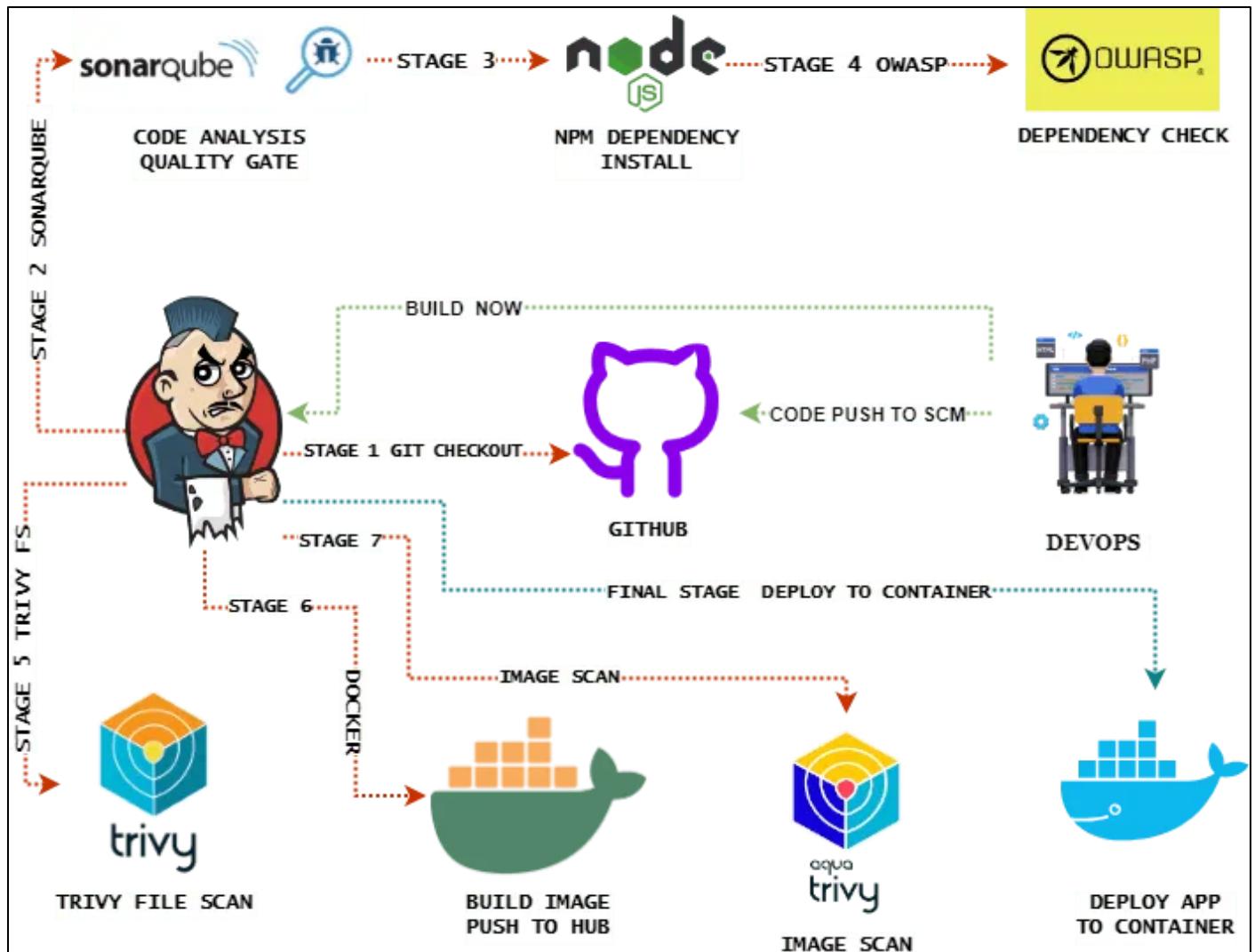


Project

Deploying Zomato Clone App with DevSecOps CI/CD

[Ajay Gaur]

Zomato Clone Architecture –



Steps:

- Step 1** — Launch an Ubuntu (22.04) T2 Large Instance
- Step 2** — Install Jenkins, Docker and Trivy. Create a Sonarqube Container using Docker.
- Step 3** — Install Plugins like JDK, Sonarqube Scanner, Nodejs, & OWASP Dependency Check.
- Step 4** — Create a Pipeline Project in Jenkins using a Declarative Pipeline
- Step 5** — Install OWASP Dependency Check Plugins
- Step 6** — Docker Image Build and Push
- Step 7** — Deploy the image using Docker
- Step 8** — Terminate the AWS EC2 Instances.

Now, let's get started and dig deeper into each of these steps.

Step 1 - Launch an Ubuntu (22.04) T2 Large Instance

Launch an AWS T2 Large Instance. Use the image as Ubuntu. You can create a new key pair or use an existing one. Enable HTTP and HTTPS settings in the Security Group and open all ports (not best case to open all ports but just for learning purposes it's okay).

The screenshot shows the AWS EC2 Instances page. On the left, there is a navigation sidebar with various options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and AMI Catalog. The main content area displays a table titled 'Instances (1/1) Info'. It shows one instance: 'zomato-app-clone' (Instance ID: i-044f7b2215c8b20ea), which is 'Running' and has an 't2.large' instance type. Below the table, there is a detailed view for the instance 'i-044f7b2215c8b20ea (zomato-app-clone)'. The 'Details' tab is selected, showing information such as Public IPv4 address (18.144.5.6), Instance state (Running), Private IP DNS name (ip-172-31-13-176.us-west-1.compute.internal), and Instance type (t2.large). Other tabs include Status and alarms, Monitoring, Security, Networking, Storage, and Tags.

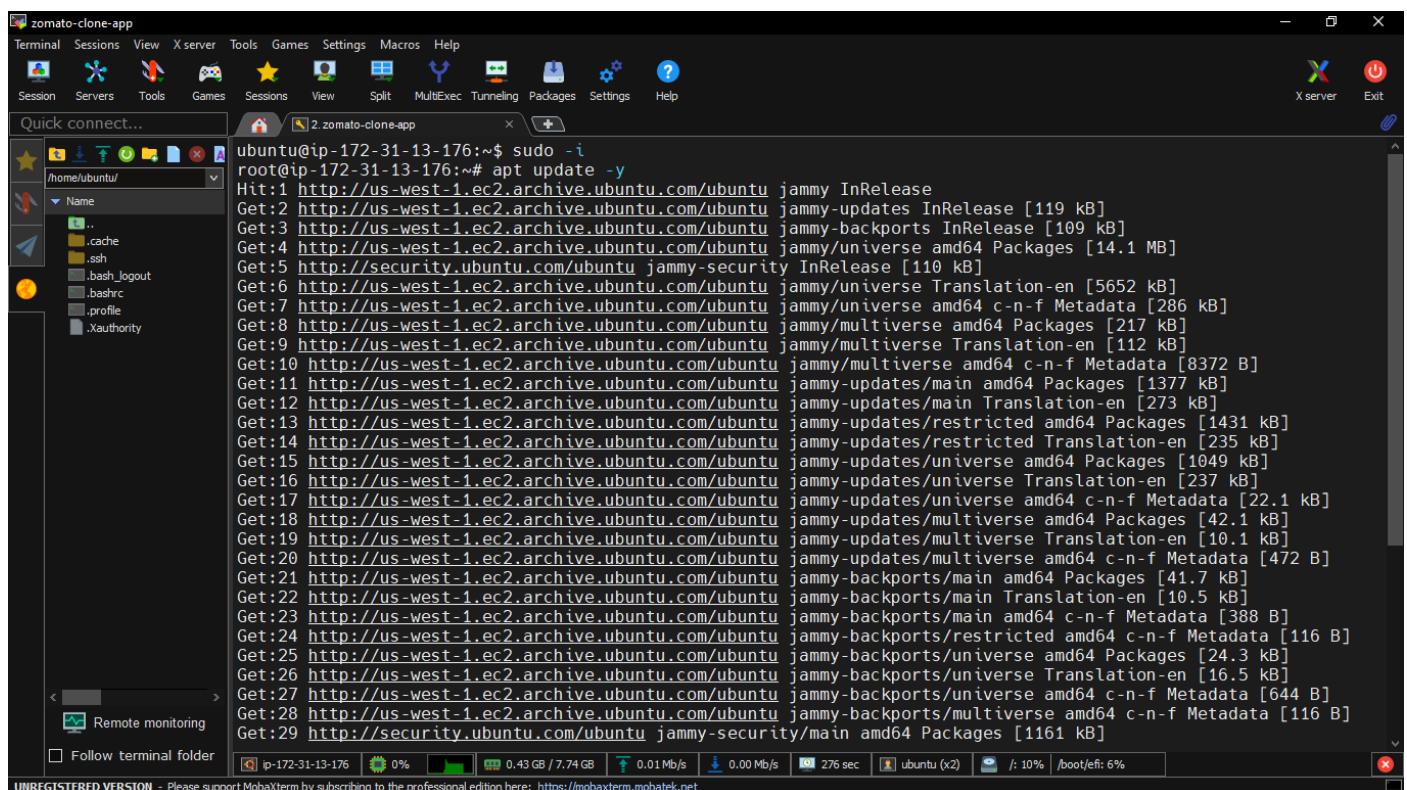
The screenshot shows the AWS Security Groups page. The left sidebar includes EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, and Network & Security (Security Groups selected). The main content shows a security group named 'sg-0983987dca7fb3518 - launch-wizard-5'. The 'Details' section provides information like Security group name (launch-wizard-5), Security group ID (sg-0983987dca7fb3518), Description (launch-wizard-5 created 2024-02-22T04:57:45.48Z), Owner (639711838771), Inbound rules count (4 Permission entries), and Outbound rules count (1 Permission entry). The 'Inbound rules' tab is selected, displaying four rules:

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
-	sgr-0b613c8a548b9b0f5	IPv4	SSH	TCP	22	0.0.0.0/0
-	sgr-02bd3f0cd0a89a71	IPv4	All traffic	All	All	0.0.0.0/0
-	sgr-05a875b82bcd905f	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-06fcf5dc276abb3f	IPv4	HTTPS	TCP	443	0.0.0.0/0

Step 2 - Install Jenkins, Docker and Trivy

→ To Install Jenkins

```
> sudo apt update -y
> wget -O - https://packages.adoptium.net/artifactory/api/gpg/key/public | tee
  /etc/apt/keyrings/adoptium.asc
> echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc]
https://packages.adoptium.net/artifactory/deb $(awk -F=
  '/^VERSION_CODENAME/{print$2}' /etc/os-release) main" | tee
  /etc/apt/sources.list.d/adoptium.list
> sudo apt update -y
> sudo apt install temurin-17-jdk -y
> /usr/bin/java -version
> curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
> echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
> sudo apt-get update -y
> sudo apt-get install jenkins -y
> sudo systemctl start Jenkins
> sudo systemctl status Jenkins
```



The screenshot shows a terminal window titled 'zomato-clone-app' in the MobaXterm interface. The terminal is running on a session named 'ubuntu@ip-172-31-13-176'. The user has run several commands to update the system, add the Adoptium Java repository, and install Jenkins. The output of the 'apt update' command is visible, showing numerous package downloads from the 'jammy' archive. The Jenkins installation command 'sudo apt-get install jenkins -y' is also present in the history.

```
ubuntu@ip-172-31-13-176:~$ sudo -i
root@ip-172-31-13-176:~# apt update -y
Get:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1377 kB]
Get:12 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [273 kB]
Get:13 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1431 kB]
Get:14 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [235 kB]
Get:15 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1049 kB]
Get:16 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [237 kB]
Get:17 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]
Get:18 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.1 kB]
Get:19 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.1 kB]
Get:20 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 B]
Get:21 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [41.7 kB]
Get:22 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [10.5 kB]
Get:23 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:24 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:25 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [24.3 kB]
Get:26 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.5 kB]
Get:27 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 B]
Get:28 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:29 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1161 kB]
```

```

zomato-clone-app
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2.zomato-clone-app
root@ip-172-31-13-176:~# wget -O - https://packages.adoptium.net/artifactory/api/gpg/key/public | tee /etc/apt/keys.d/adoptium.asc
--2024-02-22 05:05:15-- https://packages.adoptium.net/artifactory/api/gpg/key/public
Resolving packages.adoptium.net (packages.adoptium.net)... 151.101.43.42
Connecting to packages.adoptium.net (packages.adoptium.net)|151.101.43.42|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1793 (1.8K) [text/plain]
Saving to: 'STDOUT'

[Progress Bar] 1.75K --.-KB/s in 0s

2024-02-22 05:05:15 (51.5 MB/s) - written to stdout [1793/1793]

-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBGGTvTQBAC6ey144n7CG8foaff6mwgIBN1fIm1ILZDuGS4tMr0/XI8pgJnT
QvsPxZWevtSm7bEM0bzEoZjCwxBcJl1B0ui8k5kHMTI75gCmZPsOKLFWIEpuRBQ
PBoCuSw80apDmLnNDQLVQdFtEua5gaNa/rw9YsmBoxBqvgrijFU1dgYoQvH5+a
90YLWDn5Vv0gnVMb+aclwVzB/zJw3kHGSguMtlAheQiah7y8yomQn/UIX8yqDf
+11sP3+c87YcjkRqImRTtmKEDcEtGPAIXC6SYA+uEEkbYE0Fy0chkvtnWJ597fa
Epa14rnICU8z0j6X5z3v1aM2whrX9oq9X8PABEBAAQ0QEfkB3B0aXViEdQryBL
ZXkgKERFQi9SUE0gU2lnbmLuZyBLZKkpIDx0Zw11cmLuLWRldkBLy2xpcHNLLm9y
Zz6JAVIEewIADwWIQ07BNdTyQUNm100PzmEPeilZfjwSwUCYZ09NAiBawULCQgh
AgMiAgEGFQoJCAsCBByCAwECHgcCF4AACgkQhDxIpWX48Et4AggAjjJzYVuKV3nG
7ngInng18G/m9Johr7BmwgcQXYhdy5hVkmCx5JLxEz2LMBUH/F2nD595hgjMabk
KVib20X8lq9RsNbdfc2hBcWU6qyHKxsIq14boi2/XDyEzzMyyZWNGo/27Ci7Xmj
pwu3nh0pDdpqdyWDIKojbVnxLCRY8as8Sm+1ufi709Kci4MuwhNsULCswb/fju
NKeHkrHbLcHKUUIEcmtSKRWrpMYBz1HYOBGz4xPuElwUp71ehfoyZBzp6RDRf
Klqi5KSZgrkbDQRhk700AqgAq14okly8cFrpYVenEQPib75AUZfKRpMduiR6IxAj
SKch7aSoFZ9aubUEBVPzsy15svxoEPe1i4TdbF+m9Gy42Ec0Lla3ArLjt5H8fRL

```

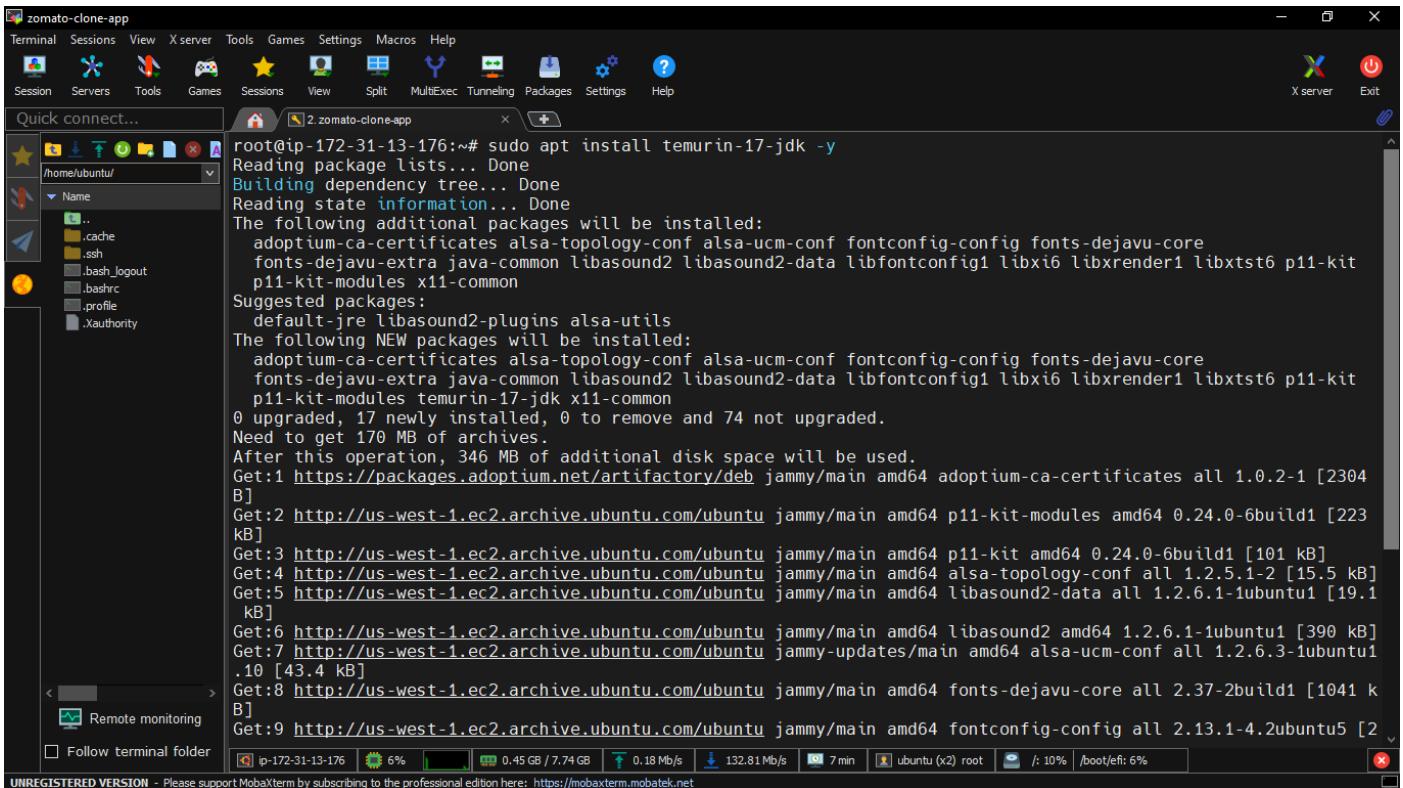
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

```

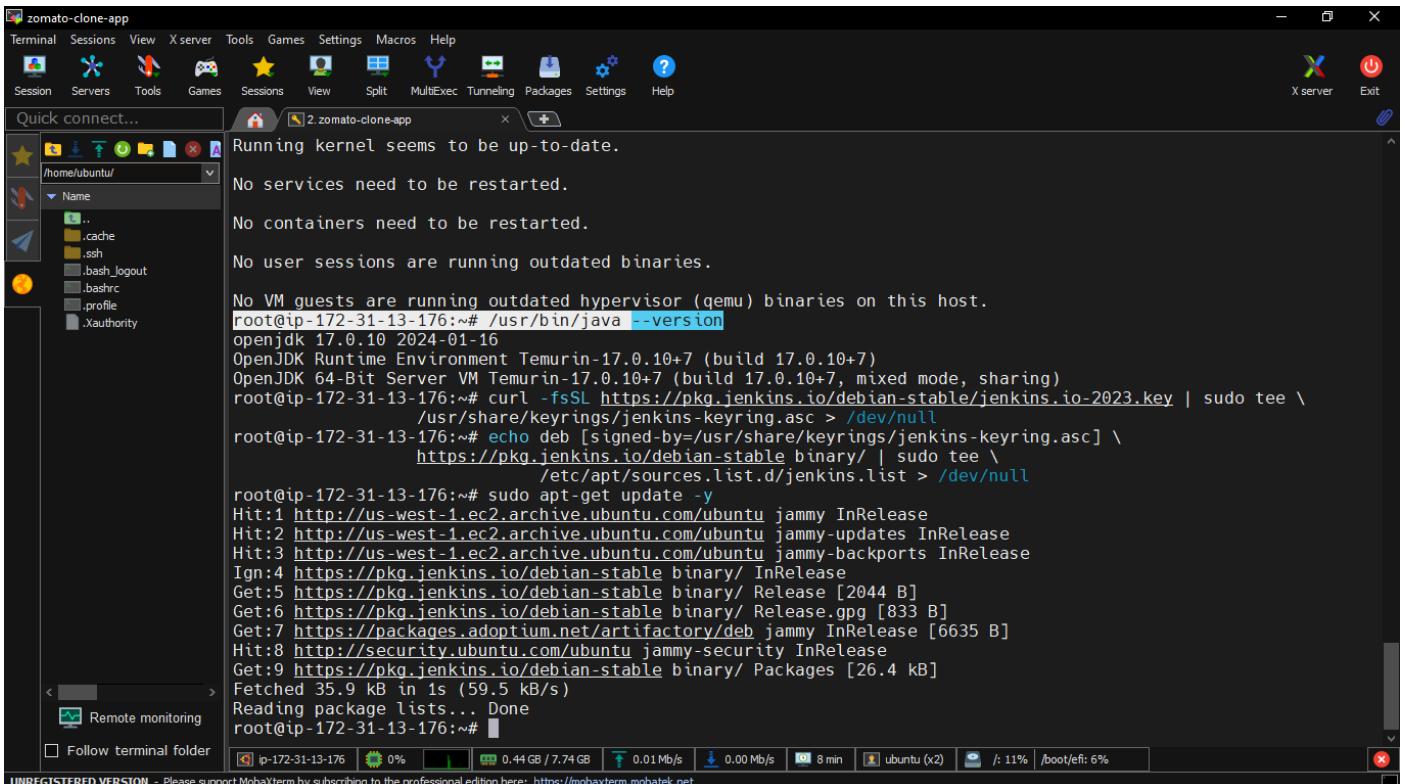
zomato-clone-app
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2.zomato-clone-app
root@ip-172-31-13-176:~# echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packages.adoptium.net/artifactory/deb $(awk -F '/^VERSION_CODENAME/{print$2}' /etc/os-release) main" | tee /etc/apt/sources.list.d/adoptum.list
deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packages.adoptium.net/artifactory/deb jammy main
root@ip-172-31-13-176:~# sudo apt update -y
Hit:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://packages.adoptium.net/artifactory/deb jammy InRelease [6635 B]
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://packages.adoptium.net/artifactory/deb jammy/main amd64 Packages [6965 B]
Fetched 13.6 kB in 1s (25.0 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
74 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-13-176:~#

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>



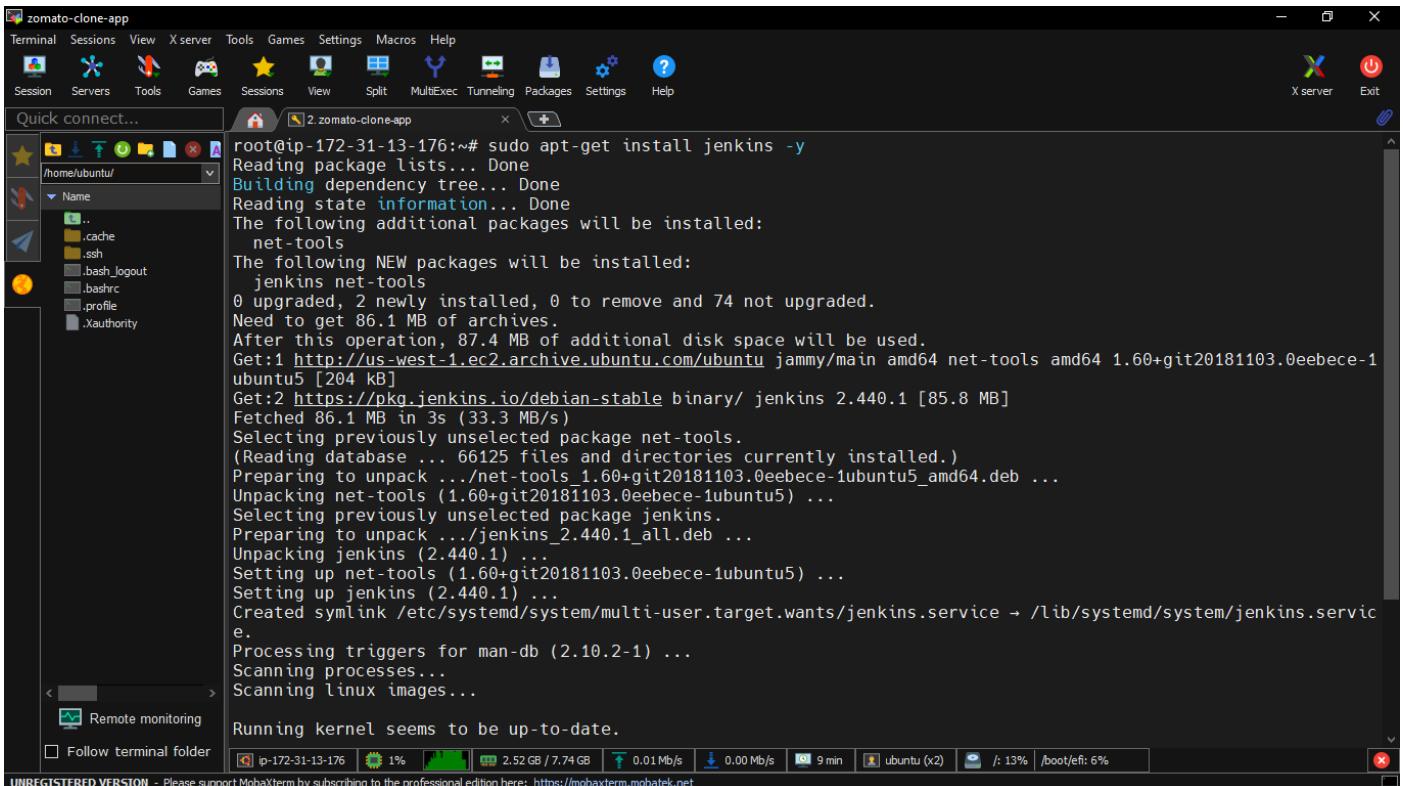
```
zomato-clone-app
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2.zomato-clone-app
root@ip-172-31-13-176:~# sudo apt install temurin-17-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
adoptium-ca-certificates alsal-topology-conf alsal-ucm-conf fontconfig-config fonts-dejavu-core
fonts-dejavu-extra java-common libasound2 libasound2-data libfontconfig1 libxi6 libxrender1 libxtst6 p11-kit
p11-kit-modules x11-common
Suggested packages:
default-jre libasound2-plugins alsal-utils
The following NEW packages will be installed:
adoptium-ca-certificates alsal-topology-conf alsal-ucm-conf fontconfig-config fonts-dejavu-core
fonts-dejavu-extra java-common libasound2 libasound2-data libfontconfig1 libxi6 libxrender1 libxtst6 p11-kit
p11-kit-modules temurin-17-jdk x11-common
0 upgraded, 17 newly installed, 0 to remove and 74 not upgraded.
Need to get 170 MB of archives.
After this operation, 346 MB of additional disk space will be used.
Get:1 https://packages.adoptium.net/artifactory/deb jammy/main amd64 adoptium-ca-certificates all 1.0.2-1 [2304
B]
Get:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 p11-kit-modules amd64 0.24.0-6build1 [223
KB]
Get:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 p11-kit amd64 0.24.0-6build1 [101 KB]
Get:4 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 alsal-topology-conf all 1.2.5.1-2 [15.5 KB]
Get:5 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libasound2-data all 1.2.6.1-1ubuntu1 [19.1
KB]
Get:6 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libasound2 amd64 1.2.6.1-1ubuntu1 [390 kB]
Get:7 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 alsal-ucm-conf all 1.2.6.3-1ubuntu1
.10 [43.4 kB]
Get:8 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-core all 2.37-2build1 [1041 k
B]
Get:9 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fontconfig-config all 2.13.1-4.2ubuntu5 [2
]
ip-172-31-13-176 6% 0.45 GB / 7.74 GB 0.18 Mb/s 132.81 Mb/s 7 min ubuntu (x2) root :/ 10% /boot/efi: 6%
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```



```
zomato-clone-app
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2.zomato-clone-app
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-13-176:~# /usr/bin/java --version
openjdk 17.0.10 2024-01-16
OpenJDK Runtime Environment Temurin-17.0.10+7 (build 17.0.10+7)
OpenJDK 64-Bit Server VM Temurin-17.0.10+7 (build 17.0.10+7, mixed mode, sharing)
root@ip-172-31-13-176:~# curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
root@ip-172-31-13-176:~# echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
root@ip-172-31-13-176:~# sudo apt-get update -y
Hit:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:7 https://packages.adoptium.net/artifactory/deb jammy InRelease [6635 B]
Hit:8 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:9 https://pkg.jenkins.io/debian-stable binary/ Packages [26.4 kB]
Fetched 35.9 kB in 1s (59.5 kB/s)
Reading package lists... Done
root@ip-172-31-13-176:~#
```

ip-172-31-13-176 0% 0.44 GB / 7.74 GB 0.01 Mb/s 0.00 Mb/s 8 min ubuntu (x2) root :/ 11% /boot/efi: 6%

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net



```

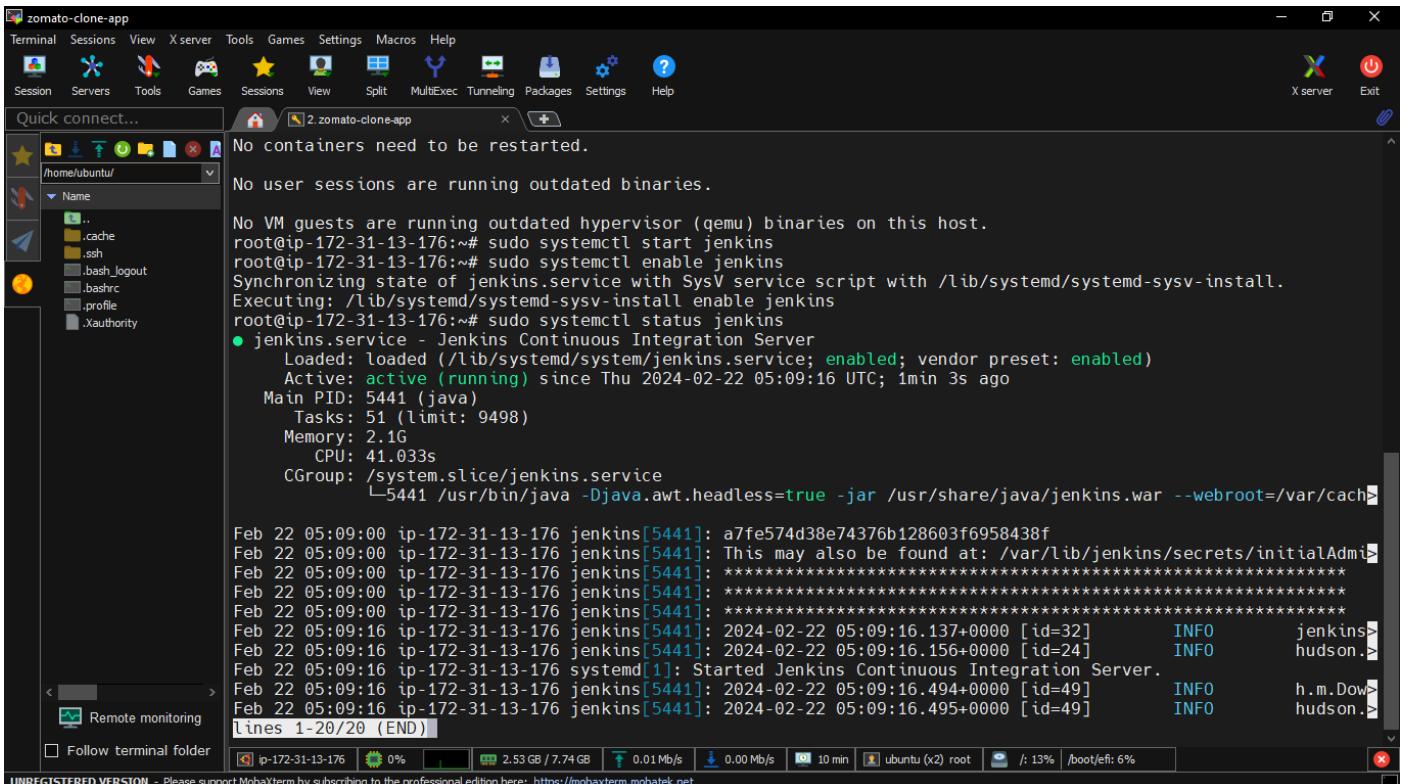
zomato-clone-app
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2.zomato-clone-app
root@ip-172-31-13-176:~# sudo apt-get install jenkins -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 74 not upgraded.
Need to get 86.1 MB of archives.
After this operation, 87.4 MB of additional disk space will be used.
Get:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1
ubuntu5 [204 kB]
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.440.1 [85.8 MB]
Fetched 86.1 MB in 3s (33.3 MB/s)
Selecting previously unselected package net-tools.
(Reading database ... 66125 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.440.1_all.deb ...
Unpacking jenkins (2.440.1) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.440.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

ip-172-31-13-176 1% 2.52 GB / 7.74 GB 0.01 Mb/s 0.00 Mb/s 9 min ubuntu (x2) : 13% /boot/efi: 6%

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>



```

zomato-clone-app
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2.zomato-clone-app
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-13-176:~# sudo systemctl start jenkins
root@ip-172-31-13-176:~# sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
root@ip-172-31-13-176:~# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
    Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
    Active: active (running) since Thu 2024-02-22 05:09:16 UTC; 1min 3s ago
      Main PID: 5441 (java)
        Tasks: 51 (limit: 9498)
       Memory: 2.1G
          CPU: 41.033s
        CGroup: /system.slice/jenkins.service
           └─5441 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cach

Feb 22 05:09:00 ip-172-31-13-176 jenkins[5441]: a7fe574d38e74376b128603f6958438f
Feb 22 05:09:00 ip-172-31-13-176 jenkins[5441]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPw
Feb 22 05:09:00 ip-172-31-13-176 jenkins[5441]: ****
Feb 22 05:09:00 ip-172-31-13-176 jenkins[5441]: ****
Feb 22 05:09:00 ip-172-31-13-176 jenkins[5441]: ****
Feb 22 05:09:16 ip-172-31-13-176 jenkins[5441]: 2024-02-22 05:09:16.137+0000 [id=32]      INFO  jenkins
Feb 22 05:09:16 ip-172-31-13-176 jenkins[5441]: 2024-02-22 05:09:16.156+0000 [id=24]      INFO  hudson
Feb 22 05:09:16 ip-172-31-13-176 systemd[1]: Started Jenkins Continuous Integration Server.
Feb 22 05:09:16 ip-172-31-13-176 jenkins[5441]: 2024-02-22 05:09:16.494+0000 [id=49]      INFO  h.m.Dow
Feb 22 05:09:16 ip-172-31-13-176 jenkins[5441]: 2024-02-22 05:09:16.495+0000 [id=49]      INFO  hudson.

Lines 1-20/20 (END)

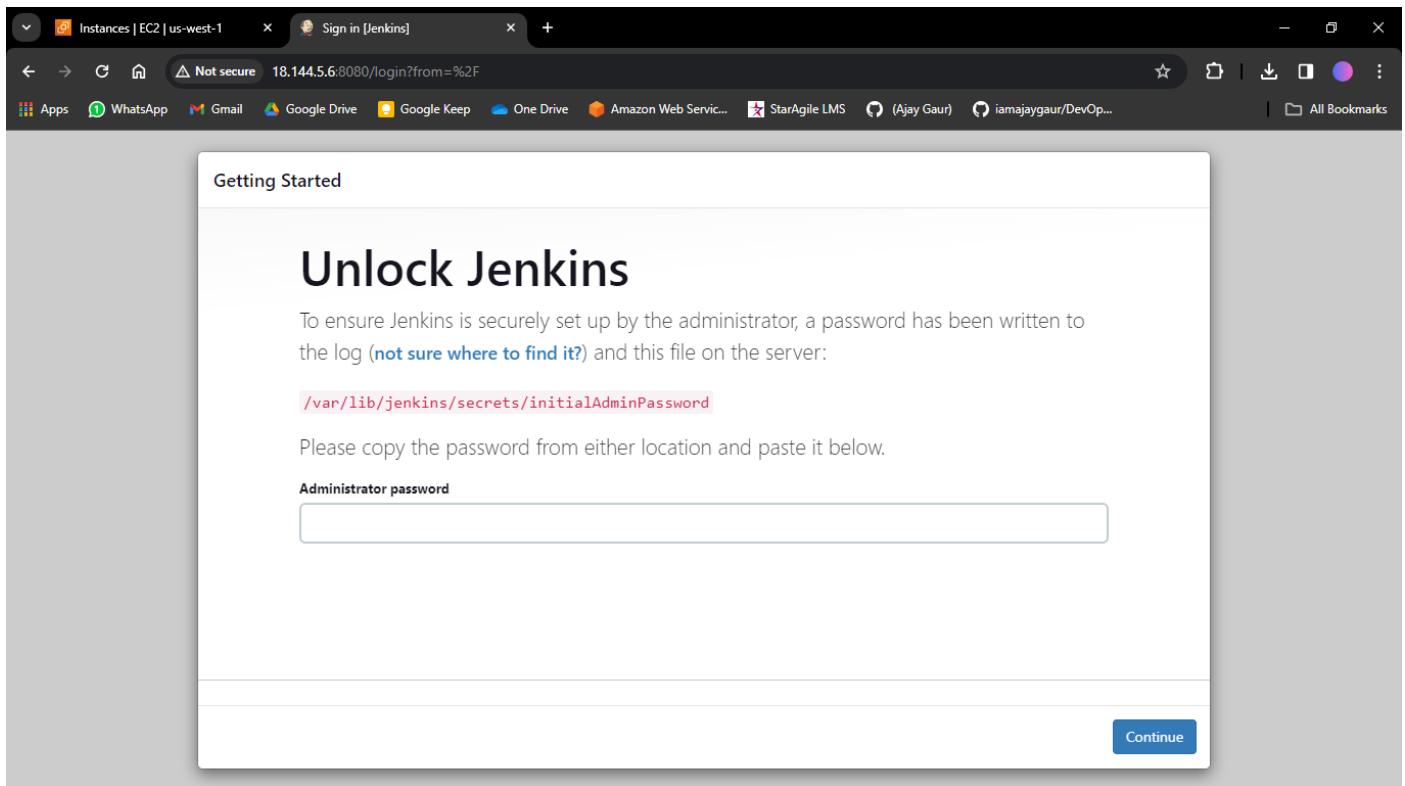
ip-172-31-13-176 0% 2.53 GB / 7.74 GB 0.01 Mb/s 0.00 Mb/s 10 min ubuntu (x2) root : 13% /boot/efi: 6%

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

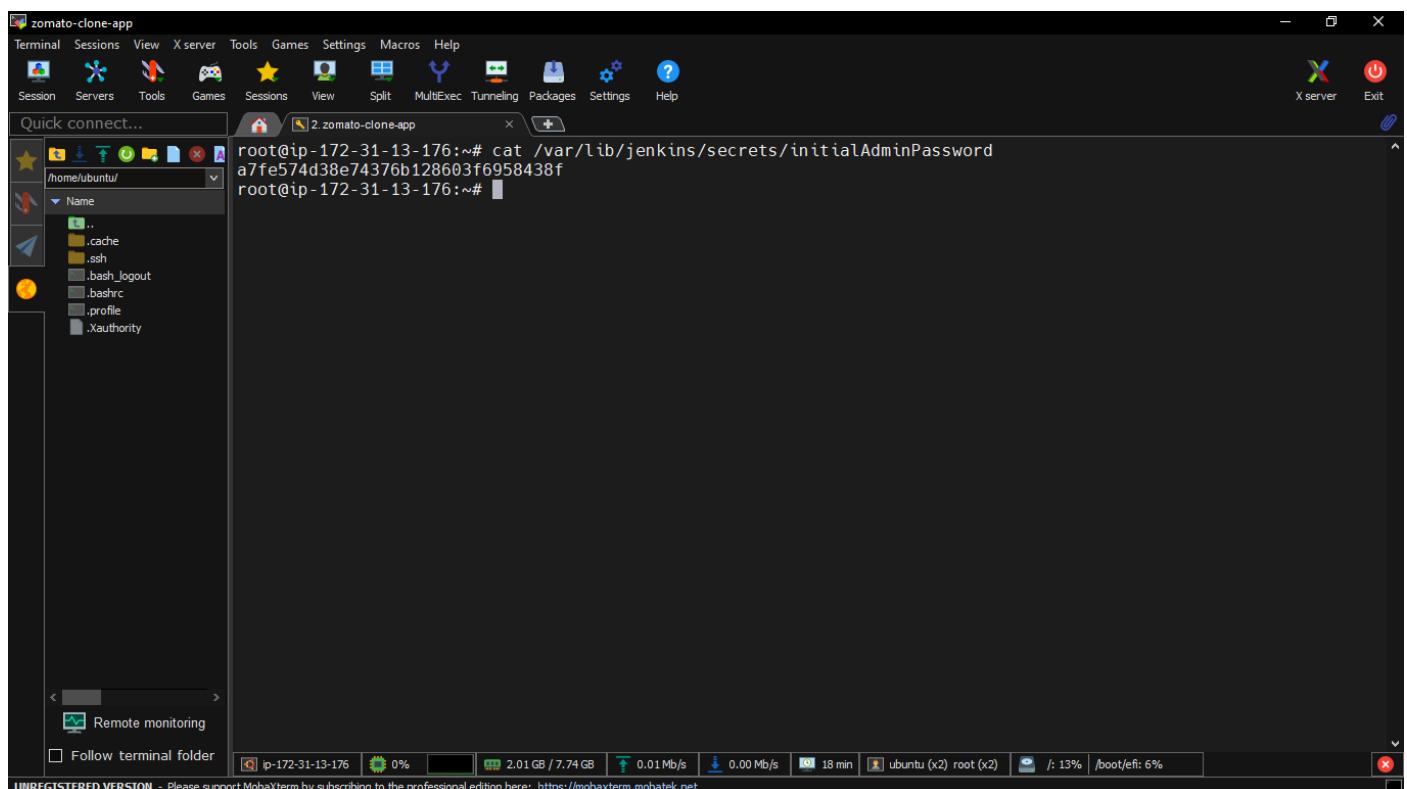
Once Jenkins is installed, you will need to go to your AWS EC2 Security Group and open Inbound Port 8080, since Jenkins works on Port 8080.

→ http://<public_ip>:8080/



> cat /var/lib/jenkins/secrets/initialAdminPassword

→ Get the Password and put into Jenkins for login.



The screenshot shows a browser window titled 'Sign in [Jenkins]' at the URL '18.144.5.6:8080/login?from=%2F'. The page is titled 'Getting Started' and features a large heading 'Unlock Jenkins'. Below it, a text block says: 'To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:' followed by the path '/var/lib/jenkins/secrets/initialAdminPassword'. A placeholder text box labeled 'Administrator password' contains several dots ('.....'). At the bottom right is a blue 'Continue' button.

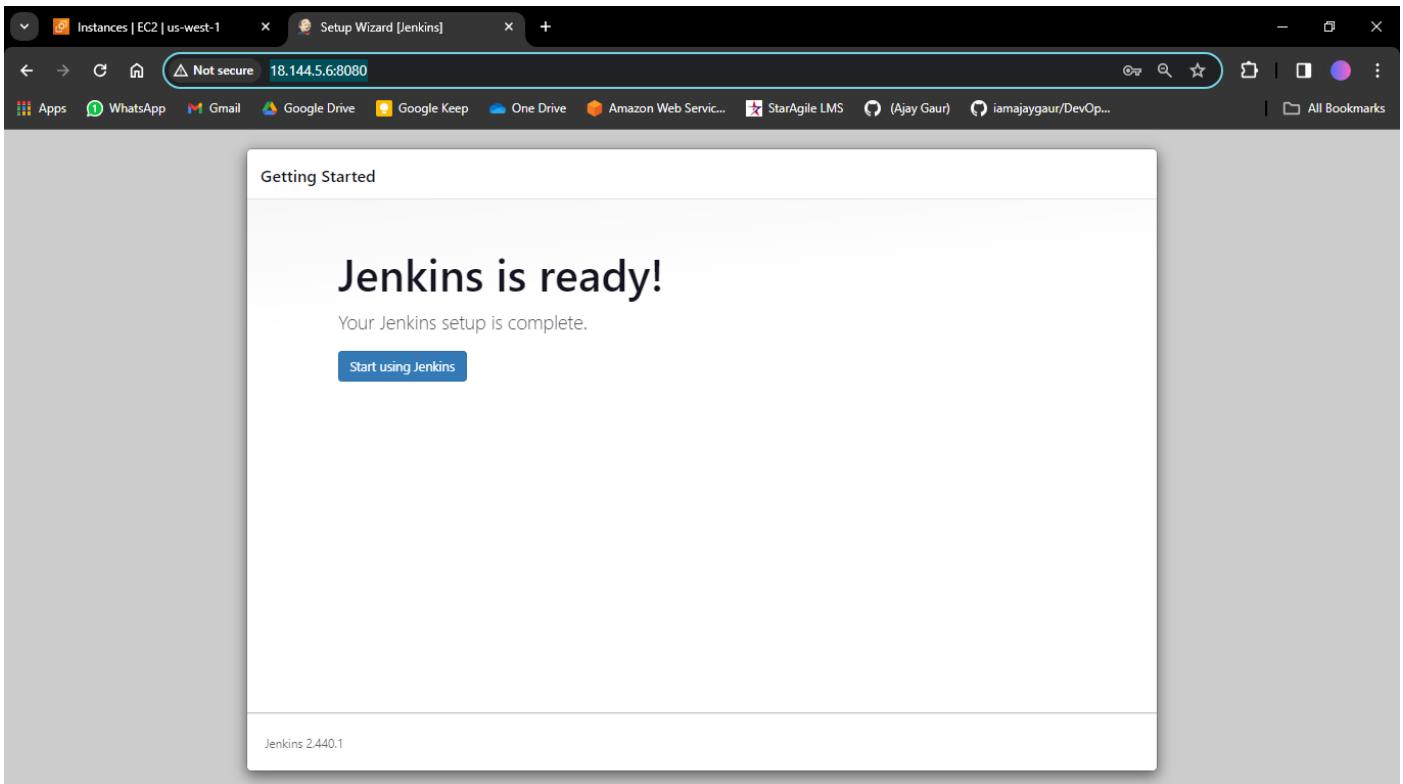
→ Install the suggested plugins.

The screenshot shows a browser window titled 'Setup Wizard [Jenkins]' at the URL '18.144.5.6:8080'. The page is titled 'Getting Started' and features a large heading 'Customize Jenkins'. Below it, a text block says: 'Plugins extend Jenkins with additional features to support many different needs.' Two options are presented in boxes: 'Install suggested plugins' (described as installing plugins the Jenkins community finds most useful) and 'Select plugins to install' (described as selecting and installing plugins most suitable for your needs). At the bottom left is a note 'Jenkins 2.440.1' and at the bottom right is the URL '18.144.5.6:8080/#'.

→ Create a user click on save and continue.

The screenshot shows the Jenkins Setup Wizard 'Getting Started' page. It contains fields for entering user information: Username (ajay), Password (three dots), Confirm password (three dots), Full name (Ajay Gaur), and E-mail address (ajay9493@gmail.com). At the bottom, there are two buttons: 'Skip and continue as admin' and a highlighted 'Save and Continue' button.

The screenshot shows the Jenkins Setup Wizard 'Instance Configuration' page. It displays the Jenkins URL field with the value 'http://18.144.5.6:8080/'. Below the field, there is explanatory text about the Jenkins URL's purpose and a note about the proposed default value. At the bottom, there are two buttons: 'Not now' and a highlighted 'Save and Finish' button.

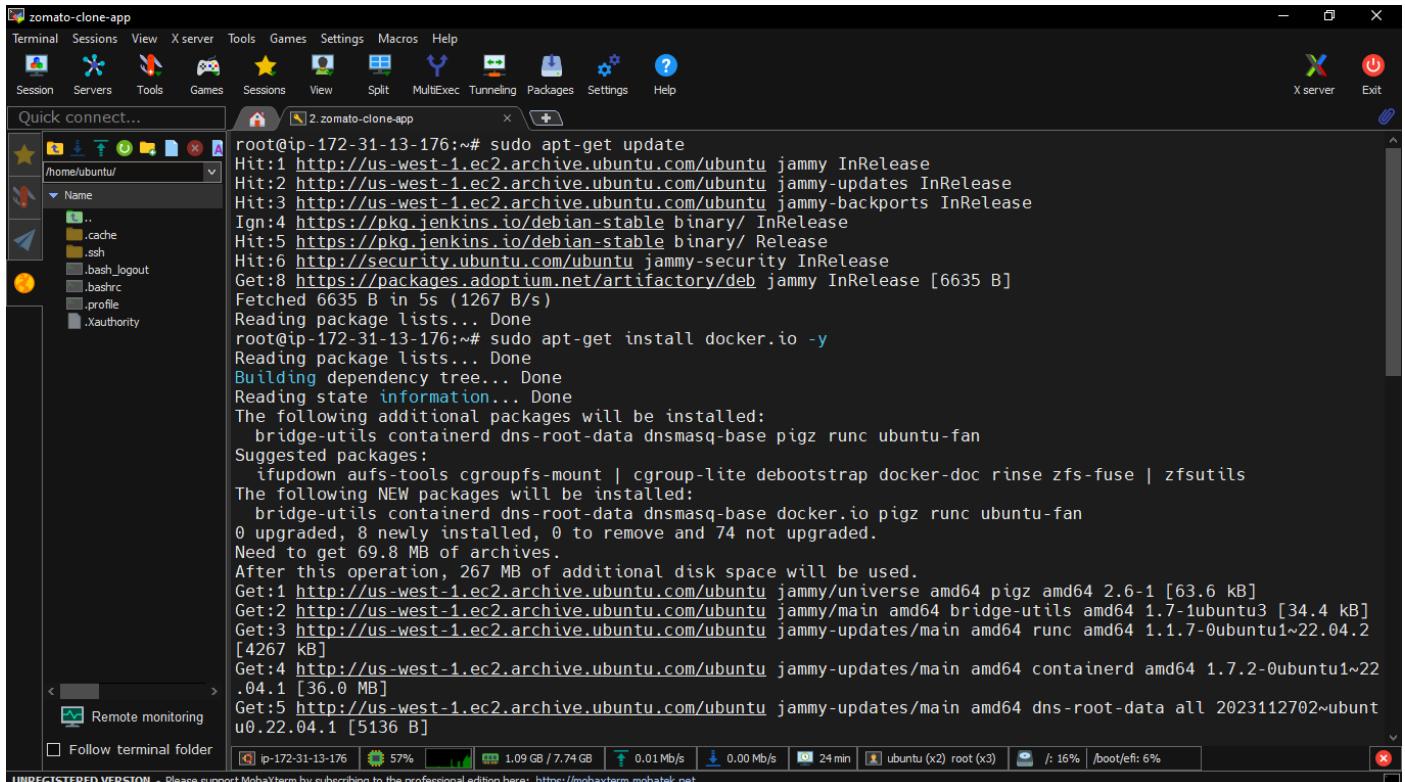


→ Jenkins Getting Started Screen.

A screenshot of a web browser showing the Jenkins 'Dashboard' page. The title bar says 'Instances | EC2 | us-west-1' and 'Dashboard [Jenkins]'. The address bar shows 'Not secure 18.144.5.6:8080'. Below the address bar is a toolbar with various icons for Google services like Apps, WhatsApp, Gmail, Google Drive, Google Keep, One Drive, Amazon Web Services, StarAgile LMS, and user profiles for Ajay Gaur and iamajaygaur/DevOp... A 'All Bookmarks' button is also present. The main content area has a dark header with the Jenkins logo and the word 'Jenkins'. Below the header, there's a navigation menu with links: '+ New Item', 'People', 'Build History', 'Manage Jenkins', and 'My Views'. On the left, there are two dropdown menus: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). In the center, there's a section titled 'Welcome to Jenkins!' with the sub-section 'Start building your software project'. It includes a 'Create a job' button, a 'Set up a distributed build' section with 'Set up an agent' and 'Configure a cloud' buttons, and a link 'Learn more about distributed builds'. At the bottom right, there are links for 'REST API' and 'Jenkins 2.440.1'.

→ To Install Docker

- > sudo apt-get update
- > sudo apt-get install docker.io -y
- > sudo usermod -aG docker \$USER #my case is ubuntu
- > newgrp docker
- > sudo chmod 777 /var/run/docker.sock

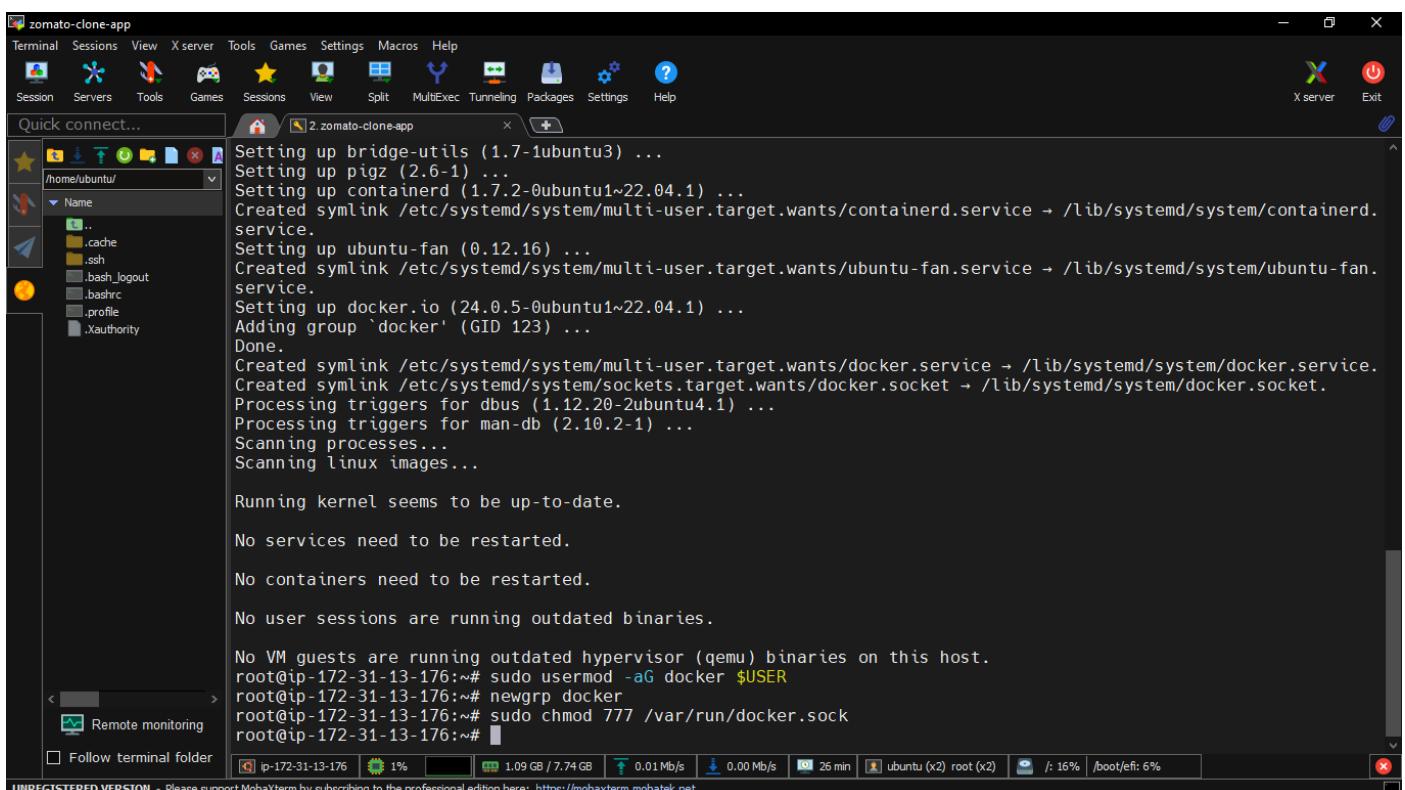


The screenshot shows a MobaXterm window titled "zomato-clone-app". The terminal session is running as root on an Ubuntu system. The user has run several commands to update the package list, install Docker, and add their user to the Docker group. The terminal output is as follows:

```

root@ip-172-31-13-176:~# sudo apt-get update
Hit:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:5 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:6 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:8 https://packages.adoptopenjdk.net/artifactory/deb jammy InRelease [6635 B]
Fetched 6635 B in 5s (1267 B/s)
Reading package lists... Done
root@ip-172-31-13-176:~# sudo apt-get install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base 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 pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 74 not upgraded.
Need to get 69.8 MB of archives.
After this operation, 267 MB of additional disk space will be used.
Get:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]
Get:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.7-0ubuntu1~22.04.2 [4267 kB]
Get:4 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.2-0ubuntu1~22.04.1 [36.0 MB]
Get:5 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dns-root-data all 2023112702~ubuntu0.22.04.1 [5136 B]
0 upgraded, 8 newly installed, 0 to remove and 74 not upgraded.
Need to get 69.8 MB of archives.
After this operation, 267 MB of additional disk space will be used.
Get:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]
Get:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.7-0ubuntu1~22.04.2 [4267 kB]
Get:4 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.2-0ubuntu1~22.04.1 [36.0 MB]
Get:5 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dns-root-data all 2023112702~ubuntu0.22.04.1 [5136 B]
```

At the bottom of the terminal, there is a status bar showing network usage and battery level.



The screenshot shows the same MobaXterm window as the previous one, but the terminal session is now completed. The user has run the commands to update the package list, install Docker, and add their user to the Docker group. The terminal output is as follows:

```

Setting up bridge-utils (1.7-1ubuntu3) ...
Setting up pigz (2.6-1) ...
Setting up containerd (1.7.2-0ubuntu1~22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (24.0.5-0ubuntu1~22.04.1) ...
Adding group 'docker' (GID 123) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

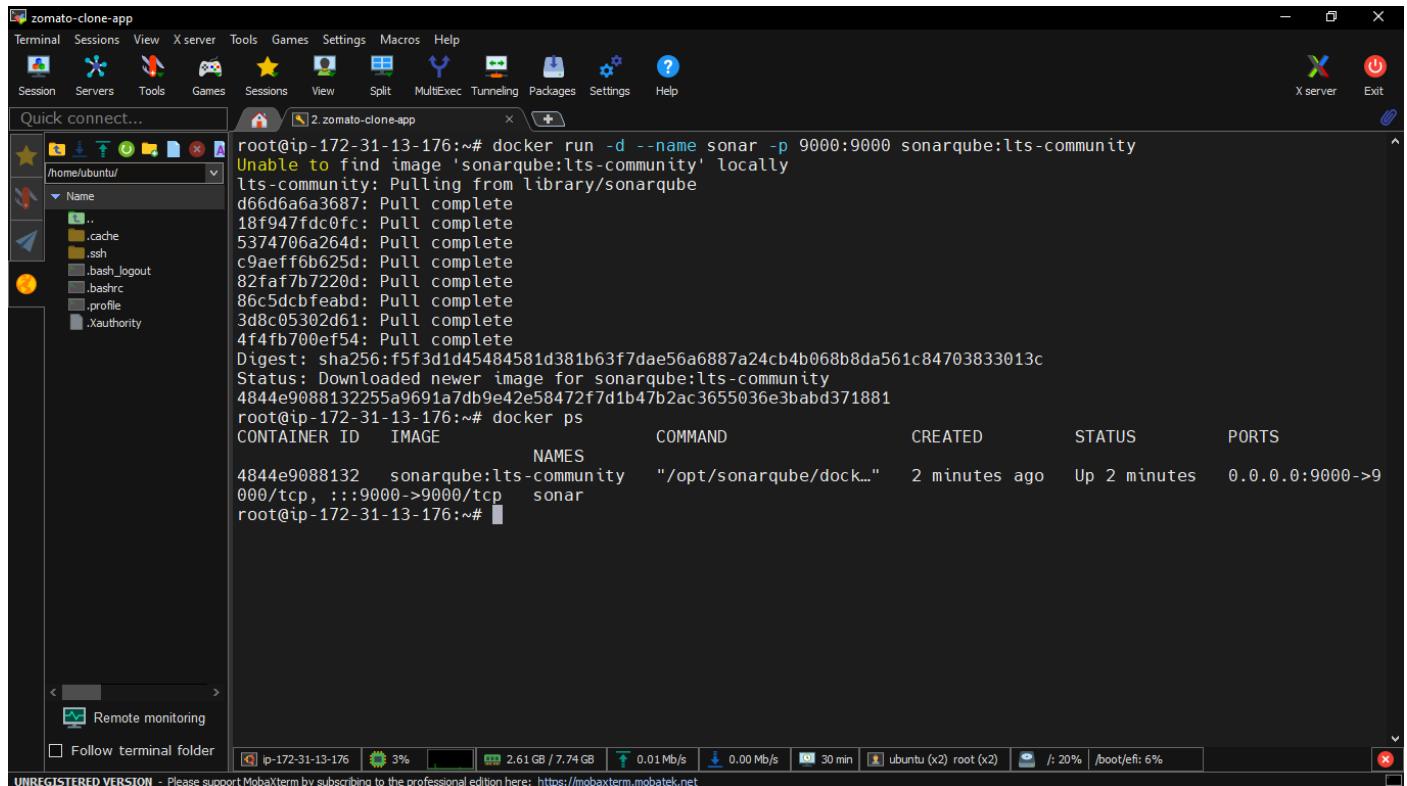
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-13-176:~# sudo usermod -aG docker $USER
root@ip-172-31-13-176:~# newgrp docker
root@ip-172-31-13-176:~# sudo chmod 777 /var/run/docker.sock
root@ip-172-31-13-176:~#
```

At the bottom of the terminal, there is a status bar showing network usage and battery level.

After the docker installation, we create a sonarqube container (Remember to add 9000 ports in the security group).

```
> docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
> docker ps
```



The screenshot shows a terminal window titled 'zomato-clone-app' in the MobaXterm interface. The terminal displays the following command and its output:

```
root@ip-172-31-13-176:~# docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
d66d6a6a3687: Pull complete
18f947fdc0fc: Pull complete
5374706a264d: Pull complete
c9aeff6b625d: Pull complete
82faf7b7220d: Pull complete
86c5dcfeabd: Pull complete
3d8c05302d61: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:f5f3d1d45484581d381b63f7dae56a6887a24cb4b068b8da561c84703833013c
Status: Downloaded newer image for sonarqube:lts-community
4844e9088132255a9691a7db9e42e58472f7d1b47b2ac3655036e3babd371881
root@ip-172-31-13-176:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
4844e9088132 sonarqube:lts-community "/opt/sonarqube/dock..." 2 minutes ago Up 2 minutes 0.0.0.0:9000->
000/tcp, :::9000->9000/tcp sonar
root@ip-172-31-13-176:~#
```

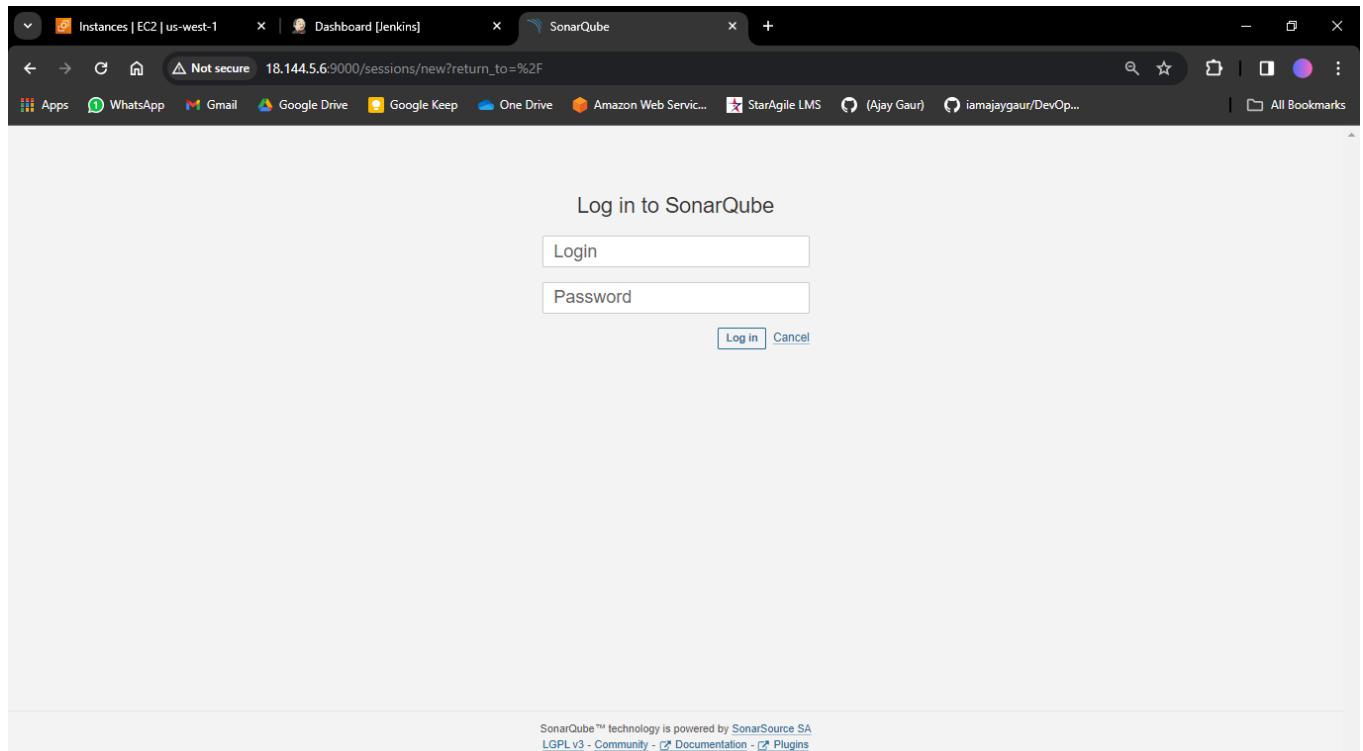
The terminal also shows system monitoring information at the bottom:

- ip-172-31-13-176
- 3% CPU usage
- 2.61 GB / 7.74 GB RAM
- 0.01 Mb/s download, 0.00 Mb/s upload
- 30 min uptime
- ubuntu (x2) root (x2)
- /: 20% /boot/efi: 6%

A message at the bottom left says: UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

→ Now our sonarqube is up and running

→ http://<public_ip>/sessions/new?return_to=%2F



→ Enter username and password, click on login and change password

→ username - admin

→ password – admin

Log in to SonarQube

admin

Log in Cancel

SonarQube™ technology is powered by SonarSource SA
LGPL v3 - Community - Documentation - Plugins

→ Update New password

Update your password

This account should not use the default password.

Enter a new password

All fields marked with * are required

Old Password *

New Password *

Confirm Password *

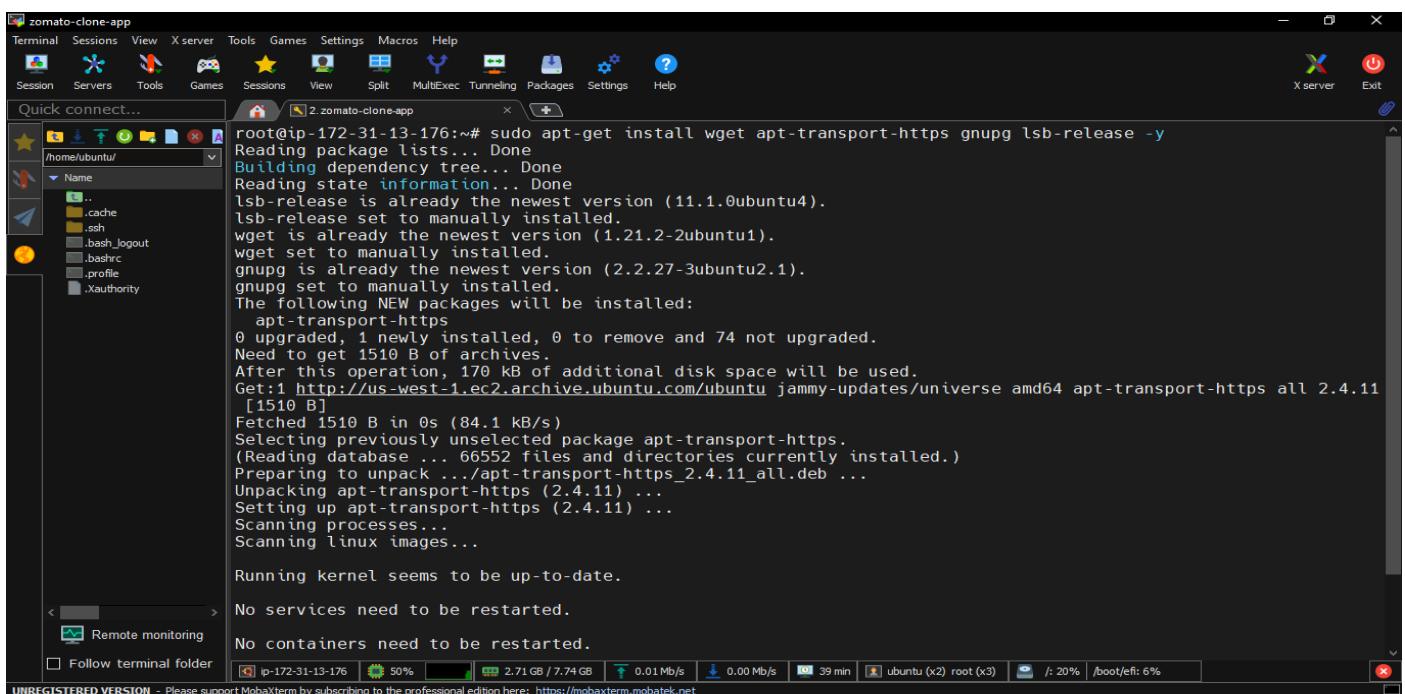
Update

→ This is Sonar Dashboard.

The screenshot shows a web browser window with the URL 18.144.5.6:9000/projects/create. The page title is "How do you want to create your project?". The header includes the SonarQube logo, navigation links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and a search bar. Below the header, there is a section titled "How do you want to create your project?" with a sub-section about creating from DevOps platforms. It features five cards: "From Azure DevOps" (with a Microsoft Azure icon), "From Bitbucket Server" (with a Bitbucket icon), "From Bitbucket Cloud" (with a Bitbucket icon), "From GitHub" (with a GitHub icon), and "From GitLab" (with a GitLab icon). Each card has a "Set up global configuration" link below it. At the bottom left, there is a "Manually" button with a double-angle bracket icon.

→ To Install Trivy

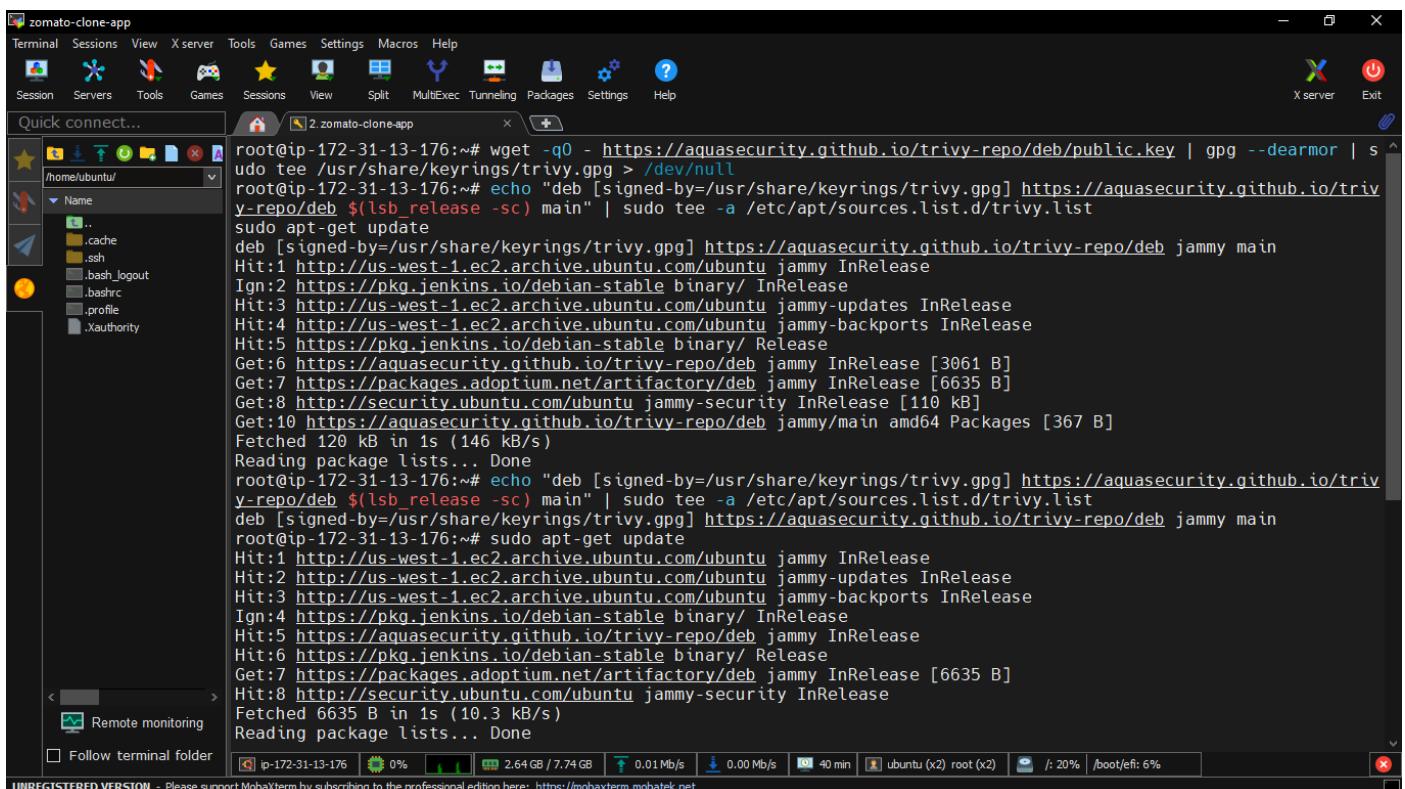
- > sudo apt-get install wget apt-transport-https gnupg lsb-release -y
- > wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null
- > echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb \$(lsb_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.list
- > sudo apt-get update
- > sudo apt-get install trivy -y



The screenshot shows a terminal window titled 'zomato-clone-app' running on a Linux system. The command entered is:

```
root@ip-172-31-13-176:~# sudo apt-get install wget apt-transport-https gnupg lsb-release -y
```

The terminal output shows the process of building the dependency tree, reading package lists, and installing the required packages. It also shows the download of the apt-transport-https package from the US West Archive.



The screenshot shows a terminal window titled 'zomato-clone-app' running on a Linux system. The commands entered are:

```
root@ip-172-31-13-176:~# wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null
root@ip-172-31-13-176:~# echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.list
sudo apt-get update
```

The terminal output shows the process of adding the Trivy repository, hitting several mirrors for the package, and then updating the package lists. The final command is 'sudo apt-get update'.

```
root@ip-172-31-13-176:~# sudo apt-get install trivy -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  trivy
0 upgraded, 1 newly installed, 0 to remove and 74 not upgraded.
Need to get 54.6 MB of additional disk space will be used.
Get:1 https://aquasecurity.github.io/trivy-repo/deb jammy/main amd64 trivy amd64 0.49.1 [54.6 MB]
Fetched 54.6 MB in 1s (72.1 MB/s)
Selecting previously unselected package trivy.
(Reading database ... 66556 files and directories currently installed.)
Preparing to unpack .../trivy_0.49.1_amd64.deb ...
Unpacking trivy (0.49.1) ...
Setting up trivy (0.49.1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-13-176:~# trivy --version
Version: 0.49.1
root@ip-172-31-13-176:~#
```

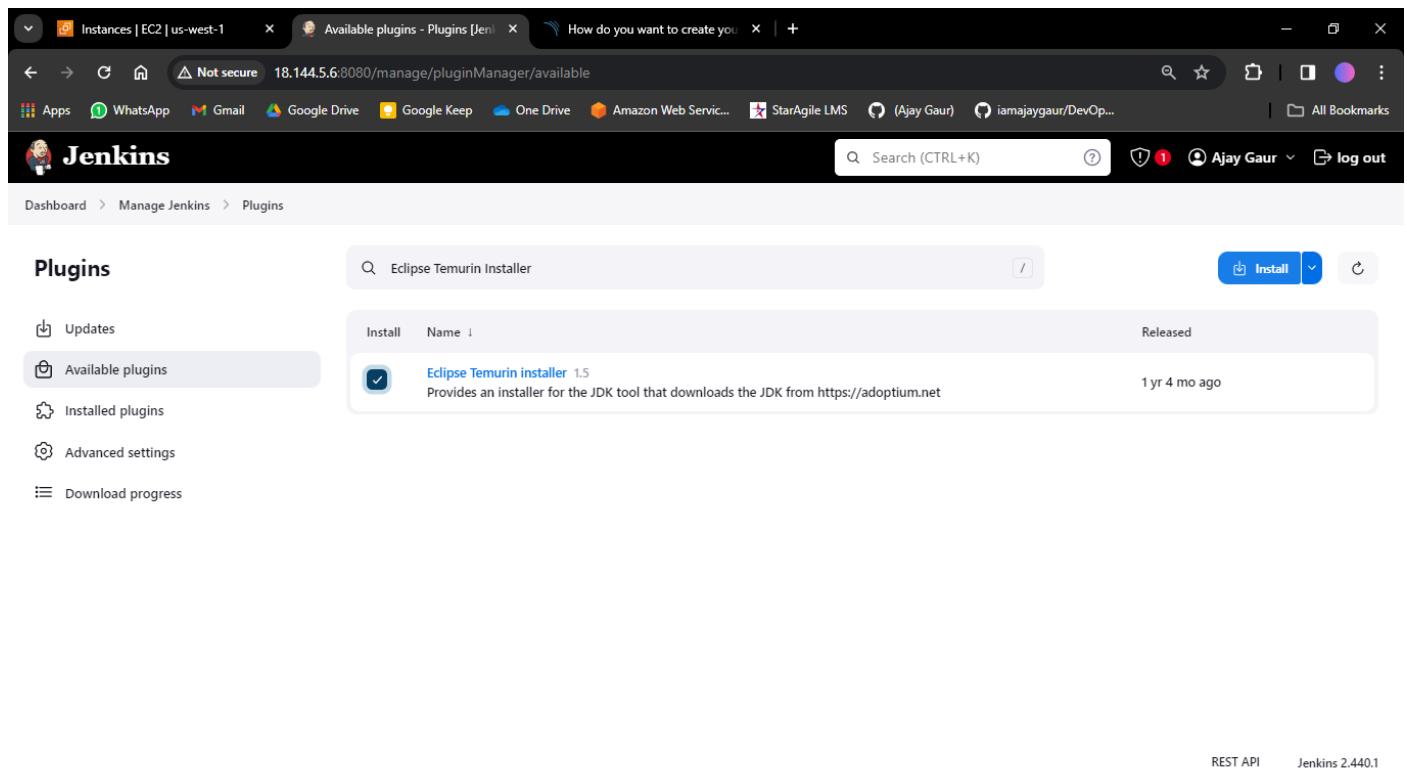
→ Next, we will log in to Jenkins and start to configure our Pipeline in Jenkins.

Step 3 - Install Plugins like JDK, Sonarqube Scanner, NodeJs, OWASP Dependency Check

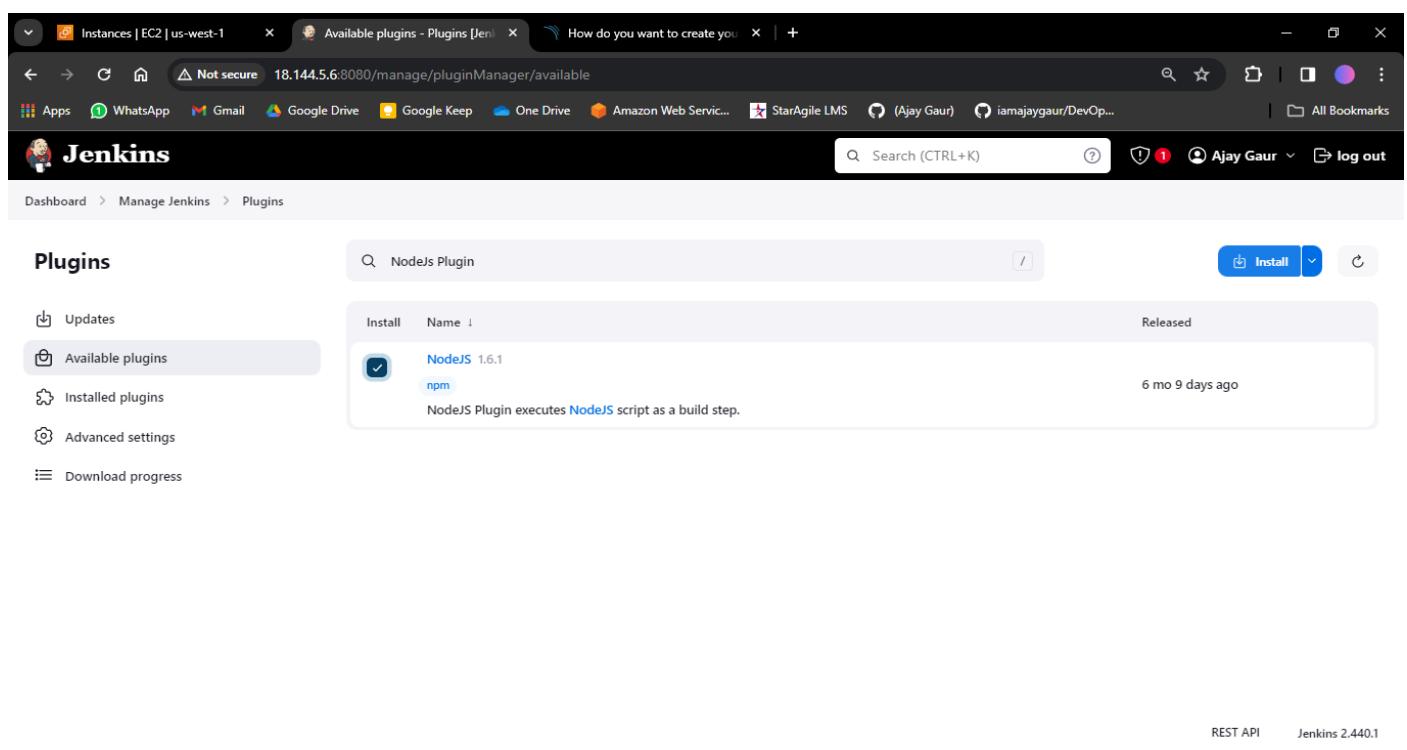
→ Goto Manage Jenkins → Plugins → Available Plugins

Install below plugins

1. Eclipse Temurin Installer (Install without restart)
2. SonarQube Scanner (Install without restart)
3. NodeJs Plugin (Install Without restart)



The screenshot shows the Jenkins plugin manager interface. The search bar at the top contains the text "Eclipse Temurin Installer". Below the search bar is a results table with one row. The table has columns for "Install", "Name", and "Released". The "Install" column has a blue "Install" button. The "Name" column shows "Eclipse Temurin installer 1.5" with a description: "Provides an installer for the JDK tool that downloads the JDK from https://adoptium.net". The "Released" column shows "1 yr 4 mo ago". On the left side of the screen, there is a sidebar with links: "Updates", "Available plugins" (which is selected and highlighted in grey), "Installed plugins", "Advanced settings", and "Download progress". At the bottom right of the screen, there are links for "REST API" and "Jenkins 2.440.1".



The screenshot shows the Jenkins plugin manager interface. The search bar at the top contains the text "NodeJs Plugin". Below the search bar is a results table with one row. The table has columns for "Install", "Name", and "Released". The "Install" column has a blue "Install" button. The "Name" column shows "NodeJS 1.6.1 npm" with a description: "NodeJS Plugin executes NodeJS script as a build step.". The "Released" column shows "6 mo 9 days ago". On the left side of the screen, there is a sidebar with links: "Updates", "Available plugins" (which is selected and highlighted in grey), "Installed plugins", "Advanced settings", and "Download progress". At the bottom right of the screen, there are links for "REST API" and "Jenkins 2.440.1".

The screenshot shows the Jenkins Plugins management interface. On the left, a sidebar lists options: Updates, Available plugins, **Installed plugins** (which is selected and highlighted in grey), Advanced settings, and Download progress. The main content area displays a search bar with the query "Sona". A table lists the "SonarQube Scanner for Jenkins" plugin, version 2.17.2, which is described as allowing an easy integration of SonarQube for code quality inspection. The plugin is marked as "Enabled" with a green toggle switch. At the bottom right of the main area, there are "REST API" and "Jenkins 2.440.1" links.

→ To Configure Java and Nodejs in Global Tool Configuration

→ Goto Manage Jenkins → Tools → Install JDK(17) and NodeJs(16) → Click on Apply and Save

The screenshot shows the Jenkins Tools configuration interface. Under the "JDK installations" section, a new entry is being added. The "Name" field contains "jdk17". The "Install automatically" checkbox is checked. Below it, under "Install from adoptium.net", the "Version" dropdown is set to "jdk-17.0.8+1". There is also an "Add Installer" button. At the bottom of the form, there are "Save" and "Apply" buttons.

Name: node16

Install automatically

Install from nodejs.org

Version: NodeJS 16.2.0

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail
 Force 32bit architecture

Global npm packages to install
Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax 'packageName@version'

Global npm packages refresh hours
Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache
72

Save Apply

→ To Create a Job

→ create a job as zomato Name, select pipeline and click on ok.

Enter an item name
zomato » Required field

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

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

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

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.

OK Multibranch Pipeline

Step 4 - Configure Sonar Server in Manage Jenkins

→ Grab the Public IP Address of your EC2 Instance, Sonarqube works on Port 9000, so <Public IP>:9000. Goto your Sonarqube Server.

Click on Administration → Security → Users → Click on Tokens and Update Token → Give it a name → and click on Generate Token

The screenshot shows the SonarQube Administration interface at the URL 18.144.5.6:9000/admin/settings. The 'Administration' tab is selected. In the top navigation bar, 'Security' is the active tab under 'Configuration'. A dropdown menu is open over the 'Users' link, showing options: General, Edit global, Groups, Global Permissions, and Permission Templates. The main content area displays the 'Cross project duplication detection' section, which is marked as 'DEPRECATED'. It explains that by default, SonarQube detects duplicates at project level, but activating the 'true' setting allows detection across projects, which increases analysis time. A toggle switch is shown as '(default)'. Below this, there are sections for 'Email' and 'SMTP host', with a note: 'For example "smtp.outlook.com". Leave blank to disable.'

→ click on update Token

The screenshot shows the SonarQube Administration interface at the URL 18.144.5.6:9000/admin/users. The 'Administration' tab is selected. In the top navigation bar, 'Users' is the active tab under 'Security'. The main content area shows a table of users. For the 'Administrator admin' user, the 'Tokens' column shows '0' and has a 'Update Tokens' button. A warning message at the bottom left states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

→ Create a token with a name and generate and copy token.

The screenshot shows the SonarQube Administration interface. A modal window titled "Tokens of Administrator" is open. It contains a "Generate Tokens" section with fields for "Name" (Enter Token Name) and "Expires in" (30 days). Below this, a message says "New token "jenkins" has been created. Make sure you copy it now, you won't be able to see it again!" with a "Copy" button next to the token ID "squ_ae6d123c0afce8737ddbecc34f2ec2bbd861c1a3". A table lists the token: "jenkins" (User type), Last used: Never, Created: February 22, 2024, Expiration: March 23, 2024. A "Revoke" button is also present. At the bottom, a note says "Embedded database should be used for evaluation purposes only" and "SonarQube™ technology is powered by SonarSource SA".

→ Goto Jenkins Dashboard → Manage Jenkins → Credentials → System → Global credentials (unrestricted) Add the Following things

Kind – Select Secret text

Secret – post the generated token

ID – sonar

Description – sonar

The screenshot shows the Jenkins Manage Jenkins interface under the Credentials section. A "New credentials" form is open. The "Kind" field is set to "Secret text". The "Scope" dropdown is set to "Global (Jenkins, nodes, items, all child items, etc)". The "Secret" field contains a masked password. The "ID" field is set to "sonar". The "Description" field is set to "sonar". At the bottom, a "Create" button is visible.

→ You will get this page once you click on create

The screenshot shows the Jenkins Global credentials (unrestricted) page. It displays a table with one row containing the following data:

ID	Name	Kind	Description
	sonar	Secret text	sonar

Below the table, there are icons for S, M, and L. At the top right, there is a blue button labeled "+ Add Credentials". At the bottom right, it says "REST API Jenkins 2.440.1".

→ Now, go to Dashboard → Manage Jenkins → System and Add like the below image.
→ Click Apply and Save

The screenshot shows the Jenkins System configuration page under SonarQube installations. It displays the following fields:

- Name:** sonar-server
- Server URL:** http://18.144.5.6:9000
- Server authentication token:** sonar

At the bottom, there are "Save" and "Apply" buttons.

- The Configure System option is used in Jenkins to configure different server. Global Tool Configuration is used to configure different tools that we install using Plugins.
- We will install a sonar scanner in the tools.
- Go to Dashboard → Manage Jenkins → Tools
- Click on Apply and Save

The screenshot shows the Jenkins 'Tools' configuration page at 18.144.5.6:8080/manage/configureTools/. The 'SonarQube Scanner installations' section is open, displaying a form to add a new scanner. The 'Name' field contains 'sonar-scanner'. The 'Install automatically' checkbox is checked. Under 'Install from Maven Central', the 'Version' dropdown is set to 'SonarQube Scanner 5.0.1.3006'. There is also an 'Add Installer' button. At the bottom are 'Save' and 'Apply' buttons.

- In the Sonarqube Dashboard add a quality gate also
- Go to Administration → Configuration → Webhooks

The screenshot shows the SonarQube Administration interface at 18.144.5.6:9000/admin/users. The 'Administration' tab is selected. The 'Webhooks' option under 'Configuration' is highlighted. The main table lists a single user: 'Administrator admin'. The table includes columns for 'SCM Accounts', 'Last connection', 'Groups', and 'Tokens'. A note at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.' The footer shows the URL 18.144.5.6:9000/admin/webhooks.

→ Click on Create

The screenshot shows the SonarQube administration interface for webhooks. The URL is <http://18.144.5.6:9000/admin/webhooks>. The page title is "Webhooks - Administration". A prominent "Create" button is located in the top right corner of the main content area. Below it, a message states "No webhook defined." A warning message in a yellow box at the bottom left says: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine." At the bottom center, it says "SonarQube™ technology is powered by SonarSource SA Community Edition - Version 9.9.4 (build 87374) - [LGPL v3](#) - [Community](#) - [Documentation](#) - [Plugins](#) - [Web API](#)".

→ Add details

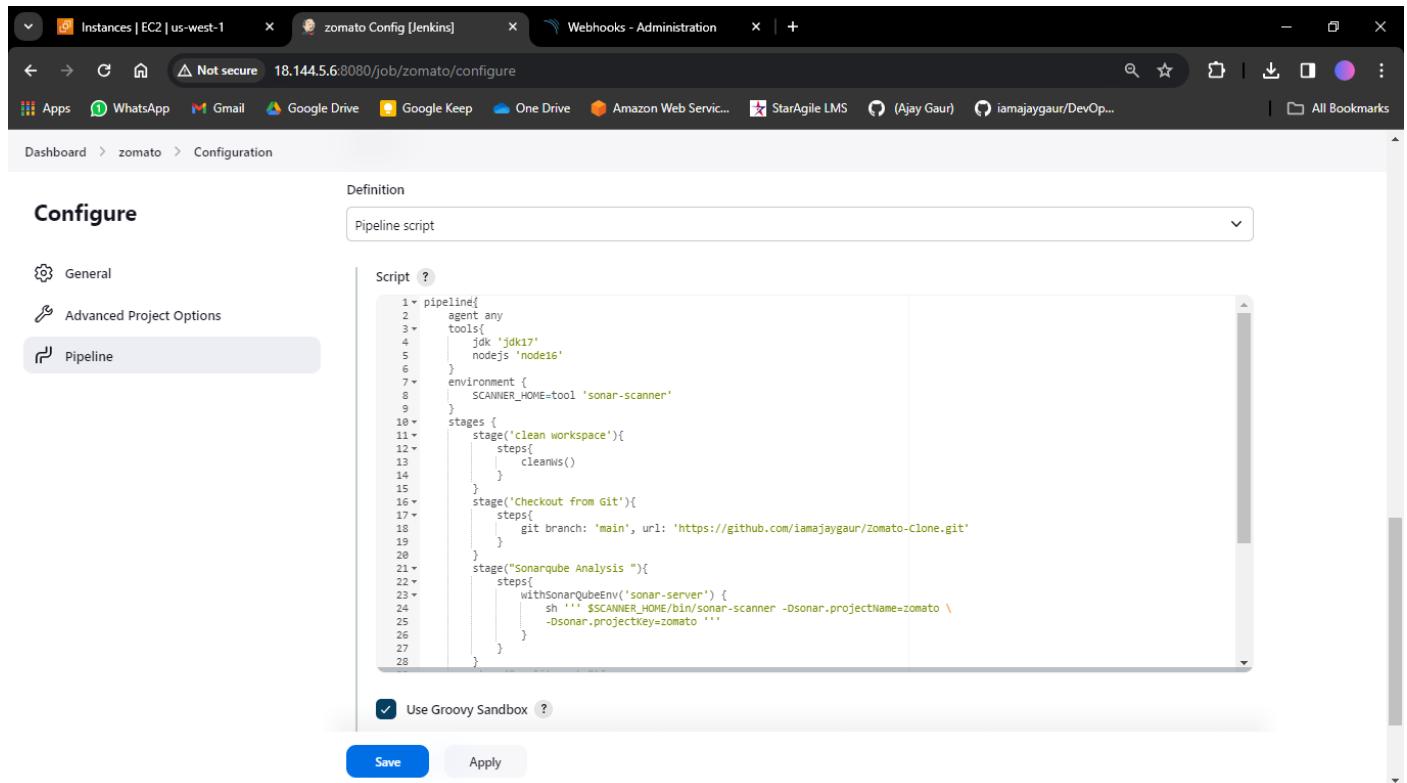
#in url section of quality gate

→ <http://<jenkins-public-ip:8080>/sonarqube-webhook/>

→ Click on Create

The screenshot shows the "Create Webhook" dialog box overlaid on the SonarQube administration interface. The dialog has fields for "Name" (set to "jenkins") and "URL" (set to "http://18.144.5.6:8080/sonarqube-webhook/"). It also includes a "Secret" field and a note about HMAC hex digests. At the bottom are "Create" and "Cancel" buttons. A yellow warning box at the bottom left of the dialog says: "All fields marked with * are required". Another yellow warning box at the very bottom of the page says: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine." The footer of the page includes links to "SonarQube™ technology is powered by SonarSource SA Community Edition - Version 9.9.4 (build 87374) - [LGPL v3](#) - [Community](#) - [Documentation](#) - [Plugins](#) - [Web API](#)".

- Let's go to our Pipeline and add the script in our Pipeline Script.
- Add pipeline script from file name jenkinsfile from github repository
- <https://github.com/iamajaygaur/zomato-DevSecOps-CICD.git>
- Go to Dashboard → Zomato → Configuration and Paste the Script
- Then Click on Apply and Save



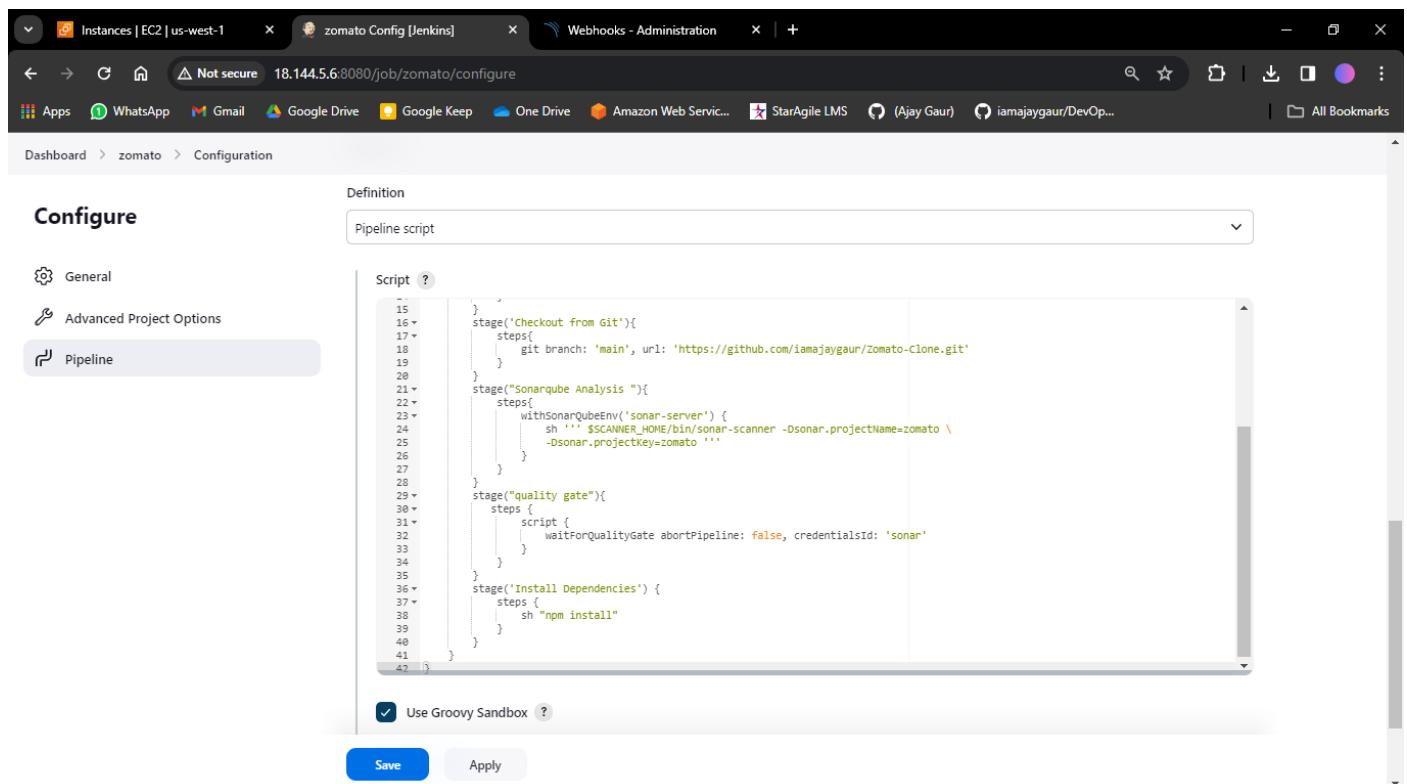
The screenshot shows the Jenkins Pipeline configuration page for a project named 'zomato'. The 'Pipeline' tab is selected. In the 'Script' section, the Jenkinsfile code is pasted:

```

1+ pipeline{
2+   agent any
3+   tools{
4+     jdk 'jdk17'
5+     nodejs 'node16'
6+   }
7+   environment {
8+     SCANNER_HOME=tool 'sonar-scanner'
9+   }
10+  stages {
11+    stage('clean workspace'){
12+      steps{
13+        cleanWs()
14+      }
15+    }
16+    stage('Checkout from Git'){
17+      steps{
18+        git branch: 'main', url: 'https://github.com/iamajaygaur/zomato-Clone.git'
19+      }
20+    }
21+    stage("Sonarqube Analysis"){
22+      steps{
23+        withSonarQubeEnv('sonar-server') {
24+          sh """$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=zomato \
25+            -Dsonar.projectKey=zomato"""
26+        }
27+      }
28+    }
29+  }
30+ }
31+
32+
33+
34+
35+
36+
37+
38+
39+
40+
41+
42+

```

Below the script, there is a checkbox labeled 'Use Groovy Sandbox' with a question mark icon. At the bottom are 'Save' and 'Apply' buttons.



The screenshot shows the Jenkins Pipeline configuration page for the same project 'zomato'. The 'Pipeline' tab is selected. The Jenkinsfile code now includes several new stages:

```

15+
16+
17+
18+
19+
20+
21+
22+
23+
24+
25+
26+
27+
28+
29+
30+
31+
32+
33+
34+
35+
36+
37+
38+
39+
40+
41+
42+

```

The new stages include 'quality_gate', 'Install Dependencies', and 'npm install'. The 'Use Groovy Sandbox' checkbox is checked. At the bottom are 'Save' and 'Apply' buttons.

→ Click on Build now, you will see the stage view like this

The screenshot shows the Jenkins interface for the 'zomato' pipeline. On the left, there's a sidebar with various options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Rename, and Pipeline Syntax. The main area is titled 'Stage View' and displays a timeline of stages: Declarative: Tool Install (2min 2s), clean workspace (489ms), Checkout from Git (4s), Sonarqube Analysis (27s), quality gate (651ms), and Install Dependencies (46s). Below this, a table provides detailed timing for each stage. A 'SonarQube Quality Gate' section indicates a 'Passed' status. At the bottom, there's a build history table showing one successful build (#1) from Feb 22 at 12:41.

→ To see the report, you can go to Sonarqube Server and go to Projects.

The screenshot shows the SonarQube interface for the 'zomato' project. The top navigation bar includes links for sonarqube, Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. The main content area shows a summary for the 'zomato' project, which is 'Passed'. It displays metrics such as Bugs (1 C), Vulnerabilities (0 A), Hotspots Reviewed (0.0% E), Code Smells (0 A), Coverage (0.0%), Duplications (0.0%), and Lines (1.3k S CSS, Java). On the left, there are filters for Quality Gate (Passed, Failed), Reliability (A-E rating), and Security (A-E rating). A note at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.' The footer includes the text: 'SonarQube™ technology is powered by SonarSource SA Community Edition - Version 9.9.4 (build 87374) - LGPL v3 - Community - Documentation - Plugins - Web API'.

You can see the report has been generated and the status shows as passed. You can see that there are 1.3k lines. To see a detailed report, you can go to issues.

Step 5 - Install OWASP Dependency Check Plugins

→ Goto Dashboard → Manage Jenkins → Plugins → OWASP Dependency-Check. Click on it and install it without restart.

The screenshot shows the Jenkins Plugin Manager interface. On the left, there's a sidebar with options like 'Updates', 'Available plugins' (which is selected), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main area has a search bar at the top with 'OWASP Dependency-Check' typed in. Below the search bar, there's a table with columns 'Install', 'Name', and 'Released'. A row for 'OWASP Dependency-Check 5.4.6' is selected, indicated by a blue checkmark icon. The details for this plugin are shown: 'Security', 'DevOps', 'Build Tools', and 'Build Reports' tabs are available. A description states: 'This plug-in can independently execute a [Dependency-Check](#) analysis and visualize results. Dependency-Check is a utility that identifies project dependencies and checks if there are any known, publicly disclosed, vulnerabilities.' The release date is '14 days ago'. At the bottom right of the main area, there are 'Install' and 'Cancel' buttons. The status bar at the bottom right shows 'REST API' and 'Jenkins 2.440.1'.

First, we configured the Plugin and next, we had to configure the Tool

→ Goto Dashboard → Manage Jenkins → Tools

→ Click Apply and Save

The screenshot shows the Jenkins Tools configuration page. The URL in the browser is '18.144.5.6:8080/manage/configureTools/'. The left sidebar shows 'Dashboard', 'Manage Jenkins', and 'Tools'. The main content area is titled 'Dependency-Check installations'. It features a 'Add Dependency-Check' button. Below it, there's a section for 'Dependency-Check' with a 'Name' field containing 'DP-Check' and an 'Install automatically' checkbox which is checked. Underneath, there's a 'Install from github.com' section with a 'Version' dropdown set to 'dependency-check 6.5.1'. There are 'Add Installer' and 'Save' buttons at the bottom. The status bar at the bottom right shows 'Page | 28'.

→ Now go configure → Pipeline and add this stage to your pipeline apply and save.

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

```

41 stage('OWASP FS SCAN') {
42     steps {
43         dependencyCheck additionalArguments: '--scan ./ --disableYarnAudit --disableNodeAudit', odcInstallation: 'DP-Check'
44         dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
45     }
46 }
47 stage('TRIVY FS SCAN') {
48     steps {
49         sh "trivy fs . > trivyfs.txt"
50     }
51 }
52 }
53

```

Below the script, there is a checked checkbox labeled 'Use Groovy Sandbox'. At the bottom are 'Save' and 'Apply' buttons.

→ Click on Build Now

→ The stage view would look like this,

The screenshot shows the Jenkins Stage View for the 'zomato' job. On the left, there is a sidebar with options: Configure, Delete Pipeline, Full Stage View, SonarQube, Rename, Pipeline Syntax, Build History, and Permalinks. The Build History section shows two builds: #2 (22 Feb 2024, 07:27) and #1 (22 Feb 2024, 07:11). Both builds are marked as 'Passed'. The Stage View table has the following data:

	Declarative: Tool Install	clean workspace	Checkout from Git	Sonarqube Analysis	quality gate	Install Dependencies	OWASP FS SCAN	TRIVY FS SCAN
#2 Feb 22 12:57	1min 1s	421ms	3s	25s	532ms	35s	9min 34s	43s
#1 Feb 22 12:41	172ms	353ms	1s	24s	413ms	23s	9min 34s	43s
	2min 2s	489ms	4s	27s	651ms (passed for 465ms)	46s		

Below the table, the SonarQube Quality Gate status is shown as 'Passed' with a green button labeled 'Success'. The Permalinks section lists the last four completed builds.

- Last build (#1), 15 min ago
- Last stable build (#1), 15 min ago
- Last successful build (#1), 15 min ago
- Last completed build (#1), 15 min ago

→ You will see that in status, a graph will also be generated and Vulnerabilities.

→ Refresh the page after build.

The screenshot shows the Jenkins Pipeline Status page for the 'zomato' project. It displays a timeline of stages with their execution times: #2 (Feb 22, 12:57) took 1min 1s, #1 (Feb 22, 12:41) took 2min 2s. Below the timeline is the 'SonarQube Quality Gate' section, which shows a green 'Passed' status with 'Success' server-side processing. A 'Latest Dependency-Check' link is also present. The 'Permalinks' section lists recent builds. At the bottom, the URL is 18.144.5.6:8080/job/zomato/lastCompletedBuild/dependency-check-findings/ and the page version is Jenkins 2.440.1.

→ Click on the Latest Dependency-Check

The screenshot shows the 'Dependency-Check Results' page for build #2. On the left, there's a sidebar with options like Status, Changes, Console Output, Edit Build Information, Delete build, Git Build Data, Dependency-Check (which is selected), Restart from Stage, Replay, Pipeline Steps, Workspaces, and Previous Build. The main area has a 'Dependency-Check Results' title and a 'SEVERITY DISTRIBUTION' chart with bars for 5, 8, and 6. Below the chart is a table of vulnerabilities:

File Name	Vulnerability	Severity	Weakness
css-what:3.4.2	OSSINDEX CVE-2022-21222	High	CWE-1333
ejs:3.1.8	OSSINDEX CVE-2023-29827	High	CWE-74
follow-redirects:1.15.2	NVD CVE-2023-26159	Medium	CWE-601
json5:1.0.1	NVD CVE-2022-46175	High	CWE-1321
jsonpointer:5.0.1	NVD CVE-2022-4742	Critical	CWE-1321
nth-check:1.0.2	NVD CVE-2021-3803	High	CWE-1333
parseuri:1.3.3	NVD CVE-2022-0722	High	CWE-200
parseuri:1.3.3	NVD CVE-2022-2216	Critical	CWE-918
parseuri:1.3.3	NVD CVE-2022-2217	Medium	CWE-79
parseuri:1.3.3	NVD CVE-2022-2218	Medium	CWE-79

Step 6 - Docker Image Build and Push

→ We need to install the Docker tool in our system in jenkins, Goto Dashboard → Manage Plugins → Available plugins → Search for Docker and install these plugins-

→ Docker

→ Docker Commons

→ Docker Pipeline

→ Docker API

→ docker-build-step

and click on install for all plugins without restart

Plugin Name	Version	Last Updated
Docker	1.6	10 days ago
Docker Commons	439.va_3cb_0a_6a_fb_29	7 mo 16 days ago
Docker Pipeline	572.v950f58993843	6 mo 14 days ago
Docker API	3.3.4-86.v39b_a_5ede342c	2 mo 25 days ago
docker-build-step	2.11	1 mo 18 days ago

→ Now, Goto Dashboard → Manage Jenkins → Tools

→ Click on Apply and Save

Docker installations

Add Docker

Docker

Name: docker

Install automatically

Download from docker.com

Docker version: latest

Add Installer

Add Docker

Save **Apply**

→ Now, Goto Dashboard → Manage Jenkins → Credentials → System → Global credentials (unrestricted)

→ Add DockerHub Username and Password under Global Credentials

→ Click on Create

New credentials [Jenkins]

Not secure 18.144.5.6:8080/manage/credentials/store/system/domain/_/newCredentials

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: iamajaygaur

Treat username as secret

Password:

ID: docker

Description: docker

Create

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
sonar	sonar	Secret text	sonar
docker	iamajaygaur/******** (docker)	Username with password	docker

Icon: S M L

REST API Jenkins 2.440.1

→ Add this stage to Pipeline Script

→ Click on Apply and Save

Definition Pipeline script

Configure

General Advanced Project Options Pipeline

```

53 ~
54 ~
55 ~
56 ~
57 ~
58 ~
59 ~
60 ~
61 ~
62 ~
63 ~
64 ~
65 ~
66 ~
67 ~
68 ~
69 ~

```

Script ? try sample Pipeline...

```

steps{
    script{
        withDockerRegistry(credentialsId: 'docker', toolName: 'docker'){
            sh "docker build -t zomato ."
            sh "docker tag zomato iamajaygaur/zomato:latest"
            sh "docker push iamajaygaur/zomato:latest"
        }
    }
}
stage("TRIVY"){
    steps{
        sh "trivy image iamajaygaur/zomato:latest > trivy.txt"
    }
}

```

Use Groovy Sandbox ?

Pipeline Syntax

Save Apply

REST API Jenkins 2.440.1

→ Build Now

→ You will see the output below, with a dependency trend.

→ Refresh the page

The screenshot shows the Jenkins interface for the 'zomato' project. On the left, there's a sidebar with options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Rename, and Pipeline Syntax. The main area is titled 'Stage View' and displays a table of build stages and their execution times. A 'Dependency-Check Trend' chart is visible on the right, showing a horizontal line at level 4. The table data is as follows:

	Declarative: Tool Install	clean workspace	Checkout from Git	Sonarqube Analysis	quality gate	Install Dependencies	OWASP FS SCAN	TRIVY FS SCAN	Docker Build & Push	TRIVY
#3	40s	395ms	2s	25s	474ms	31s	8min 14s	38s	4min 25s	1min 38s
#2	150ms	345ms	1s	23s	359ms	23s	6min 53s	33s	4min 25s	1min 38s
#1	Feb 22 13:29	No Changes								

Build History on the left shows three builds: #3 (22 Feb 2024, 07:59), #2 (22 Feb 2024, 07:27), and #1 (22 Feb 2024, 07:11). The 'Dependency-Check Trend' chart has a legend for Unassigned (grey), Low (green), Medium (yellow), High (orange), and Critical (red).

This screenshot shows the Jenkins interface for the 'zomato' project after a refresh. The 'Stage View' table data remains the same as in the previous screenshot. Below the table, the 'SonarQube Quality Gate' section shows the status as 'Passed' with a green button labeled 'Success'. The 'Permalinks' section includes a link to the latest dependency check: 'Latest Dependency-Check'.

	Declarative: Tool Install	clean workspace	Checkout from Git	Sonarqube Analysis	quality gate	Install Dependencies	OWASP FS SCAN	TRIVY FS SCAN	Docker Build & Push	TRIVY
#3	40s	395ms	2s	25s	474ms	31s	8min 14s	38s	4min 25s	1min 38s
#2	150ms	345ms	1s	23s	359ms	23s	6min 53s	33s	4min 25s	1min 38s
#1	Feb 22 12:57	No Changes								
	172ms	353ms	1s	24s	413ms	23s	9min 34s	43s		
	Feb 22 12:41	No Changes								
	2min 2s	489ms	4s	27s	651ms (paused for 46ms)	46s				

SonarQube Quality Gate
zomato Passed
server-side processing: Success

Permalinks
Latest Dependency-Check
[Permalink](#)

→ When you log in to Dockerhub, you will see a new image is created.

The screenshot shows the Dockerhub interface. At the top, there are tabs for Instances, Jenkins, Projects, and iamajaygaur/zomato general. The main navigation bar includes Dockerhub, Explore, Repositories (which is selected), Organizations, a search bar for Docker Hub, and various icons. Below the navigation, it shows the user's repositories: iamajaygaur / Repositories / zomato / General. It indicates 0 of 1 private repositories with a link to Get more. The repository details for iamajaygaur/zomato show it was updated 5 minutes ago. A 'General' tab is selected, showing a single tag: latest (OS: Alpine, Type: Image, Pulled: ..., Pushed: 5 minutes ago). To the right, under 'Docker commands', is a box containing 'docker push iamajaygaur/zomato:tagname'. Another section, 'Automated Builds', discusses connecting to GitHub or Bitbucket for automatic builds, available with Pro, Team, and Business subscriptions, with a 'Upgrade' button.

→ Now Run the container to see if the app coming up or not by adding the below stage

→ Click on Apply and Save

The screenshot shows the Jenkins configuration page for a job named zomato Config. The top navigation bar includes Instances, Jenkins, Projects, and iamajaygaur/zomato general. The main content area has tabs for General, Advanced Project Options (selected), Pipeline, and Pipeline Syntax. Under Pipeline, the 'Definition' is set to 'Pipeline script'. The script editor contains the following Groovy code:

```

stage('Deploy to container'){
    steps{
        sh 'docker run -d --name zomato -p 3000:3000 iamajaygaur/zomato:latest'
    }
}

```

A checkbox 'Use Groovy Sandbox' is checked. At the bottom, there are 'Save' and 'Apply' buttons.

→ Build Now

→ You can see the Container is deployed successfully in the below stages

The screenshot shows the Jenkins Pipeline Stage View. On the left, there's a sidebar with 'Build History' and a 'trend' dropdown. Below it is a 'Filter...' input field. The main area displays a table of build stages:

	Declarative: Tool Install	clean workspace	Checkout from Git	Sonarqube Analysis	quality gate	Install Dependencies	OWASP FS SCAN	TRIVY FS SCAN	Docker Build & Push	TRIVY	Deploy to container
Average stage times: (Average full run time: ~10min 24s)	30s	368ms	2s	25s	439ms	29s	7min 50s	36s	3min 22s	1min 45s	1s
#4 Feb 22 13:50 No Changes	162ms	288ms	1s	26s	334ms	22s	7min 2s	33s	2min 18s	1min 52s	1s
#3 Feb 22 13:29 No Changes	150ms	345ms	1s	23s	359ms	23s	6min 53s	33s	4min 25s	1min 38s	
#2 Feb 22 12:57 No Changes	172ms	353ms	1s	24s	413ms	23s	9min 34s	43s			
#1 Feb 22 12:41 No Changes	2min 2s	489ms	4s	27s	651ms (paused for 465ms)	46s					

Below the table, there's a section titled 'SonarQube Quality Gate' with status indicators for 'zomato' (Passed) and 'server-side processing' (Success). A 'Latest Dependency-Check' badge is also present.

→ You can see now the Zomato app is deployed Successfully

→ <http://<Jenkins-public-ip>:3000>

→ You will get this output

The screenshot shows the Zomato React App homepage. The background features various food images. At the top, there's a navigation bar with links for 'Investor Relations', 'Add restaurant', 'Log in', and 'Sign up'. Below the navigation is a search bar with the placeholder 'Chennai' and a dropdown menu. The main heading 'zomato' is prominently displayed, followed by the tagline 'Discover the best food & drinks in Patna'. Below the heading, there's a search bar with the placeholder 'Chennai' and a dropdown menu. At the bottom, there are three cards: 'Order Online', 'Nightlife and Clubs', and 'Dinning'.

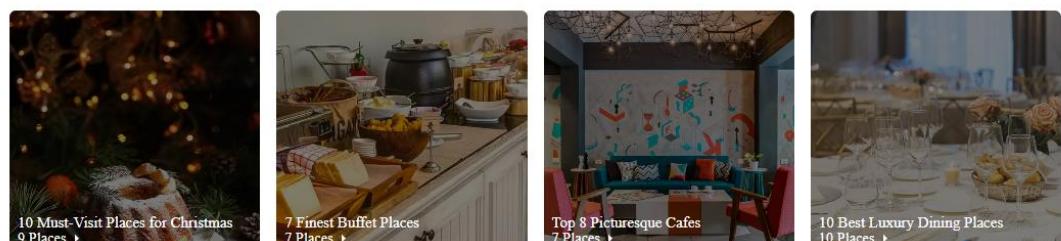
The screenshot shows a grid of three cards:

- Order Online**: Stay home and order to your doorstep.
- Nightlife and Clubs**: Explore the city's top nightlife outlets.
- Dining**: Views the city's favourite venues.

Collection

Explore curated lists of top restaurants, cafes, pubs, and bars in Ahmedabad, based on trends

[All collection in Ahmedabad](#)



Popular localities in and around Ahmedabad

Bodakdev 345 Places	Setellite 336 Places	Gurukul 83 Places	Navrangpura 302 Places
Vastrapur 217 Places	Thaltej 222 Places	Prahalad Nagar 181 Places	C G Road 94 Places

[See more](#)

Get the Zomato app

We will send you a link, open it on your phone to download the app

Email Phone

[Share App Link](#)

Download app from

[GET IT ON](#) [Download on the](#)

The screenshot shows a search interface with three dropdown menus:

- Popular cuisines near me**
- Popular restaurant types near me**
- Top Restaurant Chains**

Below this, there is a link to the **Zomato** website.

Zomato

ABOUT ZOMATO

- Who We Are
- Blog
- Work With Us
- Investor Relations
- Report Fraud
- Contact Us

ZOMAVERSE

- Zomato
- Blinkit
- Feeding India
- HyperPure
- Zomaland

FOR RESTAURANTS

- Partner With Us
- Apps For You
- For Enterprises
- Zomato For Work

LEARN MORE

- Privacy
- Security
- Terms
- Sitemap

SOCIAL LINKS

- India
- English
- LinkedIn
- Twitter
- Instagram
- YouTube
- Facebook

Download on the App Store

GET IT ON Google Play

Step - 8: Terminate instances.

The screenshot shows the AWS EC2 Instances page. A green notification bar at the top says "Successfully terminated i-044f7b2215c8b20ea".

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
zomato-app-clone	i-044f7b2215c8b20ea	Shutting-down	t2.large	2/2 checks passed	View alarms	us-west-1b

Instance: i-044f7b2215c8b20ea (zomato-app-clone)

Details

Instance ID	Public IPv4 address	Private IPv4 addresses
i-044f7b2215c8b20ea (zomato-app-clone)	18.144.5.6 [open address]	172.31.13.176
IPv6 address	Instance state	Public IPv4 DNS
-	Shutting-down	ec2-18-144-5-6.us-west-1.compute.amazonaws.com [open address]
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-13-176.us-west-1.compute.internal	ip-172-31-13-176.us-west-1.compute.internal	-
Answer private resource DNS name	Instance type	
IPv4 (A)	t2.large	