

Assignment–5

Module-8: Continuous Integration using Jenkins

Submitted by : Shaik Junaid Adil

Date of Submission: 26-06-2024

Submitted to: Vikul

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

Step-1: Login to AWS Console and Launch 2 Instances- “Jenkins_Master” and “Jenkins_Slave”.

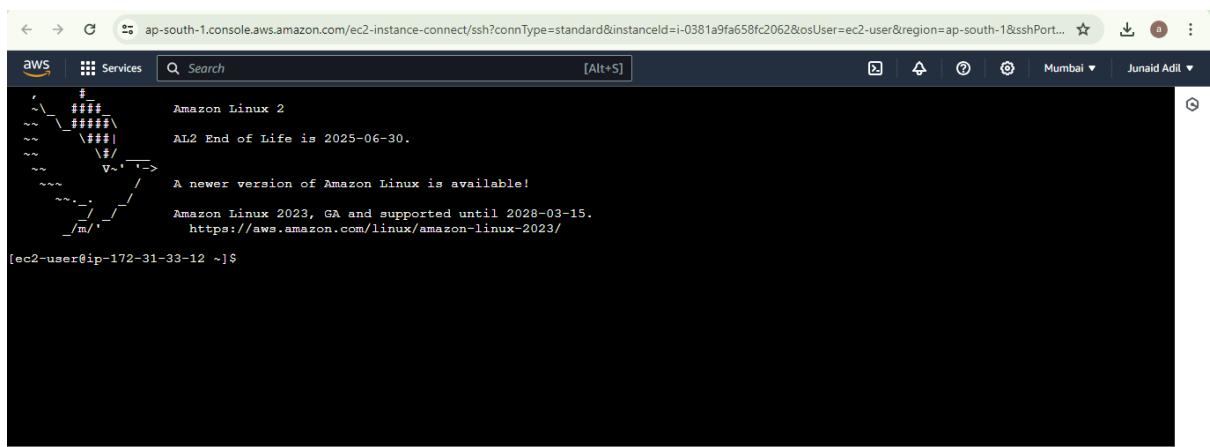
The screenshot shows the AWS EC2 Dashboard. On the left, a sidebar lists options like EC2 Global View, Events, Instances (with sub-options for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). The main area has a "Resources" section with a table showing 0 instances (running), 0 auto scaling groups, 0 dedicated hosts, 0 elastic IPs, 0 instances, 2 key pairs, 0 load balancers, 0 placement groups, 14 security groups, 0 snapshots, and 0 volumes. Below this is a "Launch instance" section with a "Launch instance" button. To the right, there's a "Service health" section with a "AWS Health Dashboard" button and a "EC2 Free Tier" section showing 1 EC2 free tier offers in use, with details about end-of-month forecasts and usage on EBS (13% storage space remaining).

The screenshot shows the "Launch an instance" wizard. The left panel has sections for "Name and tags" (with "Jenkins_Master" entered), "Application and OS Images (Amazon Machine Image)" (with a search bar), and "Quick Start". The right panel is titled "Summary" and includes fields for "Number of instances" (set to 2), "Software Image (AMI)" (Amazon Linux 2 Kernel 5.10 AMI...), "Virtual server type (instance type)" (t2.micro), "Firewall (security group)" (New security group), and "Storage (volumes)" (1 volume(s) - 8 GiB). At the bottom are "Cancel", "Launch instance" (highlighted in orange), and "Review commands" buttons.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like 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), and Elastic Block Store (Volumes). The main content area displays a table titled 'Instances (2) Info' with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. Two instances are listed: 'Jenkins_Master' (i-0381a9fa658fc2062) and 'Jenkins_Slave' (i-04ab5ef9c024b3590). Both instances are shown as 'Running'. The 'Status check' column indicates they are 'Initializing'. The 'Alarm status' and 'Availability Zone' columns show 'ap-south-1a'.

Step-2: Connect to Jenkins Master Instance

The screenshot shows the AWS EC2 Instance Details page for the Jenkins_Master instance (i-0381a9fa658fc2062). The left sidebar is identical to the previous screenshot. The main content area shows the instance details for 'i-0381a9fa658fc2062 (Jenkins_Master)'. It includes tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under the Details tab, there's an 'Instance summary' section with fields for Instance ID (i-0381a9fa658fc2062 (Jenkins_Master)), Public IPv4 address (13.233.245.185), Private IPv4 addresses (172.31.33.12), Instance state (Running), Public IPv4 DNS (ec2-13-233-245-185.ap-south-1.compute.amazonaws.com), and a note about IPv6 address (-).

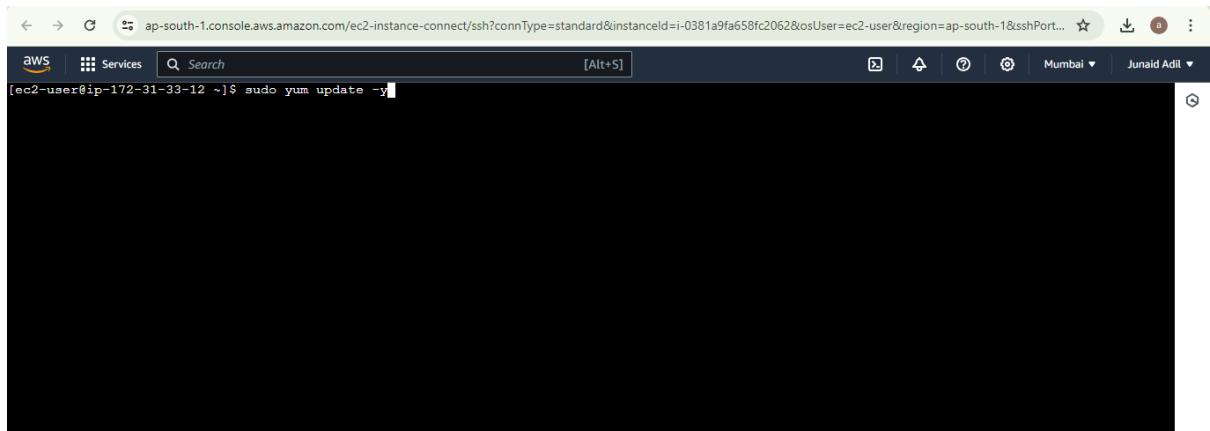


```
[ec2-user@ip-172-31-33-12 ~]$
```

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step-3: Update all the software packages on the Instance using command “ sudo yum update -y”



```
[ec2-user@ip-172-31-33-12 ~]$ sudo yum update -y
```

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows a terminal window within the AWS CloudShell interface. The terminal output is as follows:

```
[ec2-user@ip-172-31-33-12 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No packages marked for update
[ec2-user@ip-172-31-33-12 ~]$
```

Below the terminal, a summary of the instance is displayed:

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

At the bottom of the screen, there are navigation links for CloudShell and Feedback, along with copyright information and links for Privacy, Terms, and Cookie preferences.

Step-4: Add the Jenkins repo using command “**sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo**”

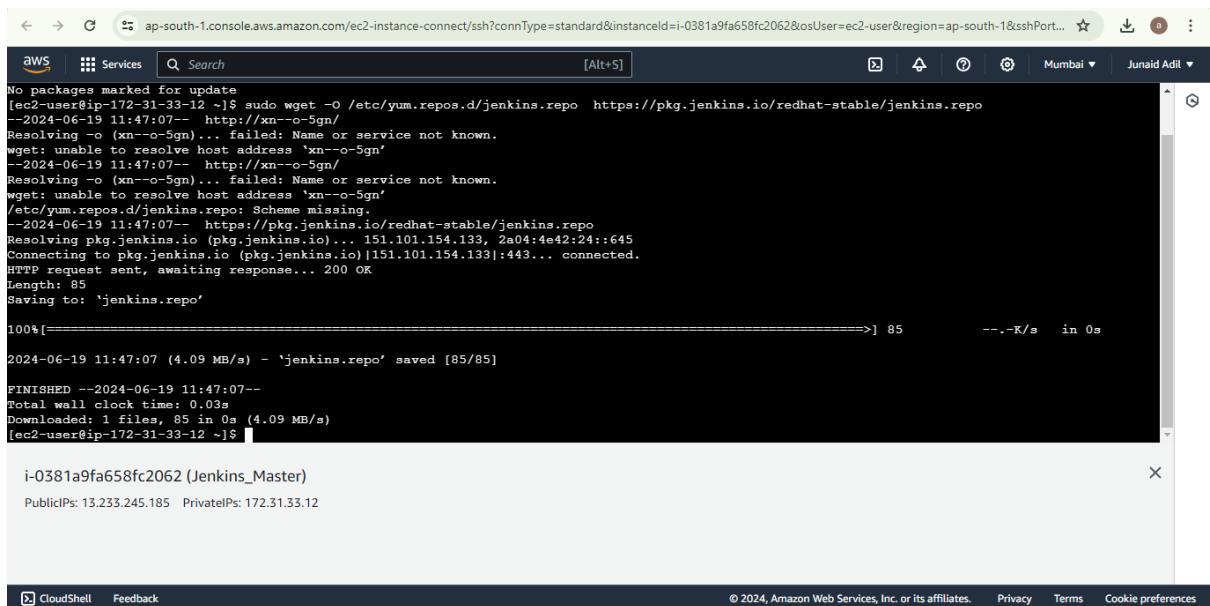
The screenshot shows a terminal window within the AWS CloudShell interface. The terminal output is as follows:

```
[ec2-user@ip-172-31-33-12 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No packages marked for update
[ec2-user@ip-172-31-33-12 ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
[ec2-user@ip-172-31-33-12 ~]$
```

Below the terminal, a summary of the instance is displayed:

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

At the bottom of the screen, there are navigation links for CloudShell and Feedback, along with copyright information and links for Privacy, Terms, and Cookie preferences.



```
No packages marked for update
[ec2-user@ip-172-31-33-12 ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-06-19 11:47:07-- http://xn--o-5gn/
Resolving -- (xn--o-5gn)... failed: Name or service not known.
wget: unable to resolve host address 'xn--o-5gn'
--2024-06-19 11:47:07-- http://xn--o-5gn/
Resolving --o (xn--o-5gn)... failed: Name or service not known.
wget: unable to resolve host address 'xn--o-5gn'
/etc/yum.repos.d/jenkins.repo: Scheme missing.
--2024-06-19 11:47:07-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
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: 85
Saving to: 'jenkins.repo'

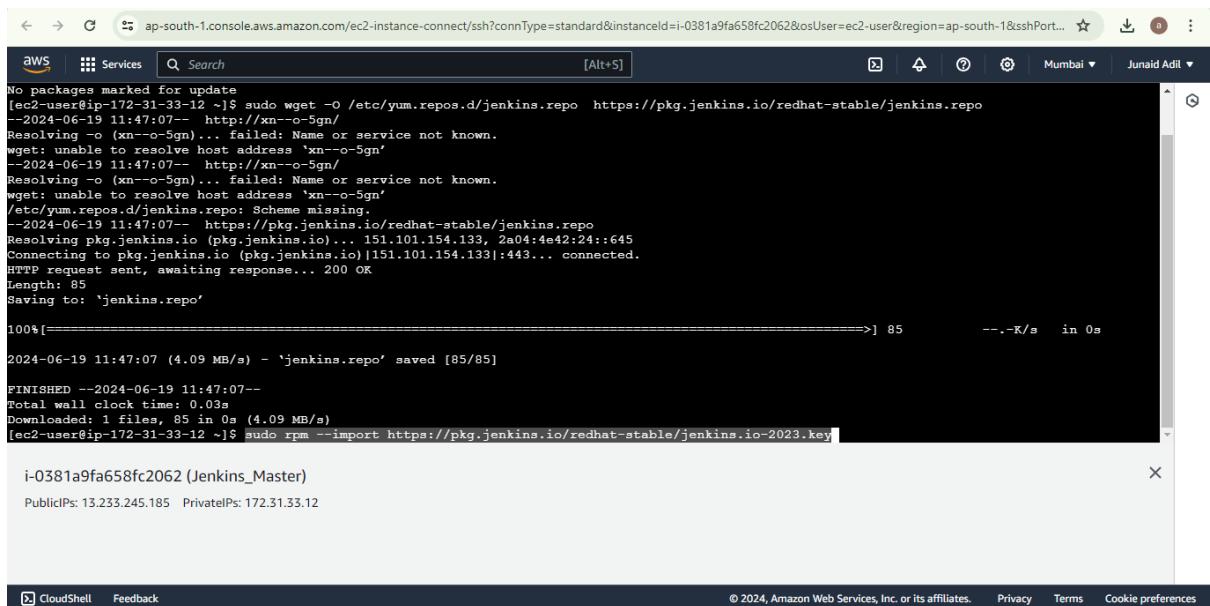
100%[=====] 85      --.-K/s   in 0s

2024-06-19 11:47:07 (4.09 MB/s) - 'jenkins.repo' saved [85/85]

FINISHED --2024-06-19 11:47:07--
Total wall clock time: 0.03s
Downloaded: 1 files, 85 in 0s (4.09 MB/s)
[ec2-user@ip-172-31-33-12 ~]$
```

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

Step 5: Import a key-file from Jenkins-CI to enable the installation from the package. Using command “**sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key**”



```
No packages marked for update
[ec2-user@ip-172-31-33-12 ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-06-19 11:47:07-- http://xn--o-5gn/
Resolving -- (xn--o-5gn)... failed: Name or service not known.
wget: unable to resolve host address 'xn--o-5gn'
--2024-06-19 11:47:07-- http://xn--o-5gn/
Resolving --o (xn--o-5gn)... failed: Name or service not known.
wget: unable to resolve host address 'xn--o-5gn'
/etc/yum.repos.d/jenkins.repo: Scheme missing.
--2024-06-19 11:47:07-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
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: 85
Saving to: 'jenkins.repo'

100%[=====] 85      --.-K/s   in 0s

2024-06-19 11:47:07 (4.09 MB/s) - 'jenkins.repo' saved [85/85]

FINISHED --2024-06-19 11:47:07--
Total wall clock time: 0.03s
Downloaded: 1 files, 85 in 0s (4.09 MB/s)
[ec2-user@ip-172-31-33-12 ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

```
[ec2-user@ip-172-31-33-12 ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-06-19 11:47:07-- http://xn--o-5gn/
Resolving -o (xn--o-5gn)... failed: Name or service not known.
wget: unable to resolve host address 'xn--o-5gn'
--2024-06-19 11:47:07-- http://xn--o-5gn/
Resolving -o (xn--o-5gn)... failed: Name or service not known.
wget: unable to resolve host address 'xn--o-5gn'
/etc/yum.repos.d/jenkins.repo: Scheme missing.
--2024-06-19 11:47:07-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
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: 85
Saving to: 'jenkins.repo'

100%[=====] 85      --.-K/s   in 0s

2024-06-19 11:47:07 (4.09 MB/s) - 'jenkins.repo' saved [85/85]

FINISHED --2024-06-19 11:47:07--
Total wall clock time: 0.03s
Downloaded: 1 files, 85 in 0s (4.09 MB/s)
[ec2-user@ip-172-31-33-12 ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[ec2-user@ip-172-31-33-12 ~]$
```

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

Step 6: run command “**sudo yum upgrade**”

```
--2024-06-19 11:47:07-- http://xn--o-5gn/
Resolving -o (xn--o-5gn)... failed: Name or service not known.
wget: unable to resolve host address 'xn--o-5gn'
/etc/yum.repos.d/jenkins.repo: Scheme missing.
--2024-06-19 11:47:07-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
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: 85
Saving to: 'jenkins.repo'

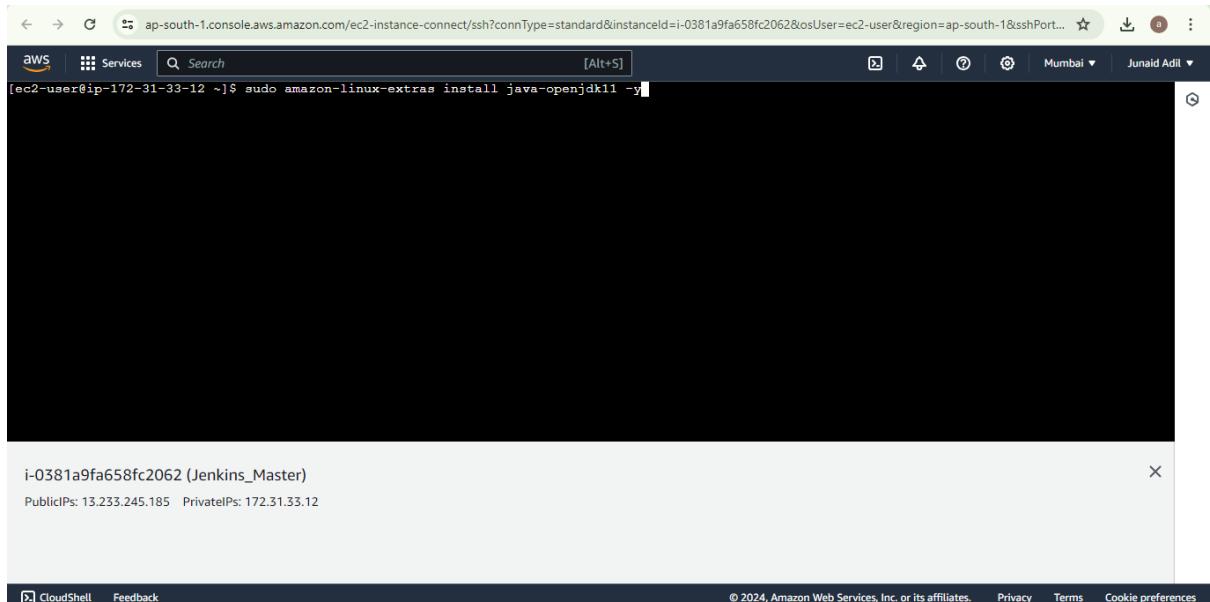
100%[=====] 85      --.-K/s   in 0s

2024-06-19 11:47:07 (4.09 MB/s) - 'jenkins.repo' saved [85/85]

FINISHED --2024-06-19 11:47:07--
Total wall clock time: 0.03s
Downloaded: 1 files, 85 in 0s (4.09 MB/s)
[ec2-user@ip-172-31-33-12 ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[ec2-user@ip-172-31-33-12 ~]$ sudo yum upgrade
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No packages marked for update
[ec2-user@ip-172-31-33-12 ~]$
```

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

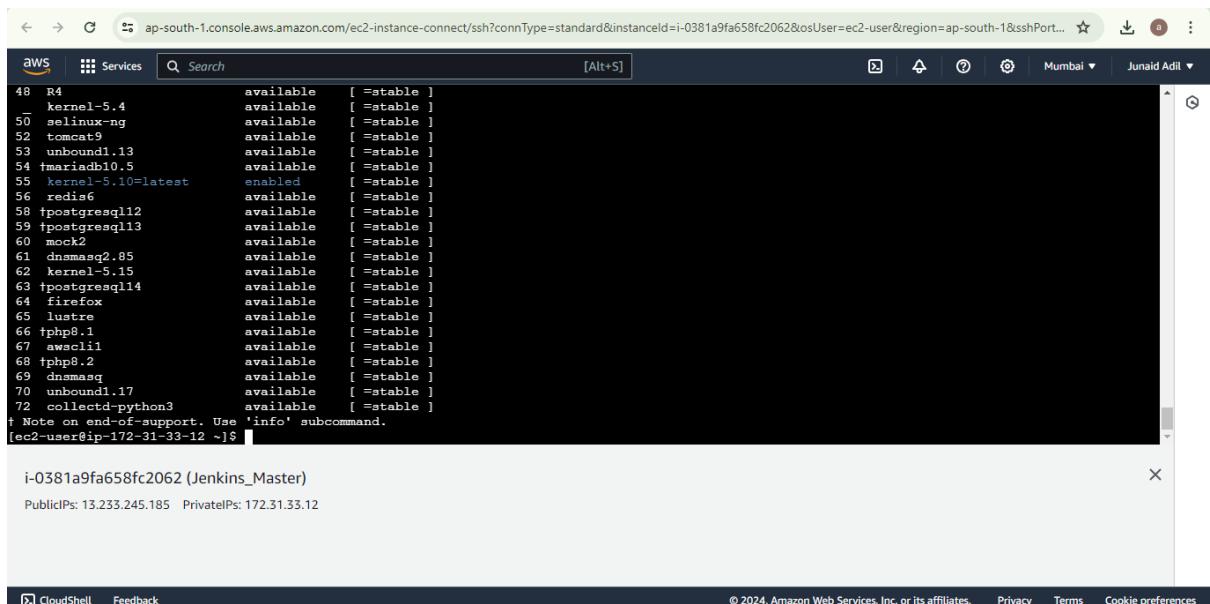
Step 7: Install Java using command “ sudo amazon-linux-extras install java-openjdk11 -y ”



```
[ec2-user@ip-172-31-33-12 ~]$ sudo amazon-linux-extras install java-openjdk11 -y
```

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

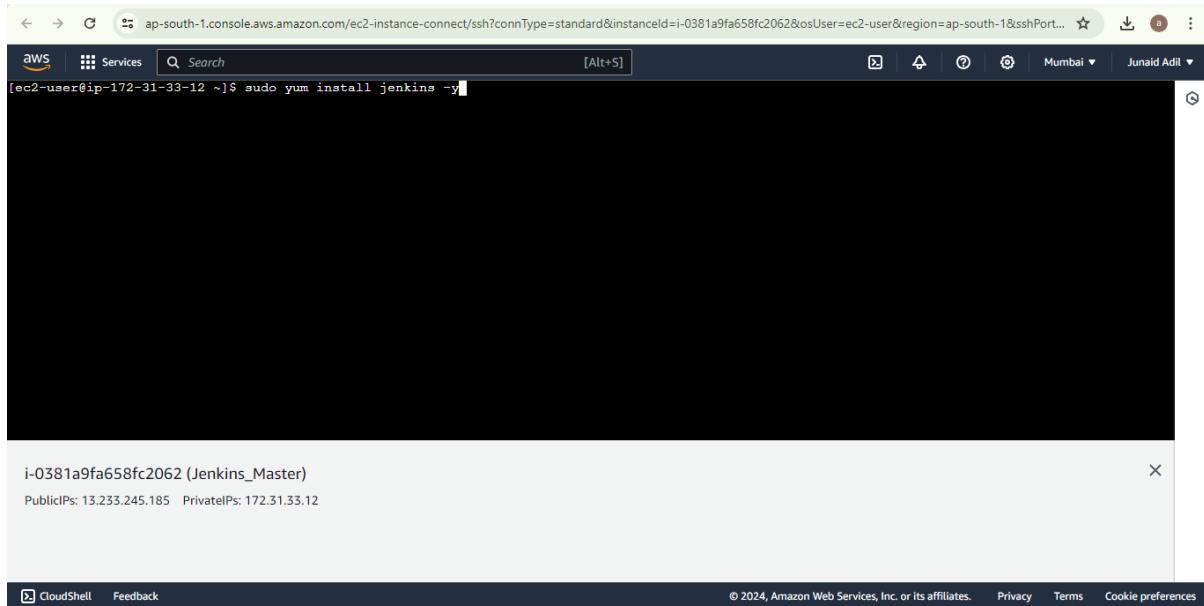


```
48 R4 available [ =stable ]
49 kernel-5.4 available [ =stable ]
50 selinux-ng available [ =stable ]
52 tomcat9 available [ =stable ]
53 unbound1.13 available [ =stable ]
54 mariadb10.5 available [ =stable ]
55 kernel-5.10+latest enabled [ =stable ]
56 redis6 available [ =stable ]
58 tpostgresql12 available [ =stable ]
59 tpostgresql13 available [ =stable ]
60 mock2 available [ =stable ]
61 dnsmasq2.85 available [ =stable ]
62 kernel-5.15 available [ =stable ]
63 tpostgresql14 available [ =stable ]
64 firefox available [ =stable ]
65 lustre available [ =stable ]
66 tphp8.1 available [ =stable ]
67 awsccli1 available [ =stable ]
68 tphp8.2 available [ =stable ]
69 dnsmasq available [ =stable ]
70 unbound1.17 available [ =stable ]
72 collectd-python3 available [ =stable ]
+ Note on end-of-support. Use 'info' subcommand.
[ec2-user@ip-172-31-33-12 ~]$
```

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

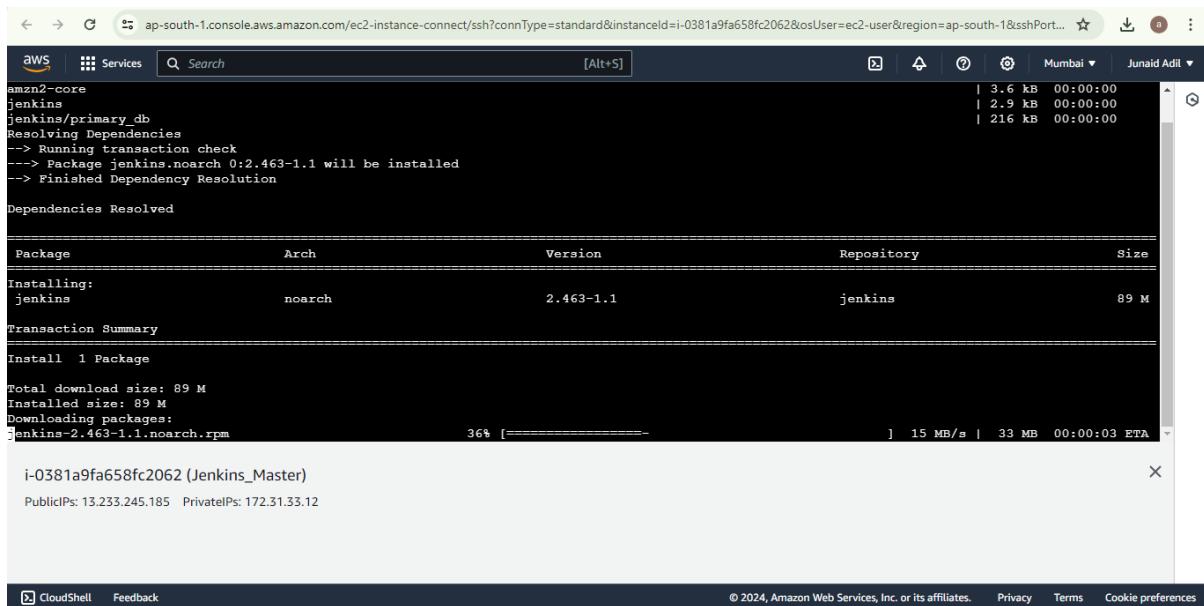
Step-8: Install Jenkins using command “**`sudo yum install jenkins -y`**”



```
[ec2-user@ip-172-31-33-12 ~]$ sudo yum install jenkins -y
```

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



```
amzn2-core
jenkins
jenkins/primary_db
Resolving Dependencies
--> Running transaction check
--> Package jenkins.noarch 0:2.463-1.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

Package           Arch      Version       Repository      Size
Installing:
jenkins          noarch   2.463-1.1   jenkins        89 M

Transaction Summary
Install 1 Package

Total download size: 89 M
Installed size: 89 M
Downloading packages:
jenkins-2.463-1.1.noarch.rpm          36% [=====] 15 MB/s | 33 MB 00:00:03 ETA
```

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows a terminal window in the AWS CloudShell interface. The user is installing the Jenkins package via RPM. The terminal output is as follows:

```
Installing:
jenkins           noarch        2.463-1.1          jenkins          89 M
Transaction Summary
Install 1 Package

Total download size: 89 M
Installed size: 89 M
Downloading packages:
jenkins-2.463-1.1.noarch.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.463-1.1.noarch
  Verifying  : jenkins-2.463-1.1.noarch
Installed:
jenkins.noarch 0:2.463-1.1
Complete!
[ec2-user@ip-172-31-33-12 ~]$
```

At the bottom of the terminal, it says "Complete!" followed by the command entered: "[ec2-user@ip-172-31-33-12 ~]\$ sudo systemctl enable Jenkins".

Step-9: Enable the Jenkins using command “ sudo systemctl enable Jenkins ”

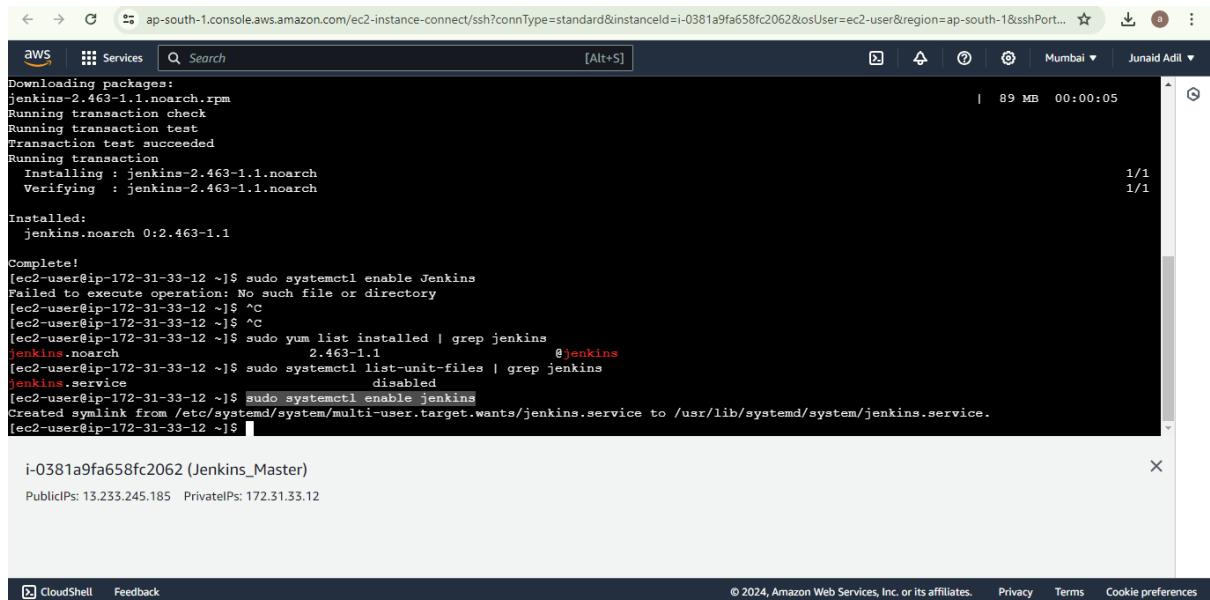
The screenshot shows a terminal window in the AWS CloudShell interface. The user has run the command "sudo systemctl enable Jenkins" after the installation was completed. The terminal output is as follows:

```
Installing:
jenkins           noarch        2.463-1.1          jenkins          89 M
Transaction Summary
Install 1 Package

Total download size: 89 M
Installed size: 89 M
Downloading packages:
jenkins-2.463-1.1.noarch.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.463-1.1.noarch
  Verifying  : jenkins-2.463-1.1.noarch
Installed:
jenkins.noarch 0:2.463-1.1
Complete!
[ec2-user@ip-172-31-33-12 ~]$ sudo systemctl enable Jenkins
```

At the bottom of the terminal, it says "Complete!" followed by the command entered: "[ec2-user@ip-172-31-33-12 ~]\$ sudo systemctl enable Jenkins".

I Was facing some error while enabling the jenkins. So tried to check the list and rerun the command to enable the jenkins. (not sure but when I copy pasted the command i got an error. But when tried manually writing the command then it worked.)



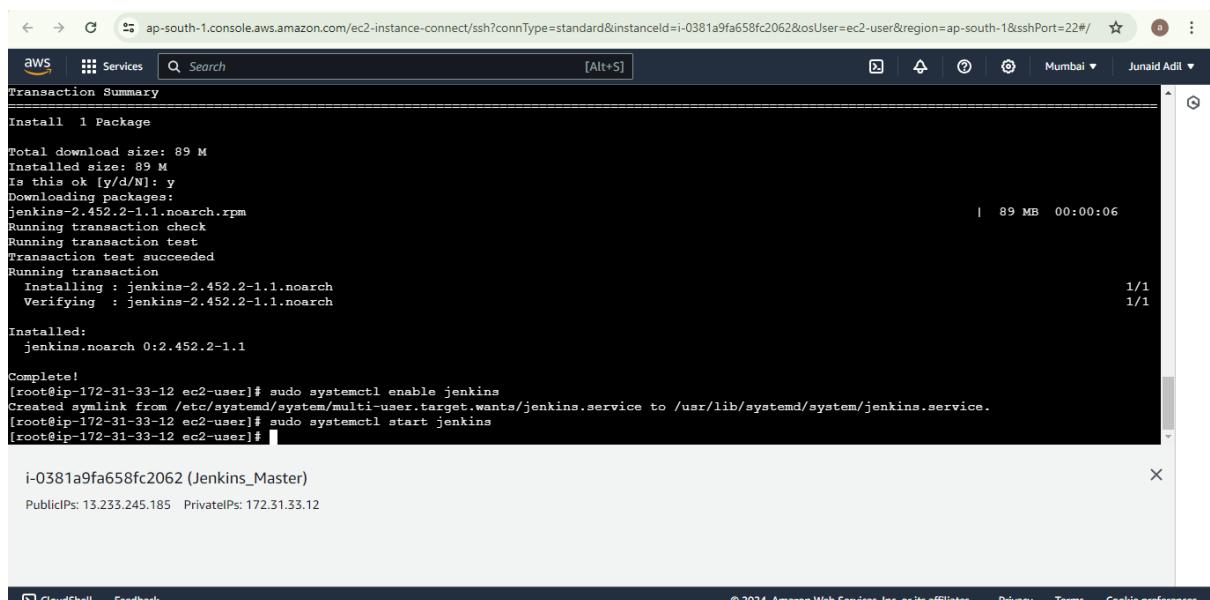
The screenshot shows a terminal window in the AWS CloudShell interface. The user has run several commands to install Jenkins:

```
aws Services Search [Alt+S] Mumbai Junaid Adil
AWS
Download packages:
jenkins-2.463-1.1.noarch.rpm
Running transaction check
Running transaction test
transaction test succeeded
Running transaction
  Installing : jenkins-2.463-1.1.noarch
  Verifying  : jenkins-2.463-1.1.noarch
Installed:
  jenkins.noarch 0:2.463-1.1

Complete!
[ec2-user@ip-172-31-33-12 ~]$ sudo systemctl enable Jenkins
Failed to execute operation: No such file or directory
[ec2-user@ip-172-31-33-12 ~]$ " "
[ec2-user@ip-172-31-33-12 ~]$ " "
[ec2-user@ip-172-31-33-12 ~]$ sudo yum list installed | grep jenkins
jenkins.noarch      2.463-1.1
[ec2-user@ip-172-31-33-12 ~]$ sudo systemctl list-unit-files | grep jenkins
jenkins.service          disabled
[ec2-user@ip-172-31-33-12 ~]$ sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[ec2-user@ip-172-31-33-12 ~]$ 
```

At the bottom, the user is identified as "Junaid Adil" and the instance ID is listed as "i-0381a9fa658fc2062 (Jenkins_Master)".

Step-10: Start Jenkins using command “**sudo systemctl start Jenkins**”



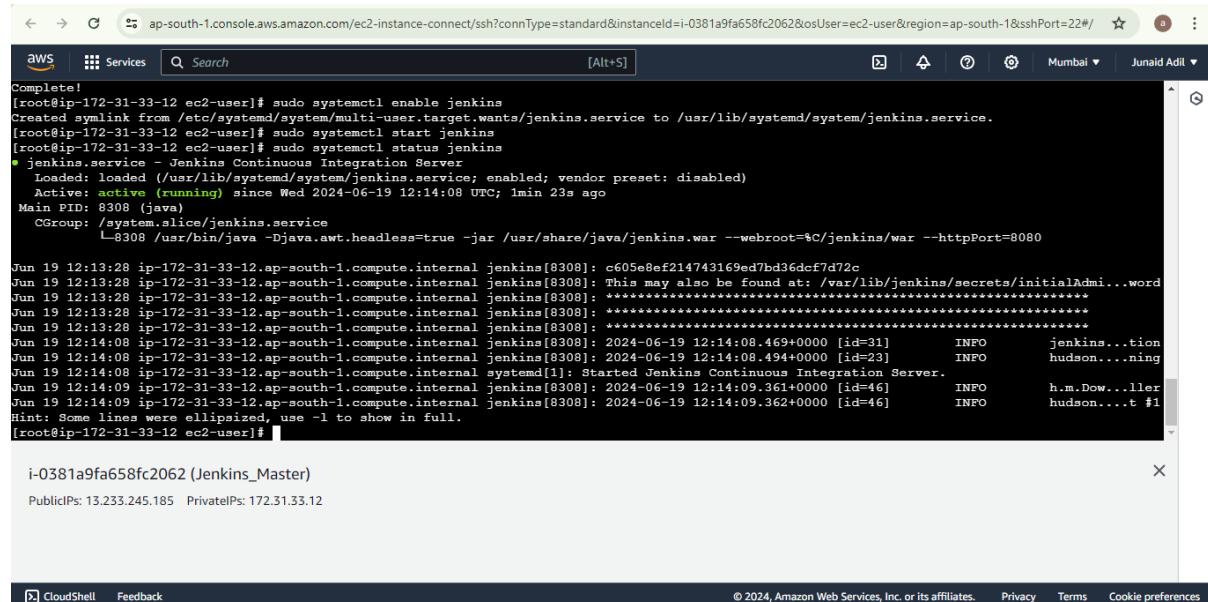
The screenshot shows a terminal window in the AWS CloudShell interface. The user has run commands to start the Jenkins service:

```
aws Services Search [Alt+S] Mumbai Junaid Adil
AWS
Transaction Summary
Install 1 Package
Total download size: 89 M
Installed size: 89 M
Is this ok [y/d/N]: y
Downloading packages:
jenkins-2.452.2-1.1.noarch.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.452.2-1.1.noarch
  Verifying  : jenkins-2.452.2-1.1.noarch
Installed:
  jenkins.noarch 0:2.452.2-1.1

Complete!
[root@ip-172-31-33-12 ec2-user]# sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[root@ip-172-31-33-12 ec2-user]# sudo systemctl start jenkins
[root@ip-172-31-33-12 ec2-user]# 
```

At the bottom, the user is identified as "Junaid Adil" and the instance ID is listed as "i-0381a9fa658fc2062 (Jenkins_Master)".

Step-11: We can check the status of the Jenkins service using command “`sudo systemctl status jenkins`”



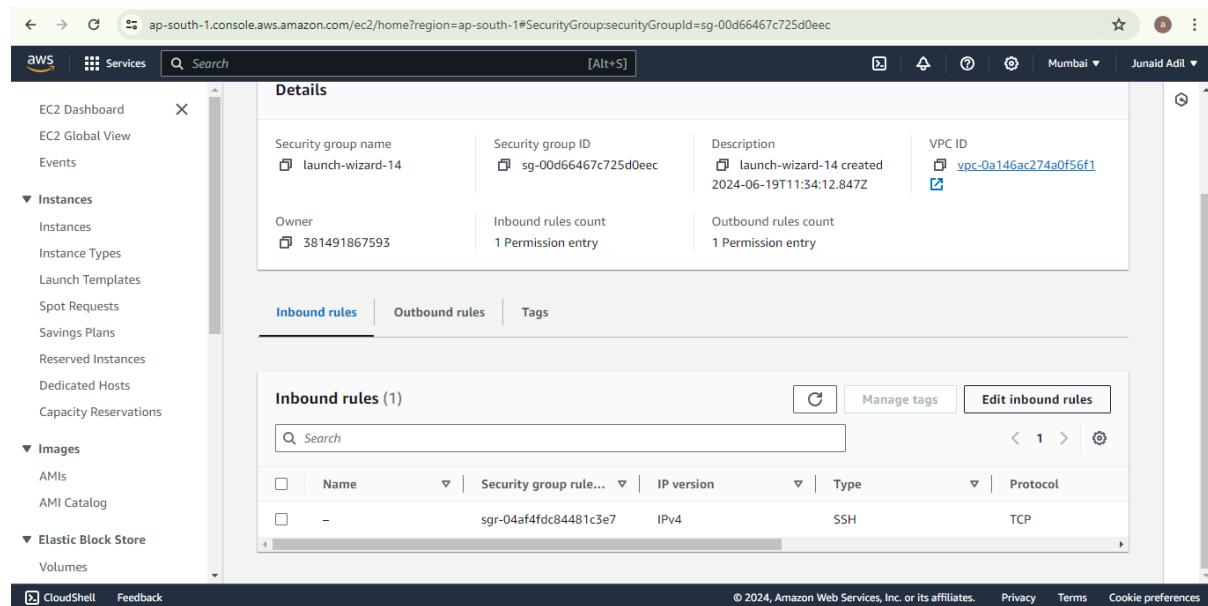
```
Complete!
[root@ip-172-31-33-12 ec2-user]# sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[root@ip-172-31-33-12 ec2-user]# sudo systemctl start jenkins
[root@ip-172-31-33-12 ec2-user]# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
     Active: active (running) since Wed 2024-06-19 12:14:08 UTC; 1min 23s ago
       Main PID: 8308 (java)
      CGroup: /system.slice/jenkins.service
             └─8308 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Jun 19 12:13:28 ip-172-31-33-12.ap-south-1.compute.internal jenkins[8308]: c605e8ef214743169ed7bd36dcf7d72c
Jun 19 12:13:28 ip-172-31-33-12.ap-south-1.compute.internal jenkins[8308]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Jun 19 12:13:28 ip-172-31-33-12.ap-south-1.compute.internal jenkins[8308]: ****
Jun 19 12:13:28 ip-172-31-33-12.ap-south-1.compute.internal jenkins[8308]: 2024-06-19 12:14:08.469+0000 [id=31]           INFO    jenkins...tion
Jun 19 12:14:08 ip-172-31-33-12.ap-south-1.compute.internal jenkins[8308]: 2024-06-19 12:14:08.494+0000 [id=23]           INFO    hudson...ning
Jun 19 12:14:08 ip-172-31-33-12.ap-south-1.compute.internal systemd[1]: Started Jenkins Continuous Integration Server.
Jun 19 12:14:09 ip-172-31-33-12.ap-south-1.compute.internal jenkins[8308]: 2024-06-19 12:14:09.361+0000 [id=46]           INFO    h.m.Dow...ller
Jun 19 12:14:09 ip-172-31-33-12.ap-south-1.compute.internal jenkins[8308]: 2024-06-19 12:14:09.362+0000 [id=46]           INFO    hudson...t #1
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-33-12 ec2-user]#
```

i-0381a9fa658fc2062 (Jenkins_Master)
PublicIPs: 13.233.245.185 PrivateIPs: 172.31.33.12

We can see Jenkins is active, in running status

Step-12: As Jenkins run on default port: 8080, we need to enable/give access to port number 8080 in security groups → Inbound rules



EC2 Dashboard EC2 Global View Events

Instances

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations

Images

- AMIs
- AMI Catalog

Elastic Block Store

- Volumes

CloudShell Feedback

Details

Security group name	launch-wizard-14	Security group ID	sg-00d66467c725d0eec	Description	launch-wizard-14 created 2024-06-19T11:34:12.847Z	VPC ID	vpc-0a146ac274a0f56f1
Owner	381491867593	Inbound rules count	1 Permission entry	Outbound rules count	1 Permission entry		

Inbound rules | Outbound rules | Tags

Inbound rules (1)

Name	Security group rule...	IP version	Type	Protocol
-	sgr-04af4fdc84481c3e7	IPv4	SSH	TCP

Manage tags Edit inbound rules

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

Inbound rules Info

Security group rule ID	Type <small>Info</small>	Protocol	Port range	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-04af4fdc84481c3e7	SSH <small>Info</small>	TCP	22	Cust... <small>Info</small>	<input type="text"/> 0.0.0.0/0 <small>X</small>
-	Custom TCP <small>Info</small>	TCP	8080	Any... <small>Info</small>	<input type="text"/> 0.0.0.0/0 <small>X</small>

[Add rule](#)

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Preview changes](#) [Save rules](#)

[CloudShell](#) [Feedback](#) © 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Instances (1/2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
Jenkins_Master	i-0381a9fa658fc2062	Running <small>View details</small> <small>View metrics</small>	t2.micro	2/2 checks passed <small>View alarms</small> +	ap-south1	
Jenkins_Slave	i-04ab5ef9c024b3590	Running <small>View details</small> <small>View metrics</small>	t2.micro	2/2 checks passed <small>View alarms</small> +	ap-south1	

[Launch instances](#)

[Filter rules](#)

Name	Security group rule ID	Port range	Protocol	Source
-	sgr-04af4fdc84481c3e7	22	TCP	0.0.0.0/0
-	sgr-02d4f7d5355d07b53	8080	TCP	0.0.0.0/0

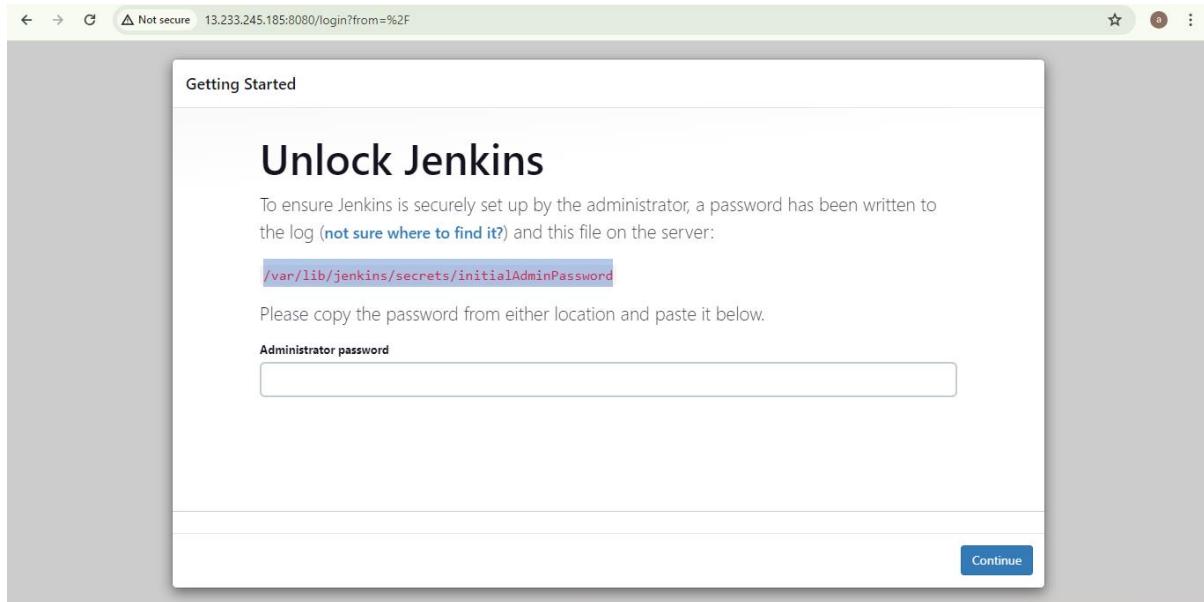
[Outbound rules](#)

[Filter rules](#)

[CloudShell](#) [Feedback](#) © 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

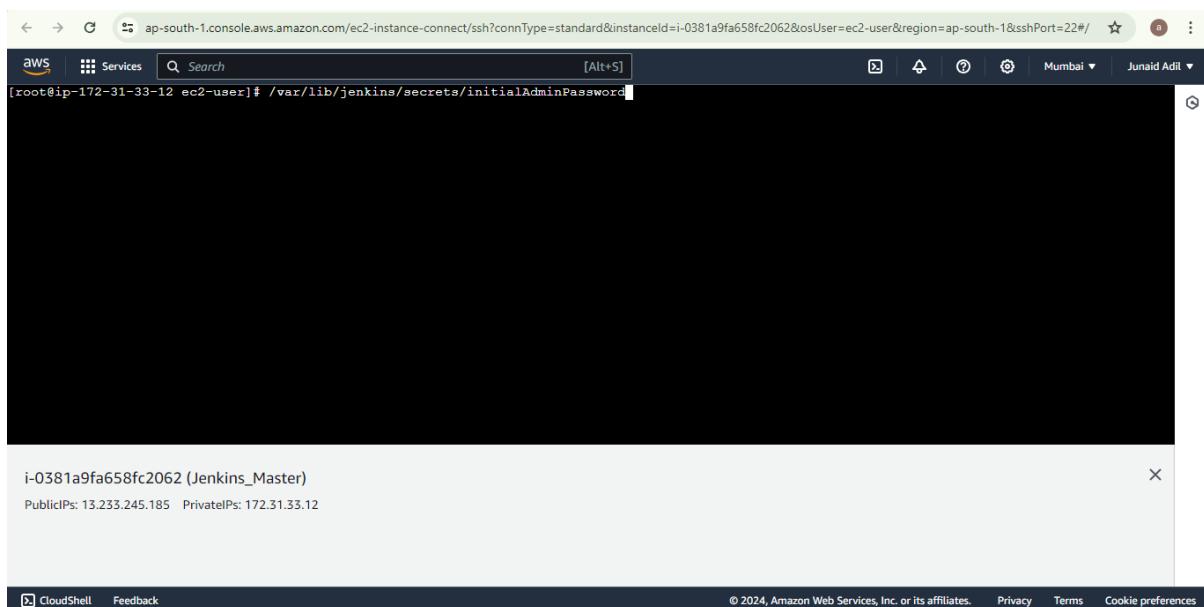
Step-13: Copy the public IP of the server and add Jenkins port number along with IP.

“ **13.233.245.185:8080** ” and paste it in browser

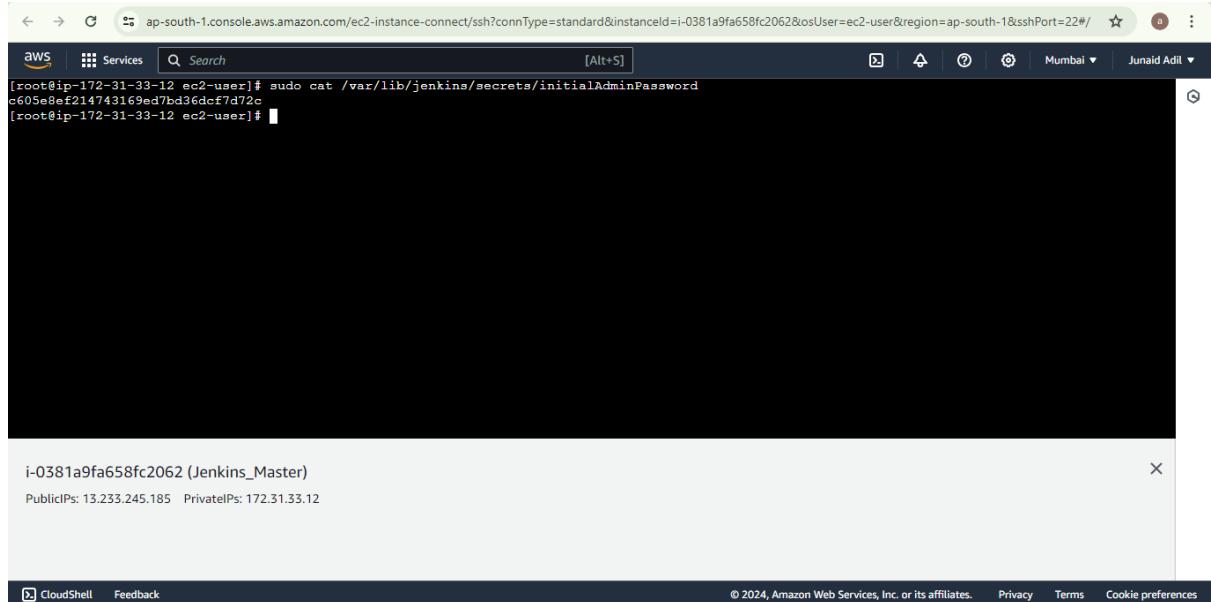


Step-14: Now copy the path which is in red colour and run in the server as

“ **sudo cat /var/lib/jenkins/secrets/initialAdminPassword** ”



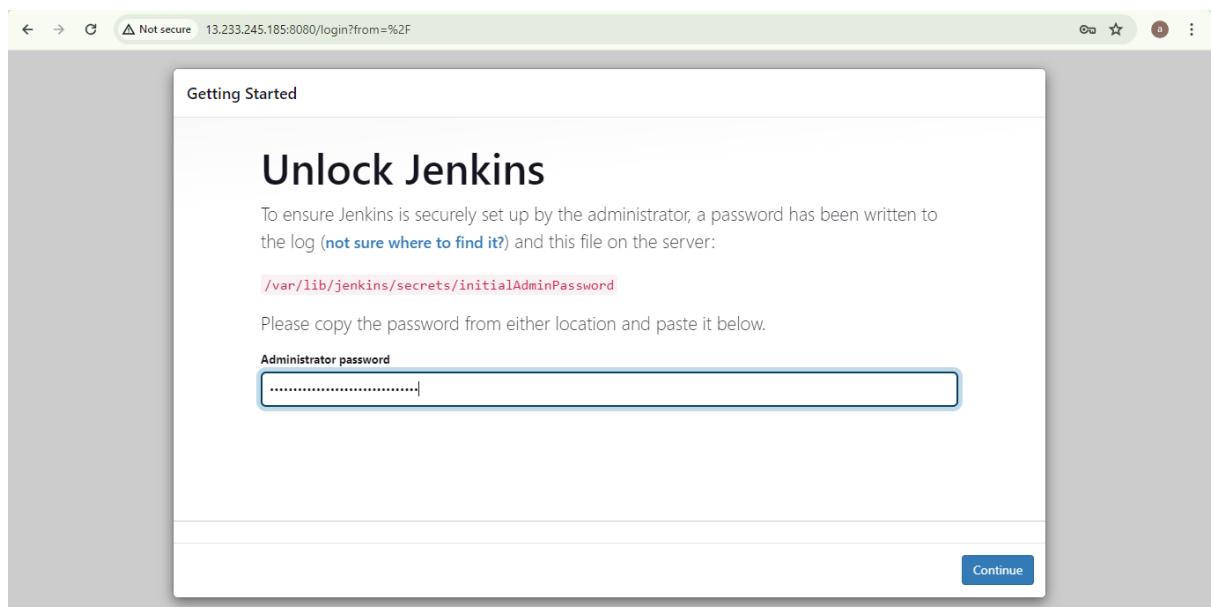
Step-15: Paste the command “ `sudo cat /var/lib/jenkins/secrets/initialAdminPassword` ” in server



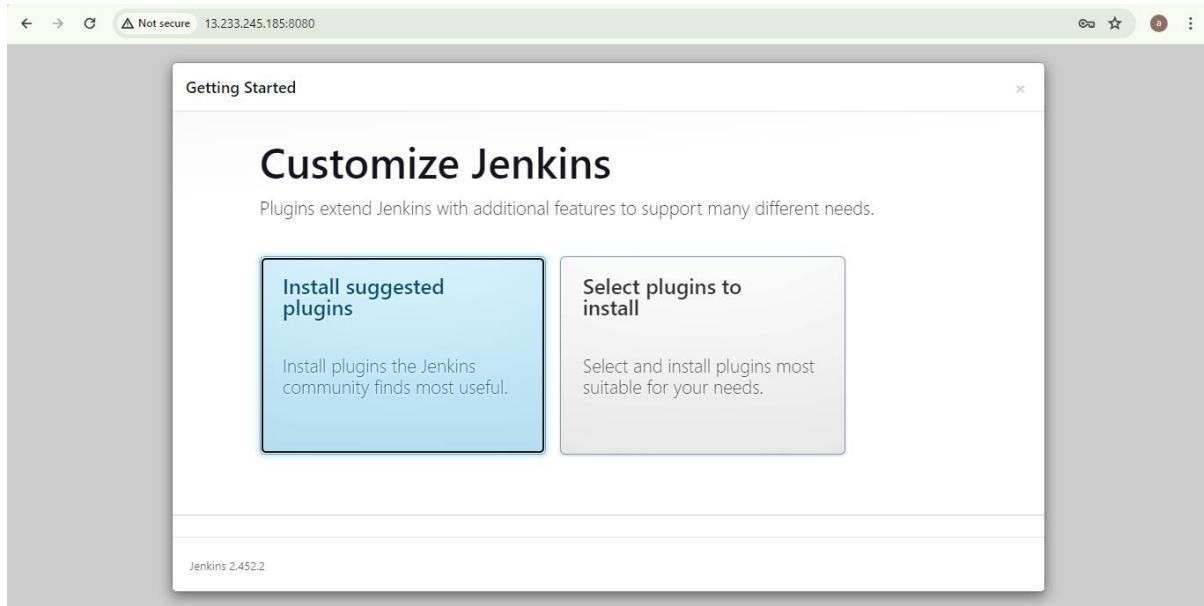
A screenshot of the AWS CloudShell interface. The terminal window shows the command `sudo cat /var/lib/jenkins/secrets/initialAdminPassword` being run, which outputs the password `c605e8ef214743169ed7bd36dcf7d72c`. The AWS logo and services navigation bar are visible at the top. Below the terminal, the instance details are shown: i-0381a9fa658fc2062 (Jenkins_Master), Public IPs: 13.233.245.185, Private IPs: 172.31.33.12. The bottom of the screen includes CloudShell, Feedback, and cookie preference links.

Step-16: Copy the password which generated in server

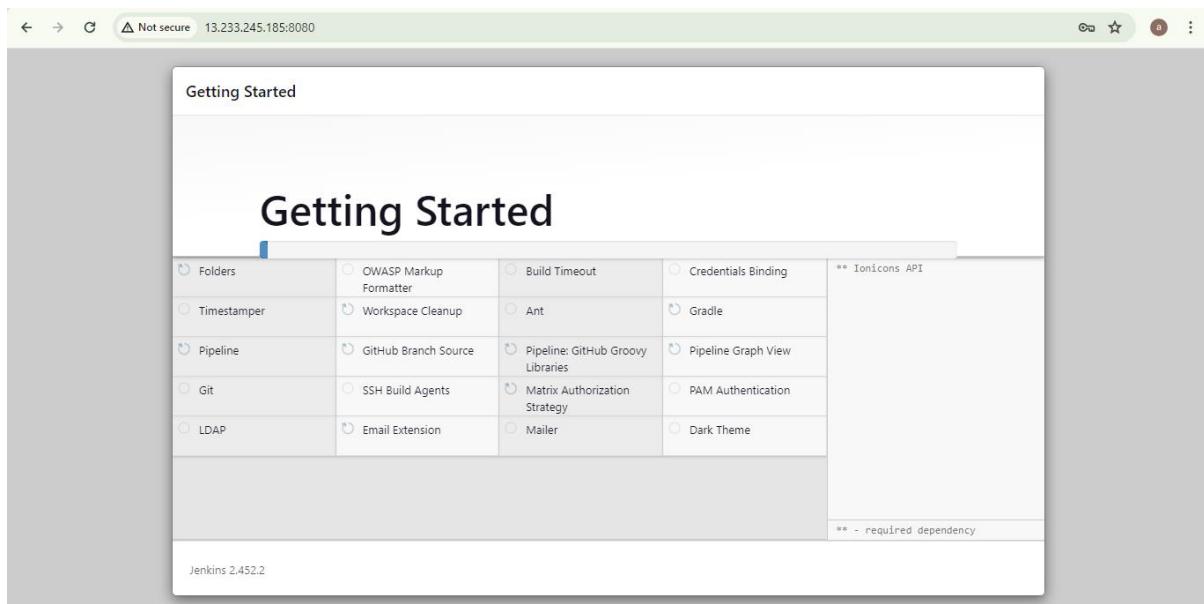
“`c605e8ef214743169ed7bd36dcf7d72c`” and paste it in the Jenkins page Administrator password field → Continue



Step-17: select Install suggested plugins



We can see all suggested plugins getting installed by default



Step-18: create username and give password

Getting Started

Create First Admin User

Username

Password

Confirm password

Full name

Jenkins 2.452.2

Skip and continue as admin

Save and Continue

Getting Started

Create First Admin User

Username

junaidadadil

Password

Confirm password

Full name

junaidadadil

E-mail address

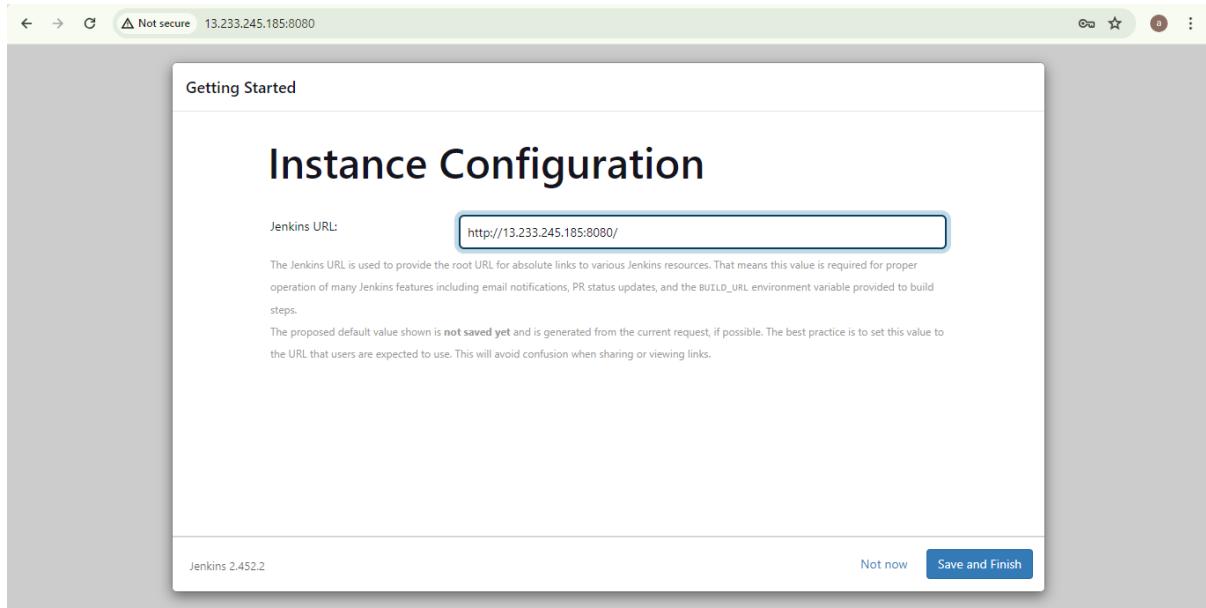
junaidadadil459@gmail.com

Jenkins 2.452.2

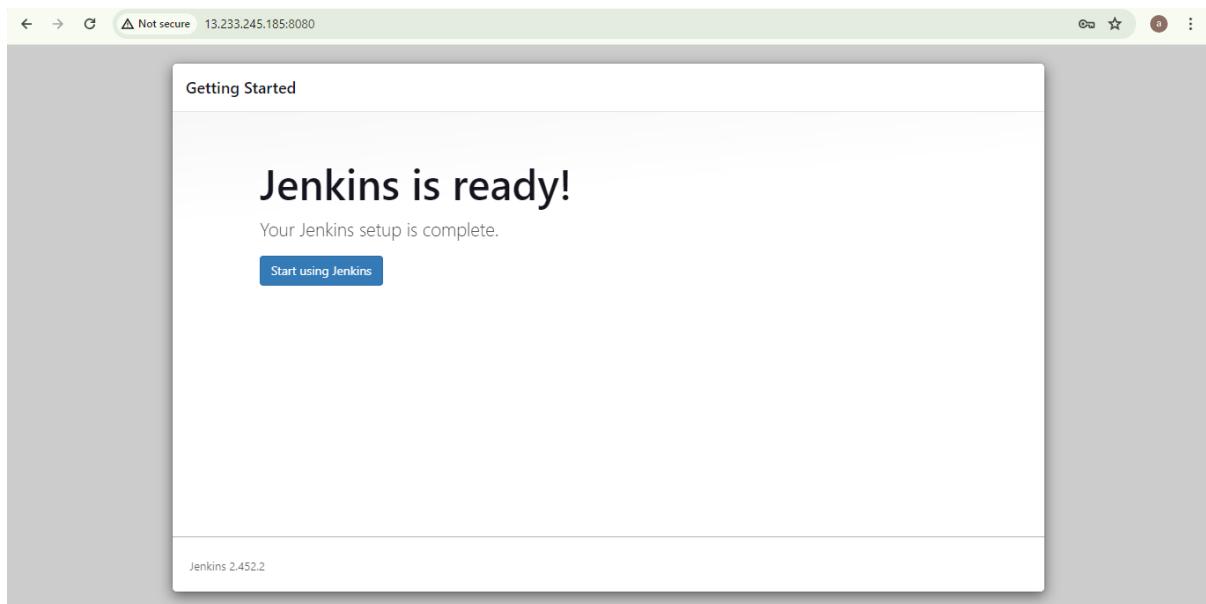
Skip and continue as admin

Save and Continue

We get Jenkins URL.



Now we can start using Jenkins



The screenshot shows the Jenkins dashboard at the URL 13.233.245.185:8080. The top navigation bar includes a star icon, a refresh button, and the URL. On the right, there are icons for notifications, security, and user 'junaidadil'. A dropdown menu for 'Junaid Adil' and a 'log out' link are also present.

The left sidebar contains links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below these are two sections: 'Build Queue' (which says 'No builds in the queue.') and 'Build Executor Status' (which lists 1 Idle and 2 Idle). The main content area features a 'Welcome to Jenkins!' message, a 'Start building your software project' call-to-action, and a 'Set up a distributed build' section with links to 'Create a job', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.

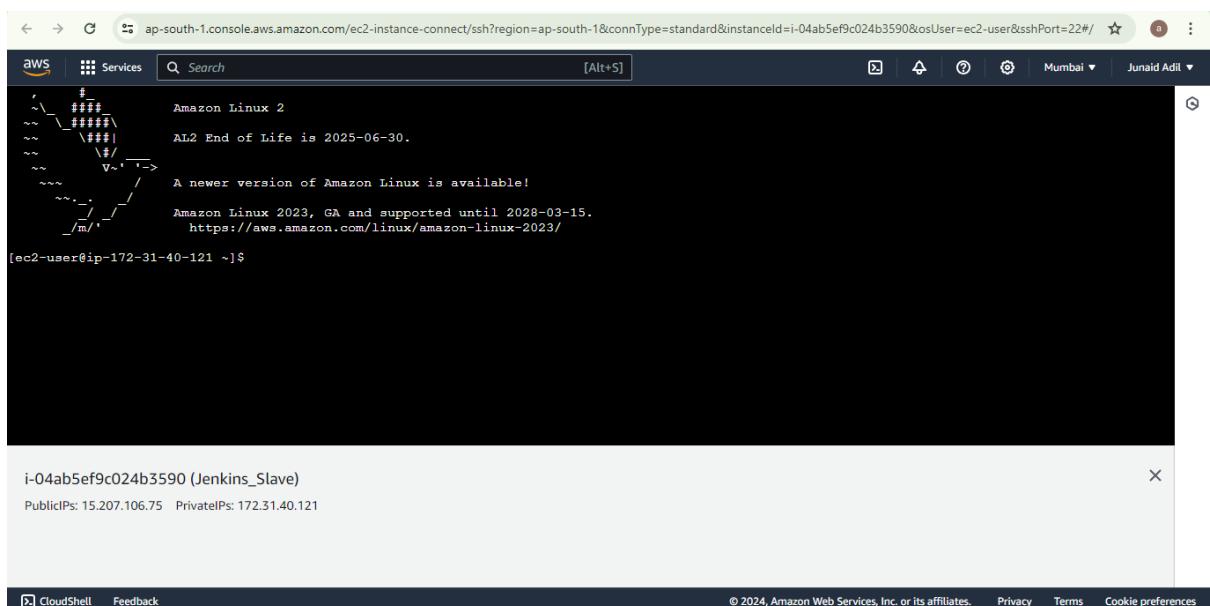
Step-19: Go to Jenkins_Slave Instance and connect

The screenshot shows the AWS EC2 Instances page at the URL <https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:instanceState=running>. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes).

The main content displays a table of running instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
Jenkins_Master	i-0381a9fa658fc2062	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1
<input checked="" type="checkbox"/> Jenkins_Slave	i-04ab5ef9c024b3590	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1

A modal window for the Jenkins_Slave instance (i-04ab5ef9c024b3590) is open, showing the security group rules tab. It indicates that security group rules are being loaded.

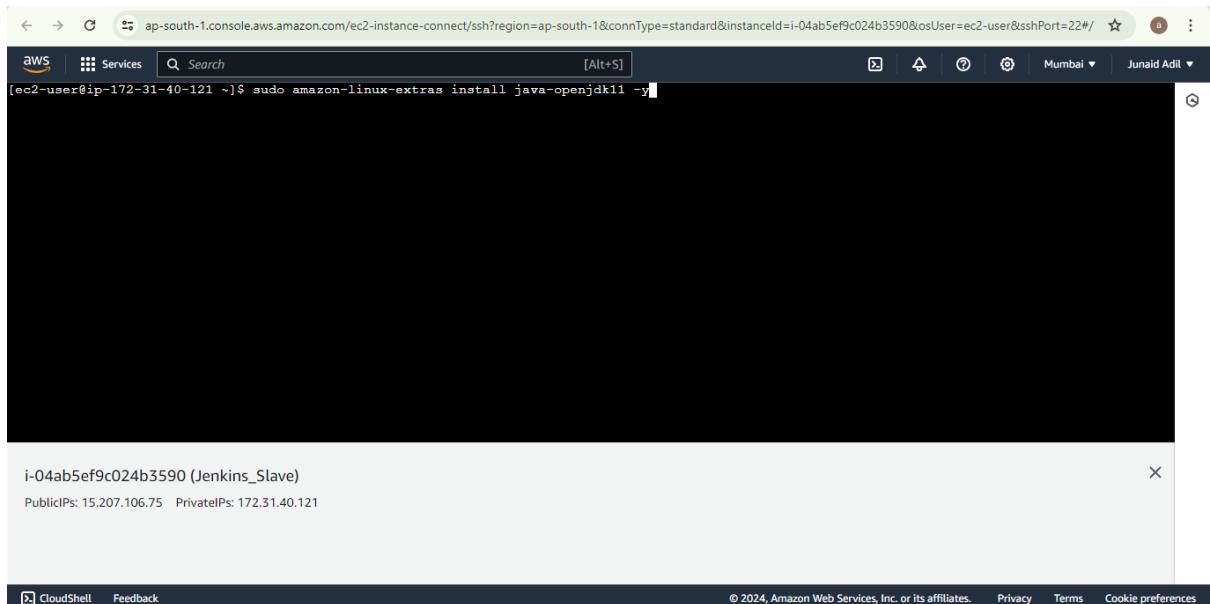


The screenshot shows a terminal window in the AWS CloudShell interface. The terminal output displays a notice about the end-of-life for Amazon Linux 2, dated June 30, 2025. It encourages users to upgrade to Amazon Linux 2023, which is supported until March 15, 2028. A link to the documentation is provided.

```
Amazon Linux 2
AL2 End of Life is 2025-06-30.
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-40-121 ~]$
```

i-04ab5ef9c024b3590 (Jenkins_Slave)
PublicIPs: 15.207.106.75 PrivateIPs: 172.31.40.121

Step-20: Install java in Jenkins_slave instance using command “ **sudo amazon-linux-extras install java-openjdk11 -y** ”



The screenshot shows a terminal window in the AWS CloudShell interface. The user has run the command "sudo amazon-linux-extras install java-openjdk11 -y" to install Java 11. The command is visible at the bottom of the terminal window.

```
[ec2-user@ip-172-31-40-121 ~]$ sudo amazon-linux-extras install java-openjdk11 -y
```

i-04ab5ef9c024b3590 (Jenkins_Slave)
PublicIPs: 15.207.106.75 PrivateIPs: 172.31.40.121

```
← → G ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-04ab5ef9c024b3590&osUser=ec2-user&sshPort=22#/  
AWS Services Search [Alt+S] [ ] [ ] Mumbai Junaid Adil  
  
Installed:  
java-11-openjdk.x86_64 1:11.0.23.0.9-2.amzn2.0.1  
  
Dependency Installed:  
alsa-lib.x86_64 0:1.1.4.1-2.amzn2  
copy-jdk-configs.noarch 0:3.3-10.amzn2  
dejavu-fon...-common.noarch 0:2.33-6.amzn2  
fontconfig.x86_64 0:2.13.0-4.3.amzn2  
glib2.x86_64 0:4.1.6-9.amzn2.0.2  
harfbuzz.x86_64 0:1.7.5-2.amzn2  
javapackages-tools.noarch 0:3.4.1-11.amzn2  
libSM.x86_64 0:1.2.2-2.amzn2.0.2  
libX11-common.noarch 0:1.6.7-3.amzn2.0.5  
libXext.x86_64 0:1.3.3-3.amzn2.0.2  
libXrender.x86_64 0:0.9.10-1.amzn2.0.2  
libfontenc.x86_64 0:1.1.3-3.amzn2.0.2  
libXslt.x86_64 0:1.1.28-6.amzn2  
log4j-cve-2021-44228-hotpatch.noarch 0:1.3-7.amzn2  
python-javapackages.noarch 0:3.4.1-11.amzn2  
ttmkfdir.x86_64 0:3.0.9-42.amzn2.0.2  
xorg-x11-font-utils.x86_64 1:7.5-21.amzn2  
avahi-libs.x86_64 0:0.6.31-20.amzn2.0.5  
cups-libs.x86_64 1:1.6.3-51.amzn2.0.4  
dejavu-sans-fonts.noarch 0:2.33-6.amzn2  
fontpackages-fs...-file...-noarch 0:1.44-8.amzn2  
graphite2.x86_64 0:1.3.10-1.amzn2.0.2  
java-11-openjdk-headless.x86_64 1:11.0.23.0.9-2.amzn2.0.1  
libICE.x86_64 0:1.0.9-9.amzn2.0.2  
libX11.x86_64 0:1.6.7-3.amzn2.0.5  
libXau.x86_64 0:1.0.8-2.1.amzn2.0.2  
libXi.x86_64 0:1.7.9-1.amzn2.0.2  
libXtst.x86_64 0:1.2.3-1.amzn2.0.2  
libxcb.x86_64 0:1.12-1.amzn2.0.2  
lksctp-tools.x86_64 0:1.0.17-2.amzn2.0.2  
pcsc-lite-libs.x86_64 0:1.8.8-7.amzn2  
python-lxml.x86_64 0:3.2.1-4.amzn2.0.5  
tzdata-java.noarch 0:2024a-1.amzn2.0.1  
xorg-x11-fonts-Type1.noarch 0:7.5-9.amzn2  
xorg-x11-fonts-Type1.noarch 0:7.5-9.amzn2  
  
Complete!  
  
i-04ab5ef9c024b3590 (Jenkins_Slave)  
PublicIPs: 15.207.106.75 PrivateIPs: 172.31.40.121  
  
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

```
← → G ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-04ab5ef9c024b3590&osUser=ec2-user&sshPort=22#/  
AWS Services Search [Alt+S] [ ] [ ] Mumbai Junaid Adil  
  
48 R4 available [ =stable ]  
49 kernel-5.4 available [ =stable ]  
50 selinux-ng available [ =stable ]  
51 tomcat9 available [ =stable ]  
52 unbound1.13 available [ =stable ]  
53 mariadb10.5 available [ =stable ]  
54 kernel-5.10=latest enabled [ =stable ]  
55 redis6 available [ =stable ]  
56 tpostgresql12 available [ =stable ]  
57 tpostgresql13 available [ =stable ]  
58 mock2 available [ =stable ]  
59 dnsmasq2.85 available [ =stable ]  
60 kernel-5.15 available [ =stable ]  
61 tpostgresql14 available [ =stable ]  
62 firefox available [ =stable ]  
63 lustre available [ =stable ]  
64 tphp8.1 available [ =stable ]  
65 awacs11 available [ =stable ]  
66 tphp8.2 available [ =stable ]  
67 dnsmasq available [ =stable ]  
68 unbound1.17 available [ =stable ]  
69 collectd-python3 available [ =stable ]  
70 Note on end-of-support. Use 'info' subcommand.  
[ec2-user@ip-172-31-40-121 ~]$  
  
i-04ab5ef9c024b3590 (Jenkins_Slave)  
PublicIPs: 15.207.106.75 PrivateIPs: 172.31.40.121  
  
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Step-21: Run below commands to create keygen and convert to .PEM file—

“ssh-keygen -t rsa -b 4096 -f my-slave-node-key -C "jenkins_slave_node"” and then run command

“openssl rsa -in my-slave-node-key -outform pem -out my-slave-node-key.pem” to convert the key file to .pem file

```
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-04ab5ef9c024b3590&osUser=ec2-user&sshPort=22#/
[ec2-user@ip-172-31-40-121 ~]$ ssh-keygen -t rsa -b 4096 -f my-slave-node-key -C "jenkins_slave_node"
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in my-slave-node-key.
Your public key has been saved in my-slave-node-key.pub.
The key fingerprint is:
SHA256:3jRftd8CDsLRMhmxxI2y2rKXpep73V8PoUpLusUSyQ jenkins_slave_node
The key's randomart image is:
+---[RSA 4096]---+
| .o
| *+ . o
| o oE.o o + .
| ..+ + o o
| ...8 * +
| o= X = +
| ..+ * = + o ..
| .oo + . o .
| .+o..o
+---[SHA256]---+
[ec2-user@ip-172-31-40-121 ~]$ openssl rsa -in my-slave-node-key -outform pem -out my-slave-node-key.pem
writing RSA key
[ec2-user@ip-172-31-40-121 ~]$ 
```

i-04ab5ef9c024b3590 (Jenkins_Slave)
PublicIPs: 15.207.106.75 PrivateIPs: 172.31.40.121

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step-22: Then Restrict the permissions to the file

```
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-04ab5ef9c024b3590&osUser=ec2-user&sshPort=22#/
[ec2-user@ip-172-31-40-121 ~]$ chmod 400 my-slave-node-key
[ec2-user@ip-172-31-40-121 ~]$ chmod 400 my-slave-node-key.pem
[ec2-user@ip-172-31-40-121 ~]$ 
```

i-04ab5ef9c024b3590 (Jenkins_Slave)
PublicIPs: 15.207.106.75 PrivateIPs: 172.31.40.121

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step-23: Now go to Jenkins → manage Jenkins

The screenshot shows the Jenkins dashboard at the URL <http://13.233.245.185:8080>. The title bar says "Not secure". The main header has the Jenkins logo and the text "Dashboard >". A search bar says "Search (CTRL+K)". On the right, there are notifications and a user account "junaidadil". A "log out" button is also present.

The dashboard features a sidebar with links: "+ New Item", "Build History", "Manage Jenkins" (which is currently selected), and "My Views". Below the sidebar, there's a "Build Queue" section showing "No builds in the queue." and a "Build Executor Status" section showing "1 Idle" and "2 Idle".

The central area has a "Welcome to Jenkins!" message: "This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project." It includes a "Start building your software project" button and a "Create a job" button. A "Set up a distributed build" section contains "Set up an agent" and "Configure a cloud" buttons, along with a link to "Learn more about distributed builds".

Step-24: select Nodes

The screenshot shows the "Manage Jenkins" page at the URL <http://13.233.245.185:8080/manage/>. The title bar says "Not secure". The main header has the Jenkins logo and the text "Dashboard > Manage Jenkins". A search bar says "Search settings".

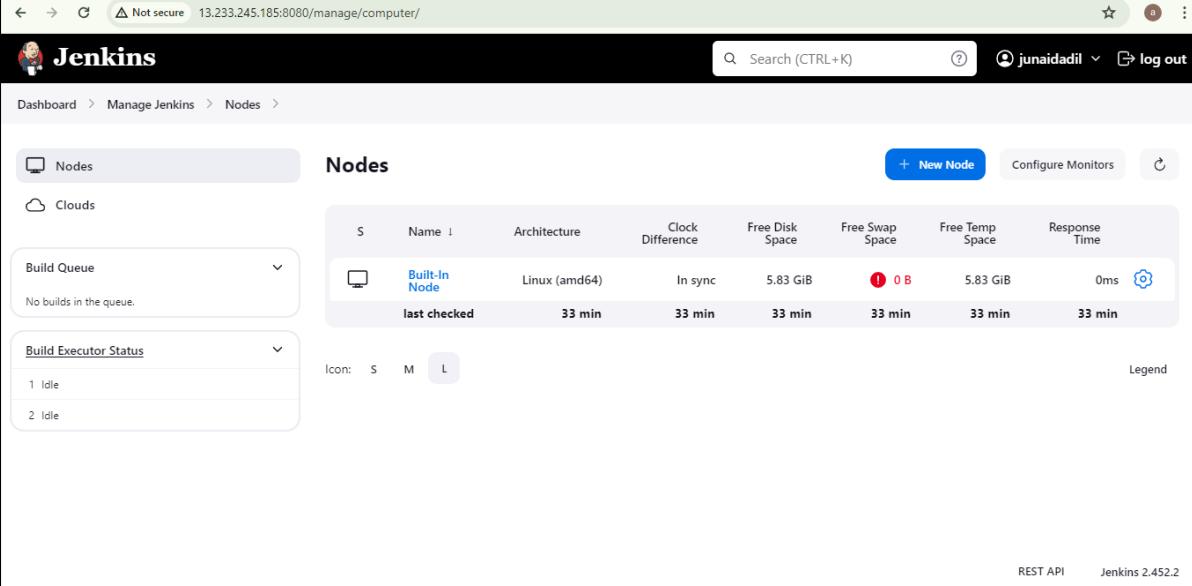
The sidebar on the left is identical to the dashboard, with links for "+ New Item", "Build History", "Manage Jenkins" (selected), and "My Views".

The central area is titled "Manage Jenkins" and "System Configuration". It contains several configuration sections:

- System**: Configure global settings and paths.
- Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on. This section is highlighted with a green border.
- Tools**: Configure tools, their locations and automatic installers.
- Clouds**: Add, remove, and configure cloud instances to provision agents on-demand.
- Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Appearance**: Configure the look and feel of Jenkins.

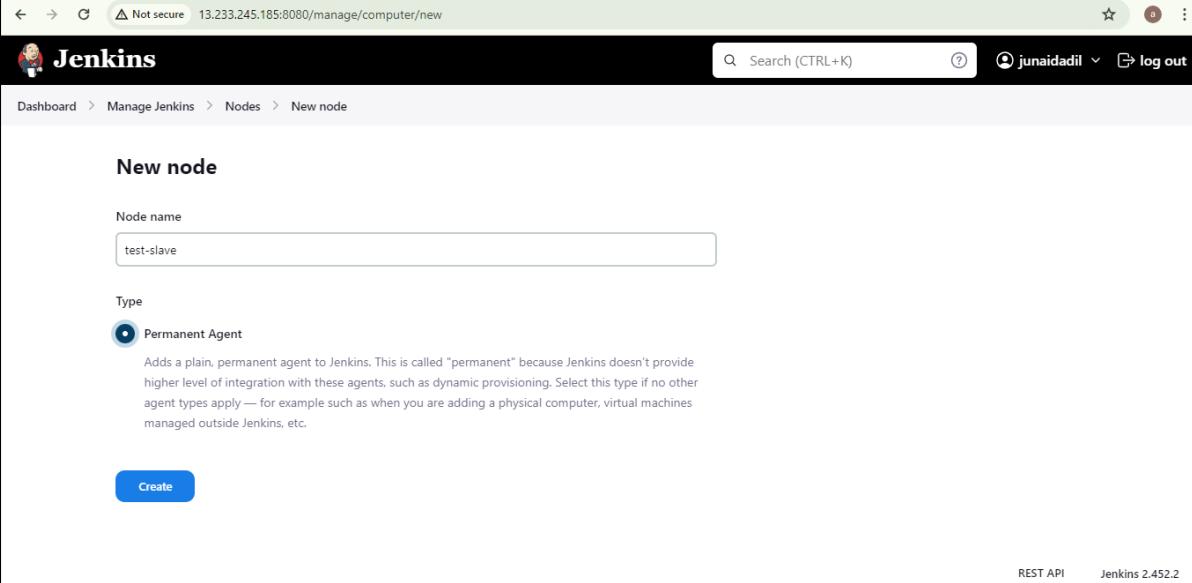
Below these, there are sections for "Security", "Credentials", and "Credential Providers".

Step-25: click on New node



The screenshot shows the Jenkins 'Nodes' page. On the left, there are two tabs: 'Nodes' (which is selected) and 'Clouds'. Below these are two dropdown menus: '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. The table columns are: S (Status), Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. The single node listed is 'Built-In Node' (Name), which is a 'Linux (amd64)' machine. It is marked as 'In sync' with a clock icon. The status bar indicates 'last checked' was 33 min ago. Resource usage is shown as 5.83 GiB free disk space, 5.83 GiB free swap space, and 0ms response time. A red warning icon with '0 B' is present under 'Free Swap Space'. A blue link icon is next to 'Response Time'. At the top right of the table header is a blue button labeled '+ New Node'. To the right of the table is a 'Configure Monitors' link and a refresh icon. At the bottom right of the page are links for 'REST API' and 'Jenkins 2.452.2'.

Step-26: Enter the name and 'check' – Permanent Agent → Create



The screenshot shows the 'New node' configuration page. The title is 'New node'. There are two sections: 'Node name' (containing the value 'test-slave') and 'Type' (with 'Permanent Agent' selected). A detailed description of 'Permanent Agent' is provided: '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.' At the bottom is a blue 'Create' button. The page includes standard navigation and search bars at the top, and links for 'REST API' and 'Jenkins 2.452.2' at the bottom right.

Not secure 13.233.245.185:8080/manage/computer/createItem

Jenkins

Dashboard > Manage Jenkins > Nodes >

Name ?
test-slave

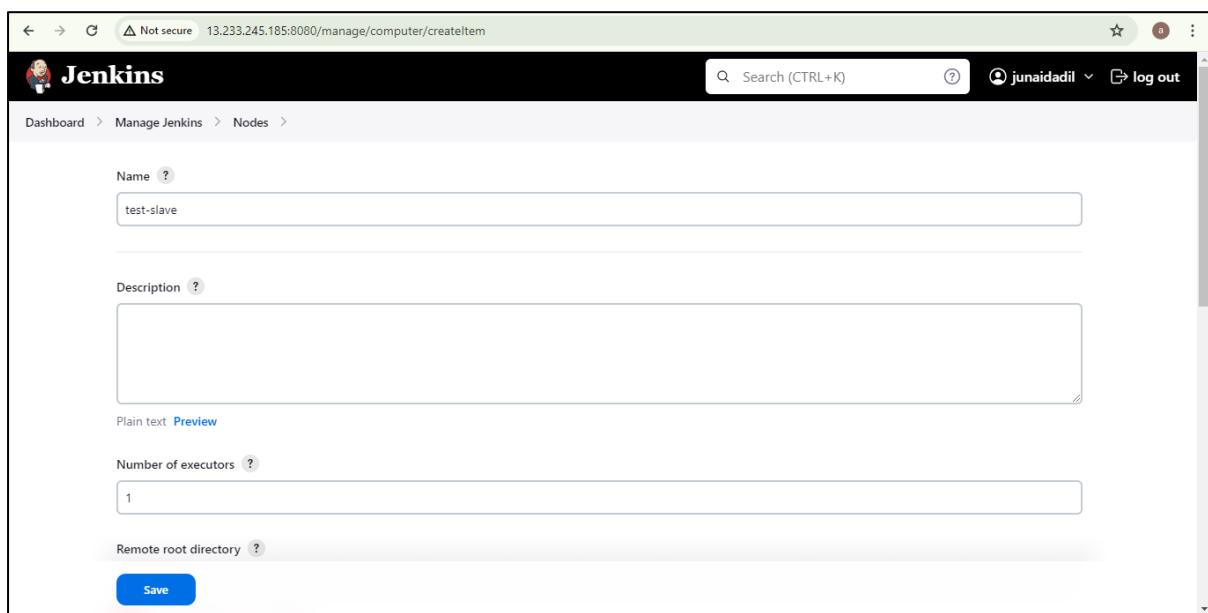
Description ?

Plain text [Preview](#)

Number of executors ?
1

Remote root directory ?

Save



Step-27: Enter the details, Label: Slave1.

Not secure 13.233.245.185:8080/manage/computer/createItem

Dashboard > Manage Jenkins > Nodes >

Remote directory is mandatory

Labels ?
slave1

Usage ?
Only build jobs with label expressions matching this node

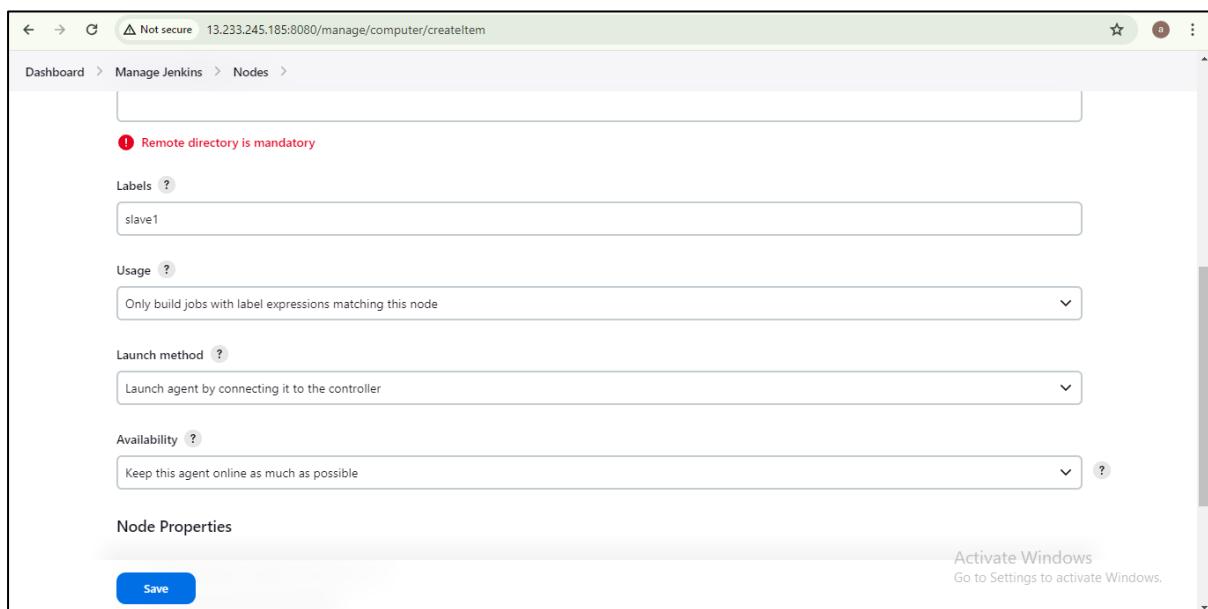
Launch method ?
Launch agent by connecting it to the controller

Availability ?
Keep this agent online as much as possible

Node Properties

Activate Windows
Go to Settings to activate Windows.

Save



Step-28: In Launch method select “ Launch agents via SSH ”

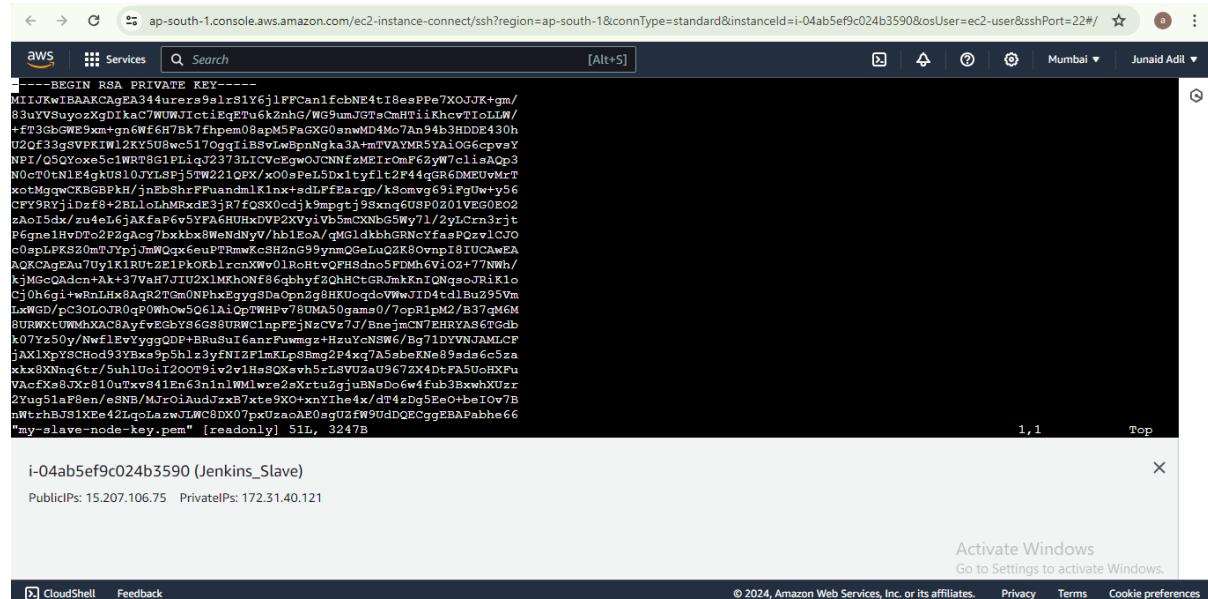
And then enter the details, change Kind field to “ SSH username with private key ” and enter username: ec2-user (using Linux) , in private key field ‘check’ Enter directly, and paste the content of .pem file used for slave node

The screenshot shows the Jenkins 'Add Credentials' dialog. The 'Kind' dropdown is set to 'SSH Username with private key'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'ID' and 'Description' fields are empty. A 'Save' button is visible at the bottom left.

The screenshot shows the Jenkins 'Add Credentials' dialog with more detailed settings. The 'Username' field contains 'ec2-user'. The 'Private Key' section has a radio button selected for 'Enter directly' and a 'Key' input field containing 'No Stored Value'. A blue 'Add' button is visible next to the key field. The 'Passphrase' field is empty. A 'Save' button is visible at the bottom left.

Step-29: Paste the key created from .pem file → Add.

Step: open the created .pem file in the Instance and from there copy the key and paste it in Jenkins credentials provider-private key



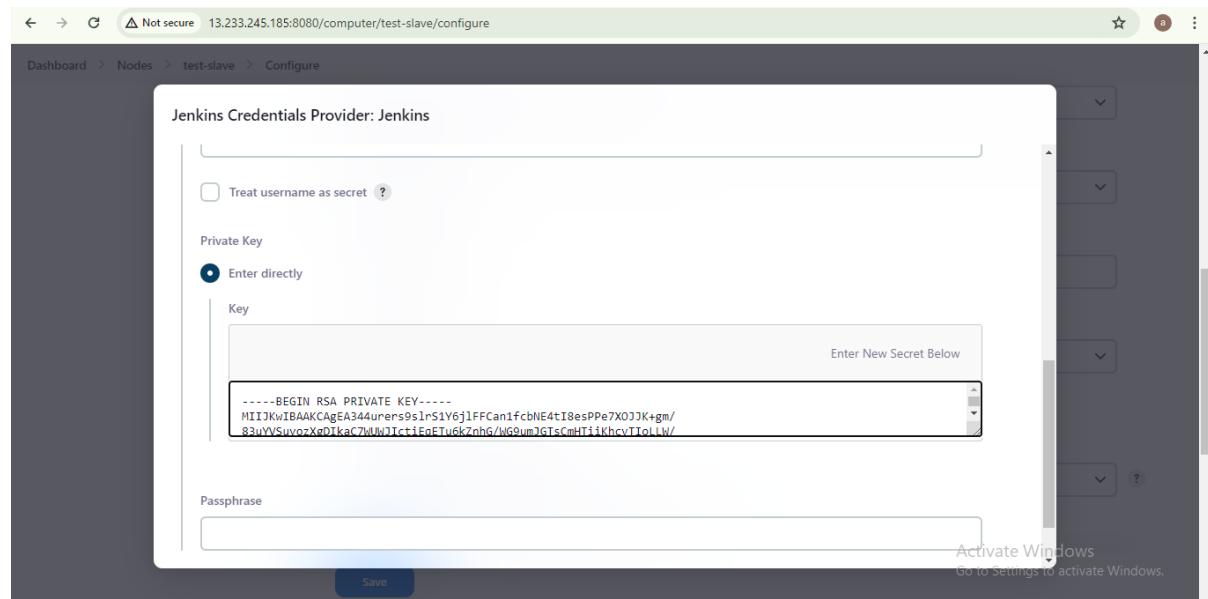
```
--BEGIN RSA PRIVATE KEY--
MIIEKwIBAAKCAgEA344urers9s1rS1Y6j1FFFCan1fcbNE4tI8esPPe7XOJJK+gm/
93uVSuyoczgDkaC7WUWJctiqTuwkzhnG/WG9umJGTscMmTiikhcvTloLW/
+f73GbWE9xm+m=gn6WF6R7Bk7hpem08apM5PaGXG0snwMD4Mo7An94b3HDE430h
U2Qf33gsVPKIWl2K2Y5Uswe517OqgiBSvlwBpnNgka3A+mtVAXMR5YAlOG6cpv8Y
NPI/O5OYoxe5c1WRt8G1PLiqz2373lICVcEgw0J/CNNfzMEIrOmF6zyw7clisAcP3
N0cn0tN1L4pkUS10JYLSPl5TW2210PKXx0oPe15px1tyflt2P44gGR6DMRUvMr7
xotMggwCKBGBBkH/jnEbShrFPuandmlK1nx+sd1PFEarqp/k8omvg69iPgUw+y56
CFY9RXj1dzf8+2BLlchMrxdE3jR7fQSX0cdjk9mpgtj98nxqGU8P0201VEG0B02
zAo15dx/zu4e16jAkFaP6v5YFA6HUHxDVF2XVyiVb5mCXNbGSMy7l/2yLCrn3rjt
P6gneiHvd7ro2P2ZgAcg7bxbsxWeNdnyv/hb1EcA/qMsl1dkbhGRNCyfas0zv1cJO
c0spLPKSZ0mTVYpjJmWCpx6euP7RmwKcSHznmG99ymmGeLaq2x8ovnpI8lUCAwEA
AQK0AgEAu70y1K1RtZE1p1K0kb1rcnXwv0lRoHtvQFI8dnos5FIMh6Vi02+77nhw/
kjMgcAAdcn+Ak+37VaH/JIUZX1MK0NF8qhyfZ0hCTGRUmKniQNqsoURk1o
Cj0h6gi+wRnLhx8q27Gm0NPhxEyggsDaOp2g8HKUoqdovWwJID4t1lBu295Vm
LxWGD/pc3OL0J0RqP0WhOw5061ai1OpTWHFv7UMA50gams0/7opR1pm2/B37qM6M
BURWxtUWMNxAC83ayfEgbYS6GS8URWClnpFFjNzCVz7J/BnejmCN7EHRYAS6TGdb
k07xz5Oy/Nwf1EyYygqQDP+BRuSu16anrFuwmgzzHzuY-N8W6/Bg71dYVNJAMLCF
jAx1XpY8CHcd93XBxs9p5h/z3yfNTZFlmkLpSHmq2P4x7qA5abekNe89da6c5za
xkx8XNng6tr/Suh1Uci12OOt9iv2v1Hs8CXvh5rlSVU2aU9672X4DtFA50UoHXFu
VAcfx8JXr810utxv841En63nlwlWMIwre2xxttu2giuBNSnGwifub33xwhXUzr
2Yug51aPRen/+SN/MJrO1AudJx87xt+e5X0+xnYIhe4x/d74zDg5FeO+betIOr7B
nWtrhBJS1XEE42LqoLazwJLWC8DX07pxUsaoAE0sgU2fW9UdDQEcgEBAPabhe66
"my_slave_node-key.pem" [readonly] S1L, 3247B
```

i-04ab5ef9c024b3590 (Jenkins_Slave)

PublicIPs: 15.207.106.75 PrivateIPs: 172.31.40.121

Activate Windows
Go to Settings to activate Windows.

Step-30: Paste the copied key in Jenkins credentials provider-private key



Step-31: In Host key verification strategy: select ‘Manually trusted key verification strategy’ → in Host: enter private IP Address → save

Not secure 13.233.245.185:8080/manage/computer/createItem

Dashboard > Manage Jenkins > Nodes >

Launch agents via SSH

Host ?
15.207.106.75

Credentials ?
ec2-user
+ Add ▾

Host Key Verification Strategy ?
Known hosts file Verification Strategy
Manually provided key Verification Strategy
Manually trusted key Verification Strategy
Non verifying Verification Strategy

Availability ?
Keep this agent online as much as possible

Activate Windows
Go to Settings to activate Windows.

Save

Step-32: Enter remote root directory → Save.

Not secure 13.233.245.185:8080/computer/test-slave/configure

Dashboard > Nodes > test-slave > Configure

Remote root directory ?
/home/ec2-user/slavenode

Labels ?
slave1

Usage ?
Only build jobs with label expressions matching this node

Launch method ?
Launch agents via SSH

Host ?
15.207.106.75

Activate Windows
Go to Settings to activate Windows.

Save

We can see the created node has started

The screenshot shows the Jenkins interface for managing nodes. On the left, there's a sidebar with sections for 'Build Queue' (empty), 'Build Executor Status' (showing 'Built-In Node' with 1 idle and 2 idle executors), and 'Clouds'. The main area is titled 'Nodes' and lists two entries:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	5.57 GiB	1 0 B	5.57 GiB	0ms
	test-slave	Linux (amd64)	In sync	6.00 GiB	1 0 B	6.00 GiB	28ms

Details for the 'test-slave' node: last checked 6.3 sec, response time 6.3 sec. Below the table are icons for sorting (S, M, L) and a legend. At the bottom right, there are links to activate Windows, REST API, and Jenkins version 2.452.2.

Step-33: Using Freestyle project, create a job – “Testslave1”

The screenshot shows the Jenkins 'New Job' creation page. The 'Item name' field contains 'Testslave1'. Below it, three project types are listed:

- Freestyle project**: Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Pipeline**: Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

At the bottom right, there are links to activate Windows, REST API, and Jenkins version 2.452.2. A blue 'OK' button is visible at the bottom of the configuration section.

Step-34: In Description we can write anything for reference (Its optional)

The screenshot shows the Jenkins configuration interface for a job named 'Testslave1'. The 'General' tab is selected. In the 'Description' field, the text 'To check slave1' is entered. There are several optional checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterised', and 'Throttle builds'. At the bottom are 'Save' and 'Apply' buttons.

Step-35: 'check' – “Restrict where this project can be run” and add the label expression: <slave node which created>

The screenshot shows the Jenkins configuration interface for the same job. The 'Restrict where this project can be run' checkbox is checked. Below it, the 'Label Expression' field contains 'test-slave'. A note below states 'Label test-slave matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.' There is also an 'Advanced' dropdown menu. At the bottom are 'Save' and 'Apply' buttons.

Step-36: In Build steps, select Execute shell and write the script to run → Apply → save.

The screenshot shows the Jenkins configuration interface for a job named 'Testslave1'. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps' (which is selected), and 'Post-build Actions'. The main area is titled 'Build Steps' and contains a single step named 'Execute shell'. The command entered is '#!/bin/bash echo "Hello, This is Junaid"'. Below the command is an 'Advanced' dropdown and a 'Save' button. A message at the bottom right says 'Activate Windows Go to Settings to activate Windows.'

Step-37: click on build now and check the build history for the status of build

The screenshot shows the Jenkins dashboard for the 'Testslave1' job. The top navigation bar includes the Jenkins logo, a search bar, and user information ('junaidadil'). The main content area shows the 'Status' of the job as 'Testslave1' (green checkmark). Below it, there are links for 'Changes', 'Workspace', 'Build Now' (which is highlighted), 'Configure', 'Delete Project', and 'Rename'. To the right, there are buttons for 'Edit description' and 'Disable Project'. A 'Permalinks' section lists recent builds: 'Last build (#1), 21 min ago', 'Last stable build (#1), 21 min ago', 'Last successful build (#1), 21 min ago', and 'Last completed build (#1), 21 min ago'. At the bottom, a 'Build History' section shows a single entry for build #1 from '19 Jun 2024, 13:49'. A message at the bottom right says 'Activate Windows Go to Settings to activate Windows.'

We can see the build got succeeded

The screenshot shows the Jenkins interface for the 'Testslave1' job. At the top, there's a navigation bar with links to 'Dashboard', 'Testslave1', 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. A prominent green checkmark icon indicates the job is healthy. Below the navigation is a section titled 'Permalinks' with a list of recent builds. A 'Build History' panel displays two builds: '#2 (19 Jun 2024, 14:10)' and '#1 (19 Jun 2024, 13:49)'. At the bottom right, there are links to 'Activate Windows', 'REST API', and 'Jenkins 2.452.2'.

Step-38: Go to build History and go to console output to check the Output

This screenshot shows the details of build #2 of the 'Testslave1' job. The top navigation bar includes links to 'Dashboard', 'Testslave1', and the specific build '#2'. The main content area shows the build status as 'Success' (#2 (19 Jun 2024, 14:10:58)). It provides a summary of the build: 'No changes.', 'Started by user junaidadil', and a breakdown of run time. On the left, a sidebar lists options like 'Status', 'Changes', 'Console Output' (which is currently selected), 'Edit Build Information', 'Delete build #2', 'Timings', and 'Previous Build'. At the bottom right, there are links to 'Activate Windows', 'REST API', and 'Jenkins 2.452.2'.

Now we can see the Build is running on Test-slave node in workspace Testslave1.

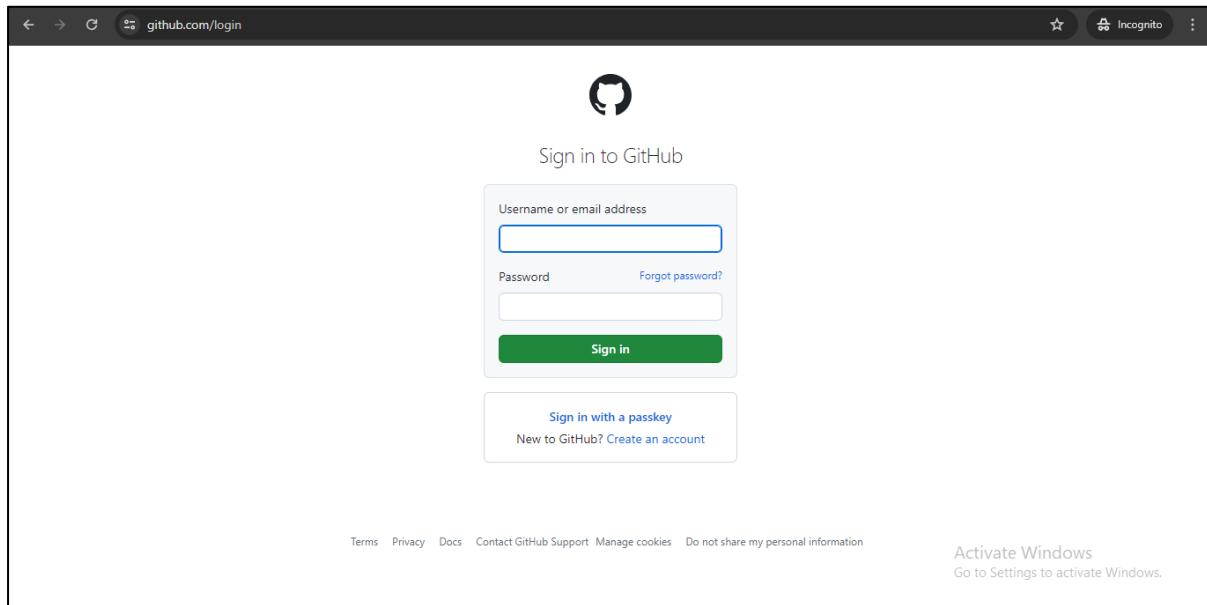
The screenshot shows the Jenkins interface for a build on the 'Testslave1' node. The left sidebar has links for Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Delete build '#2', and Timings. The main content area is titled 'Console Output' and shows the following log output:

```
Started by user junaidadil
Running as SYSTEM
Building remotely on test-slave (slave1) in workspace /home/ec2-user/slavenode/workspace/Testslave1
[Testservel] $ /bin/bash /tmp/jenkins17788599885228380747.sh
Hello, This is Junaid
Finished: SUCCESS
```

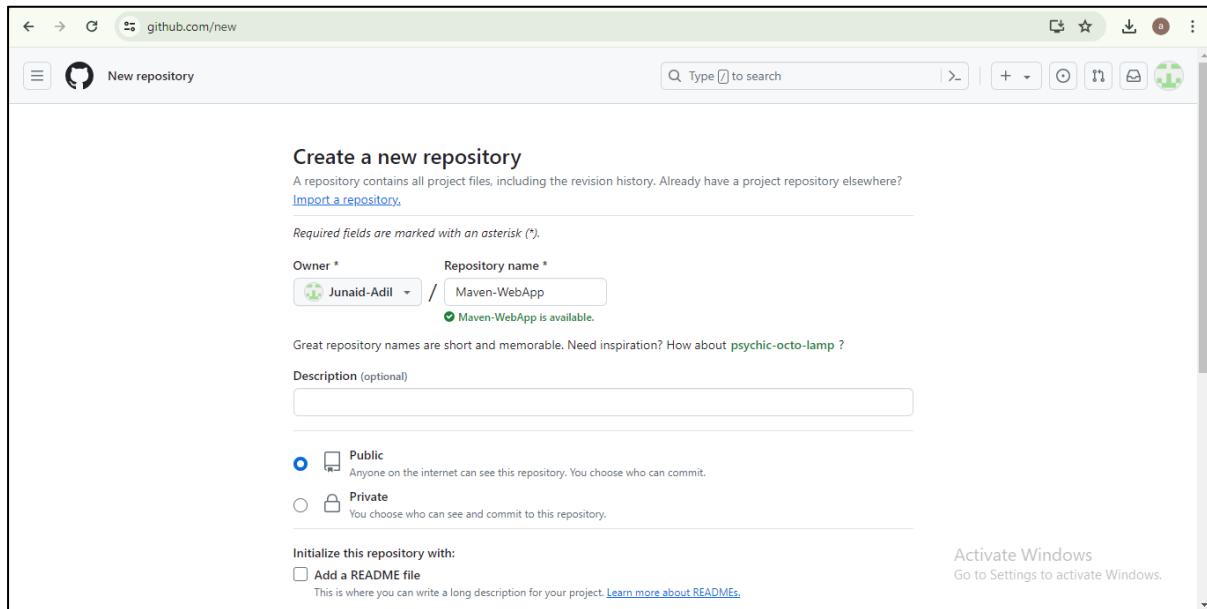
At the bottom right, there are links to Activate Windows, Go to Settings to activate Windows, REST API, and Jenkins 2.452.2.

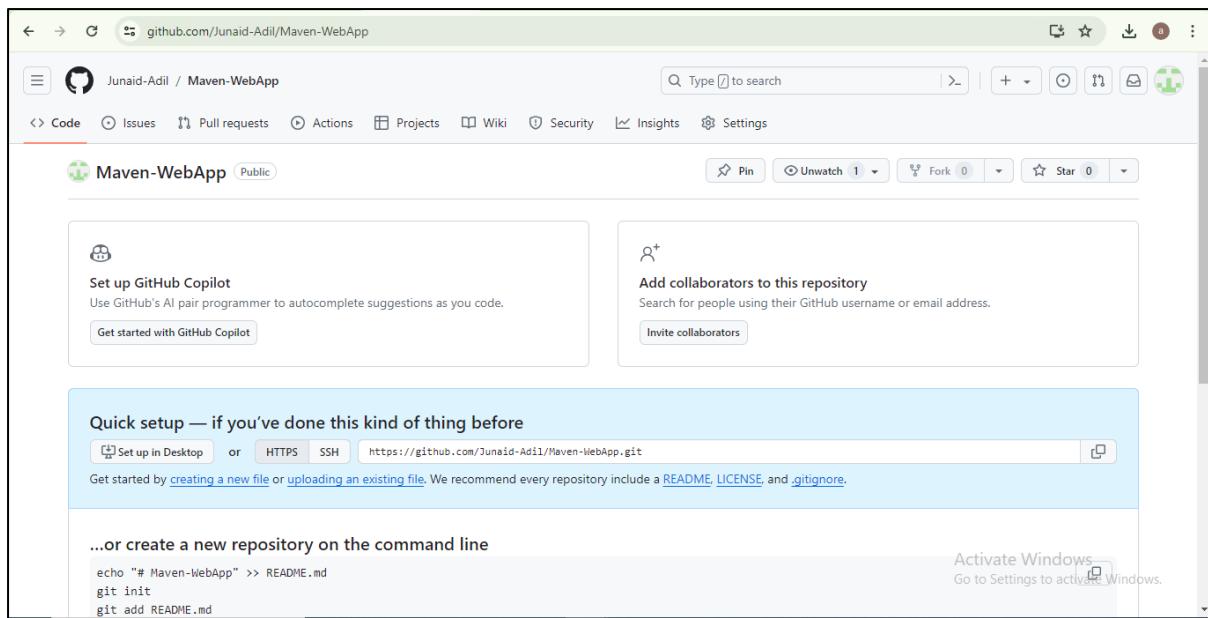
L2 - Create CICD Pipeline to Clone and Build Java Maven Web Application

Step-1: Login to GitHub

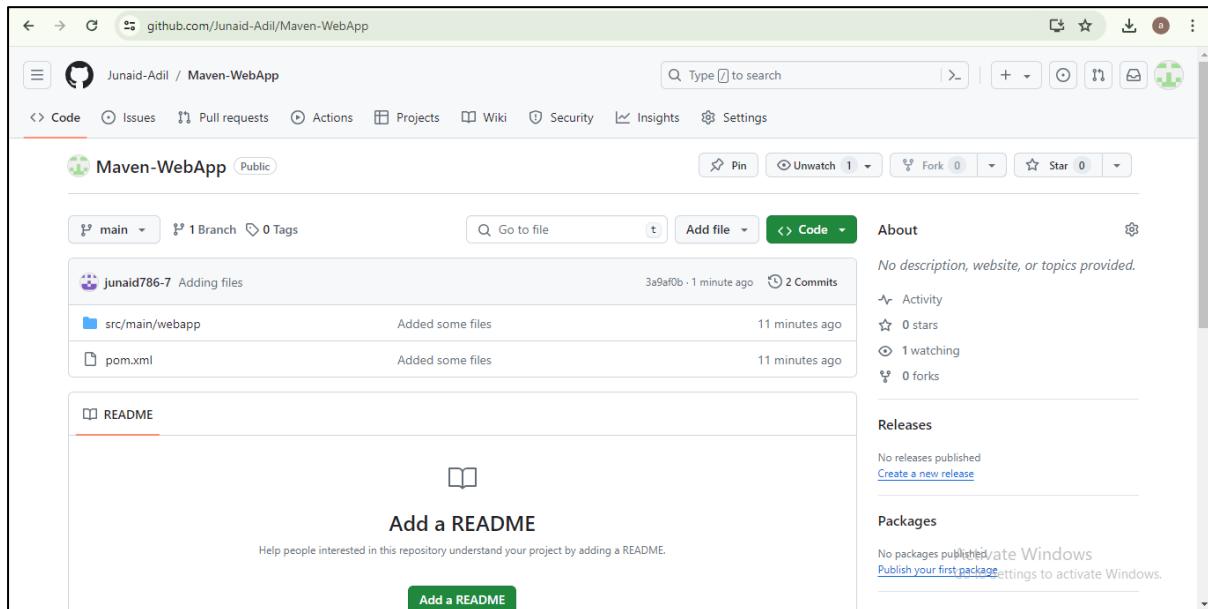


Step-2: Create a new Public Repository





Step-3: Push the source code/Project to GitHub Repository



Jenkins server Installed and running along with Git and maven installed on the server.

The screenshot shows the Jenkins dashboard at the URL 13.201.91.179:8080. The title bar says "Not secure". The main header features the Jenkins logo and the word "Jenkins". The top right includes a search bar ("Search (CTRL+K)"), a help icon, a notifications icon (2 notifications), a security icon, a user profile for "junaidadil", and a "log out" button. The left sidebar has links for "New Item", "Build History", "Manage Jenkins" (which is currently selected), and "My Views". The central content area starts with a "Welcome to Jenkins!" message: "This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project." Below this is a "Start building your software project" section with a "Create a job" button and a plus sign. Further down is a "Set up a distributed build" section with "Set up an agent" and "Configure a cloud" buttons, each with a corresponding icon. At the bottom right of this section is a "Activate Windows" link: "Go to settings to activate Windows.". On the far left, there are two collapsed sections: "Build Queue" (showing "No builds in the queue.") and "Build Executor Status" (showing "1 Idle" and "2 Idle").

Step-4: Go to manage Jenkins

This screenshot is identical to the one above, showing the Jenkins dashboard at 13.201.91.179:8080. The "Manage Jenkins" link in the sidebar is now highlighted in grey. All other elements, including the central content and sidebar sections, remain the same.

Step-5: Go to Tools → Maven Installations

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links like 'New Item', 'Build History', and 'Manage Jenkins'. Below that are sections for 'Build Queue' (empty) and 'Build Executor Status' (2 Idle). The main area is titled 'Manage Jenkins' and has a 'System Configuration' section. It includes icons and descriptions for 'System', 'Nodes', 'Tools' (which is highlighted), 'Clouds', and 'Plugins'. There's also a 'Security' section with 'Security' and 'Credentials' tabs. A bottom banner says 'Activate Windows' with a link to settings.

This screenshot shows the 'Tools' configuration page under 'Manage Jenkins'. It has sections for 'Gradle installations', 'Ant installations', and 'Maven installations'. In the 'Maven installations' section, there's a 'Save' button. A bottom banner says 'Activate Windows' with a link to settings.

Step-6: Enter name and select the version

The screenshot shows the Jenkins 'Configure Tools' page at the URL 13.201.91.179:8080/manage/configureTools/. The 'Tools' section is selected. A 'Maven' configuration is being edited. The 'Name' field contains 'Maven s/w'. The 'Install automatically' checkbox is checked. Under 'Install from Apache', the 'Version' dropdown is set to '3.9.8'. There is a 'Save' button at the bottom.

Apply → save

Step-7: Go to dashboard → new item

The screenshot shows the Jenkins dashboard at the URL 13.201.91.179:8080. The dashboard features a 'Welcome to Jenkins!' message and a 'Start building your software project' section. On the left, there are links for 'Build History', 'Manage Jenkins', and 'My Views'. A 'New Item' button is prominently displayed. The bottom right corner includes an 'Activate Windows' message: 'Go to Settings to activate Windows.' Below the dashboard, the URL 13.201.91.179:8080/view/all/newJob is visible.

Step-8: Enter name and select project type as pipeline project → Ok

The screenshot shows the Jenkins interface for creating a new job. The title bar says 'Not secure 13.201.91.179:8080/view/all/newJob'. The main area has a heading 'Enter an item name' with a text input field containing 'AssignmentL2'. Below it, a note says '> Required field'. There are three project type options: 'Freestyle project' (selected), 'Pipeline', and 'Multi-configuration project'. The 'Pipeline' option is described as 'Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.' A blue 'OK' button is at the bottom. On the right, there's an 'Activate Windows' message: 'Go to Settings to activate Windows.'

Step-9: In description enter as “Java-Maven-WebApp”

The screenshot shows the Jenkins 'General' configuration page for the 'AssignmentL2' job. The title bar says 'Not secure 13.201.91.179:8080/job/AssignmentL2/configure'. The left sidebar has tabs for 'Configure', 'General' (selected), 'Advanced Project Options', and 'Pipeline'. The 'General' tab has a 'Description' field containing 'Java-Maven-WebApp'. Below it are several checkboxes: 'Discard old builds', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', and 'GitHub project'. At the bottom are 'Save' and 'Apply' buttons. An 'Activate Windows' message is visible on the right.

Step-10: Write the pipeline script

The screenshot shows the Jenkins Pipeline configuration screen. The left sidebar has tabs for General, Advanced Project Options, and Pipeline, with Pipeline selected. The main area is titled 'Pipeline' and 'Definition'. A dropdown menu shows 'Pipeline script'. Below it is a code editor containing the following Groovy script:

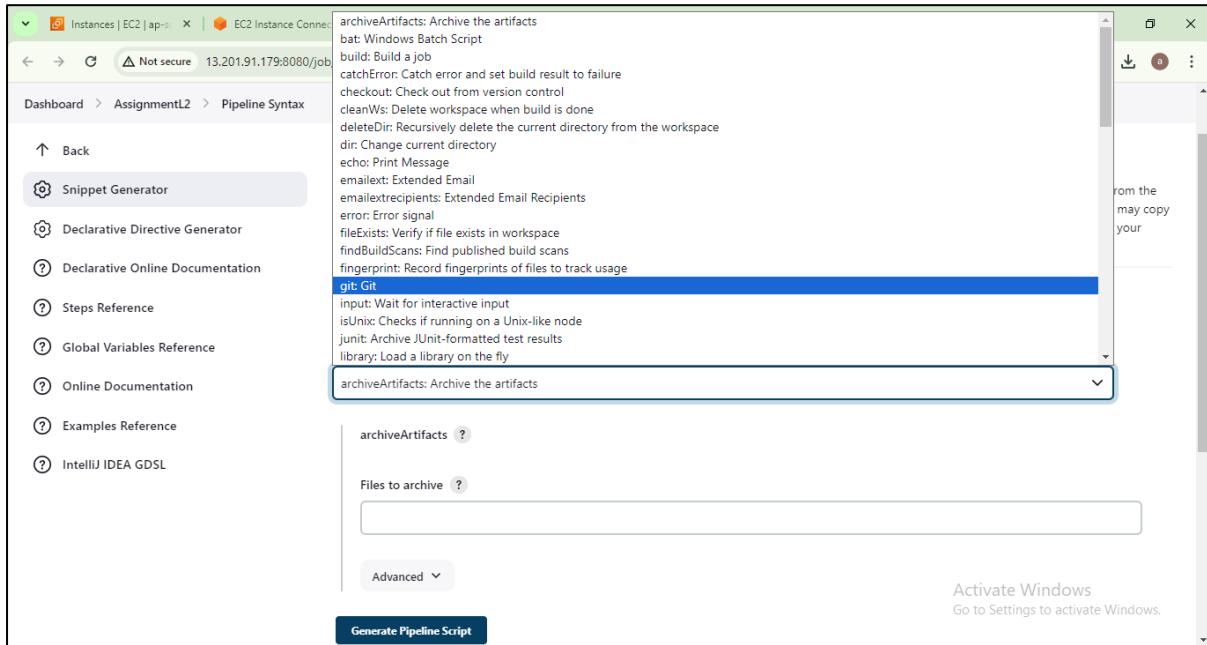
```
1 pipeline {  
2     agent any  
3  
4     stages {  
5         stage('Checkout') {  
6             steps {  
7                 git 'https://github.com/your-username/your-java-maven-webapp.git'  
8             }  
9         }  
10    }  
11    stage('Build') {  
12        steps {  
13            sh 'mvn clean install'  
14        }  
15    }  
16}  
17}
```

Below the code editor is a checkbox for 'Use Groovy Sandbox'. At the bottom are 'Save' and 'Apply' buttons.

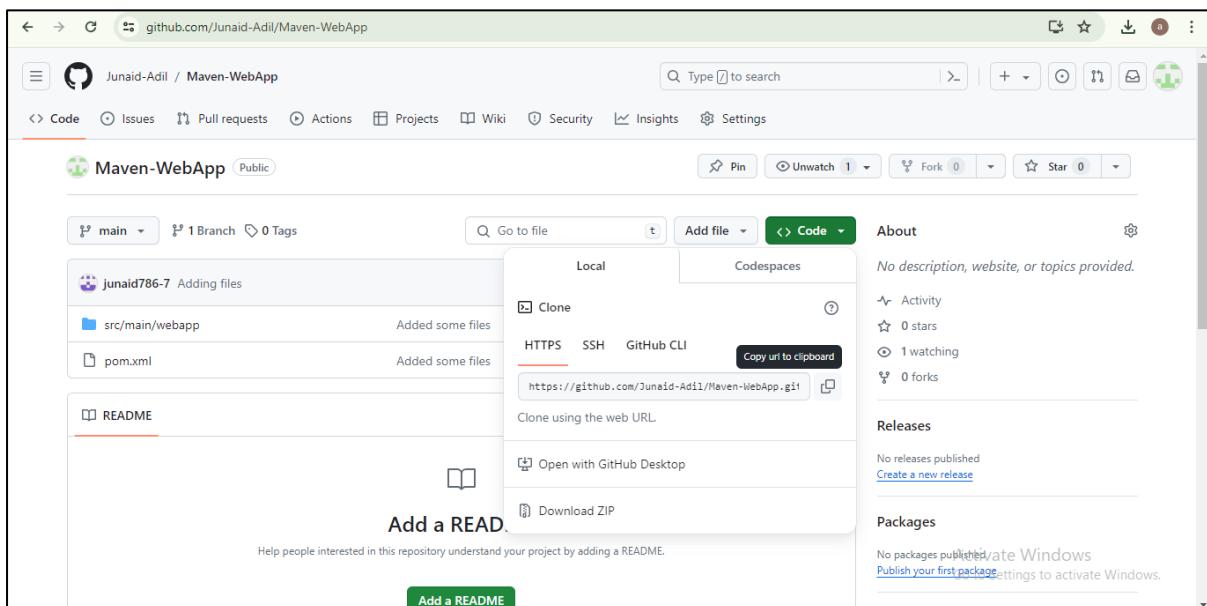
Step-11: Click on pipeline syntax

The screenshot shows the same Jenkins Pipeline configuration screen as before, but with a new link at the bottom of the code editor: 'Pipeline Syntax'. This link is underlined and appears to be the target of a click action. The rest of the interface is identical to the previous screenshot.

Step-12: In sample steps select git, as we are pulling the code from GitHub.

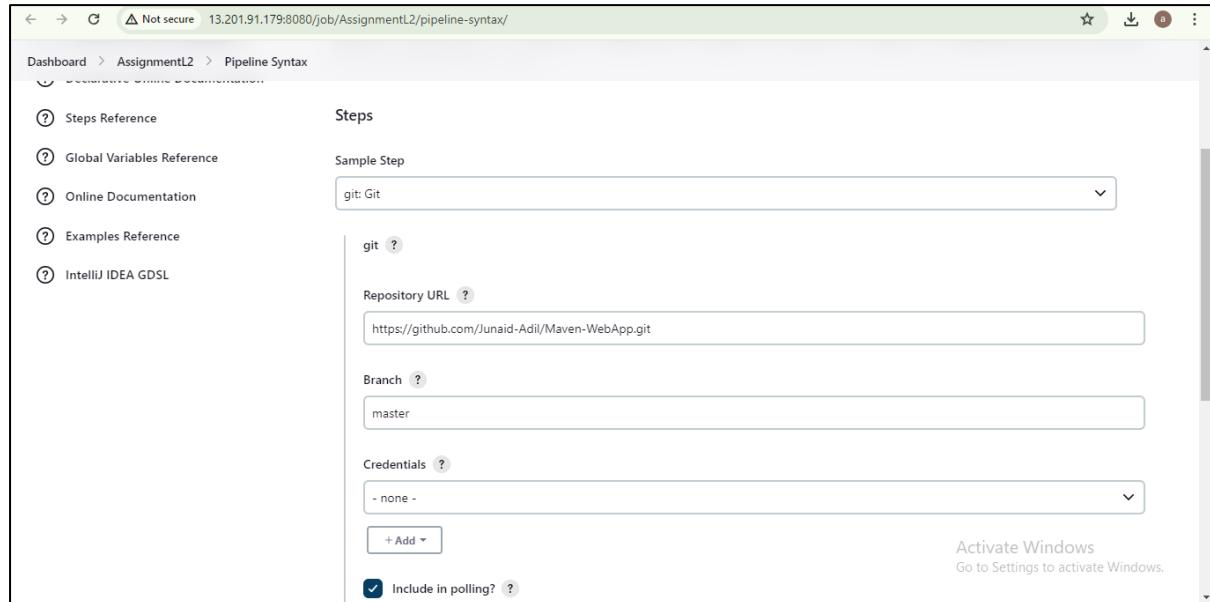


Step-13: Go to GitHub Repository and copy the HTTPS Link.

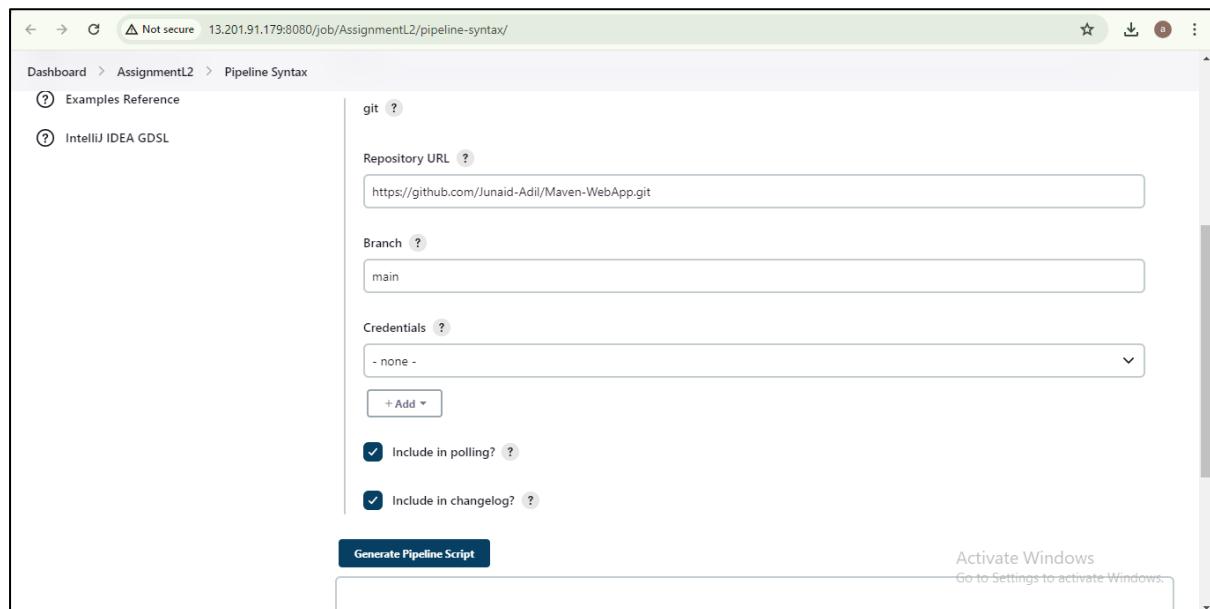


Step-14: Paste it in the pipeline syntax steps.

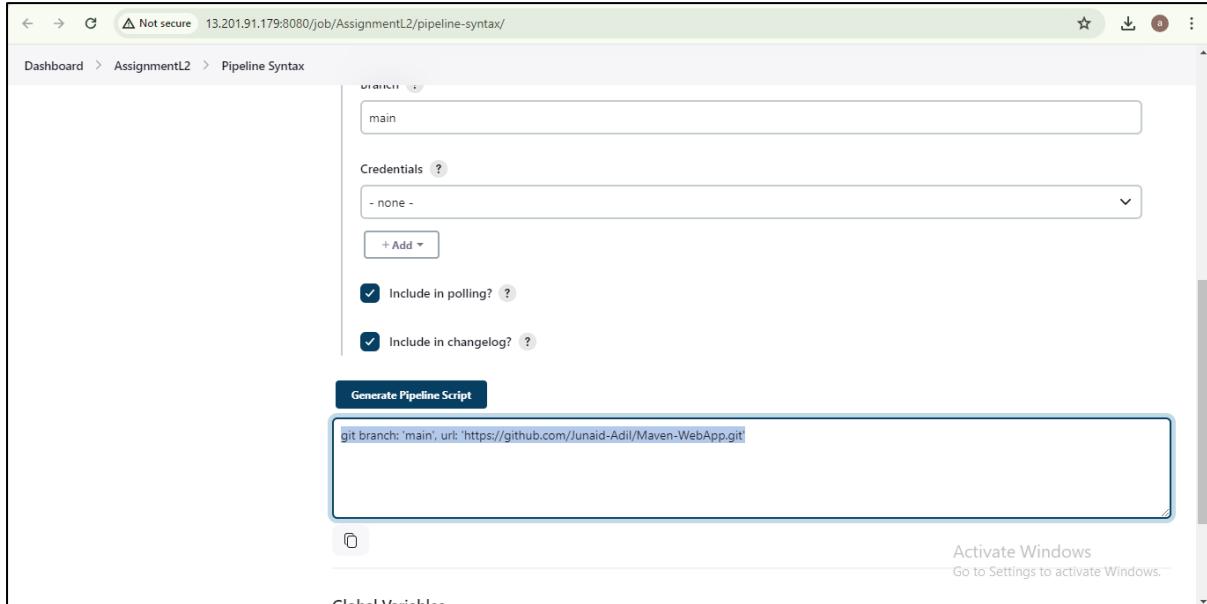
Enter the repository URL <https://github.com/Junaid-Adil/Maven-WebApp.git>



Step-15: Update the branch as per the created repository in Github and as it is public repository no need to give the credentials → click Generate Pipeline Script



Step-16: Copy the generated pipeline script URL “git branch: 'main', url: 'https://github.com/Junaid-Adil/Maven-WebApp.git' ” and paste it in the script



Dashboard > AssignmentL2 > Pipeline Syntax

Branch: main

Credentials: - none -

Include in polling?

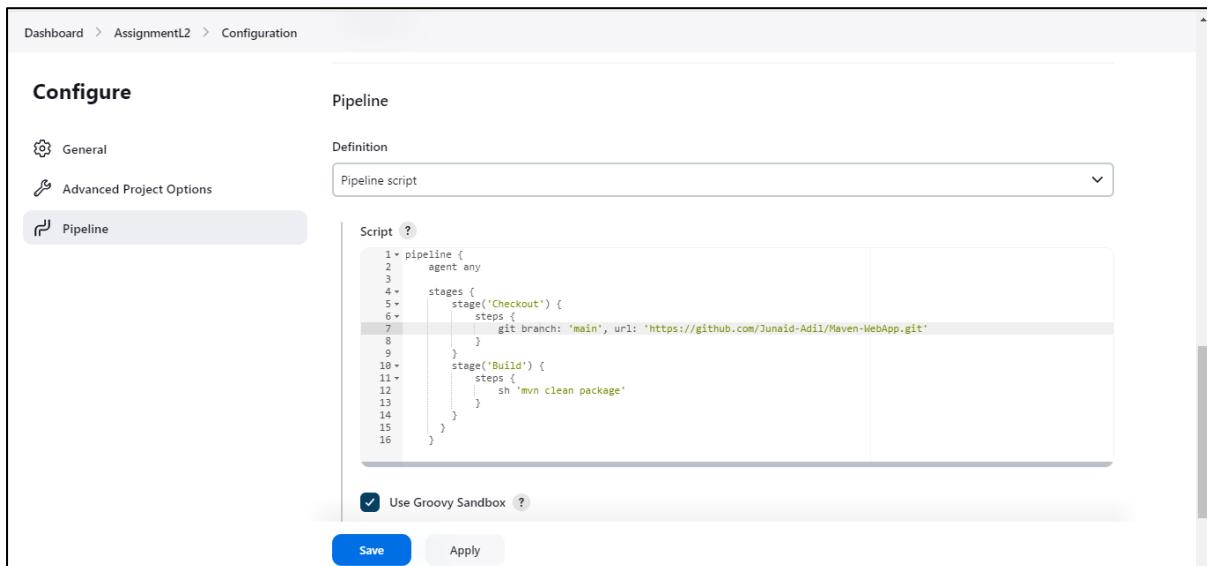
Include in changelog?

Generate Pipeline Script

```
git branch: 'main', url: 'https://github.com/Junaid-Adil/Maven-WebApp.git'
```

Activate Windows
Go to Settings to activate Windows.

Step-17: Paste it in the Pipeline script stage.



Dashboard > AssignmentL2 > Configuration

Configure

General Advanced Project Options Pipeline

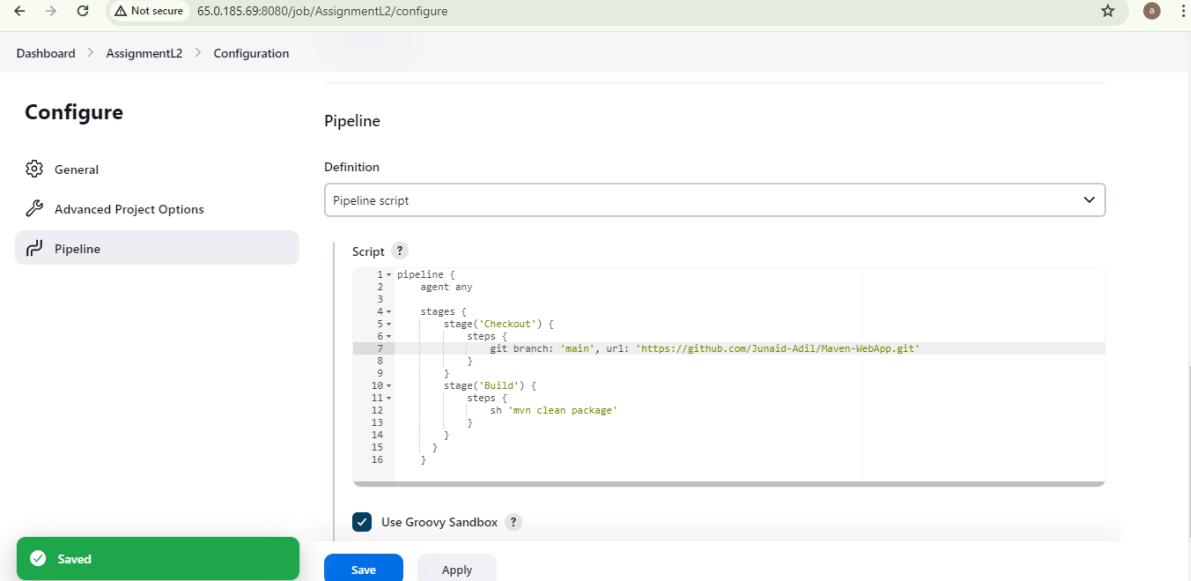
Definition: Pipeline script

```
1 > pipeline {
2   agent any
3
4   stages {
5     stage('Checkout') {
6       steps {
7         git branch: 'main', url: 'https://github.com/Junaid-Adil/Maven-WebApp.git'
8       }
9     }
10    stage('Build') {
11      steps {
12        sh 'mvn clean package'
13      }
14    }
15  }
16}
```

Use Groovy Sandbox

Save Apply

Step-18: Apply → Save

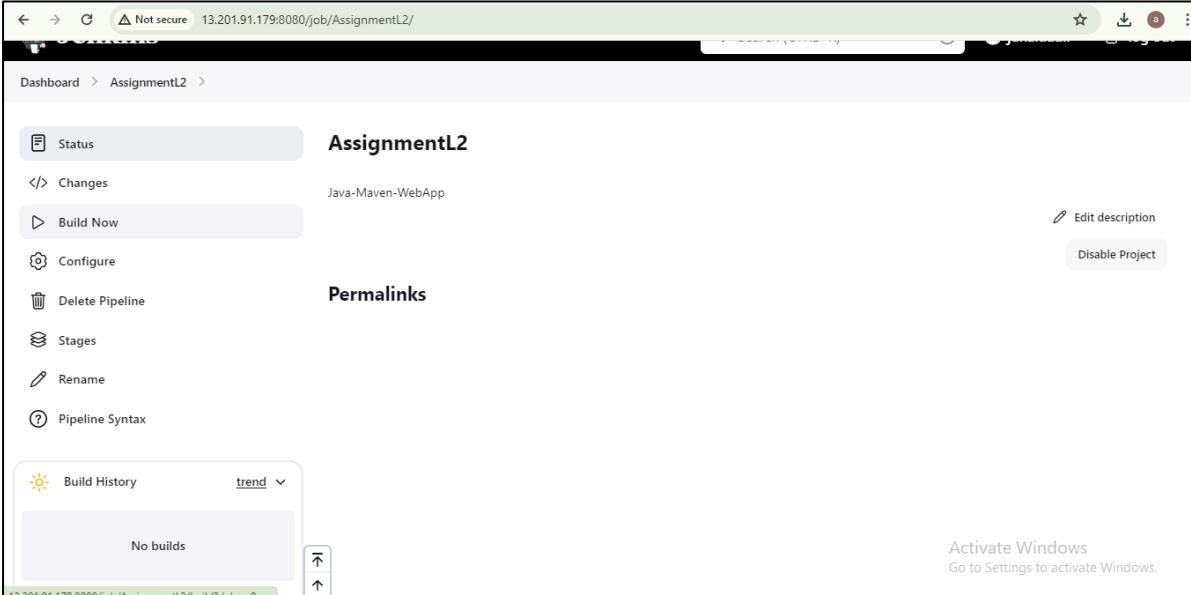


The screenshot shows the Jenkins Pipeline configuration page for a job named 'AssignmentL2'. The 'Pipeline' tab is selected. In the 'Definition' section, the 'Pipeline script' dropdown is open. The script content is as follows:

```
1 pipeline {  
2     agent any  
3  
4     stages {  
5         stage('Checkout') {  
6             steps {  
7                 git branch: 'main', url: 'https://github.com/Junaid-Adil/Maven-WebApp.git'  
8             }  
9         }  
10        stage('Build') {  
11            steps {  
12                sh 'mvn clean package'  
13            }  
14        }  
15    }  
16}
```

Below the script, there is a checked checkbox labeled 'Use Groovy Sandbox'. At the bottom, there are three buttons: a green 'Saved' button with a checkmark, a blue 'Save' button, and a grey 'Apply' button.

Step-19: Now click on Build Now



The screenshot shows the Jenkins job status page for 'AssignmentL2'. The 'Build Now' button is highlighted with a red box. The page also displays the job name 'AssignmentL2', the description 'Java-Maven-WebApp', and various configuration options like 'Edit description' and 'Disable Project'. On the right side, there is a message about activating Windows.

In build history we can see the log

The screenshot shows the Jenkins interface for the 'AssignmentL2' job. On the left, there's a sidebar with options like 'Changes', 'Build Now', 'Configure', 'Delete Pipeline', 'Stages', 'Rename', and 'Pipeline Syntax'. The main area is titled 'Permalinks' and contains a 'Build History' section. A single build is listed with the timestamp '19 Jun 2024, 21:41'. Below the build list are links for 'Atom feed for all' and 'Atom feed for failures'. At the bottom right of the main area, there are links for 'Activate Windows', 'Go to Settings to activate Windows.', 'REST API', and 'Jenkins 2.452.2'.

Step-20: Build got succeeded. Go to console output to check the details

The screenshot shows the Jenkins interface for the 'AssignmentL2' job, specifically the 'Console Output' tab. The left sidebar includes 'Status', 'Changes', 'Console Output' (which is selected), 'View as plain text', 'Edit Build Information', 'Delete build '#4', 'Timings', 'Git Build Data', 'Pipeline Overview', 'Pipeline Console', 'Restart from Stage', and 'Replay'. The main content area is titled 'Console Output' and displays the build log. The log starts with 'Started by user junaidadil' and shows the Jenkins pipeline executing commands to clone a GitHub repository and checkout the 'main' branch. There are also links for 'Activate Windows' and 'Go to Settings to activate Windows.' at the bottom right.

Dashboard > AssignmentL2 > #4

Results :

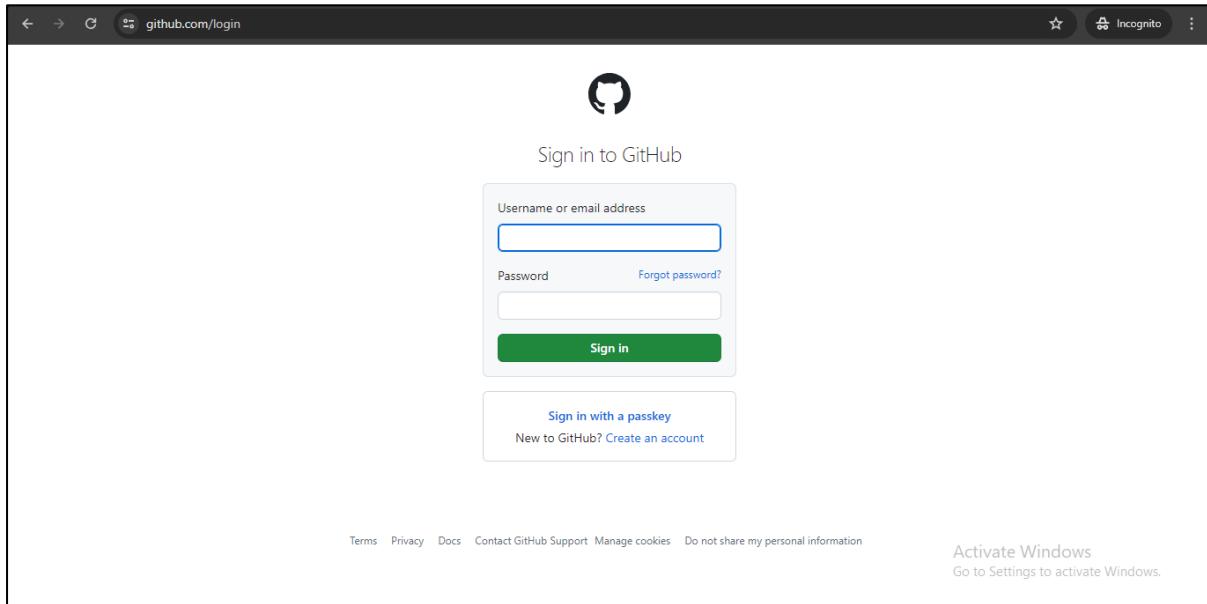
```
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-war-plugin:3.3.1:war (default-war) @ 01-maven-web-app ---
[INFO] Packaging webapp
[INFO] Assembling webapp [01-maven-web-app] in [/var/lib/jenkins/workspace/AssignmentL2/target/maven-web-app]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/AssignmentL2/src/main/webapp]
[INFO] Building war: /var/lib/jenkins/workspace/AssignmentL2/target/maven-web-app.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.165s
[INFO] Finished at: Wed Jun 19 22:00:48 UTC 2024
[INFO] Final Memory: 9M/93M
[INFO] -----
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

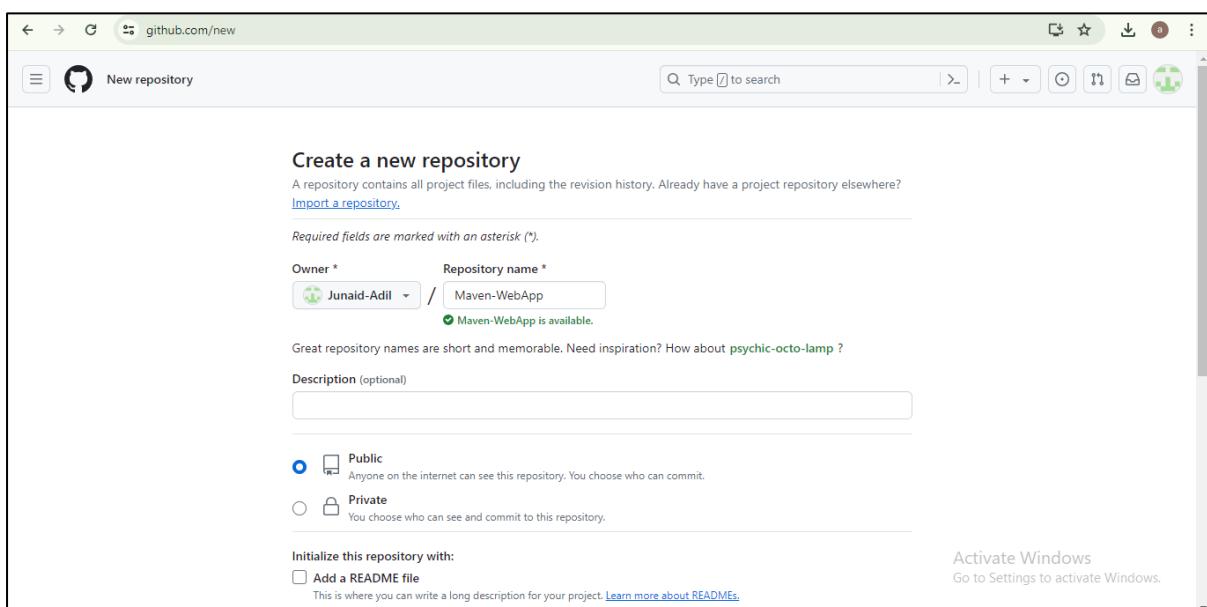
Activate Windows
Go to Settings to activate Windows.

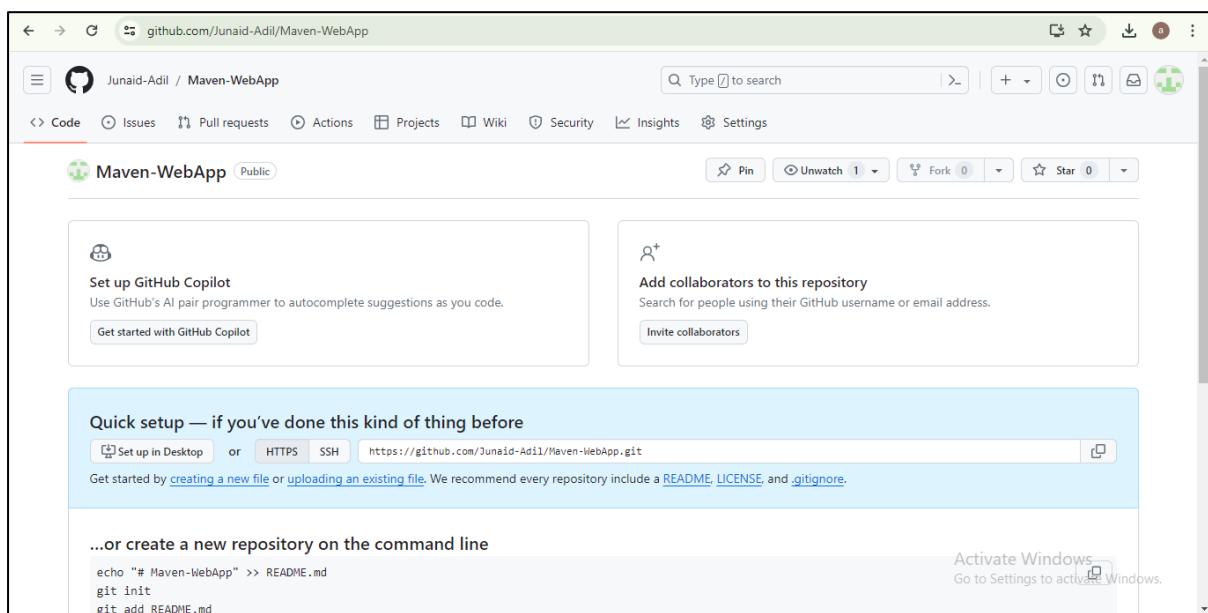
L3 - Create CICD Pipeline to Deploy the Maven Web Application in Tomcat Server

Step-1: Login to GitHub

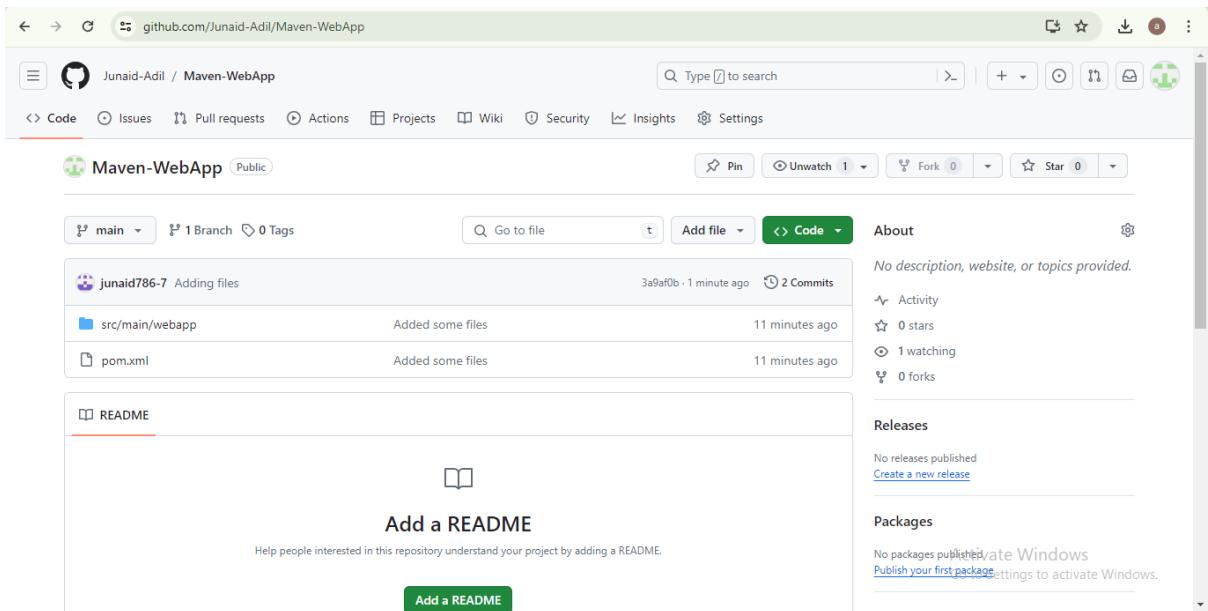


Step-2: Create a new Public Repository





Step-3: Push the source code/Project to GitHub Repository



→ Jenkins server Installed and running along with Git and maven installed on the server.

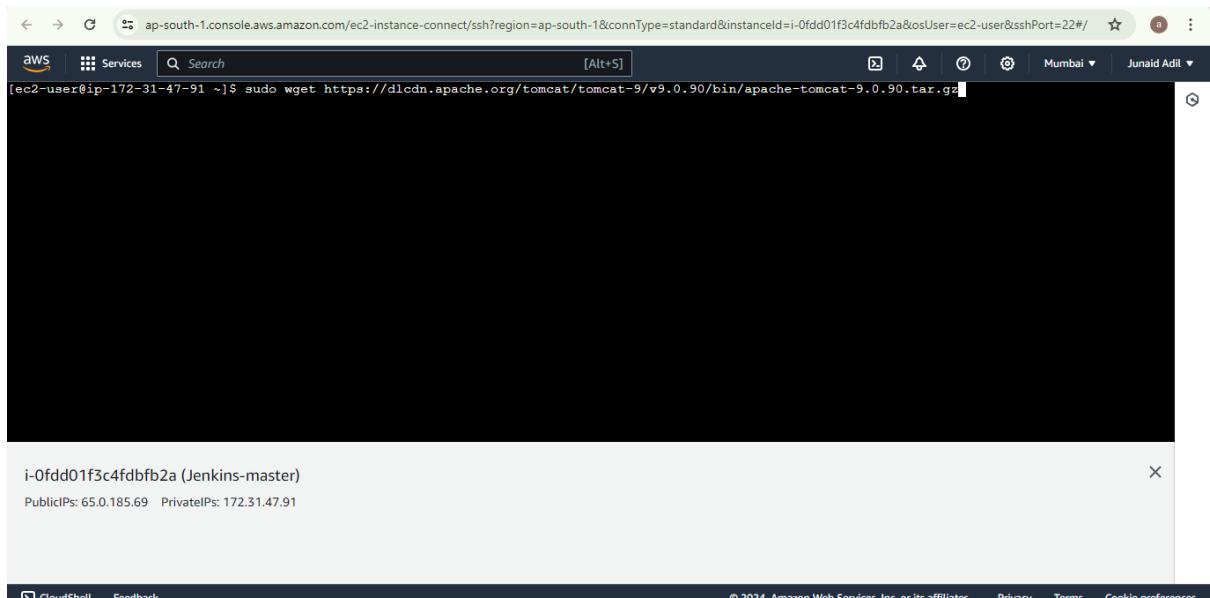
The screenshot shows the Jenkins dashboard at the URL <http://65.0.185.69:8080>. The dashboard includes a sidebar with links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. The main area displays a table of builds with columns for Status (S), Workstation (W), Name, Last Success, Last Failure, and Last Duration. A single build named 'AssignmentL2' is listed with a green checkmark icon, indicating success. The status bar at the bottom right shows 'Jenkins 2.452.2'.

Step-4: Now install Tomcat on the server,

Go to “tomcat.apache.org/download-90.cgi” in browser → on **tar.gz** right click and select – copy link address

The screenshot shows the Tomcat download page at the URL <https://tomcat.apache.org/download-90.cgi>. The page lists various Tomcat versions and documentation. On the right, there's a section for 'Binary Distributions' under the '9.0.90' heading. A context menu is open over a 'tar.gz' link, with the option 'Copy link address' highlighted. The status bar at the bottom right shows 'Jenkins 2.452.2'.

Step-5: Paste the link as “sudo wget <Link Address> “ in instance

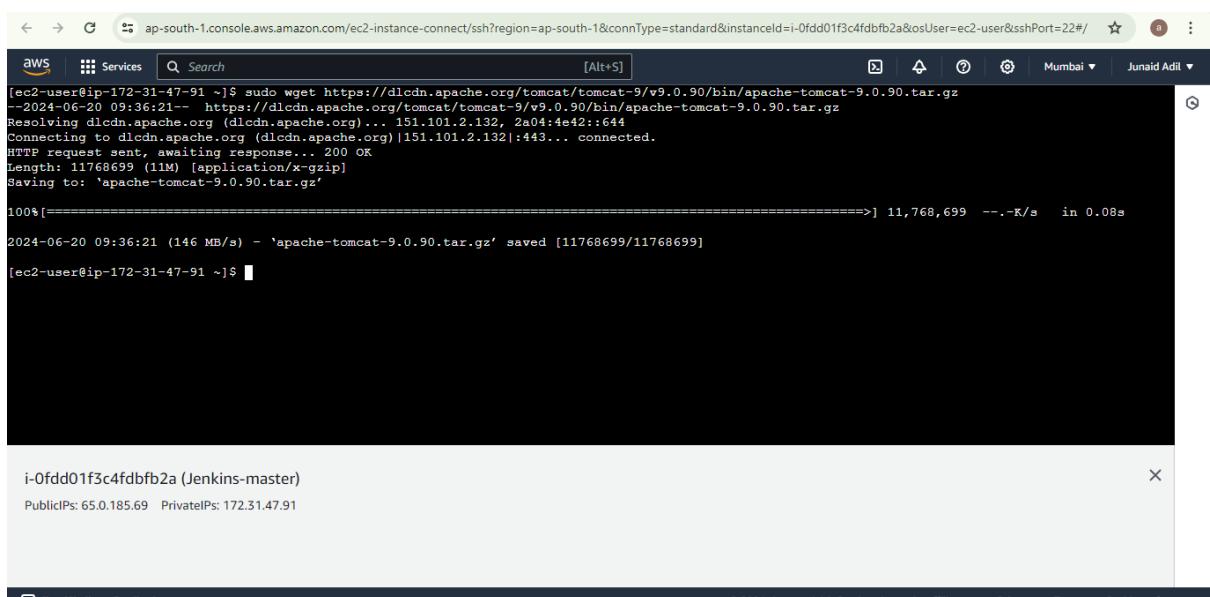


```
[ec2-user@ip-172-31-47-91 ~]$ sudo wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.90/bin/apache-tomcat-9.0.90.tar.gz
```

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

We can see the tomcat has been installed in the same server.



```
[ec2-user@ip-172-31-47-91 ~]$ sudo wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.90/bin/apache-tomcat-9.0.90.tar.gz
--2024-06-20 09:36:21-- https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.90/bin/apache-tomcat-9.0.90.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42:644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11768699 (11M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.90.tar.gz'

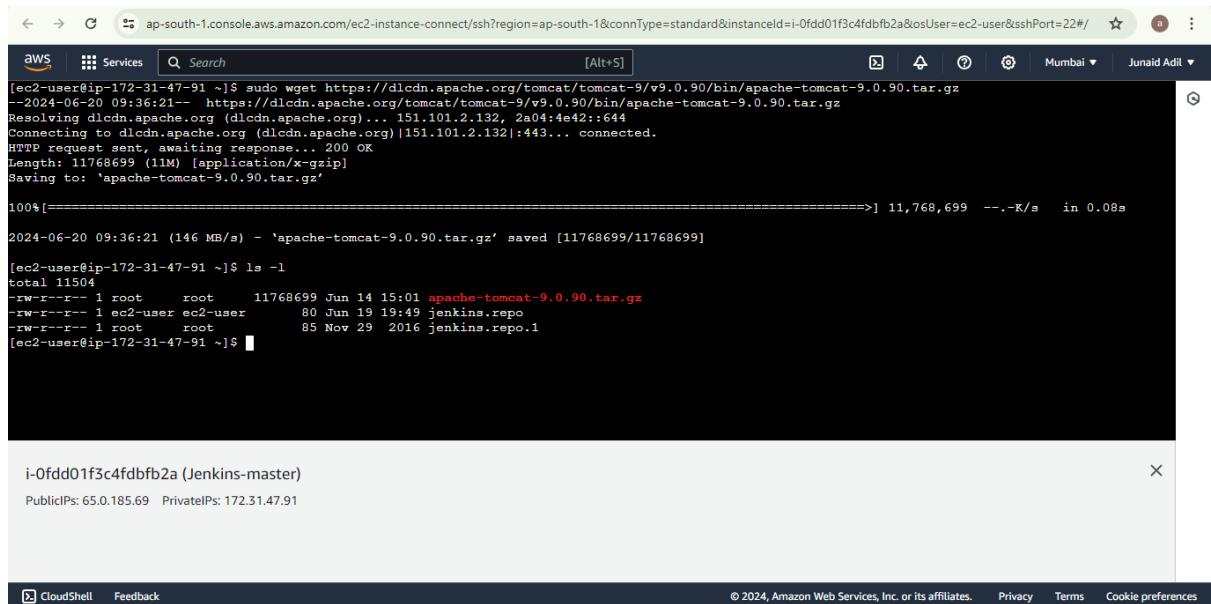
100%[=====] 11,768,699  --.-K/s   in 0.08s

2024-06-20 09:36:21 (146 MB/s) - 'apache-tomcat-9.0.90.tar.gz' saved [11768699/11768699]

[ec2-user@ip-172-31-47-91 ~]$
```

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



```
[ec2-user@ip-172-31-47-91 ~]$ sudo wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.90/bin/apache-tomcat-9.0.90.tar.gz
--2024-06-20 09:36:21-- https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.90/bin/apache-tomcat-9.0.90.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11768699 (11M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.90.tar.gz'

100%[=====] 11,768,699  --.-K/s   in 0.08s

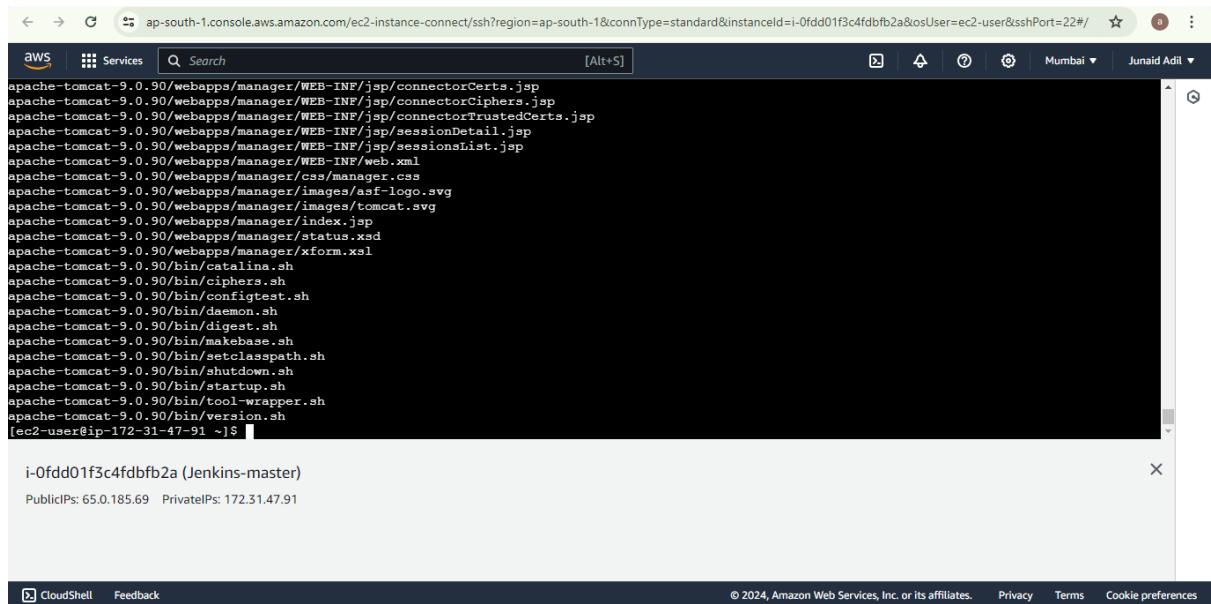
2024-06-20 09:36:21 (146 MB/s) - 'apache-tomcat-9.0.90.tar.gz' saved [11768699/11768699]

[ec2-user@ip-172-31-47-91 ~]$ ls -l
total 11504
-rw-r--r-- 1 root      root    11768699 Jun 14 15:01 apache-tomcat-9.0.90.tar.gz
-rw-r--r-- 1 ec2-user  ec2-user     80 Jun 19 19:49 jenkins.repo
-rw-r--r-- 1 root      root    85 Nov 29  2016 jenkins.repo.1
[ec2-user@ip-172-31-47-91 ~]$
```

i-0fdd01f3c4fdbfb2a (Jenkins-master)

PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

Step-6: Now extract the Tar file using command “ **tar -xvf <name of file>**”

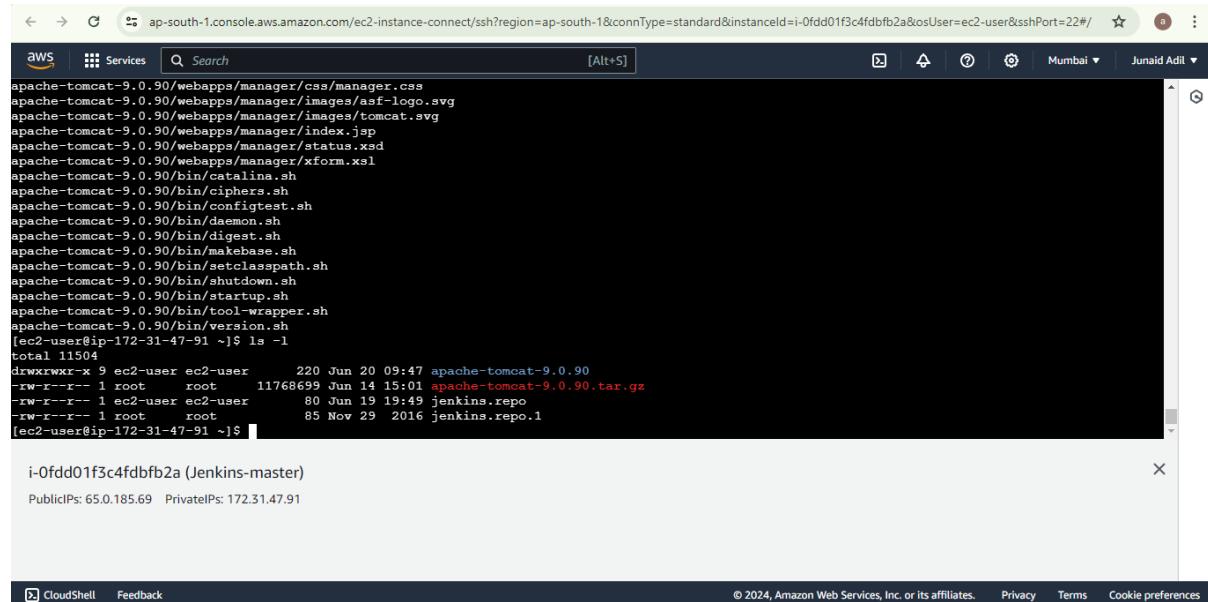


```
apache-tomcat-9.0.90/webapps/manager/WEB-INF/jsp/connectorCerts.jsp
apache-tomcat-9.0.90/webapps/manager/WEB-INF/jsp/connectorCiphers.jsp
apache-tomcat-9.0.90/webapps/manager/WEB-INF/jsp/connectorTrustedCerts.jsp
apache-tomcat-9.0.90/webapps/manager/WEB-INF/jsp/sessionDetail.jsp
apache-tomcat-9.0.90/webapps/manager/WEB-INF/jsp/sessionsList.jsp
apache-tomcat-9.0.90/webapps/manager/WEB-INF/web.xml
apache-tomcat-9.0.90/webapps/manager/css/manager.css
apache-tomcat-9.0.90/webapps/manager/images/asf-logo.svg
apache-tomcat-9.0.90/webapps/manager/images/tomcat.svg
apache-tomcat-9.0.90/webapps/manager/index.jsp
apache-tomcat-9.0.90/webapps/manager/status.xsd
apache-tomcat-9.0.90/webapps/manager/xform.xsl
apache-tomcat-9.0.90/bin/catalina.sh
apache-tomcat-9.0.90/bin/ciphers.sh
apache-tomcat-9.0.90/bin/configtest.sh
apache-tomcat-9.0.90/bin/daemon.sh
apache-tomcat-9.0.90/bin/digest.sh
apache-tomcat-9.0.90/bin/makebase.sh
apache-tomcat-9.0.90/bin/setclaspath.sh
apache-tomcat-9.0.90/bin/shutdown.sh
apache-tomcat-9.0.90/bin/startup.sh
apache-tomcat-9.0.90/bin/tool-wrapper.sh
apache-tomcat-9.0.90/bin/version.sh
[ec2-user@ip-172-31-47-91 ~]$
```

i-0fdd01f3c4fdbfb2a (Jenkins-master)

PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

Step-7: Run “ls -l” to see the extracted file. The file in Blue colour is the extracted file.

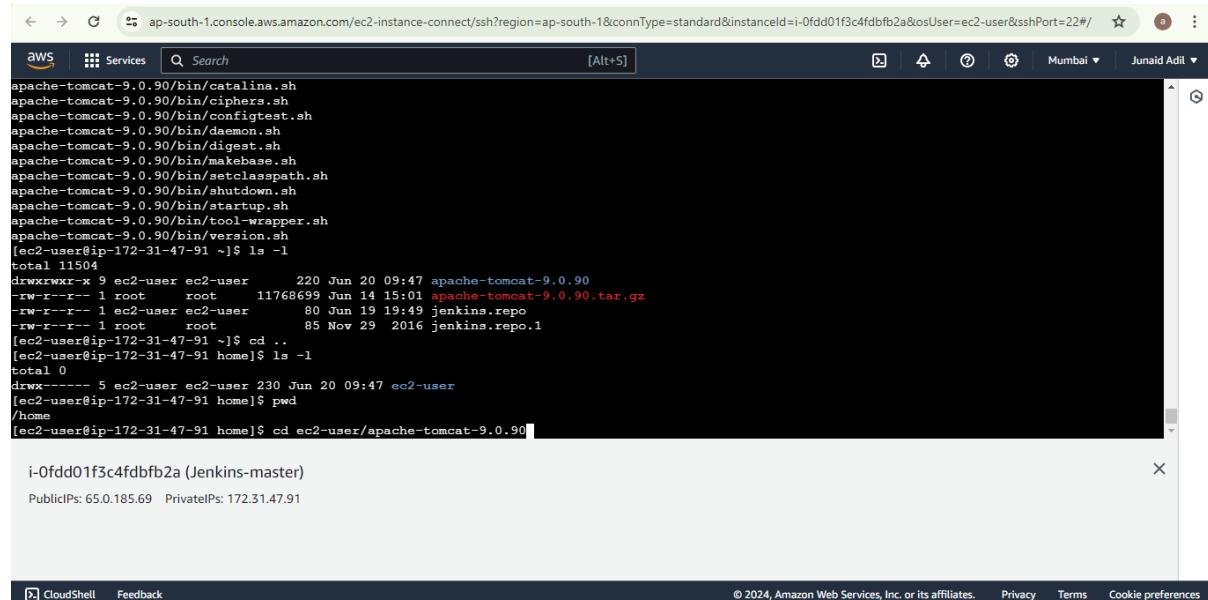


```
apache-tomcat-9.0.90/webapps/manager/css/manager.css
apache-tomcat-9.0.90/webapps/manager/images/asf-logo.svg
apache-tomcat-9.0.90/webapps/manager/images/tomcat.svg
apache-tomcat-9.0.90/webapps/manager/index.jsp
apache-tomcat-9.0.90/webapps/manager/status.xsd
apache-tomcat-9.0.90/webapps/manager/xform.xsl
apache-tomcat-9.0.90/bin/catalina.sh
apache-tomcat-9.0.90/bin/ciphers.sh
apache-tomcat-9.0.90/bin/configtest.sh
apache-tomcat-9.0.90/bin/daemon.sh
apache-tomcat-9.0.90/bin/digest.sh
apache-tomcat-9.0.90/bin/makebase.sh
apache-tomcat-9.0.90/bin/setclasspath.sh
apache-tomcat-9.0.90/bin/shutdown.sh
apache-tomcat-9.0.90/bin/startup.sh
apache-tomcat-9.0.90/bin/tool-wrapper.sh
apache-tomcat-9.0.90/bin/version.sh
[ec2-user@ip-172-31-47-91 ~]$ ls -l
total 11504
drwxrwx-- 9 ec2-user ec2-user 220 Jun 20 09:47 apache-tomcat-9.0.90
-rw-r--r-- 1 root root 11768699 Jun 14 15:01 apache-tomcat-9.0.90.tar.gz
-rw-r--r-- 1 ec2-user ec2-user 80 Jun 19 19:49 jenkins.repo
-rw-r--r-- 1 root root 85 Nov 29 2016 jenkins.repo.1
[ec2-user@ip-172-31-47-91 ~]$
```

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

As Jenkins and Tomcat server runs on same port number: 8080, we cannot run both the Jenkins and Tomcat on same port number. So we have to change the Tomcat port number to 9090.

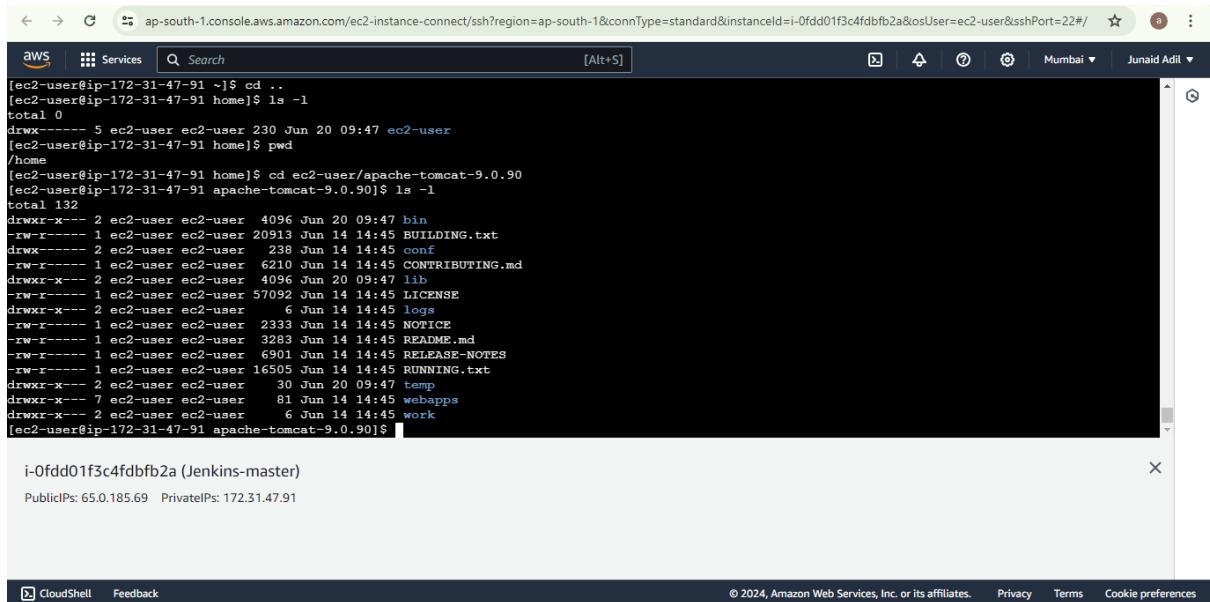
Step-8: Go to /home/ec2-user/apache-tomcat-9.0.90



```
apache-tomcat-9.0.90/bin/catalina.sh
apache-tomcat-9.0.90/bin/ciphers.sh
apache-tomcat-9.0.90/bin/configtest.sh
apache-tomcat-9.0.90/bin/daemon.sh
apache-tomcat-9.0.90/bin/digest.sh
apache-tomcat-9.0.90/bin/makebase.sh
apache-tomcat-9.0.90/bin/setclasspath.sh
apache-tomcat-9.0.90/bin/shutdown.sh
apache-tomcat-9.0.90/bin/startup.sh
apache-tomcat-9.0.90/bin/tool-wrapper.sh
apache-tomcat-9.0.90/bin/version.sh
[ec2-user@ip-172-31-47-91 ~]$ ls -l
total 11504
drwxrwxr-x 9 ec2-user ec2-user 220 Jun 20 09:47 apache-tomcat-9.0.90
-rw-r--r-- 1 root root 11768699 Jun 14 15:01 apache-tomcat-9.0.90.tar.gz
-rw-r--r-- 1 ec2-user ec2-user 80 Jun 19 19:49 jenkins.repo
-rw-r--r-- 1 root root 85 Nov 29 2016 jenkins.repo.1
[ec2-user@ip-172-31-47-91 ~]$ cd ..
[ec2-user@ip-172-31-47-91 home]$ ls -l
total 0
drwx----- 5 ec2-user ec2-user 230 Jun 20 09:47 ec2-user
[ec2-user@ip-172-31-47-91 home]$ pwd
/home
[ec2-user@ip-172-31-47-91 home]$ cd ec2-user/apache-tomcat-9.0.90
```

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

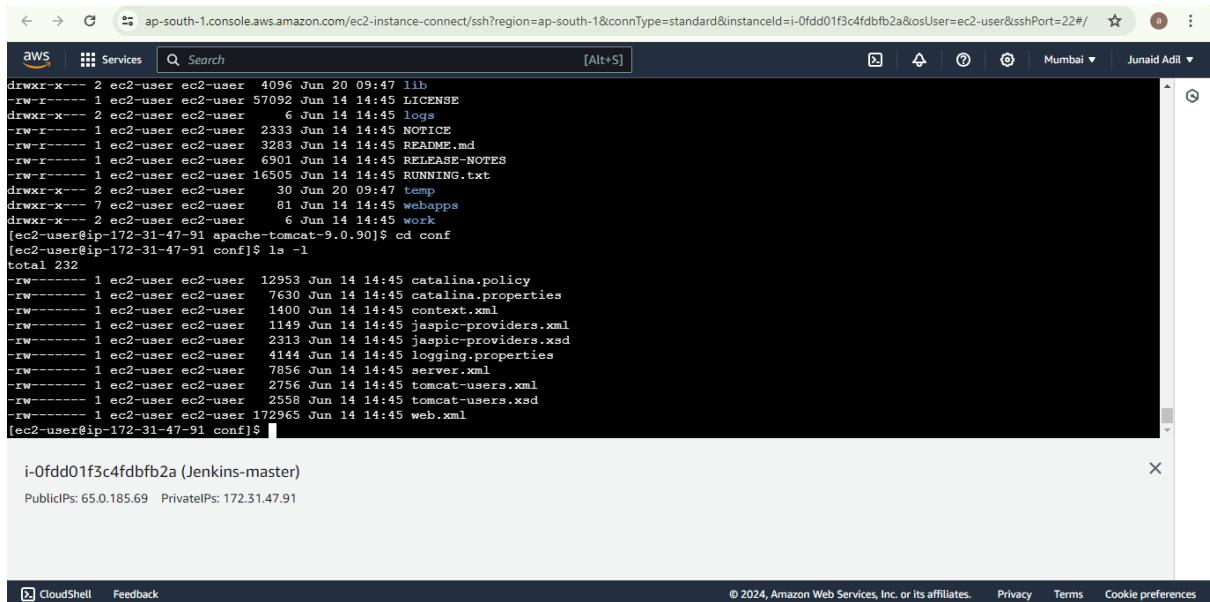
Run “ls -l” to see the files and folders of the current directory



```
[ec2-user@ip-172-31-47-91 ~]$ cd ..
[ec2-user@ip-172-31-47-91 home]$ ls -l
total 0
drwxr-x--- 5 ec2-user ec2-user 230 Jun 20 09:47 ec2-user
[ec2-user@ip-172-31-47-91 home]$ cd ec2-user/apache-tomcat-9.0.90
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ ls -l
total 132
drwxr-x--- 2 ec2-user ec2-user 4096 Jun 20 09:47 bin
-rw-r---- 1 ec2-user ec2-user 20913 Jun 14 14:45 BUILDING.txt
drwxr-x--- 2 ec2-user ec2-user 238 Jun 14 14:45 conf
-rw-r---- 1 ec2-user ec2-user 6210 Jun 14 14:45 CONTRIBUTING.md
drwxr-x--- 2 ec2-user ec2-user 4096 Jun 20 09:47 lib
-rw-r---- 1 ec2-user ec2-user 57092 Jun 14 14:45 LICENSE
drwxr-x--- 2 ec2-user ec2-user 6 Jun 14 14:45 logs
-rw-r---- 1 ec2-user ec2-user 2333 Jun 14 14:45 NOTICE
-rw-r---- 1 ec2-user ec2-user 3283 Jun 14 14:45 README.md
-rw-r---- 1 ec2-user ec2-user 6901 Jun 14 14:45 RELEASE-NOTES
-rw-r---- 1 ec2-user ec2-user 16505 Jun 14 14:45 RUNNING.txt
drwxr-x--- 2 ec2-user ec2-user 30 Jun 20 09:47 temp
drwxr-x--- 7 ec2-user ec2-user 81 Jun 14 14:45 webapps
drwxr-x--- 2 ec2-user ec2-user 6 Jun 14 14:45 work
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$
```

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

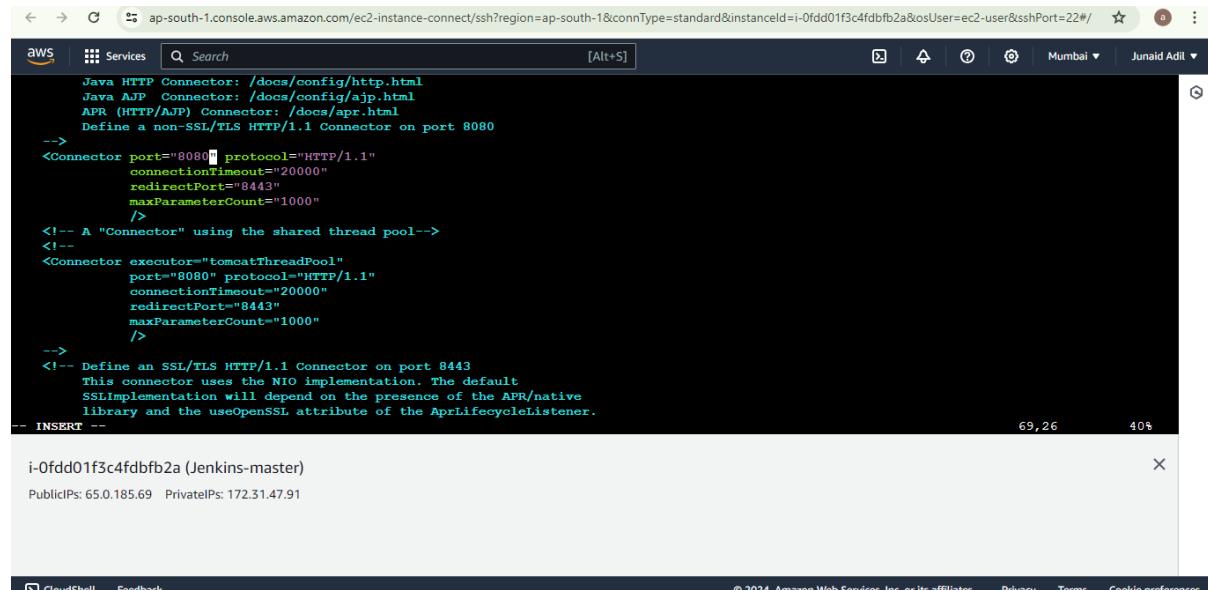
Run “cd conf” to change the PWD to config folder.



```
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ cd conf
[ec2-user@ip-172-31-47-91 conf]$ ls -l
total 232
-rw-r---- 1 ec2-user ec2-user 12953 Jun 14 14:45 catalina.policy
-rw-r---- 1 ec2-user ec2-user 7630 Jun 14 14:45 catalina.properties
-rw-r---- 1 ec2-user ec2-user 1400 Jun 14 14:45 context.xml
-rw-r---- 1 ec2-user ec2-user 1149 Jun 14 14:45 jaspic-providers.xml
-rw-r---- 1 ec2-user ec2-user 2313 Jun 14 14:45 jaspic-providers.xsd
-rw-r---- 1 ec2-user ec2-user 4144 Jun 14 14:45 logging.properties
-rw-r---- 1 ec2-user ec2-user 7856 Jun 14 14:45 server.xml
-rw-r---- 1 ec2-user ec2-user 2756 Jun 14 14:45 tomcat-users.xml
-rw-r---- 1 ec2-user ec2-user 2598 Jun 14 14:45 tomcat-users.xsd
-rw-r---- 1 ec2-user ec2-user 172965 Jun 14 14:45 web.xml
[ec2-user@ip-172-31-47-91 conf]$
```

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

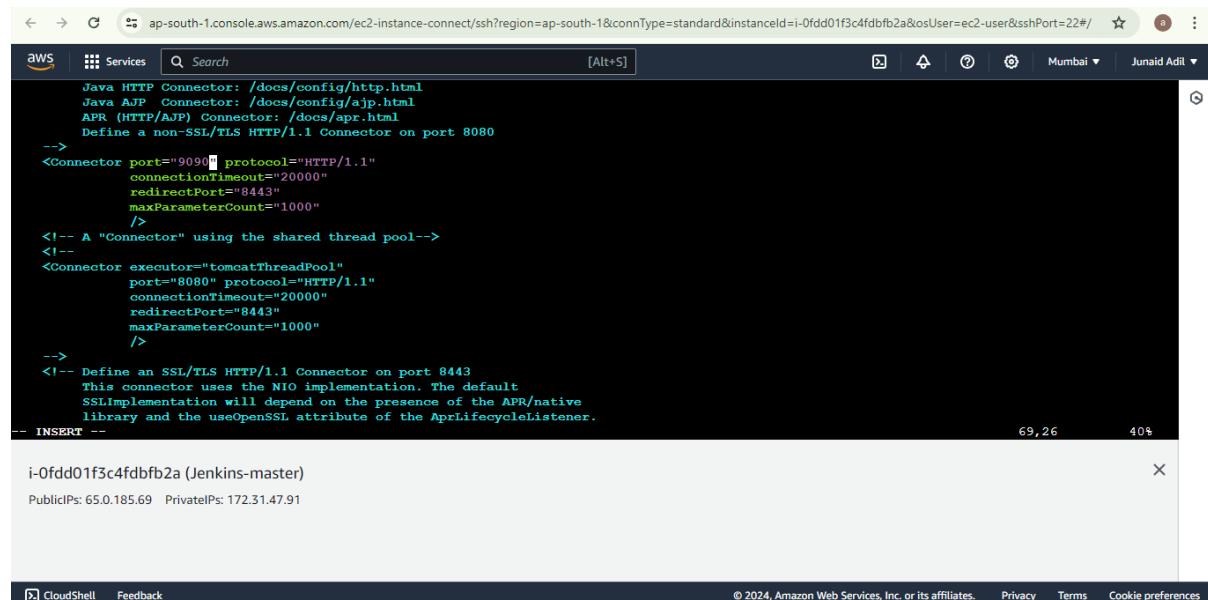
Step-9: Execute “ vi server.xml” to edit the port number



```
Java HTTP Connector: /docs/config/http.html
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443"
           maxParameterCount="1000"
       />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
           port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443"
           maxParameterCount="1000"
       />
-->
<!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
     This connector uses the NIO implementation. The default
     SSLImplementation will depend on the presence of the APR/native
     library and the useOpenSSL attribute of the AprLifecycleListener.
-- INSERT -->

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91
```

Updated the port number to “9090”.



```
Java HTTP Connector: /docs/config/http.html
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector port="9090" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443"
           maxParameterCount="1000"
       />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
           port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443"
           maxParameterCount="1000"
       />
-->
<!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
     This connector uses the NIO implementation. The default
     SSLImplementation will depend on the presence of the APR/native
     library and the useOpenSSL attribute of the AprLifecycleListener.
-- INSERT -->

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91
```

```

-rw-r----- 1 ec2-user ec2-user 57092 Jun 14 14:45 LICENSE
drwxr-x--- 2 ec2-user ec2-user 6 Jun 14 14:45 logs
-rw-r----- 1 ec2-user ec2-user 2333 Jun 14 14:45 NOTICE
-rw-r----- 1 ec2-user ec2-user 3283 Jun 14 14:45 README.md
-rw-r----- 1 ec2-user ec2-user 6901 Jun 14 14:45 RELEASE-NOTES
-rw-r----- 1 ec2-user ec2-user 16505 Jun 14 14:45 RUNNING.txt
drwxr-x--- 2 ec2-user ec2-user 30 Jun 20 09:47 temp
drwxr-x--- 7 ec2-user ec2-user 81 Jun 14 14:45 webapps
drwxr-x--- 2 ec2-user ec2-user 6 Jun 14 14:45 work
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ cd conf
[ec2-user@ip-172-31-47-91 conf]$ ls -l
total 232
-rw-r----- 1 ec2-user ec2-user 12953 Jun 14 14:45 catalina.policy
-rw-r----- 1 ec2-user ec2-user 7630 Jun 14 14:45 catalina.properties
-rw-r----- 1 ec2-user ec2-user 1400 Jun 14 14:45 context.xml
-rw-r----- 1 ec2-user ec2-user 1149 Jun 14 14:45 jaspic-providers.xml
-rw-r----- 1 ec2-user ec2-user 2313 Jun 14 14:45 jaspic-providers.xsd
-rw-r----- 1 ec2-user ec2-user 4144 Jun 14 14:45 logging.properties
-rw-r----- 1 ec2-user ec2-user 7856 Jun 14 14:45 server.xml
-rw-r----- 1 ec2-user ec2-user 2756 Jun 14 14:45 tomcat-users.xml
-rw-r----- 1 ec2-user ec2-user 2558 Jun 14 14:45 tomcat-users.xsd
-rw-r----- 1 ec2-user ec2-user 172965 Jun 14 14:45 web.xml
[ec2-user@ip-172-31-47-91 conf]$ ls
[ec2-user@ip-172-31-47-91 conf]$ 

```

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

Step-10: Tomcat port number has been updated to 9090. So now we have to give access to 9090 in Security group, Inbound Rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0ecdadb81b7c8a6d36	Custom TCP	TCP	8080	Cust... ▾	0.0.0.0/0
sgr-09ca4381181c7affa	SSH	TCP	22	Cust... ▾	0.0.0.0/0
-	Custom TCP	TCP	9090	Any... ▾	0.0.0.0/0

Add rule

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Preview changes Save rules

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Available
Jenkins-master	i-0fdd01f3c4fdbfb2a	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source
-	sgr-0cadb81b7c8a6d36	8080	TCP	0.0.0.0/0
-	sgr-07951814d596c304e	9090	TCP	0.0.0.0/0
-	sgr-09ca4381181c7affa	22	TCP	0.0.0.0/0

Step-11: In order to get the access to use Tomcat we need to give permission.

Edit context.xml file and add users.

Go to webapps/manager/META-INF/

```

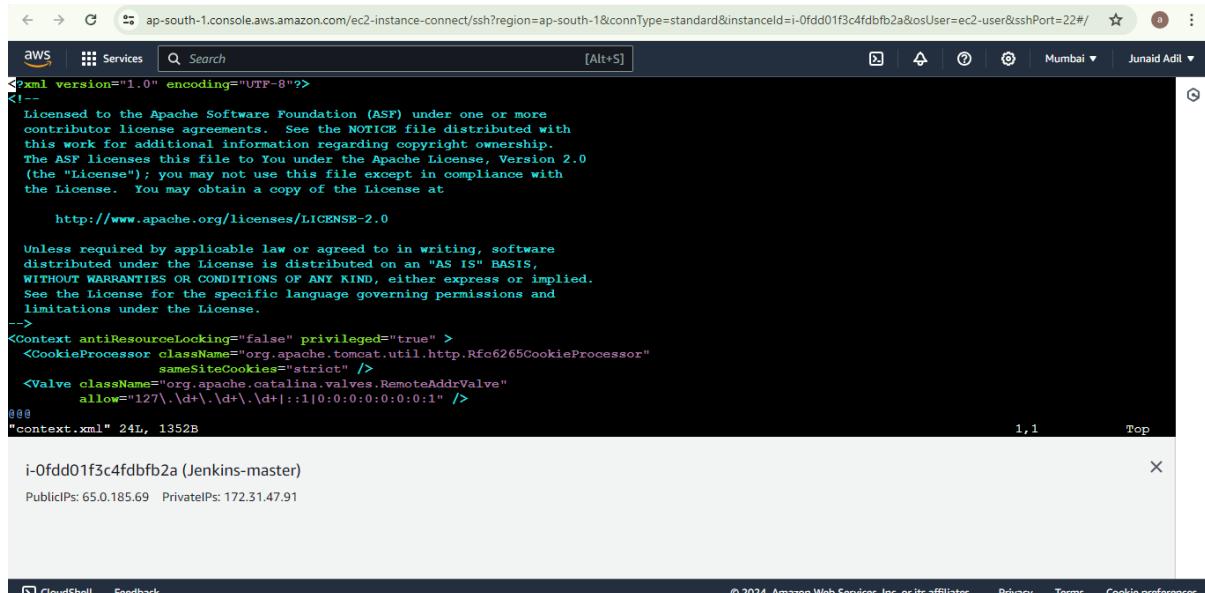
-rw----- 1 ec2-user ec2-user 2558 Jun 14 14:45 tomcat-users.xml
-rw----- 1 ec2-user ec2-user 172965 Jun 14 14:45 web.xml
[ec2-user@ip-172-31-47-91 conf]$ cd ../../
[ec2-user@ip-172-31-47-91 ~]$ ls
apache-tomcat-9.0.90 apache-tomcat-9.0.90.tar.gz jenkins.repo jenkins.repo.1
[ec2-user@ip-172-31-47-91 ~]$ cd ..
[ec2-user@ip-172-31-47-91 home]$ ls
ec2-user
[ec2-user@ip-172-31-47-91 home]$ cd ec2-user
[ec2-user@ip-172-31-47-91 ~]$ ls
apache-tomcat-9.0.90 apache-tomcat-9.0.90.tar.gz jenkins.repo jenkins.repo.1
[ec2-user@ip-172-31-47-91 ~]$ cd apache-tomcat-9.0.90
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ cd webapps
[ec2-user@ip-172-31-47-91 webapps]$ ls
docs examples host-manager manager ROOT
[ec2-user@ip-172-31-47-91 webapps]$ cd manager
[ec2-user@ip-172-31-47-91 manager]$ ls
css images index.jsp META-INF status.xml WEB-INF xform.xsl
[ec2-user@ip-172-31-47-91 manager]$ cd META-INF
[ec2-user@ip-172-31-47-91 META-INF]$ ls
context.xml
[ec2-user@ip-172-31-47-91 META-INF]$ 

```

i-0fdd01f3c4fdbfb2a (Jenkins-master)

Public IPs: 65.0.185.69 Private IPs: 172.31.47.91

Step-12: Edit the context.xml file → edit valve tag → in valve allow remove all the content and enter “.*”

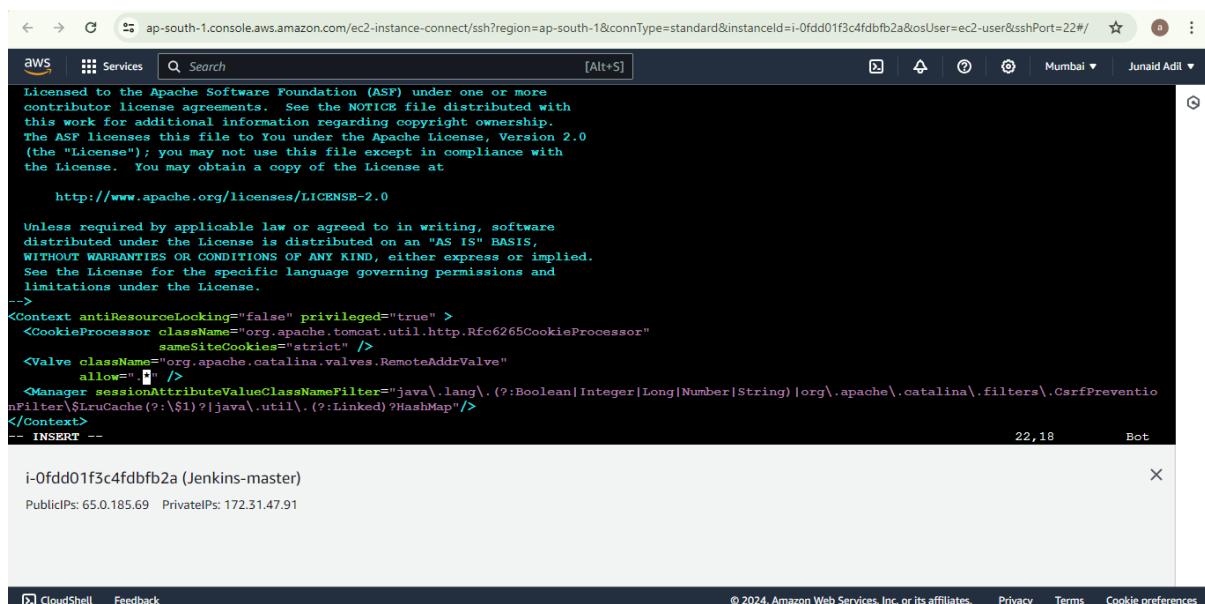


```
xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
    sameSiteCookies="strict" />
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\\.\\d+\\.\\d+\\.\\d+:1|0:0:0:0:0:1" />
</Context>
"context.xml" 24L, 1352B
```

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91



```
xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
    sameSiteCookies="strict" />
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow=".*" />
  <Manager sessionAttributeValueClassNameFilter="java\\.lang\\.\\?Boolean|\\?Integer|\\?Long|\\?Number|\\?String|\\?org\\.apache\\.\\?catalina\\.\\?filters\\\\.CsrfPreventionFilter\\?LruCache\\?\\?java\\.util\\\\.\\?Linked\\?HashMap" />
</Context>
-- INSERT --
```

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

Updated to “.*”

The screenshot shows a terminal session in AWS CloudShell. The user is navigating through the Apache Tomcat 9.0.90 directory structure. They are in the 'webapps' directory, specifically under the 'manager' subdirectory. The user runs 'vi context.xml' to edit the context configuration file. The terminal shows the file being opened and modified, with changes like 'Stopped' and 'Started' status indicators. The session ends with the user exiting the editor.

```
apache-tomcat-9.0.90 apache-tomcat-9.0.90.tar.gz jenkins.repo jenkins.repo.1
[ec2-user@ip-172-31-47-91 ~]$ cd ..
[ec2-user@ip-172-31-47-91 home]$ ls
ec2-user
[ec2-user@ip-172-31-47-91 home]$ cd ec2-user
[ec2-user@ip-172-31-47-91 ~]$ ls
apache-tomcat-9.0.90 apache-tomcat-9.0.90.tar.gz jenkins.repo jenkins.repo.1
[ec2-user@ip-172-31-47-91 ~]$ cd apache-tomcat-9.0.90
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ cd webapps
[ec2-user@ip-172-31-47-91 webapps]$ ls
docs examples host-manager manager ROOT
[ec2-user@ip-172-31-47-91 webapps]$ cd manager
[ec2-user@ip-172-31-47-91 manager]$ ls
css images index.jsp META-INF status.xsd WEB-INF xform.xsl
[ec2-user@ip-172-31-47-91 manager]$ cd META-INF
[ec2-user@ip-172-31-47-91 META-INF]$ ls
context.xml
[ec2-user@ip-172-31-47-91 META-INF]$ vi context.xml

[1]+ Stopped                  vi context.xml
[ec2-user@ip-172-31-47-91 META-INF]$ vi context.xml
[ec2-user@ip-172-31-47-91 META-INF]$
```

Step-13: Now to configure the user. Go to Configuration directory.

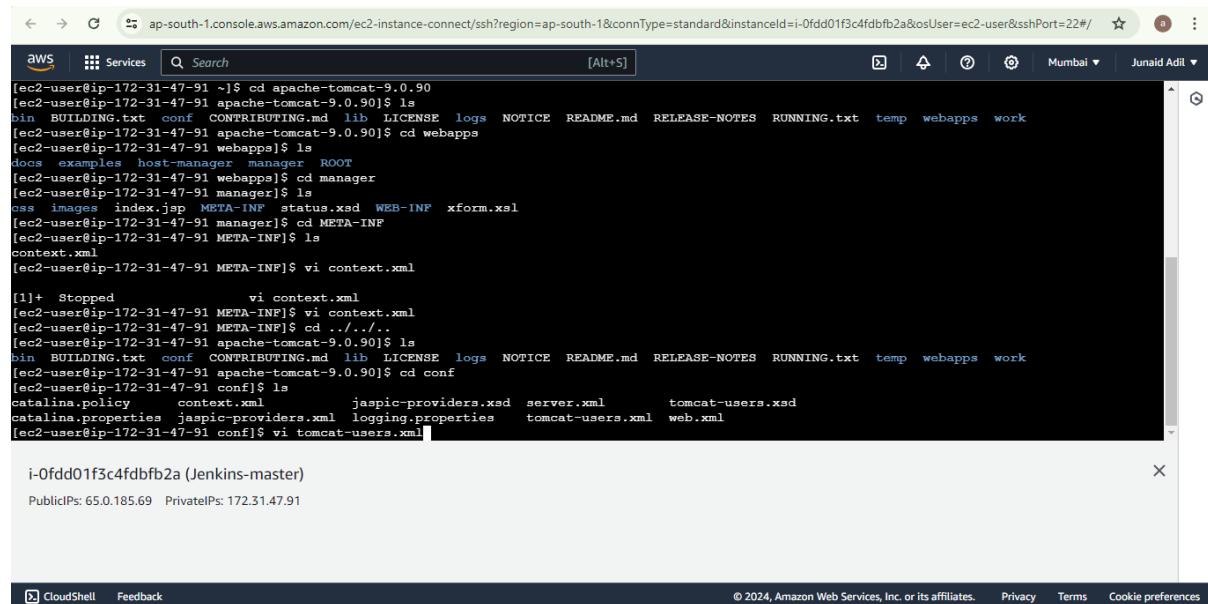
“Cd conf”

The screenshot shows a terminal session in AWS CloudShell. The user has navigated to the 'conf' directory within the Apache Tomcat 9.0.90 configuration. They run 'vi context.xml' to edit the context configuration file. The terminal shows the file being opened and modified, with changes like 'Stopped' and 'Started' status indicators. The session ends with the user exiting the editor.

```
apache-tomcat-9.0.90 apache-tomcat-9.0.90.tar.gz jenkins.repo jenkins.repo.1
[ec2-user@ip-172-31-47-91 ~]$ cd apache-tomcat-9.0.90
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ cd webapps
[ec2-user@ip-172-31-47-91 webapps]$ ls
docs examples host-manager manager ROOT
[ec2-user@ip-172-31-47-91 webapps]$ cd manager
[ec2-user@ip-172-31-47-91 manager]$ ls
css images index.jsp META-INF status.xsd WEB-INF xform.xsl
[ec2-user@ip-172-31-47-91 manager]$ cd META-INF
[ec2-user@ip-172-31-47-91 META-INF]$ ls
context.xml
[ec2-user@ip-172-31-47-91 META-INF]$ vi context.xml

[1]+ Stopped                  vi context.xml
[ec2-user@ip-172-31-47-91 META-INF]$ vi context.xml
[ec2-user@ip-172-31-47-91 META-INF]$ cd ../../..
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ cd conf
[ec2-user@ip-172-31-47-91 conf]$ ls
catalina.policy      context.xml          jaspic-providers.xsd    server.xml        tomcat-users.xml
catalina.properties  jaspic-providers.xml  logging.properties     tomcat-users.xml  web.xml
[ec2-user@ip-172-31-47-91 conf]$
```

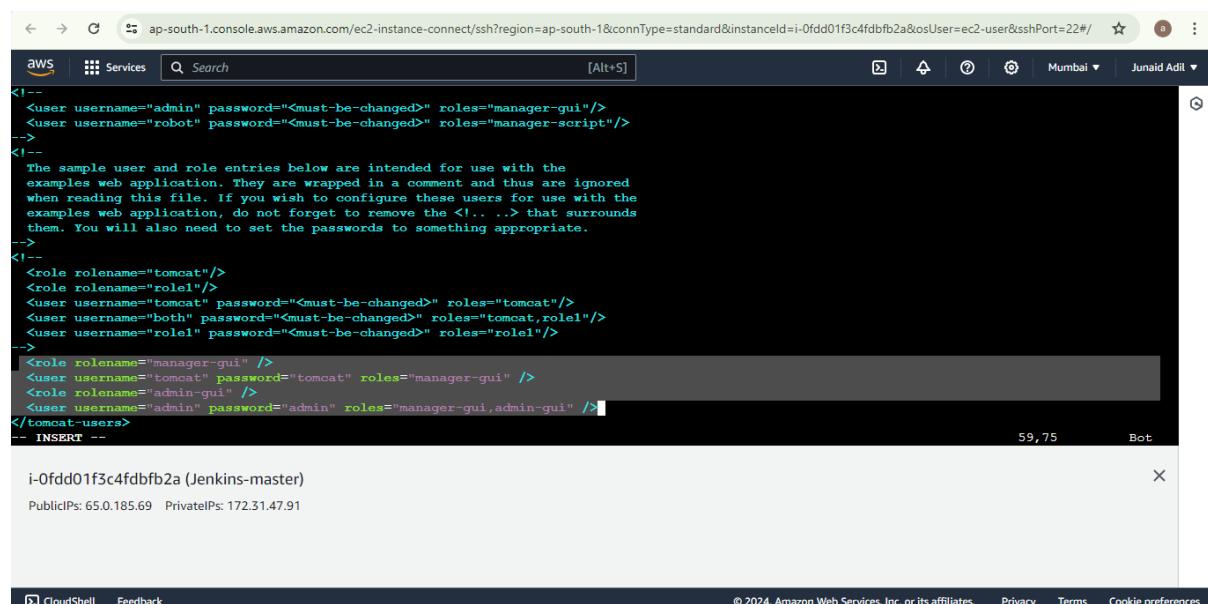
Step-14: To edit the Tomcat-users.xml, use command “vi Tomcat-users.xml”



```
[ec2-user@ip-172-31-47-91 ~]$ cd apache-tomcat-9.0.90
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ cd webapps
[ec2-user@ip-172-31-47-91 webapps]$ ls
docs examples host-manager manager ROOT
[ec2-user@ip-172-31-47-91 webapps]$ cd manager
[ec2-user@ip-172-31-47-91 manager]$ ls
css images index.jsp META-INF status.xsd WEB-INF xform.xsl
[ec2-user@ip-172-31-47-91 manager]$ cd META-INF
[ec2-user@ip-172-31-47-91 META-INF]$ ls
context.xml
[ec2-user@ip-172-31-47-91 META-INF]$ vi context.xml

[1]+ Stopped                  vi context.xml
[ec2-user@ip-172-31-47-91 META-INF]$ vi context.xml
[ec2-user@ip-172-31-47-91 META-INF]$ cd ../../..
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ cd conf
[ec2-user@ip-172-31-47-91 conf]$ ls
catalina.policy      context.xml          jaspic-providers.xsd    server.xml        tomcat-users.xml
catalina.properties   jaspic-providers.xml  logging.properties     tomcat-users.xml  web.xml
[ec2-user@ip-172-31-47-91 conf]$ vi tomcat-users.xml
```

Step-15: We need to add the credentials/users and roles along with manager-script role here as shown below



```
<!--
<user username="admin" password="" roles="manager-gui"/>
<user username="robot" password="" roles="manager-script"/>
-->
<!--
The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!...> that surrounds
them. You will also need to set the passwords to something appropriate.
-->
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="" roles="tomcat"/>
<user username="both" password="" roles="tomcat,role1"/>
<user username="role1" password="" roles="role1"/>
-->
<role rolename="manager-gui" />
<user username="tomcat" password="tomcat" roles="manager-gui" />
<role rolename="admin-gui" />
<user username="admin" password="admin" roles="manager-gui,admin-gui" />
</tomcat-users>
-- INSERT --
```

```

<!--
  The sample user and role entries below are intended for use with the
  examples web application. They are wrapped in a comment and thus are ignored
  when reading this file. If you wish to configure these users for use with the
  examples web application, do not forget to remove the <!-- ... --> that surrounds
  them. You will also need to set the passwords to something appropriate.
-->
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="" roles="tomcat"/>
<user username="both" password="" roles="tomcat,role1"/>
<user username="role1" password="" roles="role1"/>
-->
<role rolename="manager-gui" />
<user username="tomcat" password="tomcat" roles="manager-gui" />
<role rolename="admin-gui" />
<role rolename="manager-gui" />
<user username="admin" password="admin" roles="manager-gui,admin-gui,manager-script" />
</tomcat-users>
-- INSERT --

```

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

Activate Windows
Go to Settings to activate Windows.

[CloudShell](#) [Feedback](#)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Step-16: To start the Tomcat we have to execute “startup.sh” file.

Go to “cd bin”

```

bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ cd conf
[ec2-user@ip-172-31-47-91 conf]$ ls
catalina.policy context.xml jaspic-providers.xml server.xml tomcat-users.xml
catalina.properties jaspic-providers.xml logging.properties tomcat-users.xml web.xml
[ec2-user@ip-172-31-47-91 conf]$ vi tomcat-users.xml
[ec2-user@ip-172-31-47-91 conf]$ ls
catalina.policy context.xml jaspic-providers.xml server.xml tomcat-users.xml
catalina.properties jaspic-providers.xml logging.properties tomcat-users.xml web.xml
[ec2-user@ip-172-31-47-91 conf]$ cd ../..
[ec2-user@ip-172-31-47-91 ~]$ ls
apache-tomcat-9.0.90 apache-tomcat-9.0.90.tar.gz jenkins.repo jenkins.repo.1
[ec2-user@ip-172-31-47-91 ~]$ cd apache-tomcat-9.0.90
-bash: cd: apache-tomcat-9.0.9.: No such file or directory
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ cd bin
[ec2-user@ip-172-31-47-91 bin]$ ls
bootstrap.jar ciphers.bat configtest.bat digest.sh setclasspath.sh startup.sh tool-wrapper.sh
catalina.bat ciphers.sh configtest.sh makebase.bat shutdown.bat tomcat-juli.jar version.bat
catalina.sh Commons-daemon.jar daemon.sh makebase.sh shutdown.sh tomcat-native.tar.gz version.sh
catalina-tasks.xml commons-daemon-native.tar.gz digest.bat setclasspath.bat startup.bat tool-wrapper.bat
[ec2-user@ip-172-31-47-91 bin]$ 

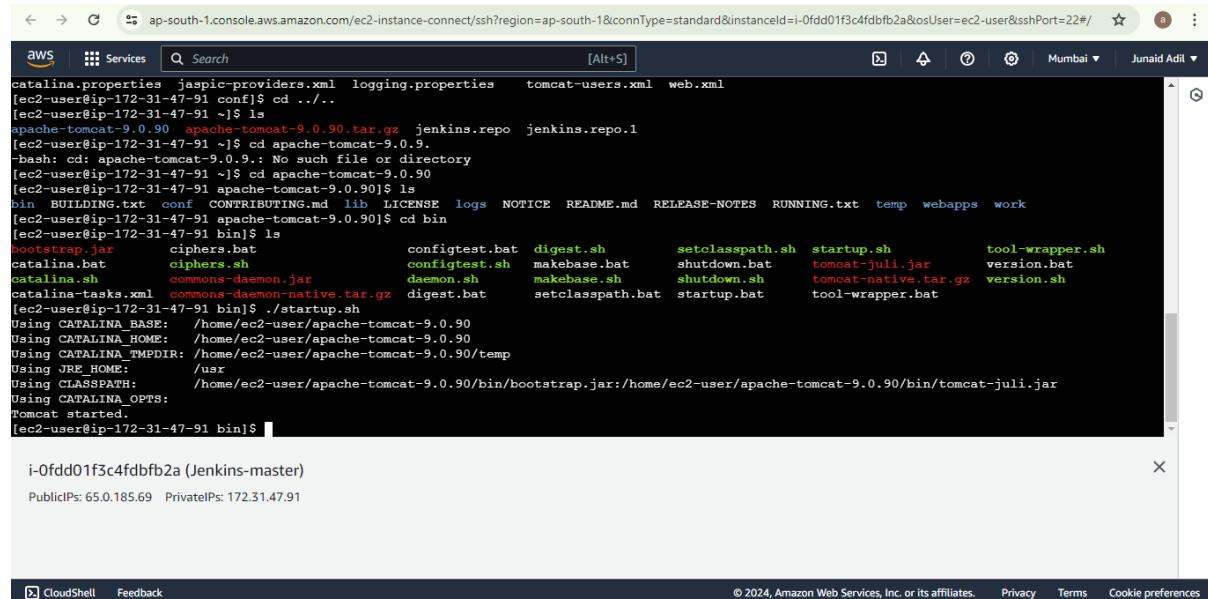
```

i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

[CloudShell](#) [Feedback](#)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Step-17: To execute stratup.sh file run command “./startup.sh “



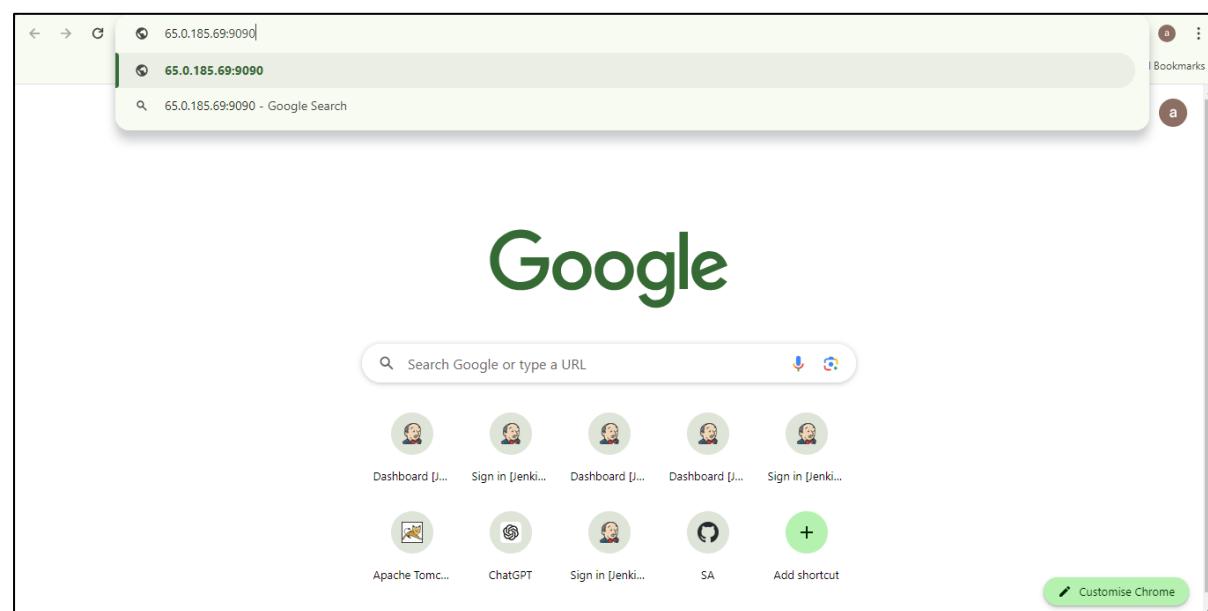
```
catalina.properties jaspic-providers.xml logging.properties tomcat-users.xml web.xml
[ec2-user@ip-172-31-47-91 conf]$ cd ../../
[ec2-user@ip-172-31-47-91 ~]$ ls
apache-tomcat-9.0.90 apache-tomcat-9.0.90.tar.gz jenkins.repo jenkins.repo.1
[ec2-user@ip-172-31-47-91 ~]$ cd apache-tomcat-9.0.90
[bash: cd: apache-tomcat-9.0.90: No such file or directory]
[ec2-user@ip-172-31-47-91 ~]$ cd apache-tomcat-9.0.90
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[ec2-user@ip-172-31-47-91 apache-tomcat-9.0.90]$ cd bin
[ec2-user@ip-172-31-47-91 bin]$ ls
bootstrap.jar ciphers.bat configtest.bat digest.sh setclasspath.sh startup.sh tool-wrapper.sh
catalina.bat ciphers.sh configtest.sh makebase.bat shutdown.bat tomcat-juli.jar version.bat
catalina.sh commons-daemon.jar daemon.sh makebase.sh shutdown.sh tomcat-native.tar.gz version.sh
catalina-tasks.xml commons-daemon-native.tar.gz digest.bat setclasspath.bat startup.bat tool-wrapper.bat
[ec2-user@ip-172-31-47-91 bin]$ ./startup.sh
Using CATALINA_BASE: /home/ec2-user/apache-tomcat-9.0.90
Using CATALINA_HOME: /home/ec2-user/apache-tomcat-9.0.90
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.90/temp
Using JRE_HOME: /usr
Using CLASSPATH: /home/ec2-user/apache-tomcat-9.0.90/bin/bootstrap.jar:/home/ec2-user/apache-tomcat-9.0.90/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[ec2-user@ip-172-31-47-91 bin]$ 
```

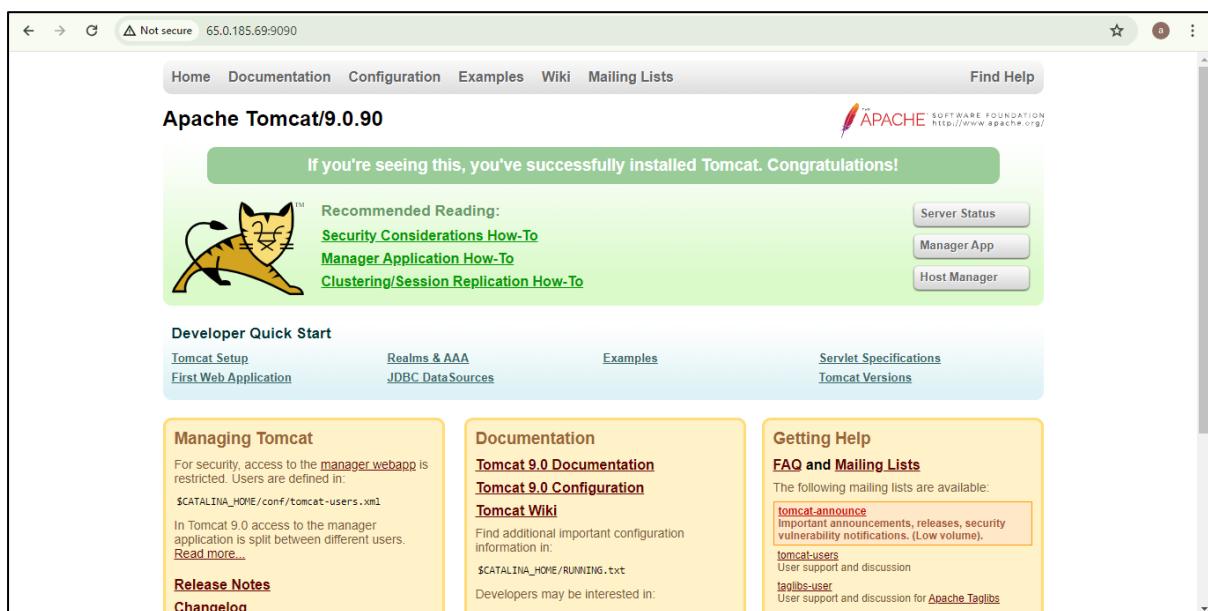
i-0fdd01f3c4fdbfb2a (Jenkins-master)
PublicIPs: 65.0.185.69 PrivateIPs: 172.31.47.91

We can see Tomcat has started.

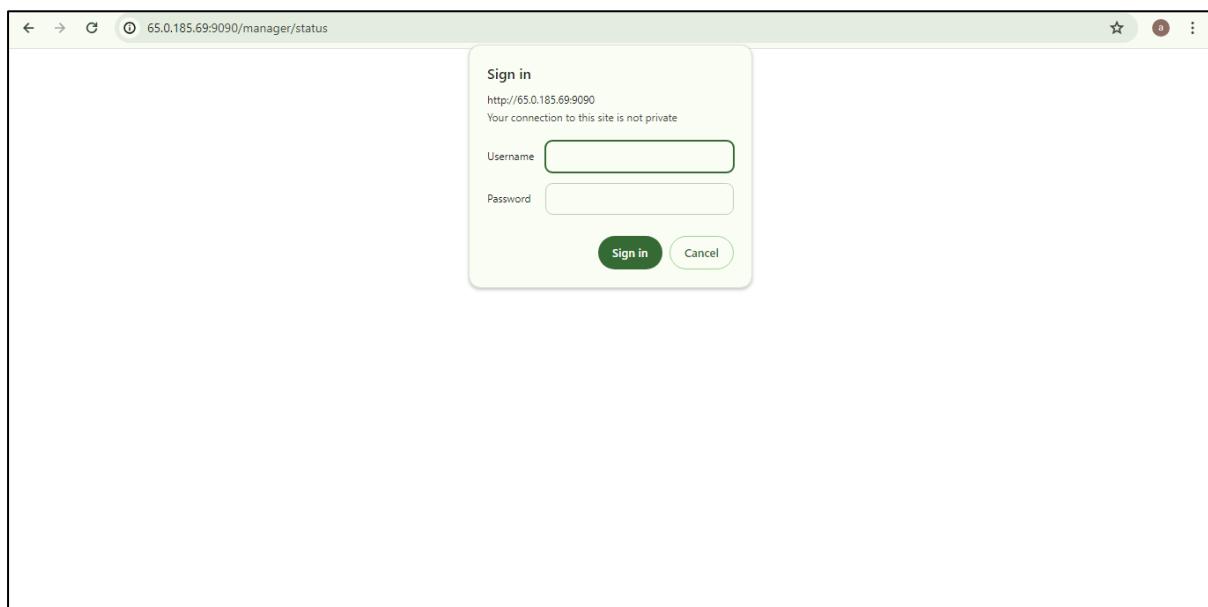
Step-18: Now copy public IP and run in browser along with port number 9090,

“65.0.185.69:9090”

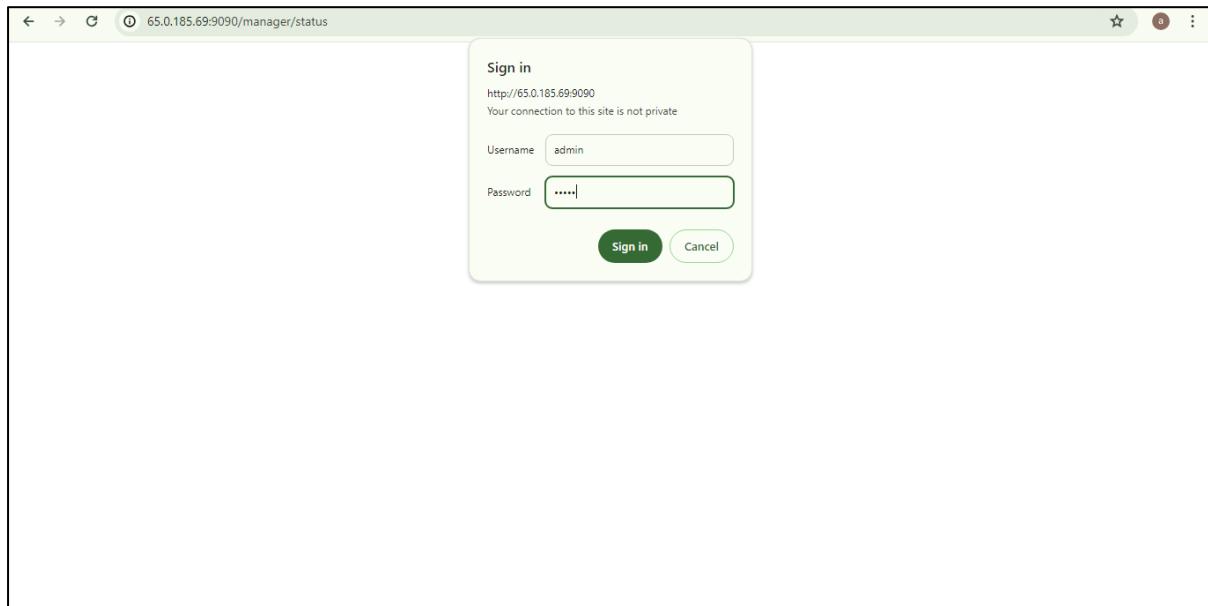




Step-19: To access tomcat we have to give login credentials which were setup in Tomcat-users.xml file



Username = admin, password = admin



Logged in to Tomcat

Manager

List Applications HTML Manager Help Manager Help Complete Server Status

Server Information

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/9.0.90	17.0.11+9-LTS	Amazon.com Inc.	Linux	5.10.218-208.862.amzn2.x86_64	amd64	ip-172-31-47-91.ap-south-1.compute.internal	172.31.47.91

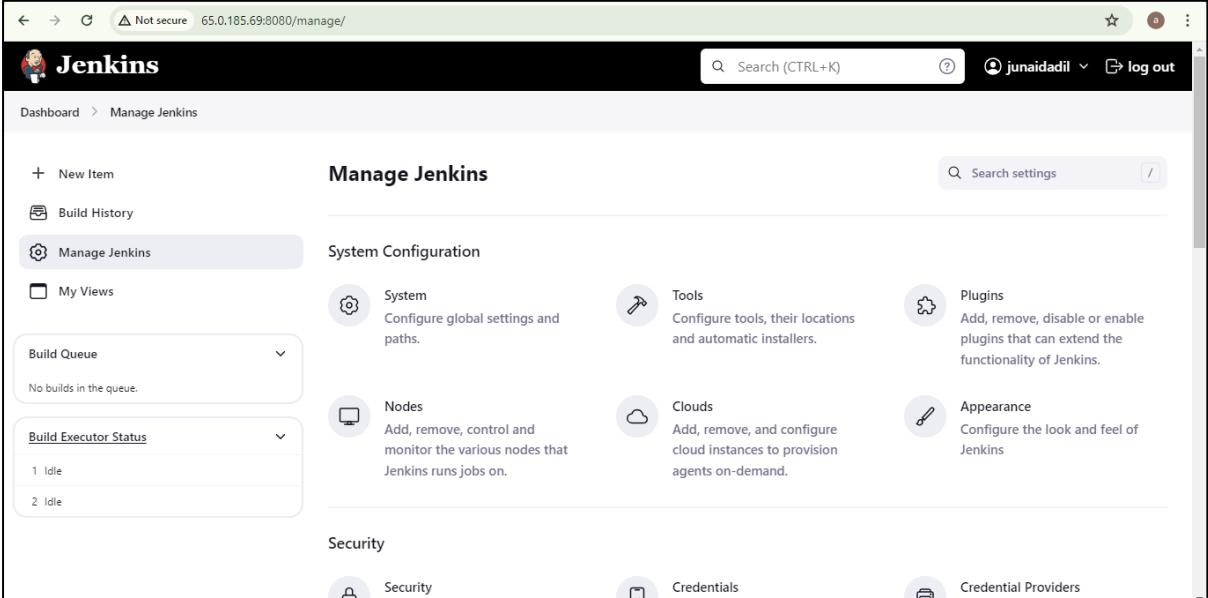
JVM

Free Memory: 8.02 MiB Total Memory: 26.19 MiB Max Memory: 232.00 MiB

Memory Pool	Type	Initial	Total	Maximum	Used
Eden Space	Heap memory	4.31 MiB	7.31 MiB	64.00 MiB	3.82 MiB (5%)
Survivor Space	Heap memory	0.50 MiB	0.87 MiB	8.00 MiB	0.87 MiB (10%)
Tenured Gen	Heap memory	10.68 MiB	18.00 MiB	160.00 MiB	13.46 MiB (8%)
CodeHeap 'non-nmethods'	Non-heap memory	2.43 MiB	2.43 MiB	5.55 MiB	1.20 MiB (21%)
CodeHeap 'non-profiled nmethods'	Non-heap memory	2.43 MiB	2.43 MiB	117.22 MiB	1.46 MiB (1%)
CodeHeap 'profiled nmethods'	Non-heap memory	2.43 MiB	7.81 MiB	117.21 MiB	7.76 MiB (6%)
Compressed Class Space	Non-heap memory	0.00 MiB	2.43 MiB	1024.00 MiB	2.33 MiB (0%)
Metaspace	Non-heap memory	0.00 MiB	24.81 MiB	-0.00 MiB	24.60 MiB

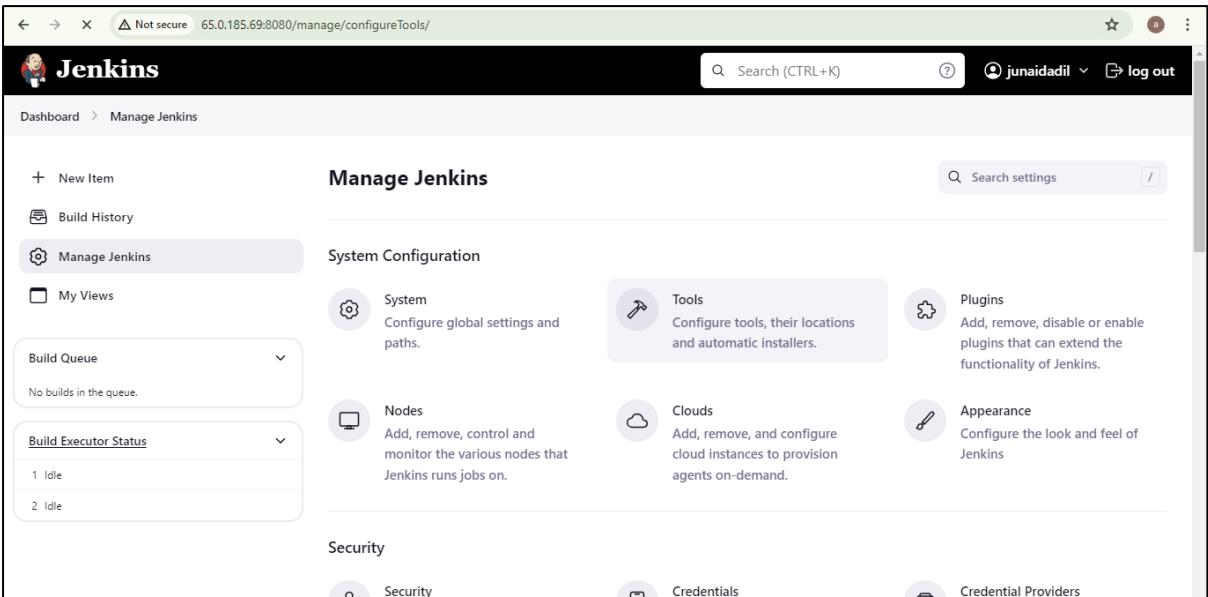
Step-20: Now Go to Jenkins and install pre requisites and create a Job.

Go to manage Jenkins.



The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links like 'New Item', 'Build History', and 'Manage Jenkins' (which is currently selected). The main area is titled 'Manage Jenkins' and has a 'System Configuration' section. This section includes icons and descriptions for 'System', 'Tools', 'Nodes', 'Clouds', and 'Plugins'. Below this is a 'Security' section with links for 'Security', 'Credentials', and 'Credential Providers'. The top right corner shows a search bar and a user profile.

Step-21: Go to Tools → Maven and JDK Installation



This screenshot is similar to the previous one, showing the Jenkins Manage Jenkins page. The 'Tools' item in the 'System Configuration' section is highlighted with a light gray background. The other items in the list ('System', 'Nodes', 'Clouds', 'Plugins', 'Appearance') are shown with their respective icons and descriptions. The rest of the interface, including the sidebar and security links, remains the same.

Step-22: In Maven – give the name and select the version → Apply → Save

The screenshot shows the Jenkins 'Configure Tools' interface. Under the 'Tools' section, there is a 'Maven' configuration. The 'Name' field is set to 'Maven s/w'. The 'Install automatically' checkbox is checked. Under 'Install from Apache', the 'Version' dropdown is set to '3.9.8'. There is also an 'Add Installer' button. At the bottom of the form are 'Save' and 'Apply' buttons.

Step-23: Enter name: Java s/w, 'check'-Install automatically → Apply → Save

The screenshot shows the Jenkins 'Configure Tools' interface. Under the 'Tools' section, there is a 'JDK installations' configuration. It is currently 'Edited'. A new 'JDK' entry is being added, with the 'Name' set to 'Java s/w'. The 'JAVA_HOME' field is empty. The 'Install automatically' checkbox is checked. There is an 'Add Installer' button. At the bottom of the form are 'Save' and 'Apply' buttons. A watermark at the bottom right corner reads 'Activate Windows Go to Settings to activate Windows.'

To deploy web application in Tomcat server we have to install plugin “Deploy to container” in Jenkins.

Step-24: Go to manage Jenkins → Plugins

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links like 'Dashboard', 'Build History', 'Manage Jenkins' (which is selected and highlighted in grey), 'My Views', 'Build Queue' (empty), and 'Build Executor Status' (2 Idle). The main area is titled 'Manage Jenkins' and contains sections for 'System Configuration' and 'Security'. 'System Configuration' includes links for 'System', 'Tools', 'Nodes', 'Clouds', and 'Appearance'. 'Security' includes 'Security' and 'Credentials'. At the bottom right, there's an 'Activate Windows' link. A search bar at the top right says 'Search (CTRL+K)'.

Step-25: Search for plugin and install

The screenshot shows the Jenkins Manage Jenkins > Plugins page. The left sidebar has 'Updates', 'Available plugins' (selected and highlighted in grey), 'Installed plugins', and 'Advanced settings'. The main area is titled 'Plugins' and shows a search bar with 'deploy' and an 'Install' button. It lists available plugins: 'Deploy to container' (version 1.16, released 3 yr 7 mo ago), 'Docker Pipeline' (version 580.vc0c340686b_54, released 29 days ago), and 'Artifactory' (version 4.0.6, released 2 mo 24 days ago). A note at the bottom right says 'Activate Windows'.

The screenshot shows the Jenkins interface for managing plugins. On the left, there's a sidebar with options like 'Updates', 'Available plugins', 'Installed plugins', and 'Advanced settings'. The 'Download progress' option is selected and highlighted with a grey background. The main content area is titled 'Download progress' and shows a list of steps under 'Preparation': 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. Below this, there are three more items: 'SSH server' (Success), 'Deploy to container' (Success), and 'Loading plugin extensions' (Success). At the bottom of the main content, there are two links: 'Go back to the top page' (with a note '(you can start using the installed plugins right away)') and 'Restart Jenkins when installation is complete and no jobs are running'. In the bottom right corner of the main panel, there's a small message: 'Activate Windows Go to Settings to activate Windows. REST API Jenkins 2.452.2'.

Plugin got Installed

Step-26: Jenkins → new item

The screenshot shows the 'Manage Jenkins' page. On the left, there's a sidebar with 'New Item', 'Build History', 'Manage Jenkins' (which is selected and highlighted with a grey background), 'My Views', 'Build Queue' (with a note 'No builds in the queue.'), and 'Build Executor Status' (with a note '1 Idle' and '2 Idle'). The main content area is titled 'Manage Jenkins' and contains several configuration sections: 'System Configuration' (with 'System' and 'Tools' options), 'Security' (with 'Security' and 'Credentials' options), and 'Clouds' (with 'Clouds' and 'Credential Providers' options). There's also a 'Plugins' section which says 'Add, remove, disable or enable plugins that can extend the functionality of Jenkins.' A search bar at the top right says 'Search settings /'.

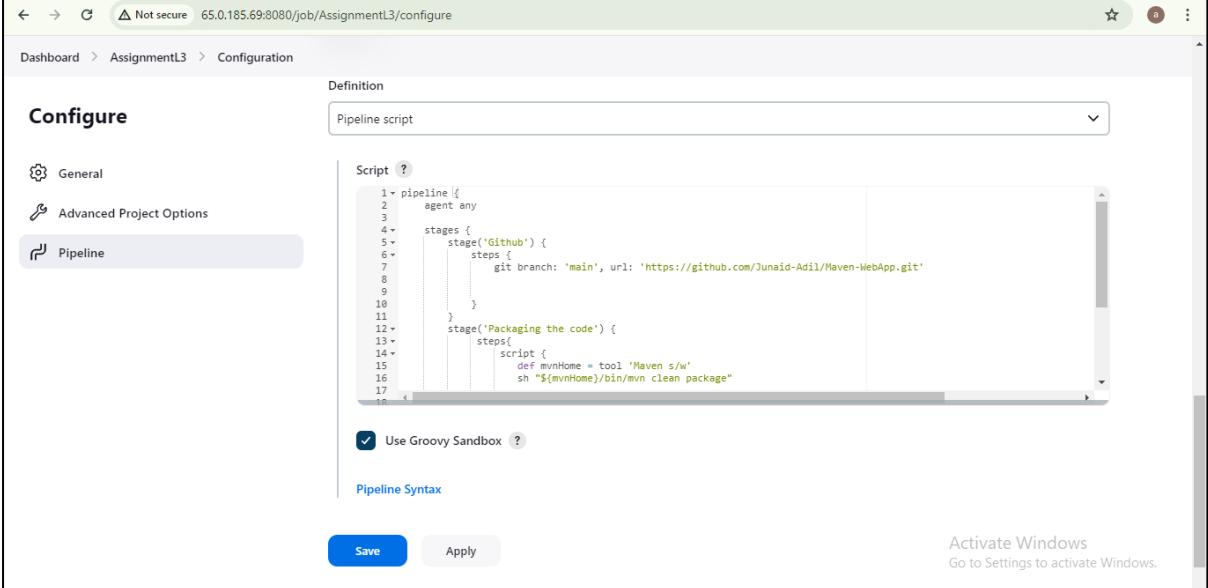
Step-27: Enter name and select project type as pipeline project → Ok

The screenshot shows the Jenkins 'New Job' creation interface. The job name 'AssignmentL3' is entered in the 'Enter an item name' field. Below the name, there are three project type options: 'Freestyle project', 'Pipeline', and 'Multi-configuration project'. The 'Pipeline' option is selected, indicated by a blue border around its icon and text. At the bottom right of the form, there is a large blue 'OK' button.

Step-28: In description enter as “Maven-WebApp”

The screenshot shows the Jenkins 'AssignmentL3' configuration page under the 'General' tab. The 'Description' field contains the text 'Maven-WebApp'. The 'Enabled' switch is turned on. On the left, there is a sidebar with 'Configure' and three sections: 'General', 'Advanced Project Options', and 'Pipeline'. At the bottom of the page are 'Save' and 'Apply' buttons.

Step-29: Write the pipeline script

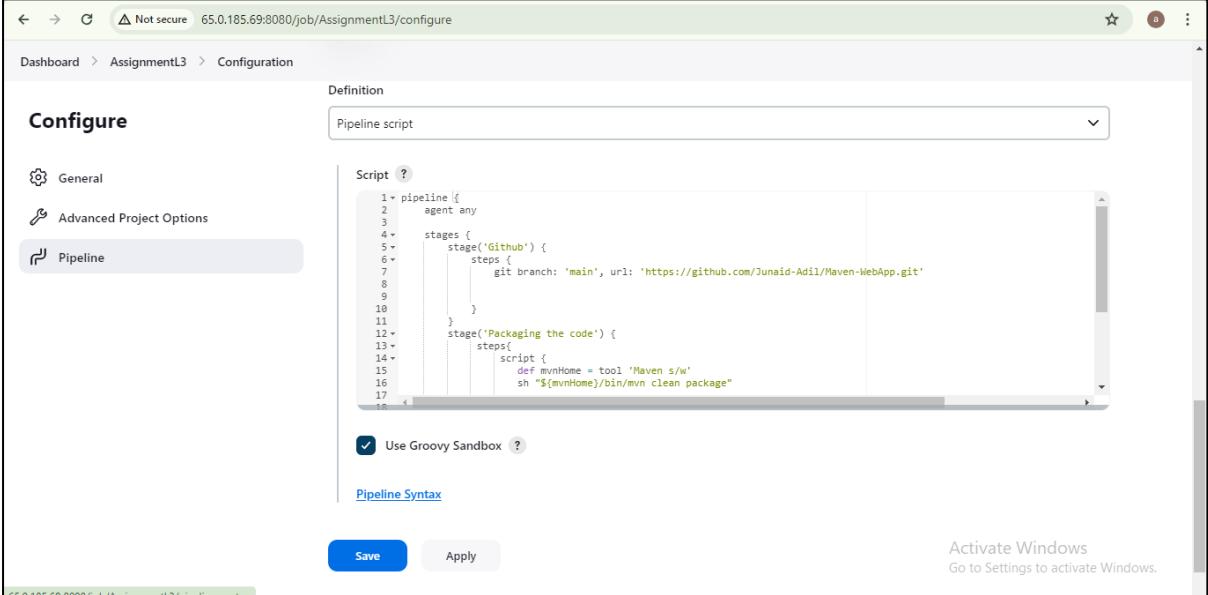


The screenshot shows the Jenkins Pipeline configuration page for a job named 'AssignmentL3'. The 'Pipeline' tab is selected in the sidebar. The main area contains a code editor with the following Groovy pipeline script:

```
1 pipeline {
2     agent any
3
4     stages {
5         stage('Github') {
6             steps {
7                 git branch: 'main', url: 'https://github.com/Junaid-Adil/Maven-WebApp.git'
8
9             }
10        }
11        stage('Packaging the code') {
12            steps{
13                script{
14                    def mvnHome = tool 'Maven s/w'
15                    sh "${mvnHome}/bin/mvn clean package"
16                }
17            }
18        }
19    }
20 }
```

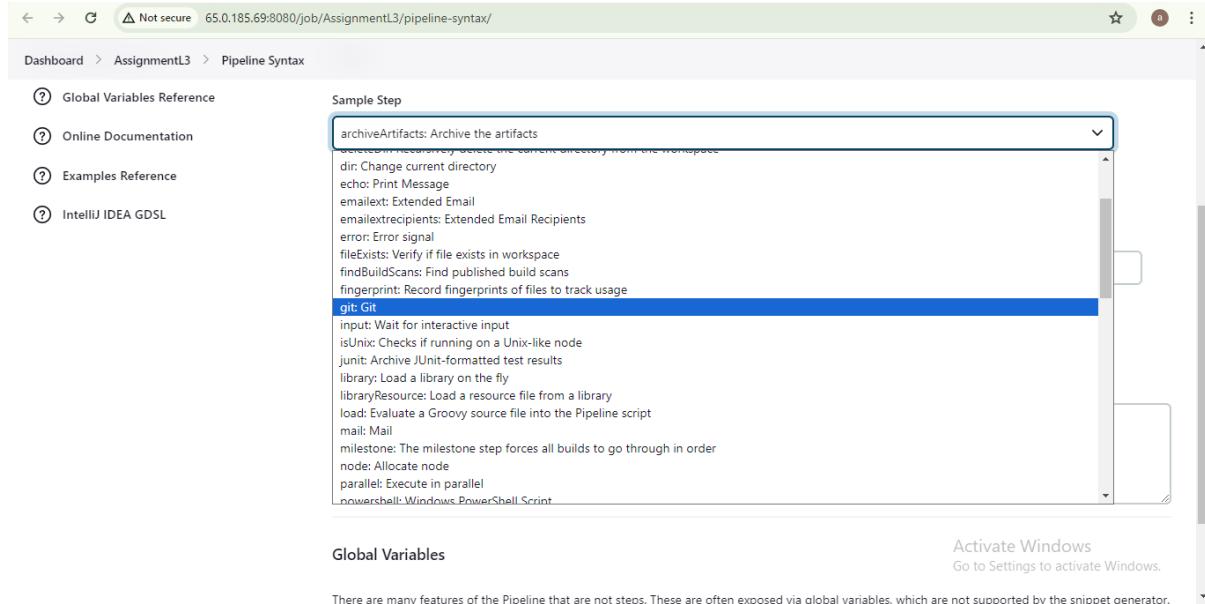
Below the code editor is a checkbox labeled 'Use Groovy Sandbox' which is checked. At the bottom are 'Save' and 'Apply' buttons.

Step-30: Click on pipeline syntax



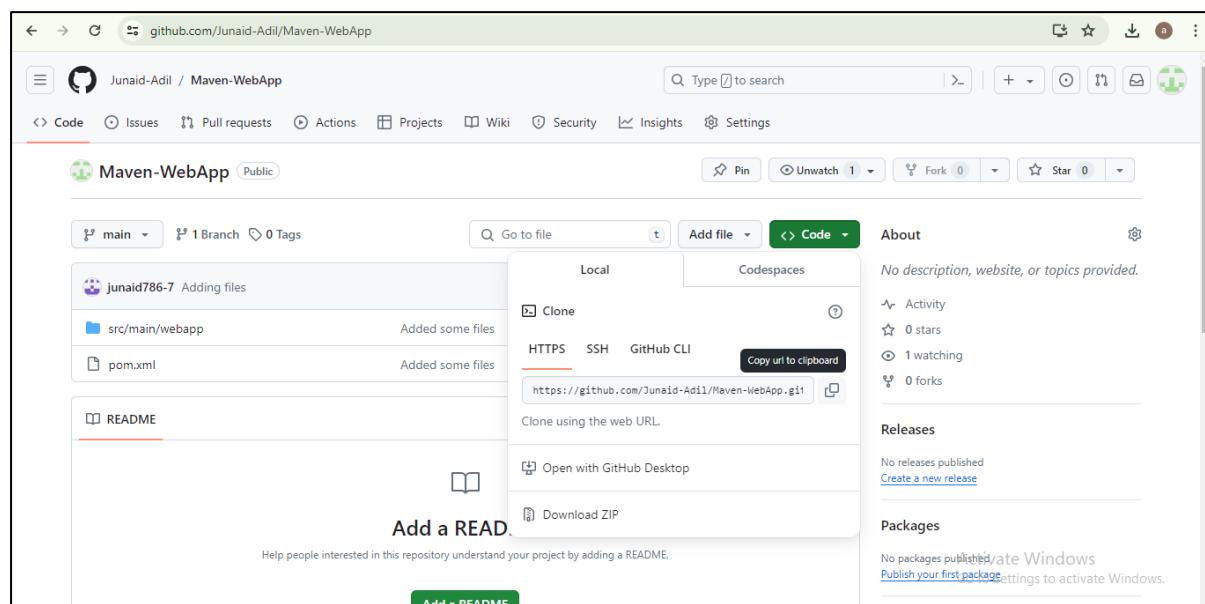
The screenshot shows the same Jenkins Pipeline configuration page as the previous one, but with a blue highlight over the 'Pipeline Syntax' link located below the code editor. The rest of the interface is identical to the first screenshot.

Step-31: In sample steps, as we are taking the source code from GitHub. select git from dropdown list



The screenshot shows a web-based interface for configuring a CI/CD pipeline. On the left, there's a sidebar with links to 'Global Variables Reference', 'Online Documentation', 'Examples Reference', and 'IntelliJ IDEA GDSDL'. The main area is titled 'Sample Step' and contains a dropdown menu with various options. The 'git' option is highlighted with a blue background. Other visible options include 'archiveArtifacts', 'archiveResources', 'dir', 'echo', 'emailExt', 'emailExtRecipients', 'error', 'fileExists', 'findBuildScans', 'fingerprint', 'git', 'input', 'isUnix', 'junit', 'library', 'libraryResource', 'load', 'mail', 'milestone', 'node', 'parallel', and 'powershell'. Below the dropdown, there's a section titled 'Global Variables' and a note about activating Windows.

Step-32: Go to GitHub Repository and copy the HTTPS Link.



The screenshot shows a GitHub repository page for 'Junaid-Adil/Maven-WebApp'. The repository has 1 branch and 0 tags. The 'Code' tab is selected. On the right side, there's a 'Clone' section with options for 'HTTPS', 'SSH', and 'GitHub CLI'. The 'HTTPS' option is highlighted, and a 'Copy url to clipboard' button is visible. The copied URL is <https://github.com/Junaid-Adil/Maven-WebApp.git>. The repository details show 0 stars, 1 watching, and 0 forks. There are sections for 'About', 'Releases', and 'Packages'.

Step-33: Paste it in the pipeline syntax steps.

Enter the repository URL <https://github.com/Junaid-Adil/Maven-WebApp.git>

The screenshot shows the 'Pipeline Syntax' configuration page. On the left, there's a sidebar with links like 'Steps Reference', 'Global Variables Reference', 'Online Documentation', 'Examples Reference', and 'IntelliJ IDEA GDSL'. The main area is titled 'Steps' and contains a 'Sample Step' section. A dropdown menu is open, showing 'git: Git'. Below it, the 'Repository URL' field is populated with 'https://github.com/Junaid-Adil/Maven-WebApp.git'. The 'Branch' field has 'master' selected. Under 'Credentials', there's a dropdown menu with '- none -'. At the bottom of the step configuration, there's a checkbox for 'Include in polling?' which is checked. A message at the bottom right says 'Activate Windows Go to Settings to activate Windows.'

Step-34: Update the branch as per the created repository in Github and as it is public repository no need to give the credentials → click Generate Pipeline Script

This screenshot is similar to the previous one but shows the 'Branch' field changed to 'main'. The 'Credentials' dropdown menu is highlighted with a blue border, indicating it is selected or being focused. The other fields ('Repository URL' and 'Include in polling?') remain the same as in the previous screenshot.

Step-35: Copy the generated pipeline script URL “git branch: 'main', url: 'https://github.com/Junaid-Adil/Maven-WebApp.git' ” and paste it in the script

The screenshot shows a web-based configuration interface for a Jenkins pipeline. The URL is 65.0.185.69:8080/job/AssignmentL3/pipeline-syntax/. The page title is "Pipeline Syntax". It has sections for "Branch" (set to "main"), "Credentials" (set to "- none -"), and checkboxes for "Include in polling?" and "Include in changelog?". A large button labeled "Generate Pipeline Script" is present. Below it, the generated script is displayed in a code editor-like box:
git branch: 'main', url: 'https://github.com/Junaid-Adil/Maven-WebApp.git'

Step-36: Paste it in the Pipeline script stage.

The screenshot shows a web-based configuration interface for a Jenkins pipeline. The URL is 65.0.185.69:8080/job/AssignmentL3/configure. The left sidebar shows "Configure" with tabs for "General", "Advanced Project Options", and "Pipeline" (which is selected). The main area is titled "Definition" with "Pipeline script" selected. The "Script" section contains the following Groovy code:
1 ~ pipeline {
2 ~ agent any
3 ~
4 ~ stages {
5 ~ stage('Github') {
6 ~ steps {
7 ~ git branch: 'main', url: 'https://github.com/Junaid-Adil/Maven-WebApp.git'
8 ~ }
9 ~ }
10 ~ }
11 ~ }
12 ~ stage('Packaging the code') {
13 ~ steps{
14 ~ script {
15 ~ def mvnHome = tool 'Maven s/w'
16 ~ sh "\${mvnHome}/bin/mvn clean package"
17 ~ }
18 ~ }
19 ~ }
20 ~ }
21 ~ }
22 ~ }
23 ~ }
24 ~ }
25 ~ }
26 ~ }
27 ~ }
28 ~ }
29 ~ }
30 ~ }
31 ~ }
32 ~ }
33 ~ }
34 ~ }
35 ~ }
36 ~ }
37 ~ }
38 ~ }
39 ~ }
40 ~ }
41 ~ }
42 ~ }
43 ~ }
44 ~ }
45 ~ }
46 ~ }
47 ~ }
48 ~ }
49 ~ }
50 ~ }
51 ~ }
52 ~ }
53 ~ }
54 ~ }
55 ~ }
56 ~ }
57 ~ }
58 ~ }
59 ~ }
60 ~ }
61 ~ }
62 ~ }
63 ~ }
64 ~ }
65 ~ }
66 ~ }
67 ~ }
68 ~ }
69 ~ }
70 ~ }
71 ~ }
72 ~ }
73 ~ }
74 ~ }
75 ~ }
76 ~ }
77 ~ }
78 ~ }
79 ~ }
80 ~ }
81 ~ }
82 ~ }
83 ~ }
84 ~ }
85 ~ }
86 ~ }
87 ~ }
88 ~ }
89 ~ }
90 ~ }
91 ~ }
92 ~ }
93 ~ }
94 ~ }
95 ~ }
96 ~ }
97 ~ }
98 ~ }
99 ~ }
100 ~ }
101 ~ }
102 ~ }
103 ~ }
104 ~ }
105 ~ }
106 ~ }
107 ~ }
108 ~ }
109 ~ }
110 ~ }
111 ~ }
112 ~ }
113 ~ }
114 ~ }
115 ~ }
116 ~ }
117 ~ }
118 ~ }
119 ~ }
120 ~ }
121 ~ }
122 ~ }
123 ~ }
124 ~ }
125 ~ }
126 ~ }
127 ~ }
128 ~ }
129 ~ }
130 ~ }
131 ~ }
132 ~ }
133 ~ }
134 ~ }
135 ~ }
136 ~ }
137 ~ }
138 ~ }
139 ~ }
140 ~ }
141 ~ }
142 ~ }
143 ~ }
144 ~ }
145 ~ }
146 ~ }
147 ~ }
148 ~ }
149 ~ }
150 ~ }
151 ~ }
152 ~ }
153 ~ }
154 ~ }
155 ~ }
156 ~ }
157 ~ }
158 ~ }
159 ~ }
160 ~ }
161 ~ }
162 ~ }
163 ~ }
164 ~ }
165 ~ }
166 ~ }
167 ~ }
168 ~ }
169 ~ }
170 ~ }
171 ~ }
172 ~ }
173 ~ }
174 ~ }
175 ~ }
176 ~ }
177 ~ }
178 ~ }
179 ~ }
180 ~ }
181 ~ }
182 ~ }
183 ~ }
184 ~ }
185 ~ }
186 ~ }
187 ~ }
188 ~ }
189 ~ }
190 ~ }
191 ~ }
192 ~ }
193 ~ }
194 ~ }
195 ~ }
196 ~ }
197 ~ }
198 ~ }
199 ~ }
200 ~ }
201 ~ }
202 ~ }
203 ~ }
204 ~ }
205 ~ }
206 ~ }
207 ~ }
208 ~ }
209 ~ }
210 ~ }
211 ~ }
212 ~ }
213 ~ }
214 ~ }
215 ~ }
216 ~ }
217 ~ }
218 ~ }
219 ~ }
220 ~ }
221 ~ }
222 ~ }
223 ~ }
224 ~ }
225 ~ }
226 ~ }
227 ~ }
228 ~ }
229 ~ }
230 ~ }
231 ~ }
232 ~ }
233 ~ }
234 ~ }
235 ~ }
236 ~ }
237 ~ }
238 ~ }
239 ~ }
240 ~ }
241 ~ }
242 ~ }
243 ~ }
244 ~ }
245 ~ }
246 ~ }
247 ~ }
248 ~ }
249 ~ }
250 ~ }
251 ~ }
252 ~ }
253 ~ }
254 ~ }
255 ~ }
256 ~ }
257 ~ }
258 ~ }
259 ~ }
260 ~ }
261 ~ }
262 ~ }
263 ~ }
264 ~ }
265 ~ }
266 ~ }
267 ~ }
268 ~ }
269 ~ }
270 ~ }
271 ~ }
272 ~ }
273 ~ }
274 ~ }
275 ~ }
276 ~ }
277 ~ }
278 ~ }
279 ~ }
280 ~ }
281 ~ }
282 ~ }
283 ~ }
284 ~ }
285 ~ }
286 ~ }
287 ~ }
288 ~ }
289 ~ }
290 ~ }
291 ~ }
292 ~ }
293 ~ }
294 ~ }
295 ~ }
296 ~ }
297 ~ }
298 ~ }
299 ~ }
300 ~ }
301 ~ }
302 ~ }
303 ~ }
304 ~ }
305 ~ }
306 ~ }
307 ~ }
308 ~ }
309 ~ }
310 ~ }
311 ~ }
312 ~ }
313 ~ }
314 ~ }
315 ~ }
316 ~ }
317 ~ }
318 ~ }
319 ~ }
320 ~ }
321 ~ }
322 ~ }
323 ~ }
324 ~ }
325 ~ }
326 ~ }
327 ~ }
328 ~ }
329 ~ }
330 ~ }
331 ~ }
332 ~ }
333 ~ }
334 ~ }
335 ~ }
336 ~ }
337 ~ }
338 ~ }
339 ~ }
340 ~ }
341 ~ }
342 ~ }
343 ~ }
344 ~ }
345 ~ }
346 ~ }
347 ~ }
348 ~ }
349 ~ }
350 ~ }
351 ~ }
352 ~ }
353 ~ }
354 ~ }
355 ~ }
356 ~ }
357 ~ }
358 ~ }
359 ~ }
360 ~ }
361 ~ }
362 ~ }
363 ~ }
364 ~ }
365 ~ }
366 ~ }
367 ~ }
368 ~ }
369 ~ }
370 ~ }
371 ~ }
372 ~ }
373 ~ }
374 ~ }
375 ~ }
376 ~ }
377 ~ }
378 ~ }
379 ~ }
380 ~ }
381 ~ }
382 ~ }
383 ~ }
384 ~ }
385 ~ }
386 ~ }
387 ~ }
388 ~ }
389 ~ }
390 ~ }
391 ~ }
392 ~ }
393 ~ }
394 ~ }
395 ~ }
396 ~ }
397 ~ }
398 ~ }
399 ~ }
400 ~ }
401 ~ }
402 ~ }
403 ~ }
404 ~ }
405 ~ }
406 ~ }
407 ~ }
408 ~ }
409 ~ }
410 ~ }
411 ~ }
412 ~ }
413 ~ }
414 ~ }
415 ~ }
416 ~ }
417 ~ }
418 ~ }
419 ~ }
420 ~ }
421 ~ }
422 ~ }
423 ~ }
424 ~ }
425 ~ }
426 ~ }
427 ~ }
428 ~ }
429 ~ }
430 ~ }
431 ~ }
432 ~ }
433 ~ }
434 ~ }
435 ~ }
436 ~ }
437 ~ }
438 ~ }
439 ~ }
440 ~ }
441 ~ }
442 ~ }
443 ~ }
444 ~ }
445 ~ }
446 ~ }
447 ~ }
448 ~ }
449 ~ }
450 ~ }
451 ~ }
452 ~ }
453 ~ }
454 ~ }
455 ~ }
456 ~ }
457 ~ }
458 ~ }
459 ~ }
460 ~ }
461 ~ }
462 ~ }
463 ~ }
464 ~ }
465 ~ }
466 ~ }
467 ~ }
468 ~ }
469 ~ }
470 ~ }
471 ~ }
472 ~ }
473 ~ }
474 ~ }
475 ~ }
476 ~ }
477 ~ }
478 ~ }
479 ~ }
480 ~ }
481 ~ }
482 ~ }
483 ~ }
484 ~ }
485 ~ }
486 ~ }
487 ~ }
488 ~ }
489 ~ }
490 ~ }
491 ~ }
492 ~ }
493 ~ }
494 ~ }
495 ~ }
496 ~ }
497 ~ }
498 ~ }
499 ~ }
500 ~ }
501 ~ }
502 ~ }
503 ~ }
504 ~ }
505 ~ }
506 ~ }
507 ~ }
508 ~ }
509 ~ }
510 ~ }
511 ~ }
512 ~ }
513 ~ }
514 ~ }
515 ~ }
516 ~ }
517 ~ }
518 ~ }
519 ~ }
520 ~ }
521 ~ }
522 ~ }
523 ~ }
524 ~ }
525 ~ }
526 ~ }
527 ~ }
528 ~ }
529 ~ }
530 ~ }
531 ~ }
532 ~ }
533 ~ }
534 ~ }
535 ~ }
536 ~ }
537 ~ }
538 ~ }
539 ~ }
540 ~ }
541 ~ }
542 ~ }
543 ~ }
544 ~ }
545 ~ }
546 ~ }
547 ~ }
548 ~ }
549 ~ }
550 ~ }
551 ~ }
552 ~ }
553 ~ }
554 ~ }
555 ~ }
556 ~ }
557 ~ }
558 ~ }
559 ~ }
560 ~ }
561 ~ }
562 ~ }
563 ~ }
564 ~ }
565 ~ }
566 ~ }
567 ~ }
568 ~ }
569 ~ }
570 ~ }
571 ~ }
572 ~ }
573 ~ }
574 ~ }
575 ~ }
576 ~ }
577 ~ }
578 ~ }
579 ~ }
580 ~ }
581 ~ }
582 ~ }
583 ~ }
584 ~ }
585 ~ }
586 ~ }
587 ~ }
588 ~ }
589 ~ }
590 ~ }
591 ~ }
592 ~ }
593 ~ }
594 ~ }
595 ~ }
596 ~ }
597 ~ }
598 ~ }
599 ~ }
600 ~ }
601 ~ }
602 ~ }
603 ~ }
604 ~ }
605 ~ }
606 ~ }
607 ~ }
608 ~ }
609 ~ }
610 ~ }
611 ~ }
612 ~ }
613 ~ }
614 ~ }
615 ~ }
616 ~ }
617 ~ }
618 ~ }
619 ~ }
620 ~ }
621 ~ }
622 ~ }
623 ~ }
624 ~ }
625 ~ }
626 ~ }
627 ~ }
628 ~ }
629 ~ }
630 ~ }
631 ~ }
632 ~ }
633 ~ }
634 ~ }
635 ~ }
636 ~ }
637 ~ }
638 ~ }
639 ~ }
640 ~ }
641 ~ }
642 ~ }
643 ~ }
644 ~ }
645 ~ }
646 ~ }
647 ~ }
648 ~ }
649 ~ }
650 ~ }
651 ~ }
652 ~ }
653 ~ }
654 ~ }
655 ~ }
656 ~ }
657 ~ }
658 ~ }
659 ~ }
660 ~ }
661 ~ }
662 ~ }
663 ~ }
664 ~ }
665 ~ }
666 ~ }
667 ~ }
668 ~ }
669 ~ }
670 ~ }
671 ~ }
672 ~ }
673 ~ }
674 ~ }
675 ~ }
676 ~ }
677 ~ }
678 ~ }
679 ~ }
680 ~ }
681 ~ }
682 ~ }
683 ~ }
684 ~ }
685 ~ }
686 ~ }
687 ~ }
688 ~ }
689 ~ }
690 ~ }
691 ~ }
692 ~ }
693 ~ }
694 ~ }
695 ~ }
696 ~ }
697 ~ }
698 ~ }
699 ~ }
700 ~ }
701 ~ }
702 ~ }
703 ~ }
704 ~ }
705 ~ }
706 ~ }
707 ~ }
708 ~ }
709 ~ }
710 ~ }
711 ~ }
712 ~ }
713 ~ }
714 ~ }
715 ~ }
716 ~ }
717 ~ }
718 ~ }
719 ~ }
720 ~ }
721 ~ }
722 ~ }
723 ~ }
724 ~ }
725 ~ }
726 ~ }
727 ~ }
728 ~ }
729 ~ }
730 ~ }
731 ~ }
732 ~ }
733 ~ }
734 ~ }
735 ~ }
736 ~ }
737 ~ }
738 ~ }
739 ~ }
740 ~ }
741 ~ }
742 ~ }
743 ~ }
744 ~ }
745 ~ }
746 ~ }
747 ~ }
748 ~ }
749 ~ }
750 ~ }
751 ~ }
752 ~ }
753 ~ }
754 ~ }
755 ~ }
756 ~ }
757 ~ }
758 ~ }
759 ~ }
760 ~ }
761 ~ }
762 ~ }
763 ~ }
764 ~ }
765 ~ }
766 ~ }
767 ~ }
768 ~ }
769 ~ }
770 ~ }
771 ~ }
772 ~ }
773 ~ }
774 ~ }
775 ~ }
776 ~ }
777 ~ }
778 ~ }
779 ~ }
780 ~ }
781 ~ }
782 ~ }
783 ~ }
784 ~ }
785 ~ }
786 ~ }
787 ~ }
788 ~ }
789 ~ }
790 ~ }
791 ~ }
792 ~ }
793 ~ }
794 ~ }
795 ~ }
796 ~ }
797 ~ }
798 ~ }
799 ~ }
800 ~ }
801 ~ }
802 ~ }
803 ~ }
804 ~ }
805 ~ }
806 ~ }
807 ~ }
808 ~ }
809 ~ }
810 ~ }
811 ~ }
812 ~ }
813 ~ }
814 ~ }
815 ~ }
816 ~ }
817 ~ }
818 ~ }
819 ~ }
820 ~ }
821 ~ }
822 ~ }
823 ~ }
824 ~ }
825 ~ }
826 ~ }
827 ~ }
828 ~ }
829 ~ }
830 ~ }
831 ~ }
832 ~ }
833 ~ }
834 ~ }
835 ~ }
836 ~ }
837 ~ }
838 ~ }
839 ~ }
840 ~ }
841 ~ }
842 ~ }
843 ~ }
844 ~ }
845 ~ }
846 ~ }
847 ~ }
848 ~ }
849 ~ }
850 ~ }
851 ~ }
852 ~ }
853 ~ }
854 ~ }
855 ~ }
856 ~ }
857 ~ }
858 ~ }
859 ~ }
860 ~ }
861 ~ }
862 ~ }
863 ~ }
864 ~ }
865 ~ }
866 ~ }
867 ~ }
868 ~ }
869 ~ }
870 ~ }
871 ~ }
872 ~ }
873 ~ }
874 ~ }
875 ~ }
876 ~ }
877 ~ }
878 ~ }
879 ~ }
880 ~ }
881 ~ }
882 ~ }
883 ~ }
884 ~ }
885 ~ }
886 ~ }
887 ~ }
888 ~ }
889 ~ }
890 ~ }
891 ~ }
892 ~ }
893 ~ }
894 ~ }
895 ~ }
896 ~ }
897 ~ }
898 ~ }
899 ~ }
900 ~ }
901 ~ }
902 ~ }
903 ~ }
904 ~ }
905 ~ }
906 ~ }
907 ~ }
908 ~ }
909 ~ }
910 ~ }
911 ~ }
912 ~ }
913 ~ }
914 ~ }
915 ~ }
916 ~ }
917 ~ }
918 ~ }
919 ~ }
920 ~ }
921 ~ }
922 ~ }
923 ~ }
924 ~ }
925 ~ }
926 ~ }
927 ~ }
928 ~ }
929 ~ }
930 ~ }
931 ~ }
932 ~ }
933 ~ }
934 ~ }
935 ~ }
936 ~ }
937 ~ }
938 ~ }
939 ~ }
940 ~ }
941 ~ }
942 ~ }
943 ~ }
944 ~ }
945 ~ }
946 ~ }
947 ~ }
948 ~ }
949 ~ }
950 ~ }
951 ~ }
952 ~ }
953 ~ }
954 ~ }
955 ~ }
956 ~ }
957 ~ }
958 ~ }
959 ~ }
960 ~ }
961 ~ }
962 ~ }
963 ~ }
964 ~ }
965 ~ }
966 ~ }
967 ~ }
968 ~ }
969 ~ }
970 ~ }
971 ~ }
972 ~ }
973 ~ }
974 ~ }
975 ~ }
976 ~ }
977 ~ }
978 ~ }
979 ~ }
980 ~ }
981 ~ }
982 ~ }
983 ~ }
984 ~ }
985 ~ }
986 ~ }
987 ~ }
988 ~ }
989 ~ }
990 ~ }
991 ~ }
992 ~ }
993 ~ }
994 ~ }
995 ~ }
996 ~ }
997 ~ }
998 ~ }
999 ~ }

Activate Windows
Go to Settings to activate Windows.

Apply → Save

The screenshot shows the Jenkins Pipeline configuration page for a job named 'AssignmentL3'. The 'Pipeline' tab is selected in the left sidebar. The main area displays a Groovy script for the pipeline definition:

```
1 pipeline {
2     agent any
3
4     stages {
5         stage('Github') {
6             steps {
7                 git branch: 'main', url: 'https://github.com/Junaid-Adil/Maven-WebApp.git'
8
9             }
10        }
11    }
12    stage('Packaging the code') {
13        steps{
14            script {
15                def mvnHome = tool 'Maven s/w'
16                sh "${mvnHome}/bin/mvn clean package"
17            }
18        }
19    }
}
```

A checked checkbox labeled 'Use Groovy Sandbox' is present below the script editor. At the bottom, there are 'Save' and 'Apply' buttons, and a green 'Saved' button is visible on the left.

Step-37: Now click on Build Now

The screenshot shows the Jenkins Pipeline status page for the 'AssignmentL3' job. The 'Status' tab is selected in the left sidebar. The main area displays the pipeline configuration and a 'Stage View' section which states 'No data available. This Pipeline has not yet run.' Below it is a 'Permalinks' section. On the left, there is a sidebar with various project management options like 'Changes', 'Build Now', and 'Configure'. A prominent blue 'Build Now' button is located in the center of the page.

In build history we can see the log

The screenshot shows the Jenkins AssignmentL3 job dashboard. On the left, there's a sidebar with various project management options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, and Pipeline Syntax. Below this is a 'Build History' section with a dropdown set to 'trend'. A single build entry is listed: '#1 20 Jun 2024, 11:34'. The main area is titled 'AssignmentL3' and shows a 'Stage View' with three stages: 'Github', 'Packaging the code', and 'Deploy'. The 'Github' stage has an average time of 825ms. The 'Packaging the code' stage has an average time of 9s. The 'Deploy' stage has an average time of 399ms. The total average full run time is approximately 14s. The 'Stage View' also displays specific times for each build: #11 (Github: 1s, Packaging: 8s, Deploy: 320ms) and #10 (Github: 775ms, Packaging: 26s, Deploy: 352ms). At the bottom right, there's a message to 'Activate Windows'.

Step-38: Build got succeeded. Go to console output to check the details

This screenshot is identical to the one above, showing the Jenkins AssignmentL3 job dashboard. The key difference is the status indicator at the top center, which now shows a green checkmark next to the project name 'AssignmentL3', indicating that the latest build (#11) has succeeded. The rest of the interface, including the sidebar, Stage View, and build history details, remains the same.

Not secure 65.0.185.69:8080/job/AssignmentL3/11/console

Dashboard > AssignmentL3 > #11

```

> Authorization: Basic YmRtaW46YmRtaW4=
> User-Agent: curl/8.3.0
> Accept: /*
> Content-Length: 110596
>
} [44888 bytes data]
* We are completely uploaded and fine
< HTTP/1.1 200
< Cache-Control: private
< X-Frame-Options: DENY
< X-Content-Type-Options: nosniff
< Content-Type: text/plain;charset=utf-8
< Transfer-Encoding: chunked
< Date: Thu, 20 Jun 2024 13:35:35 GMT
<
{ [69 bytes data]

100 108k 0 58 100 108k 820 1527k --:--:--:--:--:-- 1543k
* Connection #0 to host 65.0.185.69 left intact
OK - Deployed application at context path [/Maven-WebApp]
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Activate Windows
Go to Settings to activate Windows.

Step-39: Go to Tomcat → List Application

Not secure 65.0.185.69:9090/manager/status




Server Status

Manager							
List Applications	HTML Manager Help		Manager Help		Complete Server Status		
Server Information							
Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/9.0.90	17.0.11+9-LTS	Amazon.com Inc.	Linux	5.10.218-208.862.amzn2.x86_64	amd64	ip-172-31-47-91.ap-south-1.compute.internal	172.31.47.91

JVM

Free Memory: 6.23 MiB Total Memory: 26.19 MiB Max Memory: 232.00 MiB

Memory Pool	Type	Initial	Total	Maximum	Used
Eden Space	Heap memory	4.31 MiB	7.31 MiB	64.00 MiB	4.84 MiB (7%)
Survivor Space	Heap memory	0.50 MiB	0.87 MiB	8.00 MiB	0.23 MiB (2%)
Tenured Gen	Heap memory	10.68 MiB	18.00 MiB	160.00 MiB	14.87 MiB (9%)
CodeHeap 'non-nmethods'	Non-heap memory	2.43 MiB	2.43 MiB	5.55 MiB	1.20 MiB (21%)
CodeHeap 'non-profiled nmethods'	Non-heap memory	2.43 MiB	2.43 MiB	117.22 MiB	2.06 MiB (1%)
CodeHeap 'profiled nmethods'	Non-heap memory	2.43 MiB	9.62 MiB	117.21 MiB	9.60 MiB (8%)
Compressed Class Space	Non-heap memory	0.00 MiB	2.50 MiB	1024.00 MiB	Settings to 2.35 MiB (0%)
Metaspace	Non-heap memory	0.00 MiB	25.81 MiB	-0.00 MiB	25.48 MiB

65.0.185.69:9090/manager/html/list

The screenshot shows the Tomcat Manager application list page. At the top, there is a message bar with 'Message: OK'. Below it, the title 'Manager' is displayed. The page has four tabs: 'List Applications' (selected), 'HTML Manager Help', 'Manager Help', and 'Server Status'. A search bar is present above the table. The table has a header row 'Applications' with columns: Path, Version, Display Name, Running, Sessions, and Commands. The 'Commands' column contains buttons for Start, Stop, Reload, Undeploy, and a session expiration dropdown set to 30 minutes. The table lists several applications:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> Expire sessions with idle ≥ [30] minutes
/Maven-WebApp	None specified	Archetype Created Web Application	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> Expire sessions with idle ≥ [30] minutes
/docs	None specified	Tomcat Documentation	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> Expire sessions with idle ≥ [30] minutes
/examples	None specified	Servlet and JSP Examples	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> Expire sessions with idle ≥ [30] minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> Expire sessions with idle ≥ [30] minutes
/manager	None specified	Tomcat Manager Application	true	1	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> Expire sessions with idle ≥ [30] minutes
/yourapp	None specified	Archetype Created Web Application	true	1	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> Expire sessions with idle ≥ [30] minutes Go to Activate Windows

We can see the “Maven-WebApp” application has been deployed

Output:

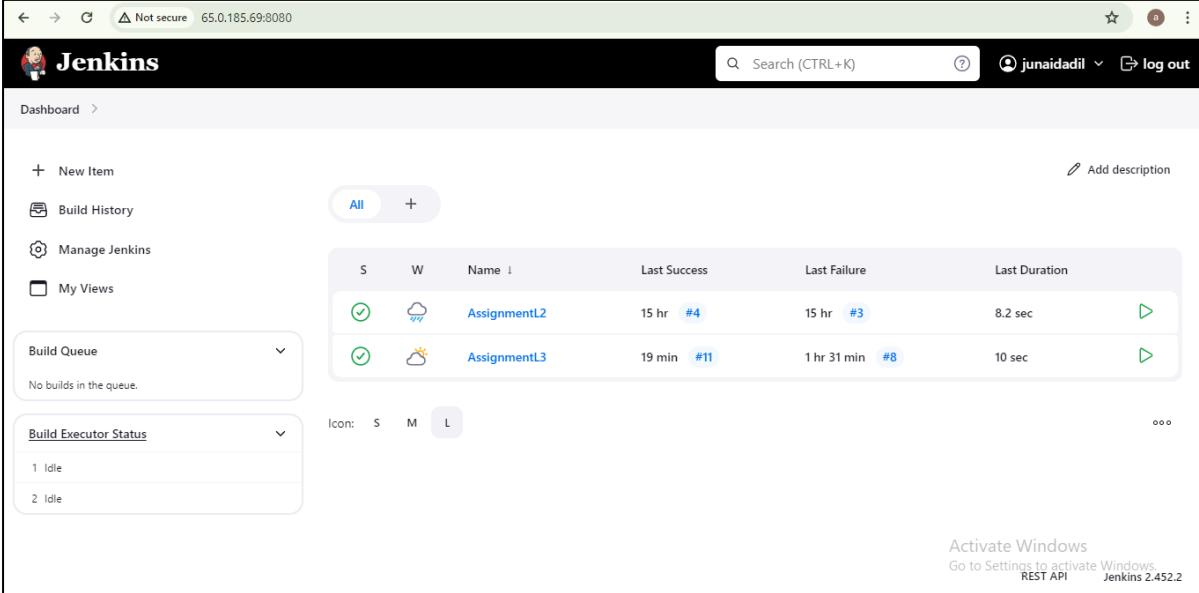
The screenshot shows the deployed Maven-WebApp at the URL 65.0.185.69:9090/Maven-WebApp/. The page displays the following content:

Hello, This is Junaid!
Iam from Hyderabad
Taking DevOps Training in StarAgile

Activate Windows
Go to Settings to activate Windows.

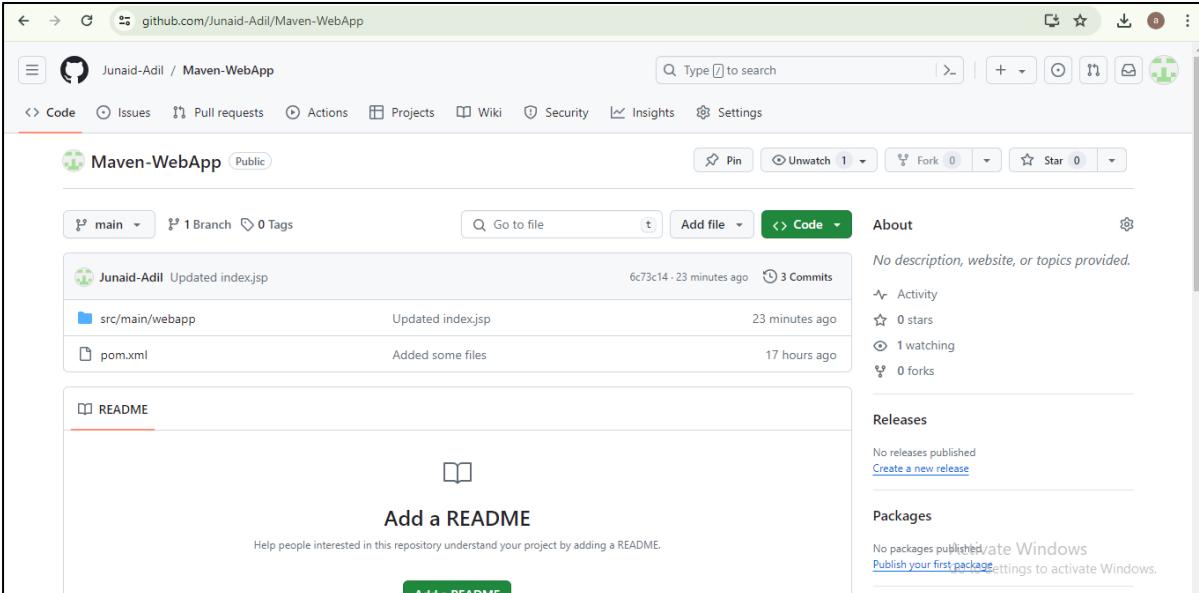
L4 - Automate the CICD Pipeline using GitHub webhook and Poll-SCM

Step-1: Install Jenkins



The screenshot shows the Jenkins dashboard at <http://65.0.185.69:8080>. The left sidebar includes links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. The main area displays a table of builds with columns for Status (S), Workstation (W), Name, Last Success, Last Failure, and Last Duration. Two builds are listed: 'AssignmentL2' (Status S, Last Success 15 hr #4, Last Failure 15 hr #3, Last Duration 8.2 sec) and 'AssignmentL3' (Status S, Last Success 19 min #11, Last Failure 1 hr 31 min #8, Last Duration 10 sec). A message at the bottom right says 'Activate Windows'.

Step-2: Create a Github Repository with the source code init.



The screenshot shows the GitHub repository page for 'Junaid-Adil/Maven-WebApp'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository name 'Maven-WebApp' is shown with a green icon. The main content area shows a commit history with three commits from 'Junaid-Adil': 'Updated index.jsp' (6c73c14, 23 minutes ago), 'src/main/webapp' (Updated index.jsp, 23 minutes ago), and 'pom.xml' (Added some files, 17 hours ago). Below the commits is a 'README' section with a 'Add a README' button. The right sidebar contains sections for About (No description, website, or topics provided), Activity (0 stars, 1 watching, 0 forks), Releases (No releases published), Packages (No packages published), and a note about activating Windows.

Step-3: Create a Webhook → Go to GitHub → settings

The screenshot shows a GitHub repository page for 'Junaid-Adil / Maven-WebApp'. The main content area displays recent commits:

- Junaid-Adil Updated index.jsp (6c73c14 · 30 minutes ago)
- src/main/webapp Updated index.jsp (30 minutes ago)
- pom.xml Added some files (17 hours ago)

Below the commits, there is a 'README' section with a placeholder 'Add a README' button.

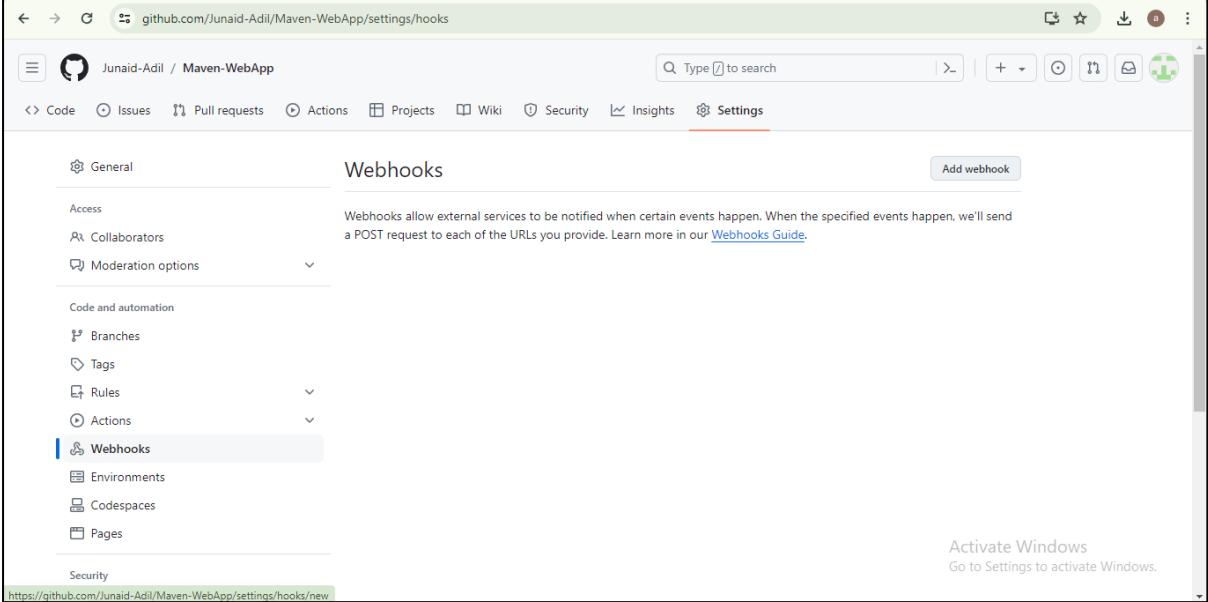
On the right side, there are sections for 'About', 'Releases', and 'Packages'. The 'About' section notes 'No description, website, or topics provided.' The 'Releases' section says 'No releases published' and has a link to 'Create a new release'. The 'Packages' section indicates 'No packages published' and provides links to 'Activate Windows' and 'Publish your first package'.

Step-4: In left pane select Webhooks

The screenshot shows the 'General' tab of the GitHub repository settings page. The left sidebar lists various settings categories, and 'Webhooks' is currently selected. The main content area contains the following information:

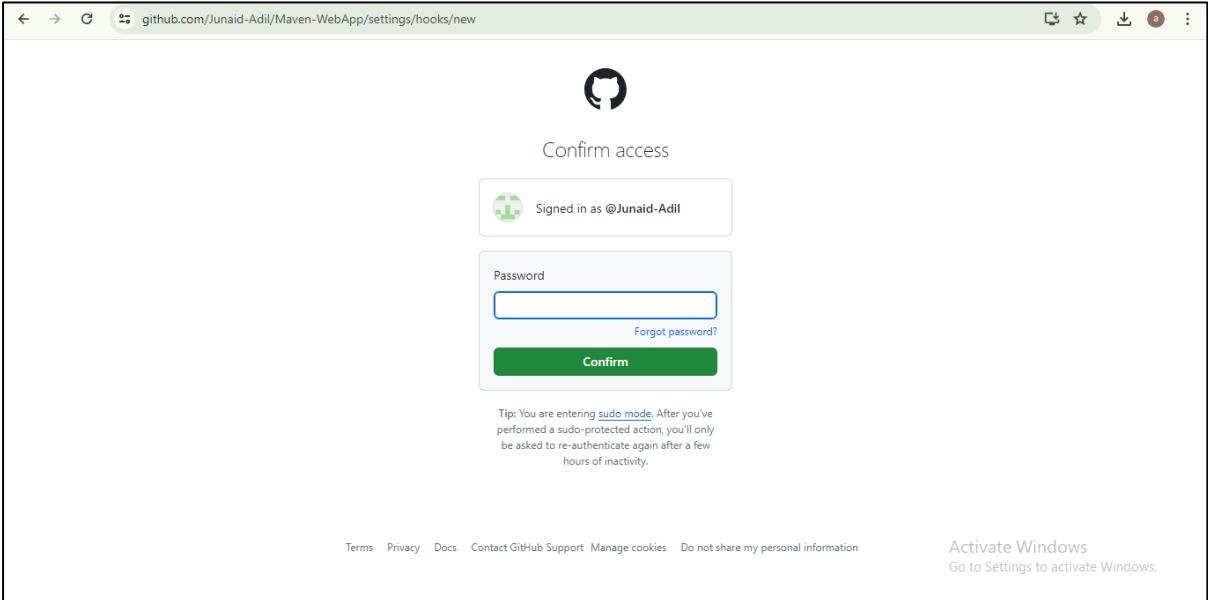
- General** tab selected.
- Access**: Shows 'Repository name' set to 'Maven-WebApp' with a 'Rename' button.
- Code and automation**: Includes 'Branches', 'Tags', 'Rules', and 'Actions' sections. 'Actions' is expanded, showing 'Webhooks' which is also selected.
- Default branch**: Set to 'main'.
- Social preview**: A section for uploading a social media preview image.
- Activate Windows**: A note to 'Go to Settings to activate Windows.'

Step-5: Click Add webhook



The screenshot shows the GitHub settings interface for a repository named 'Junaid-Adil / Maven-WebApp'. The user is on the 'Webhooks' page under the 'General' section. On the left, there's a sidebar with various settings categories like 'Access', 'Code and automation', and 'Security'. The 'Webhooks' category is currently selected and highlighted in blue. At the top right of the main content area, there is a prominent 'Add webhook' button. Below it, a brief description explains what webhooks are: 'Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#)'. The URL visible in the browser bar is <https://github.com/Junaid-Adil/Maven-WebApp/settings/hooks>.

Step-6: Enter Github password



The screenshot shows a 'Confirm access' dialog box from GitHub. It features a GitHub logo at the top. The main title is 'Confirm access'. Below it, a green button indicates the user is 'Signed in as @Junaid-Adil'. A large input field is labeled 'Password', with a 'Forgot password?' link underneath. A green 'Confirm' button is at the bottom. A tip message at the bottom left states: 'Tip: You are entering sudo mode. After you've performed a sudo-protected action, you'll only be asked to re-authenticate again after a few hours of inactivity.' At the very bottom of the page, there are links for 'Terms', 'Privacy', 'Docs', 'Contact GitHub Support', 'Manage cookies', and 'Do not share my personal information'. The URL in the browser bar is <https://github.com/Junaid-Adil/Maven-WebApp/settings/hooks/new>. An 'Activate Windows' message is also present at the bottom right.

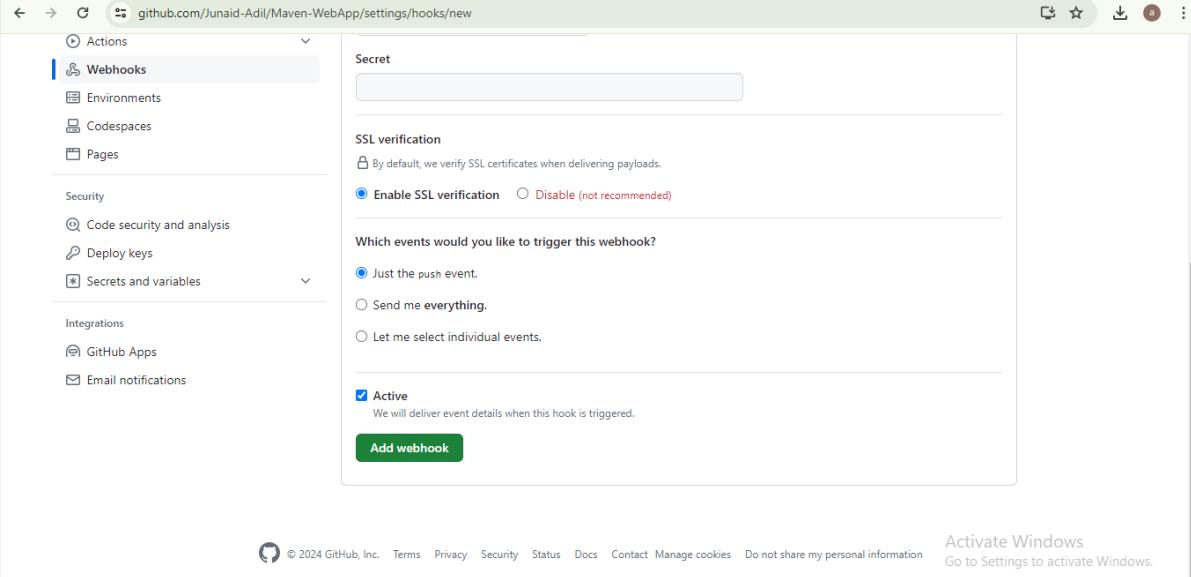
The screenshot shows the GitHub settings interface for a repository named 'Maven-WebApp'. The left sidebar is collapsed, showing options like General, Access, Code and automation, and Webhooks (which is selected). The main content area is titled 'Webhooks / Add webhook'. It contains fields for 'Payload URL' (set to 'https://example.com/postreceive'), 'Content type' (set to 'application/x-www-form-urlencoded'), and a 'Secret' field. Below these, there's a section for triggering events: 'Just the push event.' (selected) and 'Send me everything.' A message at the bottom right says 'Activate Windows Go to Settings to activate Windows.'

Step-7: In Payload URL: enter “<Jenkins URL>/github-Webhook/ ”

In Content type: select “application/json” from dropdown

This screenshot shows the same GitHub settings page as above, but with changes made to the webhook configuration. The 'Content type' dropdown has been changed from 'application/x-www-form-urlencoded' to 'application/json'. The rest of the fields and the sidebar remain the same.

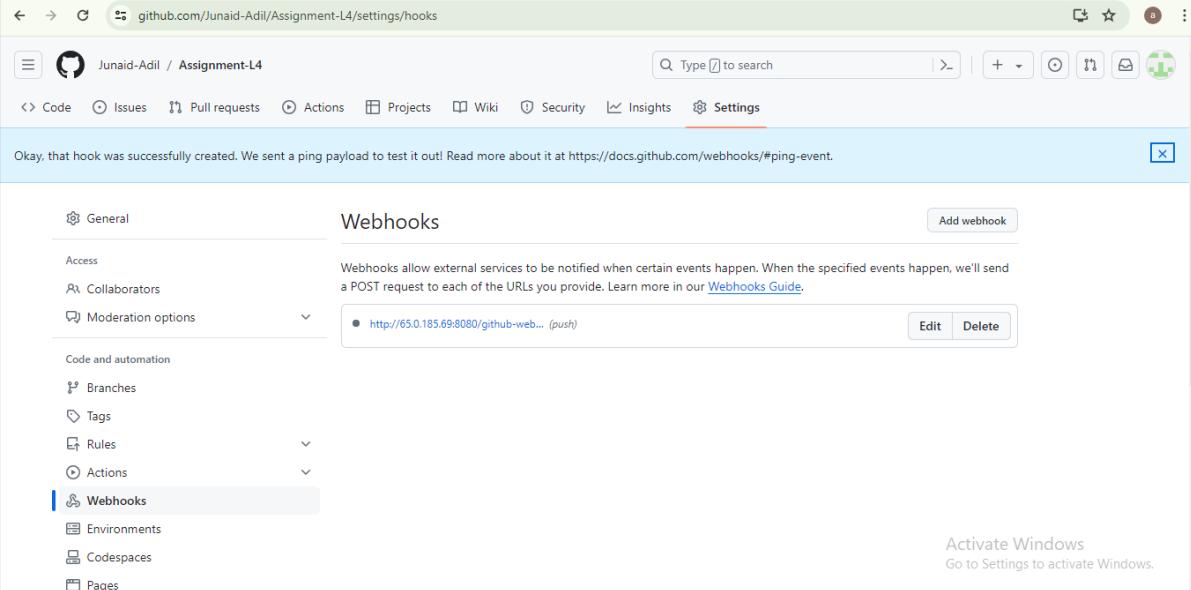
→ Add webhook



The screenshot shows the 'Add webhook' configuration page on GitHub. The URL is github.com/Junaid-Adil/Maven-WebApp/settings/hooks/new. The left sidebar has 'Actions' expanded, with 'Webhooks' selected. The main form includes:

- A 'Secret' input field.
- An 'SSL verification' section with a note about verifying certificates and two radio buttons: 'Enable SSL verification' (selected) and 'Disable (not recommended)'.
- A section for triggering events with three options: 'Just the push event.' (selected), 'Send me everything.', and 'Let me select individual events.'
- A checked 'Active' checkbox with a note: 'We will deliver event details when this hook is triggered.'
- A prominent green 'Add webhook' button at the bottom.

At the bottom of the page, there's a copyright notice: '© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information' and an 'Activate Windows' message: 'Activate Windows Go to Settings to activate Windows.'



The screenshot shows the 'Webhooks' settings page for the repository 'Assignment-L4'. The URL is github.com/Junaid-Adil/Assignment-L4/settings/hooks. The left sidebar has 'Actions' expanded, with 'Webhooks' selected. The main area displays:

- A heading 'Webhooks' with a note: 'Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#)'.
- A table showing one webhook entry:

<input checked="" type="radio"/> http://65.0.185.69:8080/github-web...	(push)	Edit	Delete
--	--------	----------------------	------------------------
- A green 'Add webhook' button.

At the bottom of the page, there's a copyright notice: '© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information' and an 'Activate Windows' message: 'Activate Windows Go to Settings to activate Windows.'

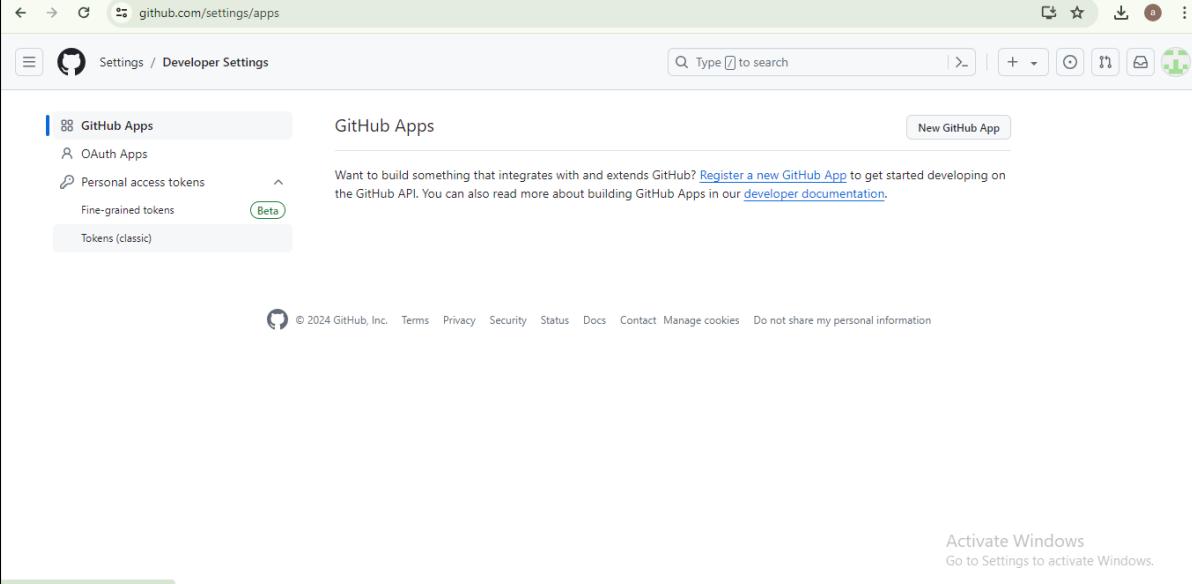
Step-8: To create a Personal Access Token (if it is a private Repository) -

Go to setting → Developer settings

The screenshot shows a GitHub repository page for 'Junaid-Adil / Maven-WebApp'. The repository is public and contains one branch ('main') and one tag ('0.0.1'). The commit history shows three commits from Junaid-Adil, all updating 'index.jsp'. Below the commits is a 'README' section with a 'Add a README' button. On the right side, there is a sidebar for 'Junaid-Adil' with various GitHub features like repositories, projects, and Copilot.

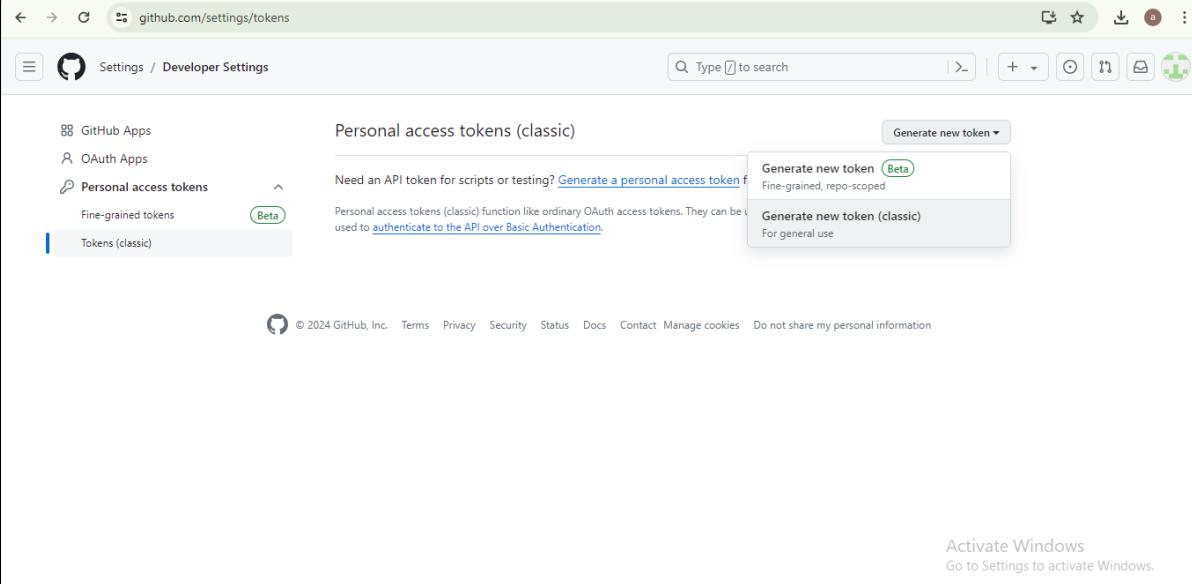
The screenshot shows the 'Developer settings' section of the GitHub profile settings. It includes fields for social links, company information, location, and time zone selection. A note states that all fields are optional and can be deleted at any time. Below this is the 'Contributions & activity' section, which contains two checkboxes: 'Make profile private and hide activity' and 'Include private contributions on my profile'. Both checkboxes have explanatory text and links to privacy statements.

Step-9: In Personal access tokens → select – Tokens (classic)



The screenshot shows the GitHub Developer Settings page at github.com/settings/apps. The left sidebar has 'GitHub Apps' selected. Under 'Personal access tokens', 'Tokens (classic)' is selected. A message at the top right says 'Want to build something that integrates with and extends GitHub? Register a new GitHub App to get started developing on the GitHub API. You can also read more about building GitHub Apps in our developer documentation.' Below this, there's a 'New GitHub App' button. At the bottom right, there's an 'Activate Windows' link.

Step-10: Click on Generate new token → Generate new token (classic)



The screenshot shows the GitHub Personal access tokens (classic) page at github.com/settings/tokens. The left sidebar has 'Personal access tokens' selected. A dropdown menu titled 'Generate new token' is open, showing two options: 'Generate new token (Beta)' (selected) and 'Generate new token (classic)'. The main content area has a message about API tokens and a note about using tokens for Basic Authentication. At the bottom right, there's an 'Activate Windows' link.

Step-11: Enter the note as “Jenkins Webhook” → Scope: ‘check’ repo

The screenshot shows the GitHub 'Personal access token (classic)' creation interface. In the 'Note' field, 'Jenkins Webhook' is entered. Under 'Select scopes', the 'repo' checkbox is checked, and several sub-options under it are also checked: 'repo:status', 'repo_deployment', 'public_repo', 'repo:invite', and 'security_events'. Other scopes like 'workflow', 'write:packages', and 'read:packages' are unchecked. The expiration is set to '30 days'. A note at the bottom right says 'Activate Windows Go to Settings to activate Windows.'

Step-12: Click on Generate Token

The screenshot shows the same GitHub token creation interface after the scopes have been selected. At the bottom, there are two buttons: 'Generate token' (highlighted in green) and 'Cancel'. A note at the bottom right says 'Activate Windows Go to Settings to activate Windows.'

Token is generated.

The screenshot shows the GitHub Developer Settings page under 'Personal access tokens (classic)'. A single token is listed with the value `ghp_Uh1CkxzZsCSevgQsz6xH7TD9Mfgr0i1c7w0y`. A blue button labeled 'Delete' is visible next to the token value. Below the token, a note states: 'Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.' At the bottom of the page, there is a note about activating Windows.

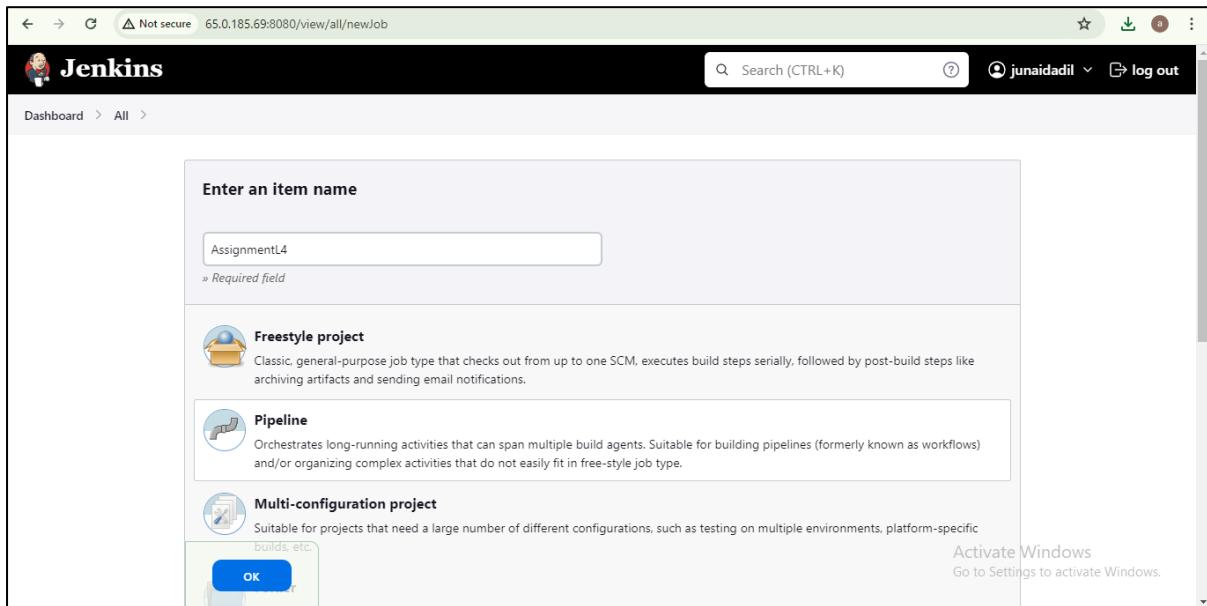
This token must be copied and pasted in a safe place. Once we reload the page we won't be able to access the token code.

Step-13: Now create a Jenkins Job

→ New Item

The screenshot shows the Jenkins dashboard at the URL `65.0.185.69:8080`. On the left, there is a sidebar with options like 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. The main area displays a table of existing Jenkins jobs. The table has columns for Status (S), Workstation (W), Name, Last Success, Last Failure, and Last Duration. Two jobs are listed: 'AssignmentL2' and 'AssignmentL3'. Both jobs show a green checkmark icon and a cloud icon. The last success time for AssignmentL2 is 16 hr #4, and for AssignmentL3 it is 41 min #11. The last failure time for AssignmentL2 is 16 hr #3, and for AssignmentL3 it is 1 hr 54 min #8. The last duration for AssignmentL2 is 8.2 sec, and for AssignmentL3 it is 10 sec. At the bottom of the page, there is a note about activating Windows.

Step-14: Name: AssignmentL4 → select: Pipeline project → Ok



Enter an item name

AssignmentL4

» Required field

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

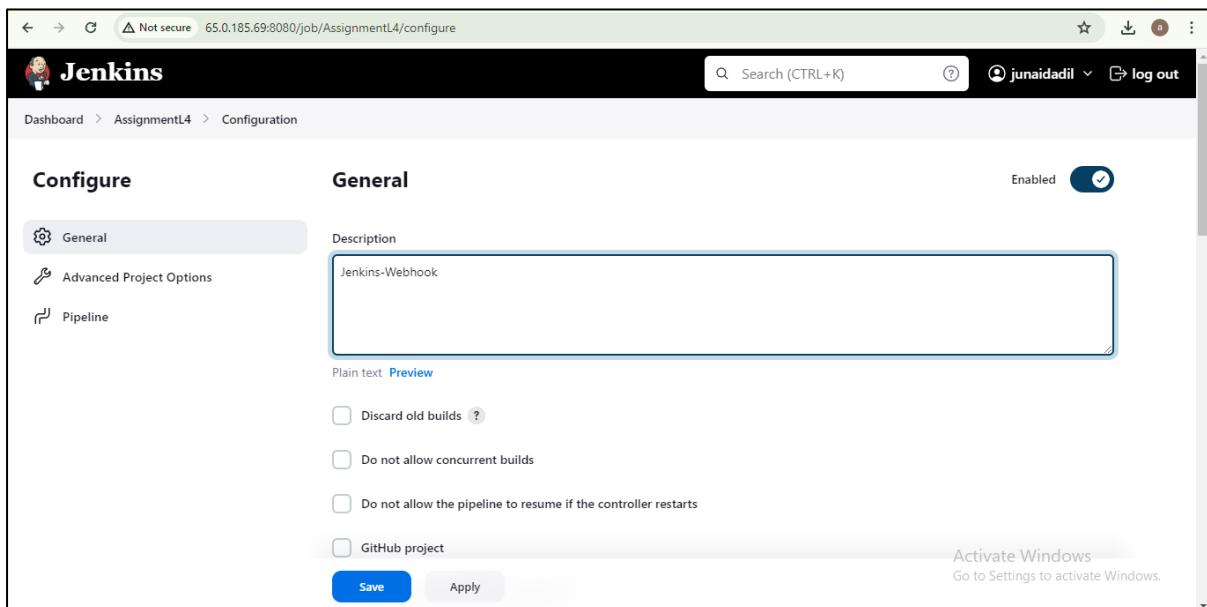
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

Activate Windows
Go to Settings to activate Windows.

Step-15: In description enter: “Jenkins-Webhook”



Configure General

Enabled

Description

Jenkins-Webhook

Plain text [Preview](#)

Discard old builds ?

Do not allow concurrent builds

Do not allow the pipeline to resume if the controller restarts

GitHub project

Save Apply

Activate Windows
Go to Settings to activate Windows.

Step-16: In “Build Triggers”: ‘check’ GitHub hook trigger for GITScm polling and Poll SCM.

In Poll SCM- Schedule: enter “ * * * * * ” as it says, to build the job per minute

The screenshot shows the Jenkins job configuration page for 'AssignmentL4'. Under the 'Build Triggers' section, two options are selected: 'GitHub hook trigger for GITScm polling' and 'Poll SCM'. A warning message at the bottom states: '⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * * *" to poll once per hour'. Below the triggers, there is a 'Schedule' field containing 'x * * * *'. At the bottom of the page are 'Save' and 'Apply' buttons.

Step-17: Enter the Pipeline Script

The screenshot shows the Jenkins job configuration page for 'AssignmentL4'. Under the 'Pipeline' section, the 'Definition' dropdown is set to 'Pipeline script'. The 'Script' editor contains the following Groovy pipeline code:

```
1 pipeline {
2     agent any
3
4     stages {
5         stage('Checkout') {
6             steps {
7                 git branch: 'main', url: 'https://github.com/Junaid-Adil/Maven-WebApp.git'
8             }
9         }
10        stage('Build') {
11            steps {
12                sh 'mvn clean package'
13            }
14        }
15    }
16}
```

A checkbox for 'Use Groovy Sandbox' is checked. At the bottom of the page are 'Save' and 'Apply' buttons.

Apply → Save

The screenshot shows the Jenkins Pipeline configuration page. On the left, there's a sidebar with tabs: General, Advanced Project Options, and Pipeline (which is selected). The main area is titled 'Pipeline' and has a 'Definition' section labeled 'Pipeline script'. A code editor contains the following Groovy script:

```
1> pipeline {
2>     agent any
3>
4>     stages {
5>         stage('Checkout') {
6>             steps {
7>                 git branch: 'main', url: 'https://github.com/Junaid-Adil/Maven-WebApp.git'
8>             }
9>         }
10>        stage('Build') {
11>            steps {
12>                sh 'mvn clean package'
13>            }
14>        }
15>    }
16>}
```

Below the script, there's a checkbox for 'Use Groovy Sandbox'. At the bottom, there are 'Save' and 'Apply' buttons, and a green 'Saved' button.

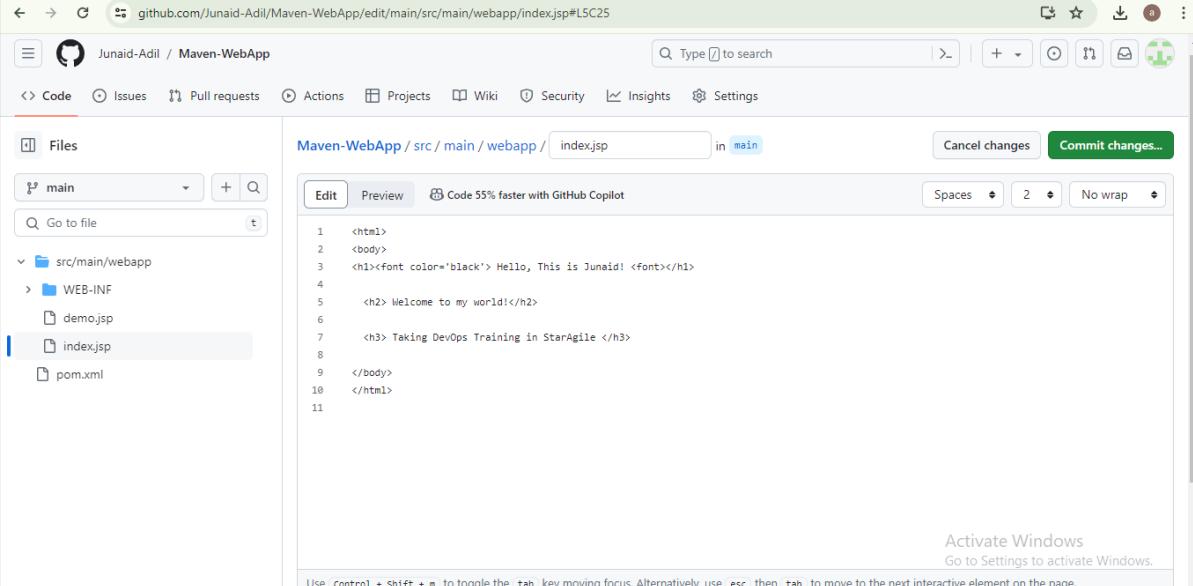
We can see Build got triggered automatically

The screenshot shows the Jenkins Stage View page for the 'AssignmentL4' pipeline. The left sidebar includes options like Delete Pipeline, Full Stage View, Stages, Rename, Pipeline Syntax, GitHub Hook Log, and Polling Log. The main area is titled 'Stage View' and displays a message: 'No data available. This Pipeline has not yet run.' Below this, there's a 'Permalinks' section. On the right, there's a 'Build History' section with a table:

Build	Date	Status
#1	20 Jun 2024, 14:45	Success

At the bottom, there are links for 'Atom feed for all' and 'Atom feed for failures'. A note at the bottom right says 'Activate Windows'.

Step-18: Now make changes in Github repository file.

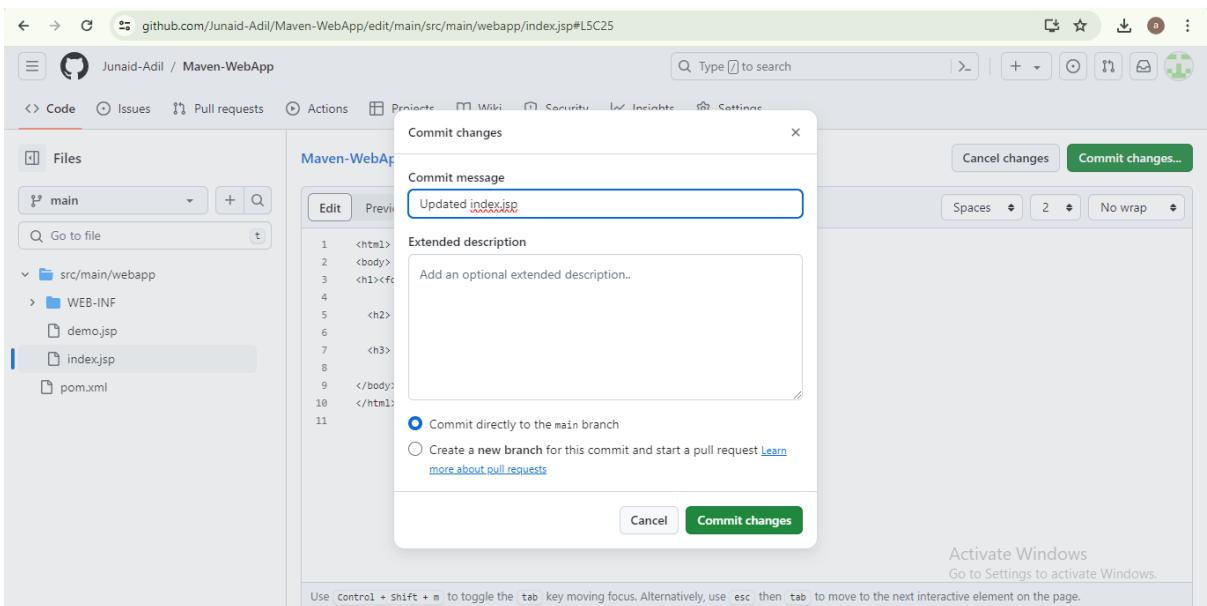


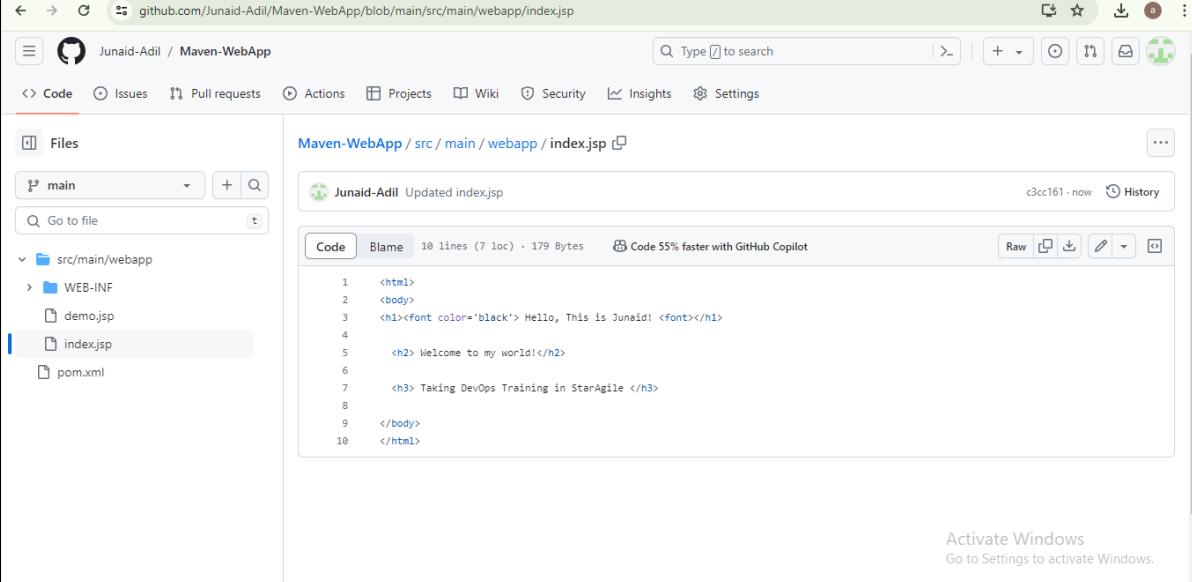
The screenshot shows the GitHub code editor interface for a Maven WebApp repository. The left sidebar displays the project structure with files like pom.xml, index.jsp, demo.jsp, and WEB-INF. The main editor area shows the content of index.jsp:

```
1 <html>
2 <body>
3 <h1><font color='black'> Hello, This is Junaid! <font></h1>
4
5 <h2> Welcome to my world!</h2>
6
7 <h3> Taking DevOps Training in StarAgile </h3>
8
9 </body>
10 </html>
11
```

At the top right of the editor, there are buttons for "Cancel changes" and "Commit changes...". Below the editor, a message from GitHub Copilot suggests "Code 55% faster with GitHub Copilot". At the bottom right, there's a note about activating Windows.

Step-19: Commit the changes





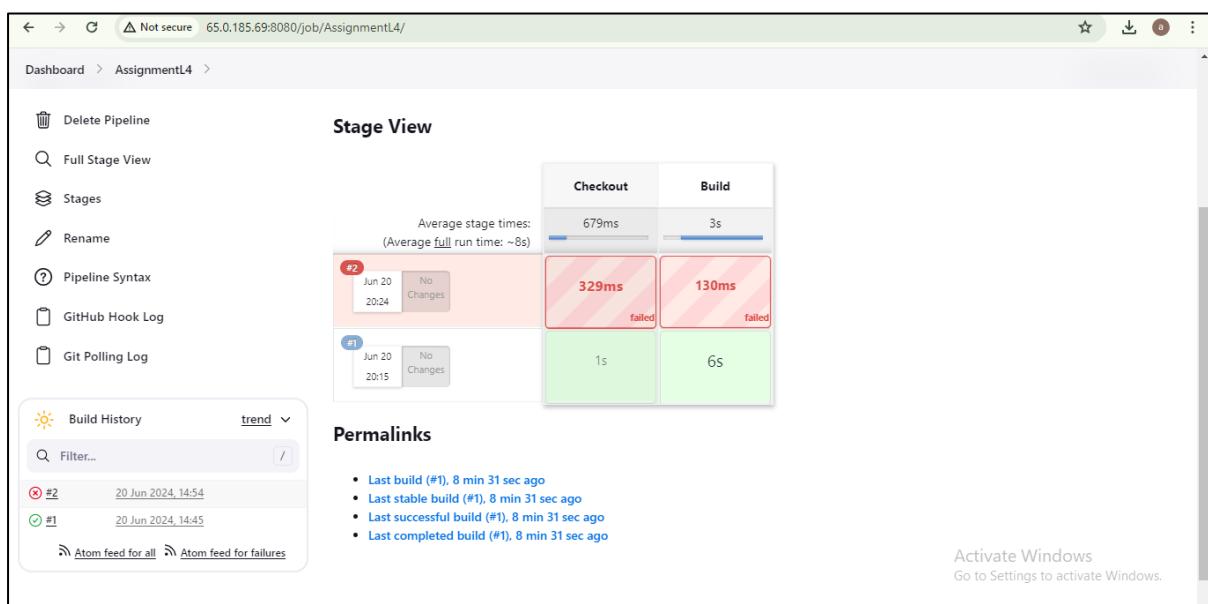
A screenshot of a GitHub repository page for 'Junaid-Adil / Maven-WebApp'. The 'Code' tab is selected, showing the file 'index.jsp'. The code content is as follows:

```
1 <html>
2 <body>
3 <h1><font color='black'> Hello, This is Junaid! </font></h1>
4
5 <h2> Welcome to my world!</h2>
6
7 <h3> Taking DevOps Training in StarAgile </h3>
8
9 </body>
10 </html>
```

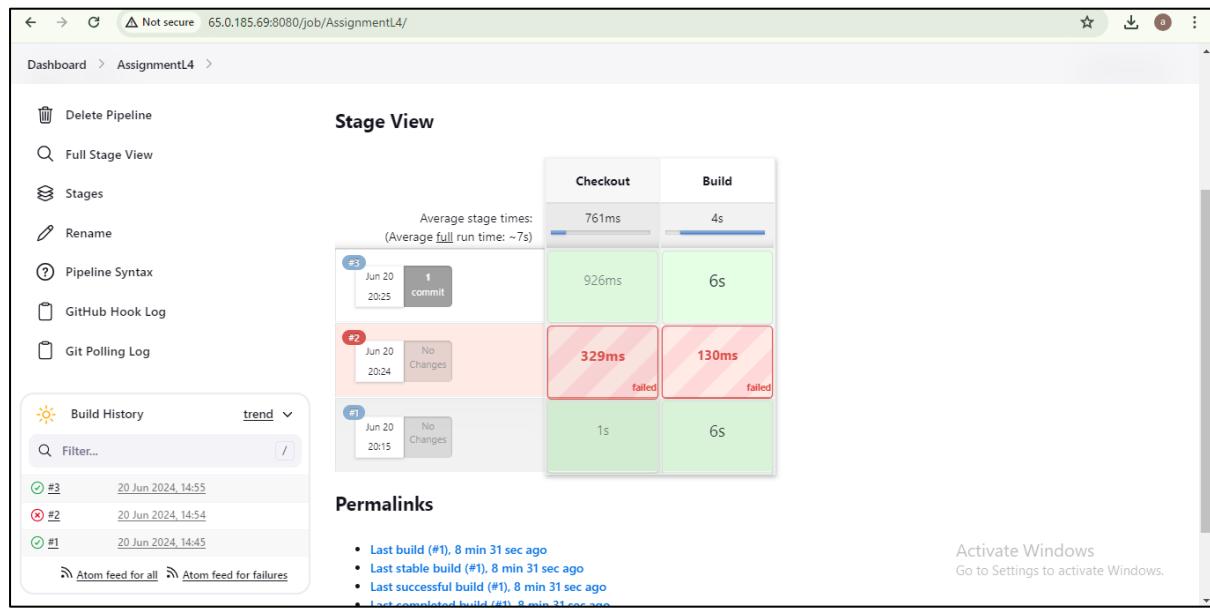
The commit message is 'Junaid-Adil Updated index.jsp' and it was made at 'c3cc161 · now'. A note indicates 'Code 55% faster with GitHub Copilot'.

Committed the changes.

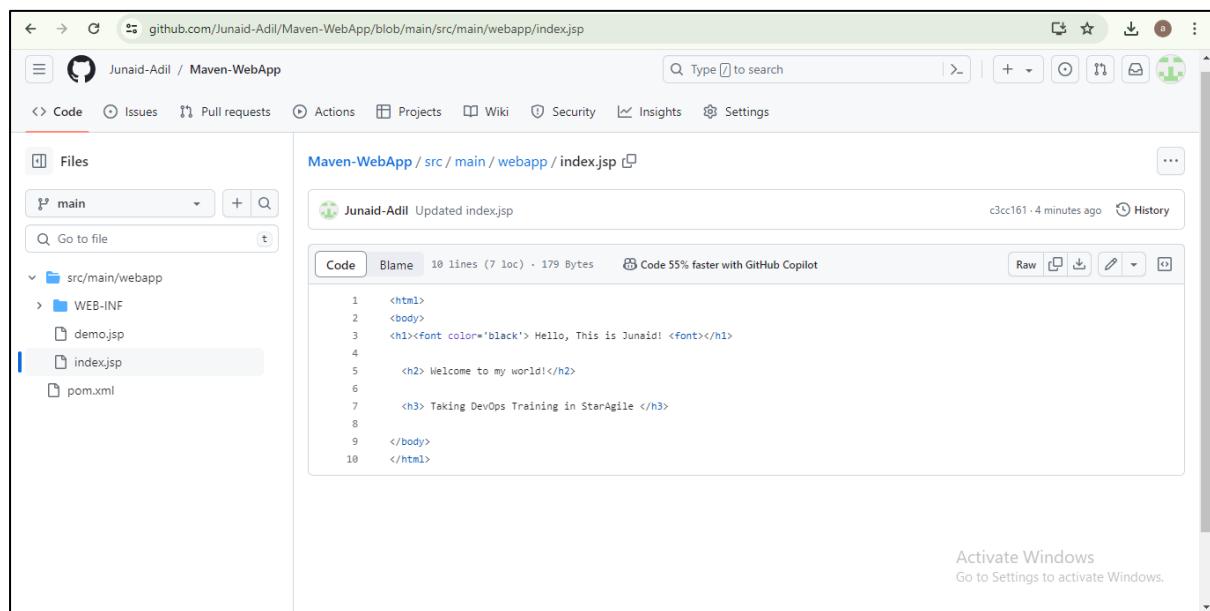
Second Build got executed



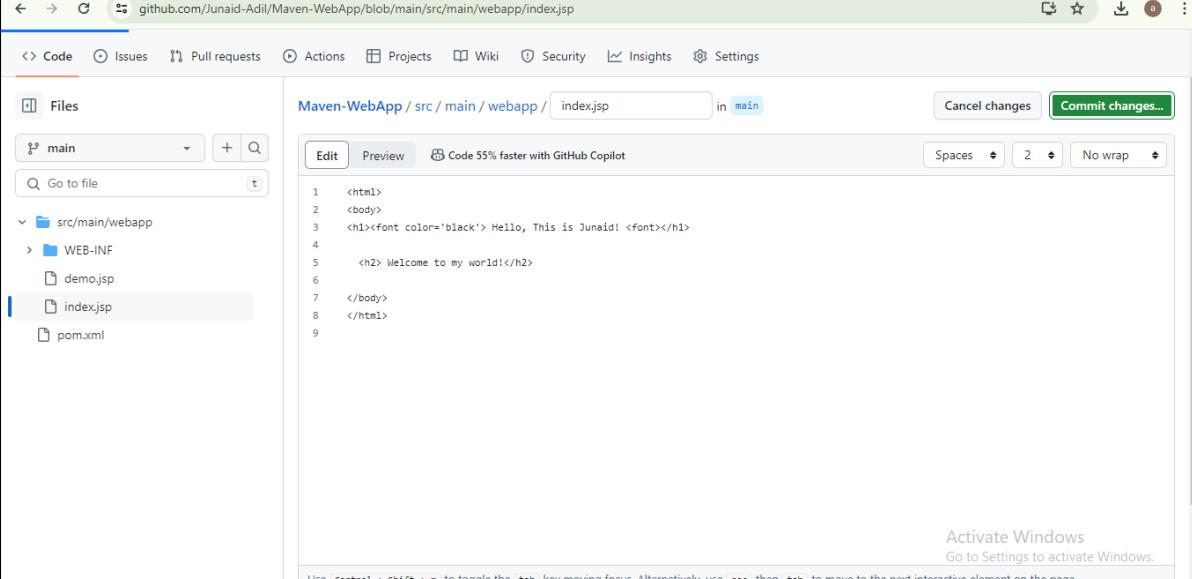
In a minute again it got triggered with the latest commit



Step-20: Update the file in Github



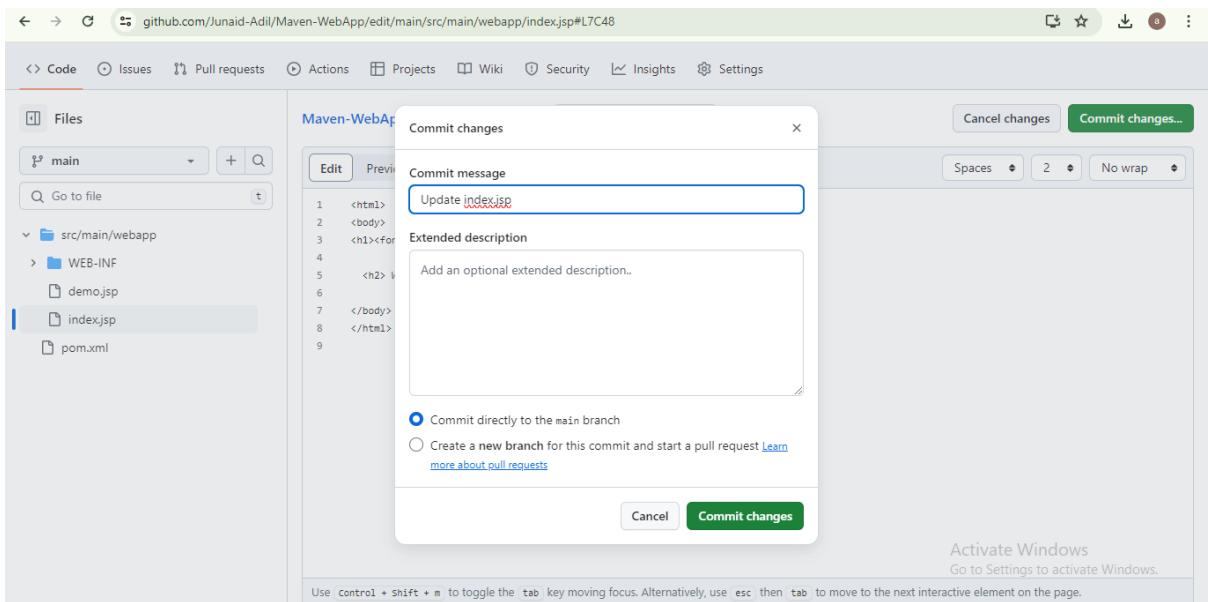
Step-21: Deleted <h3> field and committed



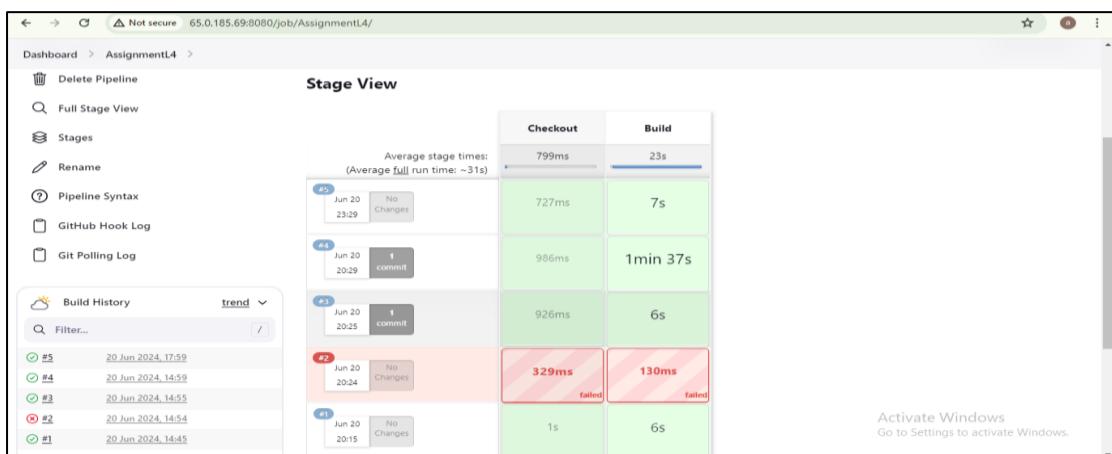
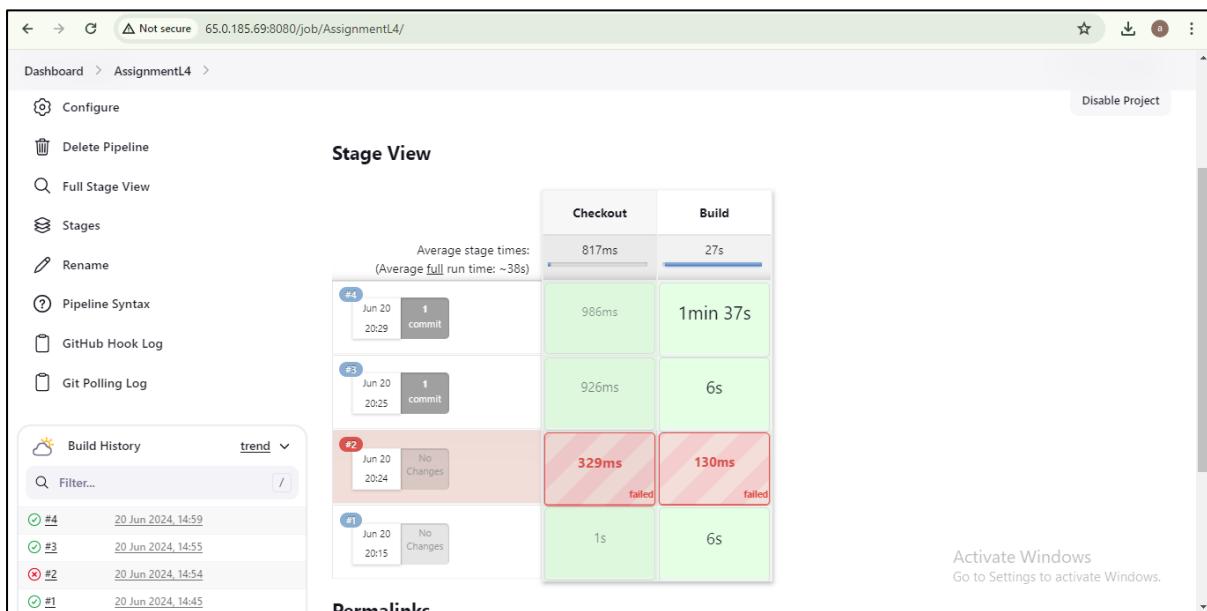
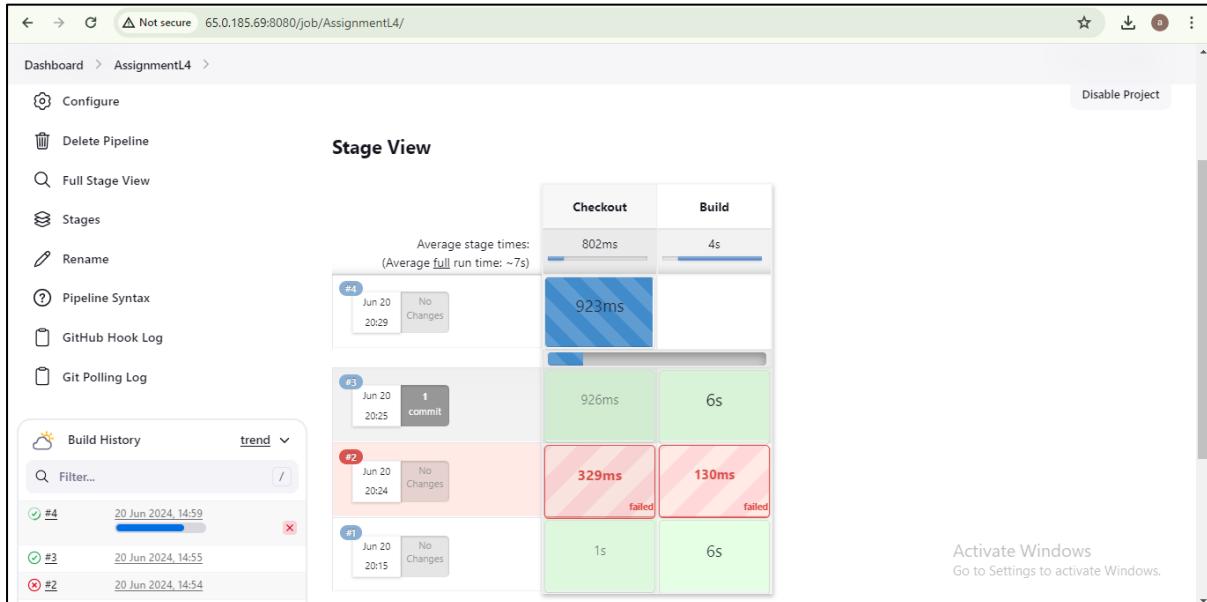
The screenshot shows the GitHub code editor interface. The left sidebar displays the project structure: main, src/main/webapp, WEB-INF, demo.jsp, index.jsp, and pom.xml. The right panel shows the content of index.jsp:

```
1 <html>
2 <body>
3 <h1><font color='black'> Hello, This is Junaid! </font></h1>
4
5 <h2> Welcome to my world!</h2>
6
7 </body>
8 </html>
```

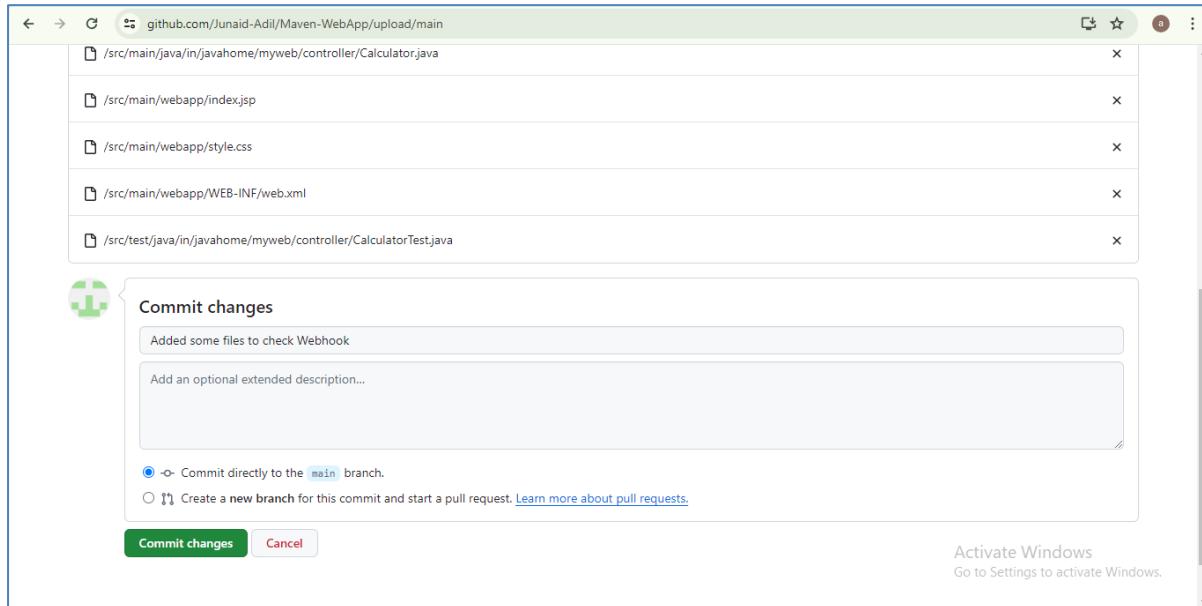
At the top right, there are 'Cancel changes' and 'Commit changes...' buttons. Below the code editor, a message says 'Activate Windows' and 'Go to Settings to activate Windows.' A note at the bottom provides keyboard navigation instructions.



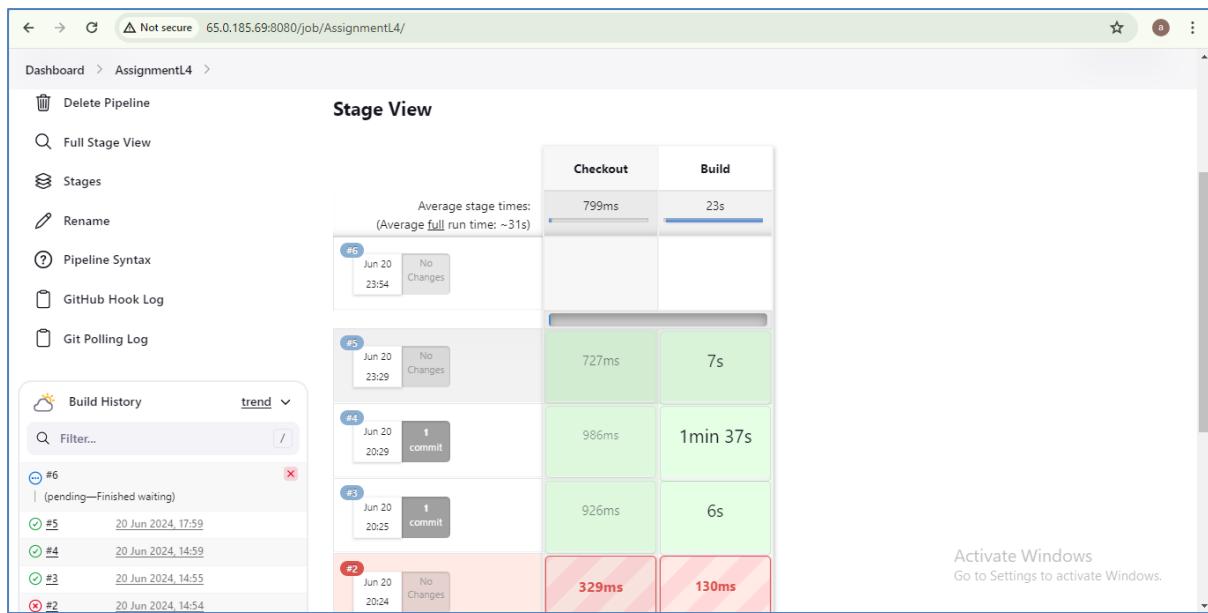
Now we can see the Build got Triggered automatically as I have made changes in source code.



Step-22: Push the new code into GitHub → commit changes



The screenshot shows the GitHub repository page for 'Maven-WebApp' owned by 'Junaid-Adil'. The repository has 1 branch and 0 tags. The main commit is by 'Junaid-Adil' with the message 'Added some files to check Webhook' and timestamp 'dd5cfa · now'. The commit details show changes to 'src' and 'pom.xml'. The repository has 6 commits in total. On the right side, there is an 'About' section with the message 'No description, website, or topics provided.' It also shows activity metrics: 0 stars, 1 watching, and 0 forks. Below that is a 'Releases' section stating 'No releases published' and a 'Packages' section stating 'No packages published'. A note at the bottom right says 'Activate Windows Go to Settings to activate Windows.'



As soon as the code was pushed to Github repository, Build got triggered. as GitHub WebHook is enabled.

Poll SCM: It makes Jenkins periodically check the version control system (like GitHub) for changes. we specify a schedule, to when Jenkins should poll the SCM for changes. If changes are detected, Jenkins triggers a build.

GitHub webhook: This is a more efficient way to trigger Jenkins builds based on the changes in the GitHub repository. When we push new code or perform certain actions, GitHub can notify Jenkins instantly via a webhook. This removes the requirement for Jenkins to continuously monitor for changes.