



INDIAN INSTITUTE OF
INFORMATION
TECHNOLOGY

CS457

DevOps Final Assignment -2

Submitted to:

Dr. Uma S

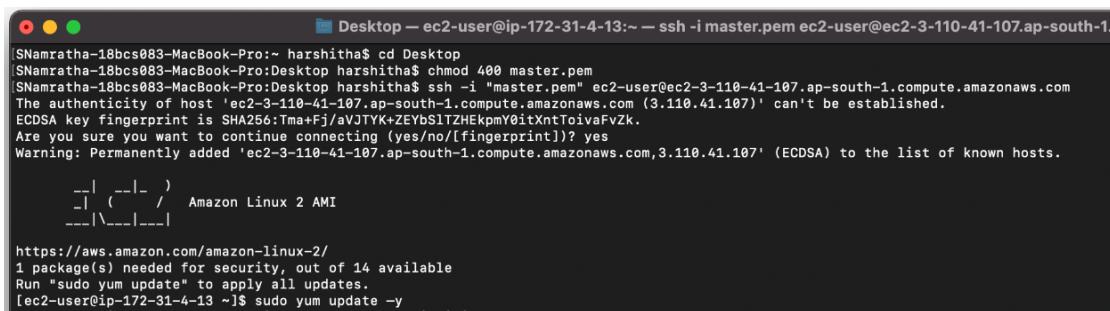
Submitted by:

S Namratha
(18BCS083)

DevOps Assignment 2

Jenkins master slave pipeline

1. Create an EC2 instance and connect it using SSH



```
[SNamratha-18bcs083-MacBook-Pro:~ harshitha$ cd Desktop
[SNamratha-18bcs083-MacBook-Pro:Desktop harshitha$ chmod 400 master.pem
[SNamratha-18bcs083-MacBook-Pro:Desktop harshitha$ ssh -i "master.pem" ec2-user@ec2-3-110-41-107.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-3-110-41-107.ap-south-1.compute.amazonaws.com (3.110.41.107)' can't be established.
ECDSA key fingerprint is SHA256:Tma+Fj/AVJTYK+ZEYbS1TZHeKpmYoitXntToivaFvZk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-110-41-107.ap-south-1.compute.amazonaws.com,3.110.41.107' (ECDSA) to the list of known hosts.

  _|_ _|_
  | (   /
  _\_\_|_|
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-4-13 ~]$ sudo yum update -y
```

2. Install and run jenkins

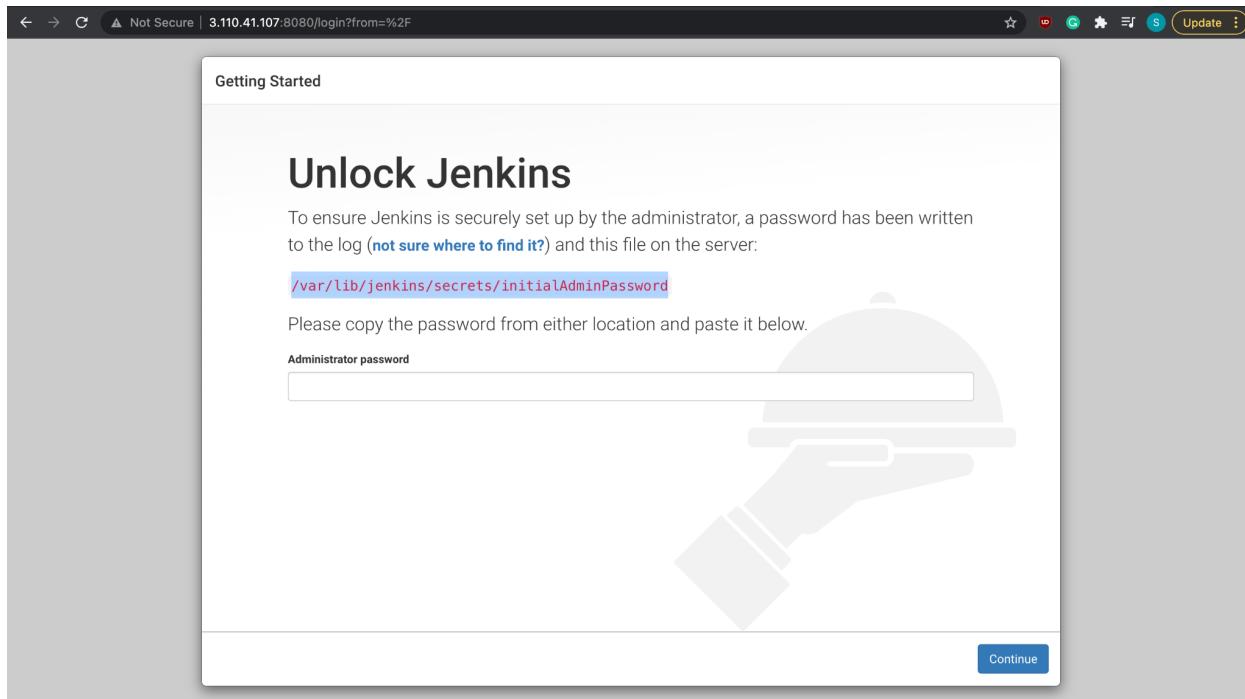
2.1. Commands to install jenkins (mac terminal)

```
2.1.1. sudo yum update -y
2.1.2. sudo wget -O /etc/yum.repos.d/jenkins.repo
        \https://pkg.jenkins.io/redhat-stable/jenkins.repo
2.1.3. sudo amazon-linux-extras install epel -y
2.1.4. sudo vi vim /etc/yum.repos.d/epelfordaemonize.repo
        and enter the following-
        [daemonize]
        baseurl=https://download-ib01.fedoraproject.org/pub/e
        pel/7/x86_64/
        gpgcheck=no
        enabled=yes
2.1.5. sudo yum install daemonize -y
2.1.6. sudo yum install jenkins java-1.8.0-openjdk-devel -y
2.1.7. sudo systemctl start jenkins
```

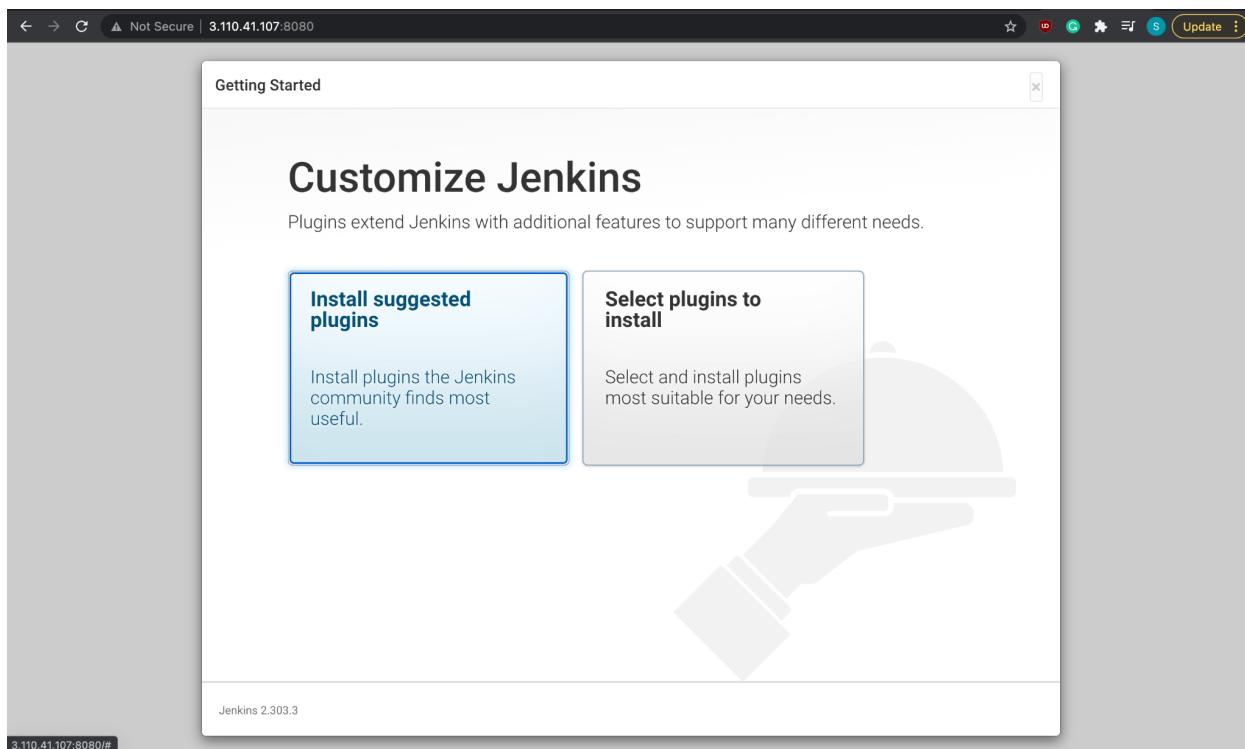
```
Desktop — ec2-user@ip-172-31-4-13:~ — ssh -i master.pem ec2-user@ec2-172-31-4-13
libXau.x86_64 0:1.0.8-2.1.amzn2.0.2
libXcomposite.x86_64 0:0.4.4-4.1.amzn2.0.2
libXcursor.x86_64 0:1.1.15-1.amzn2.0.2
libXdamage.x86_64 0:1.1.4-4.1.amzn2.0.2
libXext.x86_64 0:1.3.3-3.amzn2.0.2
libXfixes.x86_64 0:5.0.3-1.amzn2.0.2
libXft.x86_64 0:2.3.2-2.amzn2.0.2
libXi.x86_64 0:1.7.9-1.amzn2.0.2
libXinerama.x86_64 0:1.1.3-2.1.amzn2.0.2
libXrandr.x86_64 0:1.5.1-2.amzn2.0.3
libXrender.x86_64 0:0.9.10-1.amzn2.0.2
libXtst.x86_64 0:1.2.3-1.amzn2.0.2
libXxf86vm.x86_64 0:1.1.4-1.amzn2.0.2
libfontenc.x86_64 0:1.1.3-3.amzn2.0.2
libglvnd.x86_64 1:1.0.1-0.1.git5baa1e5.amzn2.0.1
libglvnd-egl.x86_64 1:1.0.1-0.1.git5baa1e5.amzn2.0.1
libglvnd-glx.x86_64 1:1.0.1-0.1.git5baa1e5.amzn2.0.1
libthai.x86_64 0:0.1.14-9.amzn2.0.2
libwayland-client.x86_64 0:1.17.0-1.amzn2
libwayland-server.x86_64 0:1.17.0-1.amzn2
libxcb.x86_64 0:1.12-1.amzn2.0.2
libxshmfence.x86_64 0:1.2-1.amzn2.0.2
libxslt.x86_64 0:1.1.28-6.amzn2
lksctp-tools.x86_64 0:1.0.17-2.amzn2.0.2
mesa-libEGL.x86_64 0:18.3.4-5.amzn2.0.1
mesa-libGL.x86_64 0:18.3.4-5.amzn2.0.1
mesa-libgbm.x86_64 0:18.3.4-5.amzn2.0.1
mesa-libglapi.x86_64 0:18.3.4-5.amzn2.0.1
pango.x86_64 0:1.42.4-4.amzn2
pcsc-lite-libs.x86_64 0:1.8.8-7.amzn2
pixman.x86_64 0:0.34.0-1.amzn2.0.2
python-javapackages.noarch 0:3.4.1-11.amzn2
python-lxml.x86_64 0:3.2.1-4.amzn2.0.3
ttmkfdir.x86_64 0:3.0.9-42.amzn2.0.2
tzdata-java.noarch 0:2021a-1.amzn2
xorg-x11-font-utils.x86_64 1:7.5-21.amzn2
xorg-x11-fonts-Type1.noarch 0:7.5-9.amzn2

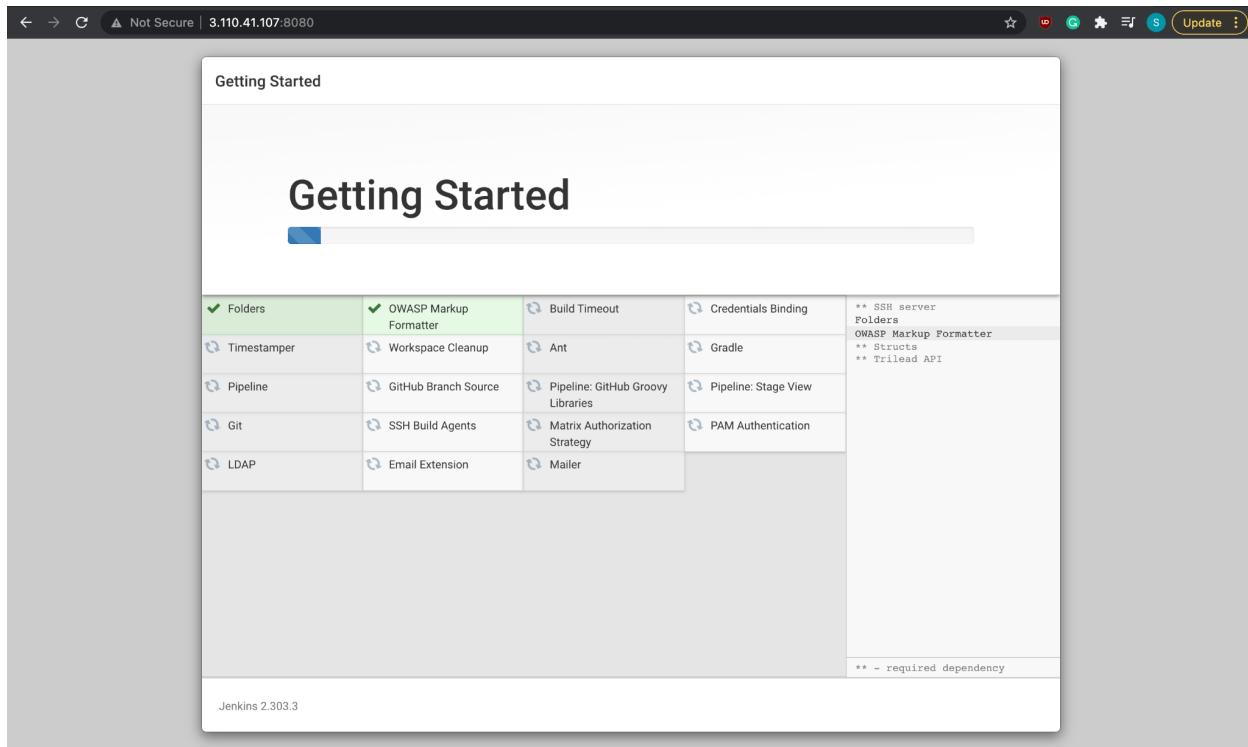
Complete!
[ec2-user@ip-172-31-4-13 ~]$ sudo systemctl start jenkins
```

2.2. Open the jenkins on browser using IP address



3. Install the suggested packages





4. Setup Jenkins

4.1. Create first admin user

Not Secure | 3.110.41.107:8080

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.303.3

Skip and continue as admin Save and Continue

Not Secure | 3.110.41.107:8080

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

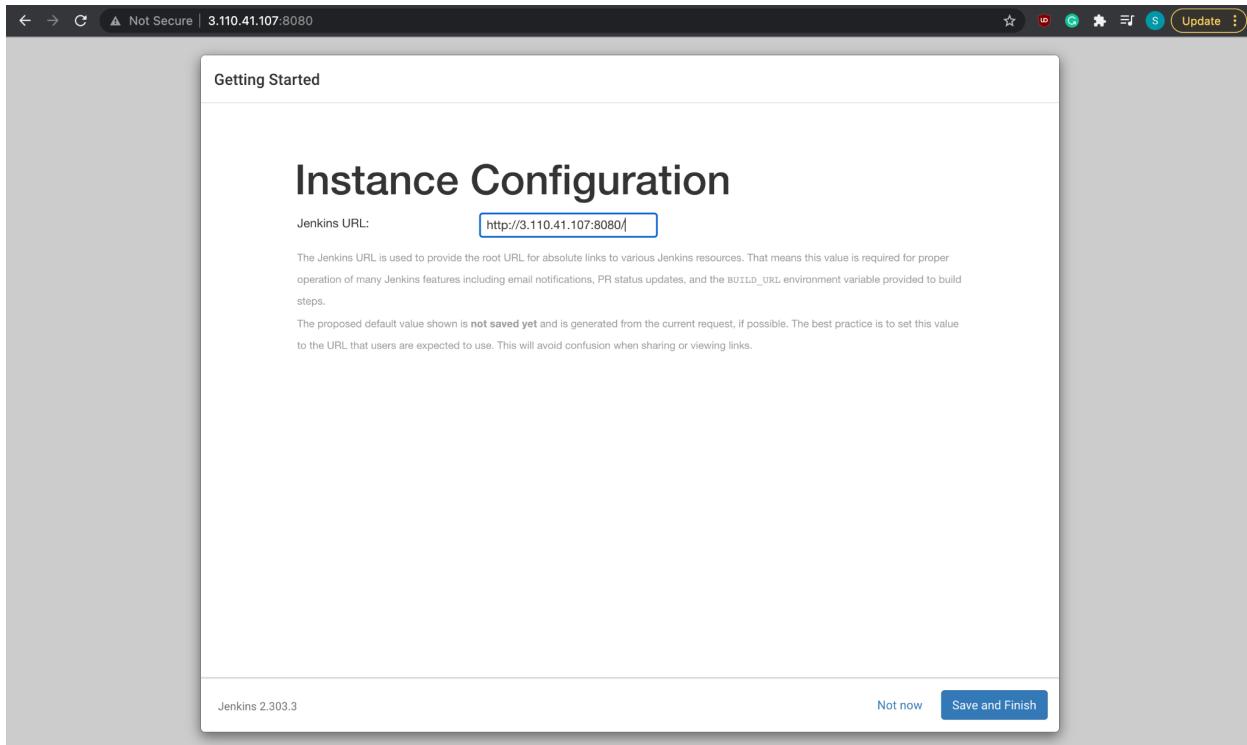
Full name:

E-mail address:

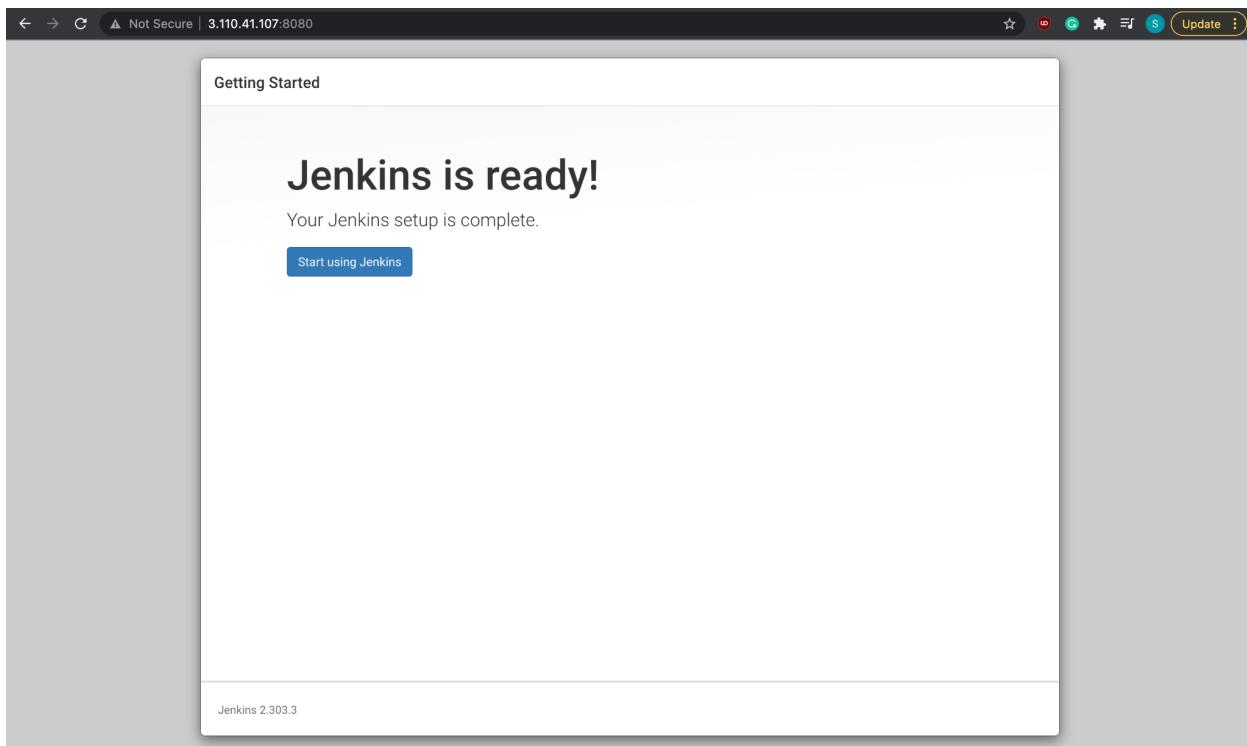
Jenkins 2.303.3

Skip and continue as admin Save and Continue

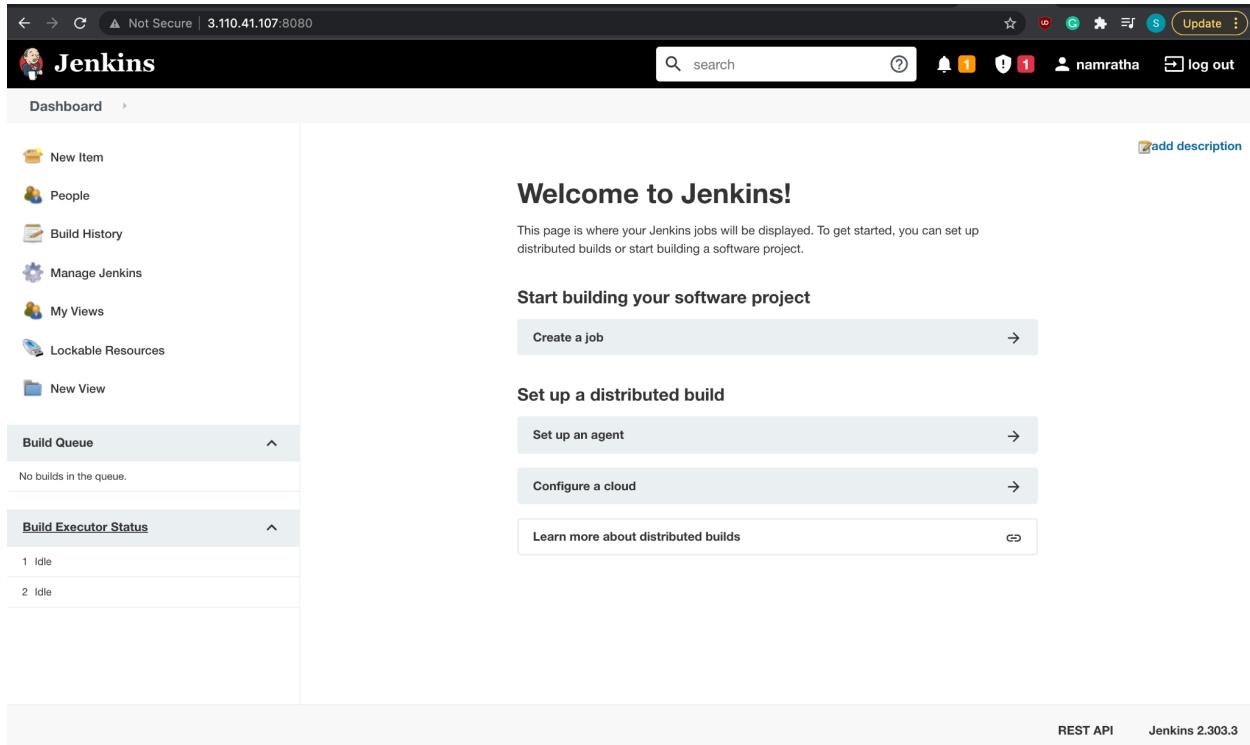
4.2. Save and finish



4.3. Jenkins is ready



4.4. Open Jenkins



5. Setup master slave on aws

- 5.1. Deploy the servers- slave 1 and slave 2.
- 5.2. Connect slaves using jnlp connection.
- 5.3. In EC2 instance go to jenkins-master, press action and select launch more like this. Under the section Configure instance, change number of instances to 2 and launch them.
- 5.4. Change the name to slave-1 and slave-2.
- 5.5. Go to jenkins -> Manage jenkins -> Configure global security. In agents, change TCP port for inbound agents to random and save.

Not Secure | 3.110.41.107:8080/configureSecurity/

Dashboard > Configure Global Security

LDAP

Unix user/group database

None

Authorization

Anyone can do anything

Legacy mode

Logged-in users can do anything

Allow anonymous read access

Matrix-based security

Project-based Matrix Authorization Strategy

Markup Formatter

Markup Formatter

Plain text

Treats all input as plain text. HTML unsafe characters like < and & are escaped to their respective character entities.

Agents

TCP port for inbound agents

Fixed : Random Disable

Agent protocols...

CSRF Protection

Crumb Issuer

Buttons

Save Apply

5.6. Go to manage nodes and add new node.

Not Secure | 3.110.41.107:8080/computer/

Jenkins

Dashboard > Nodes >

Back to Dashboard

Manage Jenkins

New Node

Configure Clouds

Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	5.89 GB	0 B	5.89 GB	0ms
		Data obtained	22 min	22 min	22 min	22 min	22 min

Refresh status

3.110.41.107:8080/computer/new

REST API Jenkins 2.303.3

5.7. Create new nodes with names slave-1 and slave-2.

Not Secure | 3.110.41.107:8080/computer/new

Jenkins

Dashboard > Nodes >

[Back to Dashboard](#)

[Manage Jenkins](#)

[New Node](#)

[Configure Clouds](#)

[Node Monitoring](#)

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

Node name: slave-1

Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

OK

Not Secure | 3.110.41.107:8080/computer/slave-1/configure

REST API Jenkins 2.303.3

Dashboard > Nodes > slave-1

[Load Statistics](#)

[Script Console](#)

[Log](#)

[System Information](#)

[Disconnect](#)

Build Executor Status
1 Idle

Remote root directory:

Remote directory is mandatory

Labels:

Usage: Use this node as much as possible

Launch method: Launch agent by connecting it to the master

Disable WorkDir

Custom WorkDir path: /home/ec2-user

Internal data directory: remoting

Fail if workspace is missing

Use WebSocket

Save

Not Secure | 3.110.41.107:8080/computer/new

Jenkins

Dashboard > Nodes >

[Back to Dashboard](#)

[Manage Jenkins](#)

[New Node](#) (selected)

[Configure Clouds](#)

[Node Monitoring](#)

Build Queue ^
No builds in the queue.

Build Executor Status ^
master
1 Idle
2 Idle
slave-1 (offline)

Node name: slave-2

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

Copy Existing Node
Copy from: slave-1|

OK

Not Secure | 3.110.41.107:8080/computer/slave-2/configure

slave-2

Dashboard > Nodes > slave-2

[Script Console](#)

[Log](#)

[System Information](#)

[Disconnect](#)

Build Executor Status ^
1 Idle

Remote root directory: /home/ec2-user

Labels:

Usage: Use this node as much as possible

Launch method: Launch agent by connecting it to the master

Disable WorkDir

Custom WorkDir path: /home/ec2-user

Internal data directory: remoting

Fail if workspace is missing

Use WebSocket

Availability

Advanced...

Save

The screenshot shows the Jenkins interface for managing nodes. On the left sidebar, there are links for Back to Dashboard, Manage Jenkins, New Node, Configure Clouds, and Node Monitoring. Below these are sections for Build Queue and Build Executor Status. The Build Queue section indicates 'No builds in the queue.' The Build Executor Status section lists three nodes: master (idle), slave-1 (offline), and slave-2 (offline). The main content area displays a table of nodes with columns: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. The master node is listed with 'Linux (amd64)' architecture, 'In sync' status, and 5.68 GB free disk space. The slave nodes are listed with 'N/A' for most metrics. A 'Refresh status' button is located at the bottom right of the table.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	5.68 GB	0 B	5.68 GB	0ms
	slave-1		N/A	N/A	N/A	N/A	N/A
	slave-2		N/A	N/A	N/A	N/A	N/A

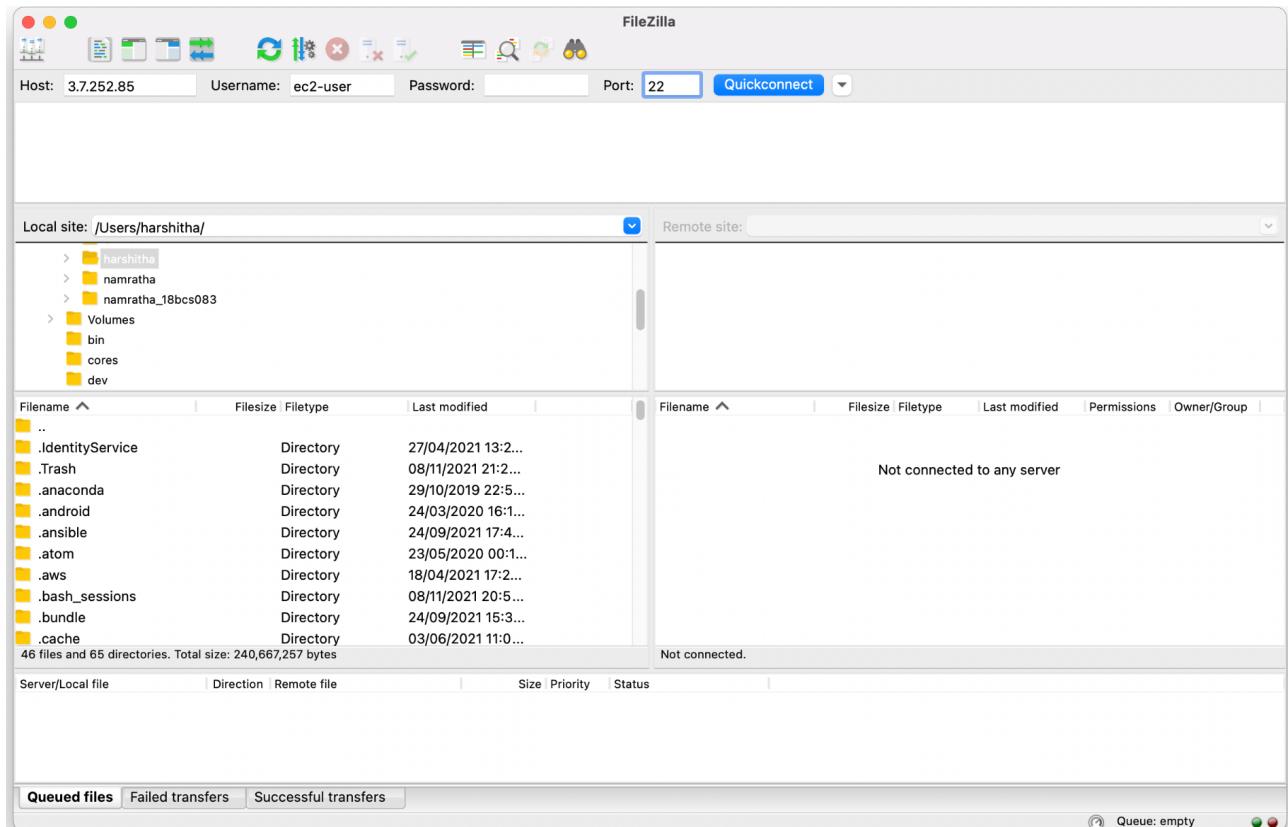
5.8. Go to slave-1 and download the agent.jar file.

5.9. Send the jar file to slave serve 1.

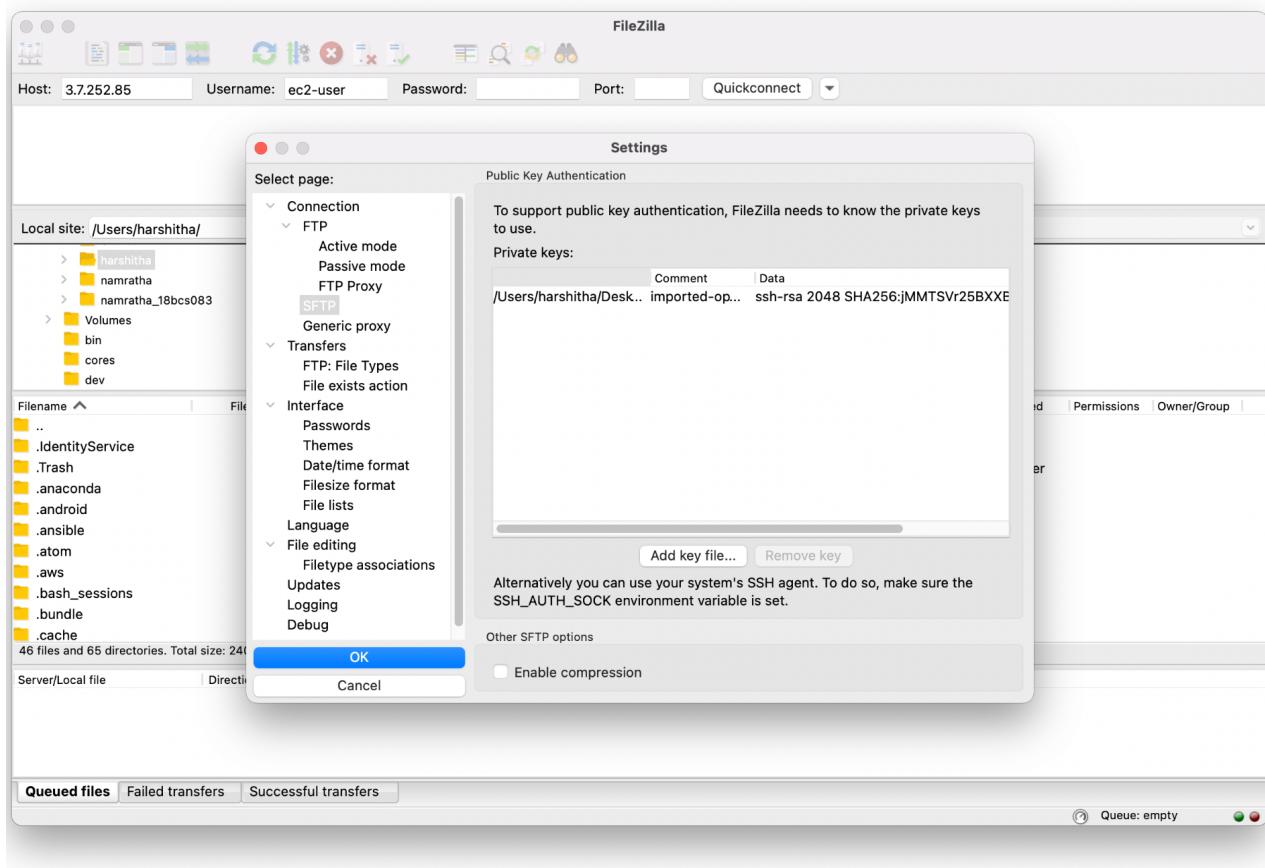
5.9.1. Go to slave1 EC2 instance and select public ip address.

5.9.2. Open filezilla.

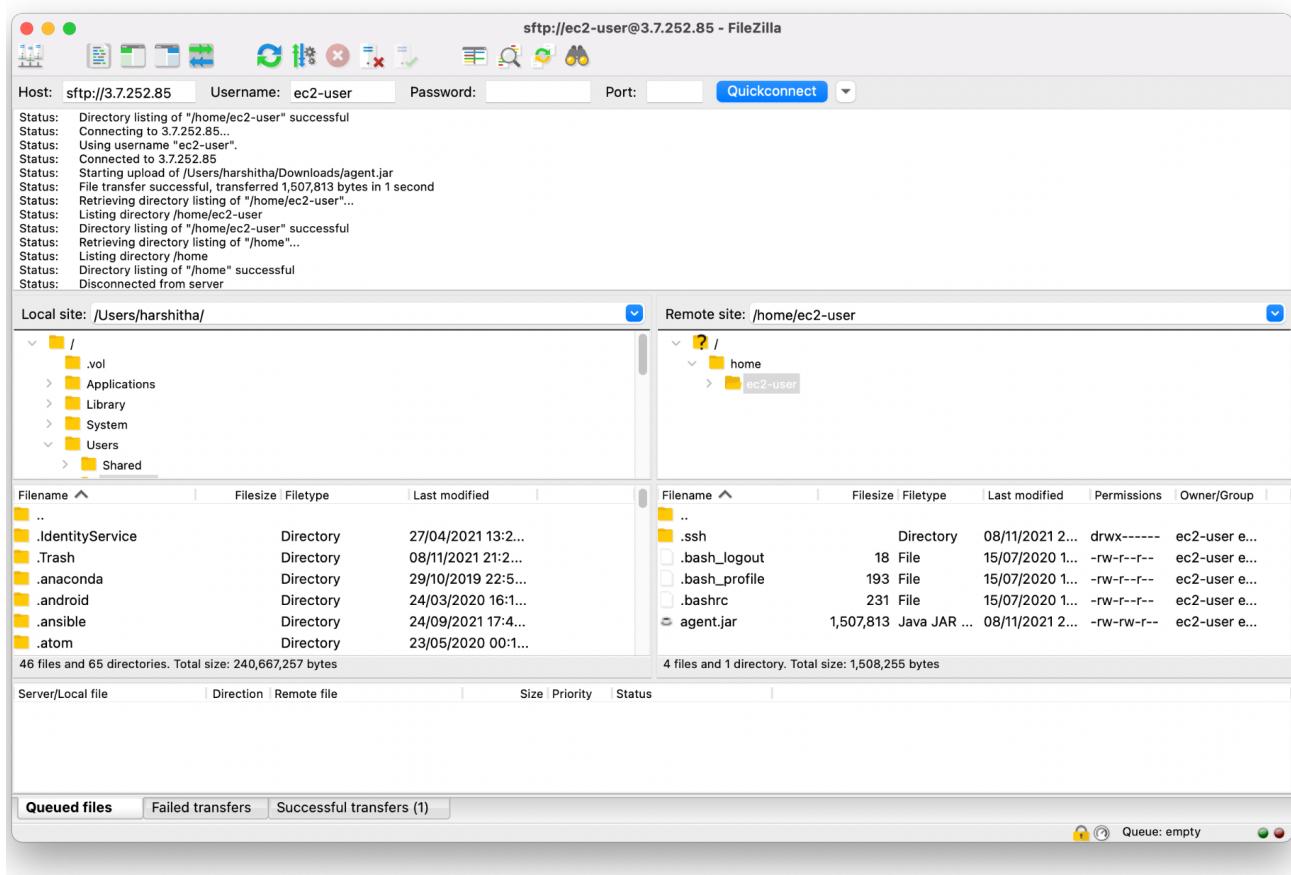
5.9.3. Paste the ip address, change username to ec2-user, and port number 22.



5.9.4. In setting -> SFTP, add the pem file.



5.9.5. After hosting, add the agent.jar file in ec2-user.



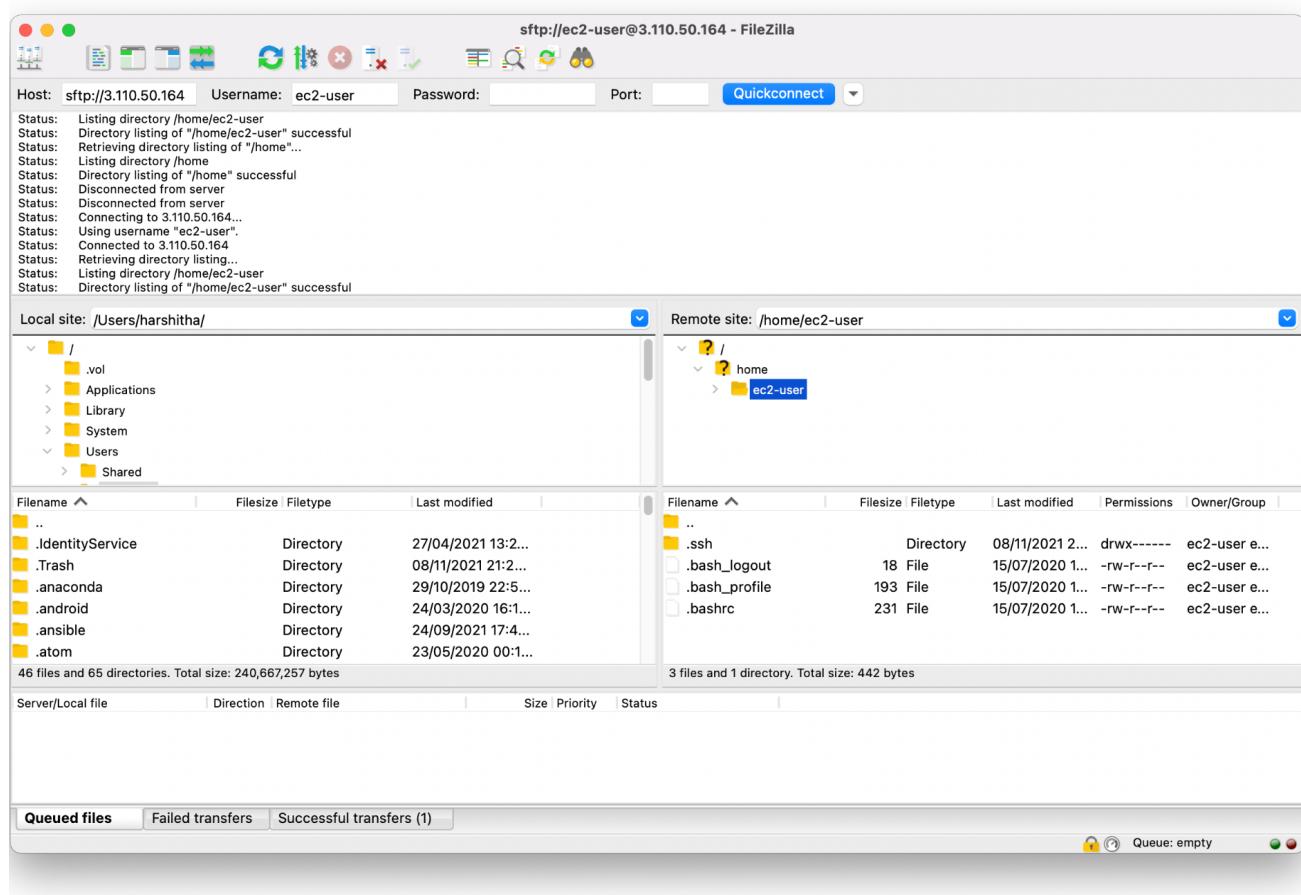
5.9.6. To verify, connect to slave-1 and type command- ls. Agent.jar is present.

```
Desktop — ec2-user@ip-172-31-13-3:~ — ssh -i master.pem ec2-user@ec2-3-7-252-85.ap-south-1.compute.amazonaws.com
Last login: Mon Nov  8 20:58:48 on ttys000
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT200050.
[Snamratha-18bc083-MacBook-Pro:~ harshitha$ cd Desktop
[Snamratha-18bc083-MacBook-Pro:Desktop harshitha$ chmod 400 master.pem
[Snamratha-18bc083-MacBook-Pro:Desktop harshitha$ ssh -i "master.pem" ec2-user@ec2-3-7-252-85.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-3-7-252-85.ap-south-1.compute.amazonaws.com (3.7.252.85)' can't be established.
ECDSA key fingerprint is SHA256:ogI2LM78ludcmKbEJc5q7B2UFxxOjo7E9Y8XPJE+q0s.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-7-252-85.ap-south-1.compute.amazonaws.com,3.7.252.85' (ECDSA) to the list of known hosts.

  _|_ _|_
  | (   /  Amazon Linux 2 AMI
  | \___|__|_

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-13-3 ~]$ ls
agent.jar
[ec2-user@ip-172-31-13-3 ~]$
```

5.10. Do the same for slave-2



sftp://ec2-user@3.7.252.85 - FileZilla

Host: 3.110.50.164 Username: ec2-user Password: Port: 22 Quickconnect

Status: Directory listing of "/home/ec2-user" successful
 Status: Connecting to 3.7.252.85...
 Status: Using username "ec2-user".
 Status: Connected to 3.7.252.85
 Status: Starting upload of /Users/harshitha/Downloads/agent.jar
 Status: File transfer successful, transferred 1,507,813 bytes in 1 second
 Status: Retrieving directory listing of "/home/ec2-user"..."
 Status: Listing directory /home/ec2-user
 Status: Directory listing of "/home/ec2-user" successful
 Status: Retrieving directory listing of "/home"..."
 Status: Listing directory /home
 Status: Directory listing of "/home" successful
 Status: Disconnected from server

Local site: /Users/harshitha/ Remote site: /home/ec2-user

Filename	Filesize	Filetype	Last modified	Permissions	Owner/Group
..					
.IdentityService		Directory	27/04/2021 13:2...		
.Trash		Directory	08/11/2021 21:2...		
.anaconda		Directory	29/10/2019 22:5...		
.android		Directory	24/03/2020 16:1...		
.ansible		Directory	24/09/2021 17:4...		
.atom		Directory	23/05/2020 00:1...		

46 files and 65 directories. Total size: 240,667,257 bytes

Server/Local file	Direction	Remote file	Size	Priority	Status
-------------------	-----------	-------------	------	----------	--------

Queued files Failed transfers Successful transfers (1)

Queue: empty

```
Desktop - ec2-user@ip-172-31-6-65:~ -- ssh -i master.pem ec2-user@ec2-3-110-50-164.ap-south1.compute.amazonaws.com
Last login: Mon Nov  8 21:59:36 on ttys001

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[SNamratha-18bcs083-MacBook-Pro:~ harshitha$ cd Desktop
[SNamratha-18bcs083-MacBook-Pro:Desktop harshitha$ chmod 400 master.pem
[SNamratha-18bcs083-MacBook-Pro:Desktop harshitha$ ssh -i "master.pem" ec2-user@ec2-3-110-50-164.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-3-110-50-164.ap-south-1.compute.amazonaws.com (3.110.50.164)' can't be established.
ECDSA key fingerprint is SHA256:QIglxlvAcrdB33zNEThzaeFNROPPsHoIiQM2D6jSA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-110-50-164.ap-south-1.compute.amazonaws.com,3.110.50.164' (ECDSA) to the list of known hosts.

  _|_ _|_
 _| ( _/_ 
---\_\_|_AMI

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-6-65 ~]$ ls
agent.jar
[ec2-user@ip-172-31-6-65 ~]$
```

- 5.11. Install open jdk on slave 1 and slave 2.
- 5.12. Now we have to execute the command given in jenkins for slave-1 and slave-2 at the place where agent.jar is present.

```
[ec2-user@ip-172-31-13-3 ~]$ java -jar agent.jar -jnlpUrl http://3.110.41.107:8080/computer/slave-1/jenkins-agent.jnlp -secret 176b64fb55870df5236c5b93c0f0fa0fd43e1fa579e1bdf53ccf81b36e13d4d -workDir "/home/ec2-user"
Nov 08, 2021 4:58:18 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ec2-user/remoting as a remoting work directory
Nov 08, 2021 4:58:19 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ec2-user/remoting
Nov 08, 2021 4:58:19 PM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: slave-1
Nov 08, 2021 4:58:19 PM org.jenkinsci.remoting.jnlp.Main$CuiListener <init>
INFO: Jenkins agent is running in headless mode.
Nov 08, 2021 4:58:19 PM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 4.10.1
Nov 08, 2021 4:58:19 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ec2-user/remoting as a remoting work directory

Nov 08, 2021 4:58:19 PM Hudson.remoting.jnlp.MainsCuiListener status
INFO: Locating server among [http://3.110.41.107:8080/]
Nov 08, 2021 4:58:19 PM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Nov 08, 2021 4:58:19 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
Agent address: 3.110.41.107
Agent port: 42959
Identity: 78:8d:f7:95:e6:e8:cd:22:8a:7c:69:e5:d5:f4:00:92
Nov 08, 2021 4:58:19 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Nov 08, 2021 4:58:19 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to 3.110.41.107:42959
Nov 08, 2021 4:58:19 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLP4-connect
Nov 08, 2021 4:58:19 PM org.jenkinsci.remoting.protocol.impl.BIONetworkLayer$Reader run
INFO: Waiting for ProtocolStack to start.
Nov 08, 2021 4:58:19 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: 78:8d:f7:95:e6:e8:cd:22:8a:7c:69:e5:d5:f4:00:92
Nov 08, 2021 4:58:21 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
[]
```

[←](#) [→](#) [C](#) Not Secure | 3.110.41.107:8080/computer/slave-1/

 Jenkins

Dashboard > Nodes > slave-1

[Back to List](#)

Status

[Delete Agent](#)

[Configure](#)

[Build History](#)

[Load Statistics](#)

[Script Console](#)

[Log](#)

[System Information](#)

[Disconnect](#)

Agent slave-1

Agent is connected.

Projects tied to slave-1

None

[Mark this node temporarily offline](#)

Build Executor Status

1 Idle

[Build History](#)

[REST API](#) Jenkins 2.30.3

3.110.41.107:8080/computer/slave-1/builds

agent.jar

FileZilla_3.5....tar.bz2

agent.jar

Show All

```
[ec2-user@ip-172-31-6-65 ~]$ java -jar agent.jar -jnlpUrl http://3.110.41.107:8080/computer/slave-2/jenkins-agent.jnlp -secret 1c82f8817d0caeac85d027732e6f38c57d08a215481d71819f37c5aa35f84b01 -workDir "/home/ec2-user"
Nov 08, 2021 4:55:48 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ec2-user/remoting as a remoting work directory
Nov 08, 2021 4:55:49 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ec2-user/remoting
Nov 08, 2021 4:55:48 PM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: slave-2
Nov 08, 2021 4:55:48 PM hudson.remoting.jnlp.MainsCuiListener <init>
INFO: Jenkins agent is running in headless mode.
Nov 08, 2021 4:55:48 PM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 4.10.1
Nov 08, 2021 4:55:48 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ec2-user/remoting as a remoting work directory
Nov 08, 2021 4:55:49 PM org.jenkinsci.remoting.jnlp.Main$CuiListener status
INFO: Locating server among [http://3.110.41.107:8080/]
Nov 08, 2021 4:55:49 PM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Nov 08, 2021 4:55:49 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
Agent address: 3.110.41.107
Agent port: 42959
Identity: 78:8d:f7:95:e6:e8:cd:22:8a:7c:69:e5:d5:f4:00:92
Nov 08, 2021 4:55:49 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Nov 08, 2021 4:55:49 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to 3.110.41.107:42959
Nov 08, 2021 4:55:49 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLP4-connect
Nov 08, 2021 4:55:49 PM org.jenkinsci.remoting.protocol.impl.BIONetworkLayer$Reader run
INFO: Waiting for ProtocolStack to start.
Nov 08, 2021 4:55:49 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: 78:8d:f7:95:e6:e8:cd:22:8a:7c:69:e5:d5:f4:00:92
Nov 08, 2021 4:55:50 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
[]
```

The screenshot shows the Jenkins interface for managing a slave node. The top navigation bar includes links for 'Dashboard', 'Nodes', and 'slave-2'. The main content area is titled 'Agent slave-2' and displays the message 'Agent is connected.' Below this, a section titled 'Projects tied to slave-2' shows 'None'. On the left, a sidebar lists various management options: 'Back to List', 'Status' (which is selected), 'Delete Agent', 'Configure', 'Build History', 'Load Statistics', 'Script Console', 'Log', 'System Information', and 'Disconnect'. A 'Build Executor Status' section indicates '1 Idle'. At the bottom, there are links for 'REST API' and 'Jenkins 2.303.3'. The browser's address bar shows the URL as 'Not Secure | 3.110.41.107:8080/computer/slave-2/'.

6. CI CD pipeline on jenkins

- 6.1. Get proj from <https://github.com/hshar/devopsIQ>
- 6.2. Install docker on both the slaves.

```
Desktop — ec2-user@ip-172-31-13-3:~ — ssh -i master.pem ec2-user@e...
Transaction test succeeded
Running transaction
  Installing : runc-1.0.0-2.amzn2.x86_64 1/5
  Installing : containerd-1.4.6-3.amzn2.x86_64 2/5
  Installing : libcgroup-0.41-21.amzn2.x86_64 3/5
  Installing : pigz-2.3.4-1.amzn2.0.1.x86_64 4/5
  Installing : docker-20.10.7-3.amzn2.x86_64 5/5
  Verifying  : docker-20.10.7-3.amzn2.x86_64 1/5
  Verifying  : containerd-1.4.6-3.amzn2.x86_64 2/5
  Verifying  : pigz-2.3.4-1.amzn2.0.1.x86_64 3/5
  Verifying  : runc-1.0.0-2.amzn2.x86_64 4/5
  Verifying  : libcgroup-0.41-21.amzn2.x86_64 5/5

Installed:
  docker.x86_64 0:20.10.7-3.amzn2

Dependency Installed:
  containerd.x86_64 0:1.4.6-3.amzn2           libcgroup.x86_64 0:0.41-21.amzn2
  pigz.x86_64 0:2.3.4-1.amzn2.0.1             runc.x86_64 0:1.0.0-2.amzn2

Complete!
[[ec2-user@ip-172-31-13-3 ~]$ docker --version
Docker version 20.10.7, build f0df350
[ec2-user@ip-172-31-13-3 ~]$ ]
```

```
Desktop — ec2-user@ip-172-31-6-65:~ — ssh -i master.pem ec2-user@e...
Transaction test succeeded
Running transaction
  Installing : runc-1.0.0-2.amzn2.x86_64 1/5
  Installing : containerd-1.4.6-3.amzn2.x86_64 2/5
  Installing : libcgroup-0.41-21.amzn2.x86_64 3/5
  Installing : pigz-2.3.4-1.amzn2.0.1.x86_64 4/5
  Installing : docker-20.10.7-3.amzn2.x86_64 5/5
  Verifying  : docker-20.10.7-3.amzn2.x86_64 1/5
  Verifying  : containerd-1.4.6-3.amzn2.x86_64 2/5
  Verifying  : pigz-2.3.4-1.amzn2.0.1.x86_64 3/5
  Verifying  : runc-1.0.0-2.amzn2.x86_64 4/5
  Verifying  : libcgroup-0.41-21.amzn2.x86_64 5/5

Installed:
  docker.x86_64 0:20.10.7-3.amzn2

Dependency Installed:
  containerd.x86_64 0:1.4.6-3.amzn2           libcgroup.x86_64 0:0.41-21.amzn2
  pigz.x86_64 0:2.3.4-1.amzn2.0.1             runc.x86_64 0:1.0.0-2.amzn2

Complete!
[[ec2-user@ip-172-31-6-65 ~]$ docker --version
Docker version 20.10.7, build f0df350
[ec2-user@ip-172-31-6-65 ~]$ ]
```

6.3. Configure jenkins to build project on slave 1, if successfully built, then do the same for slave 2.

6.3.1. Go to jenkins and create a job

6.3.1.1. Create job for slave 1

The screenshot shows the Jenkins 'Enter an item name' screen. A blue box highlights the input field containing the text 'Test'. Below the input field, a note says '» Required field'. To the right, there are five project types listed with icons: 'Freestyle project', 'Pipeline', 'Multi-configuration project', 'Folder', and 'Multibranch Pipeline'. Each type has a brief description. At the bottom right of the list is a blue 'OK' button. The background shows a dark header with the Jenkins logo and a user menu.

The screenshot shows the Jenkins 'Test/configure' screen under the 'General' tab. The 'GitHub project' section is selected, with the URL 'https://github.com/Namratha-shivaraju/task2.git' entered in the 'Project url' field. Other options like 'Discard old builds' and 'Restrict where this project can be run' are also visible. The 'Source Code Management' tab is open, showing 'Git' selected and the repository URL 'nivaraju/task2.git' entered. Buttons for 'Save' and 'Apply' are at the bottom.

Not Secure | 65.1.94.190:8080/job/Test/configure

Dashboard > Test >

General	Source Code Management	Build Triggers	Build Environment	Build	Post-build Actions
<input type="radio"/> None	<input checked="" type="radio"/> Git				
Repositories					
Repository URL <input type="text" value="https://github.com/Namratha-shivaraju/task2.git"/>					
Credentials - none - <input type="button" value="Add"/>					
<input type="button" value="Advanced..."/> <input type="button" value="Add Repository"/>					
Branches to build					
Branch Specifier (blank for 'any') <input type="text" value="/main"/>					
<input type="button" value="Add Branch"/>					
Repository browser					
(Auto) <input type="button" value="Save"/> <input type="button" value="Apply"/>					

6.3.1.2. Click build now

Not Secure | 65.1.94.190:8080/job/Test/

Jenkins

Dashboard > Test >

[Back to Dashboard](#)

- Status
- Changes
- Workspace
- Build Now**
- Configure
- Delete Project
- GitHub
- Rename

Project Test

[Workspace](#) [Recent Changes](#)

Permalinks

- Last build (#1), 2 min 39 sec ago
- Last failed build (#1), 2 min 39 sec ago
- Last unsuccessful build (#1), 2 min 39 sec ago
- Last completed build (#1), 2 min 39 sec ago

Build History [trend ^](#)

find

#2 Nov 9, 2021 9:28 AM

Success > Console Output

[Atom feed for all](#) [Atom feed for failures](#)

65.1.94.190:8080/job/Test/2/console

[add description](#) [Disable Project](#)

REST API Jenkins 2.303.3

Build #2 (Nov 9, 2021 9:28:59 AM)

No changes.

Started by user [namratha](#)

Revision: 48ee945600f47d036ba80dd44c01211b6a8a51b4
 Repository: <https://github.com/Namratha-shivaraju/task2.git>

- refs/remotes/origin/main

Console Output

```

Started by user namratha
Running as SYSTEM
Building remotely on slave-1 in workspace /home/ec2-user/workspace/Test
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/ec2-user/workspace/Test/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Namratha-shivaraju/task2.git # timeout=10
Fetching upstream changes from https://github.com/Namratha-shivaraju/task2.git
> git --version # timeout=10
> git --version # 'git version 2.32.0'
> git fetch --tags --force --progress -- https://github.com/Namratha-shivaraju/task2.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 48ee945600f47d036ba80dd44c01211b6a8a51b4 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 48ee945600f47d036ba80dd44c01211b6a8a51b4 # timeout=10
Commit message: "Add files via upload"
First time build. Skipping changelog.
Finished: SUCCESS
  
```

6.3.1.3. Check if the git repo is successfully cloned.

```
[ec2-user@ip-172-31-11-218 ~]$ ls
[agent.jar  remoting  workspace
[ec2-user@ip-172-31-11-218 ~]$ cd workspace
[ec2-user@ip-172-31-11-218 workspace]$ ls
Test
[ec2-user@ip-172-31-11-218 workspace]$ cd Test
[ec2-user@ip-172-31-11-218 Test]$ ls
docker-compose  Dockerfile  README.md  task2
[ec2-user@ip-172-31-11-218 Test]$
```

6.3.2. Deploy the website mentioned in git repo to project

Go to configure -> build -> execute shell.

The screenshot shows the Jenkins job configuration interface for a job named 'Test'. The 'Build' section is expanded, and the 'Execute shell' option is selected from a dropdown menu under 'Add build step'. Other options visible in the dropdown include 'Execute Windows batch command', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'.

Type the following code-

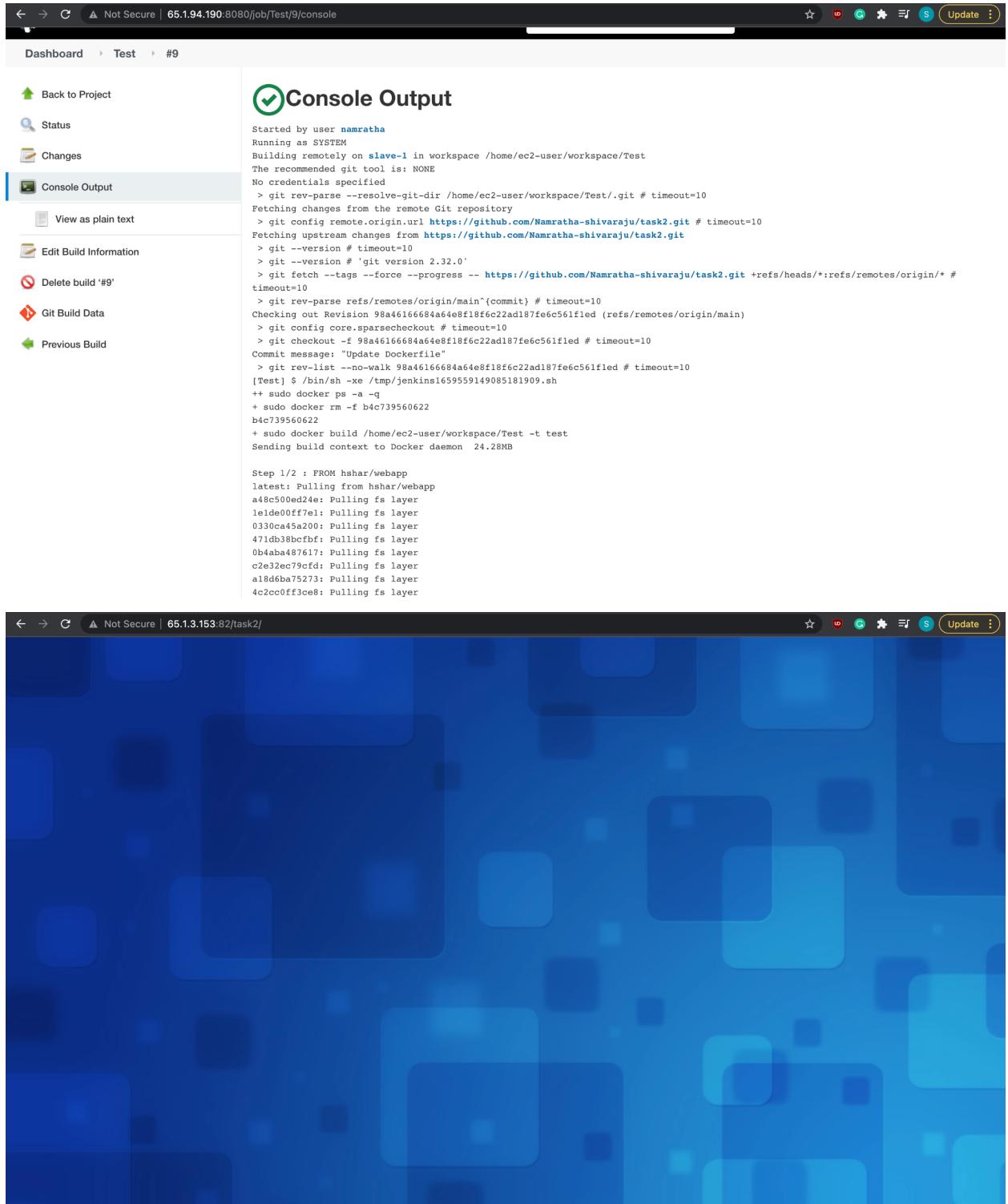
```
sudo docker rm -f $(sudo docker ps -a -q)
sudo docker build /home/ec2-user/workspace/Test -t test
sudo docker run -it -p 82:80 -d test
```

6.3.3. Save and proceed to build.

6.3.4. Before “build now”, go to slave-1 terminal and start a container. Once the container is started, go to jenkins and click on build now.

```
[[ec2-user@ip-172-31-11-218 ~]$ ls
agent.jar  remoting  workspace
[[ec2-user@ip-172-31-11-218 ~]$ cd workshop
-bash: cd: workshop: No such file or directory
[[ec2-user@ip-172-31-11-218 ~]$ cd workspace
[[ec2-user@ip-172-31-11-218 workspace]$ ls
Test
[[ec2-user@ip-172-31-11-218 workspace]$ cd Test
[[ec2-user@ip-172-31-11-218 Test]$ ls
docker-compose  Dockerfile  README.md  task2
[[ec2-user@ip-172-31-11-218 Test]$ sudo docker run -it -d ubuntu
227ab1268a92740bd78caf919d73c85d351845d5e7fc8ea19b8e183c60c279e3
[[ec2-user@ip-172-31-11-218 Test]$ ^C
[[ec2-user@ip-172-31-11-218 Test]$ sudo docker run -it -d ubuntu
b4c73956062269aa453fa138249cc16386bfdd0a36a3833e536175a0a36db937
```

6.3.5. Once the build has gone successful, verify the same.



The screenshot shows a Jenkins job console output for build #9. The left sidebar contains links for Back to Project, Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Delete build '#9', Git Build Data, and Previous Build. The main content area is titled 'Console Output' and displays the following log:

```
Started by user namratha
Running as SYSTEM
Building remotely on slave-1 in workspace /home/ec2-user/workspace/Test
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/ec2-user/workspace/Test/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Namratha-shivaraju/task2.git # timeout=10
Fetching upstream changes from https://github.com/Namratha-shivaraju/task2.git
> git --version # timeout=10
> git --version # 'git version 2.32.0'
> git fetch --tags --force --progress -- https://github.com/Namratha-shivaraju/task2.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^(commit) # timeout=10
Checking out Revision 98a46166684a64e8f18f6c22ad187fe6c561fled (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 98a46166684a64e8f18f6c22ad187fe6c561fled # timeout=10
Commit message: "Update Dockerfile"
> git rev-list --no-walk 98a46166684a64e8f18f6c22ad187fe6c561fled # timeout=10
[Test] $ /bin/sh -x /tmp/jenkins1659559149085181909.sh
++ sudo docker ps -a -q
+ sudo docker rm -f b4c739560622
b4c739560622
+ sudo docker build /home/ec2-user/workspace/Test -t test
Sending build context to Docker daemon 24.28MB

Step 1/2 : FROM hshar/webapp
latest: Pulling from hshar/webapp
a48c500ed24e: Pulling fs layer
leide00ff7e1: Pulling fs layer
0330ca45a200: Pulling fs layer
471db38bcfbf: Pulling fs layer
0b4ba487617: Pulling fs layer
c2e32ec79cf8: Pulling fs layer
a18d6ba75273: Pulling fs layer
4c2cc0ff3ce8: Pulling fs layer
```

6.3.6. Do the same configuration for slave-2

Not Secure | 65.1.94.190:8080/view/all/newJob

Jenkins

Dashboard > All >

Enter an item name

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
OK creates a set of multibranch project subfolders by scanning for repositories.

Not Secure | 65.1.94.190:8080/job/Prod/configure

Prod

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

[Plain text] Preview

Discard old builds

GitHub project

Project url

This build requires lockable resources

This project is parameterized

Throttle builds

Disable this project

Execute concurrent builds if necessary

Restrict where this project can be run

Label Expression

Label slave-2 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced...

Source Code Management

None

Git

Repositories

Save Apply

Not Secure | 65.1.94.190:8080/job/Prod/configure

Dashboard > Prod >

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

Repository URL
https://github.com/Namratha-shivaraju/task2.git

Credentials - none - Add

Advanced... Add Repository

Branches to build

Branch Specifier (blank for 'any')
*/main

Add Branch

Repository browser (Auto)

Additional Behaviours Add

Build Triggers

Save Apply

Build Environment

Add timestamps to the Console Output

Inspect build log for published Gradle build scans

With Ant

Build

Execute shell

Command

```
sudo docker rm -f $(docker ps -a -q)
sudo docker build /home/ec2-user/workspace/Prod -t production
sudo docker run -it -p 80:80 -d production
```

See the list of available environment variables

Advanced...

Add build step ▾

Post-build Actions

Add post-build action ▾

Save Apply

```
[[ec2-user@ip-172-31-13-52 ~]$ ls
agent.jar  remoting  workspace
[ec2-user@ip-172-31-13-52 ~]$ ls workspace
Prod
[ec2-user@ip-172-31-13-52 ~]$ ls Prod
ls: cannot access Prod: No such file or directory
[ec2-user@ip-172-31-13-52 ~]$ cd workspace
[[ec2-user@ip-172-31-13-52 workspace]$ ls
Prod
[[ec2-user@ip-172-31-13-52 workspace]$ cd Prod
[ec2-user@ip-172-31-13-52 Prod]$ sudo docker run -it -d ubuntu
[23889f1e713de2cf3df6cf0310042f2b52d592af9296cfe9ab3b0c44386b3221
```

← → C Not Secure | 65.1.94.190:8080/job/Prod/4/console

Dashboard > Prod > #4

[Back to Project](#)

[Status](#)

[Changes](#)

Console Output

[View as plain text](#)

[Edit Build Information](#)

[Delete build '#4'](#)

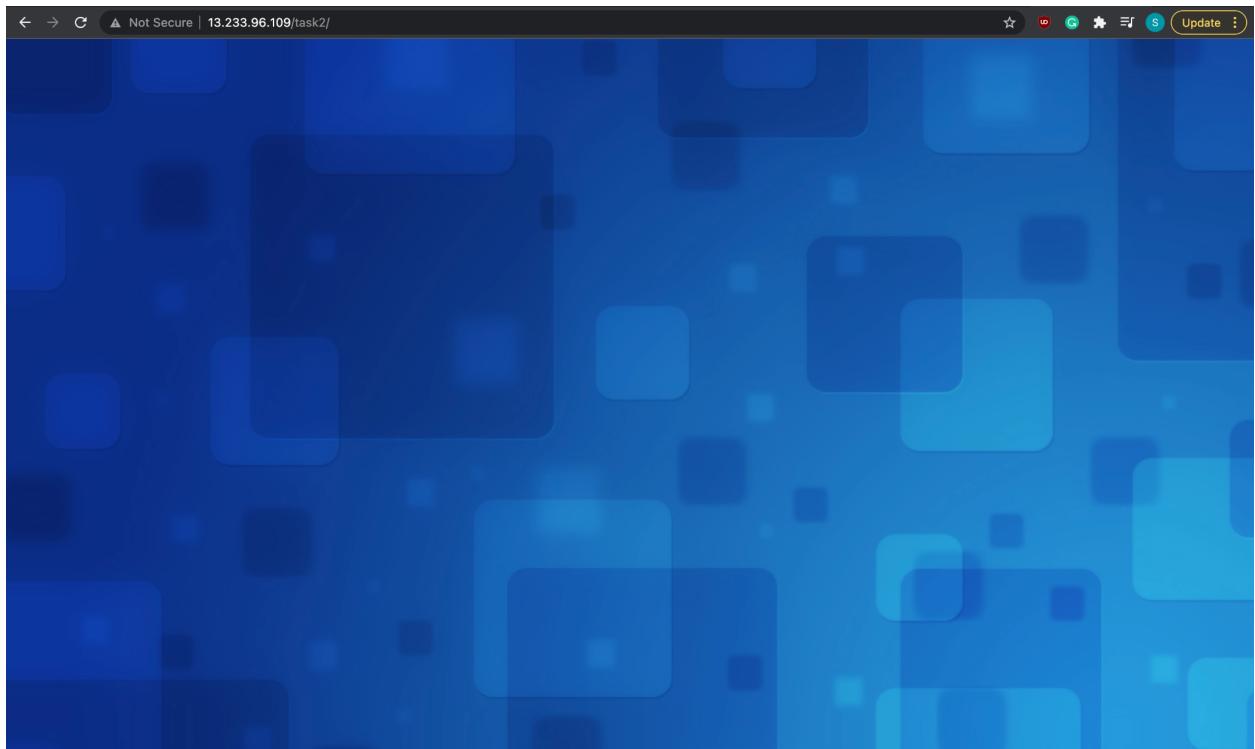
[Git Build Data](#)

[Previous Build](#)

Console Output

```
Started by user namratha
Running as SYSTEM
Building remotely on slave-2 in workspace /home/ec2-user/workspace/Prod
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/ec2-user/workspace/Prod/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Namratha-shivaraju/task2.git # timeout=10
Fetching upstream changes from https://github.com/Namratha-shivaraju/task2.git
> git --version # timeout=10
> git --version # 'git version 2.32.0'
> git fetch --tags --force --progress -- https://github.com/Namratha-shivaraju/task2.git +refs/heads/*:refs/remotes/origin/*
timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 98a46166684a64e8f18f6c22ad187fe6c561fled (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 98a46166684a64e8f18f6c22ad187fe6c561fled # timeout=10
Commit message: "Update Dockerfile"
> git rev-list --no-walk 98a46166684a64e8f18f6c22ad187fe6c561fled # timeout=10
[Prod] $ /bin/sh -x /tmp/jenkins8319007336611985584.sh
++ sudo docker ps -a -q
+ sudo docker rm -f 23889f1e713d
23889f1e713d
609e406e5595
+ sudo docker build /home/ec2-user/workspace/Prod -t production
Sending build context to Docker daemon 24.28MB

Step 1/2 : FROM hshar/webapp
--> 0cbcf1535ed8
Step 2/2 : ADD ./task2 /var/www/html/task2
--> Using cache
--> 029a689265b7
Successfully built 029a689265b7
Successfully tagged production:latest
+ sudo docker run -it -p 80:80 -d production
2aaedb5aaf583ffa83130cealab695ecd9e04583e87aecda0626f990fdf7baa
Finished: SUCCESS
```



6.3.7. Go to Test job -> configure -> add post build actions -> build other projects

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

- Abort the build if it's stuck
- Add timestamps to the Console Output
- Inspect build log for published Gradle build scans
- With Ant

Build

Execute shell

```
ps -a -q
/Workspace/Test -t test
```

Archive the artifacts

Build other projects

- Publish JUnit test result report
- Record fingerprints of files to track usage
- Git Publisher
- E-mail Notification
- Editable Email Notification
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]
- Delete workspace when build is done

Add post-build action ▾

Post-build Actions

Save Apply

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

```
sudo docker rm -f $(sudo docker ps -a -q)
sudo docker build /home/ec2-user/workspace/Test -t test
sudo docker run -it -p 82:80 -d test
```

See the list of available environment variables

Add build step ▾

Post-build Actions

Build other projects

Projects to build

Prod,

Trigger only if build is stable

Trigger even if the build is unstable

Trigger even if the build fails

Add post-build action ▾

Post-build Actions

Save Apply

6.3.8. Run the test job using pipeline view

6.3.8.1. Manage jenkins -> manage plugins -> available -> search “build pipeline” -> select “install without restart”.

Manage Jenkins

Building on the controller node can be a security issue. You should set the number of executors on the controller to 0. See [the documentation](#).

Java 11 is the recommended version to run Jenkins on; please consider upgrading.

System Configuration

- Configure System** Configure global settings and paths.
- Global Tool Configuration** Configure tools, their locations and automatic installers.
- Manage Nodes and Clouds** Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Manage Plugins** Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Security

- Configure Global Security** Secure Jenkins; define who is allowed to access/use the system.
- Manage Credentials** Configure credentials
- Manage Users** Create/delete/modify users that can log in to this Jenkins
- In-process Script Approval** Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions.

Status Information

Jenkins

Dashboard > Plugin Manager

Plugin Manager

Available

Install	Name	Version	Released
<input checked="" type="checkbox"/>	Build Pipeline	1.5.8	3 yr 11 mo ago
<input type="checkbox"/>	Webhook Step	1.4	1 yr 11 mo ago
<input type="checkbox"/>	Pipeline timeline	1.0.3	2 yr 9 mo ago

[Install without restart](#) [Download now and install after restart](#) Update information obtained: 18 hr ago [Check now](#)

Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Run Condition

Condition	Status
Javadoc	Success
Maven Integration	Installing
Conditional BuildStep	Pending
Parameterized Trigger	Pending
jQuery	Pending
Build Pipeline	Pending
Loading plugin extensions	Pending

[Go back to the top page](#)
(you can start using the installed plugins right away)

Restart Jenkins when installation is complete and no jobs are running

REST API Jenkins 2.303.3

6.3.9. After installation, go back to dashboard and click on “+”.

New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

New View

Build Queue

No builds in the queue.

Build Executor Status

Node	Status	Idle
master	idle	1
slave-1	idle	2
slave-2	idle	1

65.1.94.190:8080/newView

6.3.10. Select build pipeline view and name the view name as CICD.

The screenshot shows the Jenkins interface for creating a new view. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'Lockable Resources'. The main area is titled 'New View' and contains a 'View name' field with 'CICD' typed in. Below it, there are three radio button options: 'Build Pipeline View' (selected), 'List View', and 'My View'. Each option has a brief description. At the bottom right of the dialog is an 'OK' button. The background shows other Jenkins sections like 'Build Queue' and 'Build Executor Status'.

- 6.3.11. Enter the name- CICD.
Build pipeline view title- CICD.
Select initial job - Test
Ok

Not Secure | 65.1.94.190:8080/view/CICD/configure

Dashboard > **CICD** >

- New Item
- People
- Build History
- Edit View**
- Delete View
- Manage Jenkins
- My Views
- Lockable Resources
- New View

Build Queue
No builds in the queue.

Build Executor Status

Node	Status	Idle
master	idle	1
slave-1	idle	2
slave-2	idle	1
slave-3	idle	1

Name: CICD
Description:
 Filter build queue
 Filter build executors
Build Pipeline View Title: CICD

Pipeline Flow

Layout: Based on upstream/downstream relationship

This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for extension.

Upstream / downstream config

Select Initial Job

Trigger Options

OK Apply

Not Secure | 65.1.94.190:8080/view/CICD/

Jenkins

Dashboard > CICD >

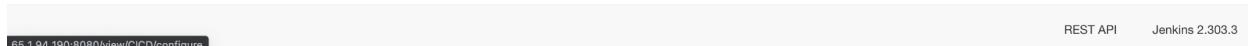
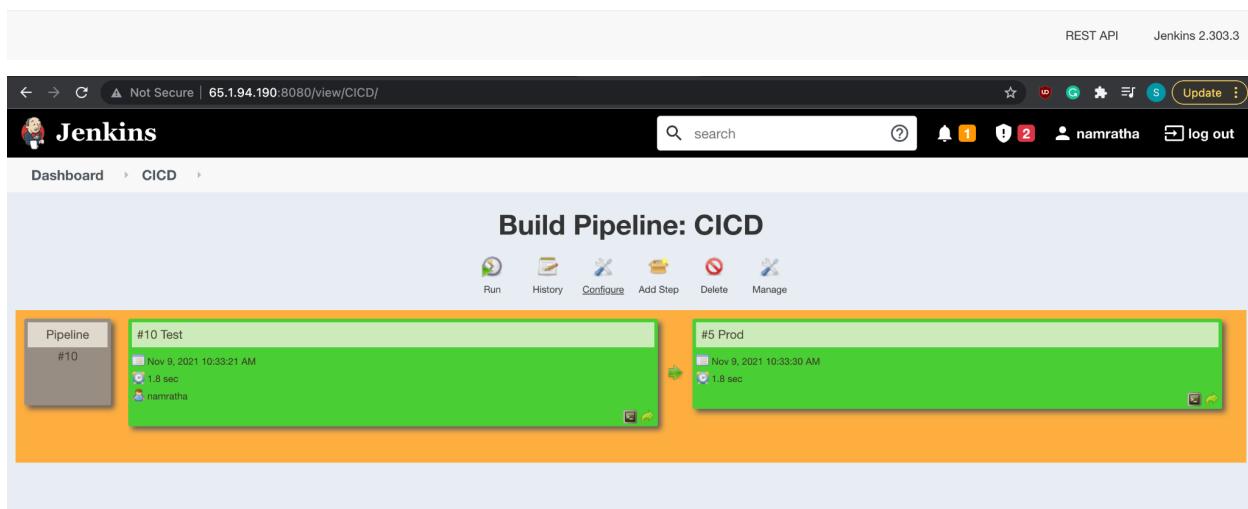
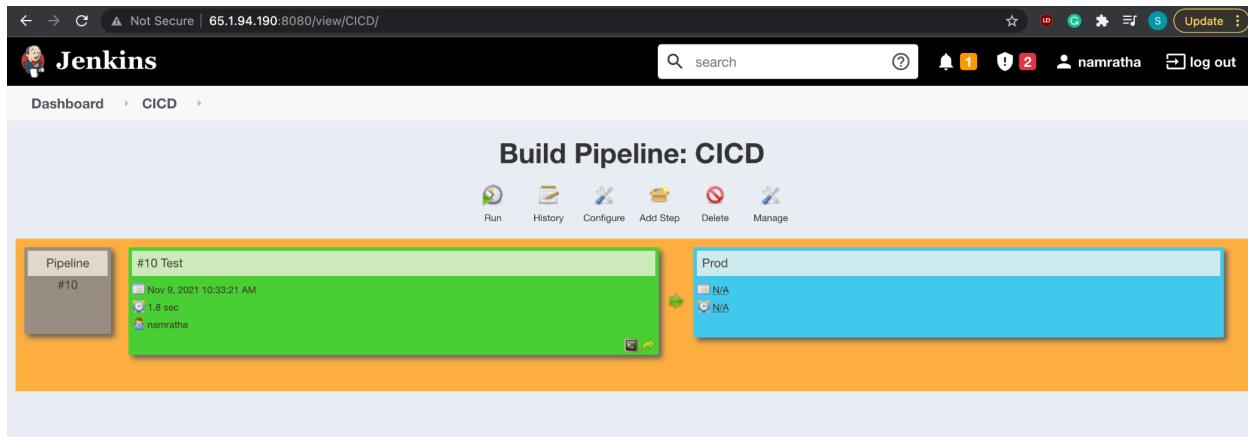
Build Pipeline: CICD

Run History Configure Add Step Delete Manage

Pipeline

- #9 Test
 - Nov 9, 2021 10:09:50 AM
 - 25 sec
 - namratha
- Prod
 - N/A
 - N/A

6.3.12. Run these jobs



6.3.13. Trigger the jenkins job when a github commit is made.

6.3.13.1. Go to test job -> configure -> build triggers -> GitHub hook trigger food GITScm polling.

The screenshot shows the Jenkins job configuration interface for a job named 'Test'. The 'Build Triggers' tab is selected. Under 'Additional Behaviours', there is an 'Add' button. The 'Build Triggers' section contains several options: 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling' (which is checked), and 'Poll SCM'. The 'Build Environment' section includes options like 'Delete workspace before build starts', 'Use secret text(s) or file(s)', 'Abort the build if it's stuck', 'Add timestamps to the Console Output', 'Inspect build log for published Gradle build scans', and 'With Ant'. The 'Build' section shows a command box containing the following script:

```
sudo docker rm -f $(sudo docker ps -a -q)
docker ps -a -q
aws s3 cp workspace/Test -t test
2:80 -d test
```

Buttons for 'Save' and 'Apply' are visible at the bottom of the command box.

6.3.13.2. Configure webhook-

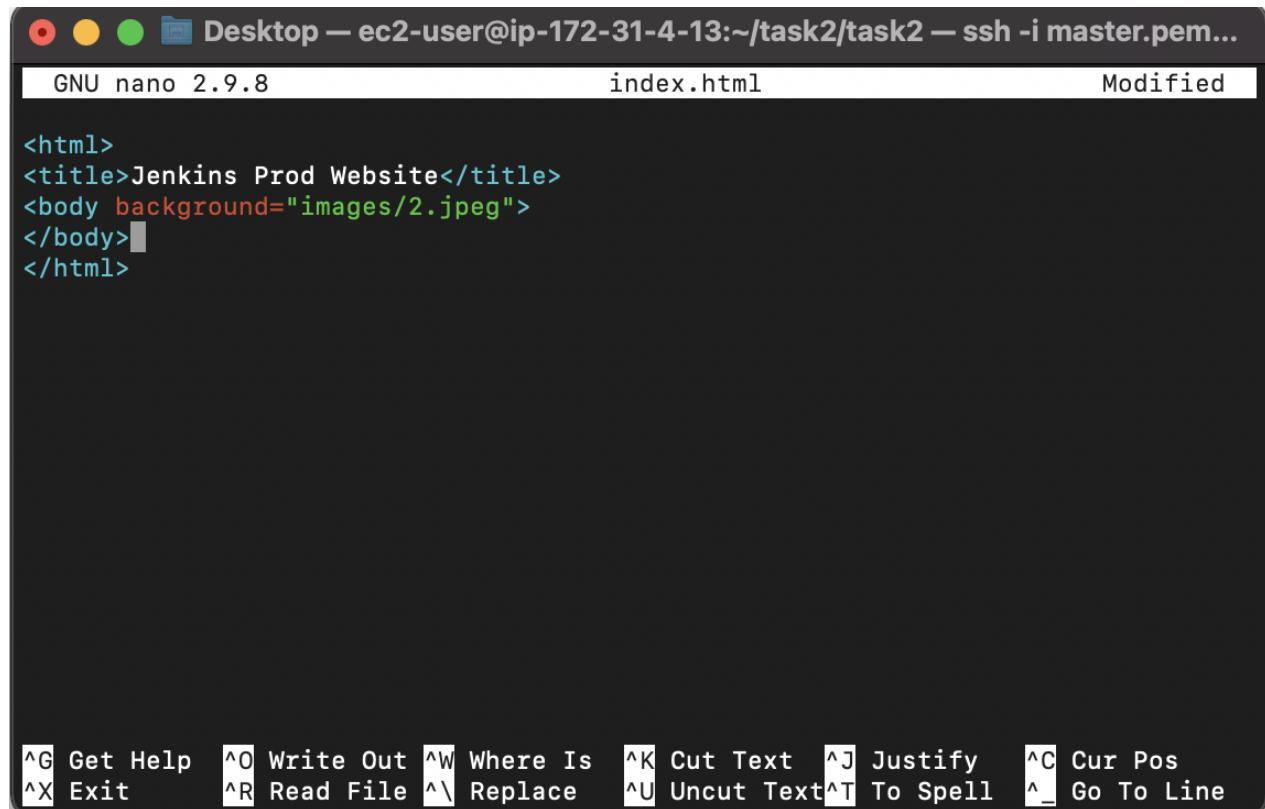
Go to github repo -> settings -> webhooks.

The screenshot shows the GitHub 'Webhooks / Add webhook' configuration page. On the left, a sidebar lists options: Options, Manage access, Security & analysis, Branches, Webhooks (which is selected and highlighted in red), Notifications, Integrations, Deploy keys, Actions, Environments, Secrets, Pages, and Moderation settings. The main right-hand panel has a heading 'Webhooks / Add webhook' and a sub-instruction: 'We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.' It contains fields for 'Payload URL' (set to 'http://65.1.94.190:8080/github-webhook/'), 'Content type' (set to 'application/x-www-form-urlencoded'), and a 'Secret' field (empty). Below these are sections for selecting events: 'Which events would you like to trigger this webhook?' with radio buttons for 'Just the push event.', 'Send me everything.', and 'Let me select individual events.' (the latter is selected), and an 'Active' checkbox (which is checked). At the bottom is a green 'Add webhook' button.

- 6.3.14. Push a commit on github to automatically trigger build. (done on master server)

```
[ec2-user@ip-172-31-4-13 ~]$ git clone https://github.com/Namratha-shivaraju/tas  
k2.git  
Cloning into 'task2'...  
remote: Enumerating objects: 28, done.  
remote: Counting objects: 100% (28/28), done.  
remote: Compressing objects: 100% (26/26), done.  
remote: Total 28 (delta 1), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (28/28), 11.45 MiB | 7.56 MiB/s, done.  
Resolving deltas: 100% (1/1), done.  
[ec2-user@ip-172-31-4-13 ~]$
```

- 6.3.15. Edit index.html to see changes.



The screenshot shows a terminal window titled "Desktop — ec2-user@ip-172-31-4-13:~/task2/task2 — ssh -i master.pem...". The window contains the following text:

```
GNU nano 2.9.8           index.html           Modified  
  
<html>  
<title>Jenkins Prod Website</title>  
<body background="images/2.jpeg">  
</body>  
</html>
```

At the bottom of the screen, there is a menu of keyboard shortcuts:

```
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```