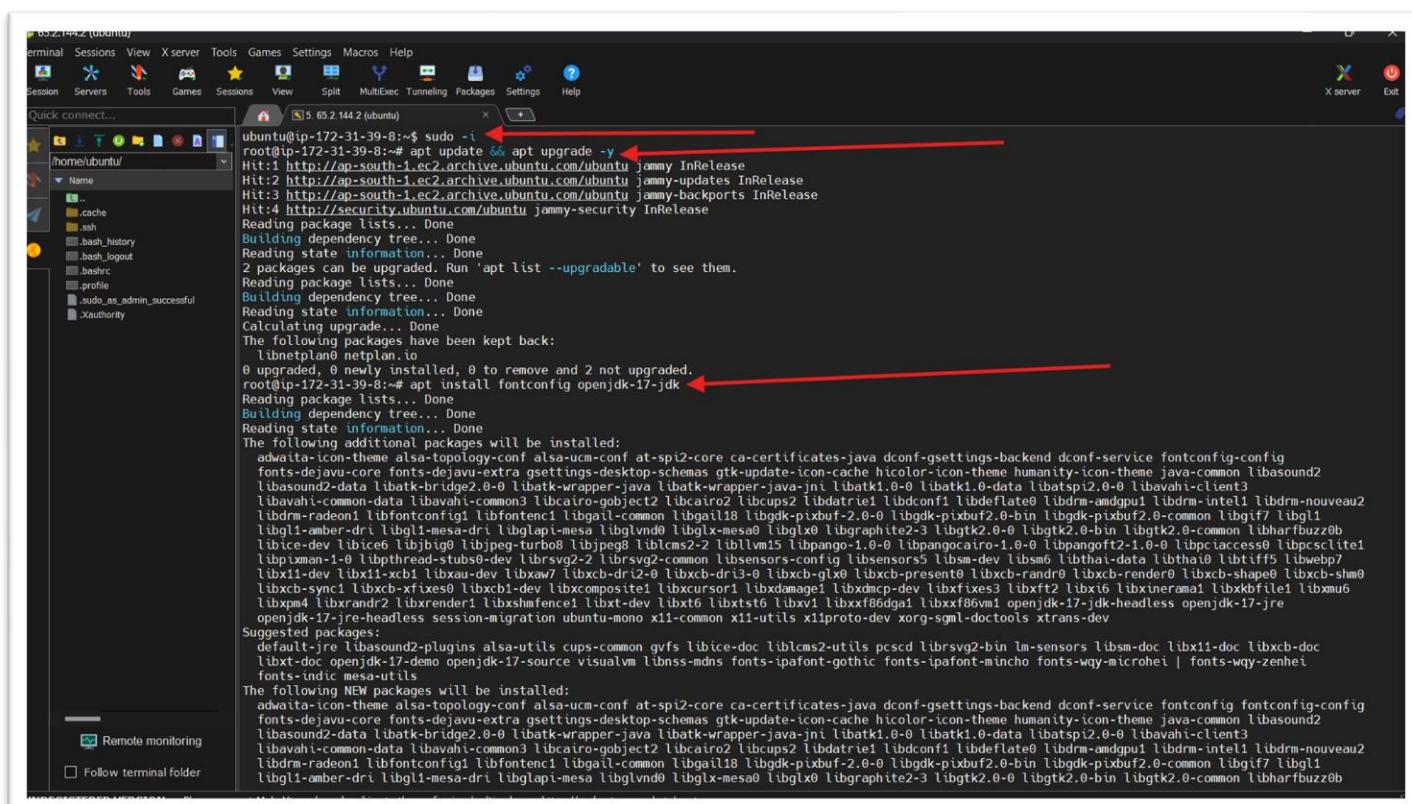


Fig. 1.04: Updating and installing required libraries (JDK-17)

```

0.0.2.144.2 (Ubuntu)
terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 5 65.2.144.2 (ubuntu) X server Exit
/home/ubuntu/
Name
.. .cache .ssh .bash_history .bash_logout .bashrc .profile .sudo_as_admin_successful .Xauthority
Setting up at-sp12-core (2.44.0-3) ...
Processing triggers for libgdk-pixbuf2.0-0:amd64 (2.42.8+dfsg-1ubuntu0.3) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-39-8:~# java -version
openjdk version "17.0.12" 2024-07-16
OpenJDK Runtime Environment (build 17.0.12+7-Ubuntu-1ubuntu22.04, mixed mode, sharing)
OpenJDK 64-Bit Server VM (build 17.0.12+7-Ubuntu-1ubuntu22.04, mixed mode, sharing)
root@ip-172-31-39-8:~# sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
--2024-10-05 14:42:11-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.154.133, 2a04:4e42:24::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.154.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins-keyring.asc 100%[=====] 3.19K --.-KB/s in 0s
2024-10-05 14:42:11 (36.2 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]

root@ip-172-31-39-8:~# echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/" | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
root@ip-172-31-39-8:~# apt-get update
Hit:1 http://ppa-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ppa-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ppa-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Hit:7 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [27.9 kB]
Fetched 30.8 kB in 1s (56.3 kB/s)
Reading package lists... Done

```

Remote monitoring Follow terminal folder

Fig. 1.05: Adding Jenkins repository

```

0.0.2.144.2 (Ubuntu)
terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 5 65.2.144.2 (ubuntu) X server Exit
/home/ubuntu/
Name
.. .cache .ssh .bash_history .bash_logout .bashrc .profile .sudo_as_admin_successful .Xauthority
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Hit:7 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [27.9 kB]
Fetched 30.8 kB in 1s (56.3 kB/s)
Reading package lists... Done
root@ip-172-31-39-8:~# apt-get install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 2 not upgraded.
Need to get 91.5 MB of archives.
After this operation, 94.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ppa-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.462.3 [91.3 MB]
Fetched 91.5 MB in 7s (12.6 MB/s)
Selecting previous unselected package net-tools.
(Reading database ... 82134 files and directories currently installed.)
Preparing to unpack .../net-tools 1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.462.3_all.deb ...
Unpacking jenkins (2.462.3) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.462.3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-39-8:~#

```

Remote monitoring Follow terminal folder

Fig. 1.06: Installing Jenkins

The screenshot shows a terminal window titled "65.2.144.2 (ubuntu)" with a dark theme. The user is root. The terminal output shows the following commands and their results:

```

Unpacking jenkins (2.462.3) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-39-8:~# whoami
root
root@ip-172-31-39-8:~# systemctl start jenkins
root@ip-172-31-39-8:~# systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
● jenkins.service - Jenkins Continuous Integration Server
    Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
      Active: active (running) since Sat 2024-10-05 14:43:50 UTC; 4min 42s ago
        Main PID: 6278 (java)
          Tasks: 46 (limit: 4676)
            Memory: 623.1M
              CPU: 19.951s
            CGroup: /system.slice/jenkins.service
                └─6278 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Oct 05 14:43:27 ip-172-31-39-8 jenkins[6278]: a6b0149c3e45404898be6713d87b5dad
Oct 05 14:43:27 ip-172-31-39-8 jenkins[6278]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Oct 05 14:43:27 ip-172-31-39-8 jenkins[6278]: ****
Oct 05 14:43:50 ip-172-31-39-8 jenkins[6278]: 2024-10-05 14:43:50.207+0000 [id=33]      INFO  jenkins.InitReactorRunner$1@onAttained: Completed initialization
Oct 05 14:43:50 ip-172-31-39-8 jenkins[6278]: 2024-10-05 14:43:50.224+0000 [id=24]      INFO  hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
Oct 05 14:43:50 ip-172-31-39-8 systemd[1]: Started Jenkins Continuous Integration Server.
Oct 05 14:43:51 ip-172-31-39-8 jenkins[6278]: 2024-10-05 14:43:51.007+0000 [id=49]      INFO  h.m.DownloadService$Downloadable#load: Obtained the update
Oct 05 14:43:51 ip-172-31-39-8 jenkins[6278]: 2024-10-05 14:43:51.008+0000 [id=49]      INFO  hudson.util.Retriger#start: Performed the action check up
[lines 1-20/20 (END)]

```

Annotations in red arrows point to the command "systemctl enable jenkins", the command "systemctl status jenkins", and the log entry "This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword".

Fig. 1.07: Enabling Jenkins on startup

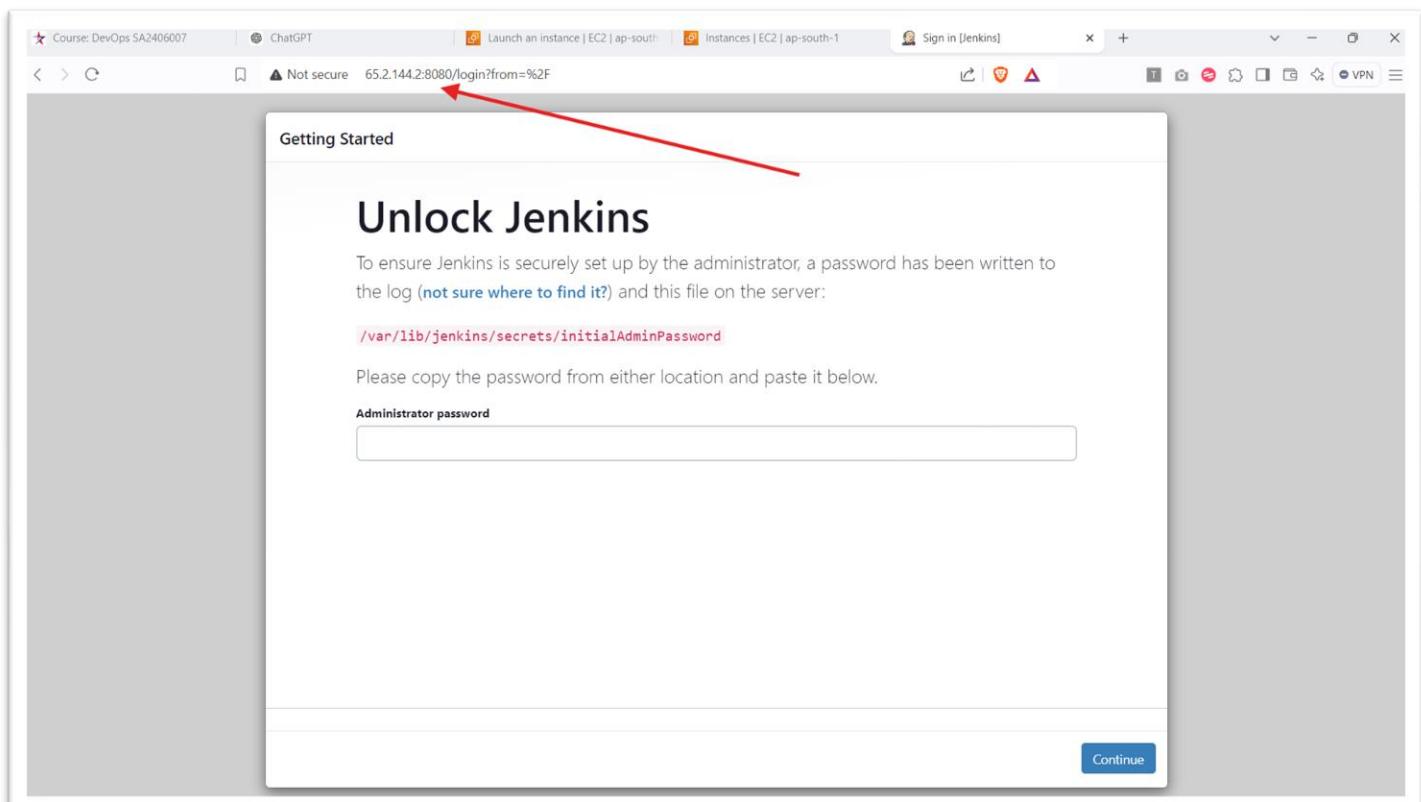


Fig. 1.08: Unlocking Jenkins

65.2.144.2 (ubuntu)

terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

X server Exit

Quick connect... 5 65.2.144.2 (ubuntu) x

```

/home/ubuntu/
└── Name
    ├── .., .cache, .ssh, .bash_history, .bash_logout, .bashrc, .profile, .sudo_as_admin_successful, .Xauthority
    └── Running kernel seems to be up-to-date.
    └── No services need to be restarted.
    └── No containers need to be restarted.
    └── No user sessions are running outdated binaries.
    └── No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-39-8:~# whoami
root
root@ip-172-31-39-8:~# systemctl start jenkins
root@ip-172-31-39-8:~# systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
root@ip-172-31-39-8:~# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
    Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
    Active: active (running) since Sat 2024-10-05 14:43:50 UTC; 4min 42s ago
      Main PID: 6278 (java)
        Tasks: 46 (limit: 4676)
       Memory: 623.1M
          CPU: 19.951s
         CGroup: /system.slice/jenkins.service
                   └─6278 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Oct 05 14:43:27 ip-172-31-39-8 jenkins[6278]: a6b149c3e45404898be6713d87b5dad
Oct 05 14:43:27 ip-172-31-39-8 jenkins[6278]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Oct 05 14:43:27 ip-172-31-39-8 jenkins[6278]: ****
Oct 05 14:43:50 ip-172-31-39-8 jenkins[6278]: 2024-10-05 14:43:50.207+0000 [id=33] INFO Jenkins.InitReactorRunner$1#onAttained: Completed init
Oct 05 14:43:50 ip-172-31-39-8 jenkins[6278]: 2024-10-05 14:43:50.224+0000 [id=24] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up
Oct 05 14:43:50 ip-172-31-39-8 systemd[1]: Started Jenkins Continuous Integration Server.
Oct 05 14:43:51 ip-172-31-39-8 jenkins[6278]: 2024-10-05 14:43:51.007+0000 [id=49] INFO h.m.DownloadService$Downloadable#load: Obtained the up
Oct 05 14:43:51 ip-172-31-39-8 jenkins[6278]: 2024-10-05 14:43:51.008+0000 [id=49] INFO hudson.util.Retrier#start: Performed the action check up

root@ip-172-31-39-8:~# cat /var/lib/jenkins/secrets/initialAdminPassword
a6b149c3e45404898be6713d87b5dad
root@ip-172-31-39-8:~#
```

Remote monitoring Follow terminal folder

Fig. 1.09: Copying initial admin password

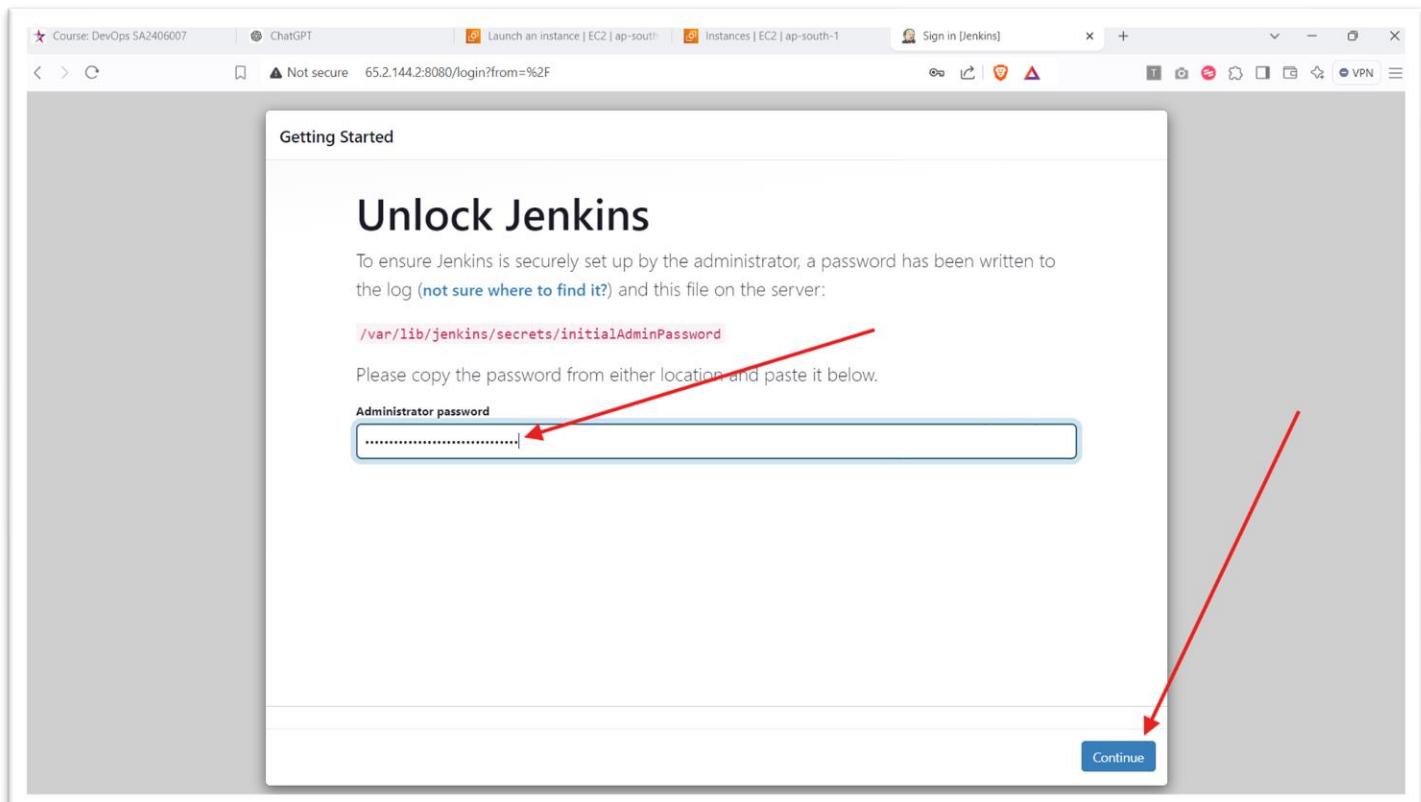


Fig. 1.10: Paste initial admin password and continue

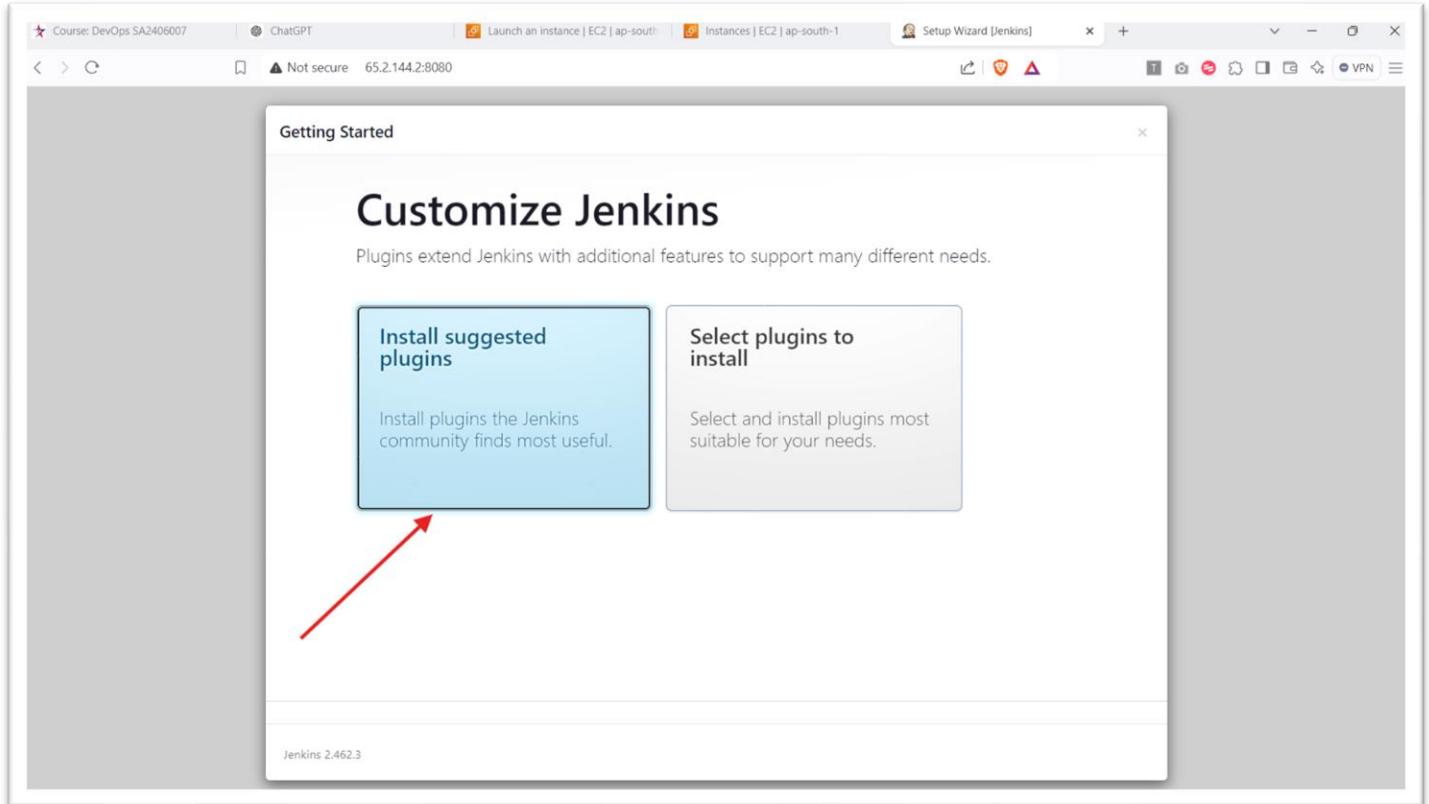


Fig. 1.11: Install suggested plugins

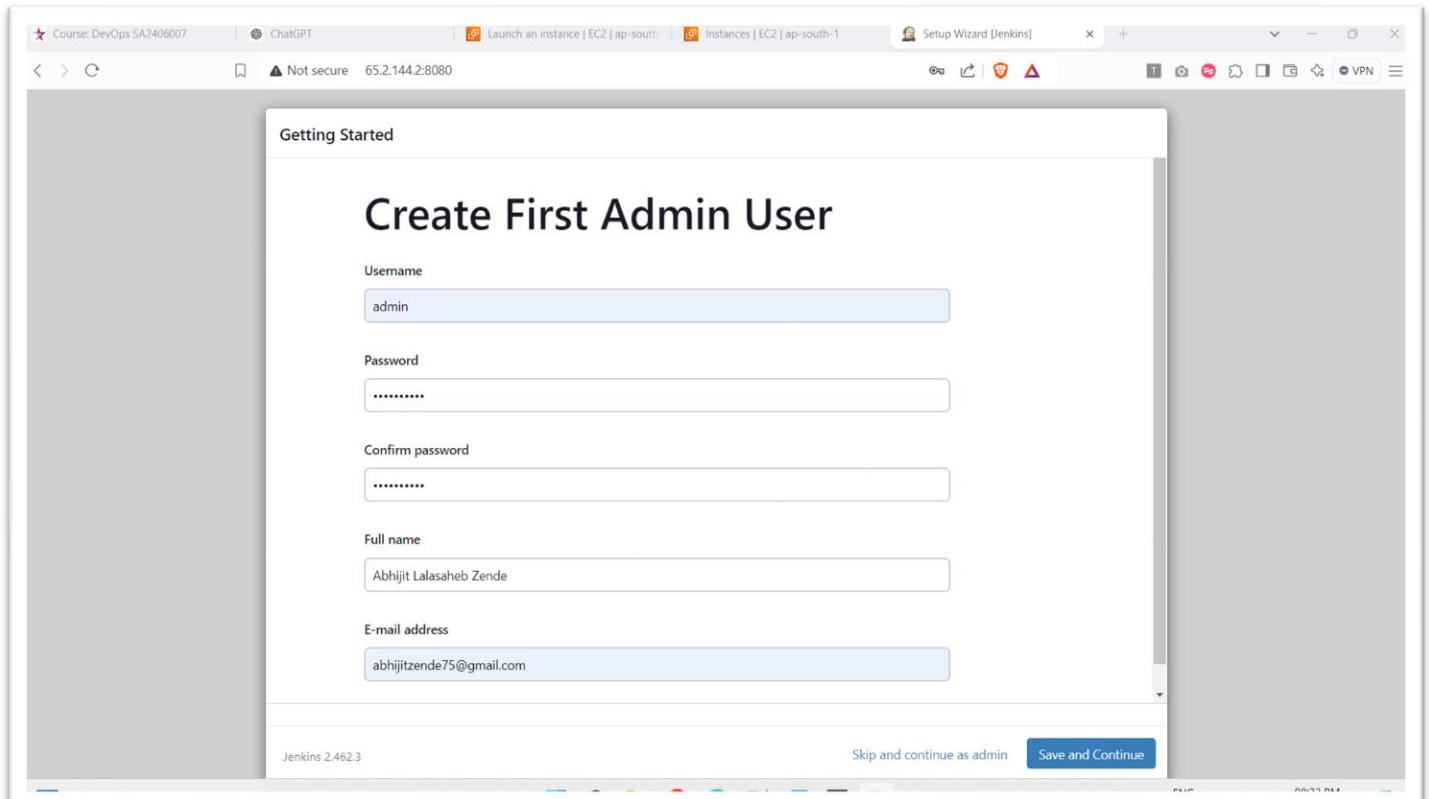


Fig. 1.12: Create first admin user

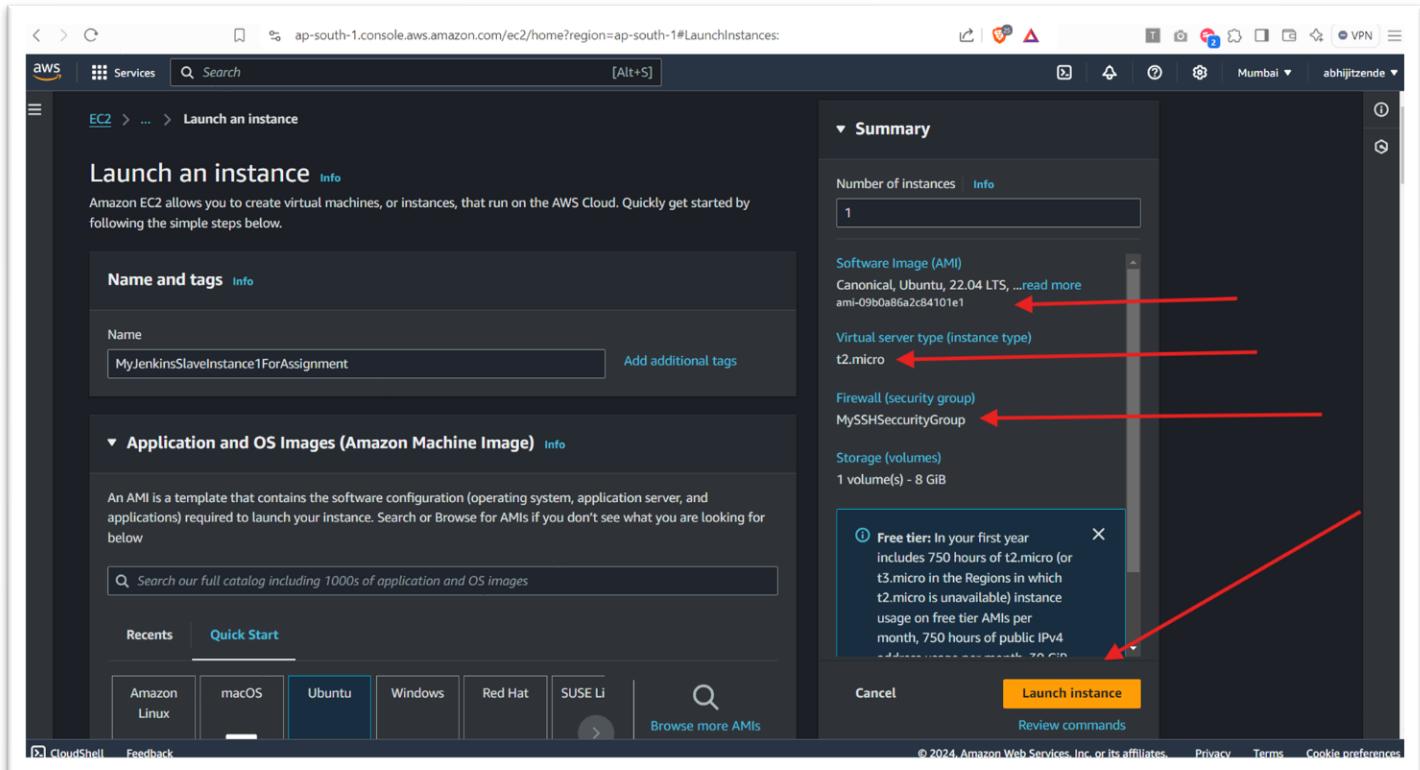


Fig. 1.12: Launch new AWS EC2 instance for Jenkins slave node

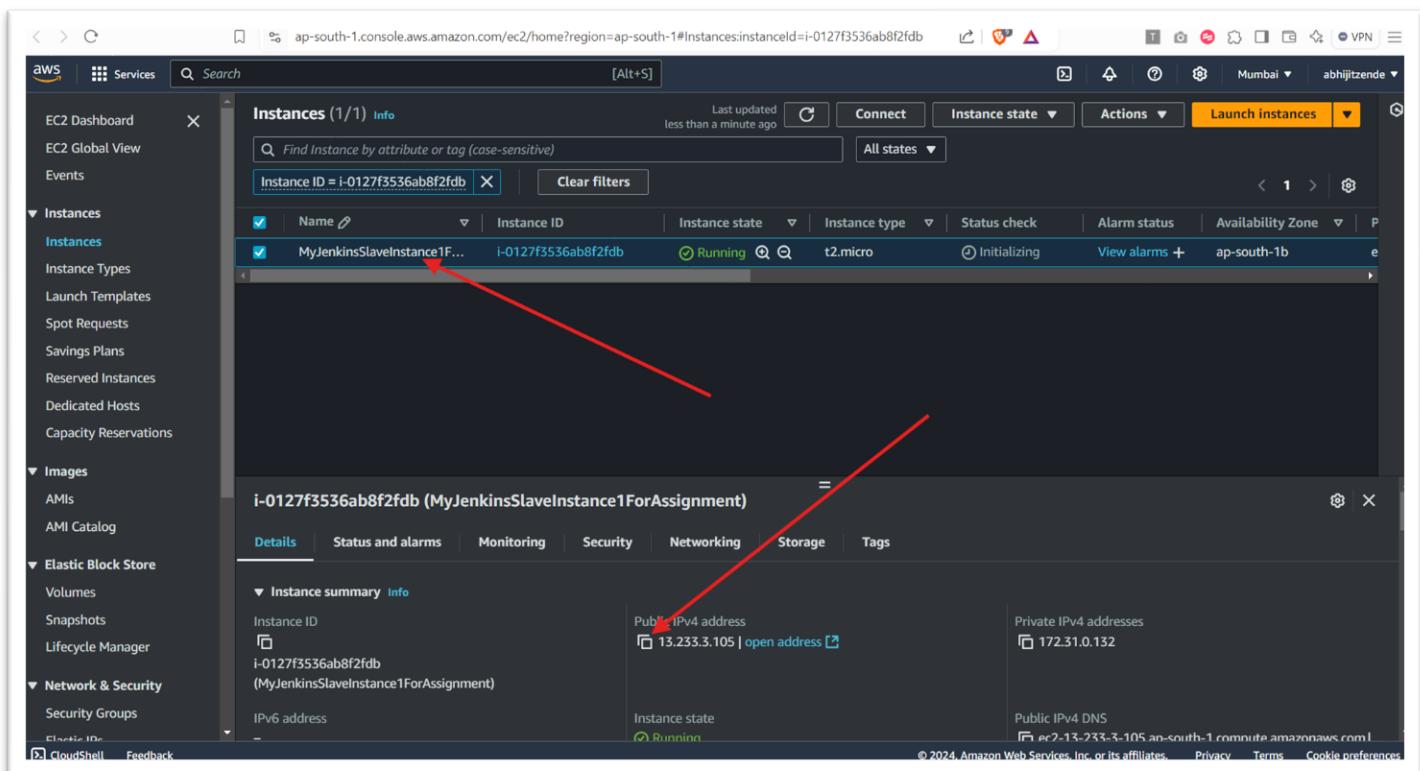


Fig. 1.13: Copy public Ipv4 address of new instance

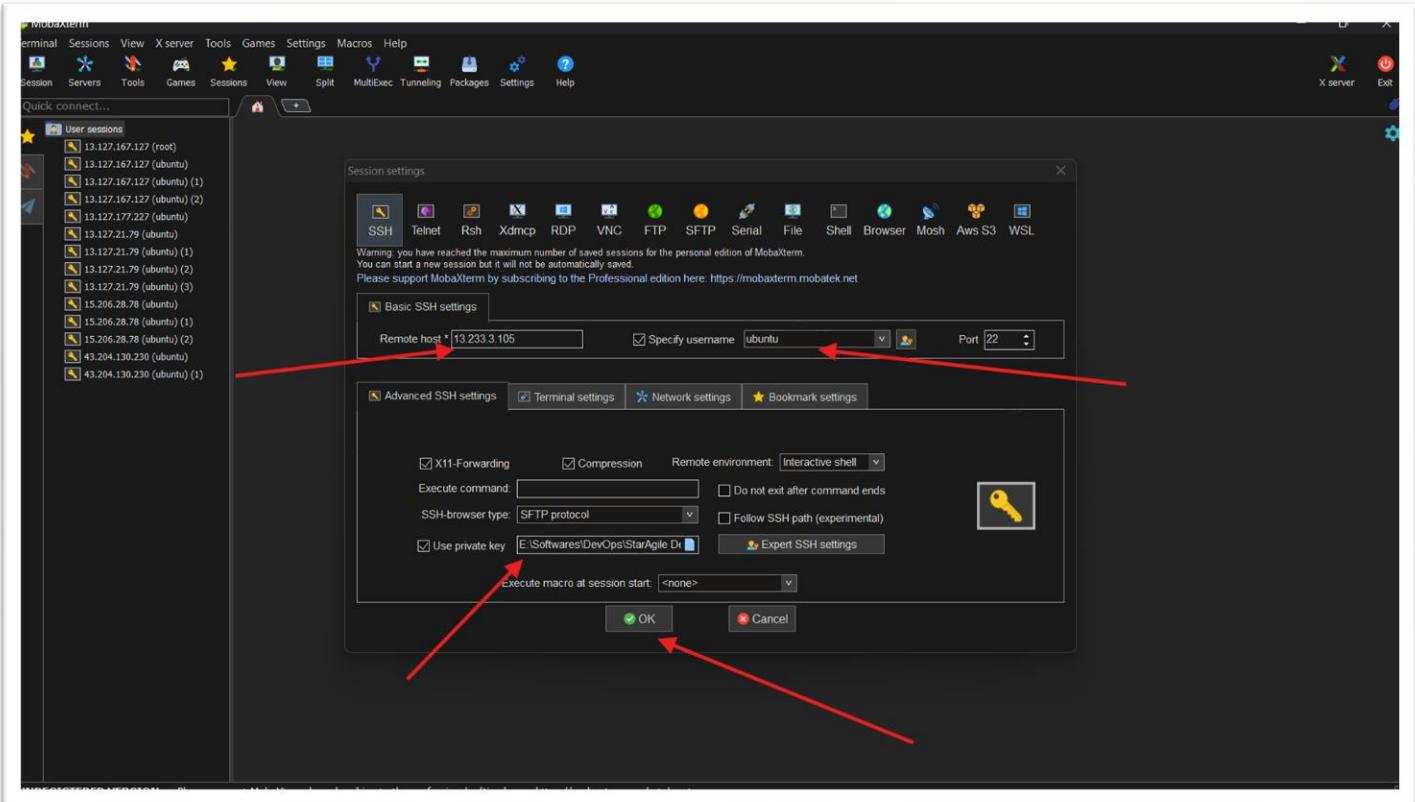


Fig. 1.14: SSH into the new instance using MobaXterm

```
ubuntu@ip-172-31-0-132:~$ whoami
ubuntu
ubuntu@ip-172-31-0-132:~$ sudo -i
sudo: invalid option -- i
usage: sudo -h [-K | -k | -V]
usage: sudo -v [-ABknS] [-g group] [-h host] [-p prompt] [-u user] [command]
usage: sudo -u [-ABknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
usage: sudo [-ABbhknPS] [-r role] [-t type] [-c num] [-D directory] [-g group] [-h host] [-p prompt] [-R directory] [-T timeout] [-u user] [VAR=value]
      [-i|-s] [<command>]
usage: sudo [-ABknS] [-r role] [-t type] [-c num] [-D directory] [-g group] [-h host] [-p prompt] [-R directory] [-T timeout] [-u user] file ...
ubuntu@ip-172-31-0-132:~$ sudo -i
root@ip-172-31-0-132:~# apt update && apt upgrade -y
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [127 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2067 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [358 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [17.9 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1130 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [264 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [26.3 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [43.3 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.8 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [444 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [67.7 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [111.1 kB]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [28.8 kB]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.5 kB]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-security/main amd64 Packages [672 B]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:28 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1848 kB]
Get:29 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [299 kB]
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [13.3 kB]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [99 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [179 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [19.4 kB]
```

Fig. 1.15: Update and upgrade

```

root@ip-172-31-0-132:~# whoami
root
root@ip-172-31-0-132:~$ sudo -i
root@ip-172-31-0-132:~# apt upgrade -y
[...]
root@ip-172-31-0-132:~# apt install openjdk-17-jdk -y
root@ip-172-31-0-132:~# git --version
git version 2.34.1
root@ip-172-31-0-132:~#

```

Fig. 1.16: Install JDK and Git

```

root@ip-172-31-0-132:~# useradd devopsadmin -s /bin/bash -m -d /home/devopsadmin
root@ip-172-31-0-132:~# su - devopsadmin
devopsadmin@ip-172-31-0-132:~$ ssh-keygen -t ecdsa -b 521
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/devopsadmin/.ssh/id_ecdsa):
Created directory '/home/devopsadmin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/devopsadmin/.ssh/id_ecdsa.
Your public key has been saved in /home/devopsadmin/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:P2BFeylJ6tvSBTkvF5o59BvOIDA66zCswfB9AmVFNO devopsadmin@ip-172-31-0-132
The key's randomart image is:
----[ECDSA 521]----+
 ..++ .o= |
 .. ..E o |
 .o .o. @ |
 .o .o .o. |
 .o .o oo S |
 .o .o .o+ |
 .o .o .o+ |
 .o .o .o= |
 .o .o .o+ |
----[SHA256]----+
devopsadmin@ip-172-31-0-132:~$ ls ~/.ssh
id_ecdsa id_ecdsa.pub
devopsadmin@ip-172-31-0-132:~$ cd .ssh
devopsadmin@ip-172-31-0-132:~/ssh$ cat id_ecdsa.pub > authorized_keys
devopsadmin@ip-172-31-0-132:~/ssh$ chmod 600 /home/devopsadmin/.ssh/*
devopsadmin@ip-172-31-0-132:~/ssh$ ll
total 20
rwx----- 2 devopsadmin devopsadmin 4096 Oct  5 15:50 ./
rwxr-x--- 3 devopsadmin devopsadmin 4096 Oct  5 15:49 ../
rw----- 1 devopsadmin devopsadmin 281 Oct  5 15:50 authorized_keys
rw----- 1 devopsadmin devopsadmin 748 Oct  5 15:49 id_ecdsa
rw----- 1 devopsadmin devopsadmin 281 Oct  5 15:49 id_ecdsa.pub
devopsadmin@ip-172-31-0-132:~/ssh$ 

```

Fig. 1.17: Create a new Linux user

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below these are two dropdown sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). The main content area has a heading 'Welcome to Jenkins!' followed by a sub-section 'Start building your software project'. It includes a 'Create a job' button, a 'Set up a distributed build' section with 'Set up an agent' and 'Configure a cloud' buttons, and a link 'Learn more about distributed builds'. At the bottom right, there are links for 'REST API' and 'Jenkins 2.462.3'.

Fig. 1.18: Manage Jenkins page

This screenshot shows the 'Manage Jenkins' page. The 'Manage Jenkins' link in the sidebar is highlighted with a red arrow pointing to it. The main content area is titled 'Manage Jenkins' and features a 'System Configuration' section with icons for 'System', 'Tools', 'Nodes', 'Clouds', 'Plugins', and 'Appearance'. Below this is a 'Security' section with icons for 'Security', 'Credentials', and 'Credential Providers'. A message at the top right says 'Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.' with buttons for 'Set up agent', 'Set up cloud', and 'Dismiss'.

Fig. 1.19: Manage Jenkins > Nodes

The screenshot shows the Jenkins 'Nodes' page. On the left, there are two collapsed sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). The main area is titled 'Nodes' and contains a table with one row for 'Built-In Node'. The columns are: S (Status), Name (Built-In Node), Architecture (Linux (amd64)), Clock Difference (In sync), Free Disk Space (4.75 GiB), Free Swap Space (10 B), Free Temp Space (4.75 GiB), and Response Time (0ms). A red arrow points from the top right towards the '+ New Node' button. The bottom right corner shows 'REST API' and 'Jenkins 2.462.3'.

Fig. 1.20: Adding New node

The screenshot shows the 'New node' configuration page. It has a 'Node name' field containing 'SlaveNode1' with a red arrow pointing to it. Below it is a 'Type' section with a radio button for 'Permanent Agent' (selected) and a description explaining it adds a plain, permanent agent to Jenkins. At the bottom is a 'Create' button with a red arrow pointing to it. The bottom right corner shows 'REST API' and 'Jenkins 2.462.3'.

Fig. 1.21: Setting New node name and making it permanent agent

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, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, and CloudShell. The main area displays a table of instances. One instance, 'MyJenkinsSlaveInstance1F...', is selected. A red arrow points to the 'Private IPv4 addresses' field in the 'Instance summary' section, which contains the value '172.31.0.132'.

Fig. 1.22: Copy Private IPv4 address of AWS EC2 instance for Jenkins slave node

```

devopsadmin@ip-172-31-0-132:~$ ssh
ubuntu@ip-172-31-0-132:~$ sudo -i
root@ip-172-31-0-132:~# su - devopsadmin
devopsadmin@ip-172-31-0-132:~$ cd .ssh/
devopsadmin@ip-172-31-0-132:~$ ls
authorized_keys  id_ecdsa  id_ecdsa.pub
devopsadmin@ip-172-31-0-132:~$ ssh cat id_ecdsa
-----BEGIN OPENSSH PRIVATE KEY-----
MIIBdAIBAAQEAj... (The key content is very long and truncated here)
-----END OPENSSH PRIVATE KEY-----
```

Fig. 1.23: SSH into the slave node AWS EC2 instance and copy the private key

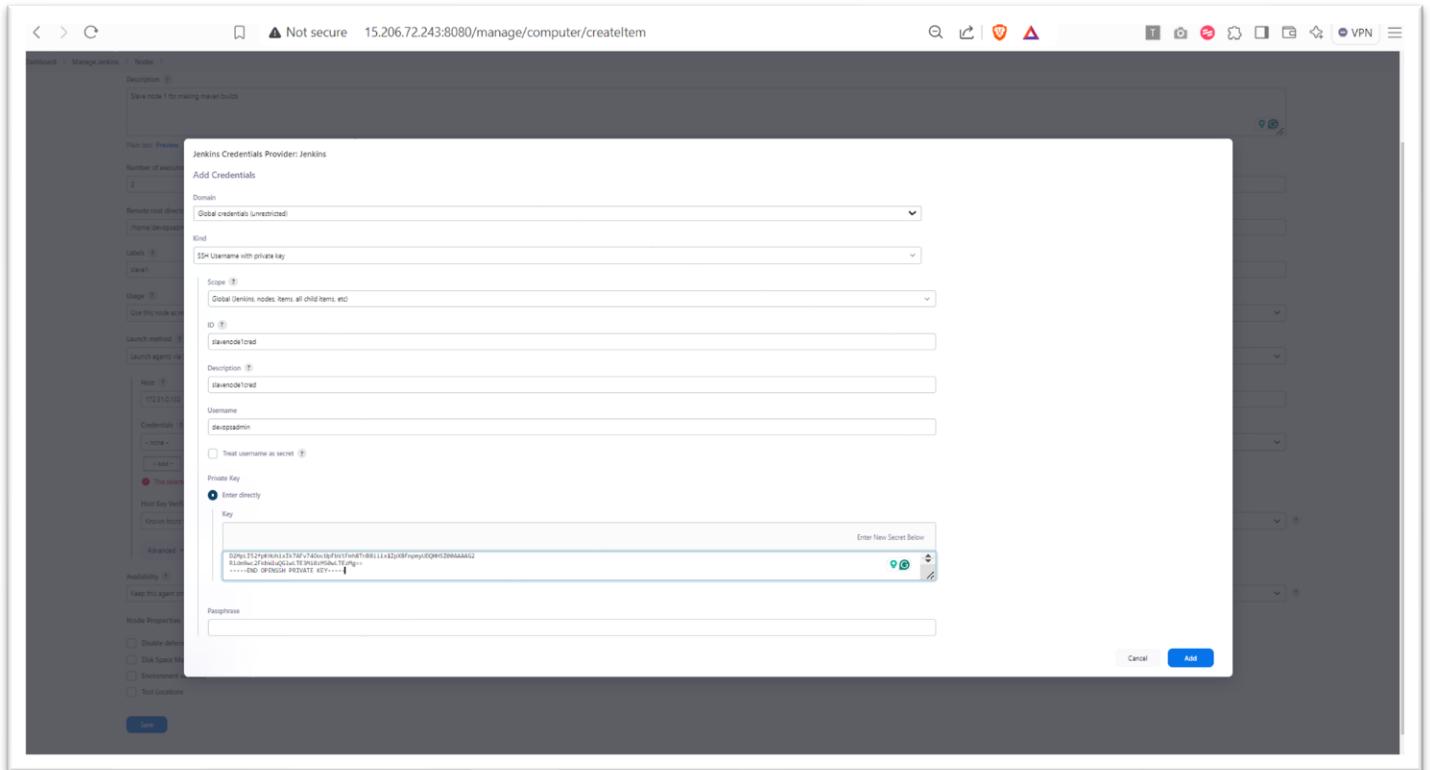


Fig. 1.24: Overview data filling of Jenkins Credentials provider



Fig. 1.25: Overview data filling of Manage Jenkins page

The screenshot shows the Jenkins 'Nodes' page. On the left, there are sections for 'Build Queue' (empty) and 'Build Executor Status' (1 Idle, 2 Idle). The main area displays a table of nodes:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.75 GiB	0 B	4.75 GiB	0ms
	SlaveNode1		N/A	N/A	N/A	N/A	N/A
			last checked	6 ms	7 ms	9 ms	7 ms
							5 ms
							1 ms

A red arrow points to the 'SlaveNode1' row. At the bottom right of the table, there is a 'Legend' section.

Fig. 1.26: Launched new node

The screenshot shows the Jenkins 'Agent SlaveNode1' page. The left sidebar includes links for Status, Delete Agent, Configure, Build History, Load Statistics, Log, and Trust SSH Host Key (which has a red arrow pointing to it). The main content area displays the agent's details:

Agent SlaveNode1

Status: Slave node 1 for making maven builds
Description: This node is being launched. See log for more details
Monitoring Data: Monitoring Data ▾

Labels: slave1

Projects tied to SlaveNode1: None

Build Executor Status: ▾

At the bottom right, there are buttons for Edit description, Relaunch agent, Mark this node temporarily offline, and a question mark icon.

Fig. 1.27: Click on 'Trust SSH Host key'

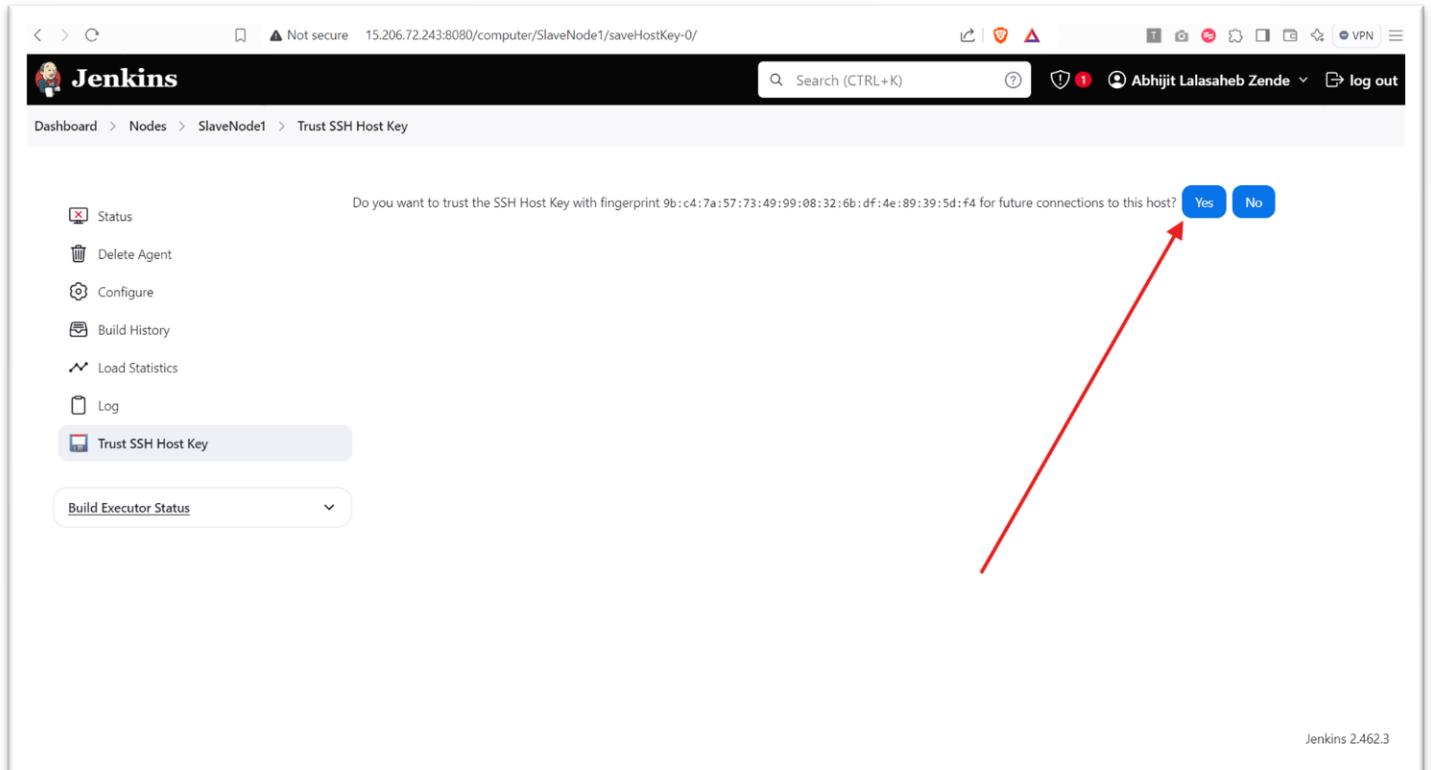


Fig. 1.28: Trust the Trust SSH Host key

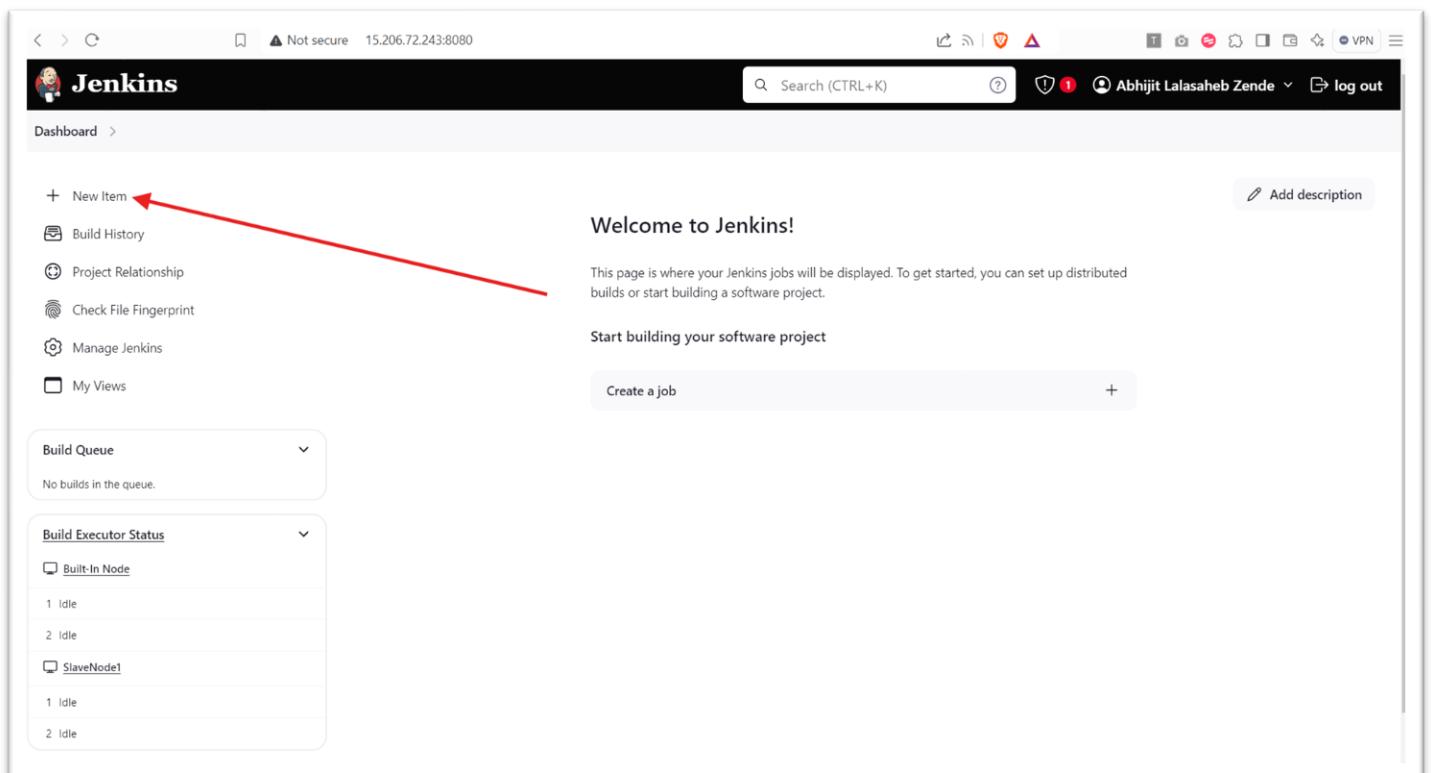


Fig. 1.29: Click on New item from Jenkins Dashboard

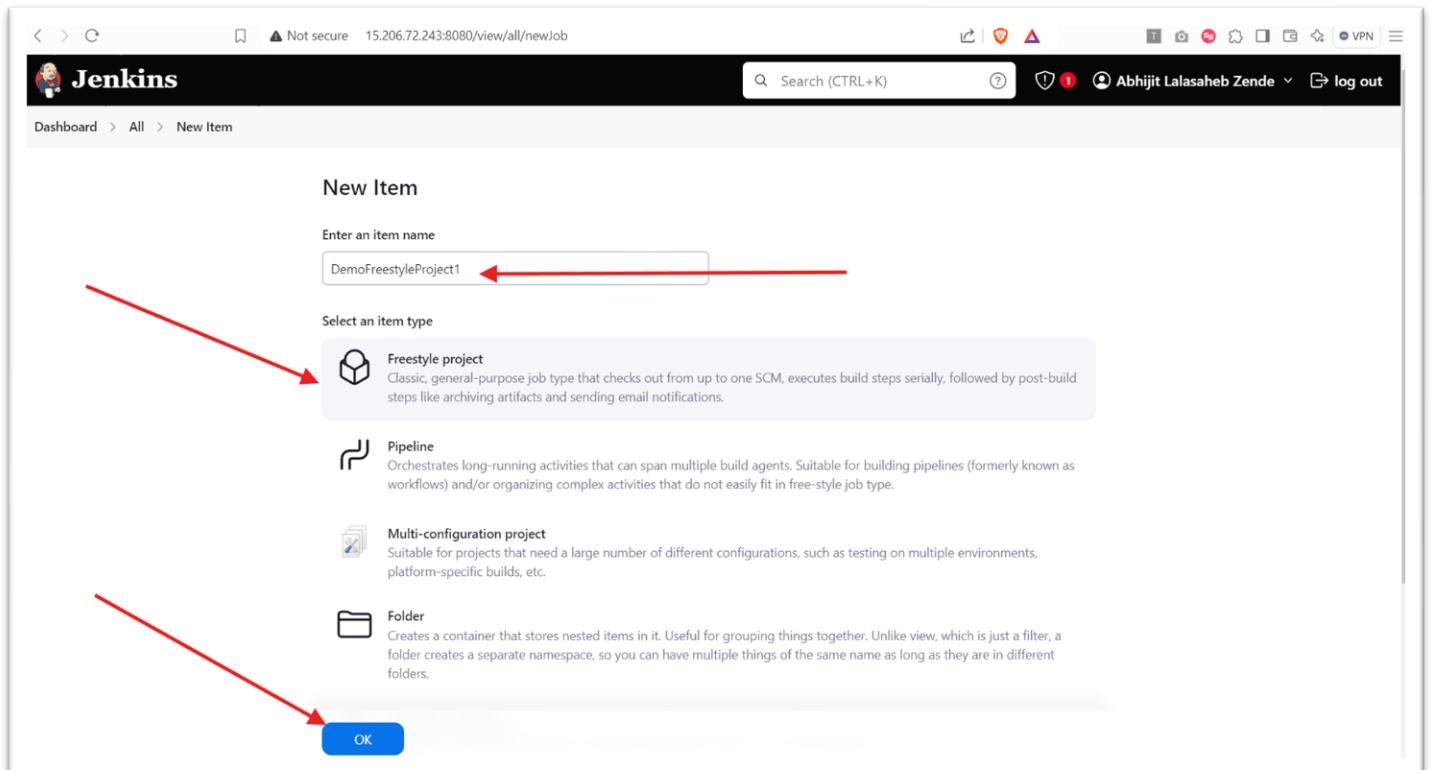


Fig. 1.30: Create a Freestyle project

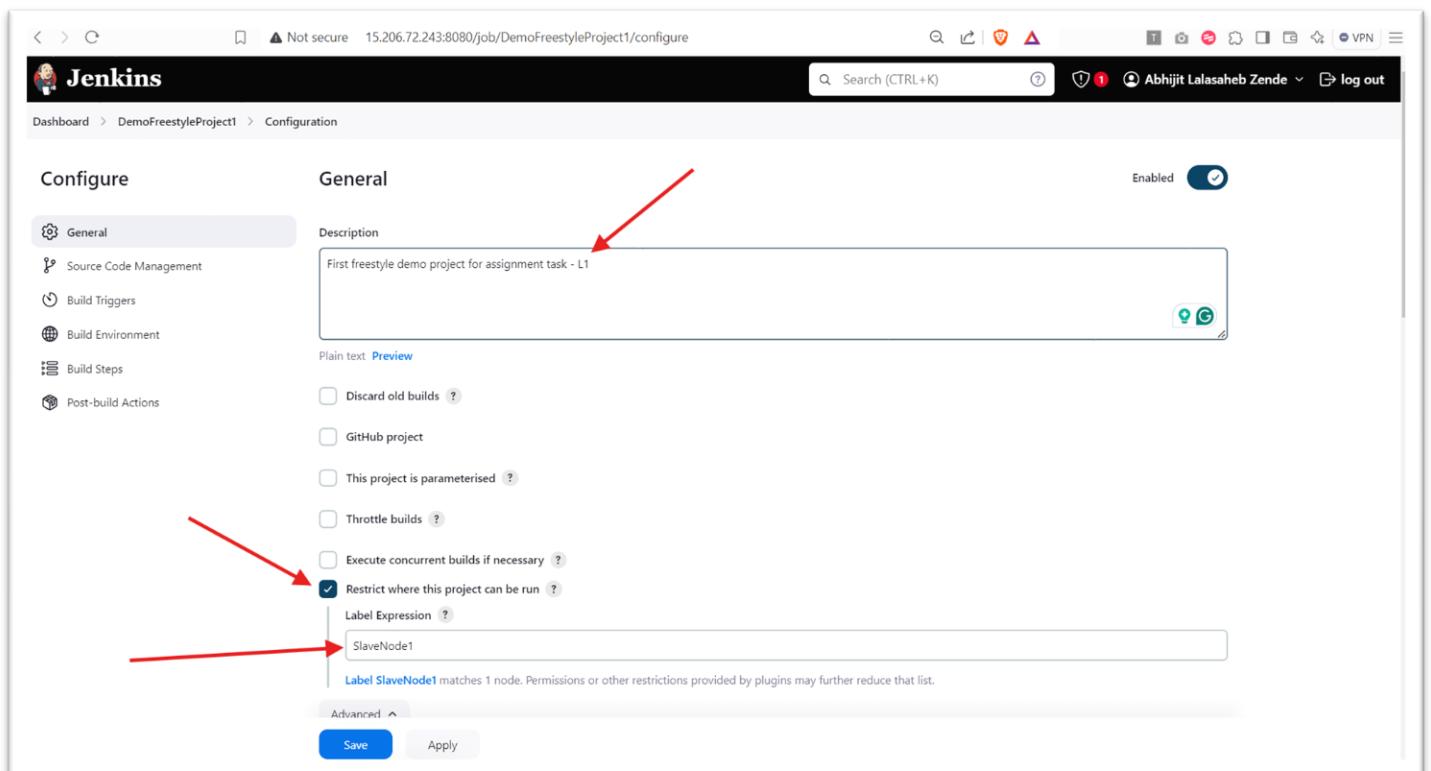


Fig. 1.31: General configuration of Freestyle project

The screenshot shows the Jenkins configuration interface for a project named 'DemoFreestyleProject1'. In the left sidebar, under 'Configure', the 'Build Environment' tab is selected. The 'Build Steps' tab is also visible. A red arrow points from the 'Add build step' button at the top of the 'Build Steps' section to a dropdown menu. The dropdown menu is titled 'Add build step ^' and contains several options: 'Execute Windows batch command', 'Execute shell' (which is highlighted with a red arrow), 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'. At the bottom right of the screen, there are links for 'REST API' and 'Jenkins 2.462.3'.

Fig. 1.31: Add a simple shell build step

The screenshot shows the Jenkins configuration interface for the same project. In the 'Build Steps' section, a 'Execute shell' step is configured with the command 'echo "Running on Slave Node 1"'. A red arrow points to the command line. Below the build steps, the 'Post-build Actions' section is shown, featuring an 'Add post-build action' dropdown. Two red arrows point to the 'Save' and 'Apply' buttons at the bottom of the configuration page.

Fig. 1.32: Hit Save

A screenshot of a Jenkins project page titled "DemoFreestyleProject1". The page shows a sidebar with options like Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. A red arrow points from the left towards the "Build Now" button. Below the sidebar is a "Build History" section with a "trend" dropdown set to "trend". The history table is empty, showing "No builds". At the bottom, there are links for "Atom feed for all" and "Atom feed for failures". The top right corner shows the user "Abhijit Lalasaheb Zende" and a "log out" link.

Fig. 1.33: Build Now

A screenshot of the same Jenkins project page after a build has been triggered. The "Build History" table now contains one entry: "#1 5 Oct 2024, 20:27". A red arrow points from the left towards this build entry. The rest of the page structure is identical to Fig. 1.33, including the sidebar, history table, and footer links.

Fig. 1.34: Successful build

The screenshot shows the Jenkins interface for a Freestyle project named 'DemoFreestyleProject1'. The build number is #1, and the 'Console Output' tab is selected. The output window displays the following log entries:

```

Started by user Abhijit Lalasaheb Zende
Running as SYSTEM
Building remotely on SlaveNode1 (slave1) in workspace /home/devopsadmin/workspace/DemoFreestyleProject1
[DemoFreestyleProject1] $ /bin/sh -xe /tmp/jenkins6339743550401635233.sh
+ echo Running on Slave Node 1
Running on Slave Node 1
Finished: SUCCESS

```

A red arrow points from the text 'Running on Slave Node 1' to the right side of the console output area.

At the bottom right of the page, there are links for 'REST API' and 'Jenkins 2.462.3'.

Fig. 1.35: Verifying console output

The screenshot shows an SSH terminal session on a Windows 10 desktop (MINGW64). The user is connected to an AWS instance. The command history and current workspace directory are as follows:

```

devopsadmin@ip-172-31-0-132: ~/workspace/DemoFreestyleProject1
$ ./aws_instance_safe_key_pair.pem ubuntu@43.204.214.49
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1015-aws x86_64)

Documentation: https://help.ubuntu.com
Management: https://landscape.canonical.com
Support: https://ubuntu.com/pro

System information as of Sat Oct 5 20:33:10 UTC 2024

System load: 0.02      Processes:           113
Usage of /: 31.7% of 7.57GB   Users logged in: 0
Memory usage: 20%
Swap usage: 0%

xpanded Security Maintenance for Applications is not enabled.
Updates can be applied immediately.

nable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

ew release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

ast login: Sat Oct 5 19:50:17 2024 from 152.58.32.37
root@ip-172-31-0-132:~ whoami
root
root@ip-172-31-0-132:~$ sudo -i
root@ip-172-31-0-132:~# su - devopsadmin
devopsadmin@ip-172-31-0-132:~$ ls /home/ubuntu/
s: cannot open directory '/home/ubuntu': Permission denied
devopsadmin@ip-172-31-0-132:~$ ls /home/devopsadmin/
emoting_remoting.jar workspace
devopsadmin@ip-172-31-0-132:~$ cd /home/devopsadmin/workspace/
devopsadmin@ip-172-31-0-132:~/workspace$ ls
DemoFreestyleProject1
devopsadmin@ip-172-31-0-132:~/workspace$ cd DemoFreestyleProject1/
devopsadmin@ip-172-31-0-132:~/workspace/DemoFreestyleProject1$ ls
total 8
-rwxrwxr-x 2 devopsadmin devopsadmin 4096 Oct 5 20:27 .
-rwxrwxr-x 3 devopsadmin devopsadmin 4096 Oct 5 20:27 ..
devopsadmin@ip-172-31-0-132:~/workspace/DemoFreestyleProject1$ |

```

Red arrows highlight the command 'su - devopsadmin', the directory '/home/devopsadmin/' which is shown as 'Permission denied', and the directory 'DemoFreestyleProject1/' which contains files '.'

Fig. 1.36: SSH into the Jenkins slave node AWS EC2 instance and verify the workspace folder

2. L2 - Create CICD Pipeline to Clone and Build Java Maven Web Application

Ans.

Step 1: Prepare the Environment:

On the Jenkins slave node instance

1. Ensure Java and Maven are Installed:

- a. Jenkins requires Java, and since we are building a Maven-based project, Maven must be installed on the Jenkins master.

b. ````

```
sudo -i  
apt update  
apt install openjdk-17-jdk -y  
```
```

Verify Java installation:

```

```
java -version  
```
```

- c. Maven: Install Maven if not already installed:

```

```
sudo apt install maven -y  
```
```

Verify Maven installation:

```

```
mvn -version  
```
```

#### 2. Ensure Jenkins Has Git Installed: Jenkins uses Git to clone the repository from GitHub.

```
` sudo apt install git -y`
```

### Step 2: Set Up a GitHub Repository:

#### 1. Prepare the Repository:

- a. Ensure your Java Maven web application is hosted on GitHub (or any Git repository).

- b. The repository should contain a valid pom.xml file (Maven project descriptor file).
2. Copy the Repository URL:
  - a. Go to your GitHub repository and copy the HTTPS clone URL (e.g., <https://github.com/your-username/your-repo.git>).
3. Create a Pipeline Job in Jenkins:
  - a. Log in to Jenkins Dashboard:  
Access Jenkins at <http://<jenkins-master-public-ip>:8080>.
  - b. Create a New Pipeline Project:
    - i. Click on New Item.
    - ii. Enter a name for the project (e.g., Maven-WebApp-Build).
    - iii. Select Pipeline and click OK.
  - c. Configure the Pipeline Project:
    - i. General Section:
      1. Add a description (optional).
      2. Check the GitHub project option and enter the GitHub repository URL.
    - ii. Pipeline Section:
      1. Scroll down to the Pipeline section.
      2. Choose Pipeline script as the definition.

### Step 3: Write the Pipeline Script:

1. Basic Groovy Script for Cloning and Building Maven Project:
  - a. Clone the Git Repository: The pipeline first needs to pull the Java Maven web application from the GitHub repository.
  - b. Build the Project Using Maven: The pipeline will then run Maven commands to build the project.

2. Here is a basic pipeline script you can use to clone and build the Java Maven project:

```
```
pipeline {
    agent {label 'slave1'}
    stages {
        stage('Clone') {
```

```
steps {
    git 'https://github.com/your-repo/web-app.git'
}
}
stage('Build') {
    steps {
        sh 'mvn clean package'
    }
}
}
}
```

```

#### Step 4: Configure Maven in Jenkins:

1. Ensure Maven is Configured in Jenkins:
  - a. Go to Manage Jenkins > Global Tool Configuration.
  - b. Under Maven, click on Add Maven.
  - c. Set the Name (e.g., Maven) and ensure that the Install automatically checkbox is checked.
  - d. Save the configuration.

#### Step 5: Test the Pipeline:

1. Save the Pipeline:
  - a. Once the pipeline script is configured, click Save.
2. Build the Project:
  - a. On the pipeline project page, click Build Now.
3. Check the Console Output:
  - a. After the build starts, click on the build number in the Build History section.
  - b. Select Console Output to monitor the build progress.

## Step 6: Review the Build Results:

1. Console Output:
  - a. Check whether the pipeline successfully cloned the repository and ran Maven build commands.

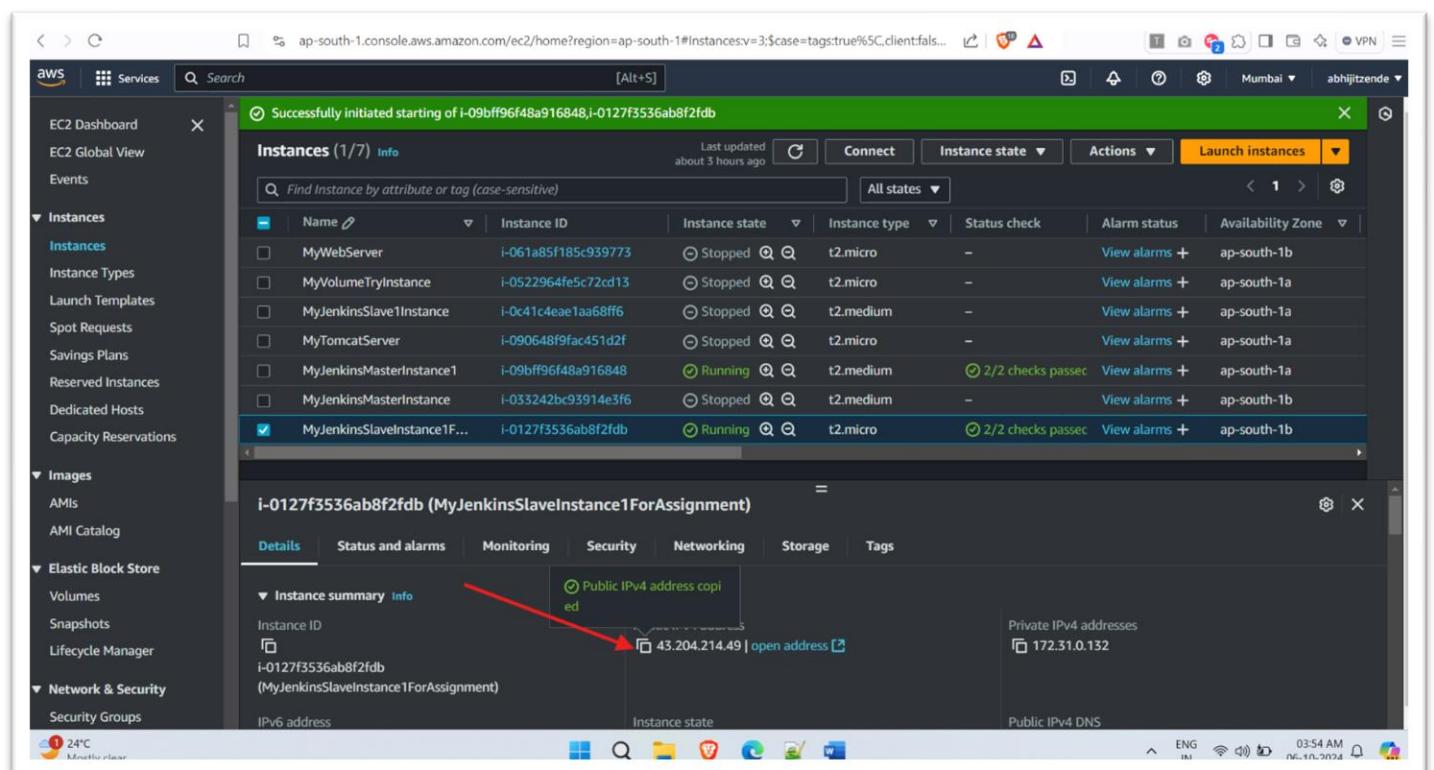


Fig. 2.01: Copy Jenkins slave node AWS EC2 instance IPv4

```

root@ip-172-31-0-132:~#
ELL@DESKTOP-RM3E0GT MINGW64 /e/Softwares/DevOps/StarAgile DevOPS/Key Pair AWS/FirstDemoInstance
ssh -i aws_instance_safe_key_pair.pem ubuntu@43.204.214.49
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1015-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sat Oct 5 22:26:46 UTC 2024

System load: 0.0 Processes: 114
Usage of /: 31.8% of 7.57GB Users logged in: 1
Memory usage: 31% IPv4 address for eth0: 172.31.0.132
Swap usage: 0%

expanded Security Maintenance for Applications is not enabled.
updates can be applied immediately.

Available ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

last login: Sat Oct 5 20:33:10 2024 from 152.58.32.37
root@ip-172-31-0-132:~# sudo -i
root@ip-172-31-0-132:~# apt update
[...]
root@ip-172-31-0-132:~# apt install openjdk-17-jdk -y
[...]
root@ip-172-31-0-132:~# apt install maven -y
[...]
root@ip-172-31-0-132:~# apt install git -y
[...]

```

*Fig. 2.02: SSH into the slave node ec2 instance and install maven*

```

root@ip-172-31-0-132:~#
unpacking maven (3.6.3-5) ...
 getting up libslf4j-api-java (1.7.32-1) ...
 getting up libplexus-utils2-java (3.0.1) ...
 getting up libplexus-classworlds-java (2.6.0-1) ...
 getting up libjsr305-java (0.1->svn49-1) ...
 getting up libjsr303-api-java (1.5.6) ...
 getting up libcommons-cli-java (1.4-2) ...
 getting up libplexus-component-annotations-java (2.1.0-1) ...
 getting up libplexus-cipher-java (1.8-2) ...
 getting up liberonimo-annotation-1.3-spec-java (1.3-1) ...
 getting up libapache-pom-java (18-1) ...
 getting up libatinject-jsr330-api-java (1.0+ds1-5) ...
 getting up libplexus-interpolation-java (1.26-1) ...
 getting up libwagon-provider-api-java (1.4-4) ...
 getting up libwagon-http-shaded-java (3.3.4-1) ...
 getting up libcdi-api-java (1.2-3) ...
 getting up libhawtjni-runtime-java (1.17-1) ...
 getting up libwagon-provider-api-java (3.3.4-1) ...
 getting up libmaven-parent-java (31-2) ...
 getting up libcommons-parent-java (43-1) ...
 getting up libsisu-inject-java (0.3.4-1) ...
 getting up libsisu-plexus-java (0.3.4-3) ...
 getting up libplexus-interpolation-java (1.4.2-3build1) ...
 getting up libquava-java (29.0-6) ...
 getting up libcommons-lang3-java (3.11-1) ...
 getting up libjansi-native-java (1.8-1) ...
 getting up libwagon-file-java (3.3.4-1) ...
 getting up libcommons-io-java (2.11.0-2) ...
 getting up libguice-java (4.2.3-2) ...
 getting up libjansi-java (1.18-1) ...
 getting up libmaven-project-classifier (3.3.0-1ubuntu0.22.04.1) ...
 getting up libmaven3-common-java (3.6.3-5) ...
 updating alternatives: using /usr/share/maven/bin/mvn to provide /usr/bin/mvn (mvn) in auto mode
 cleaning processes...
 cleaning candidates...
 cleaning linux images...

 running kernel seems to be up-to-date.

 no services need to be restarted.

 no containers need to be restarted.

 no user sessions are running outdated binaries.

5 VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-0-132:~# mvn -version
Apache Maven 3.6.3
 over home: /usr/share/maven
 maven version: 17.0.12, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
 default locale: en, platform encoding: UTF-8
 os name: "linux", version: "6.8.0-1015-aws", arch: "amd64", family: "unix"
root@ip-172-31-0-132:~# apt install git -y
[...]

```

*Fig. 2.03: Checking maven and git version*

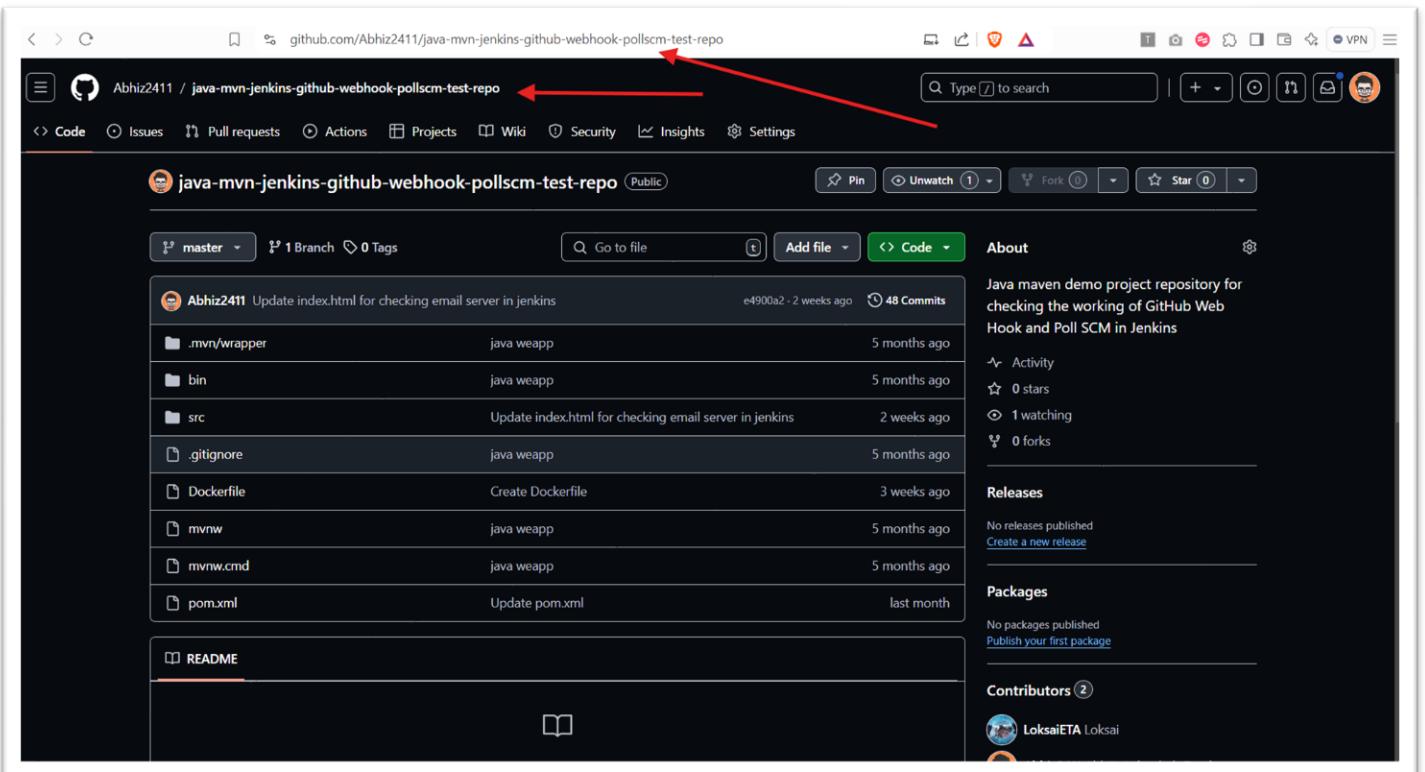


Fig. 2.04: Setting up GitHub repo for maven build

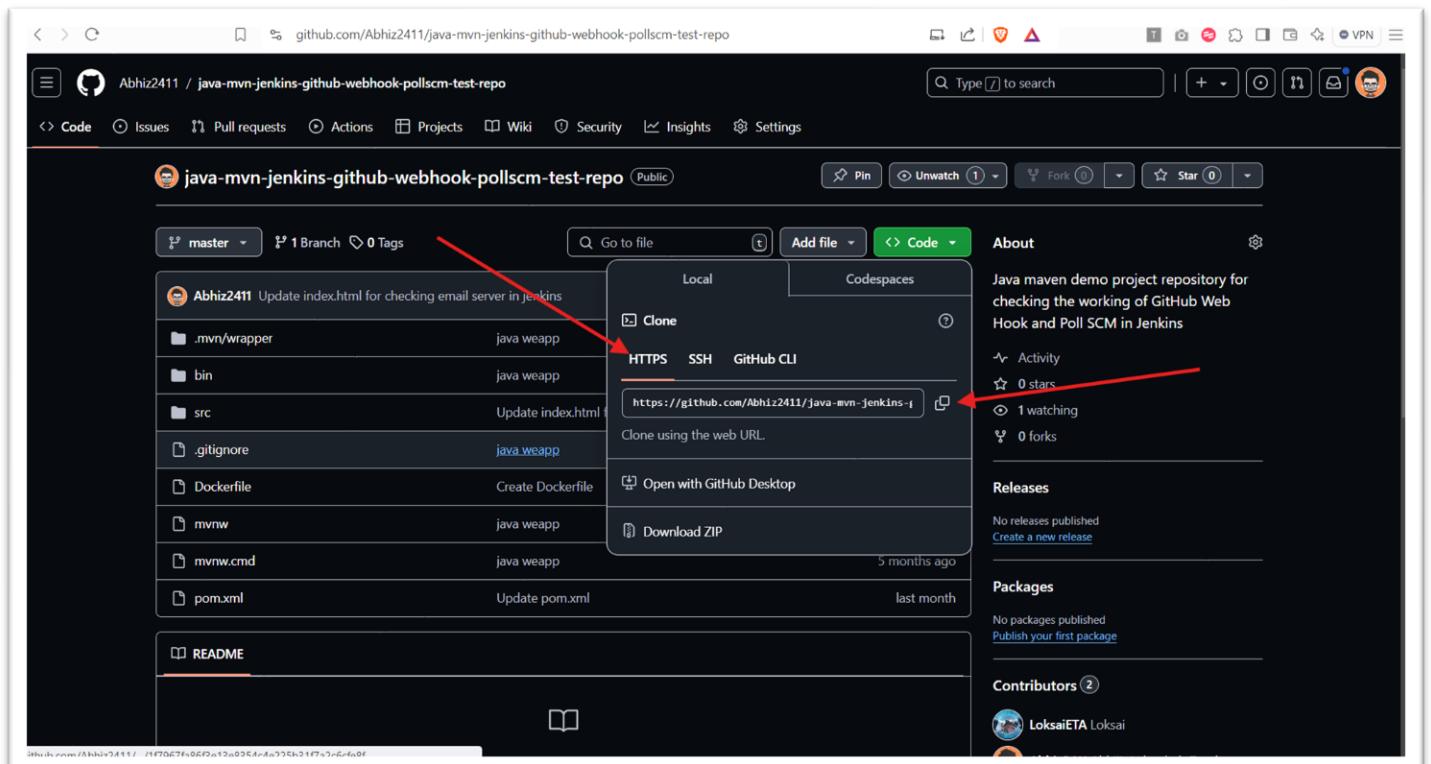


Fig. 2.05: Copying HTTPS address of the maven build GitHub repo

The screenshot shows the Jenkins dashboard. At the top right, there is a red arrow pointing to the '+ New Item' button. Below the header, there is a search bar and a user profile for 'Abhijit Lalasaheb Zende'. On the left, there are several links: 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. In the center, there is a table showing project statistics: 'DemoFreestyleProject1' has a status of 'S' (green checkmark), 'W' (yellow sun), 'Last Success' at '2 hr 14 min #1', 'Last Failure' at 'N/A', and 'Last Duration' at '0.51 sec'. Below the table, there are sections for 'Build Queue' (empty) and 'Build Executor Status' (listing 'Built-In Node' with 1 idle and 2 idle, and 'SlaveNode1' with 1 idle and 2 idle). A red arrow also points to the 'Add description' link at the top right.

Fig. 2.06: New item from Jenkins Dashboard

The screenshot shows the 'New Item' creation dialog. At the top, it says 'Enter an item name' with 'DemoPipelineProject1' typed in. Below that, it says 'Select an item type' and lists three options: 'Freestyle project' (selected), 'Pipeline', and 'Multi-configuration project'. Each option has a description below it. A red arrow points from the 'Freestyle project' section down to the 'OK' button at the bottom. Another red arrow points from the 'Pipeline' section up towards the 'Freestyle project' section.

Fig. 2.07: Pipeline project overview information

The screenshot shows the Jenkins configuration interface for a pipeline project. The top navigation bar includes links for Dashboard, DemoPipelineProject1, Configuration, and a user profile for Abhijit Lalasaheb Zende. The main title is 'Configure' and the sub-section is 'General'. A large red arrow points from the left towards the 'GitHub project' section. Another red arrow points from the bottom right towards the 'Project url' input field, which contains the URL <https://github.com/Abhiz2411/java-mvn-jenkins-github-webhook-pollscm-test-repo.git>. The 'Enabled' toggle switch is turned on. The 'Description' field contains the text 'A demo pipeline project to demonstrate maven build using Jenkins'. The 'Advanced' dropdown menu is visible at the bottom.

Fig. 2.08: Pipeline project General configuration

The screenshot shows the Jenkins configuration interface for a pipeline project, specifically under 'Advanced Project Options'. A large red arrow points from the left towards the 'Pipeline script' dropdown menu, which is set to 'Pipeline script'. Another red arrow points from the bottom right towards the 'Save' button. The 'Script' code block displays the following Groovy pipeline script:

```
1 * pipeline{
2 agent {label 'slave1'}
3
4 stages{
5 stage('SCM-checkout'){
6 steps{
7 echo 'SCM checkout stage start'
8 git 'https://github.com/Abhiz2411/java-mvn-jenkins-github-webhook-pollscm-test-repo.git'
9 echo 'SCM checkout complete'
10 }
11 }
12 stage('Maven_build'){
13 steps{
14 echo 'Maven build stage start'
15 sh 'mvn clean package'
16 echo 'Maven build complete'
17 }
18 }
19 }
20}
```

The 'Use Groovy Sandbox' checkbox is checked. The 'Pipeline Syntax' section is visible at the bottom.

Fig. 2.09: Advanced Project Options of pipeline project

The screenshot shows the Jenkins Manage Jenkins interface. On the left sidebar, 'Manage Jenkins' is highlighted with a red arrow. The main content area is titled 'Manage Jenkins' with a warning message: 'Building on the built-in node can be a security issue. You should set the number of executors on the built-in node to 0. See [the documentation](#).'. Below this, there are several configuration sections:

- System Configuration**: Includes 'System' (Configure global settings and paths), 'Nodes' (Add, remove, control and monitor nodes), 'Clouds' (Add, remove, and configure cloud instances), and 'Plugins' (Add, remove, disable or enable plugins).
- Security**: Includes 'Security' (Secure Jenkins) and 'Credentials' (Configure credentials).

On the right side of the main content area, there is a search bar labeled 'Search settings' and a user profile for 'Abhijit Lalasaheb Zende'.

Fig. 2.10: Manage Jenkins Tools

The screenshot shows the 'Configure Tools' page under 'Manage Jenkins > Tools'. A red arrow points from the previous 'Tools' section in Fig. 2.10 to this page. The page has a header with an 'Install automatically' checkbox and a dropdown menu for 'Add Git'. The main content is divided into sections for different tools:

- Gradle installations**: 'Add Gradle' button.
- Ant installations**: 'Add Ant' button.
- Maven installations**: 'Add Maven' button.

At the bottom are 'Save' and 'Apply' buttons. The footer of the page indicates 'Jenkins 2.462.3'.

Fig. 2.11: Add Maven

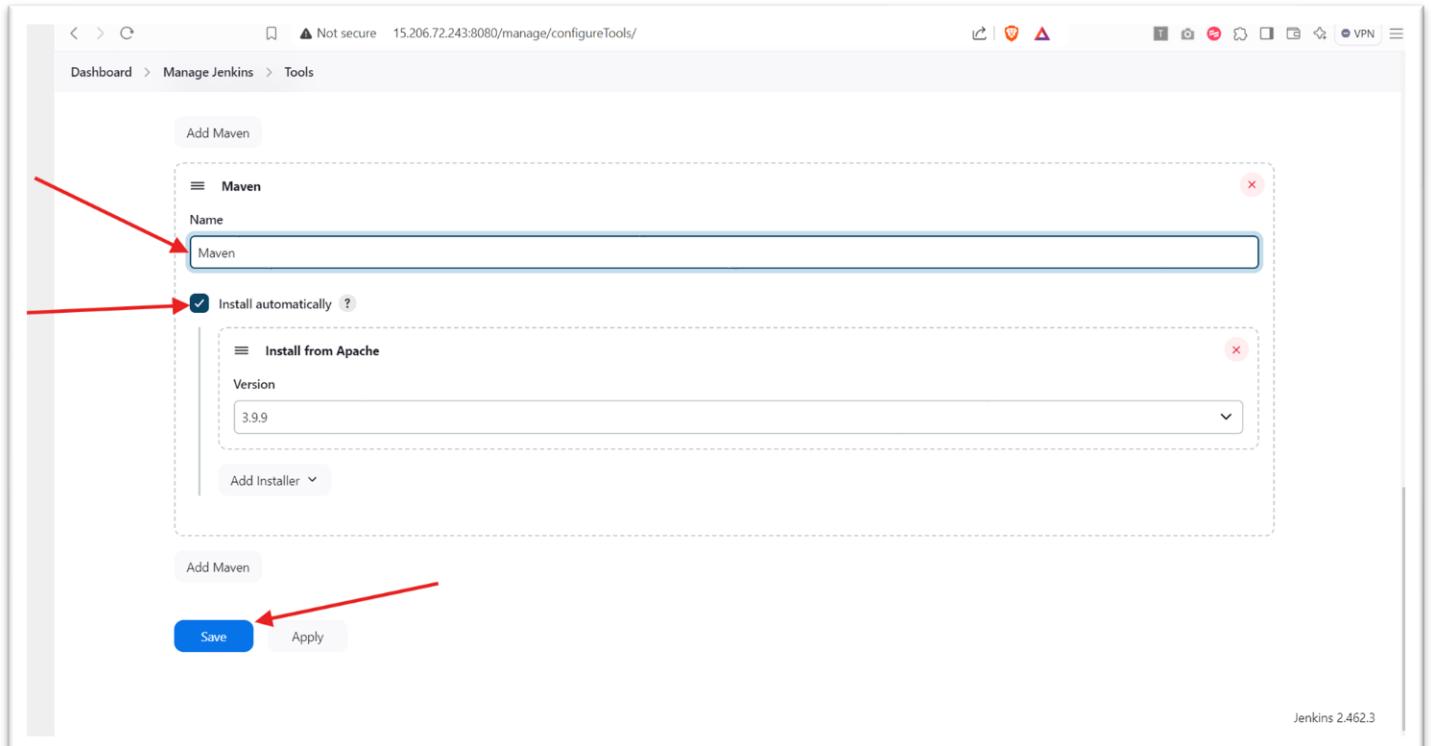


Fig. 2.12: Set Maven

The screenshot shows the Jenkins pipeline project 'DemoPipelineProject1' status page. The 'Build Now' button is highlighted with a red arrow. The page also displays the project name 'DemoPipelineProject1', a description 'A demo pipeline project to demonstrate maven build using Jenkins', and a 'Permalinks' section.

Fig. 2.13: Build now

The screenshot shows the Jenkins dashboard for the project 'DemoPipelineProject1'. On the left, a sidebar contains links like Status, Changes, Build Now, Configure, Delete Pipeline, GitHub, Stages, Rename, and Pipeline Syntax. The main area displays the title 'DemoPipelineProject1' and a brief description: 'A demo pipeline project to demonstrate maven build using Jenkins'. Below this is a section titled 'Permalinks'. A red arrow points to the 'Build History' section, which shows a single build entry for '#1' started on '5 Oct 2024, 23:04'. At the bottom right of the page are links for 'REST API' and 'Jenkins 2.462.3'.

Fig. 2.14: Build History

The screenshot shows the detailed status of Build #1 for the project 'DemoPipelineProject1'. The build was successful, indicated by a green checkmark icon. The title is 'Build #1 (5 Oct 2024, 23:04:23)'. On the left, a sidebar lists various options: Status, Changes, Console Output, Edit Build Information, Delete build '#1', Timings, Git Build Data, Pipeline Overview, Pipeline Console, Restart from Stage, Replay, Pipeline Steps, and Workspaces. To the right of the sidebar, there are sections for 'Started by user' (Abhijit Lalaheb Zende), 'This run spent:' (19 ms waiting, 32 sec build duration, 32 sec total from scheduled to completion), and 'git' details (Revision: e4900a255f0e3fd8240163504fc7a398499c7f1a, Repository: <https://github.com/Abhiz2411/java-mvn-jenkins-github-webhook-pollscm-test-repo.git>, refs/remotes/origin/master). At the top right, there are links for 'Keep this build forever', 'Add description', 'Started 3 min 24 sec ago', and 'Took 32 sec'. At the bottom right are links for 'REST API' and 'Jenkins 2.462.3'.

Fig. 2.15: First build status

The screenshot shows the AWS EC2 Instances page. A list of 7 instances is displayed, with 'MyJenkinsSlaveInstance1F' selected. The instance details page is open, showing the instance summary with the Public IPv4 address '43.204.214.49' highlighted.

Fig. 2.16: Copy IPv4 address of slave node EC2 instance for SSH access

```

$ ssh -i aws_instance_safe_key_pair.pem ubuntu@43.204.214.49
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1015-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sat Oct 5 23:12:00 UTC 2024

System load: 0.0 Processes: 111
Usage of /: 34.8% of 7.57GB Users logged in: 1
Memory usage: 33%
Swap usage: 0%

expanded Security Maintenance for Applications is not enabled.

updates can be applied immediately.

additional security updates can be applied with ESM Apps.
learn more about enabling ESM Apps service at https://ubuntu.com/esm

new release '24.04.1 LTS' available.
run 'do-release-upgrade' to upgrade to it.

last login: Sat Oct 5 22:26:47 2024 from 152.58.32.37
ubuntu@ip-172-31-0-132:~$ sudo -i
root@ip-172-31-0-132:~# su - devopsadmin
devopsadmin@ip-172-31-0-132:~$ ls /home/devopsadmin/
bin remoting remoting.jar workspace
devopsadmin@ip-172-31-0-132:~$ cd /home/devopsadmin/
devopsadmin@ip-172-31-0-132:~/workspace$ ls
devopsadmin@ip-172-31-0-132:~/workspace$ cd workspace/
devopsadmin@ip-172-31-0-132:~/workspace$ ls
DemoFreestyleProject1 DemoPipelineProject1@tmp
devopsadmin@ip-172-31-0-132:~/workspace$ cd DemoPipelineProject1
devopsadmin@ip-172-31-0-132:~/workspace/DemoPipelineProject1$ ls
pockerfile bin mvnw mvnw.cmd pom.xml src target
devopsadmin@ip-172-31-0-132:~/workspace/DemoPipelineProject1$ ls target/
classes demo-1.0-SNAPSHOT.war demo-1.0-SNAPSHOT.war.original generated-sources generated-test-sources maven-archiver maven-status surefire-reports test-classes
devopsadmin@ip-172-31-0-132:~/workspace/DemoPipelineProject1$

```

Fig. 2.17: SSH into slave node instance and verify the build

### **3. L3 - Create CICD Pipeline to Deploy the Maven Web Application in Tomcat Server**

Ans

#### Step 1: Launch a new instance for Tomcat server:

1. Launch a new AWS EC2 instance
2. Set appropriate Security group
  - a. The security group attached with the instance should have inbound rules allowing traffic to port 22 for SSH access and port 8080 which is Tomcat server port

#### Step 2: Install Tomcat server on the instance:

1. SSH into the tomcat server AWS EC2 instance:

SSH into the tomcat server AWS EC2 instance using private key pair and public IPv4 address of the instance using below command (In Git bash) or just use MobaXterm

```
'ssh -i private_key_pair username@public_ip_v_4_address_of_instance'
```

2. Update the package list

```
'''
```

```
sudo -i
apt update && apt upgrade
'''
```

3. Install the pre-requisite for Tomcat:

- a. Install java if not already installed. Preferably greater than java-11

```
'apt install openjdk-17-jdk -y'
```

4. Create dedicated Linux user for managing Tomcat server

```
'''
```

```
useradd devopsadmin -s /bin/bash -m -d /home/devopsadmin
su - devopsadmin
ssh-keygen -t ecdsa -b 521
ls ~/.ssh
cd .ssh
```

```
cat id_ecdsa.pub > authorized_keys
chmod 600 /home/devopsadmin/.ssh/*
```
```

5. Download and Install Tomcat:

Download the latest version of Tomcat (replace 9.0.XX with the latest version):

```
```
```

```
cd /opt/
```

```
wget https://archive.apache.org/dist/tomcat/tomcat-8/v8.5.99/bin/apache-tomcat-8.5.99.tar.gz
```

```
```
```

6. Extract Tomcat package:

```
```tar -xvzf /opt/apache-tomcat-8.5.99.tar.gz ```
```

## 7. Rename tomcat file:

```
`mv apache-tomcat-8.5.99 tomcat`
```

## 8. Start the Tomcat server:

```
```
```

```
cd tomcat/bin/
```

```
./startup.sh
```

```
```
```

## 9. Make the newly created user the owner of tomcat dir:

```
`chown -R devopsadmin /opt/tomcat`
```

## Step 3: Install ‘Publish over SSH’ plugin in Jenkins:

1. Navigate to Available plugins in Jenkins and search for ‘Publish over SSH’ plugin and install the plugin
2. Safe restart once plugin installed. Verify installation from installed plugins

#### Step 4: Setup Publish over SSH plugin:

1. Navigate to 'Dashboard > Manage Jenkins > System'. Scroll down to 'Publish over SSH' part and add ssh servers by clicking on 'Add' button below 'SSH servers'
  - a. Name: Any suitable name for reference for eg. Tomcat\_server\_1
  - b. Hostname: Private IP address of server instance
  - c. Username: Username with which authorized keys are made for eg. devopsadmin
  - d. Remote directory: Location where we want to save our artifacts for eg. /opt/tomcat/webapps
  - e. Advanced:
    - i. Tick the 'Use password authentication, or use a different key'
    - ii. Enter the private key for the user from the server instance
  - f. Test configuration. If success then proceed
2. Save the settings

#### Step 5: Create/Modify existing pipeline project for deployment stage:

1. Write the groovy script using the 'pipeline syntax' with 'sshPublisher'
  - a. Name: Select the name of server for eg. Tomcat\_servver
  - b. Source files: Path to artifact for eg. target/\*.war
  - c. Remove prefix: Everything in source files path except artifact for eg. target/
  - d. Remote diorectory: Since remote directory already set in Jenkis global settings of this server the path will be same so '.'
  - e. Generate pipeline script
2. Paste the generated code using script tag under steps in Deployment stage  
For e.g.  
```

```
stage('Deployment_stage') {
    steps {
        script {
            sshPublisher(publishers: [sshPublisherDesc(configName: 'Tomcat_server_1', transfers: [sshTransfer(cleanRemote: false, excludes: ''))]
```

```

    execCommand: "", execTimeout: 120000, flatten: false, makeEmptyDirs: false, noDefaultExcludes: false, patternSeparator: '[, ]+', remoteDirectory: '.', remoteDirectorySDF: false, removePrefix: 'target/', sourceFiles: 'target/*.war')], usePromotionTimestamp: false, useWorkspaceInPromotion: false, verbose: false)])
}
}
```

```

### 3. Apply and Save

#### Step 6: Build and verify:

1. Click on Build in Jenkins

2. Verify the build:

- a. You can verify based on status of the build
- b. We can also visit the tomcat server link ‘public IPv4 address of tomcat server:8080/name-of-war-file/’

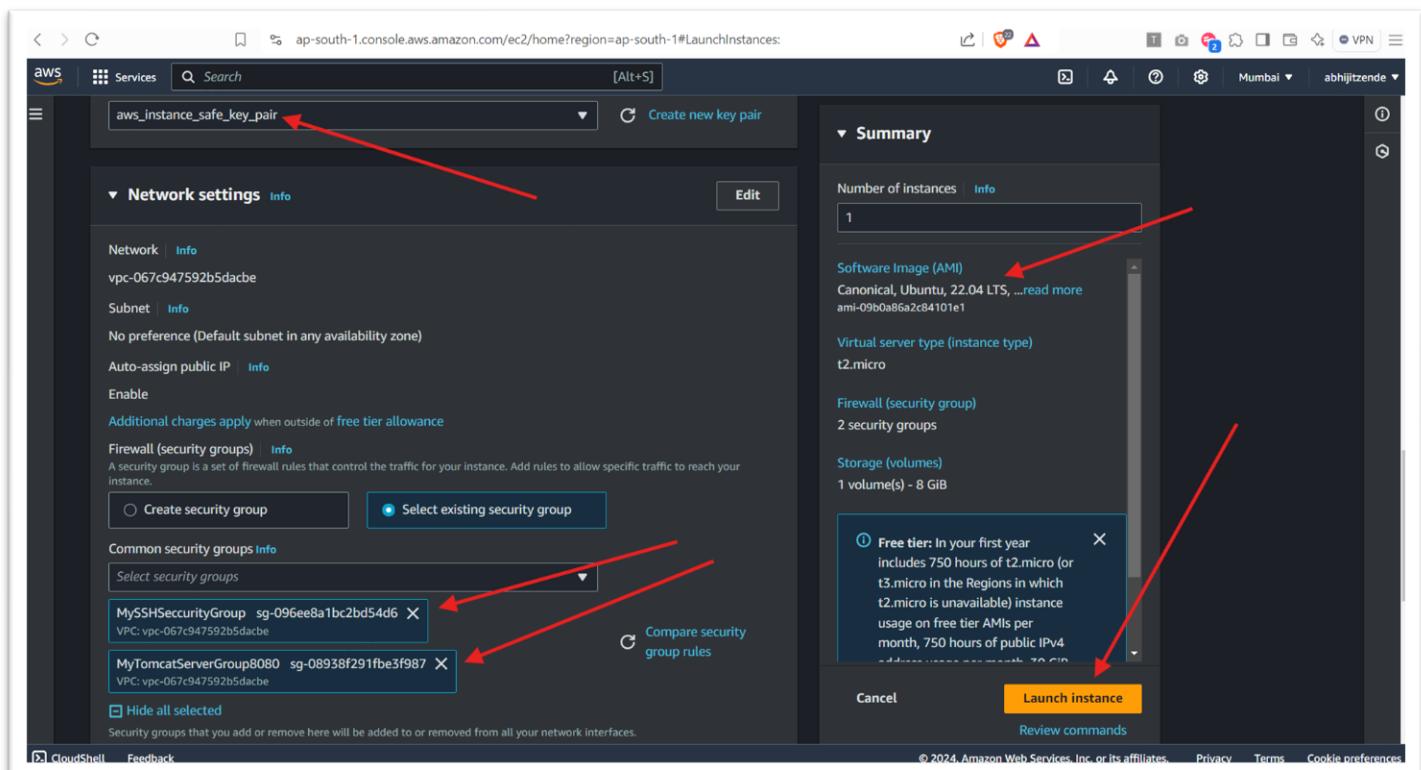


Fig. 3.01: Launch new Tomcat server AWS EC2 instance

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar lists various services: EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, and CloudShell. The main area displays 'Instances (1/1) Info' with a single instance listed: Instance ID = i-0952a236bee048f66, Name = MyTomcatServerForAssignment, Instance state = Running, Instance type = t2.micro, Status check = Initializing, Alarm status = View alarms +, Availability Z = ap-south-1b. Below this is a detailed view for instance i-0952a236bee048f66 (MyTomcatServerForAssignment). The 'Details' tab is selected, showing the Instance summary: Instance ID (i-0952a236bee048f66), Public IPv4 address (52.66.241.164), Instance state (Running). Other tabs include Status and alarms, Monitoring, Security, Networking, Storage, and Tags. A red arrow points to the instance name 'MyTomcatServerForAssignment'. Another red arrow points to the Public IPv4 address '52.66.241.164'.

Fig. 3.02: Copy IPv4 address of new Tomcat server AWS EC2 instance

```
MININOW04:e/Softwares/DevOps/StarAgile DevOPS/Key Pair AWS/FirstDemoInstance
LL0DESKTOP-RM3E0GT MINGW64 /e/Softwares/DevOps/StarAgile DevOPS/Key Pair AWS/FirstDemoInstance
ssh -i aws_instance_safe_key_pair.pem ubuntu@52.66.241.164|
```

A screenshot of a Git Bash terminal window. The command 'ssh -i aws\_instance\_safe\_key\_pair.pem ubuntu@52.66.241.164' is being typed. A red arrow points to the IP address '52.66.241.164'.

Fig. 3.03: SSH into Tomcat server AWS EC2 instance using Git Bash

*Fig. 3.04: Update and Upgrade task*

*Fif. 3.05: Install JDK*

```

❯ devopsadmin@ip-172-31-3-176:~/.ssh
processing triggers for ca-certificates (20240203~22.04.1) ...
Updating certificates in /etc/ssl/certs...
added, 0 removed; done.
running hooks in /etc/ca-certificates/update.d...
none.
none.
processing triggers for libglib2.0-0:amd64 (2.72.4-0ubuntu2.3) ...
processing triggers for libgb-bin (2.35-0ubuntu3.8) ...
processing triggers for man-db (2.10.2-1) ...
letting up at-sp12-core (2.44.0-3) ...
processing triggers for libgd-pixbuf=2.0-0:amd64 (2.42.8+dfsg-1ubuntu0.3) ...
canning processes...
canning linux images...

unning kernel seems to be up-to-date.
o services need to be restarted.
o containers need to be restarted.
o user sessions are running outdated binaries.

o VM guests are running outdated hypervisor (emu) binaries on this host.
pot@ip-172-31-3-176:~# su - devopsadmin
devopsadmin@ip-172-31-3-176:~$ ssh-keygen -t ecdsa -b 521
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/devopsadmin/.ssh/id_ecdsa):
Created directory '/home/devopsadmin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/devopsadmin/.ssh/id_ecdsa.
Your public key has been saved in /home/devopsadmin/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:xc5FqJzavzKNN3tB6G1/ImgIN1QG1CD3FGWI/hpyQ devopsadmin@ip-172-31-3-176
The key's randomart image is:
---[ECDSA 521]---
o=BB
|Eo*.
|+..o
|o = o
|S =+
| .++. o o
| .+= B.
| . .o=O
| . o...o
|
----[SHA256]-----
devopsadmin@ip-172-31-3-176:~$ ls ~/.ssh
id_ecdsa id_ecdsa.pub
devopsadmin@ip-172-31-3-176:~$ cd .ssh
devopsadmin@ip-172-31-3-176:~/ssh$ cat id_ecdsa.pub > authorized_keys
devopsadmin@ip-172-31-3-176:~/ssh$ chmod 600 /home/devopsadmin/.ssh/*
devopsadmin@ip-172-31-3-176:~/ssh$ ll /home/devopsadmin/.ssh
total 20
-rw-r--r-- 2 devopsadmin devopsadmin 4096 Oct 6 14:08 /
-rw-r-x--- 3 devopsadmin devopsadmin 4096 Oct 6 14:08 ..
-rw----- 1 devopsadmin devopsadmin 281 Oct 6 14:09 authorized_keys
-rw----- 1 devopsadmin devopsadmin 756 Oct 6 14:08 id_ecdsa
-rw----- 1 devopsadmin devopsadmin 281 Oct 6 14:08 id_ecdsa.pub
devopsadmin@ip-172-31-3-176:~/ssh$
```

Fig. 3.06: Create devopsadmin linux user

```

❯ root@ip-172-31-3-176:~$ sudo -i
root@ip-172-31-3-176:~# cd /opt/
root@ip-172-31-3-176:/opt# wget https://archive.apache.org/dist/tomcat/tomcat-8/v8.5.99/bin/apache-tomcat-8.5.99.tar.gz
2024-10-06 15:12:01-- https://archive.apache.org/dist/tomcat/tomcat-8/v8.5.99/bin/apache-tomcat-8.5.99.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1:a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
Length: 10875957 (10M) [application/x-gzip]
Saving to: 'apache-tomcat-8.5.99.tar.gz'

100%[=====] 10.37M 219KB/s in 48s
2024-10-06 15:12:49 (222 KB/s) - 'apache-tomcat-8.5.99.tar.gz' saved [10875957/10875957]

root@ip-172-31-3-176:/opt# tar -xvf /opt/apache-tomcat-8.5.99.tar.gz
apache-tomcat-8.5.99/
apache-tomcat-8.5.99/conf/catalina.policy
apache-tomcat-8.5.99/conf/catalina.properties
apache-tomcat-8.5.99/conf/jaspic-providers.xml
apache-tomcat-8.5.99/conf/jaspic-providers.xsd
apache-tomcat-8.5.99/conf/logging.properties
apache-tomcat-8.5.99/conf/server.xml
apache-tomcat-8.5.99/conf/tomcat-users.xml
apache-tomcat-8.5.99/conf/tomcat-users.xsd
apache-tomcat-8.5.99/bin/
apache-tomcat-8.5.99/lib/
apache-tomcat-8.5.99/logs/
apache-tomcat-8.5.99/temp/
apache-tomcat-8.5.99/webapps/
apache-tomcat-8.5.99/webapps/ROOT/
apache-tomcat-8.5.99/webapps/ROOT/META-INF/
apache-tomcat-8.5.99/webapps/ROOT/docs/
apache-tomcat-8.5.99/webapps/docs/META-INF/
apache-tomcat-8.5.99/webapps/docs/WEB-INF/
apache-tomcat-8.5.99/webapps/docs/WEB-INF/isp/
apache-tomcat-8.5.99/webapps/docs/annotationapi/
apache-tomcat-8.5.99/webapps/docs/api/
apache-tomcat-8.5.99/webapps/docs/appdev/
apache-tomcat-8.5.99/webapps/docs/appdev/sample/
apache-tomcat-8.5.99/webapps/docs/appdev/sample/docs/
apache-tomcat-8.5.99/webapps/docs/appdev/sample/src/
apache-tomcat-8.5.99/webapps/docs/appdev/sample/src/mypackage/
apache-tomcat-8.5.99/webapps/docs/appdev/sample/web/
apache-tomcat-8.5.99/webapps/docs/appdev/sample/web/WEB-INF/
apache-tomcat-8.5.99/webapps/docs/appdev/sample/web/images/
apache-tomcat-8.5.99/webapps/docs/architecture/
apache-tomcat-8.5.99/webapps/docs/architecture/requestProcess/
apache-tomcat-8.5.99/webapps/docs/architecture/startup/
apache-tomcat-8.5.99/webapps/docs/config/
apache-tomcat-8.5.99/webapps/docs/elapi/
apache-tomcat-8.5.99/webapps/docs/images/
apache-tomcat-8.5.99/webapps/docs/images/fonts/
apache-tomcat-8.5.99/webapps/docs/jaspicapi/
apache-tomcat-8.5.99/webapps/docs/jspapi/
apache-tomcat-8.5.99/webapps/docs/servletapi/
apache-tomcat-8.5.99/webapps/docs/tribes/
apache-tomcat-8.5.99/webapps/docs/websocketapi/
apache-tomcat-8.5.99/webapps/examples/
apache-tomcat-8.5.99/webapps/examples/META-INF/
```

Fig. 3.07: Download and unzip tomcat tar gz zip file

```

root@ip-172-31-3-176:~/opt/tomcat-8.5.99#
ls -la /opt/tomcat-8.5.99/webapps/
total 128
drwxr-xr-x 2 root root 4096 Mar 12 10:00 .
drwxr-xr-x 1 root root 4096 Mar 12 10:00 ..
drwxr-xr-x 1 root root 4096 Mar 12 10:00 examples
drwxr-xr-x 1 root root 4096 Mar 12 10:00 manager
drwxr-xr-x 1 root root 4096 Mar 12 10:00 host-manager
drwxr-xr-x 1 root root 4096 Mar 12 10:00 status
drwxr-xr-x 1 root root 4096 Mar 12 10:00 docs
drwxr-xr-x 1 root root 4096 Mar 12 10:00 samples
drwxr-xr-x 1 root root 4096 Mar 12 10:00 test
drwxr-xr-x 1 root root 4096 Mar 12 10:00 ROOT
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/host-manager/META-INF/context.xml
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/host-manager/WEB-INF/jsp/401.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/host-manager/WEB-INF/jsp/403.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/host-manager/WEB-INF/jsp/404.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/host-manager/WEB-INF/manager.xml
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/host-manager/WEB-INF/web.xml
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/host-manager/css/manager.css
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/host-manager/css/asf-logo.svg
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/host-manager/images/catalina.svg
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/host-manager/index.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/META-INF/context.xml
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/WEB-INF/jsp/401.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/WEB-INF/jsp/403.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/WEB-INF/jsp/404.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/META-INF/jsp/connectorsCerts.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/META-INF/jsp/connectorCiphers.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/META-INF/jsp/trustedCerts.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/WEB-INF/jsp/sessionDetail.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/WEB-INF/jsp/sessionsList.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/WEB-INF/web.xml
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/css/manager.css
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/images/asf-logo.svg
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/index.jsp
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/status.xsd
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/webapps/manager/xform.xsl
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/bin/Catalina.sh
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/bin/ciphers.sh
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/bin/configtest.sh
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/bin/daemon.sh
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/bin/digests.sh
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/bin/setclasspath.sh
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/bin/startup.sh
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/bin/tool-wrapper.sh
drwxr-xr-x 1 root root 4096 Mar 12 10:00 /opt/tomcat-8.5.99/bin/version.sh
root@ip-172-31-3-176:~/opt/tomcat/bin# ./startup.sh
Using CATALINA_BASE: /opt/tomcat
Using CATALINA_HOME: /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using CATALINA_PID: /usr
Using CLASSPATH: /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
root@ip-172-31-3-176:~/opt/tomcat/bin#

```

Fig. 3.08: Start Tomcat server

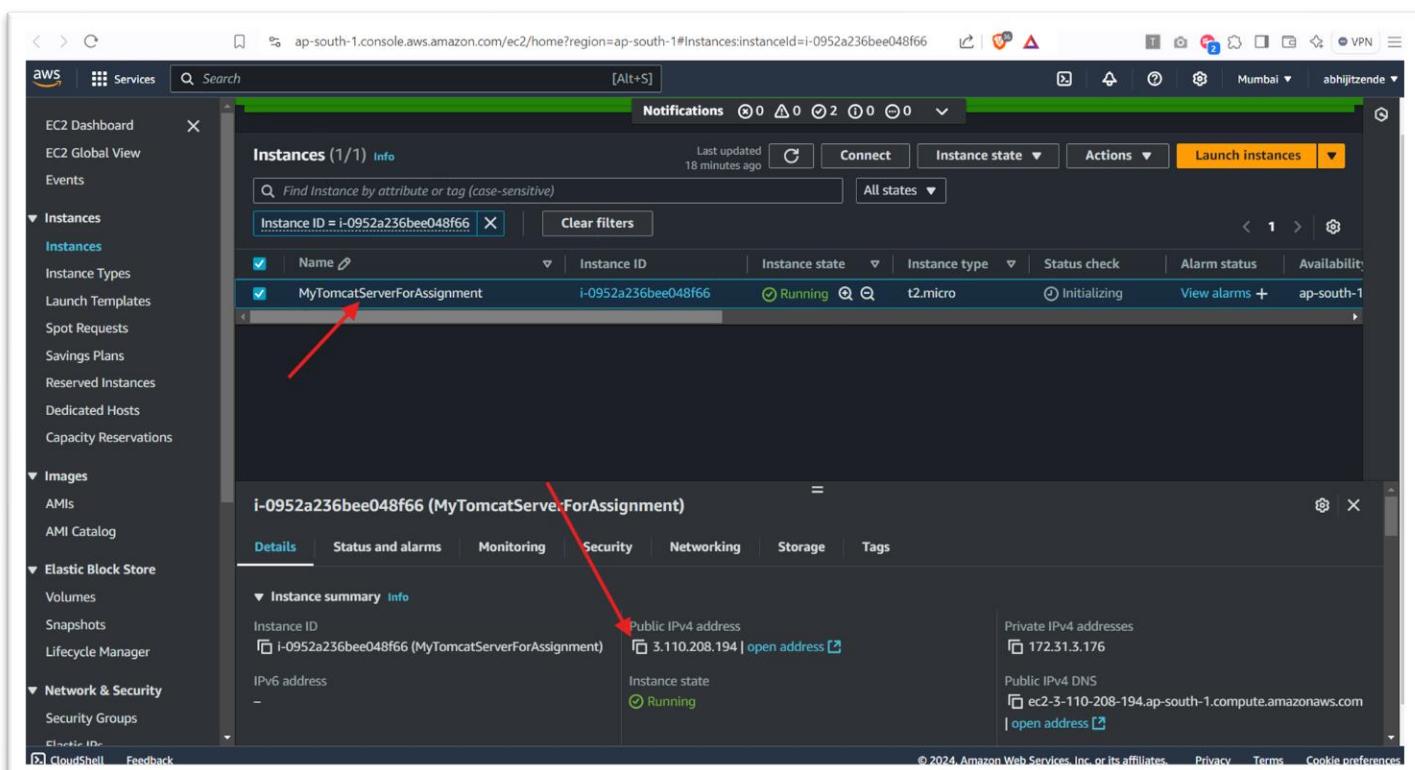


Fig. 3.09: Copy public IPv4 address of Tomcat server AWS EC2 instance

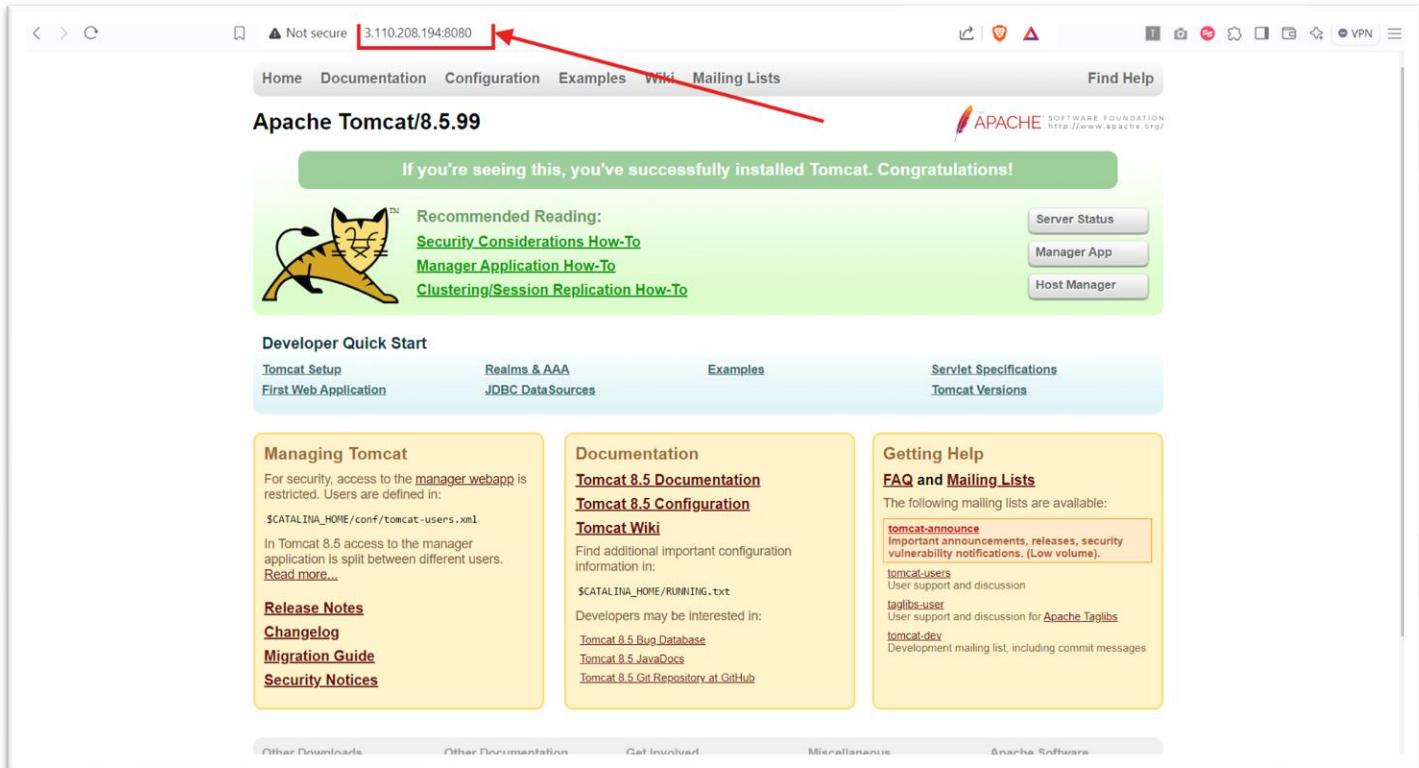


Fig. 3.10: Browser view of Tomcat server

```
root@ip-172-31-3-176:~$ sudo -i
root@ip-172-31-3-176:~# chown -R devopsadmin /opt/tomcat/
root@ip-172-31-3-176:~# ll /opt/tomcat/
total 160
drwxr-xr-x 9 devopsadmin root 4096 Oct 6 15:13 .
drwxr-xr-x 3 root root 4096 Oct 6 15:14 ..
drw-r---- 1 devopsadmin root 20913 Feb 14 2024 BUILDING.txt
drw-r---- 1 devopsadmin root 6210 Feb 14 2024 CONTRIBUTING.md
drw-r---- 1 devopsadmin root 57011 Feb 14 2024 LICENSE
drw-r---- 1 devopsadmin root 1726 Feb 14 2024 NOTICE
drw-r---- 1 devopsadmin root 3139 Feb 14 2024 RELEASE-NOTES
drw-r---- 1 devopsadmin root 16505 Feb 14 2024 RUNNING.txt
drwxr-x--- 2 devopsadmin root 4096 Oct 6 15:13 bin/
drwxr-x--- 3 devopsadmin root 4096 Oct 6 15:15 conf/
drwxr-x--- 2 devopsadmin root 4096 Oct 6 15:13 lib/
drwxr-x--- 2 devopsadmin root 4096 Oct 6 15:15 logs/
drwxr-x--- 2 devopsadmin root 4096 Oct 6 15:13 temp/
drwxr-x--- 7 devopsadmin root 4096 Feb 14 2024 webapps/
drwxr-x--- 3 devopsadmin root 4096 Oct 6 15:15 work/
```

Fig. 3.11: Changing ownership to devopsadmin user

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links like 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins' (which is selected and highlighted in blue), 'My Views', 'Build Queue' (empty), 'Build Executor Status' (showing 'Built-In Node' with 1 idle and 'SlaveNode1' with 2 idle), and 'Advanced settings'. The main area is titled 'Manage Jenkins' with a sub-section 'System Configuration'. It includes links for 'System', 'Tools', 'Nodes', 'Clouds', and 'Appearance'. A prominent red arrow points from the top right towards the 'Plugins' section. The 'Plugins' section has a sub-section 'Available plugins' with a list containing 'Publish Over SSH 1.25' (selected with a checked checkbox), 'Artifact Uploaders', and 'Build Tools'. There's also a note about security: 'Building on the built-in node can be a security issue. You should set the number of executors on the built-in node to 0. See [the documentation](#)'. At the bottom right of the page are 'Manage' and 'Dismiss' buttons.

Fig. 3.12: Manage plugin Jenkins

The screenshot shows the Jenkins Plugin Manager page for the 'Available plugins' section. A red arrow points from the top left towards the 'Available plugins' link in the sidebar. Another red arrow points from the bottom left towards the 'Install' button for the 'Publish Over SSH' plugin. The plugin details shown are: Name: 'Publish Over SSH 1.25', Version: '1.25', Released: '1 yr 3 mo ago', Category: 'Send build artifacts over SSH', Sub-categories: 'Artifact Uploaders' and 'Build Tools'. The 'Install' button is highlighted with a red arrow. The bottom right corner of the page shows 'REST API' and 'Jenkins 2.46.3'.

Fig. 3.13: Install Publish Over SSH plugin

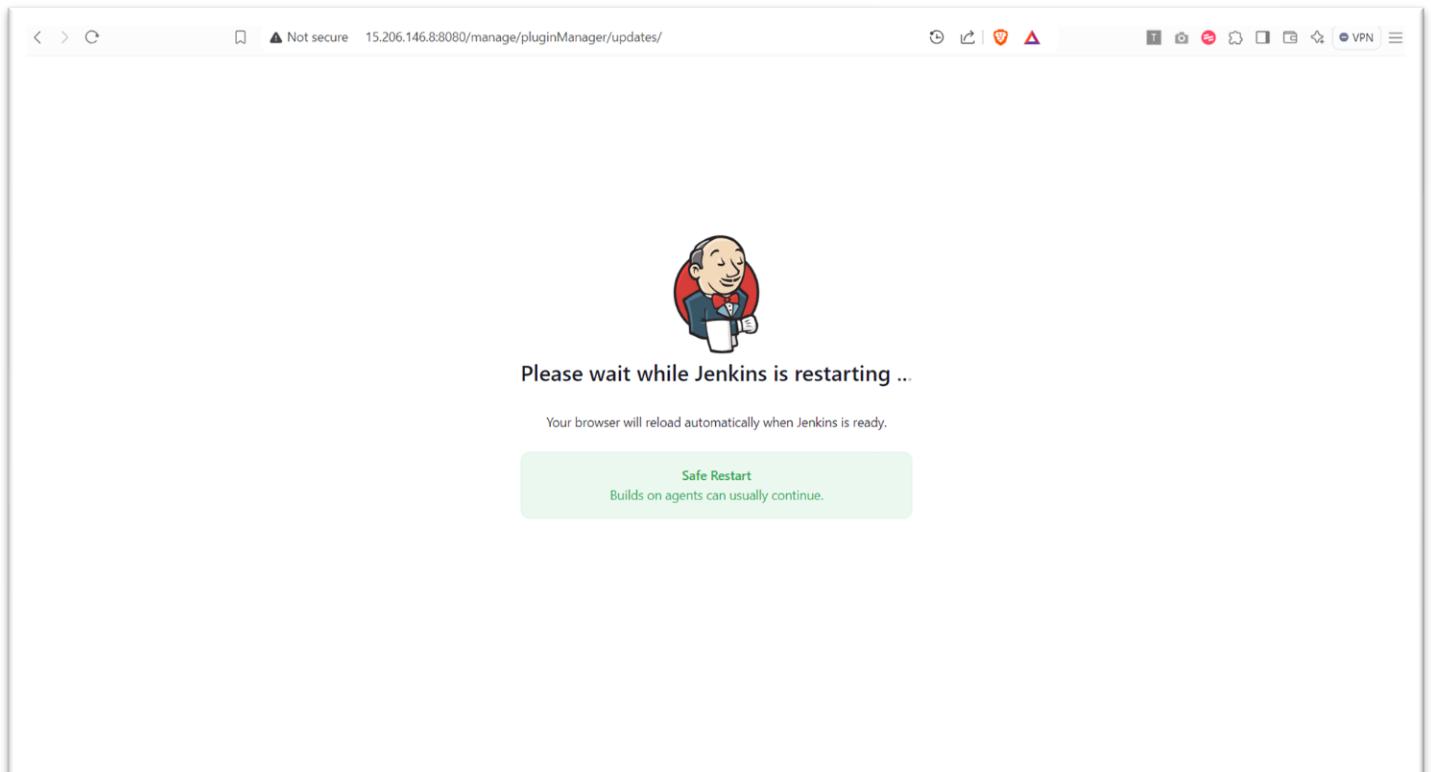


Fig. 3.14: Safe restart Jenkins

A screenshot of the Jenkins plugin manager interface. The URL is 15.206.146.8:8080/manage/pluginManager/installed. The page title is "Jenkins". The left sidebar shows "Plugins" selected, with sub-options: "Updates", "Available plugins", "Installed plugins" (which is highlighted with a red arrow), and "Advanced settings". The main content area shows a search bar with "Publish Over SSH" typed in, indicated by another red arrow. Below the search bar is a table with one row for "Publish Over SSH 1.25". The table columns are "Name" (with a dropdown arrow), "Description" (which reads "Send build artifacts over SSH" and "Report an issue with this plugin"), and "Enabled" (which has a checked checkbox and a red arrow pointing to it). At the bottom right of the page, there are links for "REST API" and "Jenkins 2.462.3".

Fig. 3.15: Verify in the installed plugins

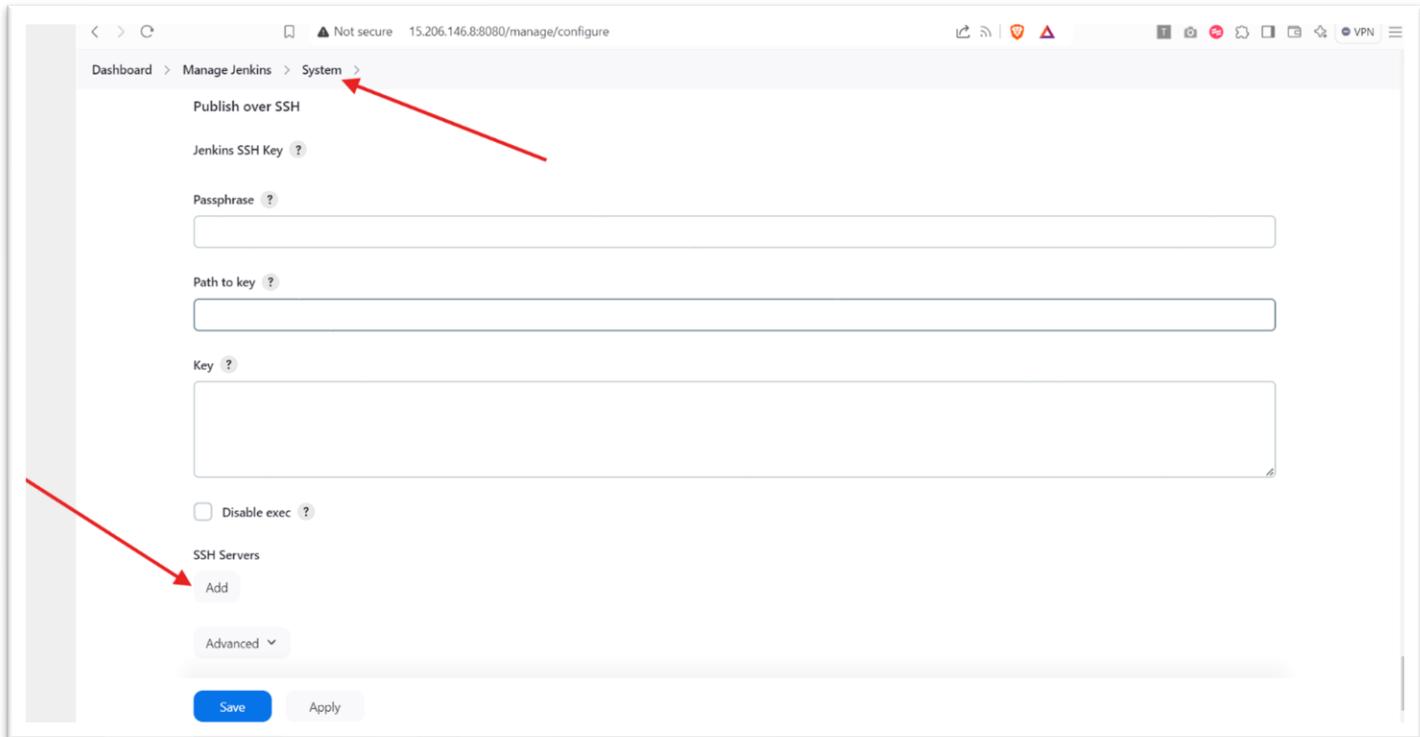


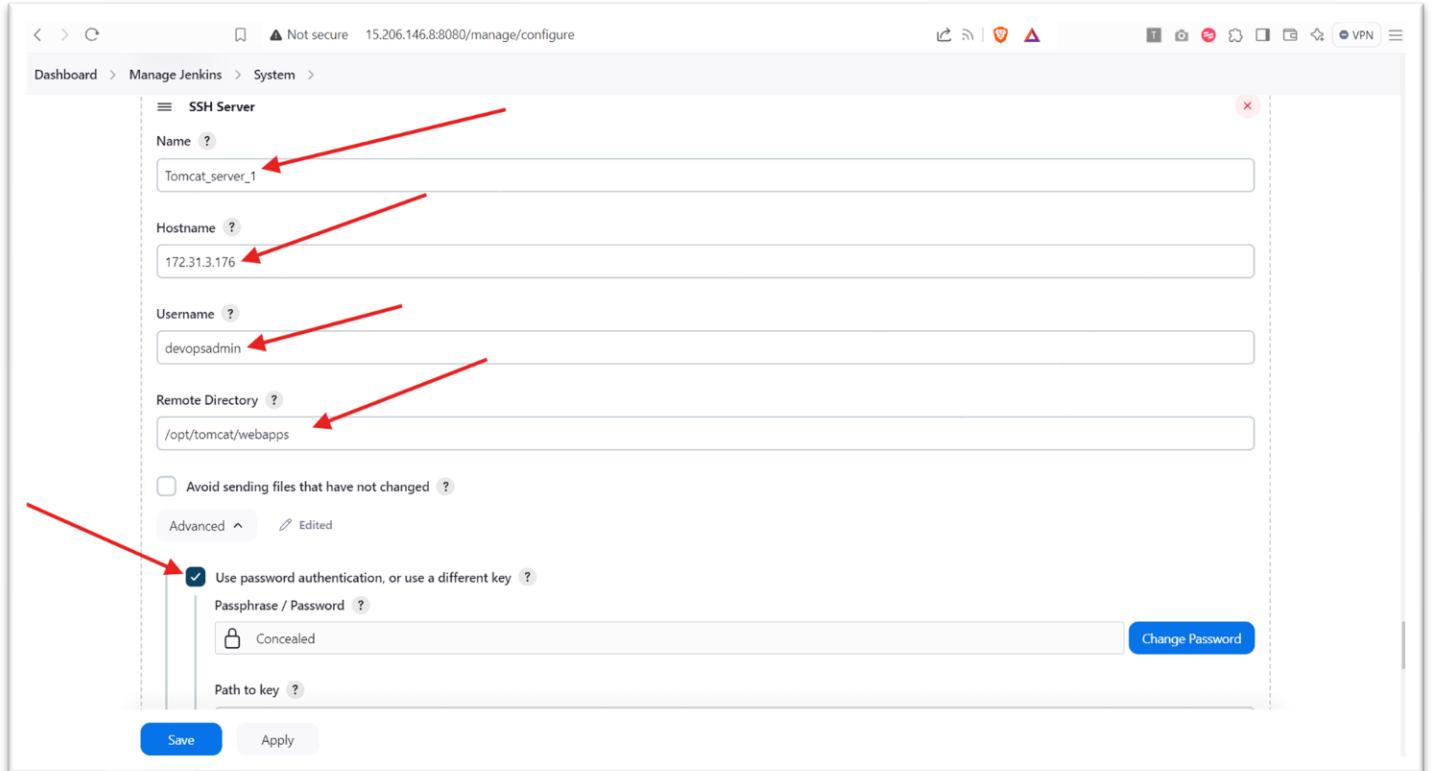
Fig. 3.16: Add SSH server

```

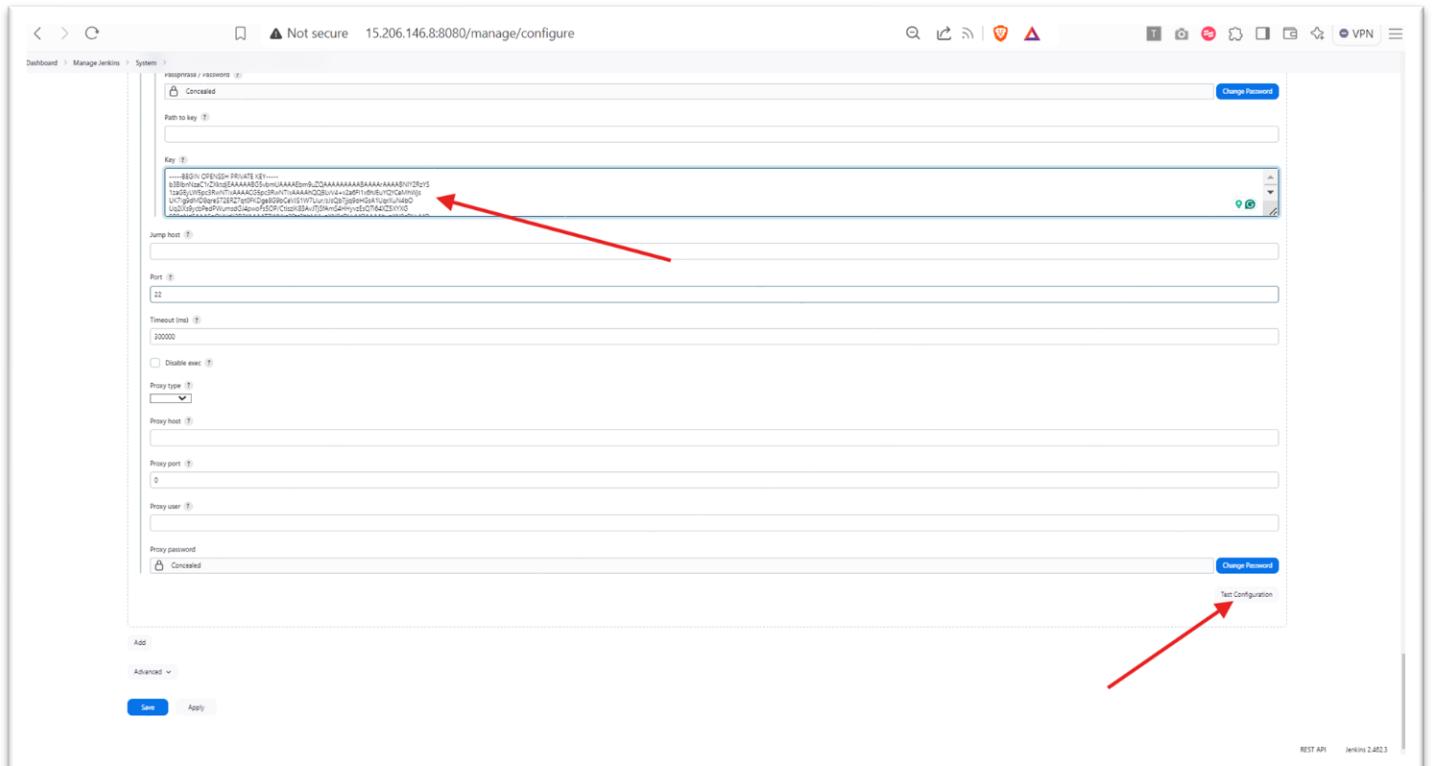
[root@ubuntuip-172-31-3-176 ~] sudo -i
[devopsadmin@ubuntuip-172-31-3-176 ~]# su - devopsadmin
[devopsadmin@ubuntuip-172-31-3-176 ~]$ cat .ssh/id_ecdsa
-----BEGIN OPENSSH PRIVATE KEY-----
MIIBdQIBAAQDwvbmUAAAEDmbuZQAAAAAAAABAAAArAAAABN1Y2RZYS
zaGEylW5pc3RwNTIxAAAACG5pc3RwNTIxAAAABAAQBLvV4+xa2a6F1x6hJEuYYCeMhwjs
C7Ig9MD08qes72ERZ7qT0FKDge8G9CeV3S1w7L1ur/S3Sob7jjq9OHGsA1uqrXuN4bo
jP8gHdAAAEGQULY/dij2P3YAAAATZwnKc2Ec2hhm1luoXw0DUw9AAAhuaxNvCDlyM0
kAIUEAS71ePsdmuhZdcceSRlnEGanj1Vo7FcuyIPXTA/k93ku9hewe6rdbSp4Hv8vwml
jtVuyyLq/7CbEG446vaBxrANVKq173egz1Kto17PcnGz3nt1prnRIekckB8OTj/wrYr
tivNw.ye40nwJkUbx8r8xLE04uut2ev2FxvlfDXRAAAQgfl.07EMlwplwR8Jcx277Y3
fFlm8c1d1utot52zMTXP1fd6G61q1hGmDz1hec32nPE3fjhvhbu8mR3uS+hq47DAAAAB
ZXZvcHNhzG1pbkbpC0NzITmZEMy0xNyBAgMEBQYH
-----END OPENSSH PRIVATE KEY-----
[devopsadmin@ubuntuip-172-31-3-176 ~]

```

Fig. 3.17: Copy Private key of Tomcat server AWS EC2 instance devopsadmin user



*Fig. 3.18: SSH server setting up*



*Fig. 3.19: Paste private key and test configuration*

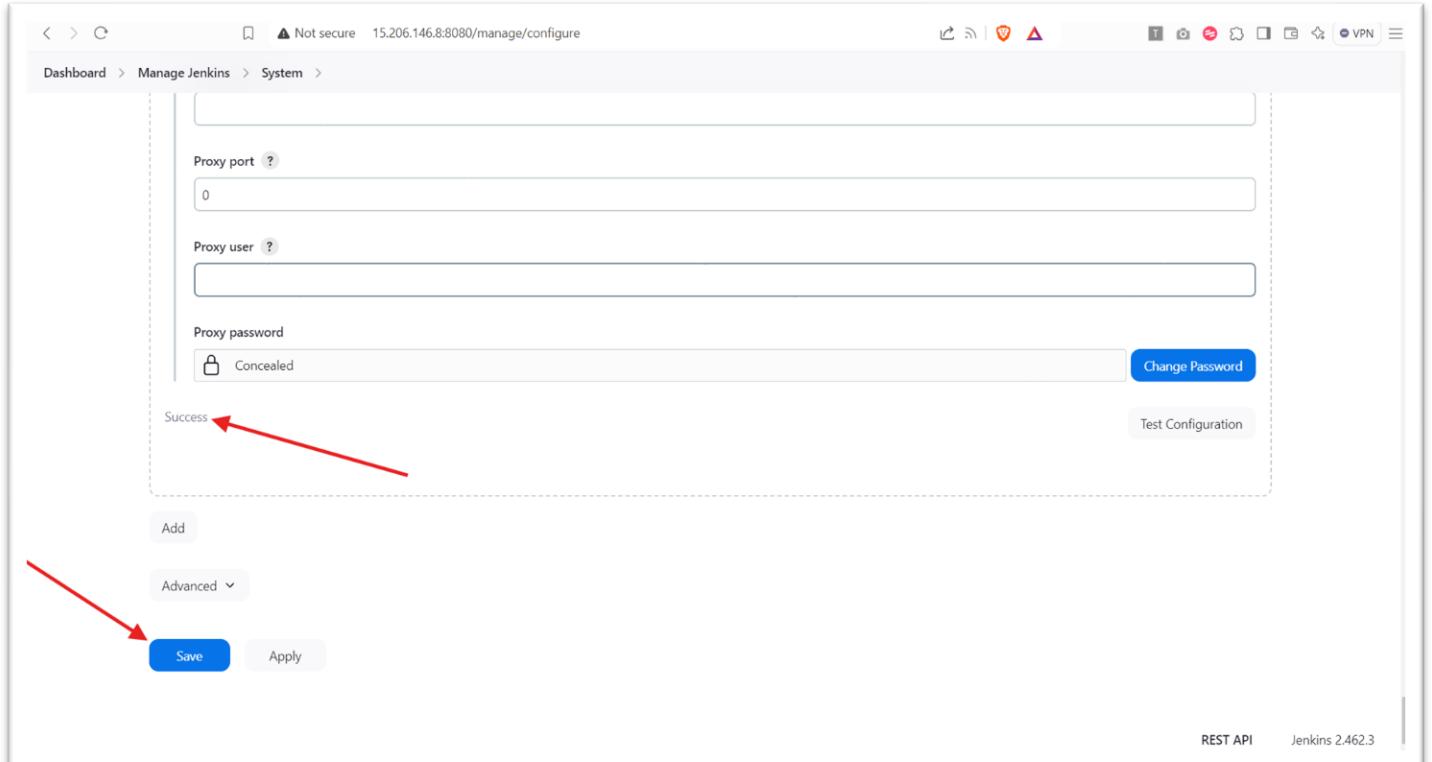


Fig. 3.20: Test configuration success

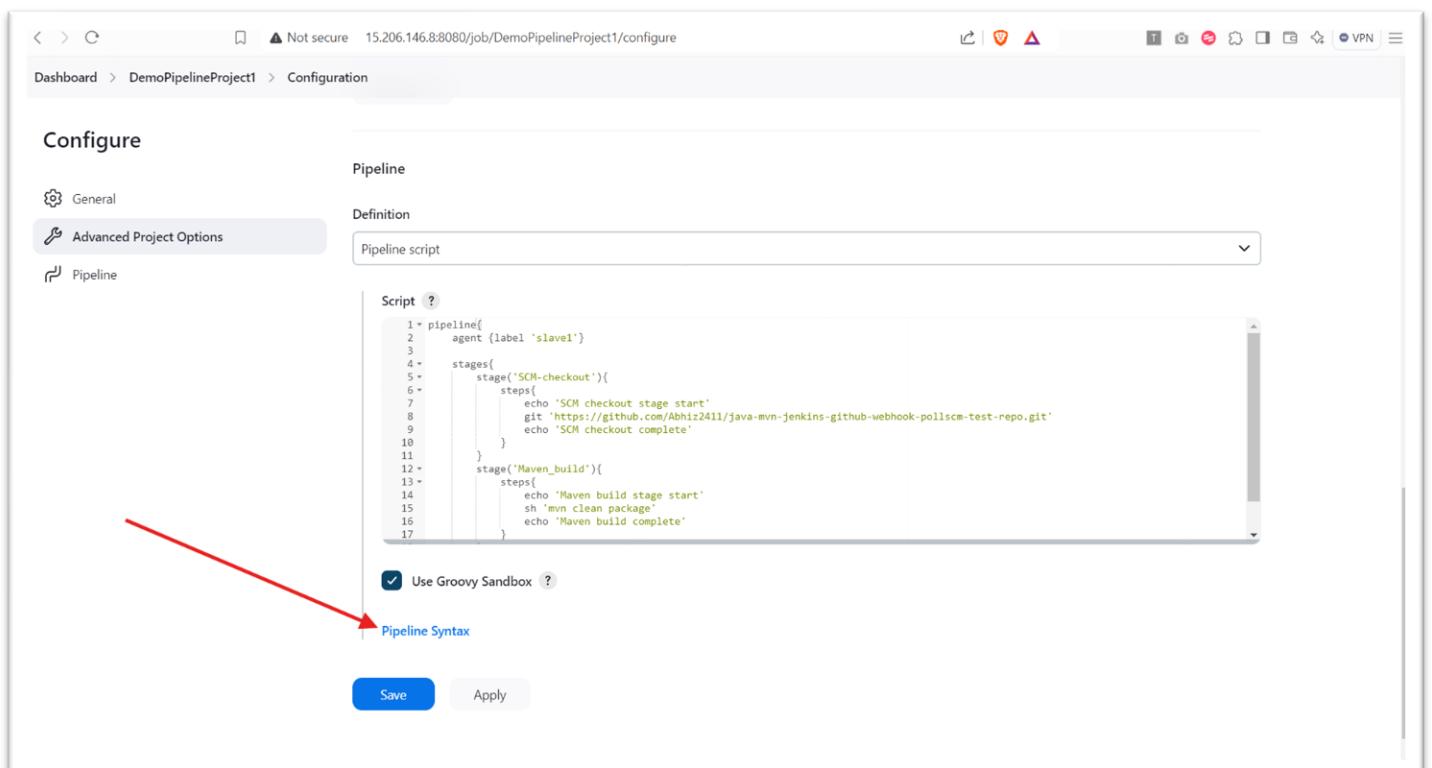


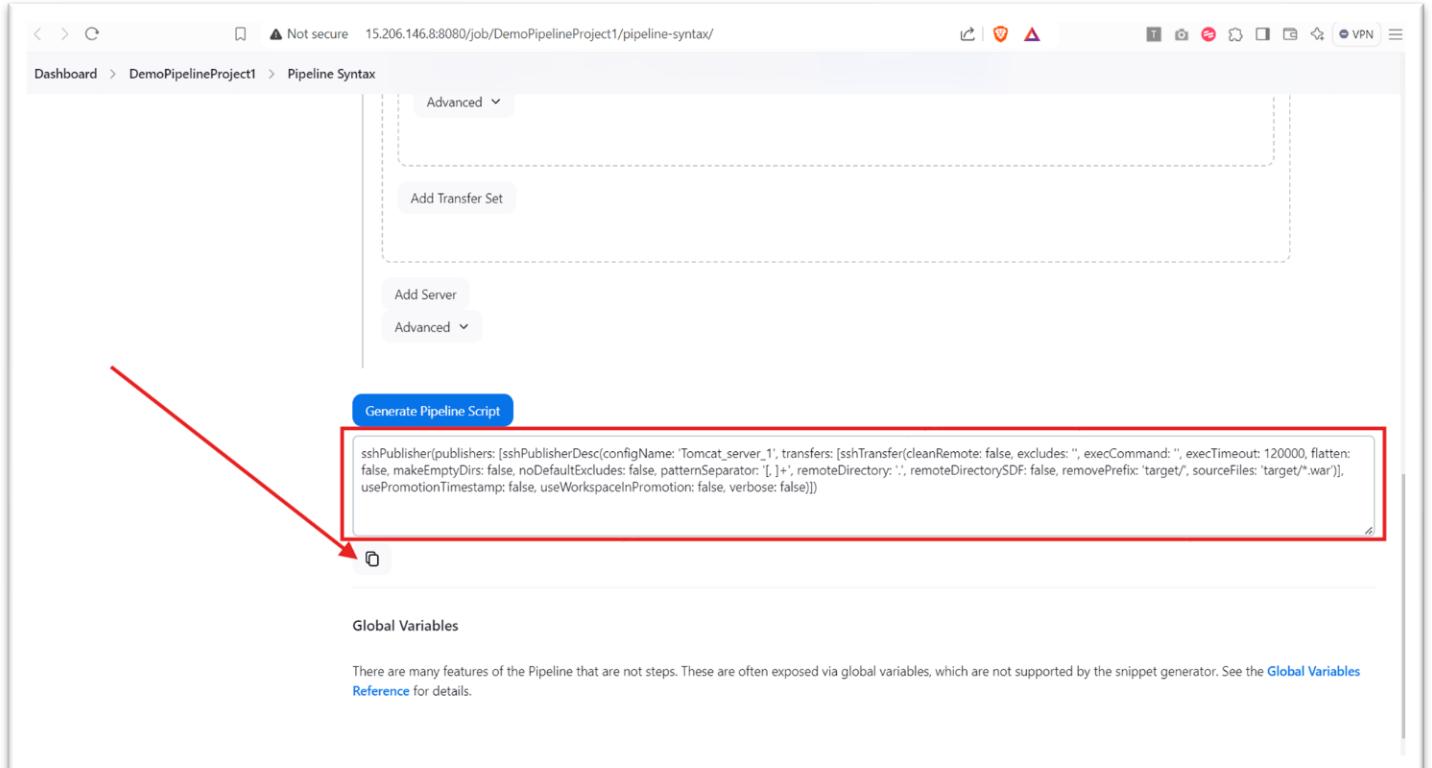
Fig. 3.21: Pipeline syntax option

This screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. On the left, a sidebar lists various snippets: Declarative Directive Generator, Declarative Online Documentation, Steps Reference, Global Variables Reference, Online Documentation, Examples Reference, and IntelliJ IDEA GDSL. The main area is titled "Overview" and contains a sample step: "sshPublisher: Send build artifacts over SSH". Below this, a detailed configuration for an "SSH Server" is shown. The "Name" field is set to "Tomcat\_server\_1". Under "Transfers", there is a "Transfer Set" section with fields for "Source files" (set to "target/\*.war"), "Remove prefix" (set to "target/"), and "Remote directory" (empty). A red arrow points from the "Name" field to the "Tomcat\_server\_1" value. Another red arrow points from the "Source files" field to "target/\*.war". A third red arrow points from the "Remove prefix" field to "target/". A fourth red arrow points from the "Remote directory" field.

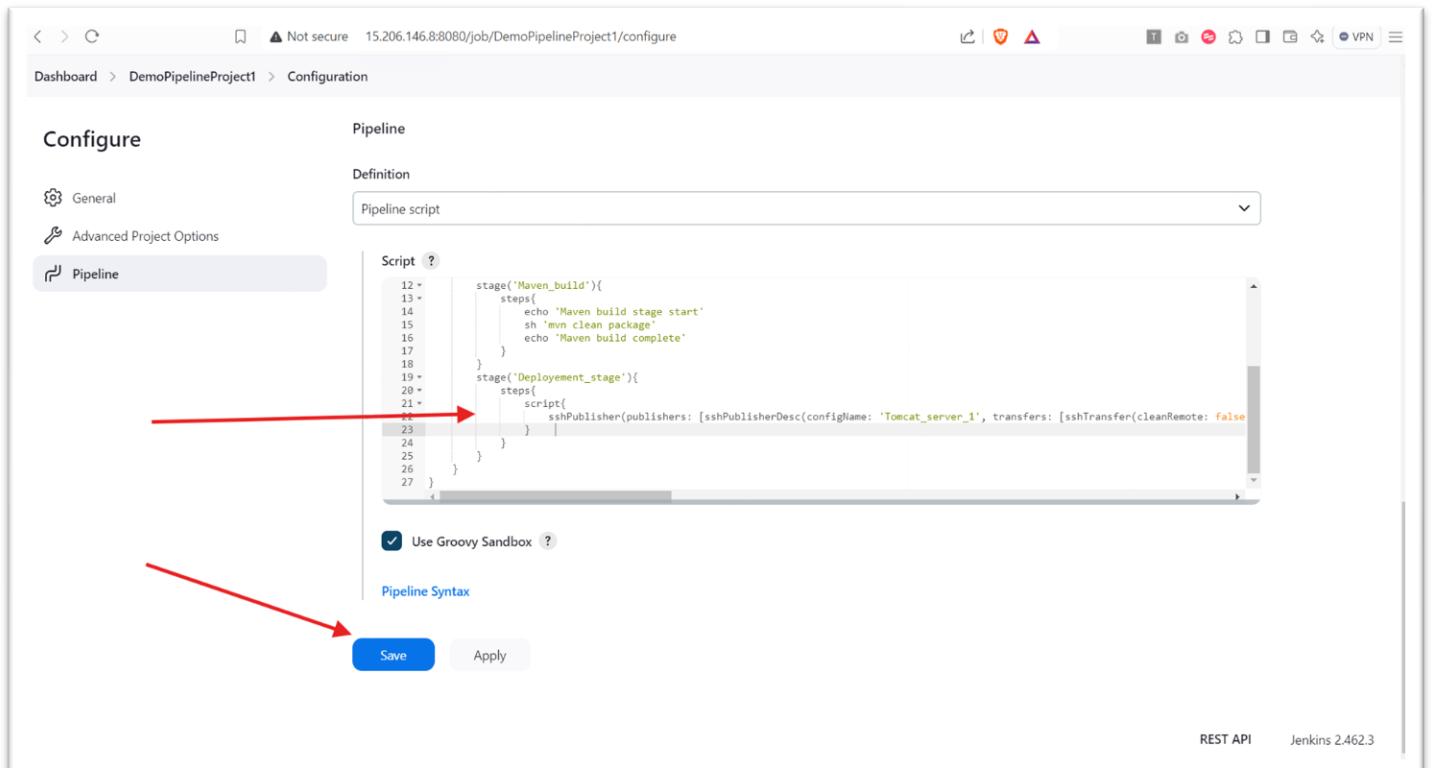
*Fig. 3.22: Publish over SSH configuration 1*

This screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. At the top, a message says "Either Source files, Exec command or both must be supplied". Below it, a note states: "All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables". There is an "Advanced" dropdown. On the left, there are buttons for "Add Transfer Set", "Add Server", and "Advanced". A large blue button at the bottom is labeled "Generate Pipeline Script". A red arrow points from the bottom-left towards this button. At the very bottom, a section titled "Global Variables" provides a note about global variables and their support by the snippet generator.

*Fig. 3.23: Generate pipeline syntax*



*Fig. 3.24: Copy generated pipeline syntax script*



*Fig. 3.25: Paste generated pipeline script*

The screenshot shows the Jenkins dashboard for the project 'DemoPipelineProject1'. The top navigation bar indicates the URL is 15.206.146.8:8080/job/DemoPipelineProject1/. The dashboard features a sidebar with various project management options like Status, Changes, Build Now (which has a red arrow pointing to it), Configure, Delete Pipeline, GitHub, Stages, Rename, and Pipeline Syntax. The main content area displays the project name 'DemoPipelineProject1' with a green checkmark icon, a brief description 'A demo pipeline project to demonstrate maven build using Jenkins', and a 'Permalinks' section listing recent builds. Below this is a 'Build History' card showing the most recent build (#1) from '5 Oct 2024, 23:04' with links for Atom feed for all and Atom feed for failures. At the bottom right, there are links for 'REST API' and 'Jenkins 2.462.3'.

Fig. 3.26: Schedule for build

The screenshot shows the Jenkins build status for 'Build #3 (6 Oct 2024, 16:23:58)'. The top navigation bar is identical to Fig. 3.26. The main content area shows the build status with a green checkmark icon. It includes a 'Status' card with options like Changes, Console Output, Edit Build Information, Timings, Git Build Data, Pipeline Overview, Pipeline Console, Thread Dump, Pause/resume, Replay, Pipeline Steps, Workspaces, and Previous Build. To the right of the status card, there are buttons for 'Keep this build forever' (disabled), 'Started 10 sec ago', 'Add description', and a link to 'Build #2'. Below the status card, detailed build information is provided: 'Started by user Abhijit Lalasaheb Zende', 'This run spent 2 ms waiting in the queue', 'Revision: e4900a255f0e3fd8240163504fc7a398499c7f1a', 'Repository: https://github.com/Abhiz2411/java-mvn-jenkins-github-webhook-pollsrm-test-repo.git', and a note about refs/remotes/origin/master. At the bottom right, there are links for 'REST API' and 'Jenkins 2.462.3'.

Fig. 3.27: Build status

```
devopsadmin@ip-172-31-3-176:~
ubuntu@ip-172-31-3-176:~$ sudo -i
root@ip-172-31-3-176:~# su - devopsadmin
devopsadmin@ip-172-31-3-176:~$ ll /opt/tomcat/webapps/
total 44688
drwxr-xr-x 8 devopsadmin root 4096 Oct 6 16:24 ./
rwxr-xr-x 9 devopsadmin root 4096 Oct 6 15:13 ../
rwxr-x--- 3 devopsadmin root 4096 Oct 6 15:13 ROOT/
rwxr-x--- 5 root root 4096 Oct 6 16:24 demo-1.0-SNAPSHOT/
rwxr-x--- 1 devopsadmin devopsadmin 45724692 Oct 6 16:24 demo-1.0-SNAPSHOT.war
rwxr-x--- 10 devopsadmin root 4096 Oct 6 15:13 docs/
rwxr-x--- 7 devopsadmin root 4096 Oct 6 15:13 examples/
rwxr-x--- 6 devopsadmin root 4096 Oct 6 15:13 host-manager/
rwxr-x--- 6 devopsadmin root 4096 Oct 6 15:13 manager/
devopsadmin@ip-172-31-3-176:~$ |
```

Fig. 3.28: Long listing of Tomcat server webapps folder

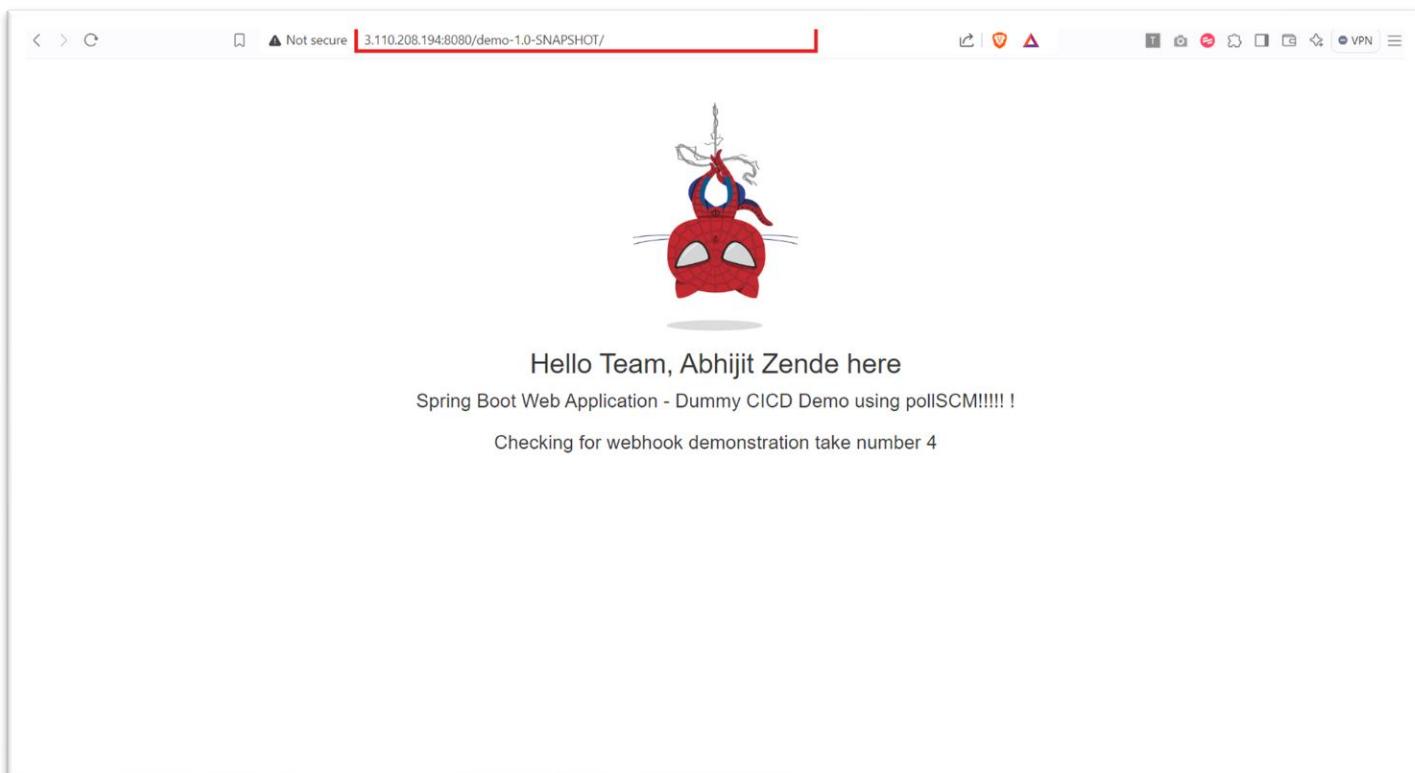


Fig. 3.29: Browser view of tomcat server after deployment

## **4. L4 - Automate the CICD Pipeline using GitHub webhook and Poll-SCM**

Ans.

### Step 1: Setup GitHub repository webhooks:

#### 1. Open the GitHub repository:

Open the desired GitHub repository which we want to SCM checkout and build

#### 2. Click on Settings of the repository

#### 3. From the left panel select ‘Webhooks’

#### 4. Click on ‘Add webhook’

#### 5. Fill in the details:

a. Payload URL: `http://<Jenkins_URL:8080>/github-webhook/`

b. Content type: application/json

c. Enable SSL certification

d. Just the push event //As per our choice

e. Check the ‘Active’ checkbox

f. Click on ‘Add Webhook’

### Step 2: Enable GitHub hook trigger in Jenkins:

#### 1. Open pipeline task in Jenkins

#### 2. Go to Job configuration

#### 3. Enable ‘GitHub hook trigger for GITScm polling’ under ‘Build Triggers’

### Step 3: Enable Poll SCM:

#### 1. Enable ‘Poll SCM’ under ‘Build Triggers’

#### 2. Write cron expression for scheduled builds

#### 3. Click save

## Step 4: Testing:

1. Make a commit to the repository
2. According to the cron expression the build will be scheduled
3. Check the build history for verification

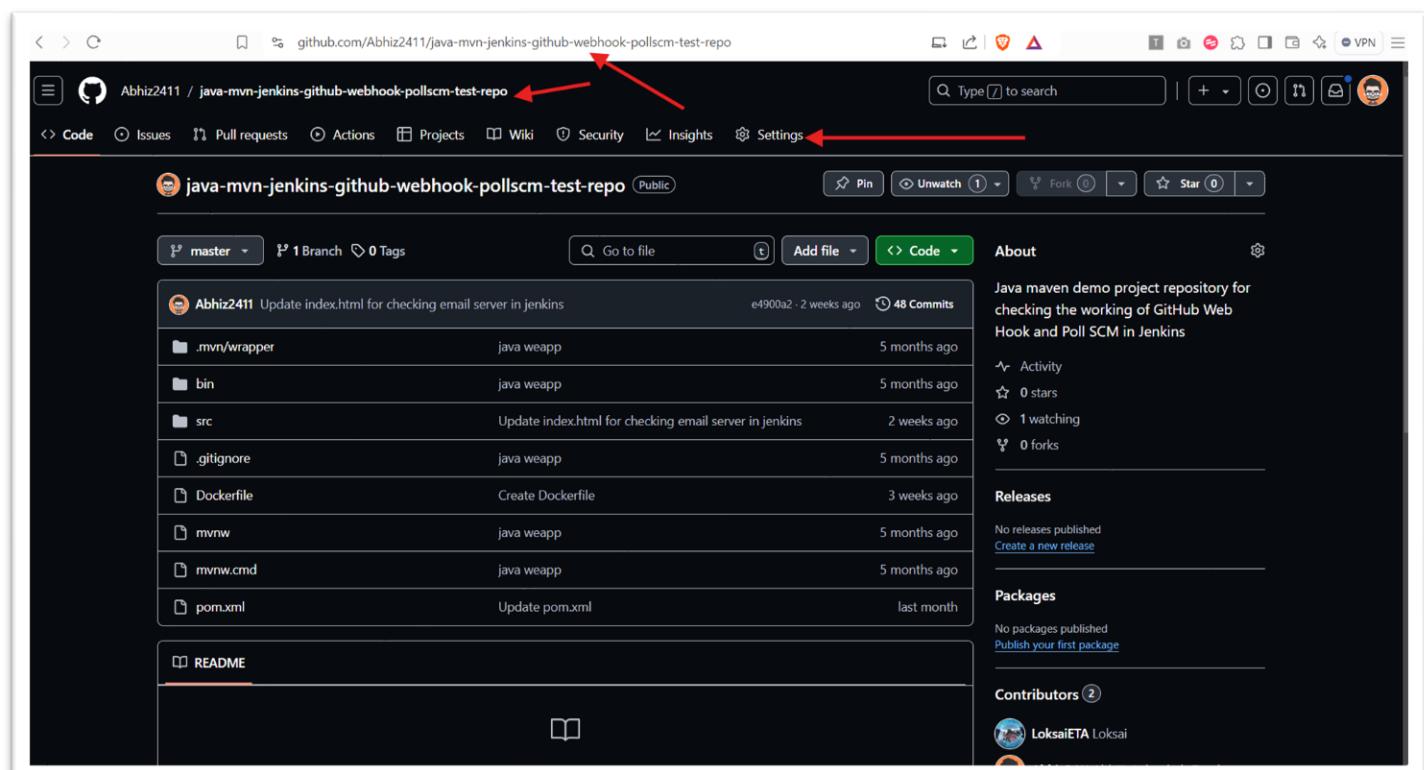


Fig. 4.01: Open GitHub repo

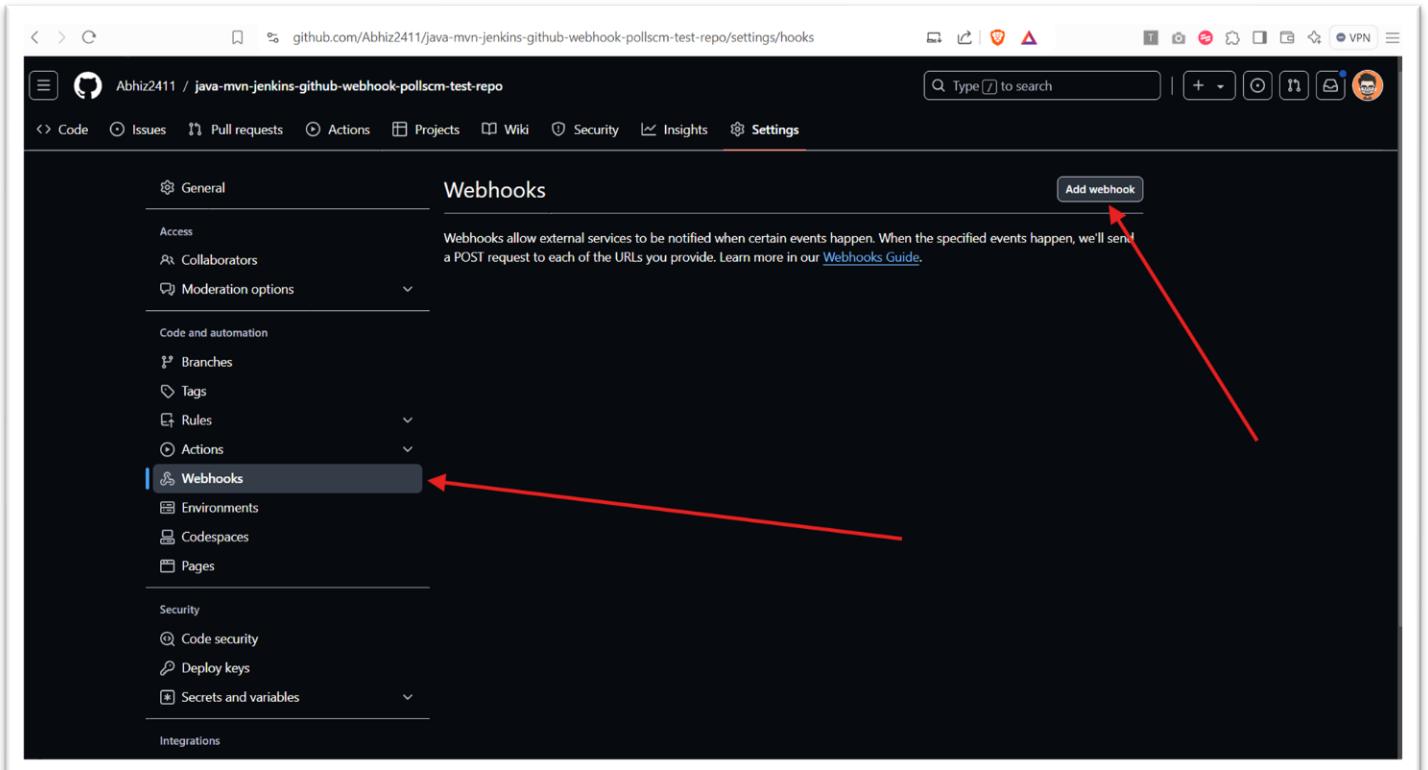


Fig. 4.02: Add webhook

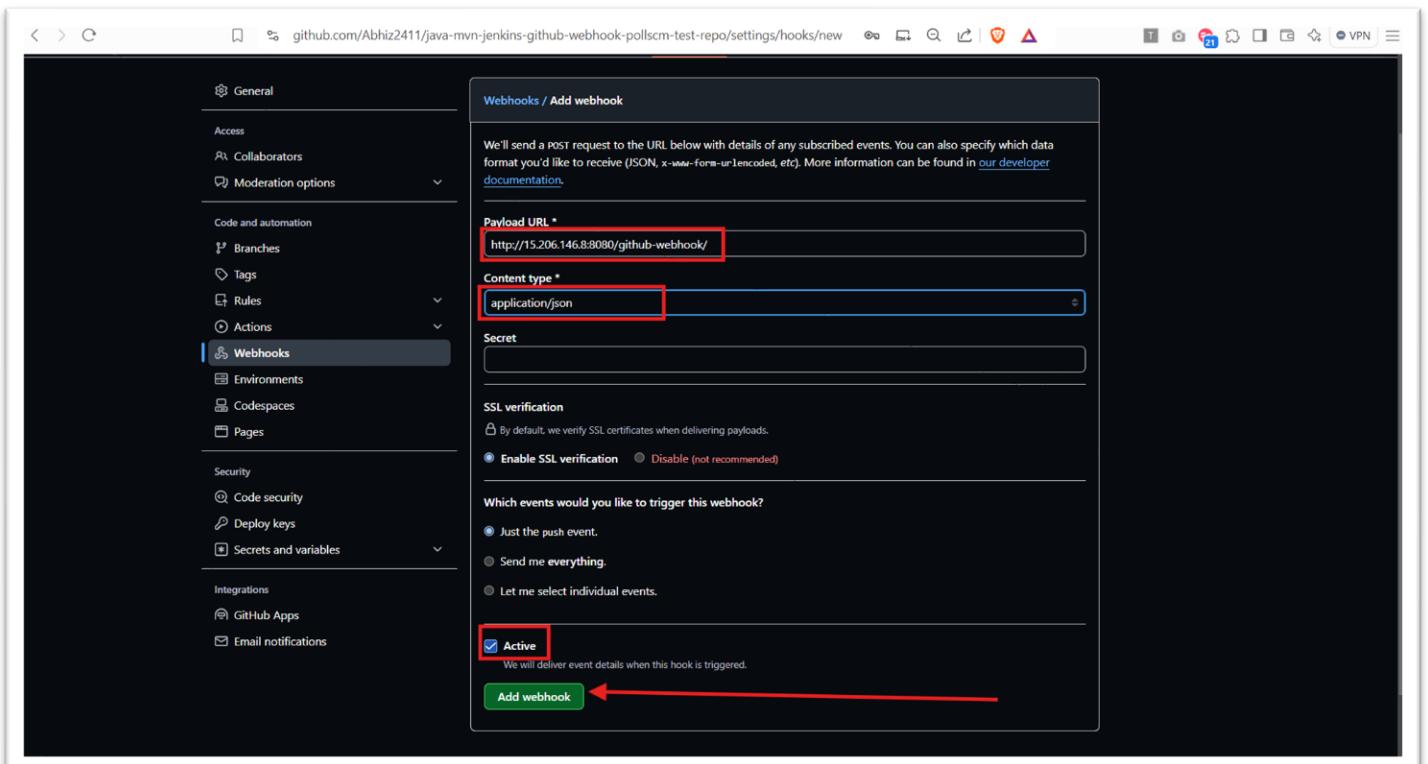


Fig. 4.03: GitHub webhook config

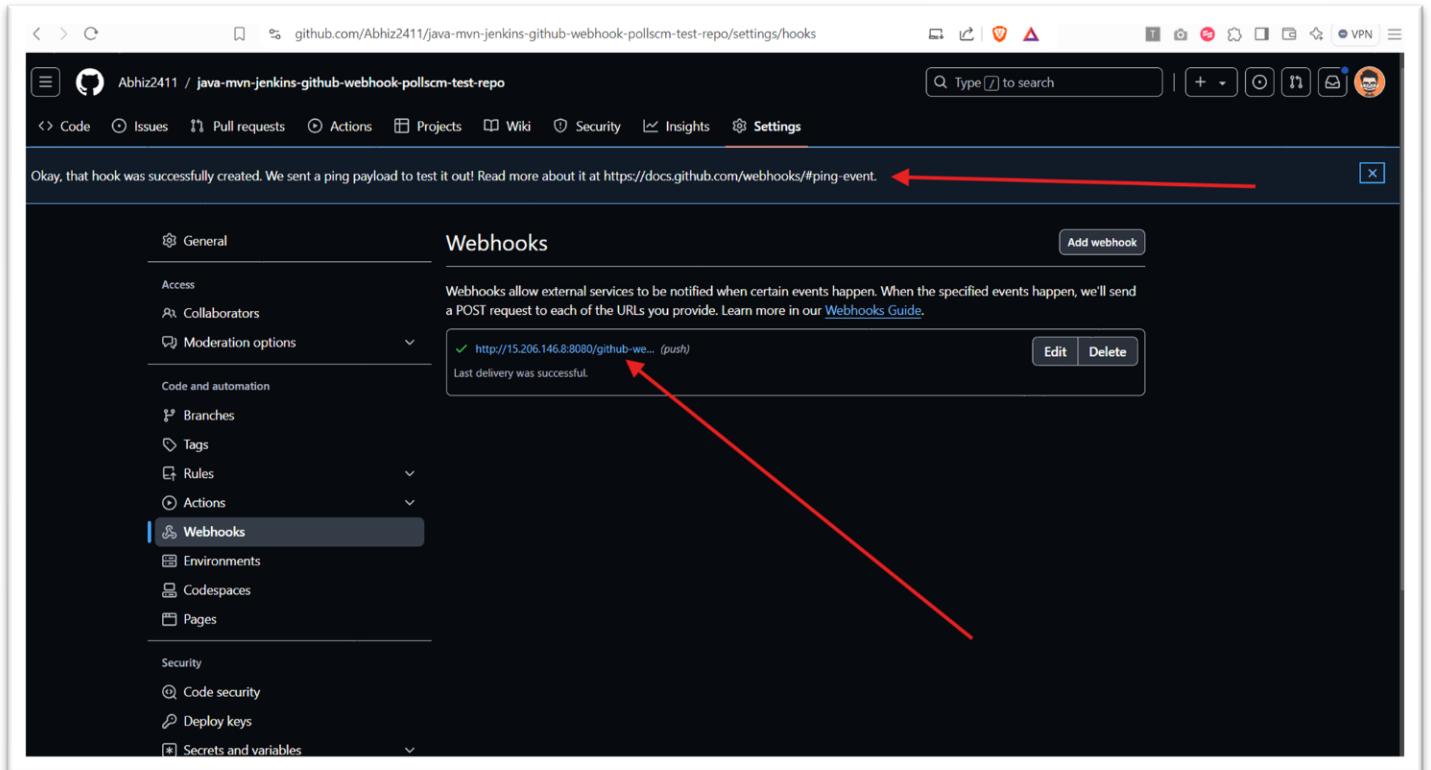


Fig. 4.04: Successful webhook creation

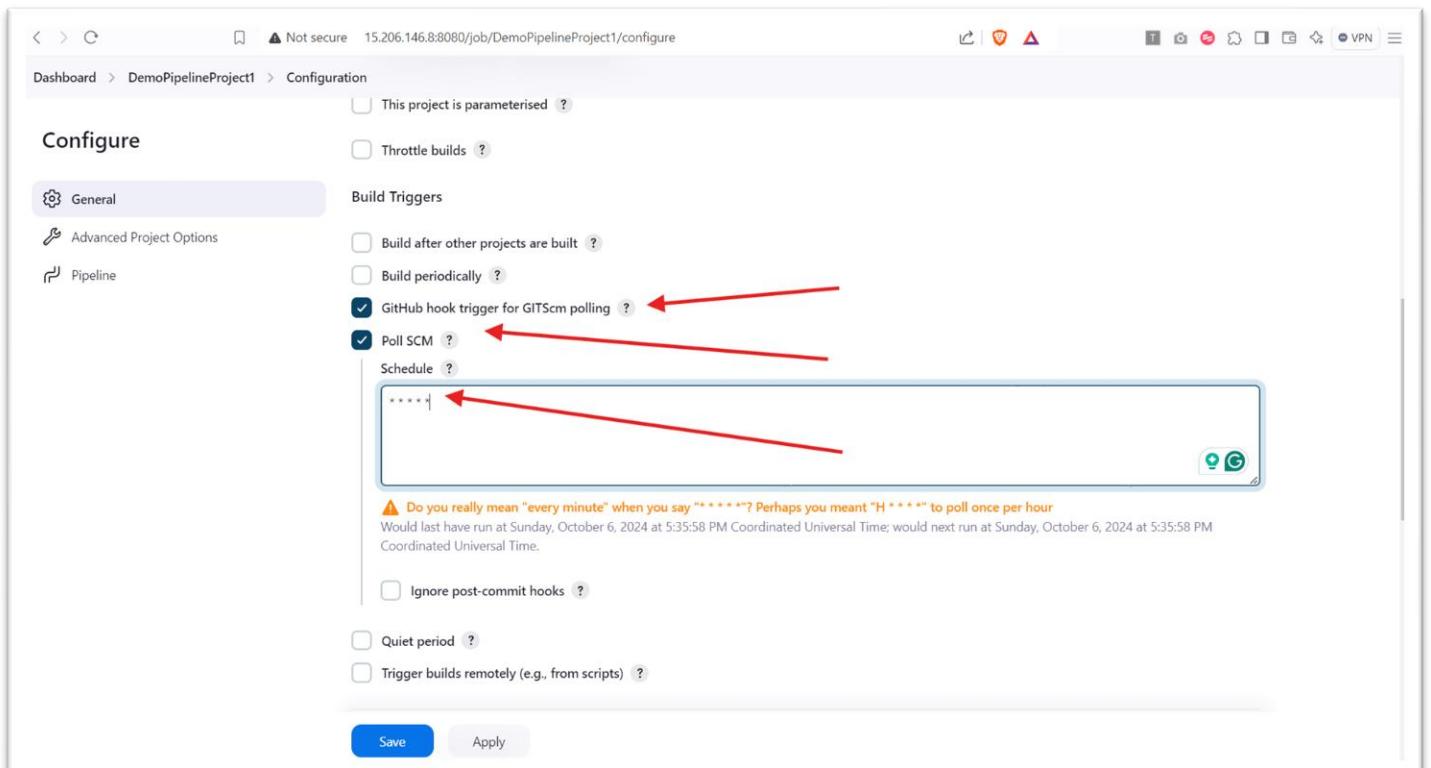


Fig. 4.05: Enable GitHub SCM polling

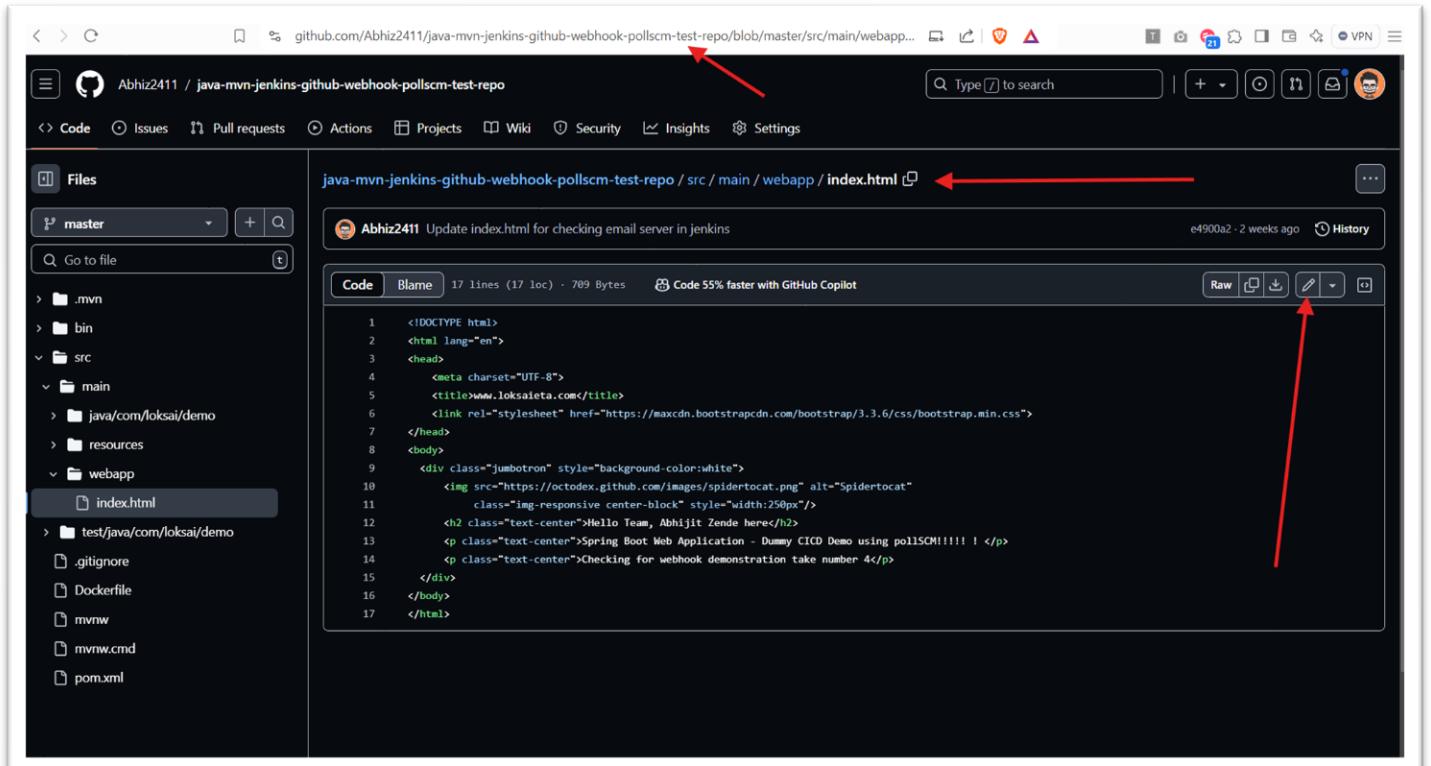


Fig. 4.06: Edit GitHub repo code for a new commit

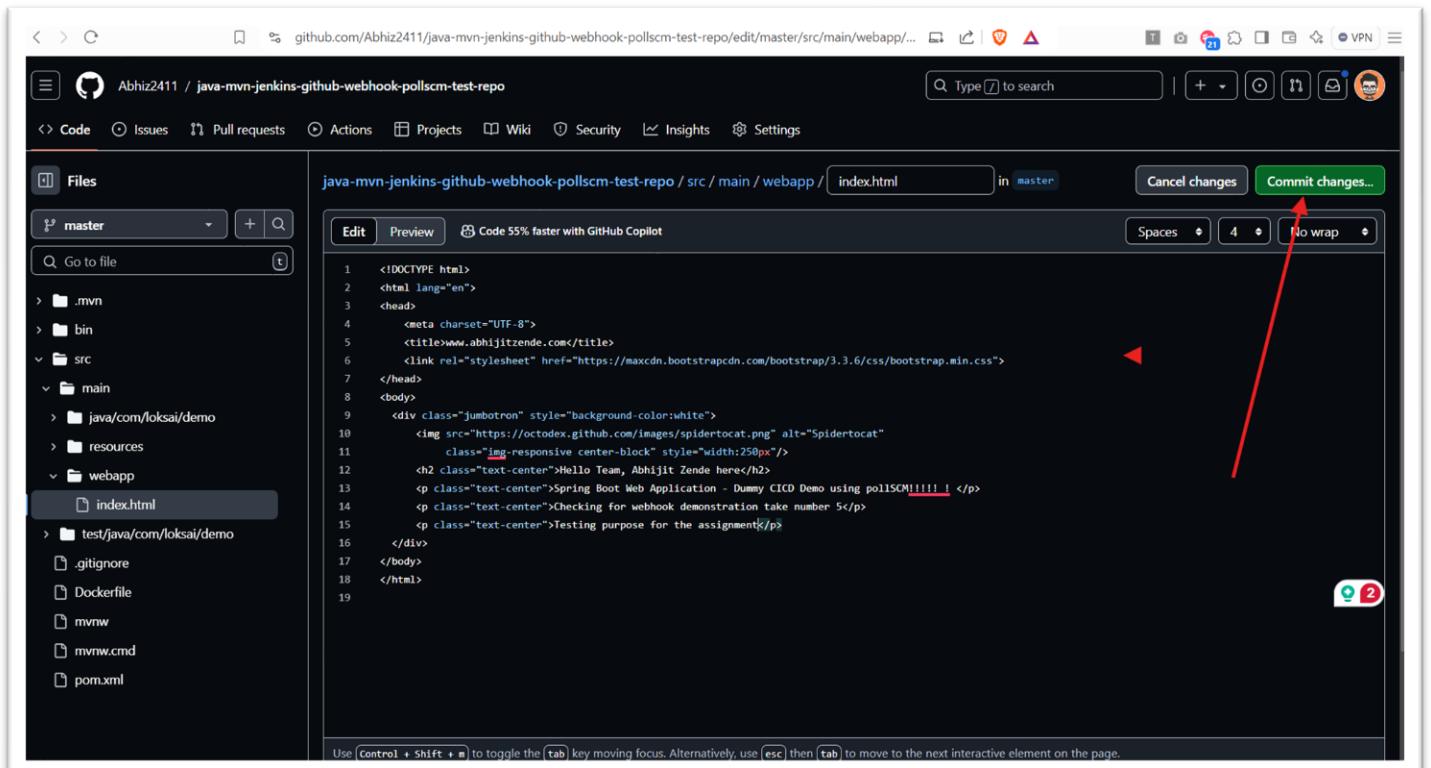


Fig. 4.07: Commit changes to main branch

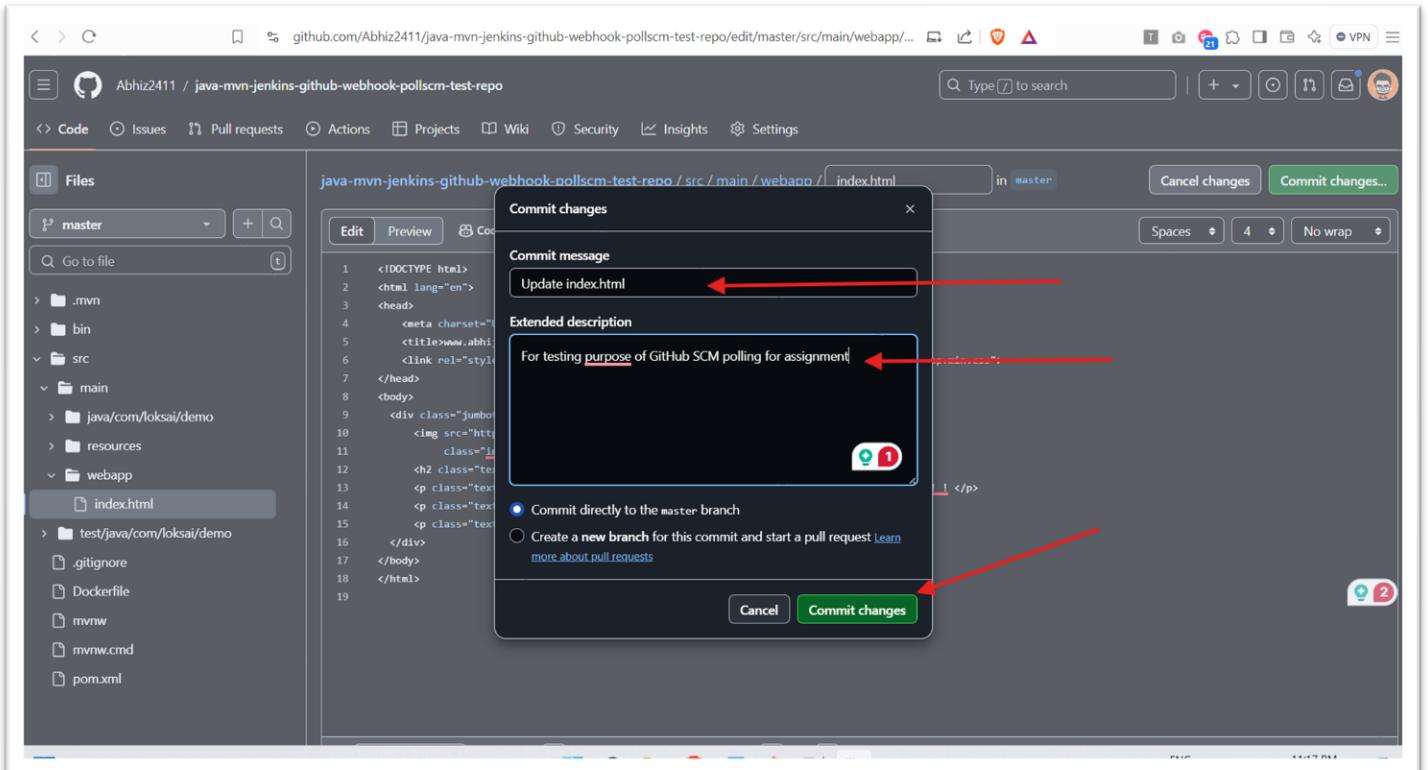


Fig. 4.08: Commit message for new commit

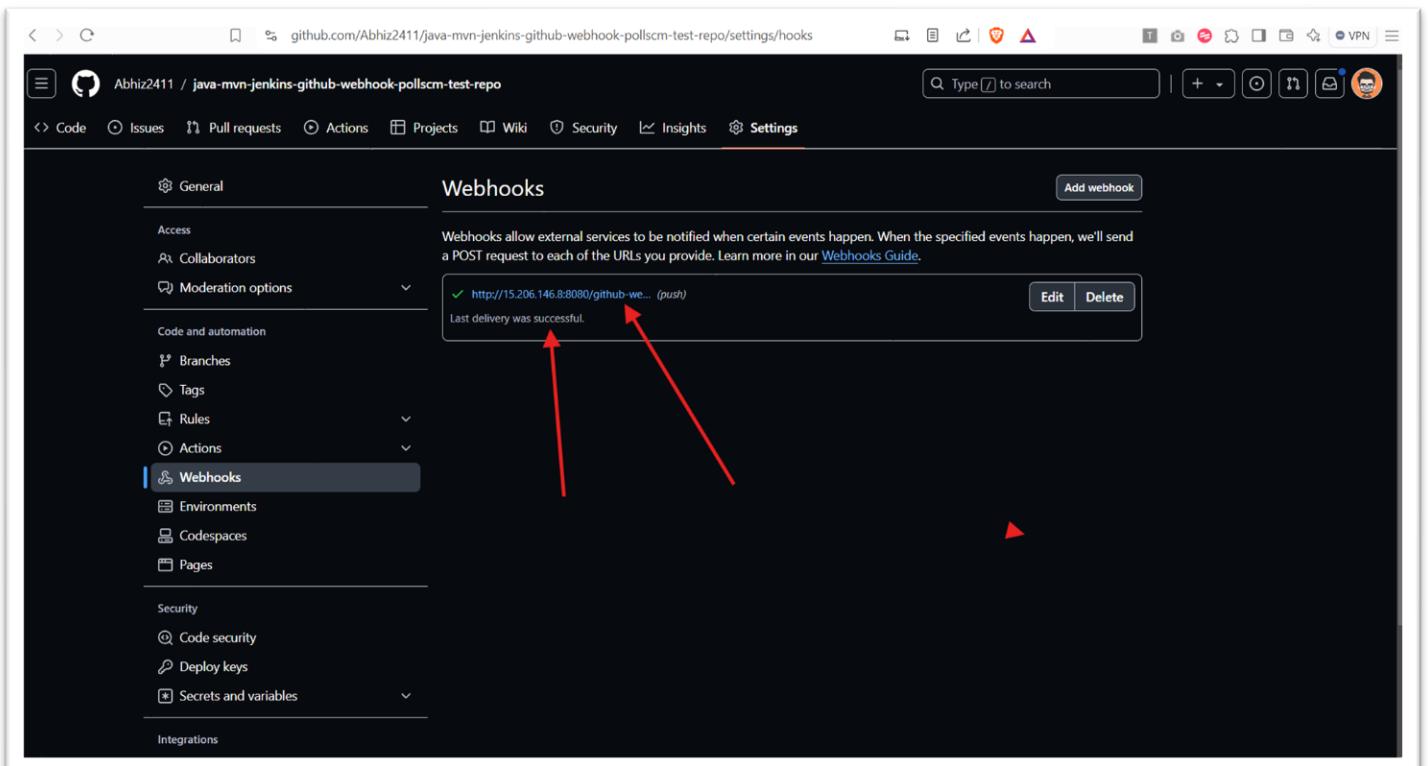


Fig. 4.09: Check GitHub webhook history

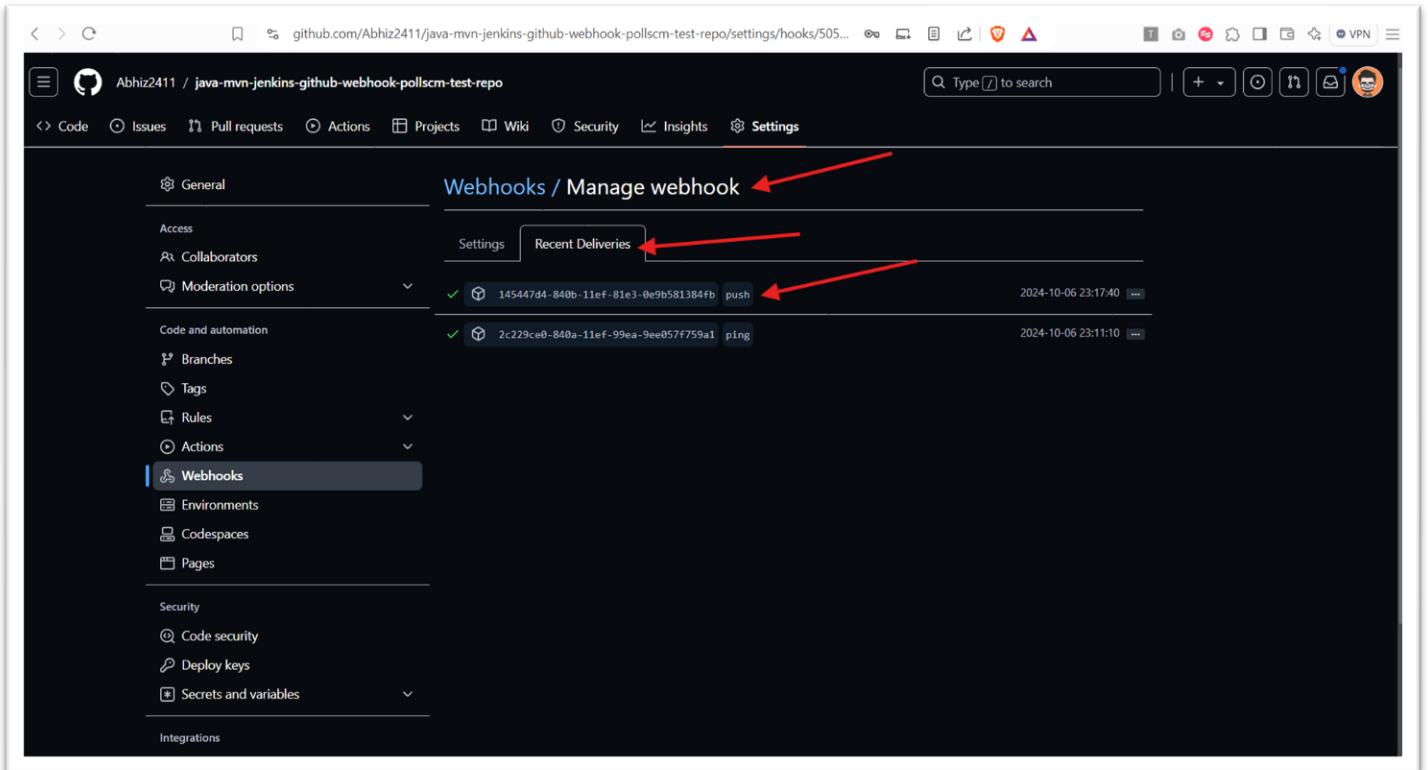


Fig. 4.10: Check recent delivery status

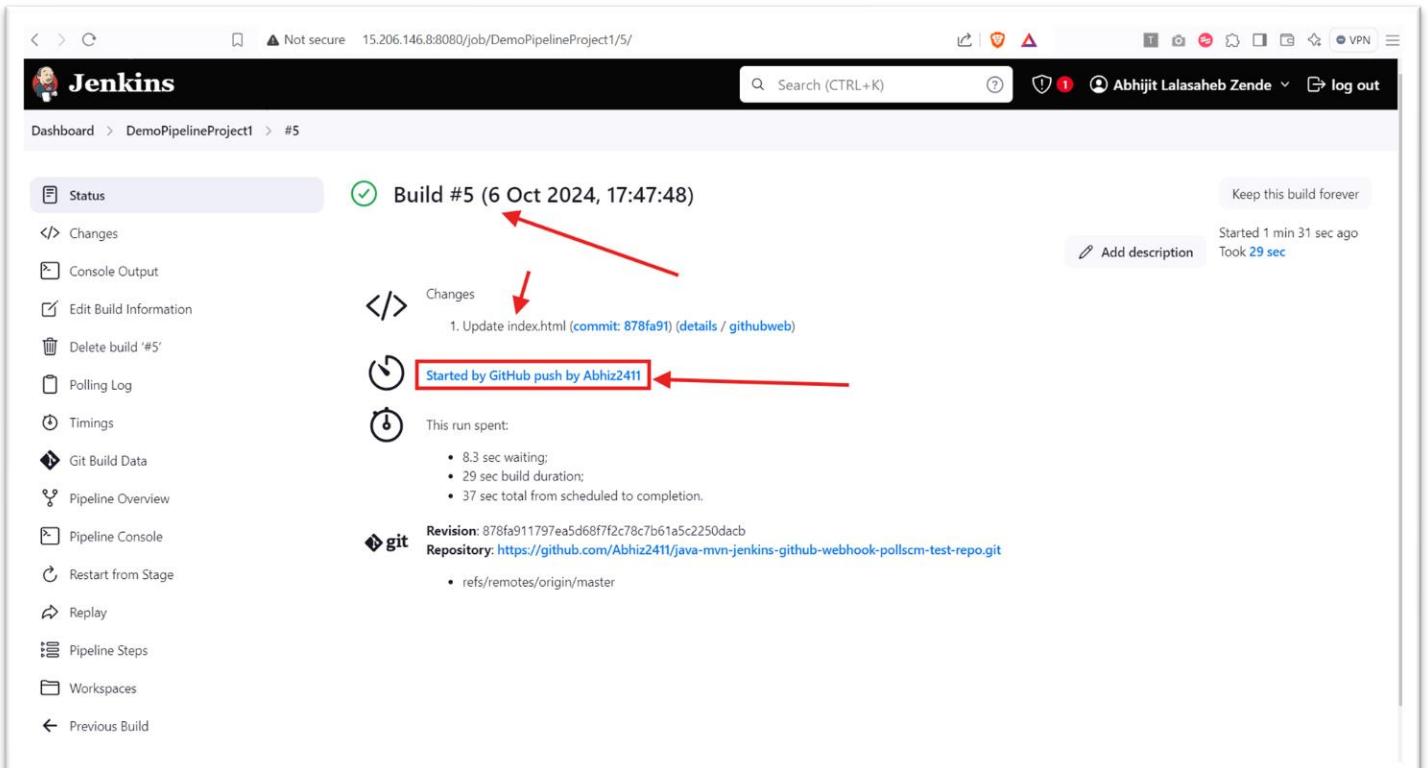
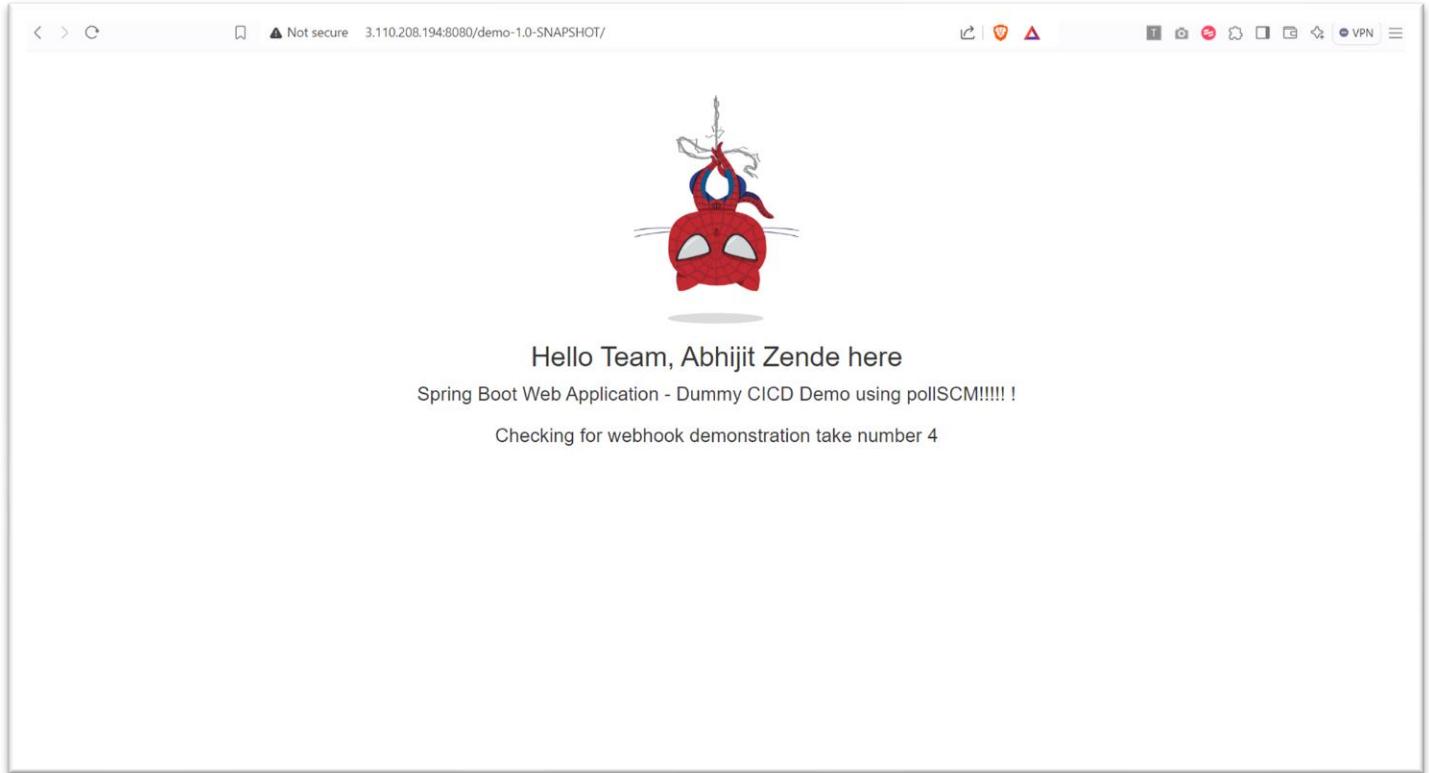


Fig. 4.11: Check Build Status for verification



*Fig. 4.12: Browser view after commit is made and deployment is done*

