



INDIAN INSTITUTE OF
INFORMATION
TECHNOLOGY

DevOps (CS457)

ASSIGNMENT-2: Task 2

Jenkins Master Slave pipeline

Submitted to:

Dr. Uma S

Submitted by Team 1:

Sumith Sai Budde (18BCS101)

Syed Sufyan Ahmed (18BCS103)

Shaik Fharook (18BCS091)

Parvati Jayakumar (18BEC036)

P Chethan Krishna (18BEC040)

G Rithika (18BCS031)

Step 1: Create Jenkins Master-Slave on AWS.

Step 1.1: Name them as ‘jenkins-master’, ‘jenkins-slave1’ and ‘jenkins-slave2’. ‘jenkins-master’ is the master node and the other two are slaves.

The screenshot shows the AWS EC2 Instances page with three instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv...
jenkins-master	i-0b14bcc92c0db90ea	Running	t2.micro	2/2 checks passed	No alarms	us-east-2b	ec2-18-218-1...
jenkins-slave2	i-019192d2cf339add3	Running	t2.micro	2/2 checks passed	No alarms	us-east-2c	ec2-18-191-1...
jenkins-slave1	i-02139ee6590573232	Running	t2.micro	2/2 checks passed	No alarms	us-east-2c	ec2-13-58-25...

Step 1.2: Make sure all the ports are accessible.

The screenshot shows the AWS EC2 Security group details page for the instance with ID i-0b14bcc92c0db90ea. The security group is named launch-wizard-21.

Inbound rules:

Security group rule ID	Port range	Protocol	Source	Security groups
sgr-0fb79b7b836e38920	All	All	::/0	launch-wizard-21
sgr-027002b33d45200c9	22	TCP	0.0.0.0/0	launch-wizard-21
sgr-0f37cf75569205d3a	All	All	0.0.0.0/0	launch-wizard-21

Outbound rules:

Security group rule ID	Port range	Protocol	Destination	Security groups
------------------------	------------	----------	-------------	-----------------

Step 1.3: Connect the ubuntu instance through a SSH client.

```
ubuntu@ip-172-31-21-118:~/team1
parvati@LAPTOP-S2R1236D:~/Team1$ ssh -i "Team1.pem" ubuntu@ec2-18-218-1-208.us-east-2.compute.amazonaws.com
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

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

System information as of Wed Nov 10 11:11:22 UTC 2021

System load:  0.05           Processes:      102
Usage of /:   17.7% of 7.69GB  Users logged in:    0
Memory usage: 20%            IPv4 address for eth0: 172.31.21.118
Swap usage:   0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
ubuntu@ip-172-31-21-118:~/Team1
```

Step 2: Install Jenkins on the master node.

Step 2.1: Get the updates

Step 2.2: Install Java

Step 2.3: Add the key, update jenkins.list and install jenkins

```
ubuntu@ip-172-31-21-118:~/Team1$ wget -q -O https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
OK
ubuntu@ip-172-31-21-118:~/Team1$ sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
ubuntu@ip-172-31-21-118:~/Team1$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:7 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [20.9 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [8844 B]
Fetched 146 kB in 1s (235 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-21-118:~/Team1$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  daemon net-tools
The following NEW packages will be installed:
  daemon jenkins net-tools
0 upgraded, 3 newly installed, 0 to remove and 18 not upgraded.
Need to get 72.2 MB of archives.
After this operation, 73.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 daemon amd64 0.6.4-1build2 [96.3 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60+git20180626.aebd88e-1ubuntu1 [196 kB]
Get:3 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.303.3 [71.9 MB]
Fetched 72.2 MB in 5s (14.5 MB/s)
Unpacking daemon (0.6.4-1build2) ...

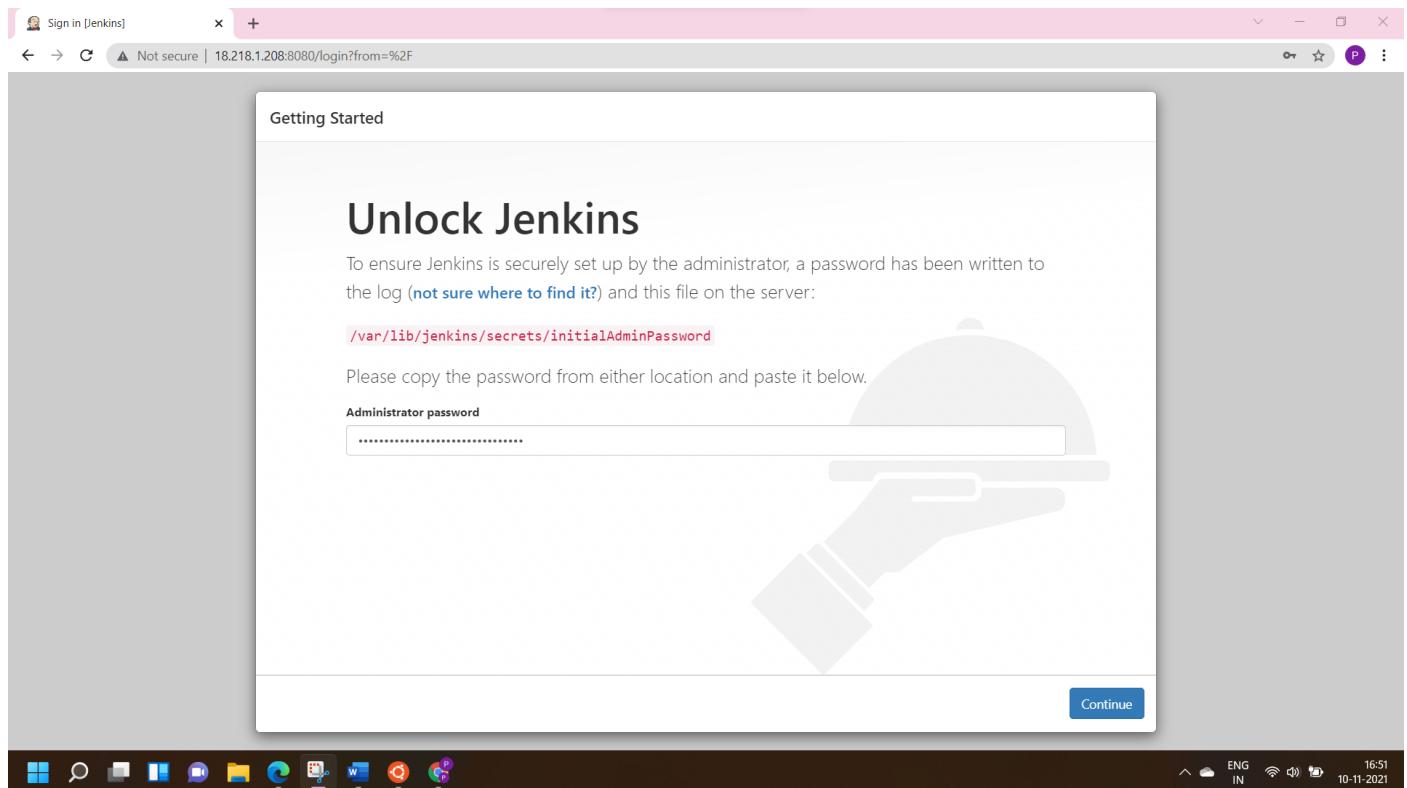
```

Step 2.4: Check if the Jenkins service is installed successfully.

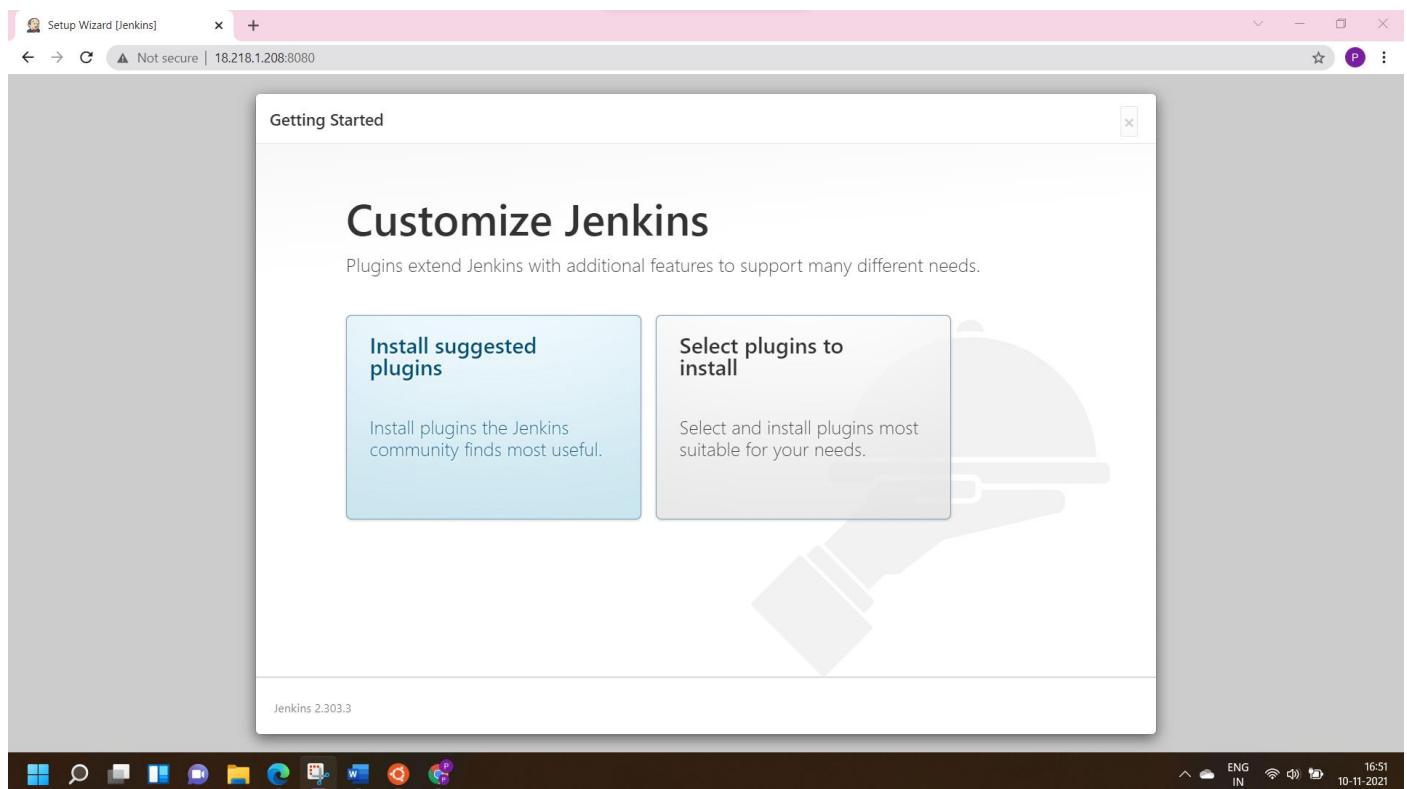
```
ubuntu@ip-172-31-21-118:~/Team1$ service jenkins status
● jenkins.service - LSB: Start Jenkins at boot time
  Loaded: loaded (/etc/init.d/jenkins; generated)
  Active: active (exited) since Wed 2021-11-10 11:16:51 UTC; 4min 2s ago
    Docs: man:systemd-sysv-generator(8)
   Tasks: 0 (limit: 1154)
  Memory: 0B
 CGroup: /system.slice/jenkins.service

Nov 10 11:16:49 ip-172-31-21-118 systemd[1]: Starting LSB: Start Jenkins at boot time...
Nov 10 11:16:49 ip-172-31-21-118 jenkins[6076]: Correct java version found
Nov 10 11:16:49 ip-172-31-21-118 jenkins[6076]: * Starting Jenkins Automation Server jenkins
Nov 10 11:16:49 ip-172-31-21-118 su[6109]: (to jenkins) root on none
Nov 10 11:16:49 ip-172-31-21-118 su[6109]: pam_unix(su-1:session): session opened for user jenkins by (uid=0)
Nov 10 11:16:49 ip-172-31-21-118 su[6109]: pam_unix(su-1:session): session closed for user jenkins
Nov 10 11:16:51 ip-172-31-21-118 jenkins[6076]: ...done.
Nov 10 11:16:51 ip-172-31-21-118 systemd[1]: Started LSB: Start Jenkins at boot time.
ubuntu@ip-172-31-21-118:~/Team1$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
a181cee95ec04dc1a214e031550be59
ubuntu@ip-172-31-21-118:~/Team1$
```

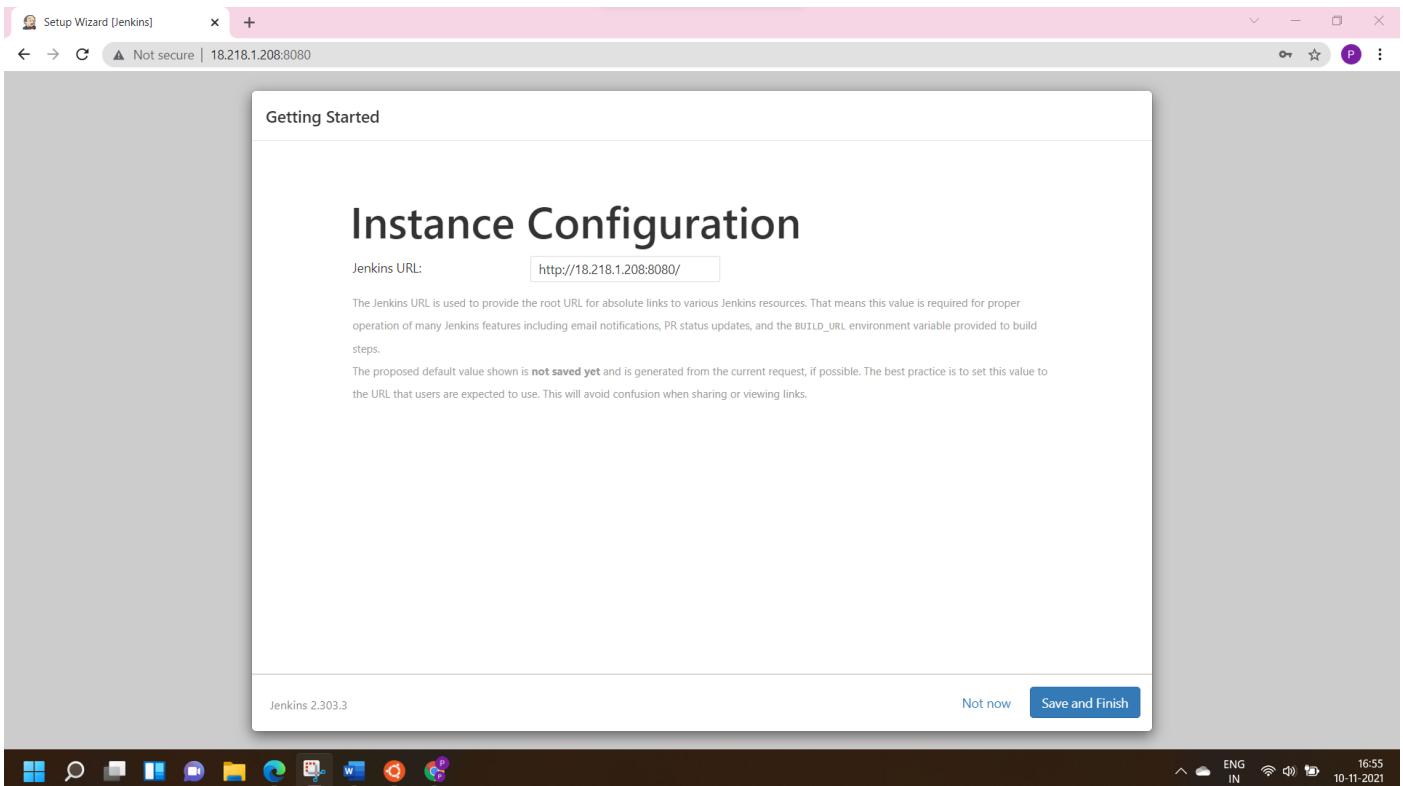
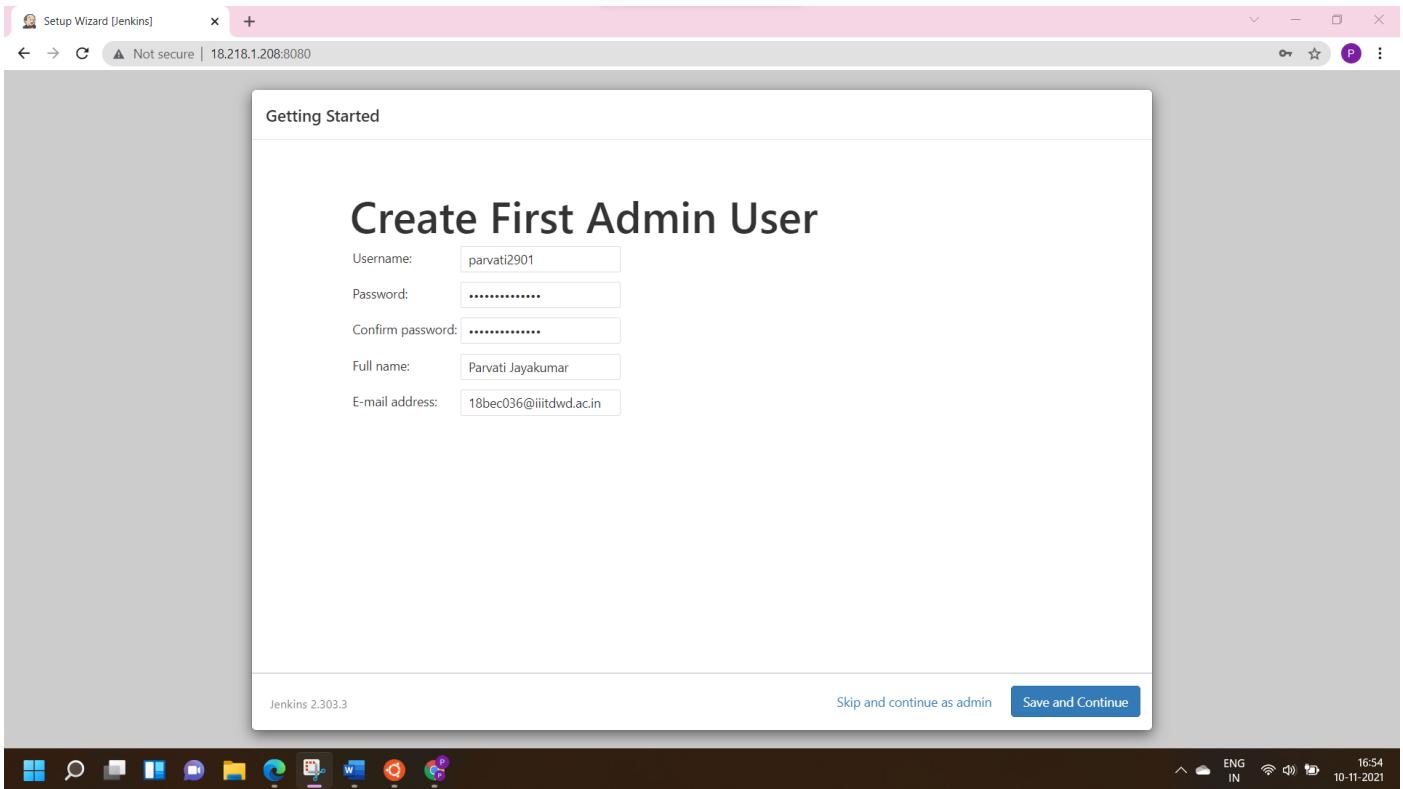
Step 2.5: On a web browser, access <ipv4_address><port_no> to unlock Jenkins and access the dashboard.

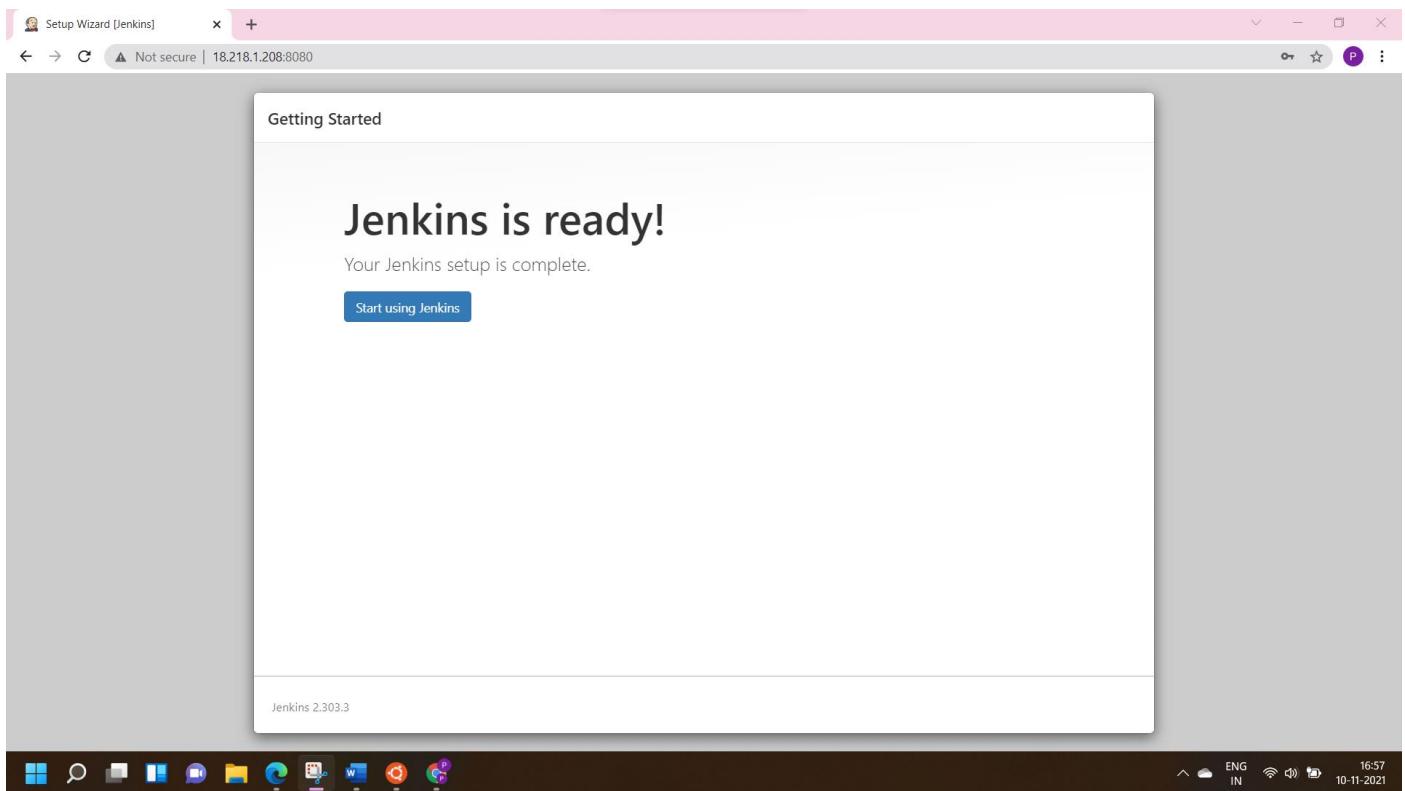


Step 2.6: Install the suggested Plugins



Step 2.7: Add in all the details to create the admin user. Further, press ‘Save and Continue’ button till the ‘Jenkins is ready screen pops up’.





Step 2.8: Configure Global security: Keep the TCP port for inbound agents as 'Random'.

The screenshot shows the Jenkins "Configure Global Security" page. The URL in the address bar is "Not secure | 18.218.1.208:8080/configureSecurity/". The page title is "Configure Global Security". The left sidebar shows "Dashboard > Configure Global Security". The main content area includes sections for "Markup Formatter" (set to "Plain text"), "Agents" (TCP port for inbound agents set to "Random"), "CSRF Protection" (Crumb Issuer settings), "Hidden security warnings" (with a "Security warnings..." button), and "API Token" (checkbox for generating a legacy API token). At the bottom are "Save" and "Apply" buttons. The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating "ENG IN" and the date "10-11-2021" at 17:06.

Step 3: Create the slaves' nodes on dashboard and copy the agent.jar file to both the slaves

Step 3.1: Create the slaves' nodes on dashboard.

The screenshot shows the Jenkins dashboard with the 'Nodes' section selected. A new node is being created with the name 'jenkins-slave1'. The 'Permanent Agent' type is selected. The 'Build Queue' and 'Build Executor Status' sections are also visible.

Step 3.2: Enter the working directory and press 'Save' button.

The screenshot shows the configuration page for the 'jenkins-slave1' node. The 'Custom WorkDir path' is set to '/home/ubuntu/Team1/jenkins'. The 'Save' button is visible at the bottom.

Step 3.3: Repeat the same steps for creating the 2nd slave node. The nodes created finally looks like this:

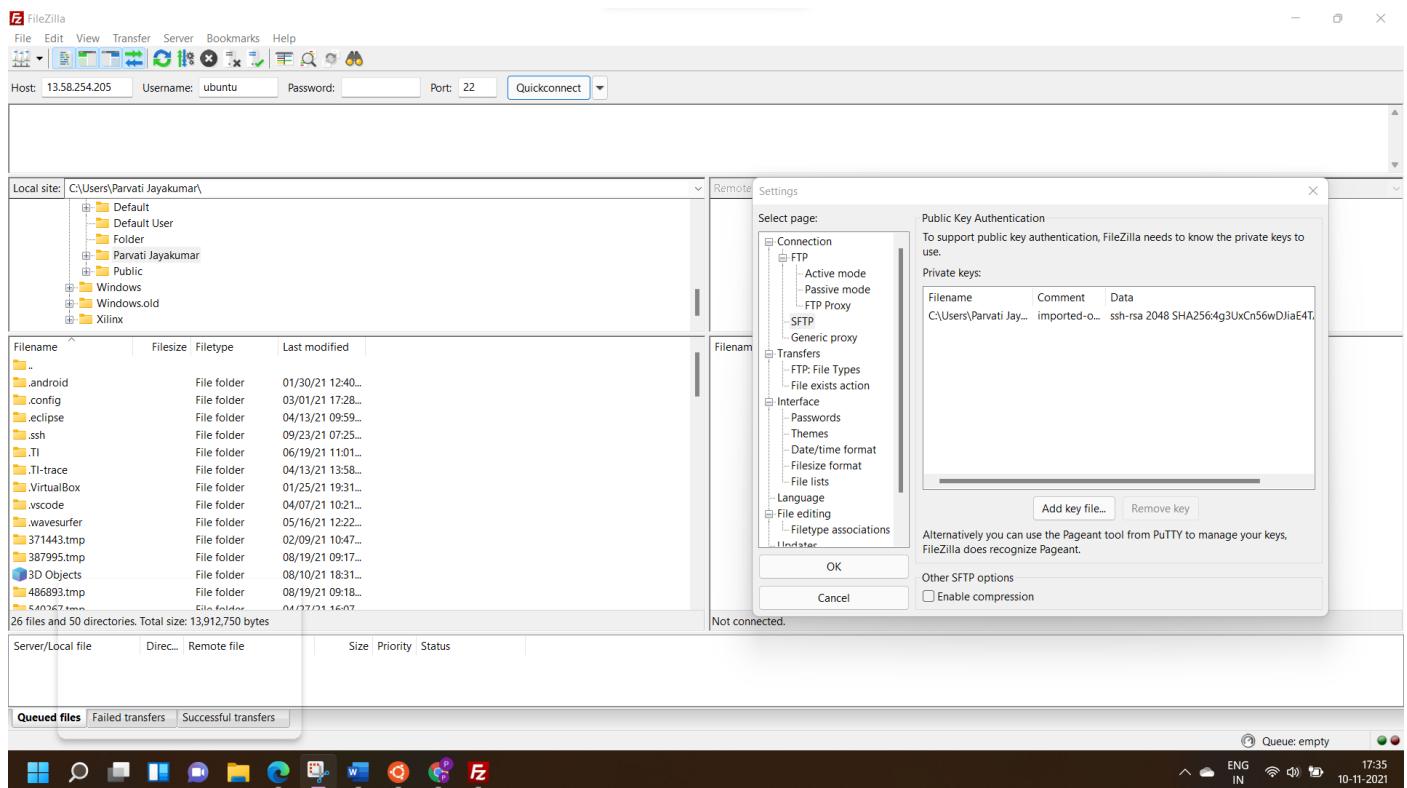
The screenshot shows the Jenkins 'Nodes' page. On the left sidebar, there are links for Back to Dashboard, Manage Jenkins, New Node, Configure Clouds, and Node Monitoring. Under 'Build Queue', it says 'No builds in the queue.' Under 'Build Executor Status', it lists 'master' (1 idle, 2 idle) and 'jenkins-slave1 (offline)' and 'jenkins-slave2 (offline)'. The main table lists three nodes: 'jenkins-slave1', 'jenkins-slave2', and 'master'. The 'master' node is highlighted. The table columns include S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, Response Time, and Data obtained. The 'master' node has a status of 'In sync', 5.31 GB free disk space, 5.31 GB free swap space, 0ms response time, and a data obtained time of 2 min 20 sec. A 'Refresh status' button is at the bottom right of the table.

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	jenkins-slave1		N/A	N/A	N/A	N/A	N/A
	jenkins-slave2		N/A	N/A	N/A	N/A	N/A
	master	Linux (amd64)	In sync	5.31 GB	0 B	5.31 GB	0ms

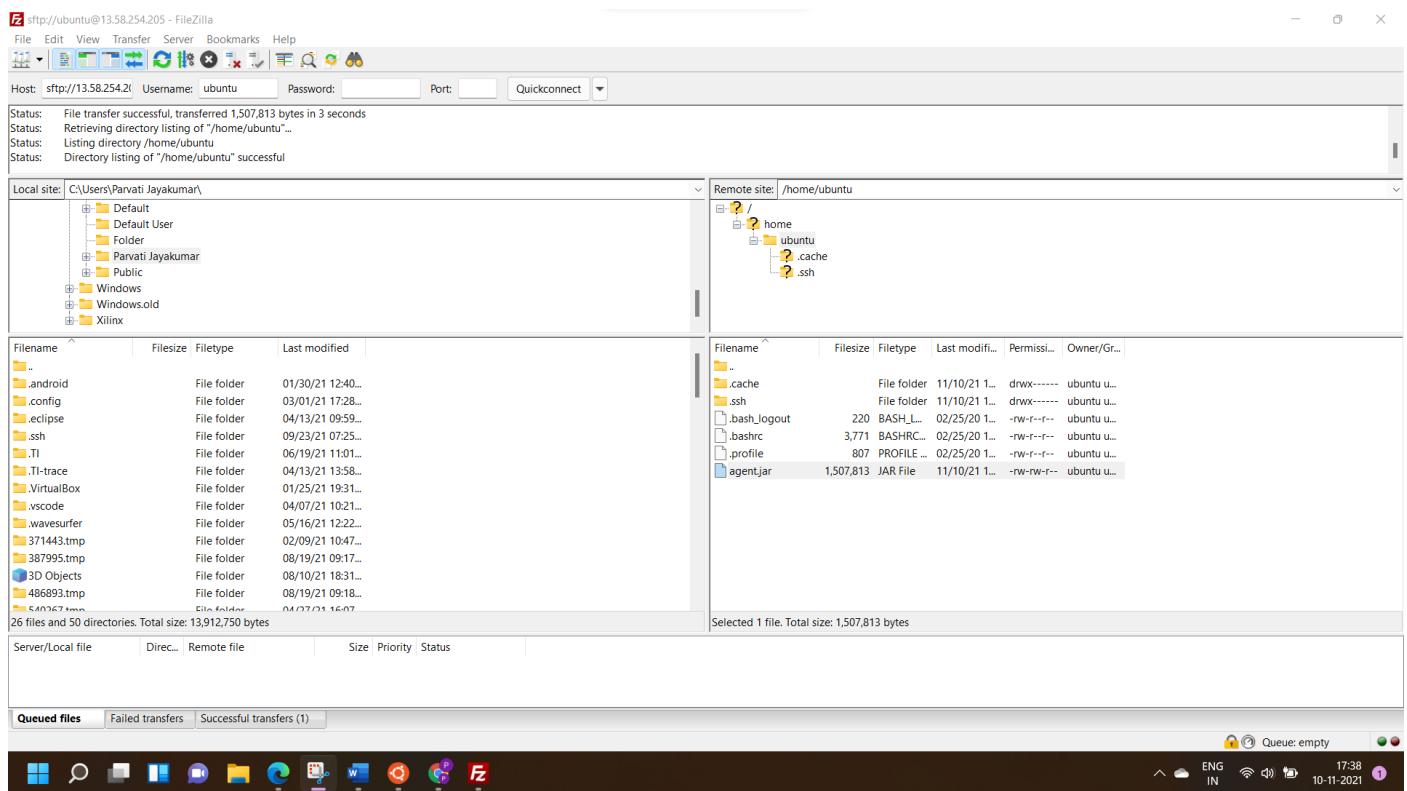
Step 3.4: Go to jenkins-slave1 and download the agent.jar file:

The screenshot shows the Jenkins 'jenkins-slave1' node page. On the left sidebar, there are links for Back to List, Status, Delete Agent, Configure, Build History, Load Statistics, and Log. Under 'Build Executor Status', it says 'Projects tied to jenkins-slave1: None'. The main content area shows instructions to connect the agent: 'Connect agent to Jenkins one of these ways:' with options to 'Launch' or 'Run from agent command line'. It provides the command: 'java -jar agent.jar -jnlpUrl http://18.218.1.208:8080/computer/jenkins-slave1/jenkins-agent.jnlp -secret c145f7677df78daedcdfac49cc1a88b63f66bbcd30e02ea00fefaf1d20c7d19b -workDir "/home/ubuntu/Team1/jenkins"' and 'Run from agent command line, with the secret stored in a file:' with the command: 'echo c145f7677df78daedcdfac49cc1a88b63f66bbcd30e02ea00fefaf1d20c7d19b > secret-file java -jar agent.jar -jnlpUrl http://18.218.1.208:8080/computer/jenkins-slave1/jenkins-agent.jnlp -secret @secret-file -workDir "/home/ubuntu/Team1/jenkins"'. A 'Mark this node temporarily offline' button is at the top right. At the bottom, there is a download link for 'agent.jar'.

Step 3.5: Our next step is to copy the agent.jar file to ‘jenkins-slave2’ machine. Install FileZilla and launch the application. Further, we will have to add the private key ‘Team1.pem’ since we are working on a remote machine.



Step 3.6: Specify the ipv4 address of ‘jenkins-slave1’, give the username as ‘ubuntu’, mention the port number as 22 and press on ‘Quickconnect’ button. Once the directory listing is successful, copy the agent.jar and place it in the ‘ubuntu’ folder.



Step 3.7: Verify that the file is placed properly in the slave1's machine.

```
ubuntu@ip-172-31-34-49:~$ cd Team1
parvati@LAPTOP-S2R1236D:~/Team1$ ssh -i "Team1.pem" ubuntu@ec2-13-58-254-205.us-east-2.compute.amazonaws.com
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

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

System information as of Wed Nov 10 12:12:55 UTC 2021

System load: 0.0          Processes:      102
Usage of /: 17.9% of 7.69GB   Users logged in:    0
Memory usage: 20%           IPv4 address for eth0: 172.31.34.49
Swap usage:  0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Wed Nov 10 12:09:56 2021 from 49.37.186.133
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

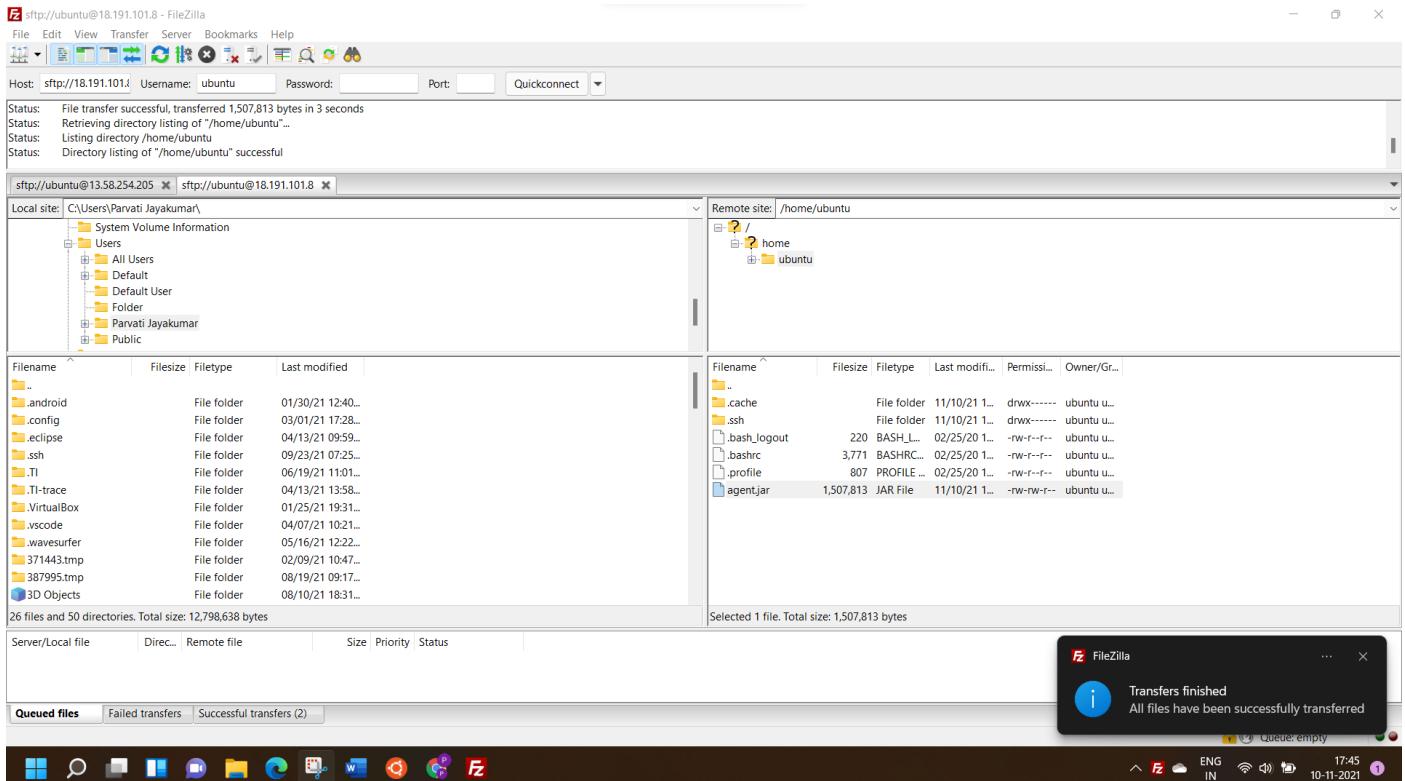
ubuntu@ip-172-31-34-49:~$ ls
agent.jar
ubuntu@ip-172-31-34-49:~$
```

Step 3.8: Repeat the same steps in 'jenkins-slave2' as well.

The screenshot shows the Jenkins master interface with the following details:

- Left Sidebar:** Includes links for Back to List, Status, Delete Agent, Configure, Build History, Load Statistics, and Log.
- Central Panel:** Title: **Agent jenkins-slave2**. Subtitle: Connect agent to Jenkins one of these ways:
 - Launch** (button) Launch agent from browser
 - Run from agent command line:

```
java -jar agent.jar -jnlpUrl http://18.218.1.208:8080/computer/jenkins-slave2/jenkins-agent.jnlp -secret bab2d70e370c4e5b94327dc343473cd4996a45806d3a232619c0f661a6702cb2 -workDir "/home/ubuntu/Team1/jenkins"
```
- Bottom Panel:** REST API Jenkins 2.303.3



```
ubuntu@ip-172-31-41-33:~$ cd Team1
parvati@LAPTOP-S2R1236D:~/Team1$ ssh -i "Team1.pem" ubuntu@ec2-18-191-101-8.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-191-101-8.us-east-2.compute.amazonaws.com (18.191.101.8)' can't be established.
EDDSA key fingerprint is SHA256:vF58wHleyE3H8KKCL3l4twsapBAR76dpctw3uOCJJoo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-191-101-8.us-east-2.compute.amazonaws.com,18.191.101.8' (EDDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

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

 System information as of Wed Nov 10 12:16:50 UTC 2021

 System load:  0.0           Processes:      102
 Usage of /:   17.9% of 7.69GB  Users logged in:     0
 Memory usage: 20%           IPv4 address for eth0: 172.31.41.33
 Swap usage:   0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-41-33:~$ ls
agent.jar
ubuntu@ip-172-31-41-33:~$
```

Step 4: Install Java and docker on both the slaves

Step 4.1: Jenkins-slave1

```
ubuntu@ip-172-31-34-49:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1341 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [275 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [14.4 kB]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [569 kB]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [81.6 kB]
Get:16 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f Metadata [528 B]
Get:17 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [871 kB]
Get:18 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [188 kB]
Get:19 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [19.5 kB]
Get:20 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [24.6 kB]
Get:21 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [6856 B]
Get:22 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [616 B]
Get:23 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [2568 B]
Get:24 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/main Translation-en [1120 B]
Get:25 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [400 B]
Get:26 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [6584 B]
Get:28 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/universe Translation-en [3292 B]
Get:29 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [580 B]
Get:30 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:31 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [985 kB]
Get:32 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [183 kB]
Get:33 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [8844 B]
Get:34 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [526 kB]
Get:35 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [75.4 kB]
Get:36 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 c-n-f Metadata [528 B]
Get:37 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [650 kB]

ubuntu@ip-172-31-34-49:~
```

```
Fetched 20.6 MB in 4s (5641 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-34-49:~$ sudo apt install openjdk-8-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
adwaita-icon-theme at-spi2-core ca-certificates-java fontconfig fontconfig-config fonts-dejavu-core fonts-dejavu-extra gtk-update-icon-cache
hicolor-icon-theme humanity-icon-theme java-common libasynccns0 libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data
libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcups2 libdatrie1 libdrm-amdgpu libdrm-intel1
libdrm-nouveau2 libdrm-radeon1 libflac8 libfontconfig1 libfontconfig1 libgail-common libgail18 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin
libgdk-pixbuf2.0-common libgif7 libgl1 libgl1-mesa-dri libgl1-mesa-glx libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3 libgtk2.0-0
libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice-dev libice6 libjbig0 libjpeg-turbo8 liblcms2-2 libl1vml2 libpango-1.0-0 libpangocairo-1.0-0
libpangoft2-1.0-0 libpiciaccess0 libpcssclite1 libpixman-1-0 libpthread-stubs0-dev libpulse0 librsvg2-2 librsvg2-common libsensors-config libsensors5
libsm-dev libsm6 libsndfile1 libthai-data libthai0 libtiff5 libvorbisenc2 libvulkan1 libwayland-client0 libwebp6 libxi1-dev libxi1-xcb1 libxau-dev libxaw7
libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcb1-dev
libcomposite1 libxcursor1 libxdamage1 libxdmcp-dev libxfices3 libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1
libxshmfence1 libxt-dev libxt6 libxtst6 libxv1 libxf86dg1 libxf86vm1 mesa-vulkan-drivers openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless
ubuntu-mono x11-common x11-utils x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
default-jre cups-common gvfs libice-doc liblcms2-utils pcscd pulseaudio librsvg2-bin lm-sensors libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-8-demo
openjdk-8-source visualvm icedtea-8-plugin libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei fonts-indic
mesa-utils
The following NEW packages will be installed:
adwaita-icon-theme at-spi2-core ca-certificates-java fontconfig fontconfig-config fonts-dejavu-core fonts-dejavu-extra gtk-update-icon-cache
hicolor-icon-theme humanity-icon-theme java-common libasynccns0 libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data
libatspi2.0-0 libavahi-client3 libavahi-common3 libcairo-gobject2 libcairo2 libcups2 libdatrie1 libdrm-amdgpu libdrm-intel1
libdrm-nouveau2 libdrm-radeon1 libflac8 libfontconfig1 libfontconfig1 libgail-common libgail18 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin
libgdk-pixbuf2.0-common libgif7 libgl1 libgl1-mesa-dri libgl1-mesa-glx libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3 libgtk2.0-0
libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice-dev libice6 libjbig0 libjpeg-turbo8 liblcms2-2 libl1vml2 libpango-1.0-0 libpangocairo-1.0-0
libpangoft2-1.0-0 libpiciaccess0 libpcssclite1 libpixman-1-0 libpthread-stubs0-dev libpulse0 librsvg2-2 librsvg2-common libsensors-config libsensors5
libsm-dev libsm6 libsndfile1 libthai-data libthai0 libtiff5 libvorbisenc2 libvulkan1 libwayland-client0 libwebp6 libxi1-dev libxi1-xcb1 libxau-dev libxaw7
libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcb1-dev
libcomposite1 libxcursor1 libxdamage1 libxdmcp-dev libxfices3 libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1
libxshmfence1 libxt-dev libxt6 libxtst6 libxv1 libxf86dg1 libxf86vm1 mesa-vulkan-drivers openjdk-8-jdk openjdk-8-jdk-headless openjdk-8-jre
openjdk-8-jre-headless ubuntu-mono x11-common x11-utils x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 126 newly installed, 0 to remove and 18 not upgraded.
Need to get 93.5 MB of archives.
```

```
ubuntu@ip-172-31-34-49: ~
ubuntu@ip-172-31-34-49:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan
Suggested packages:
ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 18 not upgraded.
Need to get 74.5 MB of archives.
After this operation, 361 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1.6-2ubuntu1 [30.5 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1.0.1-0ubuntu2~20.04.1 [4155 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 containerd amd64 1.5.5-0ubuntu3~20.04.1 [33.0 MB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 dns-root-data all 2019052802 [5300 B]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libidn11 amd64 1.33-2.2ubuntu2 [46.2 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 dnsmasq-base amd64 2.80-1.1ubuntu1.4 [315 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 docker.io amd64 20.10.7-0ubuntu5~20.04.2 [36.9 MB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 ubuntu-fan all 0.12.13 [34.5 kB]
Fetched 74.5 MB in 1s (50.9 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 79377 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.6-2ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.6-2ubuntu1) ...
Selecting previously unselected package runc.

```

```
ubuntu@ip-172-31-34-49: ~
ubuntu@ip-172-31-34-49:~$ sudo docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu5~20.04.2
ubuntu@ip-172-31-34-49:~$
```

Step 4.2: Jenkins-slave2

```
ubuntu@ip-172-31-41-33:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1341 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [275 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [14.4 kB]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [569 kB]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [81.6 kB]
Get:16 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f Metadata [528 B]
Get:17 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [871 kB]
Get:18 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [188 kB]
Get:19 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [19.5 kB]
Get:20 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [24.6 kB]
Get:21 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [6856 B]
Get:22 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [616 B]
Get:23 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [2568 B]
Get:24 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/main Translation-en [1120 B]
Get:25 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [400 B]
Get:26 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [6584 B]
Get:28 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/universe Translation-en [3292 B]
Get:29 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [580 B]
Get:30 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:31 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [985 kB]
Get:32 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [183 kB]
Get:33 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [8844 B]
Get:34 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [526 kB]
Get:35 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [75.4 kB]
Get:36 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 c-n-f Metadata [528 B]
Get:37 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [650 kB]
ubuntu@ip-172-31-41-33:~$ 17:52
10-11-2021 1
```

```
ubuntu@ip-172-31-41-33:~$ sudo apt install openjdk-8-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
adwaita-icon-theme at-spi2-core ca-certificates-java fontconfig fontconfig-config fonts-dejavu-core fonts-dejavu-extra gtk-update-icon-cache
hicolor-icon-theme humanity-icon-theme java-common libasynccns0 libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data
libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcupsc2 libdatriel1 libdrm-amdgpu1 libdrm-intel1
libdrm-nouveau2 libdrm-radeon1 libflac8 libfontconfig1 libfontenc1 libgail-common libgail18 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin
libgdk-pixbuf2.0-common libgif7 libgl1 libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa libglx0 libgraphite2-3 libgtk2.0-0
libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice6 libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2 libl1vnd12 libpango-1.0-0 libpangocairo-1.0-0
libpangoft2-1.0-0 libpcaccess0 libpcsslite1 libpixman-1-0 libpthread-stubs0-dev libpulse0 librsvg2-2 librsvg2-common libssensors-config libssensors5
libsm-dev libsm6 libsndfile1 libthai-data libtiff5 libvorbisenc2 libvulkan1 libwayland-client0 libwebp0 libxi1-dev libxi1-xcb1 libxau-dev libxaw7
libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcb1-dev
libcomposite1 libxcursor1 libxdamage1 libxdmcp-dev libxfixed3 libxft2 libxi6 libxinerama1 libxbfile1 libxmu6 libxpm4 libxrandr2 libxrender1
libxshmfence1 libxt-dev libxt6 libxtst6 libxv1 libxxf86vm1 mesa-vulkan-drivers openjdk-8-jdk-headless openjdk-8-jre-headless
ubuntu-mono x11-common x11-utils x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
default-jre cups-common gvfs libice-doc liblcms2-utils pcscd pulseaudio librsvg2-bin lm-sensors libsm-doc libxi1-doc libxcb-doc libxt-doc openjdk-8-demo
openjdk-8-source visualvm icedtea-8-plugin libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei fonts-indic
mesa-utils
The following NEW packages will be installed:
adwaita-icon-theme at-spi2-core ca-certificates-java fontconfig fontconfig-config fonts-dejavu-core fonts-dejavu-extra gtk-update-icon-cache
hicolor-icon-theme humanity-icon-theme java-common libasynccns0 libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data
libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcupsc2 libdatriel1 libdrm-amdgpu1 libdrm-intel1
libdrm-nouveau2 libdrm-radeon1 libflac8 libfontconfig1 libfontenc1 libgail-common libgail18 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin
libgdk-pixbuf2.0-common libgif7 libgl1 libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa libglx0 libgraphite2-3 libgtk2.0-0
libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice6 libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2 libl1vnd12 libpango-1.0-0 libpangocairo-1.0-0
libpangoft2-1.0-0 libpcaccess0 libpcsslite1 libpixman-1-0 libpthread-stubs0-dev libpulse0 librsvg2-2 librsvg2-common libssensors-config libssensors5
libsm-dev libsm6 libsndfile1 libthai-data libtiff5 libvorbisenc2 libvulkan1 libwayland-client0 libwebp0 libxi1-dev libxi1-xcb1 libxau-dev libxaw7
libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcb1-dev
libcomposite1 libxcursor1 libxdamage1 libxdmcp-dev libxfixed3 libxft2 libxi6 libxinerama1 libxbfile1 libxmu6 libxpm4 libxrandr2 libxrender1
libxshmfence1 libxt-dev libxt6 libxtst6 libxv1 libxxf86vm1 mesa-vulkan-drivers openjdk-8-jdk openjdk-8-jdk-headless openjdk-8-jre
openjdk-8-jre-headless ubuntu-mono x11-common x11-utils x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 126 newly installed, 0 to remove and 18 not upgraded.
Need to get 93.5 MB of archives.
After this operation, 690 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
ubuntu@ip-172-31-41-33:~$ 17:53
10-11-2021 1
```

```
ubuntu@ip-172-31-41-33:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 18 not upgraded.
Need to get 74.5 MB of archives.
After this operation, 361 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1.6-2ubuntu1 [30.5 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1.0.1-0ubuntu2~20.04.1 [4155 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 containerd amd64 1.5.5-0ubuntu3~20.04.1 [33.0 MB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 dns-root-data all 2019052802 [5300 B]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libidn11 amd64 1.33-2.2ubuntu2 [46.2 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 dnsmasq-base amd64 2.80-1.1ubuntu1.4 [315 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 docker.io amd64 20.10.7-0ubuntu5~20.04.2 [36.9 MB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 ubuntu-fan all 0.12.13 [34.5 kB]
Fetched 74.5 MB in 2s (47.2 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 79377 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.6-2ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.6-2ubuntu1) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.0.1-0ubuntu2~20.04.1_amd64.deb ...

```

```
ubuntu@ip-172-31-41-33:~$ sudo docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu5~20.04.2
ubuntu@ip-172-31-41-33:~$
```

Step 5: Connect Agent to Jenkins using JNLP connection

Step 5.1: Copy the command specified below and run from a command line to successfully connect the agent to Jenkins.

The screenshot shows the Jenkins web interface for a slave node named 'jenkins-slave1'. On the left, there's a sidebar with links like 'Back to List', 'Status', 'Delete Agent', 'Configure', 'Build History', 'Load Statistics', and 'Log'. The main content area is titled 'Agent jenkins-slave1' and contains instructions for connecting the agent. It shows two methods: launching from a browser or running from the command line. Below these instructions, a section titled 'Projects tied to jenkins-slave1' lists 'None'. At the bottom right, there are links for 'REST API' and 'Jenkins 2.303.3'.

```
ubuntu@ip-172-31-34-49:~$ java -jar agent.jar -jnlpUrl http://18.218.1.208:8080/computer/jenkins-slave1/jenkins-agent.jnlp -secret c145f7677df78daedcdfac49cc1a88b63f66bbdc30e02ea00fefa1d20c7d19b
Nov 10, 2021 12:26:15 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/Team1/jenkins/remoting as a remoting work directory
Nov 10, 2021 12:26:15 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ubuntu/Team1/jenkins/remoting
Nov 10, 2021 12:26:15 PM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: jenkins-slave1
Nov 10, 2021 12:26:15 PM hudson.remoting.jnlp.Main$CuiListener <init>
INFO: Jenkins agent is running in headless mode.
Nov 10, 2021 12:26:15 PM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 4.10.1
Nov 10, 2021 12:26:15 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/Team1/jenkins/remoting as a remoting work directory
Nov 10, 2021 12:26:16 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Locating server among [http://18.218.1.208:8080/]
Nov 10, 2021 12:26:16 PM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Nov 10, 2021 12:26:16 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
  Agent address: 18.218.1.208
  Agent port: 40847
  Identity: 66:e6:b3:59:d0:18:7e:98:63:2e:53:16:9c:23:d0:79
Nov 10, 2021 12:26:16 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Nov 10, 2021 12:26:16 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to 18.218.1.208:40847
Nov 10, 2021 12:26:16 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLP4-connect
Nov 10, 2021 12:26:16 PM org.jenkinsci.remoting.protocol.impl.BIONetworkLayer$Reader run
INFO: Waiting for ProtocolStack to start.
Nov 10, 2021 12:26:16 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: 66:e6:b3:59:d0:18:7e:98:63:2e:53:16:9c:23:d0:79
Nov 10, 2021 12:26:17 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
```

The screenshot shows the Jenkins slave configuration page for 'jenkins-slave1'. The left sidebar contains links for Back to List, Status, Delete Agent, Configure, Build History, Load Statistics, Script Console, Log, System Information, and Disconnect. The main content area shows the title 'Agent jenkins-slave1' with a note 'Agent is connected.' Below it is a section titled 'Projects tied to jenkins-slave1' with the message 'None'. At the top right is a button 'Mark this node temporarily offline'. The status bar at the bottom shows system icons and the date/time as 10-11-2021 17:56.

Step 5.2: Repeat the same steps for 'jenkins-slave2'

```

ubuntu@ip-172-31-41-33:~$ java -jar agent.jar -jnlpUrl http://18.218.1.208:8080/computer/jenkins-slave2/jenkins-agent.jnlp -secret bab2d70e370c4e5b94327dc343473cd4996a45806d3a332619c0f661a6702cb2 -workDir "/home/ubuntu/Team1/jenkins"
Nov 10, 2021 12:24:54 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/Team1/jenkins/remoting as a remoting work directory
Nov 10, 2021 12:24:54 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ubuntu/Team1/jenkins/remoting
Nov 10, 2021 12:24:54 PM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: jenkins-slave2
Nov 10, 2021 12:24:54 PM hudson.remoting.jnlp.Main$CuiListener <init>
INFO: Jenkins agent is running in headless mode.
Nov 10, 2021 12:24:54 PM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 4.10.1
Nov 10, 2021 12:24:54 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/Team1/jenkins/remoting as a remoting work directory
Nov 10, 2021 12:24:55 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Locating server among [http://18.218.1.208:8080/]
Nov 10, 2021 12:24:55 PM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Nov 10, 2021 12:24:55 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
  Agent address: 18.218.1.208
  Agent port: 40847
  Identity: 66:e6:b3:59:d0:18:7e:98:63:2e:53:16:9c:23:d0:79
Nov 10, 2021 12:24:55 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Nov 10, 2021 12:24:55 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to 18.218.1.208:40847
Nov 10, 2021 12:24:55 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLP4-connect
Nov 10, 2021 12:24:55 PM org.jenkinsci.remoting.protocol.impl.BIONetworkLayer$Reader run
INFO: Waiting for ProtocolStack to start.
Nov 10, 2021 12:24:55 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: 66:e6:b3:59:d0:18:7e:98:63:2e:53:16:9c:23:d0:79
Nov 10, 2021 12:24:56 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected

```

The screenshot shows the Jenkins interface for the slave2 node. The left sidebar has links for Back to List, Status, Delete Agent, Configure, Build History, Load Statistics, Script Console, Log, System Information, and Disconnect. The main content area is titled "Agent jenkins-slave2" and says "Agent is connected." It lists "Projects tied to jenkins-slave2" as "None". A "Build Executor Status" section shows "1 Idle". Below the main content is a toolbar with "agent (1).jar" and "agent.jar" buttons, and a "Show all" link. The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating "ENG IN" and the date "10-11-2021".

Step 5.3: Once they are successfully connected, nodes appear as shown below in the dashboard.

The screenshot shows the Jenkins dashboard with the "Nodes" link selected. The left sidebar includes Back to Dashboard, Manage Jenkins, New Node, Configure Clouds, and Node Monitoring. The main area displays a table of nodes:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	jenkins-slave1	Linux (amd64)	In sync	5.74 GB	- 0 B	5.74 GB	92ms
2	jenkins-slave2	Linux (amd64)	In sync	5.74 GB	- 0 B	5.74 GB	46ms
3	master	Linux (amd64)	In sync	5.31 GB	- 0 B	5.31 GB	0ms

Below the table, it says "Data obtained 1 min 8 sec". A "Refresh status" button is at the bottom right. The bottom of the screen shows a Windows taskbar with "agent (1).jar" and "agent.jar" buttons, and a "Show all" link. The bottom right corner shows "17:57" and the date "10-11-2021".

Step 6: Create and build projects

Step 6.1: Fork the GitHub repository: <https://github.com/hshar/devopsIQ>

The screenshot shows a GitHub repository page for 'parvatijay2901 / devopsIQ'. The repository has 1 branch and 0 tags. The 'Code' tab is selected. A message indicates that this branch is 3 commits ahead of hshar:master. The commit history shows:

- parvatijay2901 Add files via upload (991c148, 13 hours ago) - 41 commits
- devopsIQ Add files via upload (13 hours ago)
- Dockerfile added docker file (3 years ago)
- azure-pipelines.yml Set up CI with Azure Pipelines (2 years ago)
- docker-compose add (3 years ago)

Below the commit history, there's a note to 'Add a README' and a link to 'Help people interested in this repository understand your project by adding a README.' The right sidebar contains sections for 'About', 'Releases', 'Packages', and 'Languages'. The 'Languages' section shows HTML (63.6%) and Dockerfile (36.4%).

Step 6.2: Create a project 'Test1' that can only be ran on 'jenkins-slave1'

The screenshot shows the Jenkins dashboard with a new item being created. The item name is 'Test1'. The 'Freestyle project' option is selected, with a description: '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.' Other options shown include 'Pipeline', 'Multi-configuration project', 'Folder', and 'Multibranch Pipeline'. The 'OK' button is visible at the bottom of the dialog.

Test1 Config [Jenkins] Not secure | 18.218.1.208:8080/job/Test1/configure

Dashboard > Test1 >

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

[Plain text] Preview

Discard old builds GitHub project Project url
https://github.com/parvatijay2901/devopsIQ/ Advanced...

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
jenkins-slave1 Advanced...

Label jenkins-slave1 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Save **Apply**

This screenshot shows the 'General' configuration page for a Jenkins job named 'Test1'. The 'GitHub project' checkbox is checked, and the 'Project url' field contains 'https://github.com/parvatijay2901/devopsIQ/'. Under the 'Restrict where this project can be run' section, the 'Label Expression' field is set to 'jenkins-slave1'. The 'Save' and 'Apply' buttons are at the bottom.

Test1 Config [Jenkins] Not secure | 18.218.1.208:8080/job/Test1/configure

Dashboard > Test1 >

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

Source Code Management

None Git Advanced...

Repositories

Repository URL
https://github.com/parvatijay2901/devopsIQ/ Advanced... Add Repository

Credentials
- none - Add Advanced...

Branches to build

Branch Specifier (blank for 'any')
*/master Advanced... Add Branch

Save **Apply**

This screenshot shows the 'Source Code Management' configuration page for the 'Test1' job. The 'Git' radio button is selected. A single repository is configured with the URL 'https://github.com/parvatijay2901/devopsIQ/'. The 'Branches to build' section specifies the 'Branch Specifier' as '*/master'. The 'Save' and 'Apply' buttons are at the bottom.

Step 6.3: Build the project and view the console output to check the success status. By doing so, the repository will be cloned on 'jenkins-slave1'

The screenshot shows the Jenkins interface with a successful build output. The build was started by Parvati Jayakumar and ran as SYSTEM. It was built on jenkins-slave1 in workspace /home/ubuntu/workspace/Test1. The git tool used was NONE. The repository cloned was https://github.com/hshar/devopsIQ. The build log shows the git command sequence for cloning, fetching, and checking out the code, followed by a message indicating it's the first time build and skipping the changelog, resulting in a SUCCESS status.

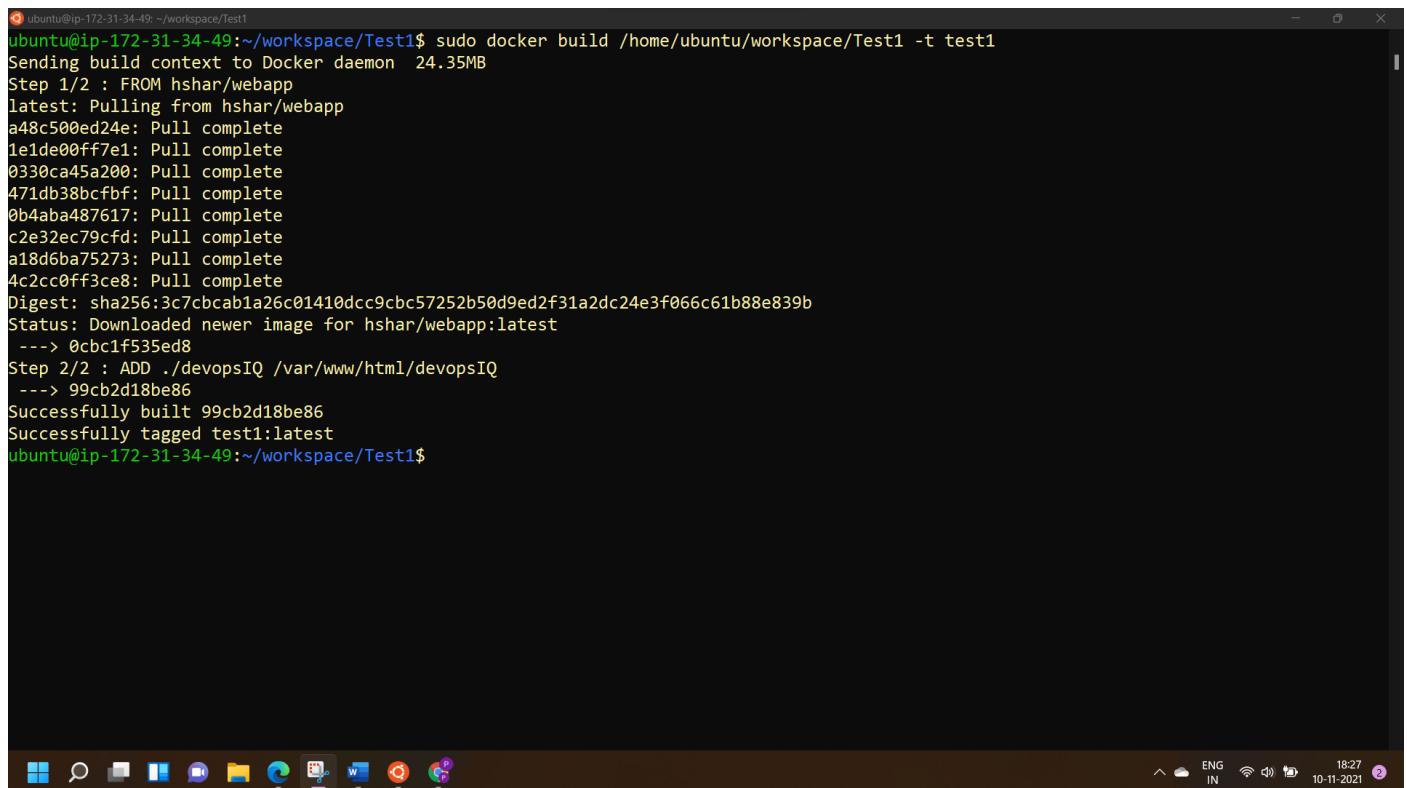
```
Started by user Parvati Jayakumar
Running as SYSTEM
Building remotely on jenkins-slave1 in workspace /home/ubuntu/workspace/Test1
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/hshar/devopsIQ
> git init /home/ubuntu/workspace/Test1 # timeout=10
Fetching upstream changes from https://github.com/hshar/devopsIQ
> git --version # timeout=10
> git --version # 'git' version 2.25.1'
> git fetch --tags --force --progress -- https://github.com/hshar/devopsIQ +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/hshar/devopsIQ # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision cc26380b7f45bfcf31831fc4f4f6e438fd6e2ab2 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f cc26380b7f45bfcf31831fc4f4f6e438fd6e2ab2 # timeout=10
Commit message: "Set up CI with Azure Pipelines"
First time build. Skipping changelog.
Finished: SUCCESS
```

Step 6.4: Make sure that the repository was cloned properly.

The screenshot shows a terminal window on an Ubuntu system (ip-172-31-34-49) displaying the directory structure of the workspace. The user navigates through the workspace, workspace/Test1, and workspace/Test1/Test1 directories, showing files like Team1, agent.jar, Dockerfile, azure-pipelines.yml, devopsIQ, docker-compose, and Test1. The pwd command shows the current working directory is /home/ubuntu/workspace/Test1.

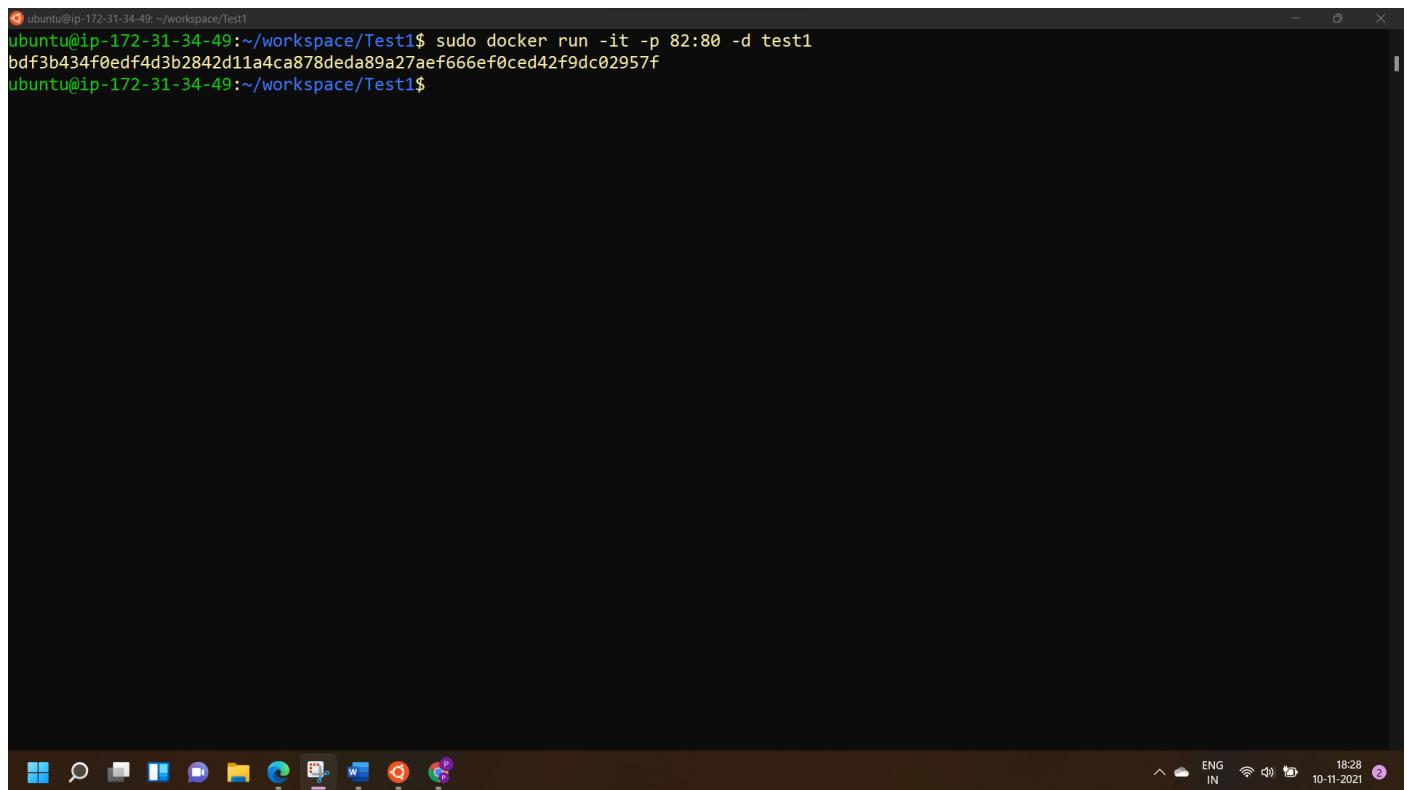
```
ubuntu@ip-172-31-34-49:~/workspace$ ls
Team1 agent.jar workspace
ubuntu@ip-172-31-34-49:~/workspace$ cd workspace
ubuntu@ip-172-31-34-49:~/workspace$ ls
Test1
ubuntu@ip-172-31-34-49:~/workspace$ cd Test1
ubuntu@ip-172-31-34-49:~/workspace/Test1$ ls
Dockerfile azure-pipelines.yml devopsIQ docker-compose
ubuntu@ip-172-31-34-49:~/workspace/Test1$ pwd
/home/ubuntu/workspace/Test1
ubuntu@ip-172-31-34-49:~/workspace/Test1$
```

Step 6.5: The repository had files with which we can deploy a website on Docker. First, we will check if the project is running successfully by running on the terminal. For that, execute the ‘build’ command.



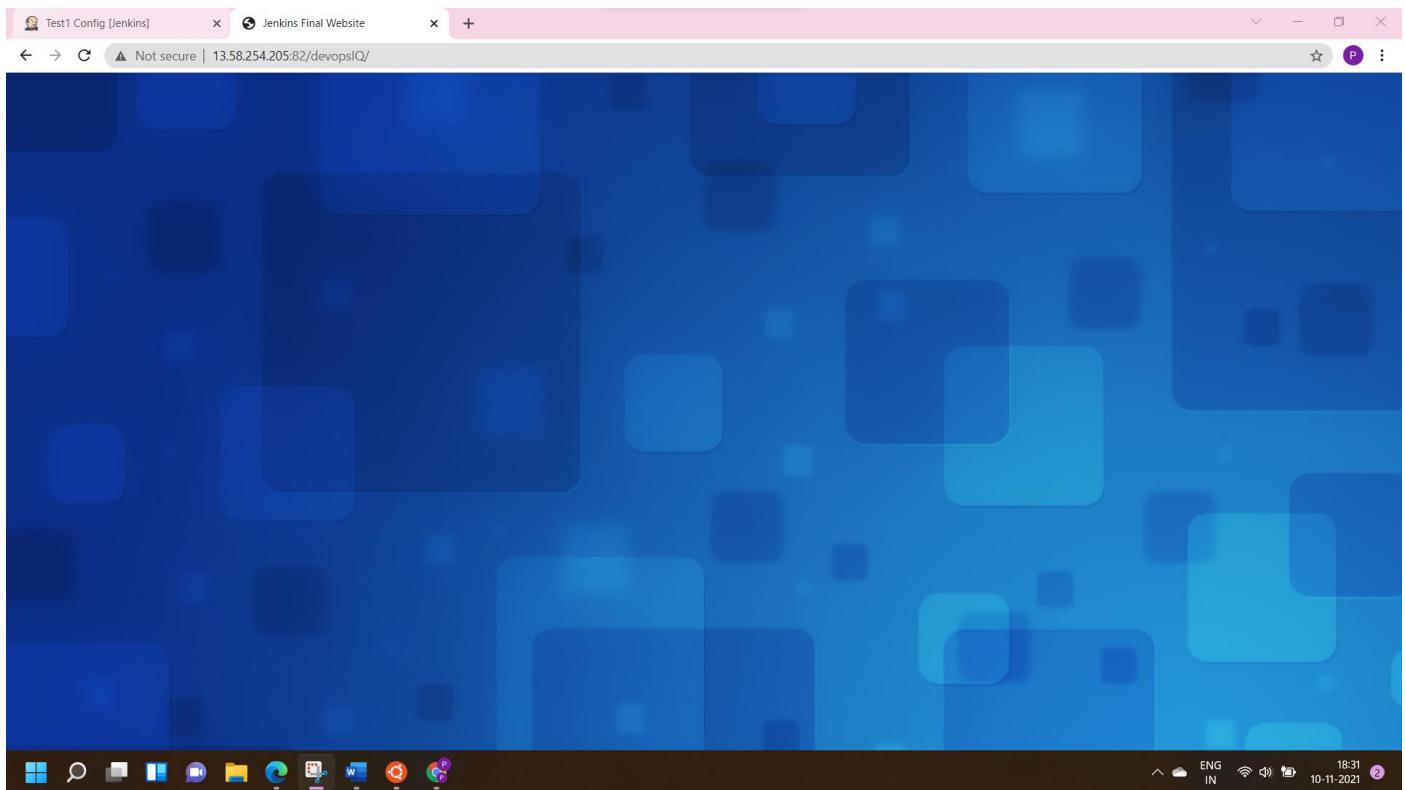
```
ubuntu@ip-172-31-34-49:~/workspace/Test1$ sudo docker build /home/ubuntu/workspace/Test1 -t test1
Sending build context to Docker daemon 24.35MB
Step 1/2 : FROM hshar/webapp
latest: Pulling from hshar/webapp
a48c500ed24e: Pull complete
1e1de00ff7e1: Pull complete
0330ca45a00: Pull complete
471db38bcfbf: Pull complete
0b4aba487617: Pull complete
c2e32ec79cf9: Pull complete
a18d6ba75273: Pull complete
4c2cc0ff3ce8: Pull complete
Digest: sha256:3c7cbcab1a26c01410dcc9cbc57252b50d9ed2f31a2dc24e3f066c61b88e839b
Status: Downloaded newer image for hshar/webapp:latest
--> 0cbc1f535ed8
Step 2/2 : ADD ./devopsIQ /var/www/html/devopsIQ
--> 99cb2d18be86
Successfully built 99cb2d18be86
Successfully tagged test1:latest
ubuntu@ip-172-31-34-49:~/workspace/Test1$
```

Step 6.6: Run the project on port-82



```
ubuntu@ip-172-31-34-49:~/workspace/Test1$ sudo docker run -it -p 82:80 -d test1
bdf3b434f0edf4d3b2842d11a4ca878deda89a27aef666ef0ced42f9dc02957f
ubuntu@ip-172-31-34-49:~/workspace/Test1$
```

Step 6.7: Verify that the website was successfully deployed.



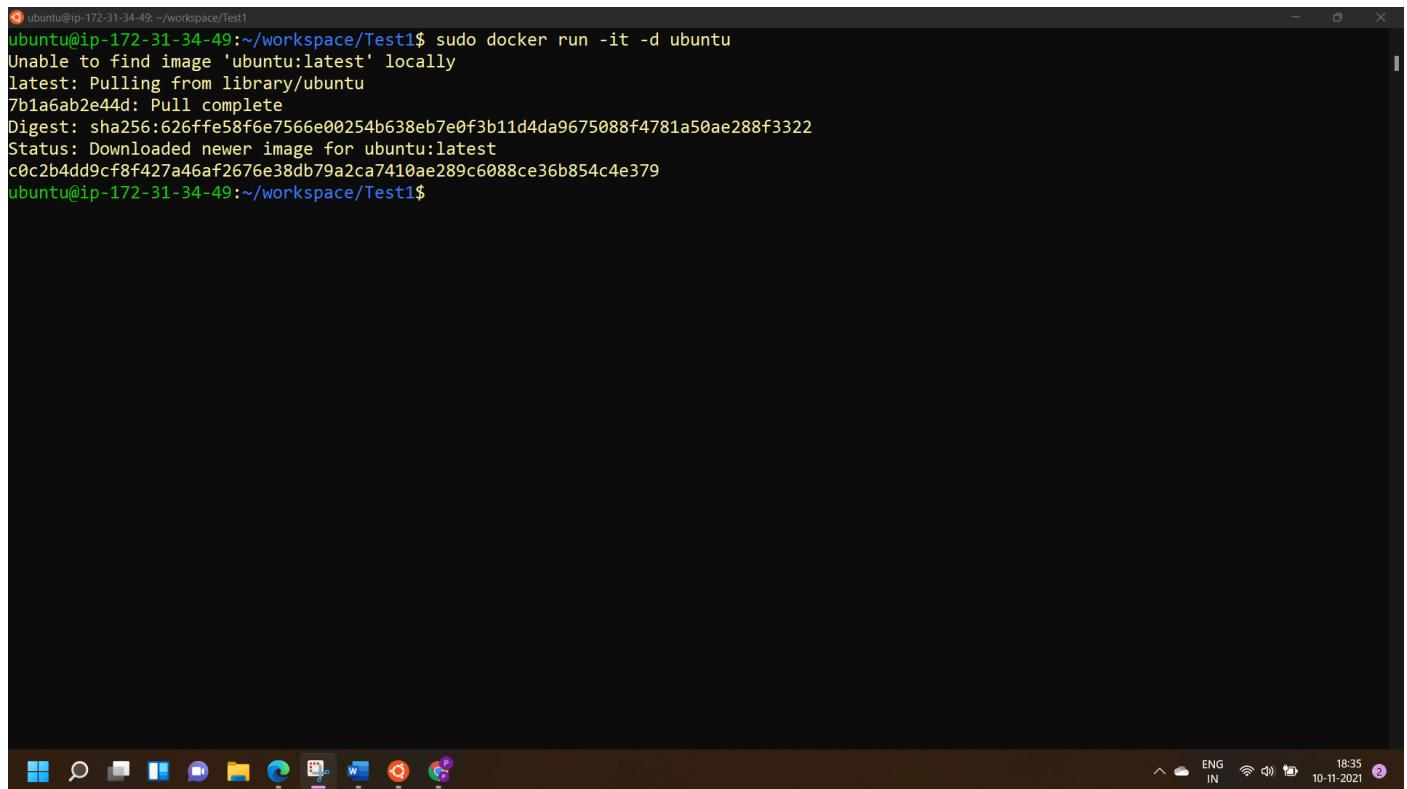
Step 6.8: Now, let us perform these actions using Jenkins. Give these commands in Test1's Build-> Execute shell option. Give port as 82.

A screenshot of the Jenkins configuration interface for a job named "Test1". The top navigation bar shows "Dashboard > Test1 >". Below this, there are tabs for "General", "Source Code Management", "Build Triggers", "Build Environment" (which is currently selected), "Build", and "Post-build Actions". In the "Build" section, there is a "Execute shell" step. The "Command" field contains the following Jenkinsfile code:

```
sudo docker rm -f $(sudo docker ps -a -q)
sudo docker build /home/ubuntu/workspace/Test1 -t test1
sudo docker run -it -p 82:80 -d test1
```

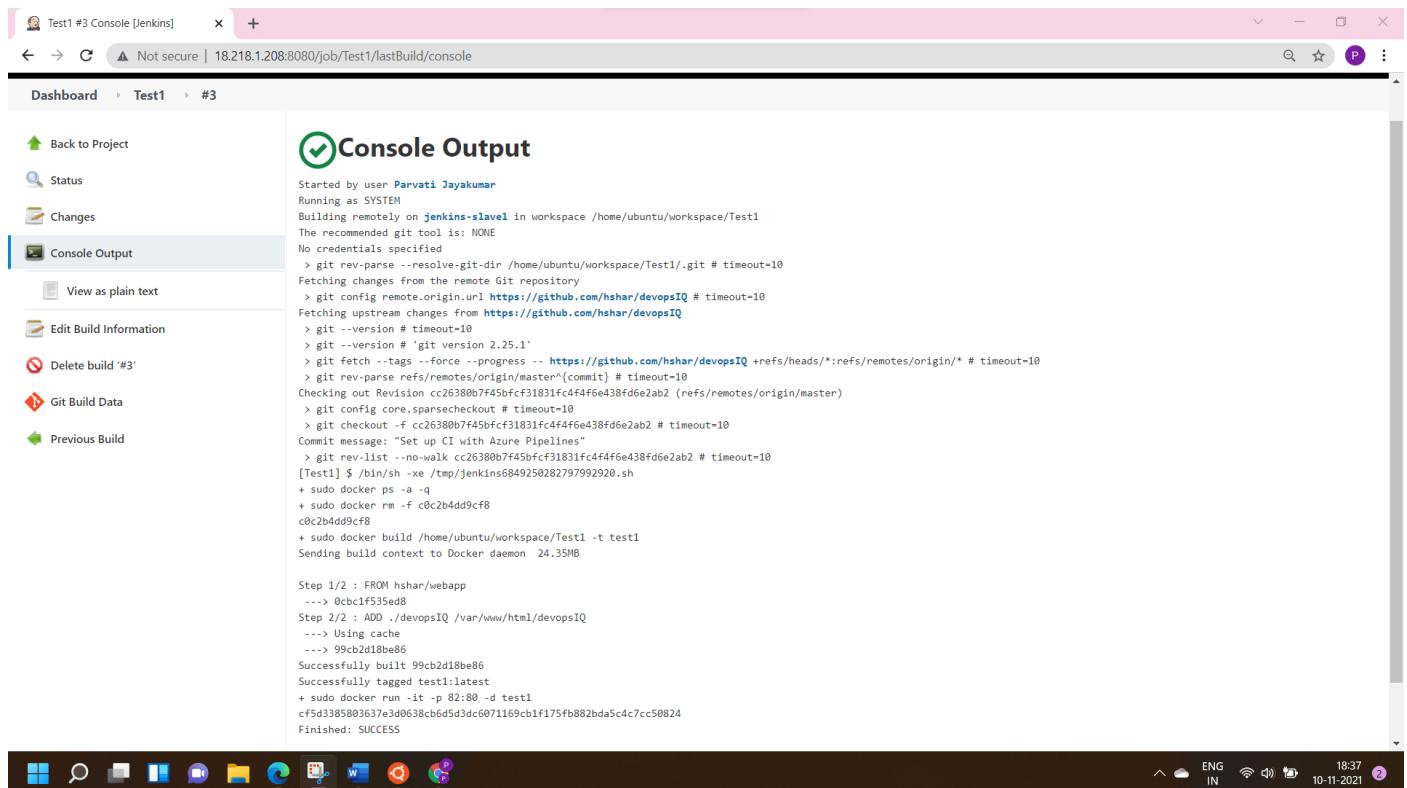
The "Post-build Actions" section is partially visible below, showing a "Save" button. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time as 10-11-2021 18:25.

Step 6.9: Atleast one docker container will be present on the system always and we have to delete the existing containers before building this project. Now, since I don't have any container running on my system, I will create one before building.



```
ubuntu@ip-172-31-34-49: ~/workspace/Test1$ sudo docker run -it -d ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
7b1a6ab2e44d: Pull complete
Digest: sha256:626ffe58f6e7566e00254b638eb7e0f3b11d4da9675088f4781a50ae288f3322
Status: Downloaded newer image for ubuntu:latest
c0c2b4dd9cf8f427a46af2676e38db79a2ca7410ae289c6088ce36b854c4e379
ubuntu@ip-172-31-34-49: ~/workspace/Test1$
```

Step 6.10: Build the project and verify the results as we did manually.



Test1 #3 Console [Jenkins] + P

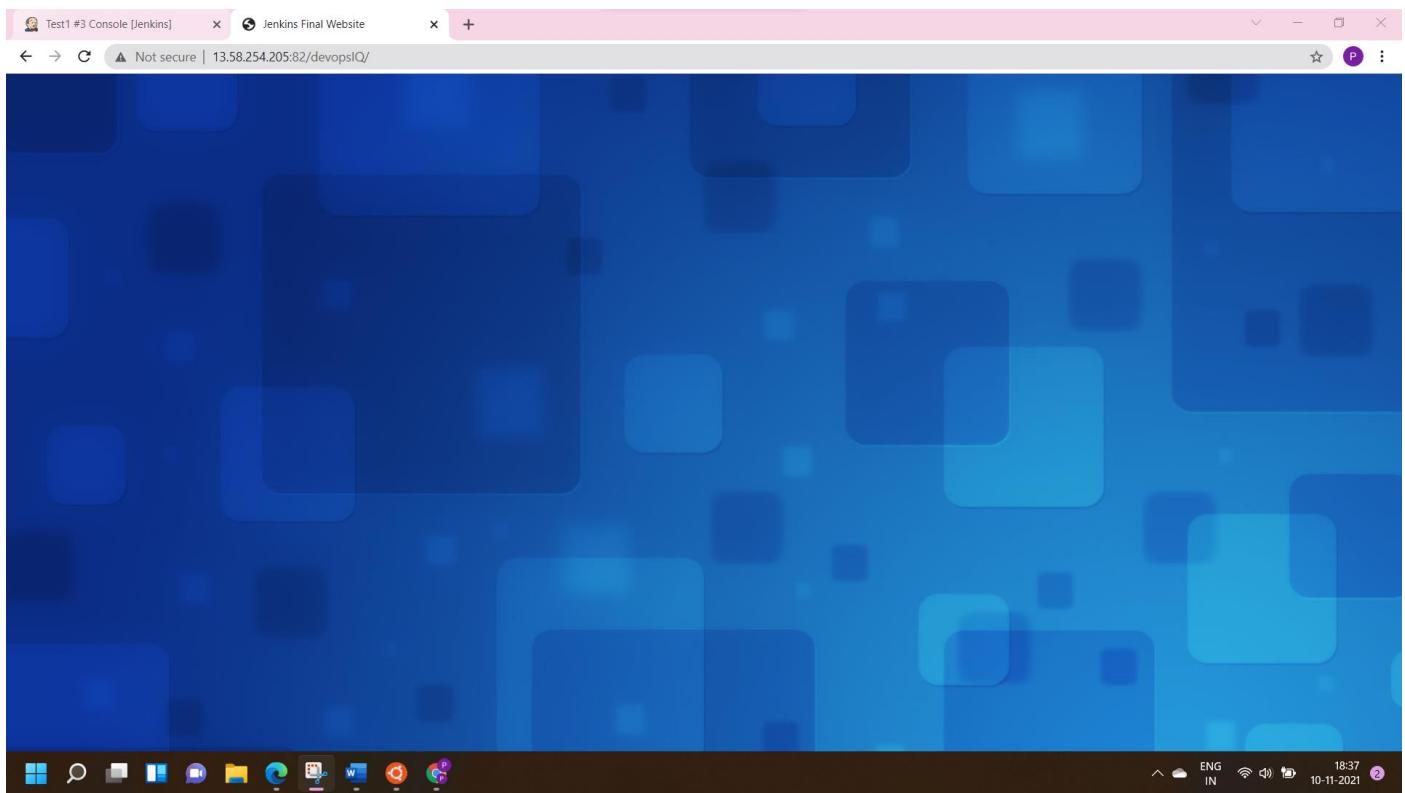
Not secure | 18.218.1.208:8080/job/Test1/lastBuild/console

Dashboard > Test1 > #3

✖ **Console Output**

Started by user **Parvati Jayakumar**
Running as SYSTEM
Building remotely on **jenkins-slave1** in workspace `/home/ubuntu/workspace/Test1`
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/ubuntu/workspace/Test1/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url <https://github.com/hshar/devopsIQ> # timeout=10
Fetching upstream changes from <https://github.com/hshar/devopsIQ>
> git -version # timeout=10
> git version # 'git version 2.25.1'
> git fetch --tags --force --progress -- <https://github.com/hshar/devopsIQ> +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^(commit) # timeout=10
Checking out Revision cc26380b7f45bfef31831fc4f4f6e438fd6e2ab2 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f cc26380b7f45bfef31831fc4f4f6e438fd6e2ab2 # timeout=10
Commit message: "Set up CI with Azure Pipelines"
> git rev-list --no-walk cc26380b7f45bfef31831fc4f4f6e438fd6e2ab2 # timeout=10
[Test1] \$ /bin/sh -xe /tmp/jenkins6849250282797992920.sh
+ sudo docker ps -a -q
+ sudo docker rm -f c0c2b4dd9cf8
+ sudo docker build /home/ubuntu/workspace/Test1 -t test1
Sending build context to Docker daemon 24.35MB

Step 1/2 : FROM hshar/webapp
--> 0cbc1f535ed8
Step 2/2 : ADD ./devopsIQ /var/www/html/devopsIQ
--> Using cache
--> 99cb2d18be86
Successfully built 99cb2d18be86
Successfully tagged test1:latest
+ sudo docker run -it -p 82:80 -d test1
cf5d3385803637e3d0638cb6d5d3dc6071169cb1f175fb882bd5c4c7cc50824
Finished: SUCCESS



Step 6.11: Perform the same steps to create a 'Test2' that runs on 'jenkins-slave2'. Make sure to give a different port. Here, I am giving port: 80.

A screenshot of the Jenkins 'New Item' creation interface. A modal dialog box is open, prompting the user to 'Enter an item name'. The input field contains the text 'Test2'. Below the input field, there is a note: 'Required field'. A list of project types is displayed: 'Freestyle project', 'Pipeline', 'Multi-configuration project', 'Folder', 'Multibranch Pipeline', and 'Organization Folder'. Each item has a small icon to its left and a brief description below it. At the bottom of the modal, there is a blue 'OK' button. The background shows the Jenkins dashboard with a user profile for 'Parvati Jayakumar'. The taskbar at the bottom shows various icons, and the system tray indicates the date as 10-11-2021 and time as 18:40.

Test2 Config [Jenkins] Not secure | 18.218.1.208:8080/job/Test2/configure

Dashboard > Test2 >

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

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
 With Ant

Build

Execute shell

Command

```
sudo docker rm -f $(sudo docker ps -a -q)
sudo docker build ./home/ubuntu/team1/jenkins/workspace/Test2 -t test2
sudo docker run -it -p 80:80 -d test2
```

See the list of available environment variables

Advanced...

Add build step ▾

Post-build Actions

Add post-build action ▾

Save Apply

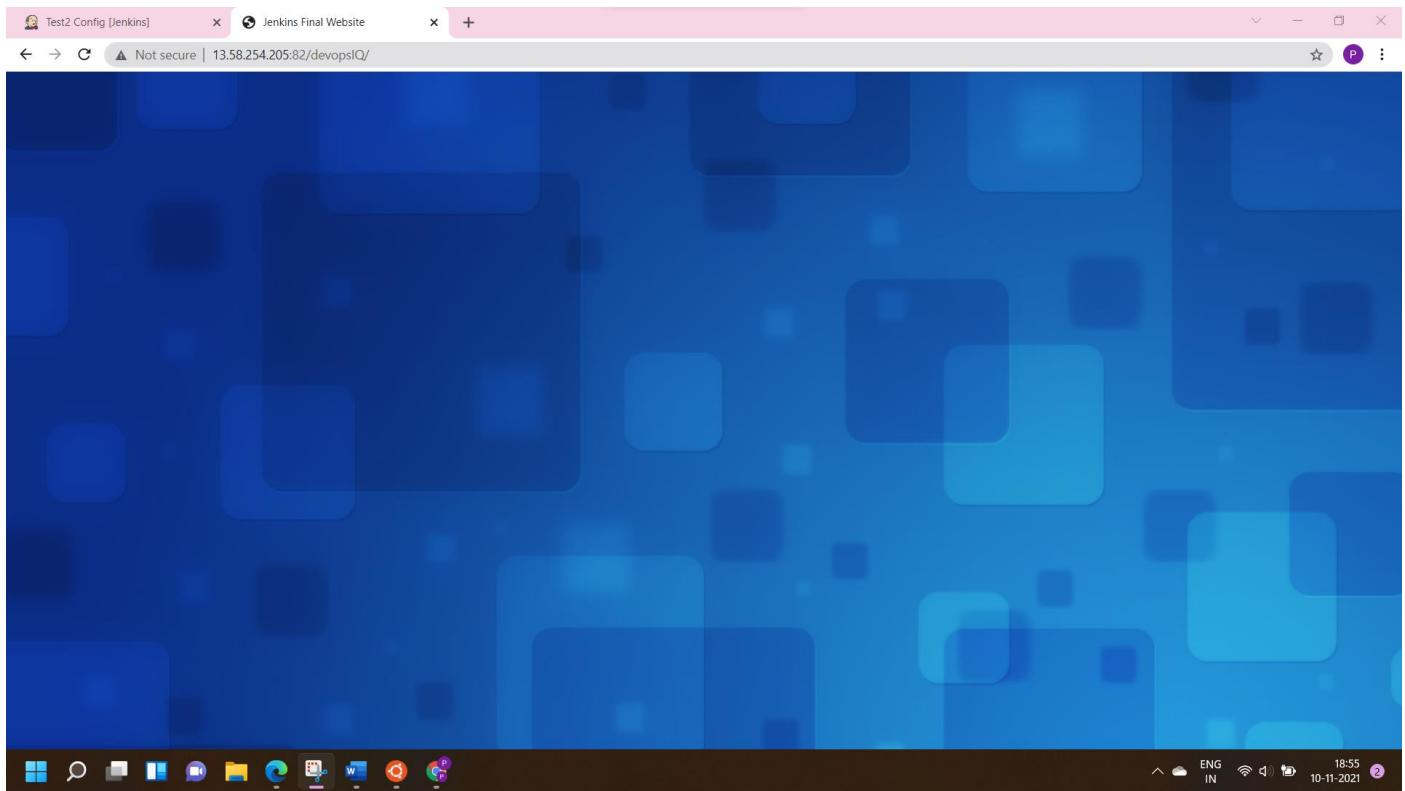
REST API Jenkins 2.305.3 18:54 10-11-2021

Test2 #4 Console [Jenkins] Not secure | 18.218.1.208:8080/job/Test2/4/console

Dashboard > Test2 > #4

```
4c2cc0ff3c8e: Waiting
0330ca45a200: Verifying Checksum
0330ca45a200: Download complete
1e1de00ff7e1: Verifying Checksum
1e1de00ff7e1: Download complete
471db38bcfcfb: Verifying Checksum
471db38bcfcfb: Download complete
0b4aba487617: Verifying Checksum
0b4aba487617: Download complete
a18d6ba75273: Verifying Checksum
a18d6ba75273: Download complete
a48c500ed24e: Verifying Checksum
a48c500ed24e: Download complete
4c2cc0ff3c8e: Verifying Checksum
4c2cc0ff3c8e: Download complete
c2e32ec79cf0: Verifying Checksum
c2e32ec79cf0: Download complete
a48c500ed24e: Pull complete
1e1de00ff7e1: Pull complete
0330ca45a200: Pull complete
471db38bcfcfb: Pull complete
0b4aba487617: Pull complete
c2e32ec79cf0: Pull complete
a18d6ba75273: Pull complete
4c2cc0ff3c8e: Pull complete
Digest: sha256:3c7cbcab1a26c01410dcc9cbc57252b50d9ed2f31a2dc24e3f066c61b88e839b
Status: Downloaded newer image for hshar/webapp:latest
-> 0cbc1f535ed8
Step 2/2 : ADD ./devops1Q /var/www/html/devops1Q
-> c54ef641a302
Successfully built c54ef641a302
Successfully tagged test2:latest
+ sudo docker run -it -p 80:80 -d test2
32fd619c2c26f3cb184adc7a22631cce25dca41176a704a587e108297b9b91a3
Finished: SUCCESS
```

REST API Jenkins 2.305.3 18:55 10-11-2021



Step 7: Configure Jenkins to build the project on Slave 1. If the step was successful, Test2 should be built on Slave 2.

Step 7.1: Install ‘Build pipeline plugin’

The screenshot shows the Jenkins Update Center interface. The main title is "Installing Plugins/Upgrades". On the left, there's a sidebar with links like "Back to Dashboard", "Manage Jenkins", and "Manage Plugins". The main content area lists various Jenkins components with their status: SSH server, Folders, OWASP Markup Formatter, Structs, Trilead API, Pipeline: Step API, Token Macro, Build Timeout, Credentials, Plain Credentials, SSH Credentials, Credentials Binding, SCM API, Pipeline: API, Timestamper, Caffeine API, Script Security, Plugin Utilities API, Font Awesome API, Popper.js API, and JClouds API. Each item has a green circle with a checkmark next to it, indicating success. At the bottom right, there's a system tray with icons for battery, signal, and time (18:58, 10-11-2021).

Step 7.2: Create a new view ‘CICD’ which shows the jobs in a build pipeline view.

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links like "New Item", "People", "Build History", "Manage Jenkins", "My Views", and "Lockable Resources". A "New View" button is highlighted with a blue border. The main content area shows a "View name" input field containing "CICD". Below it, there are three radio button options: "Build Pipeline View" (selected), "List View", and "My View". A note under "Build Pipeline View" says: "Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view." A note under "List View" says: "Shows items in a simple list format. You can choose which jobs are to be displayed in which view." A note under "My View" says: "This view automatically displays all the jobs that the current user has an access to." At the bottom of the dialog, there's an "OK" button. At the very bottom, there's a system tray with icons for battery, signal, and time (19:02, 10-11-2021).

Step 7.3: Specify the initial job – Test1

Name: CICD

Description:

[Plain text] Preview

Build Pipeline View Title: CICD

Build Queue: No builds in the queue.

Build Executor Status:

- master: 1 Idle, 2 Idle
- jenkins-slave1: 1 Idle
- jenkins-slave2: 1 Idle

Trigger Options: Select Initial Job: Test1

OK Apply

Step 7.4: On the dashboard click on ‘CICD’. Further, build the pipeline by pressing ‘Run’ option.

Pipeline #4

4 Test1
Nov 10, 2021 11:10:08 PM
21 sec
parvati2901

Test2

Build Pipeline: CICD

Run History Configure Add Step Delete Manage

REST API Jenkins 2.303.3



Step 7.5: Successfully built pipeline appears like this:

The screenshot shows the Jenkins Pipeline interface. At the top, there's a navigation bar with icons for Run, History, Configure, Add Step, Delete, and Manage. Below this, the main title is "Build Pipeline: CICD". There are two green cards representing build #5: "Test1" and "Test2". Both cards show a timestamp of "Nov 10, 2021 1:33:59 PM" and a duration of "21 sec". The Jenkins logo is visible in the top left corner.

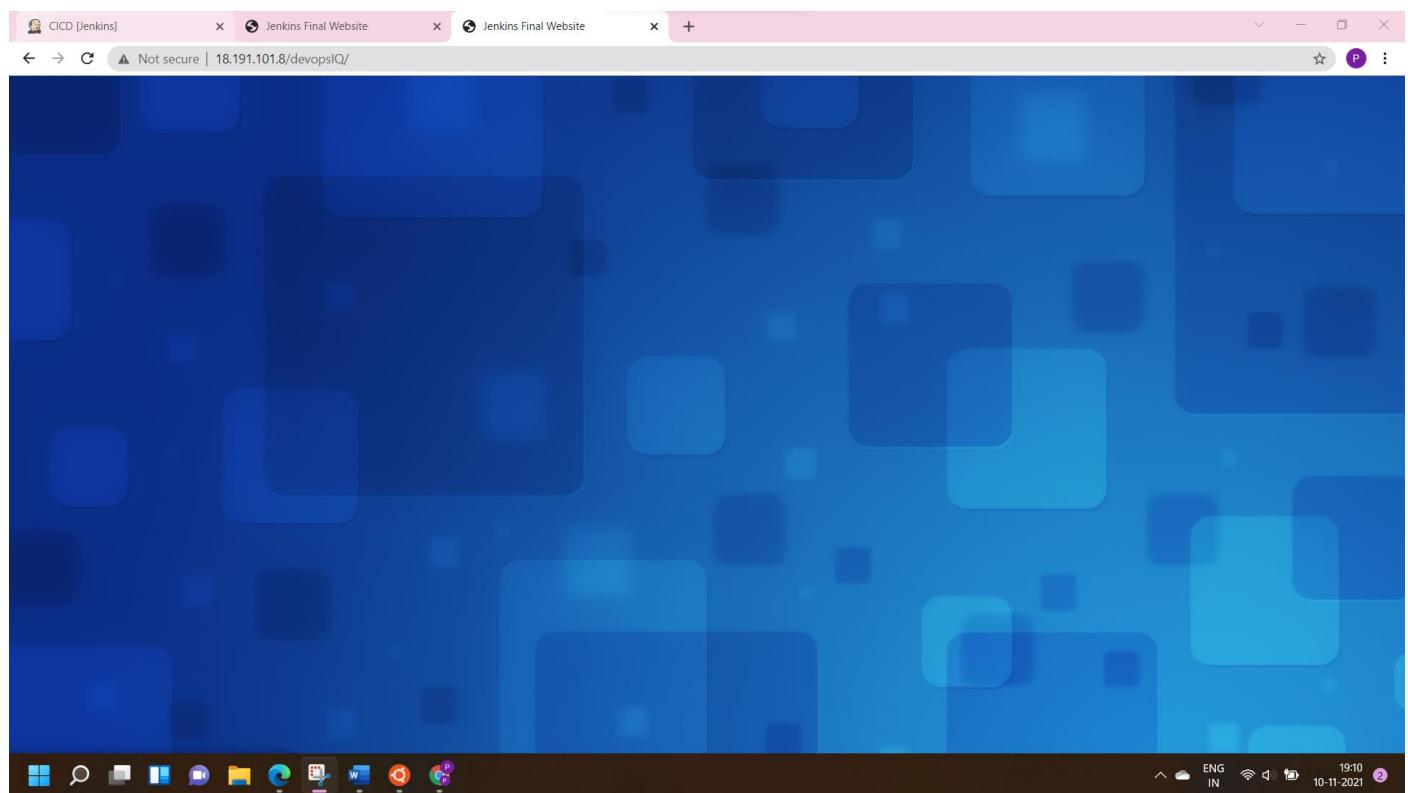


Step 7.6: Verify whether both the slaves deployed the website.

jenkins-slave1

This screenshot shows a Windows desktop with a solid blue background. The taskbar at the bottom has several pinned application icons. The system tray on the right shows "ENG IN", a battery icon, and a date/time stamp of "10-11-2021 19:05".

jenkins-slave2



Step 8: Trigger the job using git web-hooks.

Step 8.1: Select ‘GitHub gook trigger for GITScm polling’ option.

The screenshot shows the Jenkins configuration interface for the 'Test1' job. The top navigation bar includes links for 'Dashboard', 'Test1', and 'Configure'. The main content area has tabs for 'General', 'Source Code Management', 'Build Triggers' (which is selected), 'Build Environment', 'Build', and 'Post-build Actions'. Under 'Build Triggers', several options are listed: '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'. Each option has a help icon (question mark) next to it. The 'Build Environment' section contains 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 step titled 'Execute shell' with the command:

```
sudo docker rm -f $(sudo docker ps -a -q)  
sudo docker build /home/ubuntu/workspace/Test1 -t test1  
sudo docker run -it -p 82:80 -d test1
```

Buttons for 'Save' and 'Apply' are at the bottom of this step.

Step 8.2: Go to settings of the forked notebook -> Select Webhooks -> Add a new Webhook

The screenshot shows a GitHub repository settings page for 'parvatijay2901 / devopsIQ'. The left sidebar has a 'Webhooks' link selected. The main content area is titled 'Webhooks' and contains a single entry: 'http://18.218.1.208:8080/github-webhook'. There are 'Edit' and 'Delete' buttons next to it. A note below explains that webhooks notify external services about events like pushes.

parvatijay2901 / devopsIQ Public
forked from hshar/devopsIQ

Code Pull requests Actions Projects Wiki Security Insights Settings

Options
Manage access
Security & analysis
Branches
Webhooks
Notifications
Integrations
Deploy keys
Autolink references
Actions
Environments
Secrets
Pages

Watch 0 Star 0 Fork 211

Webhooks

Add webhook

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

✓ http://18.218.1.208:8080/github-webhook... (push)
Edit Delete

Step 8.3: Now, let us modify some files in the repository and build the pipeline again. Here, I have cloned the repository, added a new image '2.jpg' to the images folder, modify the title and image name in index.html.

```
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ$ git clone https://github.com/parvatijay2901/devopsIQ.git
Cloning into 'devopsIQ'...
remote: Enumerating objects: 86, done.
remote: Total 86 (delta 0), reused 0 (delta 0), pack-reused 86
Unpacking objects: 100% (86/86), 11.46 MiB | 4.09 MiB/s, done.
parvati@LAPTOP-S2R1236D:~/Team1$ ls
Team1.pem devopsIQ
parvati@LAPTOP-S2R1236D:~/Team1$ cd devopsIQ/
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ$ ls
Dockerfile azure-pipelines.yml devopsIQ docker-compose
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ$ cd devopsIQ/
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ/devopsIQ$ ls
images index.html
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ/devopsIQ$ cat index.html
<html>
<title>Jenkins Final Website</title>
<body background="images/1.jpg">
</body>
</html>
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ/devopsIQ$ vim index.html
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ/devopsIQ$ cat index.html
<html>
<title>Jenkins Test2 Website</title>
<body background="images/2.jpg">
</body>
</html>
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ/devopsIQ$
```

Step 8.4: Commit the changes and push the contents.

```
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ$ git add .
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ$ git commit -m "Modified index.html"
[master 18f5d6a] Modified index.html
 1 file changed, 2 insertions(+), 2 deletions(-)
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ$ git remote add origin https://github.com/parvatijay2901/devopsIQ.git
fatal: remote origin already exists.
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ$ git push origin master
Username for 'https://github.com': parvatijay2901
Password for 'https://parvatijay2901@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 429 bytes | 214.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/parvatijay2901/devopsIQ.git
 cc26380..18f5d6a  master -> master
parvati@LAPTOP-S2R1236D:~/Team1/devopsIQ$
```

Step 8.5: Build the pipeline CICD again.

The screenshot shows the Jenkins Pipeline CICD dashboard. At the top, there are navigation links for 'Dashboard' and 'CICD'. Below the header, the title 'Build Pipeline: CICD' is displayed. Underneath the title are several icons: Run, History, Configure, Add Step, Delete, and Manage. Two pipeline items are listed: '#9 Test1' and '#10 Test2'. Both items show a green status bar indicating they were run on November 10, 2021, at 2:04:50 PM. The duration for #9 is 1.9 sec and for #10 is 2 sec. The user 'parvati2901' is associated with both runs. The background of the dashboard has a yellow-orange gradient.



Step 8.6: Verify that the new website was hosted by the slaves.

The screenshot shows a grid of Jenkins slave host interfaces. Each host displays a circular diagram divided into four quadrants: DEV (top-left), OPS (top-right), PLAN (bottom-left), and RELEASE (bottom-right). The quadrants are further divided into sub-steps: CREATE (green), PACKAGE (blue), CONFIGURE (purple), and MONITOR (red). The 'CREATE' and 'PACKAGE' steps are typically located in the DEV quadrant, while 'CONFIGURE' and 'MONITOR' are in the OPS quadrant. The 'PLAN' and 'RELEASE' steps are often shared or placed in the OPS quadrant. The Jenkins interface includes a header with tabs for 'CICD [Jenkins]', 'Jenkins Test2 Website', and 'Jenkins Test2 Website'. The bottom of the screen shows a Windows taskbar with the same set of application icons and system status indicators as the previous screenshot.

