

Name: - B. Narasimhappa

Insurance Domain
Project-3

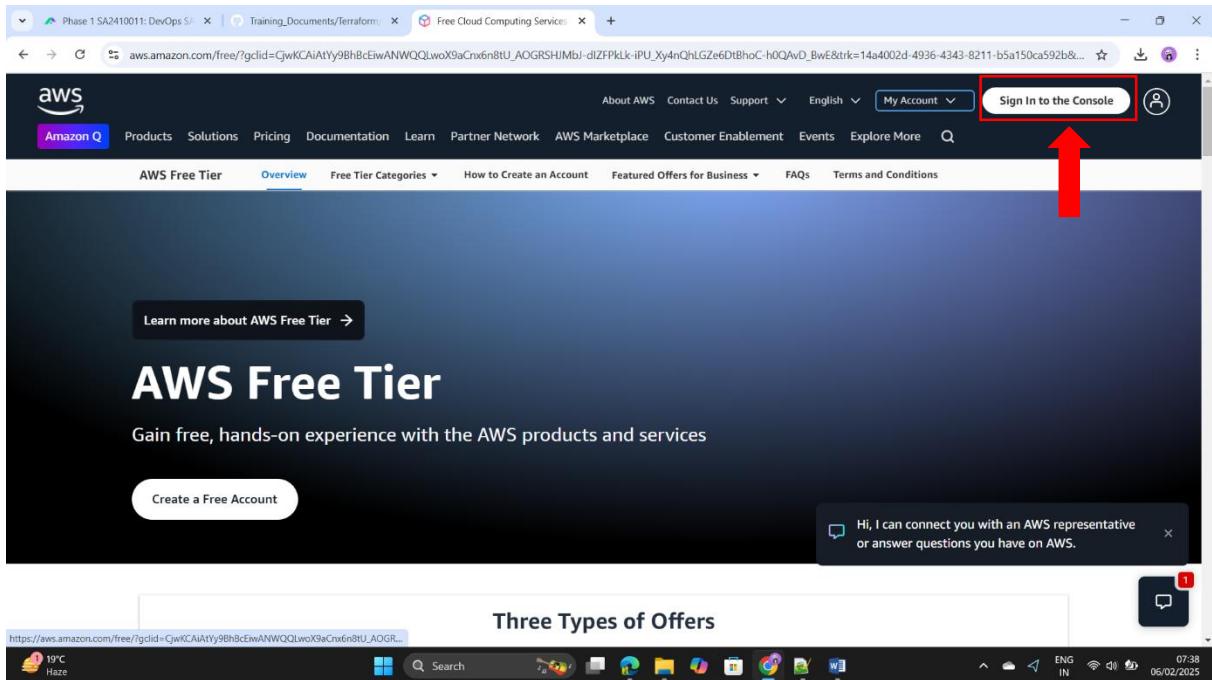
Certification Project – Insure Me

Date of submission: -

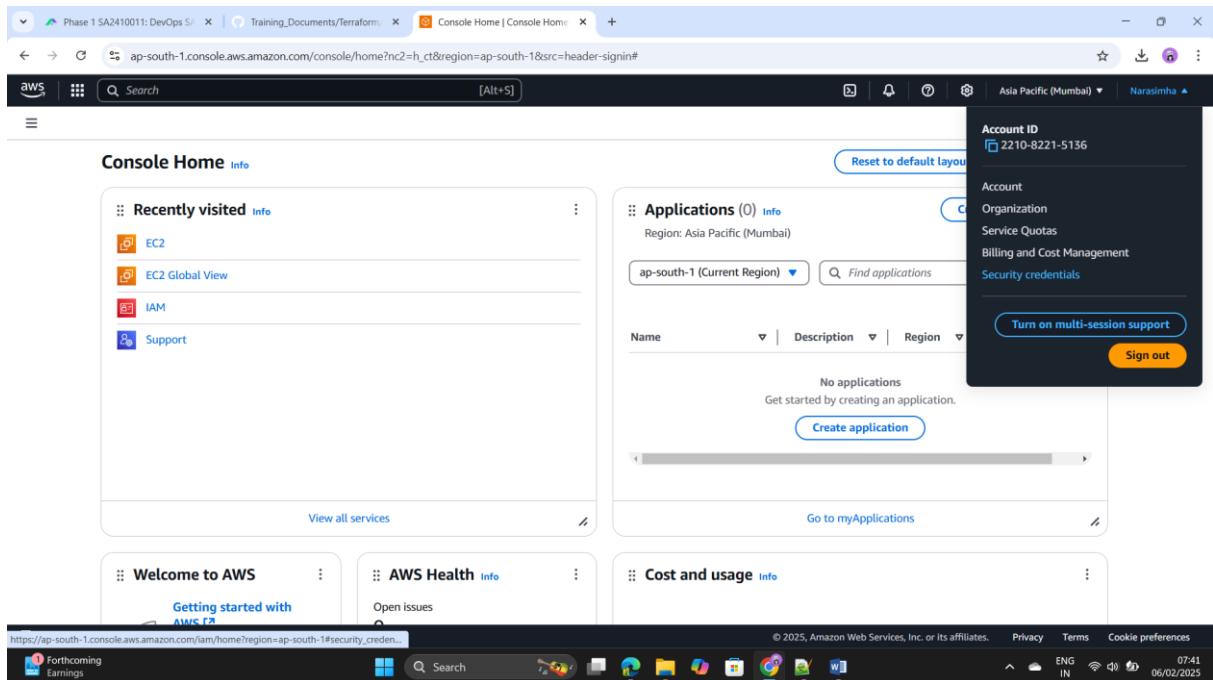
Date of resubmission: -

Submitted by: -B. Narasimhappa

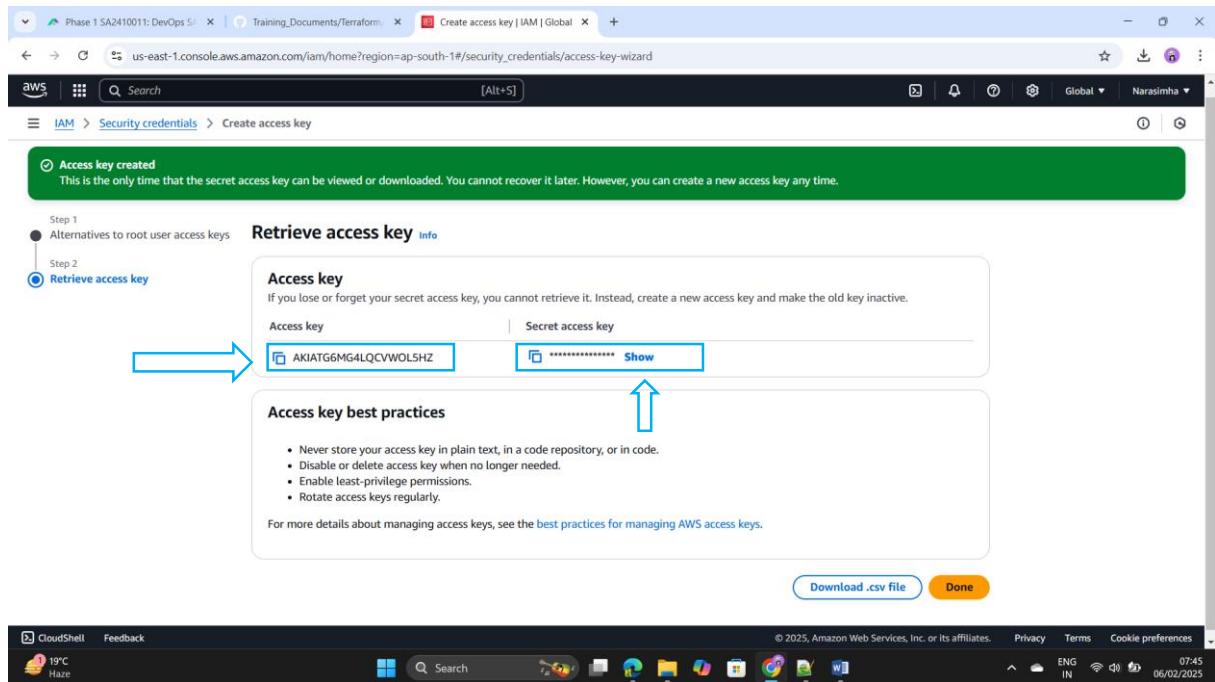
Step1: - Login to the AWS console.



Step2: - Go to the Security Credentials tab.



Step3: - Click on the Create the Access key and create the access key save the access key and secrete key.



Step4: - Open the Visual Studio code and paste these access key secrete key there and launch the 6 instance by using terraform.

```

file1.tf > ...
64 resource "aws_security_group" "mysg9" {
93   tags = {
94     Name = "mysg9"
95   }
96 }
97
98 # Create Instance
99
100 resource "aws_instance" "instance9" {
101   ami             = "ami-0198a868663199764"
102   instance_type  = "t2.micro"
103   associate_public_ip_address = true
104   subnet_id      = aws_subnet.mysubnet9.id <- #82-95 resource "aws_instance" "instance1"
105   vpc_security_group_ids = [aws_security_group.mysg9.id]
106   key_name        = "project1"
107   count           = 6
108
109   tags = {
110     Name = "Dummy_Server0"
111   }
112 }
113

```

The screenshot shows the Terraform Cloud IDE interface. The left sidebar displays the project structure with files like .terraform, providers, LICENSE.txt, terraform-provider-aws_v5.84.0_x5.exe, and terraform.lock.hcl. The main editor window contains the Terraform configuration file (file1.tf) which defines an AWS security group (mysg9) and an AWS instance (instance9). The instance is configured with specific AMI, instance type, subnet, and security group details. The bottom status bar shows the file is in 113 columns, 2 spaces, UTF-8 encoding, and was last modified on 06/02/2025 at 07:54.

Step5: - Execute the Code by using commands as terraform init and terraform plan.

The screenshot shows the Terraform Cloud IDE with the terminal tab active. The command 'terraform init' is being run, and the output indicates that the backend is being initialized and provider plugins are being initialized. It also notes that previous versions of HashiCorp/AWS were used from the dependency lock file and that previously-installed versions of HashiCorp/AWS v5.84.0 were used. The message 'Terraform has been successfully initialized!' is displayed. The bottom status bar shows the file is in 113 columns, 2 spaces, UTF-8 encoding, and was last modified on 06/02/2025 at 07:57.

```

file1.tf
resource "aws_security_group" "mysg9" {
  tags = {
    Name = "MySQL"
  }
}

resource "aws_instance" "instance9" {
  ami           = "ami-0198a868663199764"
  instance_type = "t2.micro"
  associate_public_ip_address = true
  subnet_id = aws_subnet.mysubnet9.id
  vpc_security_group_ids = [aws_security_group.mysg9.id]
  key_name = "project1"
  count = 6

  tags = {
    Name = "MySQL Server"
  }
}

```

Step6: - Use the Terraform apply command the instances has been created successfully.

```

aws_instance.instance9[1]: Creating...
aws_instance.instance9[3]: Still creating... [10s elapsed]
aws_instance.instance9[0]: Still creating... [10s elapsed]
aws_instance.instance9[4]: Still creating... [10s elapsed]
aws_instance.instance9[2]: Still creating... [10s elapsed]
aws_instance.instance9[5]: Still creating... [10s elapsed]
aws_instance.instance9[1]: Still creating... [10s elapsed]
aws_instance.instance9[3]: Creation complete after 14s [id=1-0ea711954e064dec2]
aws_instance.instance9[1]: Creation complete after 14s [id=1-02819c99b351e2b1]
aws_instance.instance9[0]: Creation complete after 14s [id=1-074c566227baa36d]
aws_instance.instance9[2]: Creation complete after 14s [id=1-0305553e75edc6e0]
aws_instance.instance9[4]: Creation complete after 14s [id=1-00a6c3f87fefed4a5]
aws_instance.instance9[5]: Creation complete after 14s [id=1-069b5725799afed7b]

Apply complete! Resources: 12 added, 0 changed, 0 destroyed.

```

The screenshot shows the AWS EC2 Instances page with the following details:

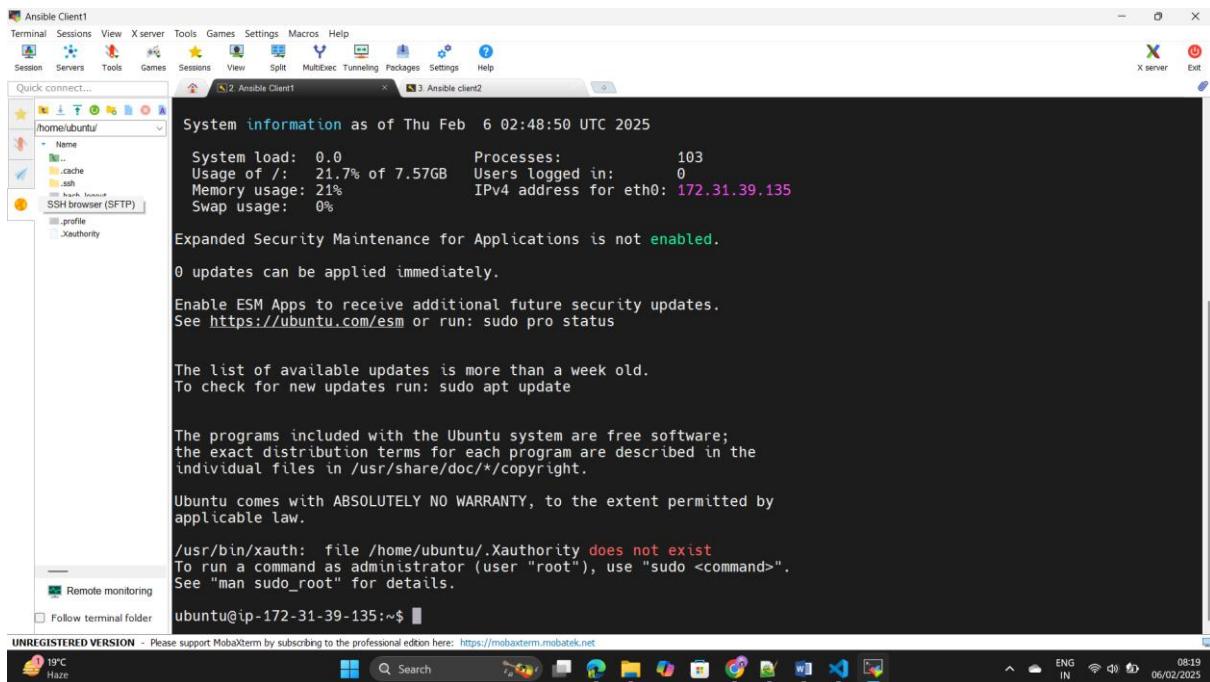
- Instances (6/6) Info:** Last updated less than a minute ago.
- Filters:** Instance state = running, Clear filters.
- Columns:** Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IP.
- Instances:**
 - Dummy_Server0 (i-069b5725799afed/b): Running, t2.micro, Initializing, View alarms, ap-southeast-1a
 - Dummy_Server0 (i-09a6c3fb7fe7fed4a5): Running, t2.micro, Initializing, View alarms, ap-southeast-1a
 - Dummy_Server0 (i-0305553e75e2dc6e0): Running, t2.micro, Initializing, View alarms, ap-southeast-1a
 - Dummy_Server0 (i-02819ce99b351e2b1): Running, t2.micro, Initializing, View alarms, ap-southeast-1a
 - Dummy_Server0 (i-074c566227baa636e): Running, t2.micro, Initializing, View alarms, ap-southeast-1a
 - Dummy_Server0 (i-08e711954e064dec2): Running, t2.micro, Initializing, View alarms, ap-southeast-1a
- Monitoring:** Configure CloudWatch agent.
- CloudShell:** Feedback.

Step7: - Update the security inbounds as enable the port 8080 & allow all the traffic in the all the instances.

The screenshot shows the AWS ModifyInboundSecurityGroup page with the following details:

- EC2 > Security Groups > sg-00ffda60a98d99c5c - default > Edit inbound rules**
- Inbound Rules:**
 - All TCP: TCP, 0 - 65535, Anywhere, 0.0.0.0/0, Delete.
 - Custom TCP: TCP, 8080, Anywhere, 0.0.0.0/0, Delete.
 - HTTPS: TCP, 443, Anywhere, 0.0.0.0/0, Delete.
 - HTTP: TCP, 80, Anywhere, 0.0.0.0/0, Delete.
 - SSH: TCP, 22, Anywhere, 0.0.0.0/0, Delete.
- Add rule:** Add a new rule.
- Warning:** Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
- Buttons:** Cancel, Preview changes, Save rules.
- CloudShell:** Feedback.

Step8: - Connect the 2 ansible client nodes by using MobaXterm.



Ansible Client1

System information as of Thu Feb 6 02:48:50 UTC 2025

```

System load: 0.0      Processes: 103
Usage of /: 21.7% of 7.57GB  Users logged in: 0
Memory usage: 21%      IPv4 address for eth0: 172.31.39.135
Swap usage: 0%

```

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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.

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

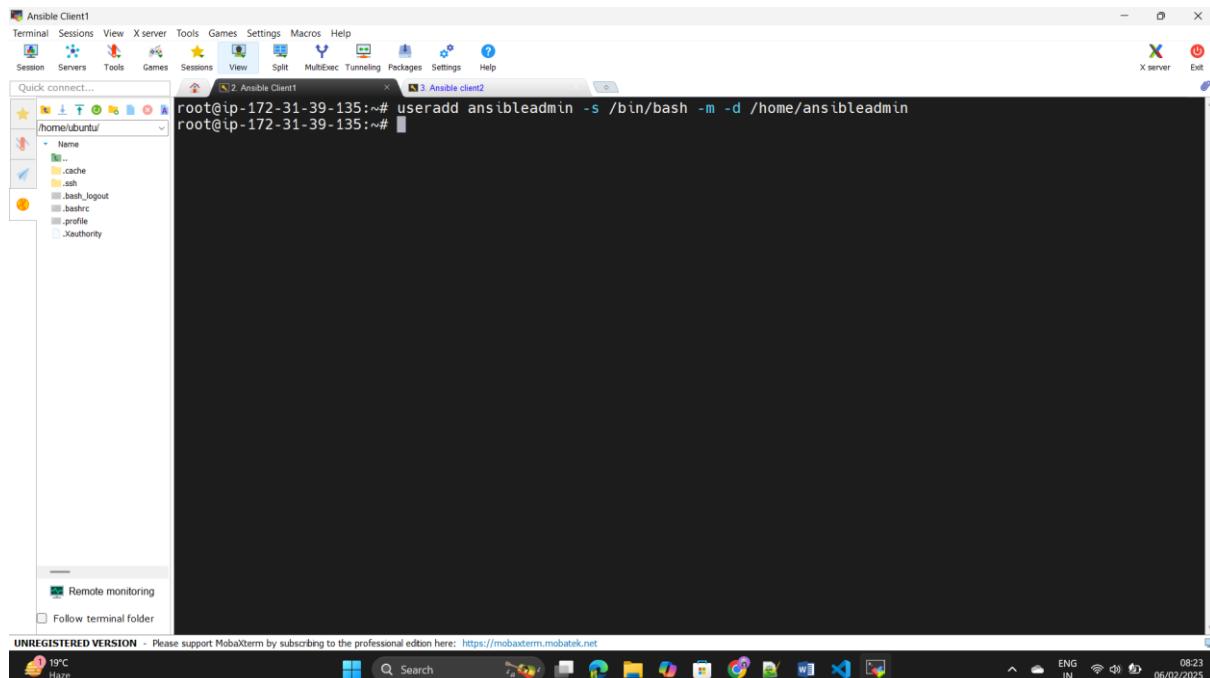
ubuntu@ip-172-31-39-135:~\$

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

19°C Haze

Step9: - Create the user in the 2 instances by using command as

```
useradd ansibleadmin -s /bin/bash -m -d /home/ansibleadmin
```



Ansible Client1

root@ip-172-31-39-135:~# useradd ansibleadmin -s /bin/bash -m -d /home/ansibleadmin

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

19°C Haze

Step10: - Edit the “/etc/ssh/sshd_config” file and enable the password authentication to yes and save the file.

```
# PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no
```

Step11: - Edit the Visudo-file and add the user to give access to the root level.

```
GNU nano 6.2                               /etc/sudoers.tmp *
# Completely harmless preservation of a user preference.
Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_"

# Per-user preferences; root won't have sensible values for them.
Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification
# User alias specification
# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
ansibleadmin ALL=(ALL) NOPASSWD: ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

Step12: - Go to the Ansibleadmin user and use the command ls -a ,we can see the root level access is there for the user.

```
root@ip-172-31-39-135:~# su - ansibleleadadmin
ansibleleadadmin@ip-172-31-39-135:~$ ls -a
. .. .bash_history .bash_logout .bashrc .profile
ansibleleadadmin@ip-172-31-39-135:~$
```

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

Step13: - Login to the Ansible_controller instance and update the instance.

```
ubuntu@ip-172-31-42-118:~$ sudo apt update -y
Hit:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:6 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2313 kB]
Get:12 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [387 kB]
Get:13 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2940 kB]
Get:14 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [515 kB]
Get:15 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1185 kB]
Get:16 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [290 kB]
Get:17 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [26.4 kB]
Get:18 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [44.5 kB]
Get:19 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [11.5 kB]
Get:20 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [440 B]
Get:21 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [67.7 kB]
Get:22 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.1 kB]
Get:23 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:24 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:25 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [30.0 kB]
Get:26 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.6 kB]
Get:27 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [672 B]
Get:28 http://ap-southeast-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 [2077 kB]
84% [29 Packages 1039 kB/2077 KB 50%]
```

4735 kB/s 1s

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

Step14: - Go to the ansible installation page on browser and copy and paste the commands from page to here.

Ansible Community Documentation

Please file a bug against the [Fedora EPEL](#) product in Red Hat Bugzilla to reach the package maintainers.

Installing Ansible on OpenSUSE Tumbleweed/Leap

```
$ sudo zypper install ansible
```

See [OpenSUSE Support Portal](#) for additional help with Ansible on OpenSUSE.

Installing Ansible on Ubuntu

Ubuntu builds are available in a PPA here.

To configure the PPA on your system and install Ansible run these commands:

```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo add-apt-repository --yes --update ppa:ansible/ansible
$ sudo apt install ansible
```

Note
On older Ubuntu distributions, "software-properties-common" is called "python-software-properties". You may want to use `apt-get` rather than `apt` in older versions. Also, be aware that only newer distributions (that is, 18.04, 18.10, and later) have a `-u` or `--update` flag. Adjust your script as needed.

File any issues in the PPA's issue tracker.

Installing Ansible on Debian

Step15: - Execute the above commands in the Ansible Controller machine.

```
root@ip-172-31-42-118:~# sudo apt update
[sudo] password for root:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.22.9).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 32 not upgraded.
Hit:1 http://deb.launchpadcontent.net/ansible/ubuntu jammy main
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications— automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/
If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository...
Adding deb entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding key to /etc/apt/trusted.gpg.d/ansible-ubuntu-ansible.gpg with fingerprint 6125E2A8C77F2818FB7BD15B93C4A3FD7BB9C367
Hit:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 https://ppa.launchpadcontent.net/ansible/ubuntu jammy InRelease [18.0 kB]
Get:6 https://ppa.launchpadcontent.net/ansible/ubuntu jammy/main amd64 Packages [1120 B]
Get:7 https://ppa.launchpadcontent.net/ansible/ubuntu jammy/main Translation-en [752 B]
Fetched 19.9 kB in 2s (10.8 kB/s)
```

Step16: - Create the one user and create the SSH key in the user.

The screenshot shows the MobaXterm interface with four windows open:

- Session 1:** Shows the file structure of the Ansible controller node, specifically navigating to the .ssh directory.
- Session 2 (Ansible Client):** A terminal window where the command `ssh-keygen -t ecdsa -b 521` is run. The output shows the key generation process, including prompts for saving the key and entering a passphrase.
- Session 3 (Ansible client):** A terminal window showing the generated public key file `id_ecdsa.pub`.
- Session 4 (Ansible_controller):** A terminal window showing the command `chmod 600 /home/devopsadmin/.ssh/*` being run to change the access mode of the SSH files.

At the bottom, the Windows taskbar shows the date as 06/02/2025 and the time as 08:49.

Step17: - Paste the id_ecdsa.pub into the authorized_keys & Change the access mod by using chmod 600 command.

The screenshot shows the MobaXterm interface with four windows open:

- Session 1:** Shows the file structure of the Ansible controller node.
- Session 2 (Ansible Client):** A terminal window where the command `cat id_ecdsa.pub > authorized_keys` is run to append the public key to the authorized_keys file.
- Session 3 (Ansible client):** A terminal window showing the command `ls` which lists the contents of the .ssh directory, including the authorized_keys file.
- Session 4 (Ansible_controller):** A terminal window showing the command `chmod 600 /home/devopsadmin/.ssh/*` being run to change the access mode of the SSH files.

At the bottom, the Windows taskbar shows the date as 06/02/2025 and the time as 08:52.

Step18: - Copy the Authorized_key and paste in the Ansible client node.

```

Ansible_controller
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
1. Ansible Client1 2. Ansible Client2 3. Ansible client2 4. Ansible_controller
/home/ubuntu/
+-- Name
  +-- .cache
  +-- .ssh
  +-- .bash_logout
  +-- .bashrc
  +-- .profile
  +-- .Xauthority

devopsadmin@ip-172-31-42-118:~/ssh$ cat authorized_keys
ecdsa-sha2-nistp521 AAAAE2VjZHNhLXNoYTItbmlzdHA1MjEAAACFBAG5eX1pDR9hGTff6sLJqtcp/HRW1RWj/m4jkRzyq8NsSHHdZXQ07
HjzSNpBDHWgpTd2/XNUecFBp3o3lrZySBaCdqATGfVZhGCoCJNhVX2o14GCP4mDgZlVowPRTXp37RxsnJNv+p0P33f7fK0L84yZ0B5j32yHgtxb5iLKGriHWtm5w
== devopsadmin@ip-172-31-42-118
devopsadmin@ip-172-31-42-118:~/ssh$ ^C
devopsadmin@ip-172-31-42-118:~/ssh$ ^C

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
20°C Haze
Ansible Client1
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
1. Ansible Client1 2. Ansible Client2 3. Ansible client2 4. Ansible_controller
/home/ubuntu/
+-- Name
  +-- .cache
  +-- .ssh
  +-- .bash_logout
  +-- .bashrc
  +-- .profile
  +-- .Xauthority

root@ip-172-31-39-135:~# su - ansibleleadadmin
ansibleleadadmin@ip-172-31-39-135:~$ mkdir .ssh
ansibleleadadmin@ip-172-31-39-135:~$ vi authorized_keys
ansibleleadadmin@ip-172-31-39-135:~$ vi authorized_keys
ansibleleadadmin@ip-172-31-39-135:~$ cat authorized_keys
ecdsa-sha2-nistp521 AAAAE2VjZHNhLXNoYTItbmlzdHA1MjEAAACFBAG5eX1pDR9hGTff6sLJqtcp/HRW1RWj/m4jkRzyq8NsSHHdZXQ07HjzSNpBDHWgpTd2/XNUecFBp3o3lrZySBaCdqATGfVZhGCoCJNhVX2o14GCP4mDgZlVowPRTXp37RxsnJNv+p0P33f7fK0L84yZ0B5j32yHgtxb5iLKGriHWtm5w== devopsadmin@ip-172-31-42-118
ansibleleadadmin@ip-172-31-39-135:~$ ^C
ansibleleadadmin@ip-172-31-39-135:~$ ^C

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
20°C Haze

```

Step19: - Connect the 2 client nodes by using command as

ssh ansibleleadadmin@ 172.31.39.135

ssh ansibleleadadmin@ 172.31.39.68

```

devopsadmin@ip-172-31-42-118:~/ssh$ ssh ansibleadmin@172.31.39.135
ssh ansibleadmin@172.31.39.135
The authenticity of host '172.31.39.135 (172.31.39.135)' can't be established.
ED25519 key fingerprint is SHA256:wdkfk1Ly4CC4A7abSZApxP9objWLoAmNh0lnofYazti.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.39.135' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

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

System information as of Thu Feb 6 06:03:53 UTC 2025

System load: 0.0 Processes: 115
Usage of /: 25.6% of 7.57GB Users logged in: 1
Memory usage: 23% IPv4 address for eth0: 172.31.39.135
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

31 updates can be applied immediately.
24 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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

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

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

```

Step20: - Edit the etc/ansible/host and add the client nodes in there as build server and Jenkins server.

```

devopsadmin@ip-172-31-42-118:/etc/ansible$ ls
ansible.cfg hosts roles
devopsadmin@ip-172-31-42-118:/etc/ansible$ cat hosts
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group:

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:
## www[001:006].example.com

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

```

```
devopsadmin@ip-172-31-42-118:/etc/ansible$ cat hosts
[node1]
samplenode1 ansible_ssh_host=172.31.39.68 ansible_ssh_user=ansibleleadmin

[node2]
samplenode2 ansible_ssh_host=172.31.39.135 ansible_ssh_user=ansibleleadmin
devopsadmin@ip-172-31-42-118:/etc/ansible$
```

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

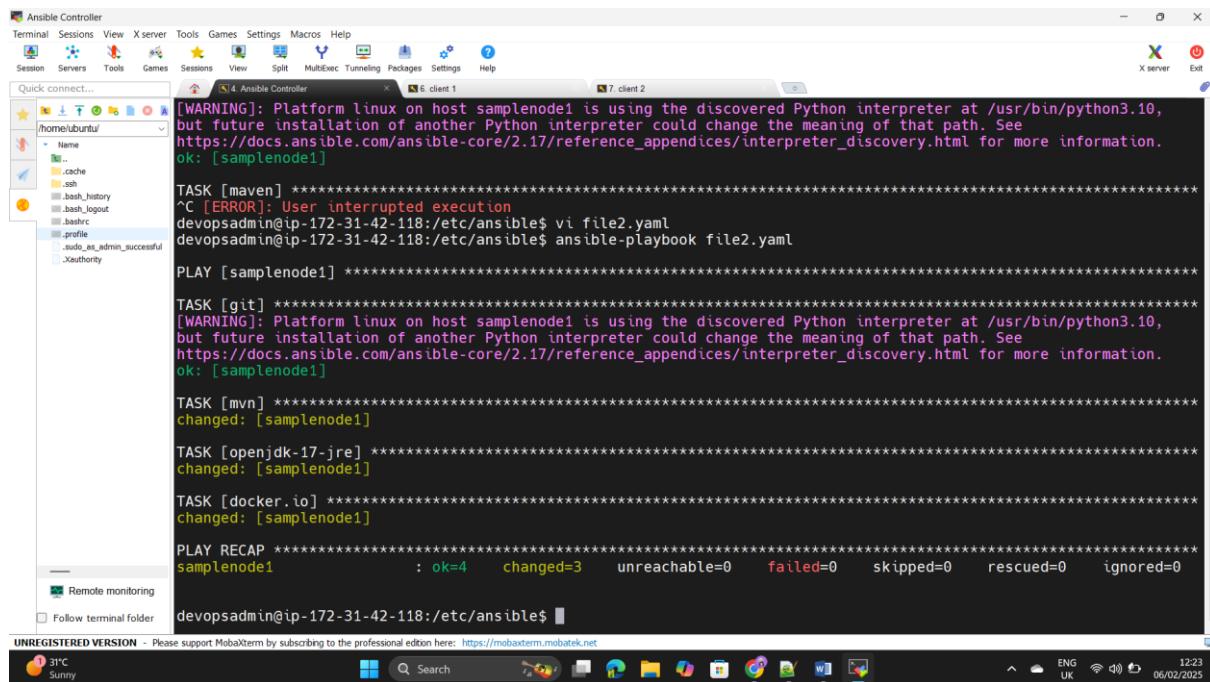
Step21: - install the tools what we want in the node 1 as a build server of Jenkins install jdk,maven,git,Docker by using playbook.

```
devopsadmin@ip-172-31-42-118:/etc/ansible$ cat hosts
[node1]
samplenode1 ansible_ssh_host=172.31.39.68 ansible_ssh_user=ansibleleadmin

[node2]
samplenode2 ansible_ssh_host=172.31.39.135 ansible_ssh_user=ansibleleadmin
devopsadmin@ip-172-31-42-118:/etc/ansible$ ls
ansible.cfg hosts roles
devopsadmin@ip-172-31-42-118:/etc/ansible$ vi file1.yaml
devopsadmin@ip-172-31-42-118:/etc/ansible$ cat file1.yaml
---
- hosts: node1
  become: yes
  tasks:
    - apt:
        name: openjdk-17-jre
        state: present
    - apt:
        name: git
        state: present
    - apt:
        name: maven
        state: present
    - apt:
        name: docker.io
        state: present
---
```

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

Step22: - Execute the file2.yaml file the needed build tools has been configured the build server.



```
[WARNING]: Platform linux on host samplenode1 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [samplenode1]

TASK [maven] ****
[WARNING]: Platform linux on host samplenode1 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [samplenode1]

TASK [git] ****
changed: [samplenode1]

TASK [openjdk-17-jre] ****
changed: [samplenode1]

TASK [docker.io] ****
changed: [samplenode1]

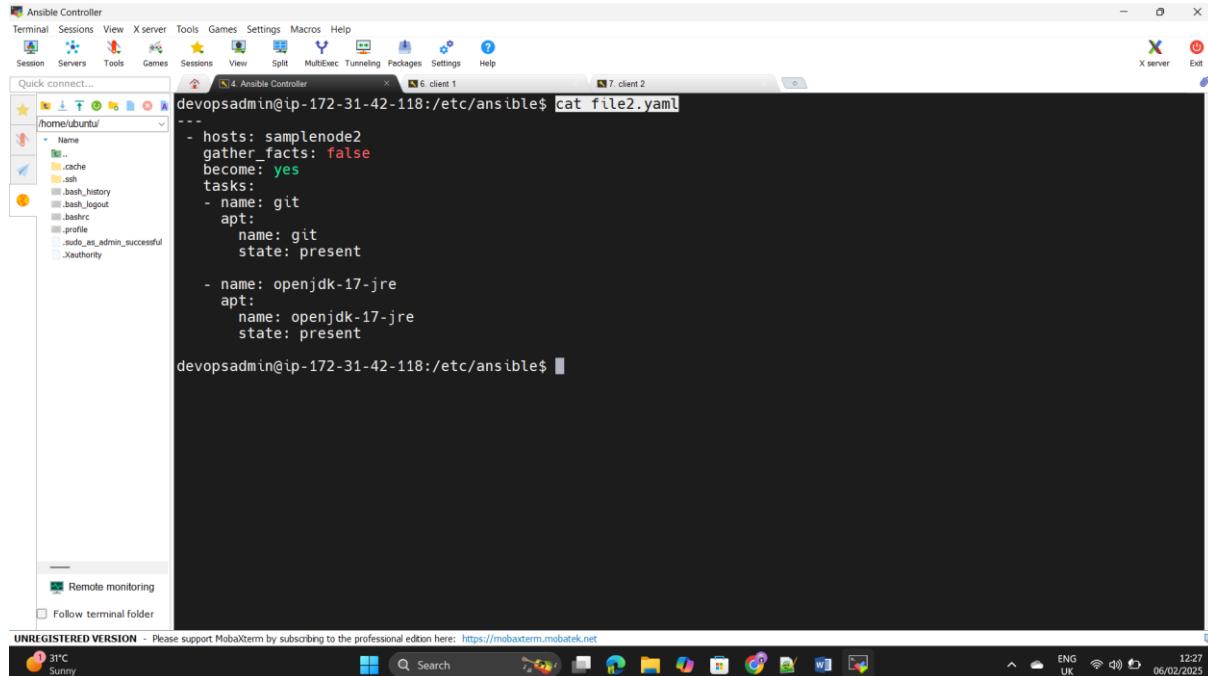
PLAY RECAP ****
samplenode1 : ok=4    changed=3    unreachable=0   failed=0    skipped=0   rescued=0   ignored=0

devopsadmin@ip-172-31-42-118:/etc/ansible$
```

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

31°C Sunny 12:23 ENG UK 06/02/2025

Step23: - Same like as install git and java in the Jenkins Server (create the yaml file and execute via playbooks).



```
devopsadmin@ip-172-31-42-118:/etc/ansible$ cat file2.yaml
---
- hosts: samplenode2
  gather_facts: false
  become: yes
  tasks:
    - name: git
      apt:
        name: git
        state: present

    - name: openjdk-17-jre
      apt:
        name: openjdk-17-jre
        state: present

devopsadmin@ip-172-31-42-118:/etc/ansible$
```

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

31°C Sunny 12:27 ENG UK 06/02/2025

Step24: - Let's get into the 2 instance and check the tools are installed or not.

```

root@ip-172-31-39-135:~# git --version
git version 2.34.1
root@ip-172-31-39-135:~# java --version
openjdk 17.0.14 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-122.04.1)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-122.04.1, mixed mode, sharing)
root@ip-172-31-39-135:~# 

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

root@ip-172-31-39-68:~# docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~22.04.1
root@ip-172-31-39-68:~# git --version
git version 2.34.1
root@ip-172-31-39-68:~# java --version
openjdk 17.0.14 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-122.04.1)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-122.04.1, mixed mode, sharing)
root@ip-172-31-39-68:~# mvn --version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 17.0.14, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1021-aws", arch: "amd64", family: "unix"
root@ip-172-31-39-68:~# 

```

Step25: - Configuring of Build tools and Jenkins tools has been configured successfully by using ansible.

Step26: - Install the Jenkins by using the url<
<https://www.jenkins.io/doc/book/installing/linux/>>.

Debian/Ubuntu

On Debian and Debian-based distributions like Ubuntu you can install Jenkins through apt.

Long Term Support release

A LTS (Long-Term Support) release is chosen every 12 weeks from the stream of regular releases as the stable release for that time period. It can be installed from the [debian-stable apt repository](#).

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

Weekly release

A new release is produced weekly to deliver bug fixes and features to users and plugin developers. It can be installed from the [debian apt repository](#).

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] * \
https://pkg.jenkins.io/debian binary/" | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
```

Step27: - Execute the above commands in the client_server1 as Jenkins masternode.

```
root@ip-172-31-39-135:~# sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
--2025-02-06 07:17:47-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.46.133, 2a04:4e42:48::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.46.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins- 100%[=====] 3.10K --KB/s   in 0s

2025-02-06 07:17:47 (60.7 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]

root@ip-172-31-39-135:~#
```

Step28: - Enter the command as – “sudo apt-get install jenkins -y”

```

root@ip-172-31-39-135:~# sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
--2025-02-06 07:17:47-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.46.133, 2a04:4e42:48:645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.46.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins- 100%[=====] 3.10K --.-KB/s   in 0s
2025-02-06 07:17:47 (60.7 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]

root@ip-172-31-39-135:~# echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
root@ip-172-31-39-135:~# sudo apt-get update
Hit:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-southeast-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 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [28.5 kB]
Fetched 160 kB in 1s (110 kB/s)
Reading package lists... Done
root@ip-172-31-39-135:~# sudo apt-get update^C
root@ip-172-31-39-135:~#

```

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

Step29: - Enable the Jenkins using the command as `systemctl enable Jenkins`.

```

Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.492.1_all.deb ...
Unpacking jenkins (2.492.1) ...
Setting up net-tools (1.60-git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.492.1) ...
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 candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...
systemctl restart packagekit.service ssh.service
Service restarts being deferred:
systemctl restart networkd-dispatcher.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

No containers need to be restarted.

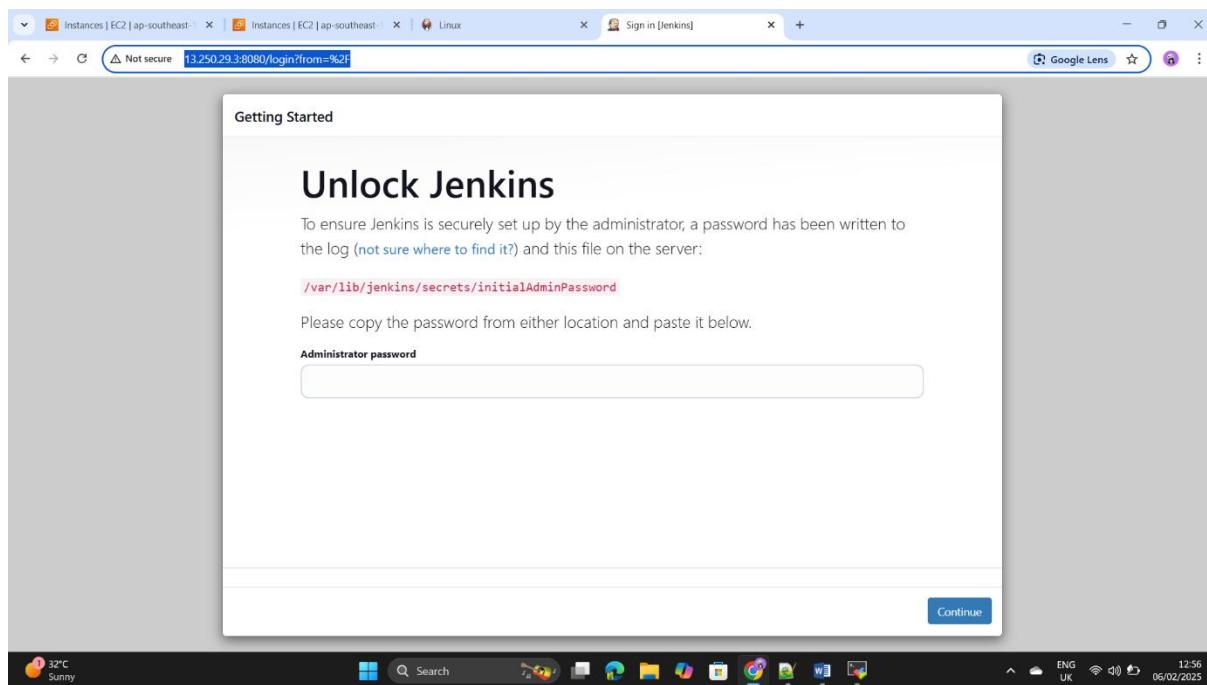
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-39-135:~# jenkins --version
2.492.1
root@ip-172-31-39-135:~# systemctl start jenkins
root@ip-172-31-39-135:~# systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
root@ip-172-31-39-135:~#

```

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

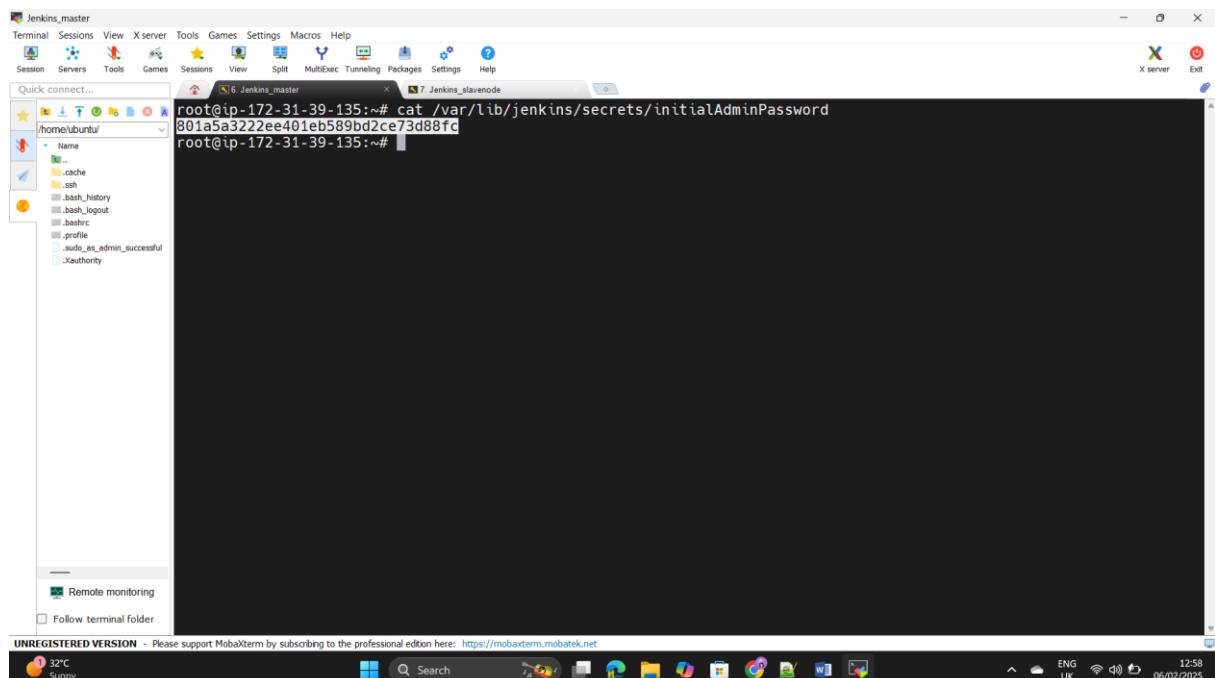
Step30: - Copy the Public Ip address of the Instance and paste in the browser and add the Jenkins port number as 8080.



Step31: - use the command as

cat /var/lib/jenkins/secrets/initialAdminPassword

we will get the password there paste in the browser and login to the Jenkins.



Step32: - Go to the manage Jenkins and nodes ,click on create node.

The screenshot shows the Jenkins 'Nodes' page. At the top, there are tabs for 'Dashboard', 'Manage Jenkins', and 'Nodes'. A yellow arrow points from the 'Manage Jenkins' tab to the 'Nodes' tab. On the right side of the page, there is a 'New Node' button with a yellow arrow pointing towards it. The main table lists one node:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.74 GiB	0 B	4.74 GiB	0ms
	Data obtained	9 min 36 sec	9 min 36 sec	9 min 36 sec	9 min 36 sec	9 min 36 sec	9 min 36 sec

Below the table, there are buttons for 'Icon: S M L' and a 'Legend' link. At the bottom of the page, there are links for 'REST API' and 'Jenkins 2.492.1'.

Step33: - Login into the Client_server2 has a Jenkins_slvานode.

The screenshot shows a MobaXterm session titled 'Jenkins_slvานode'. It has two windows open: 'ubuntu@ip-172-31-39-68:~\$' and 'root@ip-172-31-39-68:~#'. The second window is active and shows a root shell. The terminal window has a sidebar with a file tree and some configuration options. The status bar at the bottom indicates an 'UNREGISTERED VERSION' and shows system information like temperature (32°C), battery level (Sunny), and date/time (06/02/2025).

Step34: - Create the user and create the ssh key in that instance.

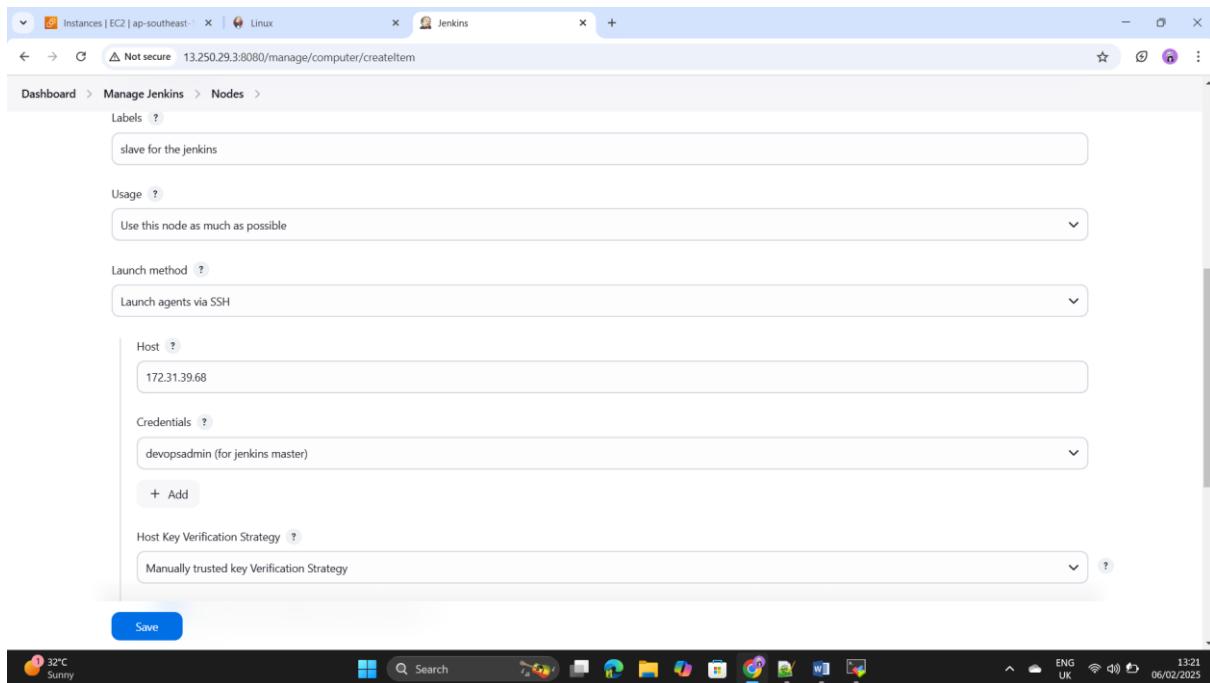
```
devopsadmin@ip-172-31-39-68:~/ssh$ ls
authorized_keys  id_ecdsa  id_ecdsa.pub
devopsadmin@ip-172-31-39-68:~/ssh$
```

Step35: - Change the access mod of the file by using command as
chmod 600 <file_path>

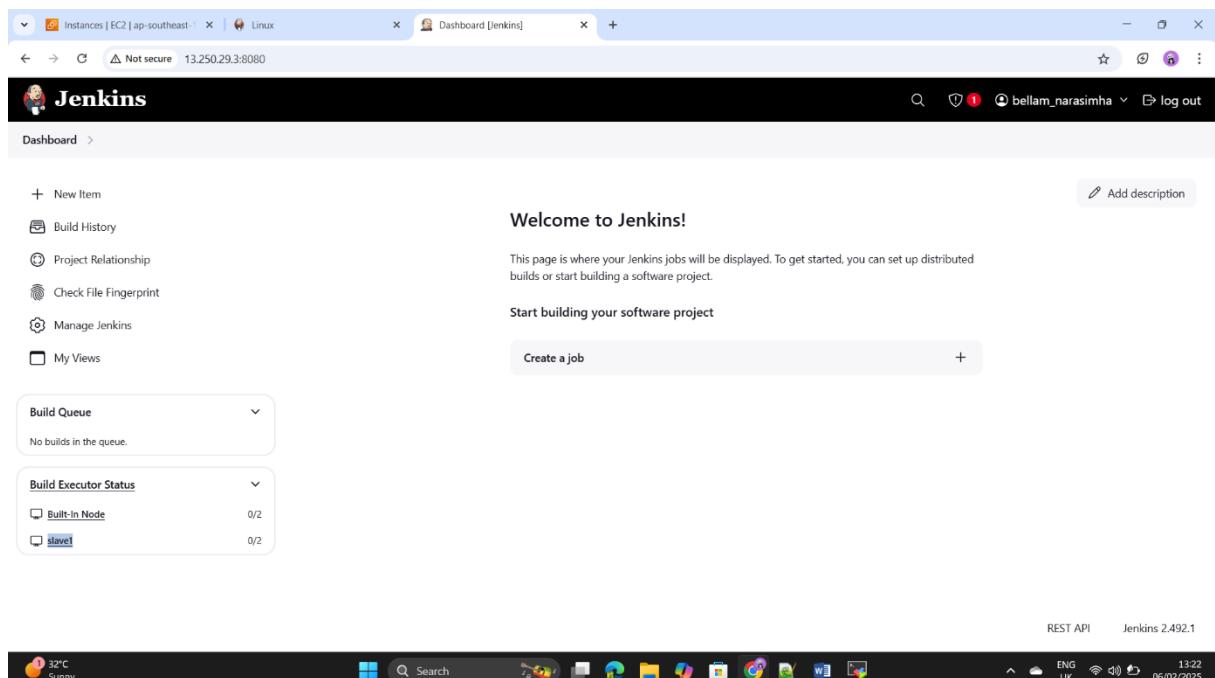
Use the command – usermod –aG docker
devopsadmin(for docker purpose).

```
devopsadmin@ip-172-31-39-68:~/ssh$ ls
authorized_keys  id_ecdsa  id_ecdsa.pub
devopsadmin@ip-172-31-39-68:~/ssh$ chmod 600 /home/devopsadmin/.ssh/*
devopsadmin@ip-172-31-39-68:~/ssh$
```

Step36: - Go to Jenkins web browser and add the slave node to the Jenkins.



Step37: - Jenkins slave node is ready for the build the code.



Step38: - Now connect the Docker hub through credentials, first login to the Docker hub go to the account settings in that select the Personal access token click on that.

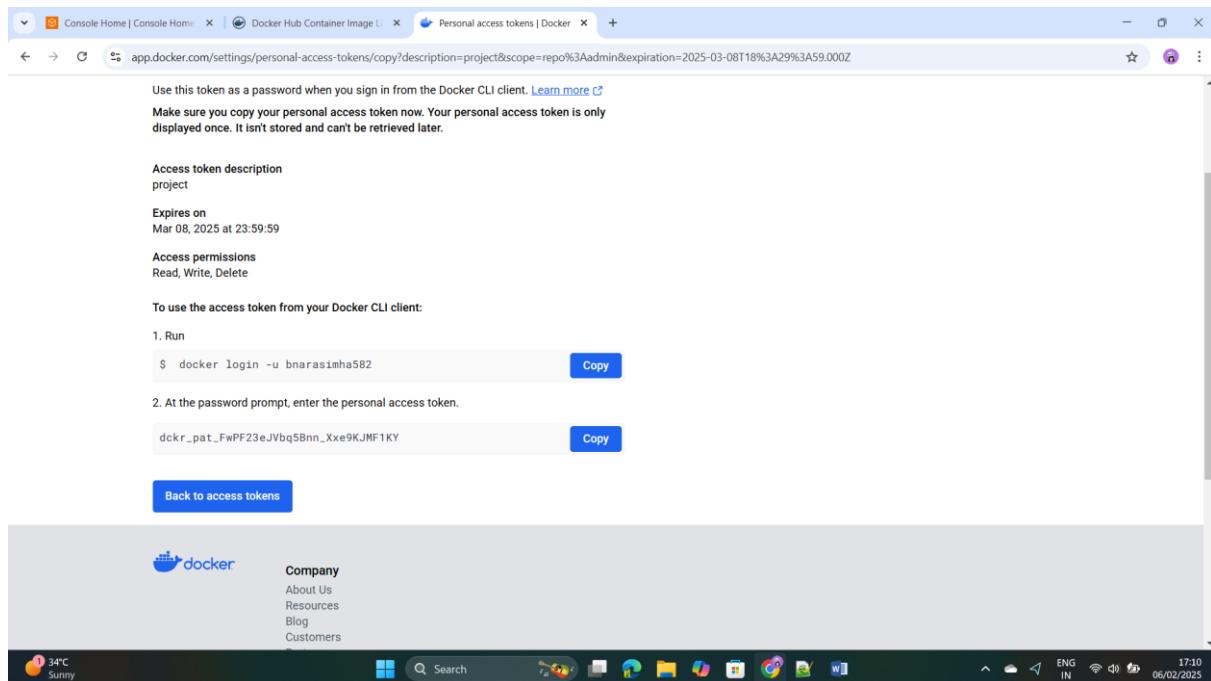
The screenshot shows the Docker settings page for a user account. The 'Personal access tokens' section is highlighted with a blue box and a red arrow pointing to it. This section displays 10 tokens associated with the account. Other sections like 'Connected accounts' and 'Account management' are also visible.

Description	Scope	Status	Source	Created	Last used	Expiration date
33	Read, Write, Delete	Active	Manual	Jan 24, 2025 at 13:57:49	Feb 05, 2025 at 18:46:43	Feb 23, 2025 at 23:59
kkk	Read, Write, Delete	Active	Manual	Jan 27, 2025 at 09:55:08	Jan 27, 2025 at 09:55:22	Feb 26, 2025 at 23:59
33	Public Repo Read-only	Active	Manual	Jan 27, 2025 at 09:54:02	Jan 27, 2025 at 09:54:20	Feb 26, 2025 at 23:59
ok	Read, Write, Delete	Active	Manual	Jan 24, 2025 at 08:44:51	Jan 24, 2025 at 08:50:32	Feb 23, 2025 at 23:59
kk	Read, Write, Delete	Active	Manual	Jan 23, 2025 at 22:11:09	Jan 23, 2025 at 22:11:53	Feb 22, 2025 at 23:59
ok	Read, Write, Delete	Active	Manual	Jan 22, 2025 at 17:10:53	Jan 22, 2025 at 18:19:03	Feb 21, 2025 at 23:59

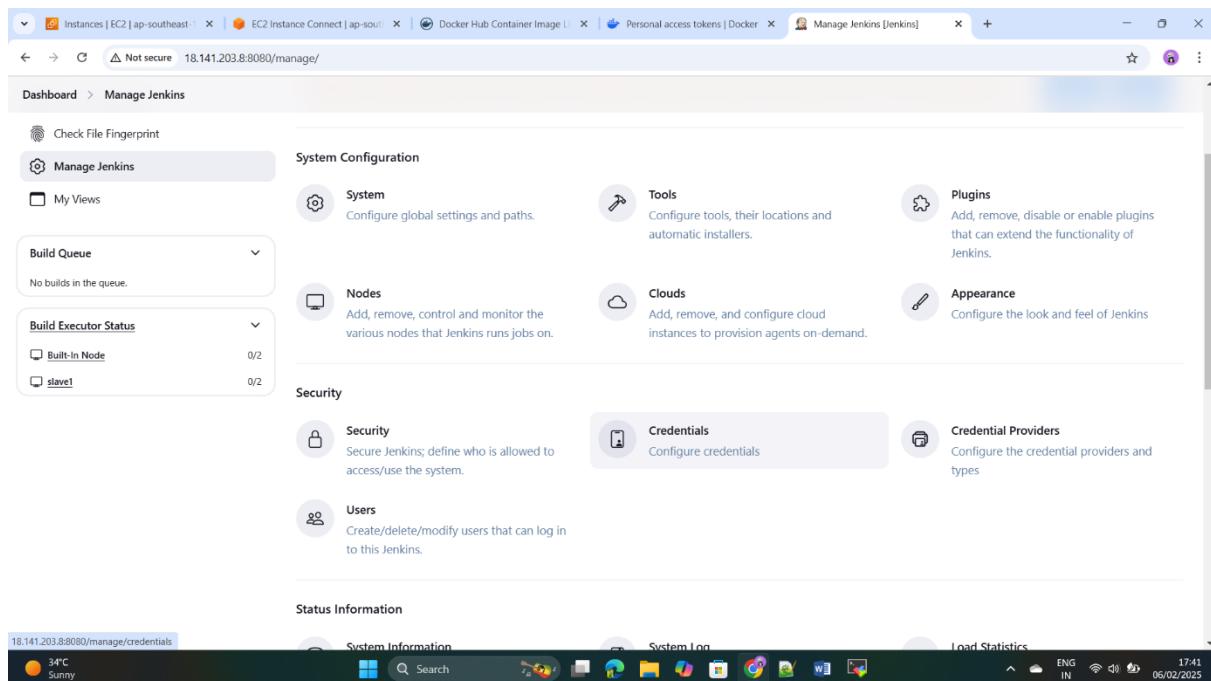
Step39: - Select the Generate the new token and create the token.

The screenshot shows the 'Personal access tokens' page with a red arrow pointing to the 'Generate new token' button. This button is located above a table listing existing tokens.

Description	Scope	Status	Source	Created	Last used	Expiration date
33	Read, Write, Delete	Active	Manual	Jan 24, 2025 at 13:57:49	Feb 05, 2025 at 18:46:43	Feb 23, 2025 at 23:59
kkk	Read, Write, Delete	Active	Manual	Jan 27, 2025 at 09:55:08	Jan 27, 2025 at 09:55:22	Feb 26, 2025 at 23:59
33	Public Repo Read-only	Active	Manual	Jan 27, 2025 at 09:54:02	Jan 27, 2025 at 09:54:20	Feb 26, 2025 at 23:59
ok	Read, Write, Delete	Active	Manual	Jan 24, 2025 at 08:44:51	Jan 24, 2025 at 08:50:32	Feb 23, 2025 at 23:59
kk	Read, Write, Delete	Active	Manual	Jan 23, 2025 at 22:11:09	Jan 23, 2025 at 22:11:53	Feb 22, 2025 at 23:59
ok	Read, Write, Delete	Active	Manual	Jan 22, 2025 at 17:10:53	Jan 22, 2025 at 18:19:03	Feb 21, 2025 at 23:59



Step40: - go to the manage Jenkins and select the credentials.



Step41: - Go to the credentials and click on add credentials.

The screenshot shows the Jenkins 'Credentials' management interface. At the top, there's a header bar with tabs for Instances, EC2, Docker Hub Container Image, Personal access tokens, and Jenkins. Below the header, the Jenkins logo is visible, followed by a search bar and user information (bellam). The main content area is titled 'Credentials' and shows a table with one row:

T	P	Store ↓	Domain	ID	Name
		System	(global)	slave1	devopsadmin (project purpose)

Below the table, a section titled 'Stores scoped to Jenkins' is shown, featuring a smaller table for 'System' credentials with '(global)' selected. A button labeled 'Add credentials' is visible. The bottom of the screen shows a Windows taskbar with various icons and system status.

Step42: - Add the username and password as we copy from the docker hub to paste it here and click on create.

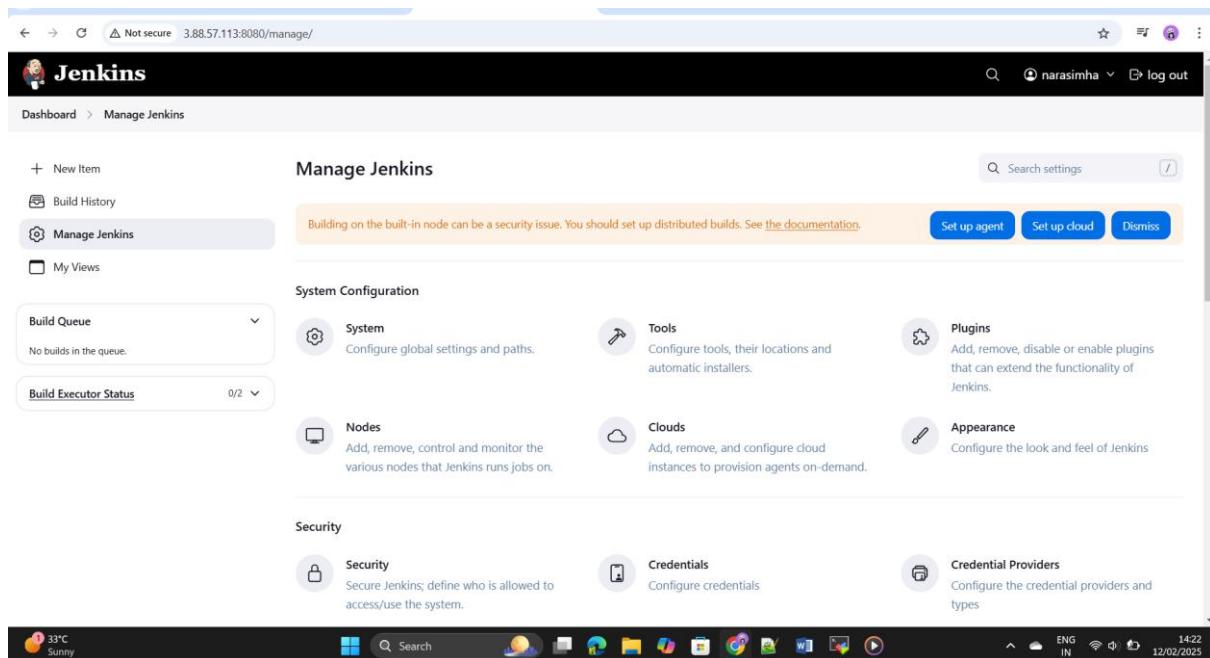
The screenshot shows the Jenkins 'New credentials' creation form for a 'Username with password' credential. The URL in the browser is 18.141.203.8:8080/manage/credentials/store/system/domain/_/newCredentials. The form fields are as follows:

- Kind:** Username with password
- Scope:** Global (Jenkins, nodes, items, all child items, etc)
- Username:** bnarasimha582
- Treat username as secret:** (unchecked)
- Password:** (redacted)
- ID:** sadocker
- Description:** Docker login

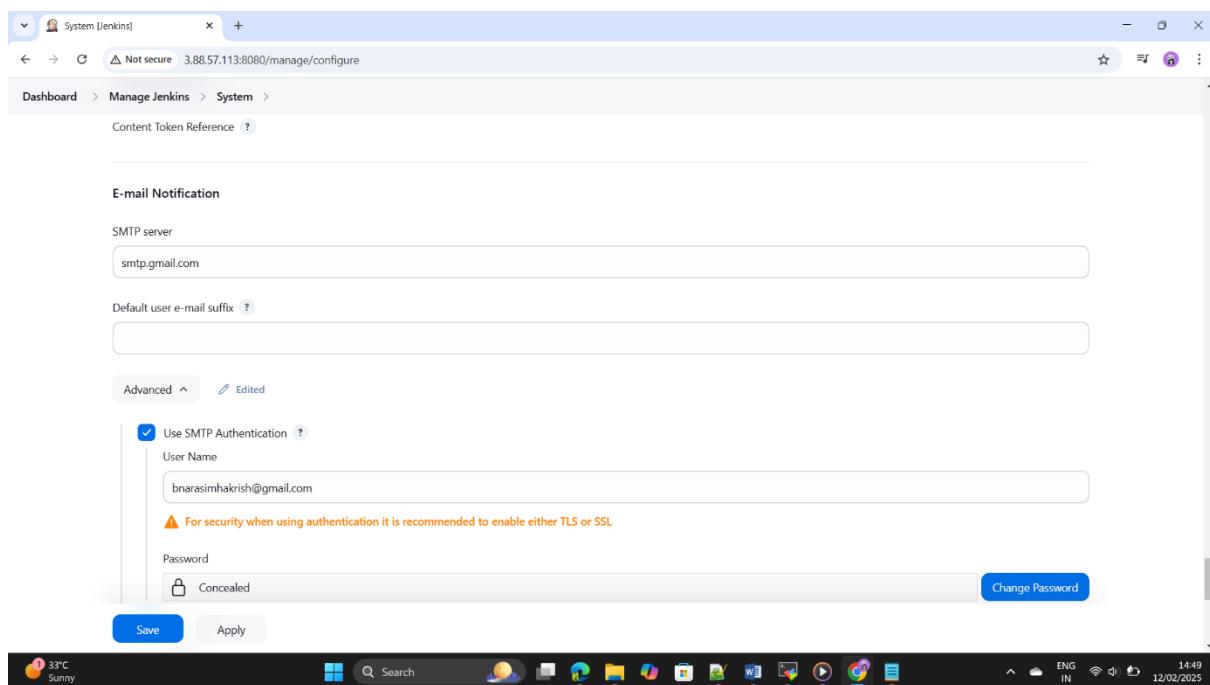
A blue 'Create' button is at the bottom left. The bottom of the screen shows a Windows taskbar with various icons and system status.

Step43: - Save the credentials for dockerhub login & image pushing.

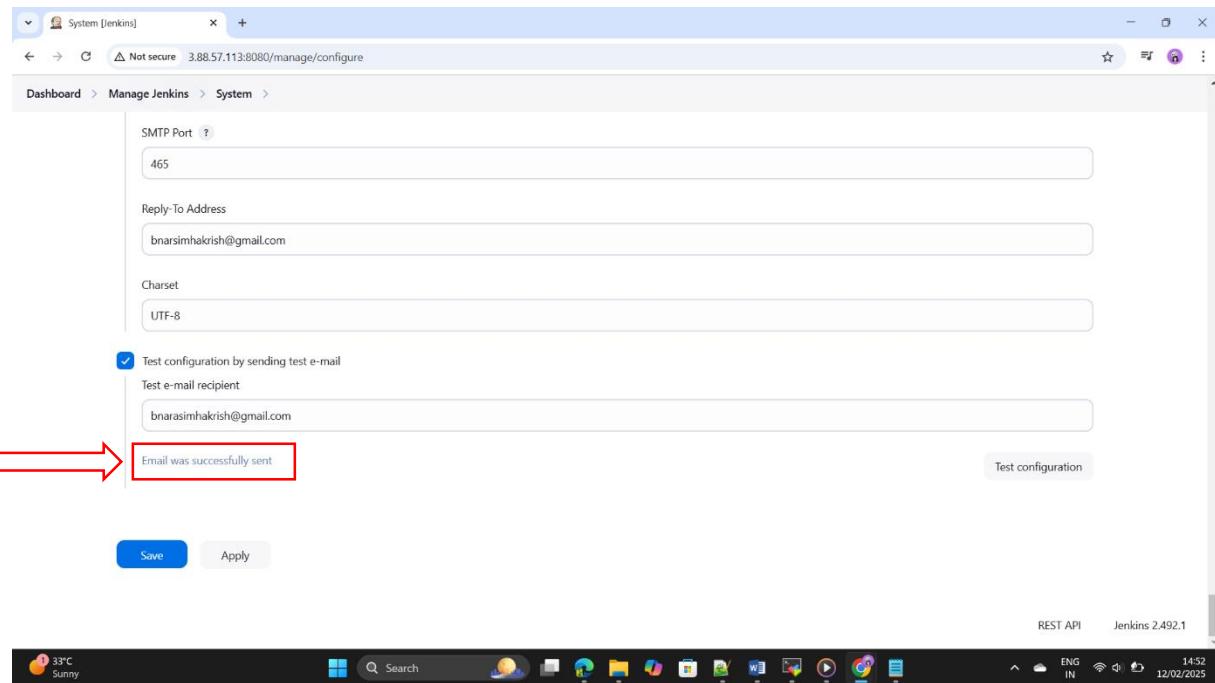
Step44: - Add the post build action to the Jenkins pipeline as a email notification regarding the pipeline status got to the Jenkins system configuration.



Step45: - Add the Email notification and save it.

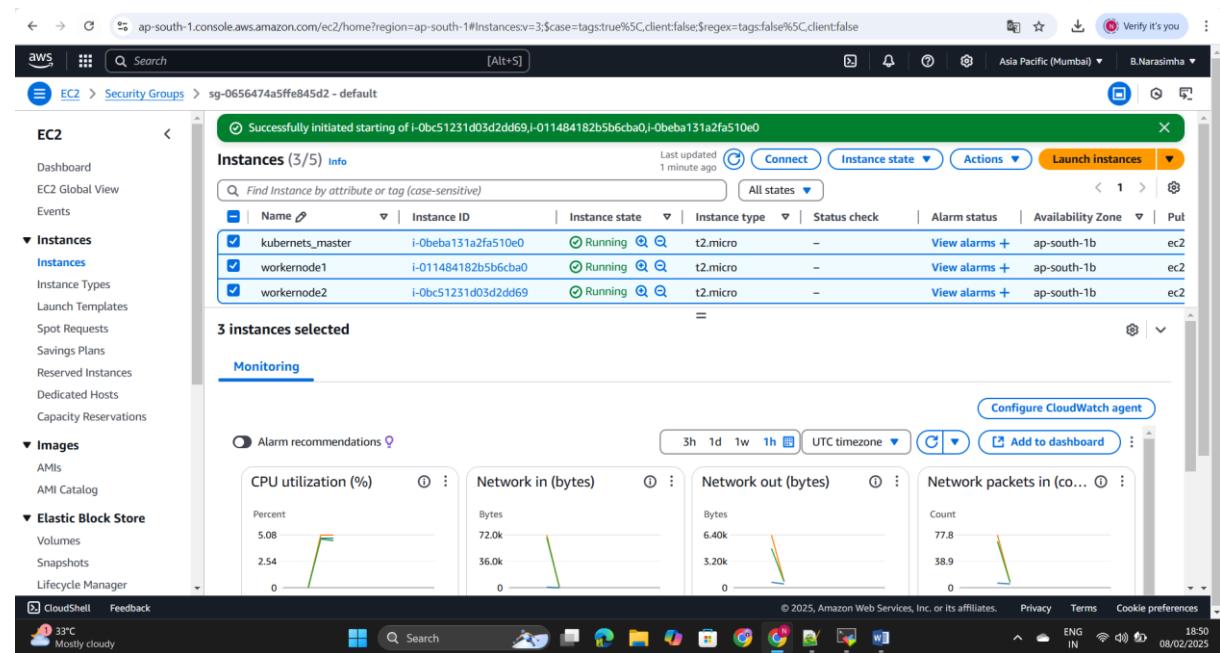


Step46: - Create the password in the gmail(go to the manage and click on 2 step verification and add the app password and copy the password and paste it here and click on the test configuration and save it.)



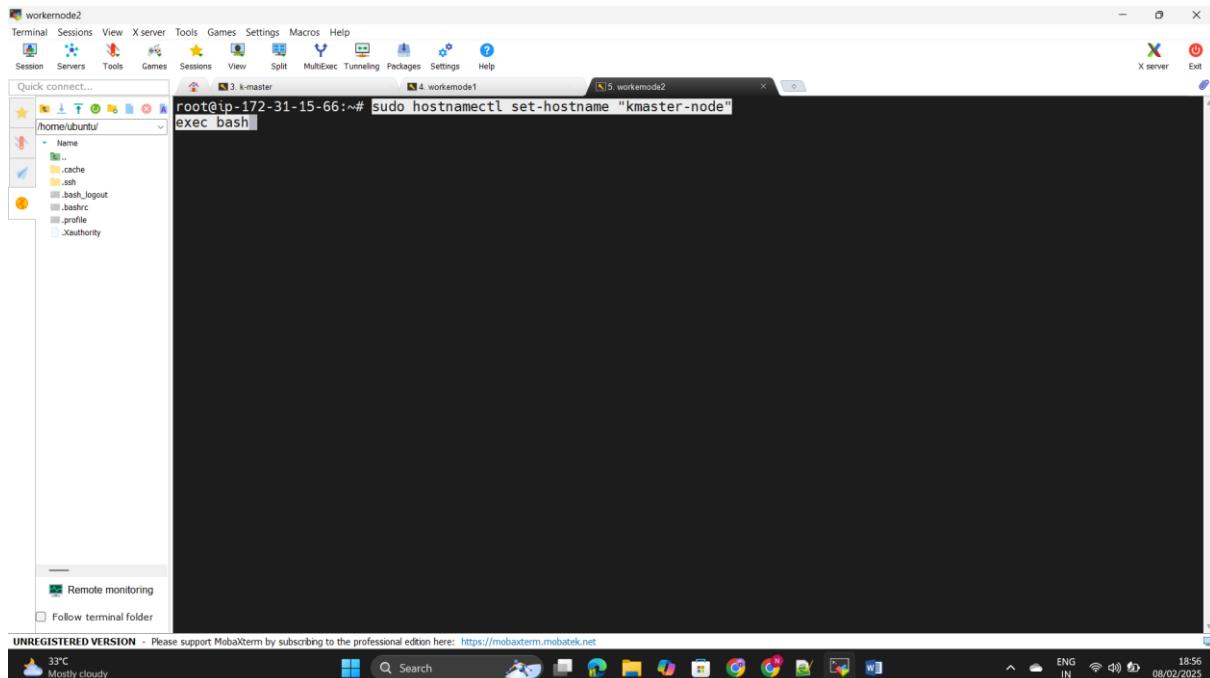
Step47: - Intstalll the kubernets for the pods deployment.

Step48: - We have to add the kubernet master node to the Jenkins master node for that install the kubernets and its worker nodes.



Step49: - Connect the 3 instances and update and change the host name by using the command as –

```
sudo hostnamectl set-hostname "kmaster-node"  
exec bash
```



Step50: - Disable the swap on both machines by using command –

```
sudo swapoff -a
```

`sudo sed -i '/ swap / s/^(\.*$)/#\1/g' /etc/fstab`

```
root@kmaster-node:~# sudo swapoff -a
sudo sed -i '/ swap / s/^(\.*$)/#\1/g' /etc/fstab
root@kmaster-node:~#
```

Step51: - Install docker by using command –

`sudo apt install docker.io`

```
root@worker-node2:~# sudo swapoff -a
sudo sed -i '/ swap / s/^(\.*$)/#\1/g' /etc/fstab
root@worker-node2:~# apt 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 containedr dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse
  | zfsutils
The following NEW packages will be installed:
  bridge-utils containedr dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 35 not upgraded.
Need to get 78.7 MB of archives.
After this operation, 301 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.12-0ubuntu2~22.04.1 [8405 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.24-0ubuntu1~22.04.1 [37.3 MB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dns-root-data all 2023112702~ubuntu0.22.04.1 [5136 B]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dnsmasq-base amd64 2.90-0ubuntu0.22.04.1 [374 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 26.1.3-0ubuntu1~22.04.1 [2.5 MB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 78.7 MB in 1s (92.2 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 65853 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
```

Step52: - install the container –D by following some steps as configure the net filters by using the below command

`sudo modprobe overlay`

```
sudo modprobe br_nf  
cat <<EOF | sudo tee /etc/modules  
load.d/containerd.conf  
overlay  
br_nf  
EOF
```

The screenshot shows a terminal window titled 'workmode2' running on a Linux system. The terminal is root-privileged, indicated by the '#'. The user has run the command 'sudo modprobe overlay' followed by 'cat <<EOF | sudo tee /etc/modules-load.d/containerd.conf' to add the 'overlay' and 'br_nf' modules to the kernel's module loading configuration. The terminal window also shows other system information like the desktop environment (KDE) and network status.

Step53: - now Configure the IP6 table ref by using below command.

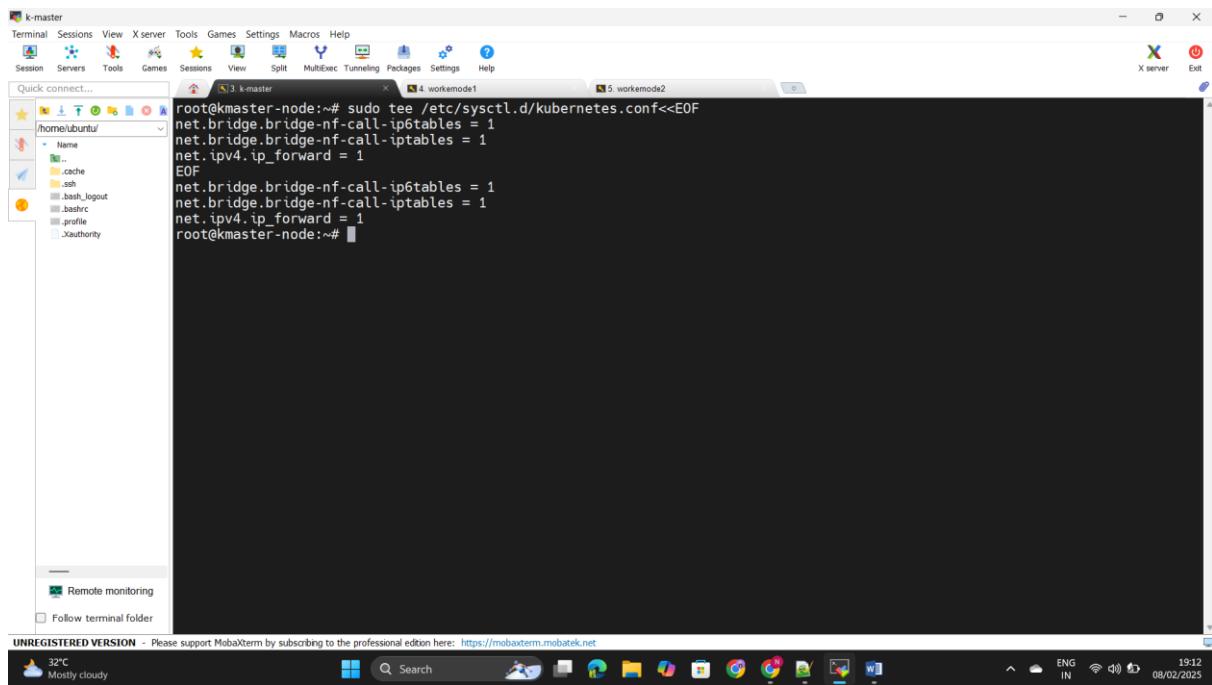
```
sudo tee /etc/sysctl.d/kubernetes.conf<<EOF
```

```
net.bridge.bridge-nf-call-ip6tables = 1
```

```
net.bridge.bridge-nf-call-iptables = 1
```

```
net.ipv4.ip_forward = 1
```

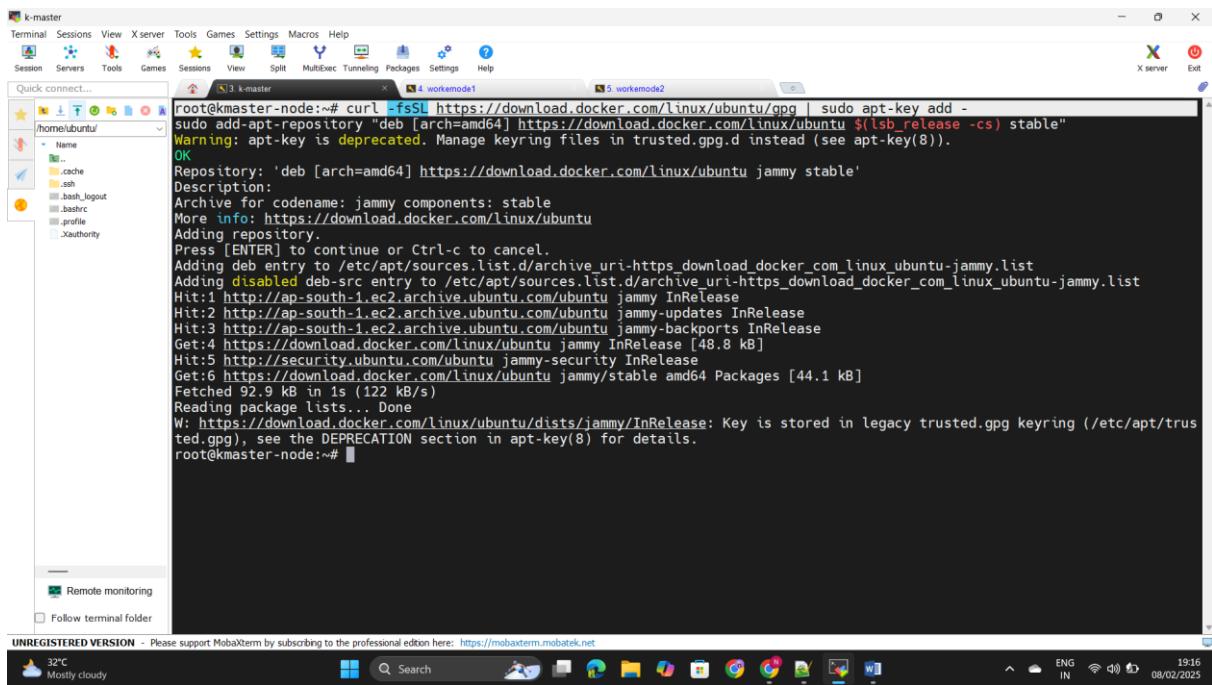
```
EOF
```



```
root@kmaster-node:~# sudo tee /etc/sysctl.d/kubernetes.conf <<EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
root@kmaster-node:~#
```

Step54: - Download the Containers by using curl command as below

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg |  
sudo apt-key add -  
  
sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu $(lsb_release  
cs) stable"
```

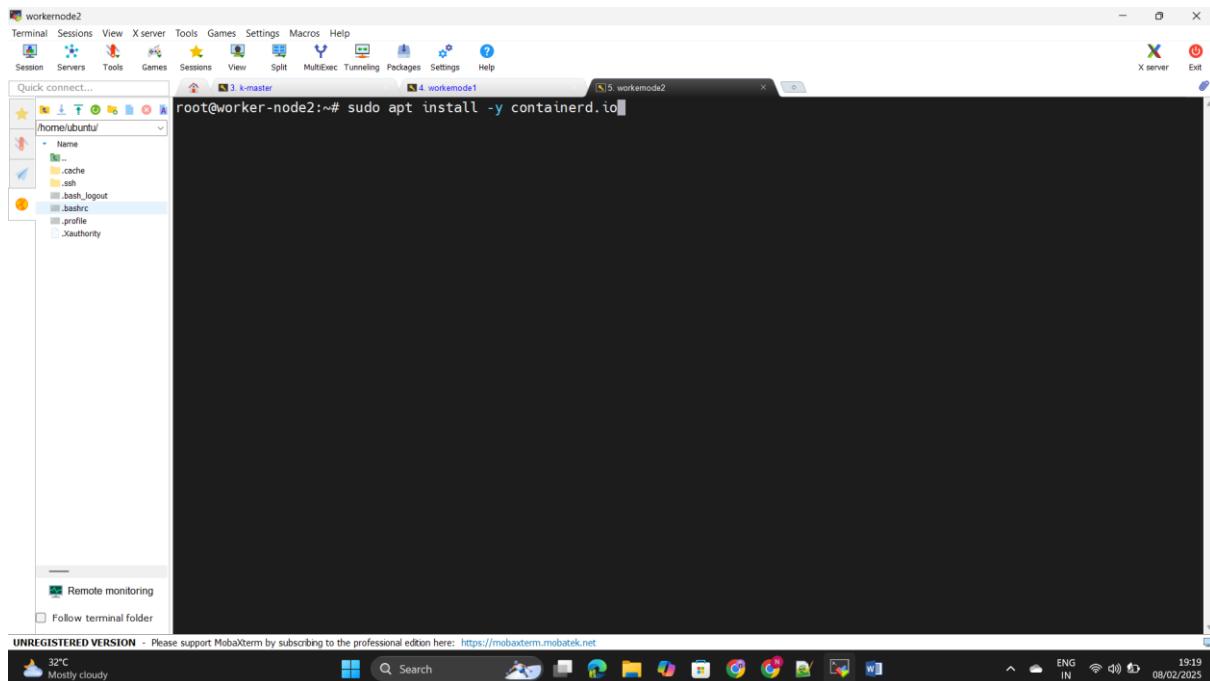


The screenshot shows a terminal window in MobaXterm with the following command and output:

```
root@kmaster-node:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Warning: apt-key is deprecated. Manage keyring files in /etc/trusted.gpg.d instead (see apt-key(8)).
OK
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu jammy stable'
Description:
Archive for codename: jammy components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository...
Press [ENTER] to continue or Ctrl-C to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Hit:5 https://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [44.1 kB]
Fetched 92.9 kB in 1s (122 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
root@kmaster-node:~#
```

Step55: - Install the Container -D by using below command.

sudo apt install -y containerd.io



The screenshot shows a terminal window in MobaXterm with the following command and output:

```
root@worker-node2:~# sudo apt install -y containerd.io
```

Step56: - Configure the .toml file by using the below command as-

```
sudo mkdir -p /etc/containerd
```

```
sudo containerd config default | sudo tee
etc/containerd/config.toml
```

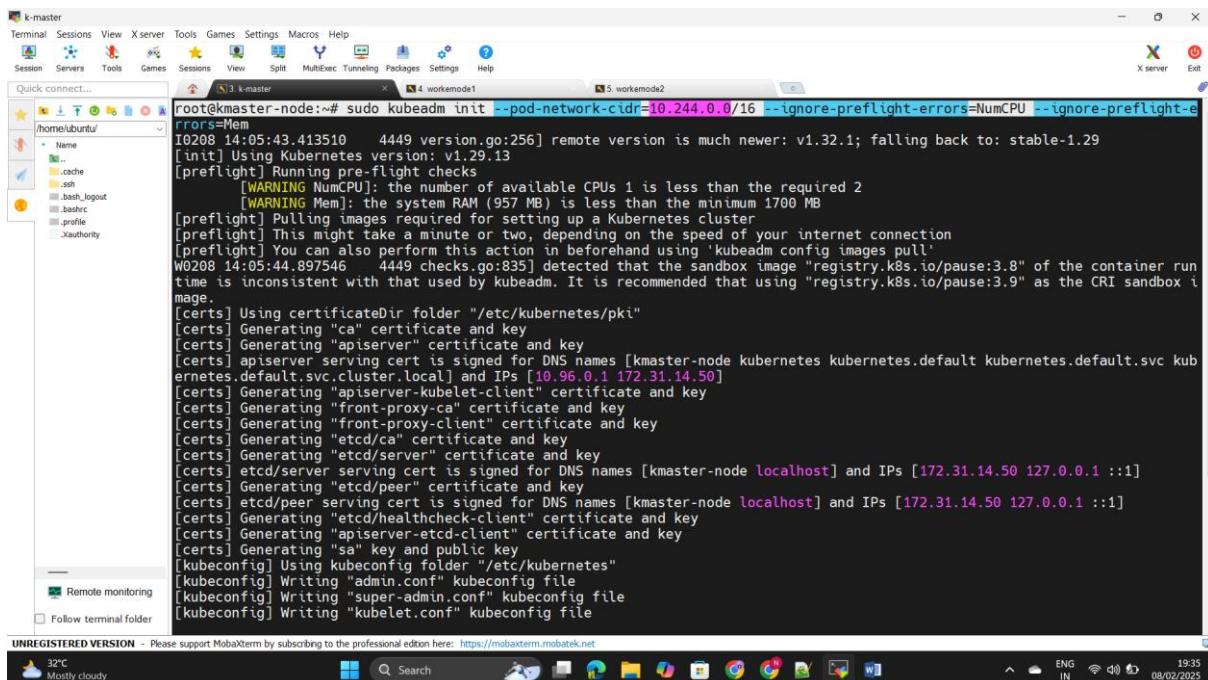
The screenshot shows a MobaXterm window titled 'k-master'. It has four tabs open: '3. k-master', '4. workemode1', and '5. workemode2'. The '3. k-master' tab is active and shows a root shell. The user runs the command 'sudo mkdir -p /etc/containerd' followed by 'sudo containerd config default | sudo tee etc/containerd/config.toml'. The terminal window also displays a message about an unregistered version and the system status bar at the bottom.

Step57: - Install the kubeadm,kubelet,kubectl by using the below command - sudo apt-get install -y kubelet kubeadm kubectl

The screenshot shows a MobaXterm window titled 'k-master'. It has four tabs open: '3. k-master', '4. workemode1', and '5. workemode2'. The '3. k-master' tab is active and shows a root shell. The user runs the command 'sudo apt-get install -y kubelet kubeadm kubectl'. The terminal window displays the output of the package installation process, including dependency resolution and file placement. The terminal window also displays a message about an unregistered version and the system status bar at the bottom.

Step58: - Execute the below command in the Masternode only

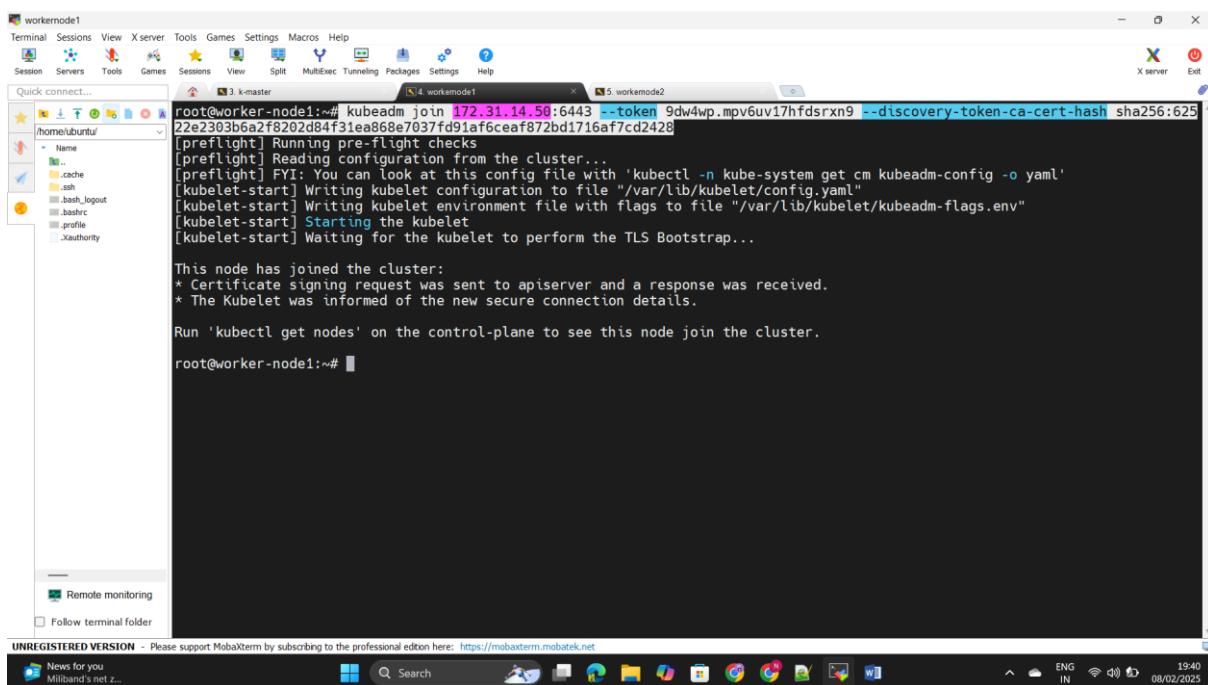
```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
ignore-preflight-errors=NumCPU --ignore-preflight
errors=Mem
```



The screenshot shows a terminal window titled 'root@kmaster-node:~#'. The command entered is 'sudo kubeadm init --pod-network-cidr=10.244.0.0/16 ignore-preflight-errors=NumCPU --ignore-preflight errors=Mem'. The output of the command is displayed, showing various logs and warnings related to certificate generation and network configuration. The terminal window is part of the MobaXterm interface, which includes a sidebar with session icons and a taskbar at the bottom.

```
root@kmaster-node:~# sudo kubeadm init --pod-network-cidr=10.244.0.0/16 ignore-preflight-errors=NumCPU --ignore-preflight errors=Mem
I0208 14:05:43.413510 4449 version.go:256] remote version is much newer: v1.32.1; falling back to: stable-1.29
[init] Using Kubernetes version: v1.29.13
[preflight] Running pre-flight checks
[WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
[WARNING Mem]: the system RAM (957 MB) is less than the minimum 1700 MB
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0208 14:05:44.897546 4449 checks.go:835] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container run time is inconsistent with that used by kubeadm. It is recommended that using "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [kmaster-node kubernetes.kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.14.50]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [kmaster-node localhost] and IPs [172.31.14.50 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [kmaster-node localhost] and IPs [172.31.14.50 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
```

Step59: - Join the kubeadm join in the workernode.



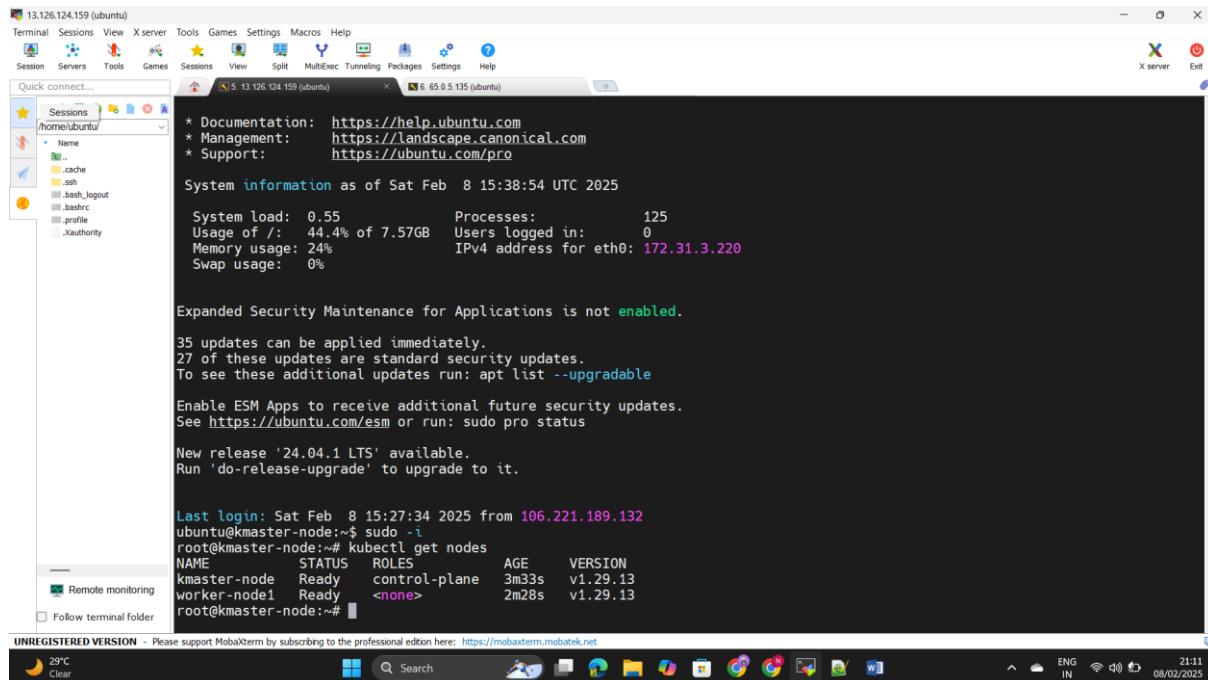
The screenshot shows a terminal window titled 'root@worker-node1:~#'. The command entered is 'kubeadm join 172.31.14.50:6443 --token 9dw4wp.mpv6uv17hfdsrnx9 --discovery-token-ca-cert-hash sha256:62522e2303b6a2f8202d84f31ea868e7037fd91af6ceaf872bd1716af7cd2428'. The output shows the node joining the cluster, including details about certificate exchange and kubelet startup. The terminal window is part of the MobaXterm interface, which includes a sidebar with session icons and a taskbar at the bottom.

```
root@worker-node1:~# kubeadm join 172.31.14.50:6443 --token 9dw4wp.mpv6uv17hfdsrnx9 --discovery-token-ca-cert-hash sha256:62522e2303b6a2f8202d84f31ea868e7037fd91af6ceaf872bd1716af7cd2428
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
root@worker-node1:~#
```

Step60: - Join the worker node to the master node by using kubeadm join command.



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

System information as of Sat Feb 8 15:38:54 UTC 2025

System load: 0.55 Processes: 125
Usage of /: 44.4% of 7.57GB Users logged in: 0
Memory usage: 24% IPv4 address for eth0: 172.31.3.220
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

35 updates can be applied immediately.
27 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

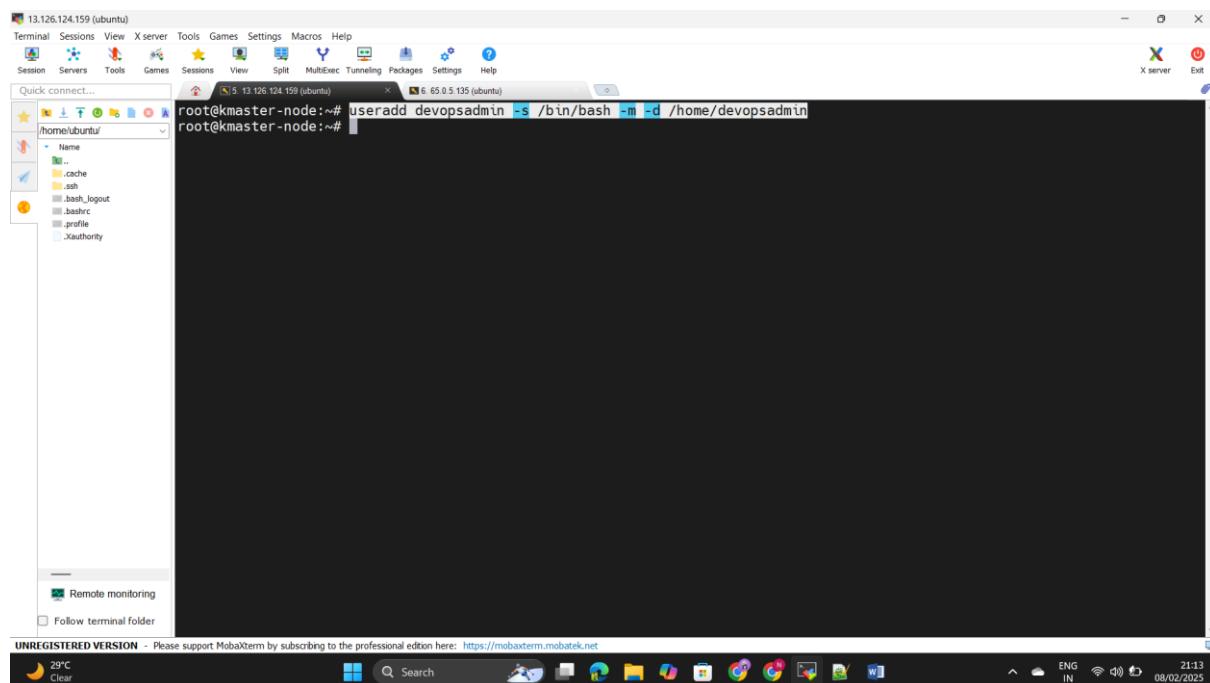
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

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

Last login: Sat Feb 8 15:27:34 2025 from 106.221.189.132
ubuntu@master-node:~$ sudo -i
root@kmaster-node:~# kubectl get nodes
NAME     STATUS   ROLES    AGE     VERSION
kmaster-node   Ready    control-plane   3m33s   v1.29.13
worker-node1   Ready    <none>    2m28s   v1.29.13
root@kmaster-node:~#
```

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

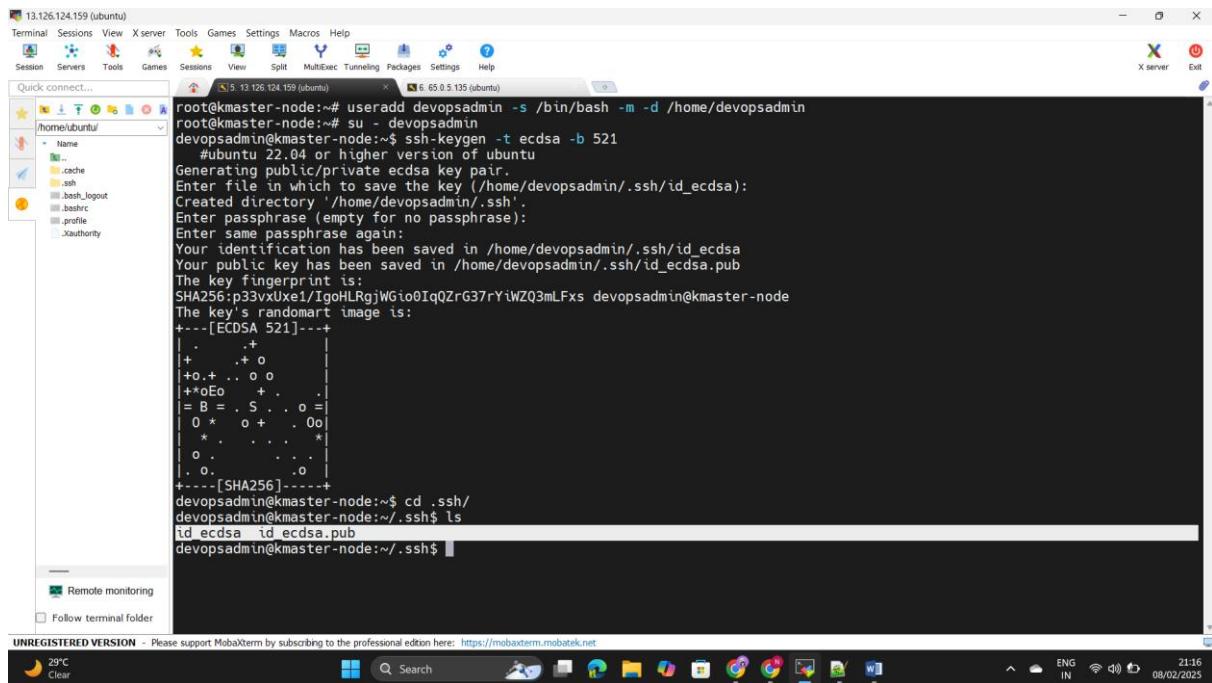
Step61: - Create the user in the Master node by using below command- useradd devopsadmin -s /bin/bash -m -d home/devopsadmin



```
root@kmaster-node:~# useradd devopsadmin -s /bin/bash -m -d /home/devopsadmin
root@kmaster-node:~#
```

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

Step62: - Create the ssh keypair by using command as - ssh-keygen -t ecdsa -b 521.

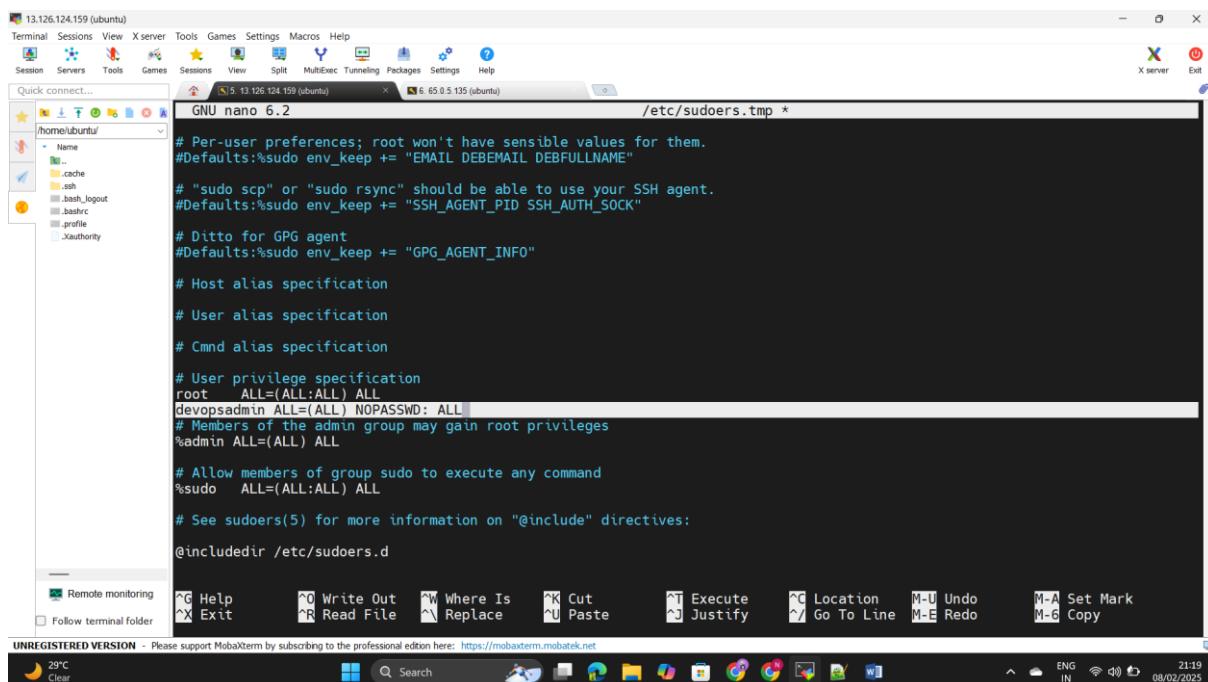


```

root@kmaster-node:~# useradd devopsadmin -s /bin/bash -m -d /home/devopsadmin
root@kmaster-node:~# su - devopsadmin
devopsadmin@kmaster-node:~$ ssh-keygen -t ecdsa -b 521
#ubuntu 22.04 or higher version of ubuntu
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/devopsadmin/.ssh/id_ecdsa):
Created directory '/home/devopsadmin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/devopsadmin/.ssh/id_ecdsa
Your public key has been saved in /home/devopsadmin/.ssh/id_ecdsa.pub
The key fingerprint is:
SHA256:p33vxUxe1/IgoHLRgjWGlo0IqZrG37rYiWZQ3mLFxs devopsadmin@kmaster-node
The key's randomart image is:
+---[ECDSA 521]---+
| . . + |
| + . + 0 |
| +o+ .. o o |
| ++*Eo + . . |
|= B = . S . o =|
| 0 * o + . 0o |
| * . . . * |
| o . . . |
| o . . . 0 |
+---[SHA256]----+
devopsadmin@kmaster-node:~$ cd .ssh/
devopsadmin@kmaster-node:~/ssh$ ls
id_ecdsa id_ecdsa.pub
devopsadmin@kmaster-node:~/ssh$ 

```

Step63: - Give the full access to the user by editing the Visudo file as - devopsadmin ALL=(ALL) NOPASSWD: ALL.



```

root@kmaster-node:~# nano /etc/sudoers.tmp
/etc/sudoers.tmp *
# Per-user preferences; root won't have sensible values for them.
Defaults: %sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
Defaults: %sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
Defaults: %sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
devopsadmin ALL=(ALL) NOPASSWD: ALL
# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:
@include /etc/sudoers.d

^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute   ^C Location   M-U Undo   M-A Set Mark
^X Exit      ^R Read File   ^Y Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo   M-G Copy

```

Step64: - Go to the user and copy paste the below code for the kubernetes access to the user

```
mkdir -p $HOME/.kube
```

```
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
```

```
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

The screenshot shows a MobaXterm interface with two terminal sessions. Session 1 (13.126.124.159) contains the following commands:

```
devopsadmin@ekmaster-node:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
devopsadmin@ekmaster-node:~$
```

Session 2 (65.0.5.135) is visible in the background. The taskbar at the bottom shows various application icons and system status.

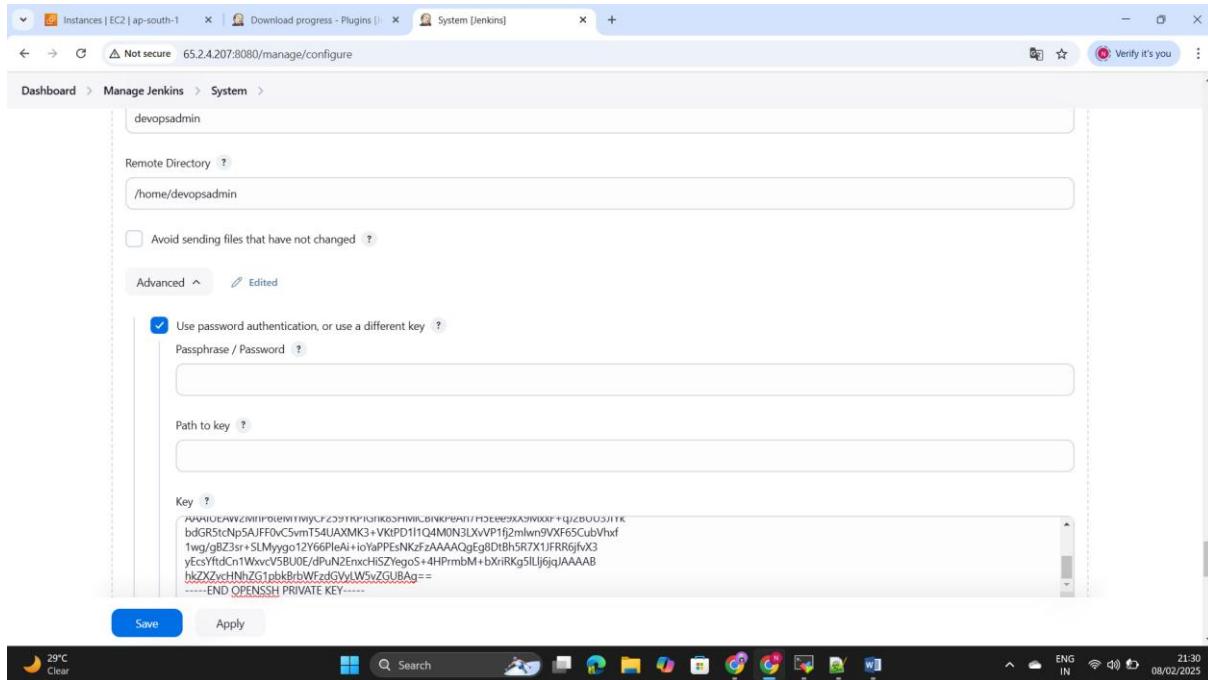
Step65: - Go to the Jenkins master and install the publish over ssh through plugins and go to the manage Jenkins under system configuration.

The screenshot shows the Jenkins System configuration page. The 'System' section includes the following settings:

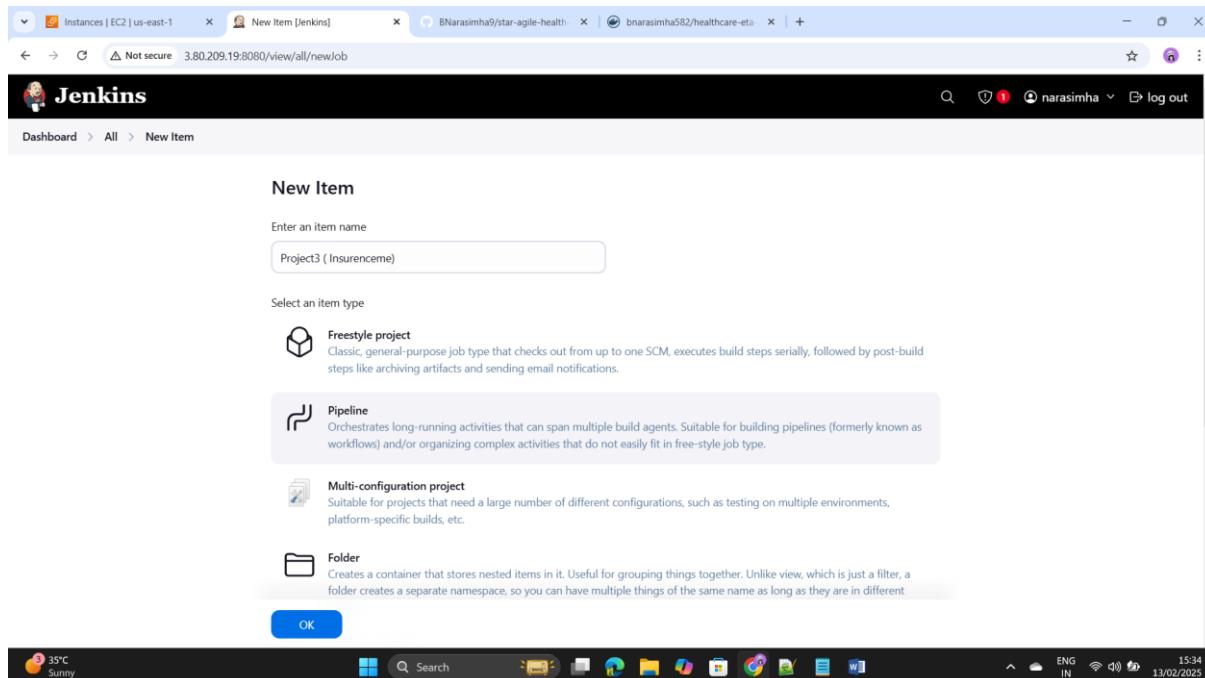
- Home directory:** /var/lib/jenkins
- System Message:** This message will be displayed at the top of the Jenkins main page. This can be useful for posting notifications to your users.
- # of executors:** 2
- Labels:** (empty)

Buttons at the bottom include 'Save' and 'Apply'.

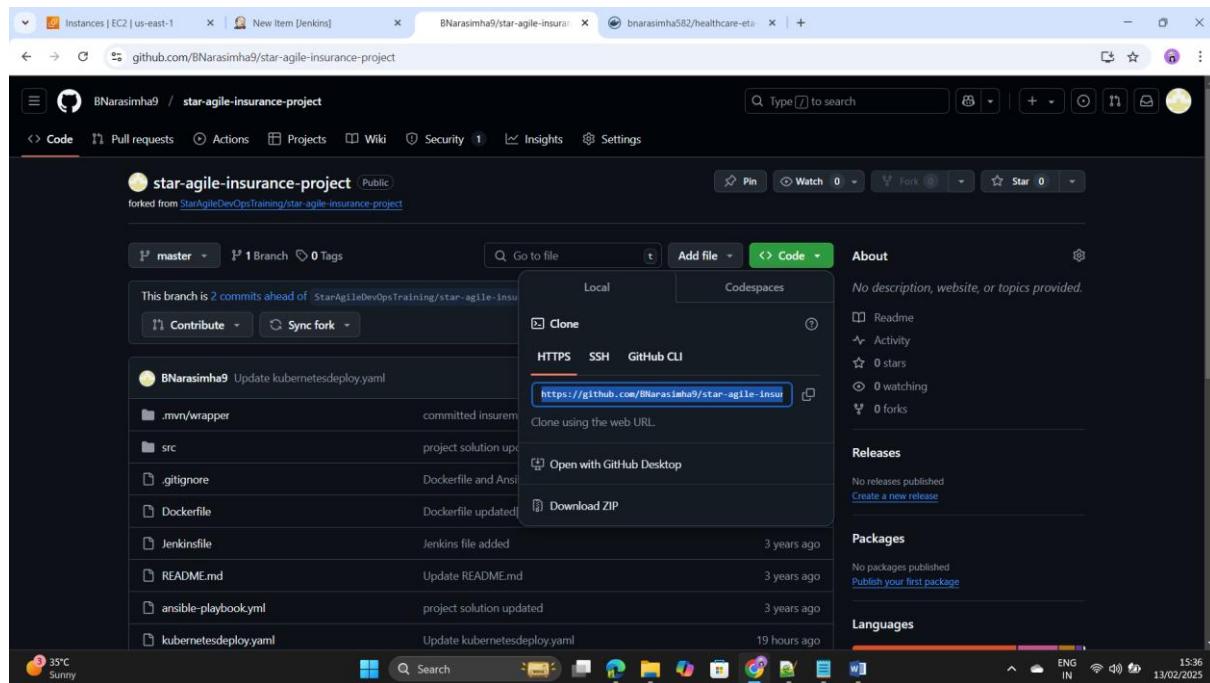
Step66: - Add the kubernet master node by using the ssh connection copy paste the private key of the user and paste here and save it).



Step67: - Create the Pipeline.



Step68: - Add the below pipeline code in the Pipeline by using the sniper Generator.(ADD the source code url)



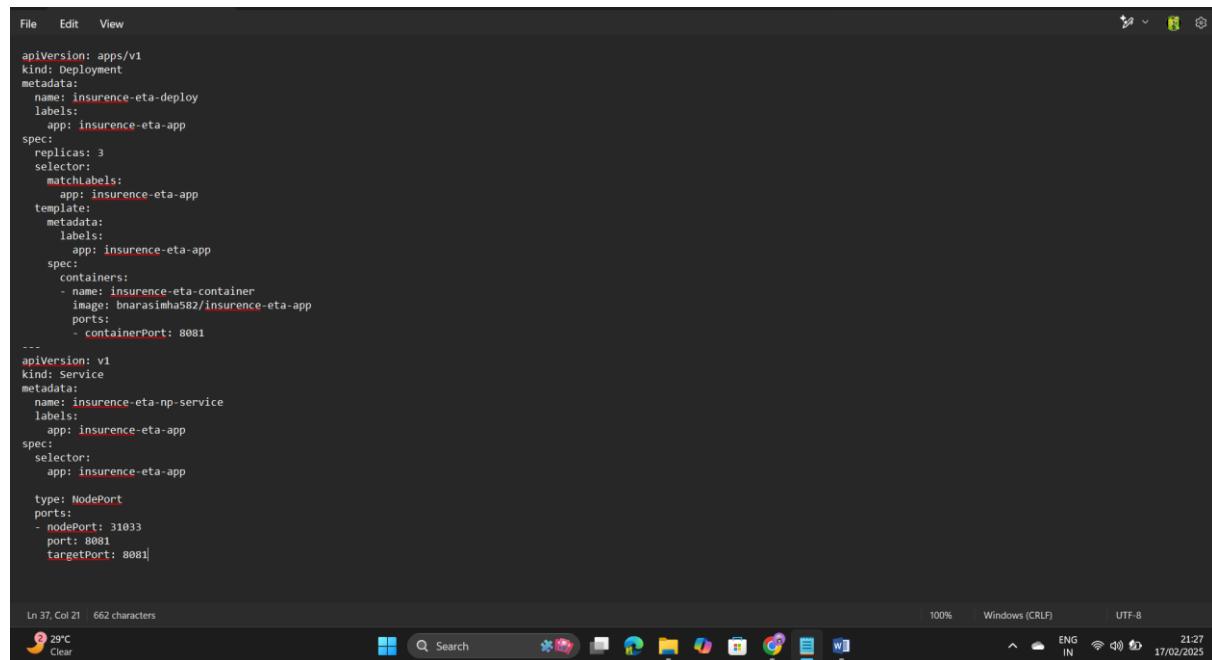
```

File Edit View
pipeline {
    agent { label 'slave1' }

    environment {
        DOCKERHUB_CREDENTIALS=credentials('sadockerlogin')
    }

    stages {
        stage('SCM_Checkout') {
            steps {
                echo "Perform SCM Checkout"
                git 'https://github.com/BNarasimha9/star-agile-insurance-project.git'
            }
        }
        stage('Application Build') {
            steps {
                echo "Perform Application Build"
                sh 'mvn clean package'
            }
        }
        stage('Build Docker Image') {
            steps {
                sh 'docker version'
                sh "docker build -t bnarasimha582/insurance-eta-app:${BUILD_NUMBER} ."
                sh 'docker image list'
                sh "docker tag bnarasimha582/insurance-eta-app:${BUILD_NUMBER} bnarasimha582/insurance-eta-app:latest"
            }
        }
        stage('Login2DockerHub') {
            steps {
                sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
            }
        }
        stage('Publish_to_Docker_Registry') {
            steps {
                sh "docker push bnarasimha582/insurance-eta-app:latest"
            }
        }
    }
    stage('Deploy to Kubernetes Cluster') {
        steps {
    
```

Ln 12, Col 75 | 63 of 1,707 characters

Step69: - Create the .yaml file for the kubernets pods deoployment.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: insurance-eta-deploy
  labels:
    app: insurance-eta-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: insurance-eta-app
  template:
    metadata:
      labels:
        app: insurance-eta-app
    spec:
      containers:
        - name: insurance-eta-container
          image: bnarasimha582/insurance-eta-app
          ports:
            - containerPort: 8081
---
apiVersion: v1
kind: Service
metadata:
  name: insurance-eta-np-service
  labels:
    app: insurance-eta-app
spec:
  selector:
    app: insurance-eta-app
  type: NodePort
  ports:
    - nodePort: 31033
      port: 8081
      targetPort: 8081
```

Step70: - Add the docker credentials as a environment and deploy the pods in the kubernets by using publish over ssh connection in the sniper generator.

The screenshot shows two windows side-by-side, both displaying the Jenkins Pipeline Syntax configuration page for a project named 'Project1.1'.

Top Window (Left):

- Shows the 'sshPublisher' configuration section.
- SSH Server:** Name is set to 'kubernetes'. An 'Advanced' dropdown is visible.
- Transfers:** A 'Transfer Set' is defined with 'Source files' set to '*yaml'. There are fields for 'Remove prefix' and 'Remote directory'.

Bottom Window (Right):

- Shows the 'Generate Pipeline Script' step.
- The generated script is as follows:

```
sshPublisher(publishers: [sshPublisherDesc(configName: 'kubernetes', transfers: [sshTransfer(cleanRemote: false, excludes: "", execCommand: "kubectl apply -f kubernetesDeploy.yaml", execTimeout: 120000, flatten: false, makeEmptyDirs: false, noDefaultExcludes: false, patternSeparator: "\\", remoteDirectory: "", remoteDirectorySDF: false, removePrefix: "", sourceFiles: "*yaml"])]), usePromotionTimestamp: false, useWorkspaceInPromotion: false, verbose: false)])
```

Step71: - Paste in the Deployment step.

The screenshot shows the Jenkins Pipeline configuration page for 'Project3 (Insurenceme)'. The 'Triggers' tab is selected. The 'Pipeline' section contains a 'Definition' dropdown set to 'Pipeline script'. Below it is a code editor showing Groovy script:

```

32      `sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
33
34
35  }
36
37  stage('Publish to Docker Registry') {
38    steps {
39      sh "docker push bnarasimha582/insurance-eta-app:latest"
40    }
41
42  stage('Deploy to Kubernetes Cluster') {
43    steps {
44      script{
45        sshPublisher(publishers: [sshPublisherDesc(configName: 'kubernetes', transfers: [sshTransfer(cleanRemote: false, e
46      }
47    }
48

```

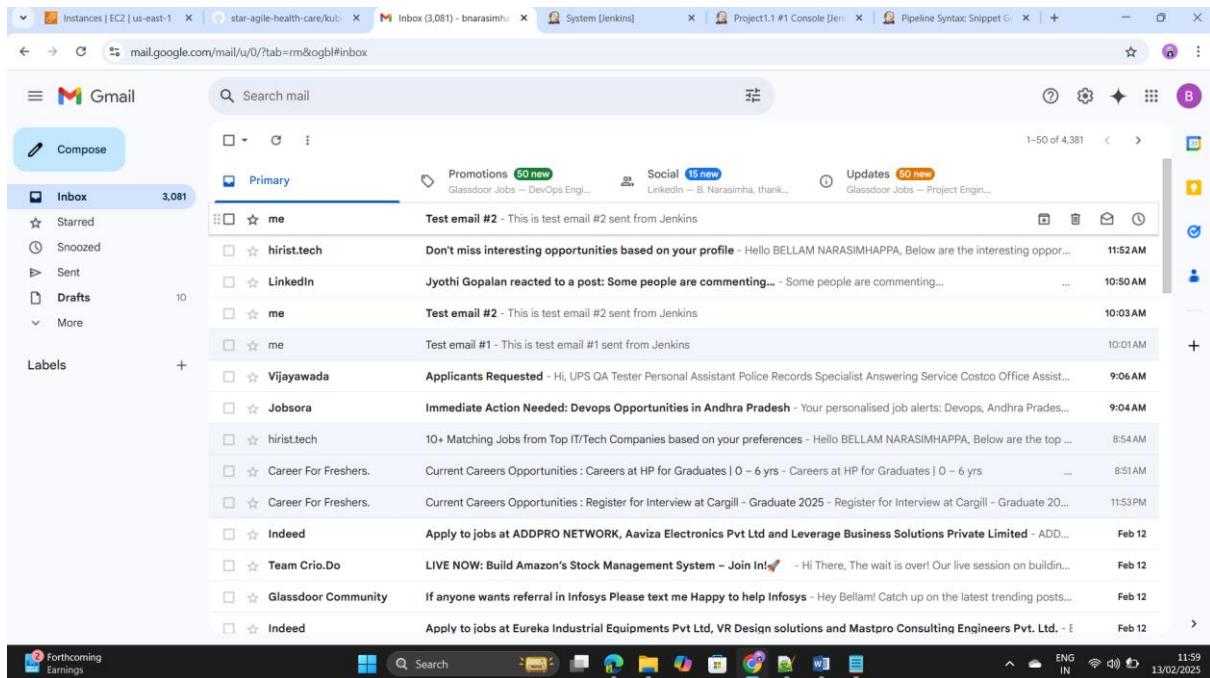
Use Groovy Sandbox

Save Apply

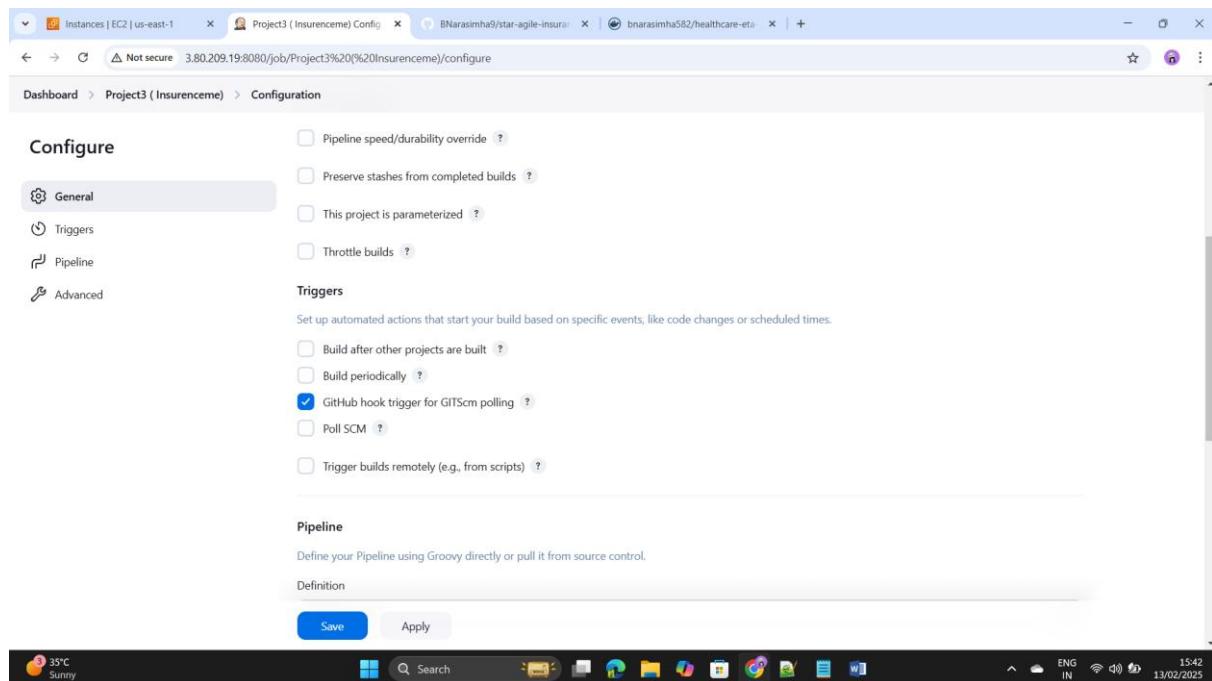
Step72: - add the post build action to this mail and send the status.

The screenshot shows the Jenkins project1 Configuration page under 'Post-build Actions'. The 'Post-build Actions' tab is selected. It displays an 'Editable Email Notification' section with fields for 'Project From' and 'Project Recipient List' (containing 'bnarasimhakrish@gmail.com').

Save Apply

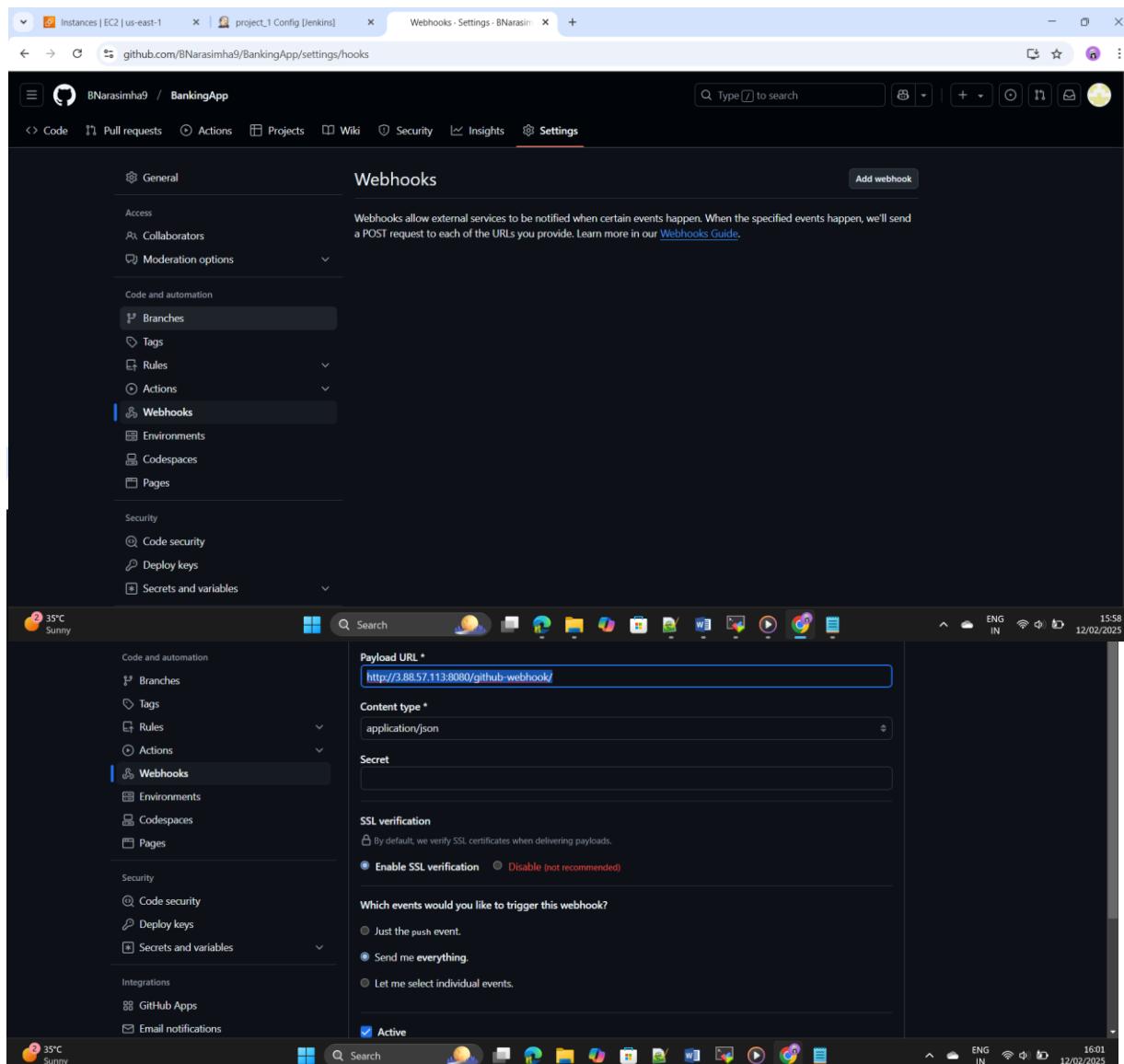


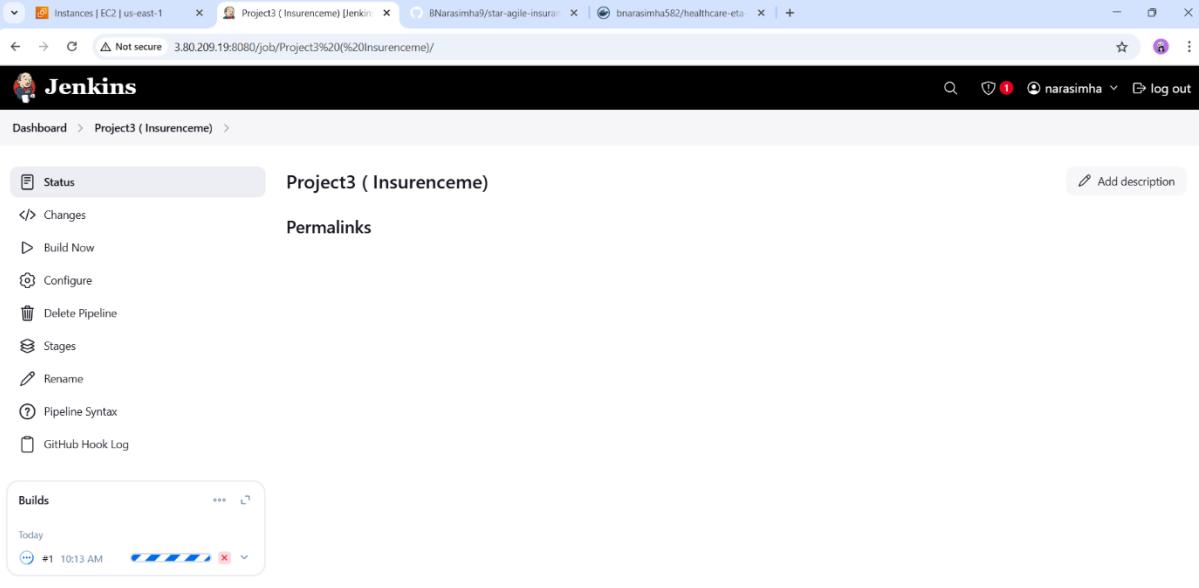
Step73: - Go to triggers in the pipeline and add the github hook.



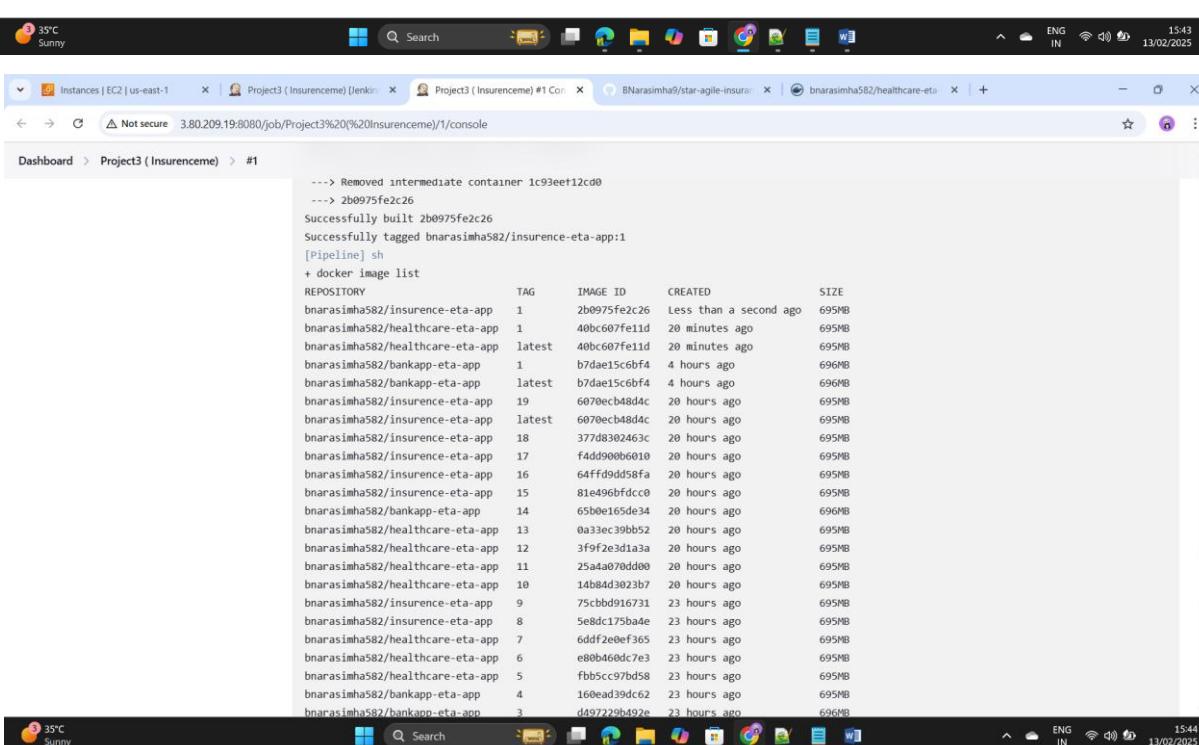
Step74: - Go to the git hub and settings and add the github webhook by using Jenkins url.

Step75: - If any changes are happened in the repository it will automatically updated in the production server by using github webhook.



Step76: - Save and apply the code click on build now.


The screenshot shows the Jenkins Project3 dashboard. The left sidebar contains links for Status, Changes, Build Now, Configure, Delete Pipeline, Stages, Rename, Pipeline Syntax, and GitHub Hook Log. The main area is titled "Project3 (Insurenceme)" and shows a "Builds" section with one entry: "#1 10:13 AM". A progress bar indicates the build is in progress. The status bar at the bottom right shows "Jenkins 2.492.1".



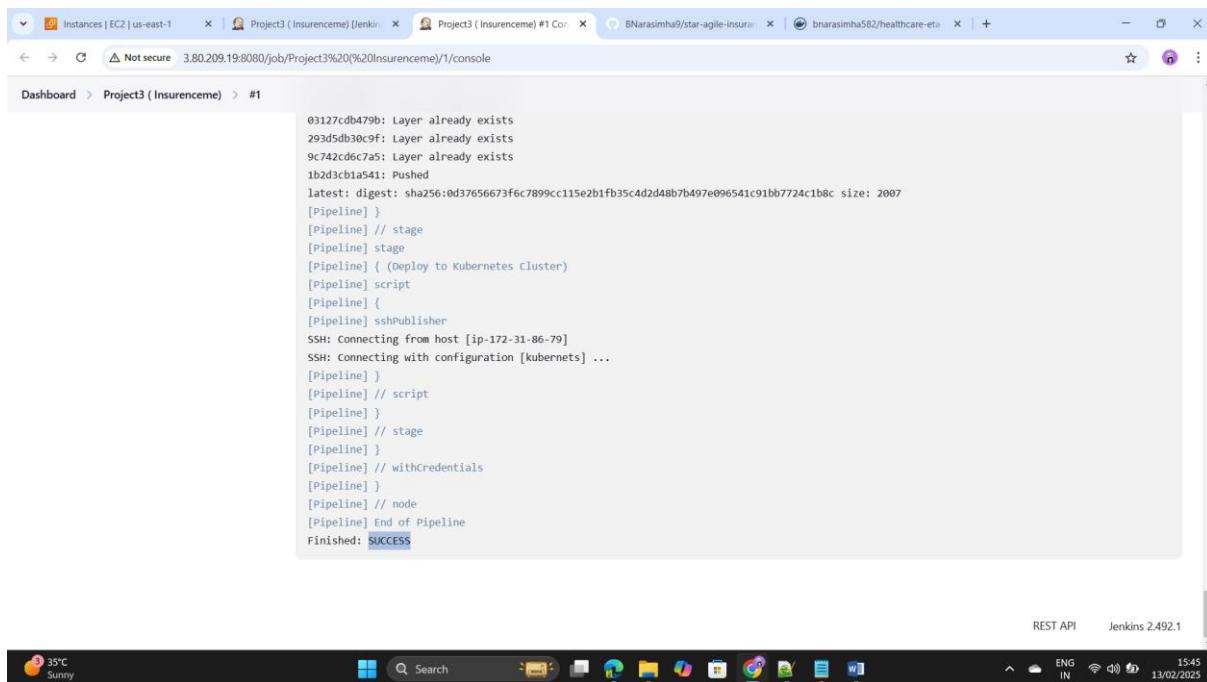
The screenshot shows a Windows desktop with a Jenkins build log window open. The log output is as follows:

```

---> Removed intermediate container 1c93eef12cd0
--> 2b0975fe2c26
Successfully built 2b0975fe2c26
Successfully tagged bnarasimha582/insurance-eta-app:1
[Pipeline] sh
+ docker image list
REPOSITORY          TAG      IMAGE ID   CREATED        SIZE
bnarasimha582/insurance-eta-app  1        2b0975fe2c26  Less than a second ago  695MB
bnarasimha582/healthcare-eta-app  1        40bc607fe11d  20 minutes ago  695MB
bnarasimha582/healthcare-eta-app  latest   40bc607fe11d  20 minutes ago  695MB
bnarasimha582/bankapp-eta-app    1        b7daef15c6bf4  4 hours ago   696MB
bnarasimha582/bankapp-eta-app    latest   b7daef15c6bf4  4 hours ago   696MB
bnarasimha582/insurance-eta-app  19       6070e0cb48d4c  20 hours ago   695MB
bnarasimha582/insurance-eta-app  latest   6070e0cb48d4c  20 hours ago   695MB
bnarasimha582/insurance-eta-app  18       377d8302463c  20 hours ago   695MB
bnarasimha582/insurance-eta-app  17       f4dd900b6010  20 hours ago   695MB
bnarasimha582/insurance-eta-app  16       64ffd9dd5fa  20 hours ago   695MB
bnarasimha582/insurance-eta-app  15       81e496b7dc0  20 hours ago   695MB
bnarasimha582/bankapp-eta-app    14       65b0e0165de34 20 hours ago   696MB
bnarasimha582/healthcare-eta-app 13       0a3e0c39bb52  20 hours ago   695MB
bnarasimha582/healthcare-eta-app 12       3f9f2e3d1a3a  20 hours ago   695MB
bnarasimha582/healthcare-eta-app 11       25aa4070dd00  20 hours ago   695MB
bnarasimha582/healthcare-eta-app 10       14b84d3023b7  20 hours ago   695MB
bnarasimha582/insurance-eta-app  9        75cbbd916731  23 hours ago   695MB
bnarasimha582/insurance-eta-app  8        5e8dc175ba4e  23 hours ago   695MB
bnarasimha582/healthcare-eta-app 7        6ddf2e0eef365 23 hours ago   695MB
bnarasimha582/healthcare-eta-app 6        e80b460dc7e3  23 hours ago   695MB
bnarasimha582/healthcare-eta-app 5        fbb5cc97bd58  23 hours ago   695MB
bnarasimha582/bankapp-eta-app    4        160ead39dc62  23 hours ago   695MB
bnarasimha582/bankapp-eta-app    3        d497229b492e  23 hours ago   696MB

```

The status bar at the bottom right shows "15:44" and "13/02/2025".



```

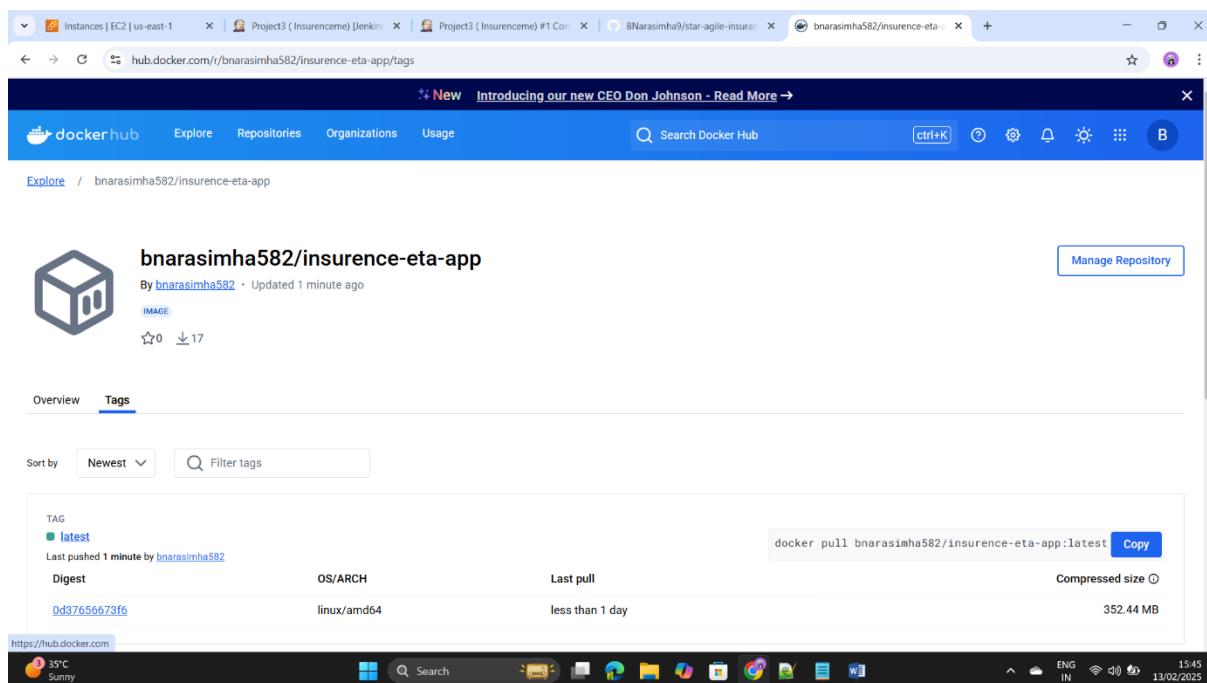
03127cdb479b: Layer already exists
293d5db30c9f: Layer already exists
9c742cd6c7a5: Layer already exists
1b2d3cb1a541: Pushed
latest: digest: sha256:0d37656673f6c7899cc115e2b1fb35c4d2d48b7b497e096541c91bb7724c1b8c size: 2007
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy to Kubernetes Cluster)
[Pipeline] script
[Pipeline] {
[Pipeline] sshpublisher
SSH: Connecting from host [ip-172-31-86-79]
SSH: Connecting with configuration [kubernetes] ...
[Pipeline]
[Pipeline] // script
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withCredentials
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

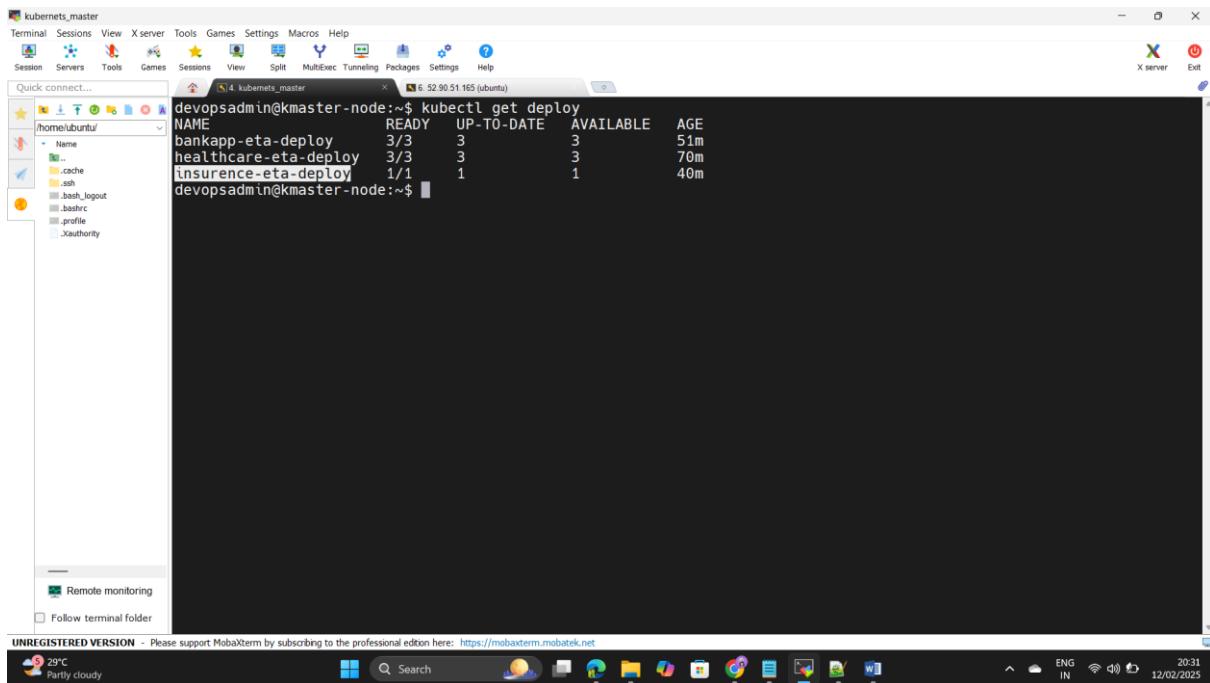
REST API Jenkins 2.492.1

35°C Sunny 15:45 ENG IN 13/02/2025

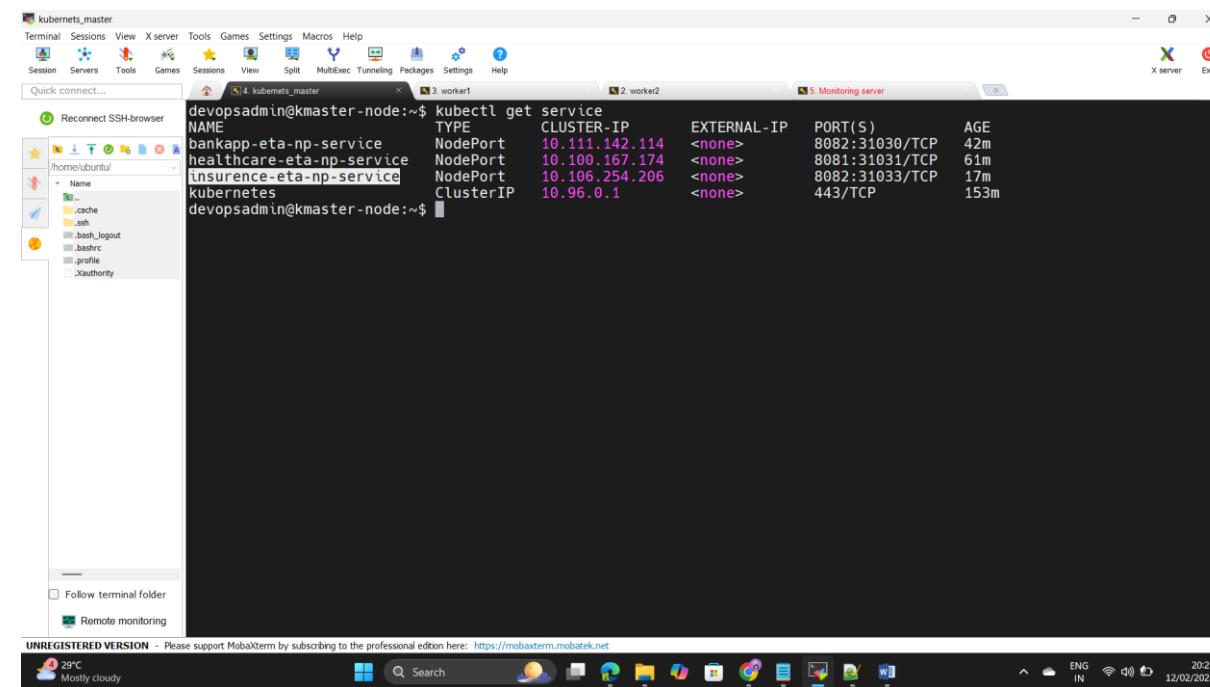
Step77: - Go to the docker hub we can see there a repository has been created as bank-app successfully.



Step78: - Go to the kubernets and we can see the pods are deployed or not by using kubectl get deploy.



```
devopsadmin@kmaster-node:~$ kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
bankapp-eta-deploy  3/3      3          3          51m
healthcare-eta-deploy 3/3      3          3          70m
insurance-eta-deploy 1/1      1          1          40m
devopsadmin@kmaster-node:~$
```



```
devopsadmin@kmaster-node:~$ kubectl get service
NAME            TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
bankapp-eta-np-service  NodePort    10.111.142.114 <none>       8082:31030/TCP   42m
healthcare-eta-np-service  NodePort    10.100.167.174 <none>       8081:31031/TCP   61m
insurance-eta-np-service  NodePort    10.106.254.206 <none>       8082:31033/TCP   17m
kubernetes       ClusterIP  10.96.0.1    <none>        443/TCP    153m
devopsadmin@kmaster-node:~$
```

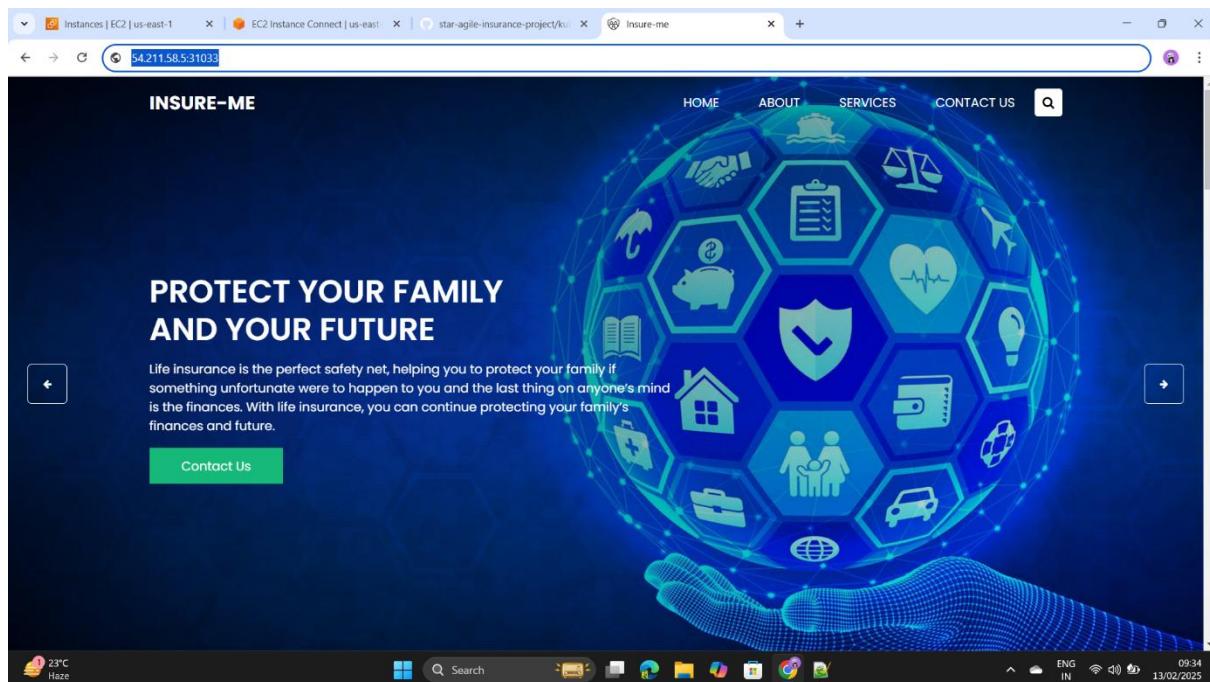
```

devopsadmin@kmaster-node:~$ kubectl get po -o wide
NAME                               READY   STATUS    RESTARTS   AGE     IP           NODE   NOMINA
TED-NODE-READINESS-GATES          1/1    Running   0          100s   10.244.3.2   worker-node3 <none>
bankapp-eta-deploy-784ccf6686-52l52 1/1    Running   0          101s   10.244.2.11   worker-node2 <none>
bankapp-eta-deploy-784ccf6686-55pnq 1/1    Running   0          100s   10.244.2.13   worker-node2 <none>
bankapp-eta-deploy-784ccf6686-6xsn2 1/1    Running   0          100s   10.244.1.12   worker-node1 <none>
bankapp-eta-deploy-784ccf6686-k4cs7 1/1    Terminating   0          13m    10.244.1.14   worker-node1 <none>
bankapp-eta-deploy-784ccf6686-nwpn8 1/1    Terminating   0          13m    10.244.1.15   worker-node1 <none>
healthcare-eta-deploy-6989db948c-55hf 1/1    Running   0          100s   10.244.3.4   worker-node3 <none>
healthcare-eta-deploy-6989db948c-kghhh 1/1    Terminating   0          13m    10.244.1.10   worker-node1 <none>
healthcare-eta-deploy-6989db948c-lbdnt 1/1    Terminating   0          13m    10.244.1.11   worker-node1 <none>
healthcare-eta-deploy-6989db948c-pt4wd 1/1    Running   0          100s   10.244.2.14   worker-node2 <none>
healthcare-eta-deploy-6989db948c-sq99c 1/1    Terminating   0          13m    10.244.1.8   worker-node1 <none>
healthcare-eta-deploy-6989db948c-xd6wv 1/1    Running   0          101s   10.244.2.12   worker-node2 <none>
Insurence-eta-deploy-c8dfcf869-gmqx4 1/1    Running   0          101s   10.244.3.3   worker-node3 <none>
Insurence-eta-deploy-c8dfcf869-r9rn8 1/1    Terminating   0          13m    10.244.1.13   worker-node1 <none>
devopsadmin@kmaster-node:~$ 

```

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

Step79: - Copt the Ip address of the Workernode and paste in the browser as a node port as mentioned above.



Reprehenderit tenetur eos
alteration in some form, by injected humour, or randomised words which don't look even slightly believable.

Numquam suscipit maxime
alteration in some form, by injected humour, or randomised words which don't look even slightly believable.

Pariatur deserunt tempora
alteration in some form, by injected humour, or randomised words which don't look even slightly believable.

28 Sep 2020 Andrew James Read More

29 Sep 2020 Andrew James Read More

30 Sep 2020 Andrew James Read More

[View More](#)

INSUDOR

Soluta odit exercitationem rerum operiam eos consectetur impedit delectus qui reiciendis, distinctio, asperiores fuga labore Magni natus.

[f](#) [t](#) [in](#) [y](#)

CONTACT US

Location: Call : +01 123455678990 Email : demo@gmail.com

SIGN UP TO OUR NEWSLETTER

Enter Your Email

[Subscribe](#)

23°C Haze

ABOUT US

Insure-Me is a leading multi-line insurer that serves its customers in global and local markets. With about 56,000 employees, it provides a wide range of property and casualty, life insurance products and services in more than 210 countries and territories. Zurich's customers include individuals, small businesses, and mid-sized and large companies, as well as multinational corporations.

[Read More](#)

BEST INSURANCE FOR YOUR FAMILY

There are many life insurance companies that provide life insurance plans globally. Life insurance is an agreement between an individual and the insurance company under which the insurance company promises to provide a death benefit to the family of life assured in case of an unfortunate demise of the life assured during the policy. We offer the best in class services.

23°C Haze

Step80: - If any deployments are or changes happen in the git hub repository by using github webhook it will automatically trigger the changes and tested and build in the build server and if the build is

successful it is pushed to the production server the refence img should be saved in the docker hub , All these things are happen automatically by using CICD pipeline in DevOps as done as successfully.

Step81: - Now we have test the production servers are working properly or monitor the servers by using Prometheus and Grafana.

Step82: - Then Install the Prometheus in the another ec2 instance or use the same build server for the monitoring.

The screenshot shows the AWS Management Console with the EC2 service selected. The left navigation pane is open, showing options like Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, and Network & Security. The main content area is titled 'Instances (1) Info' and displays a single instance named 'Monitoring'. The instance details are as follows:

Attribute	Value
Instance ID	i-00f4e3783288e81aa
Instance state	Running
Instance type	t2.micro
Status check	Initializing
Alarm status	View alarms +
Availability Zone	us-east-1a

Below the table, there is a section titled 'Select an instance' which lists the instance 'Monitoring'.

Step83: - Install the by using below

url< <https://prometheus.io/download/>>.

DOWNLOAD

We provide precompiled binaries and Docker images for most officially maintained Prometheus components. If a component is not listed here, check the respective repository on Github for further instructions.

There is also a constantly growing number of independently maintained exporters listed at Exporters and integrations.

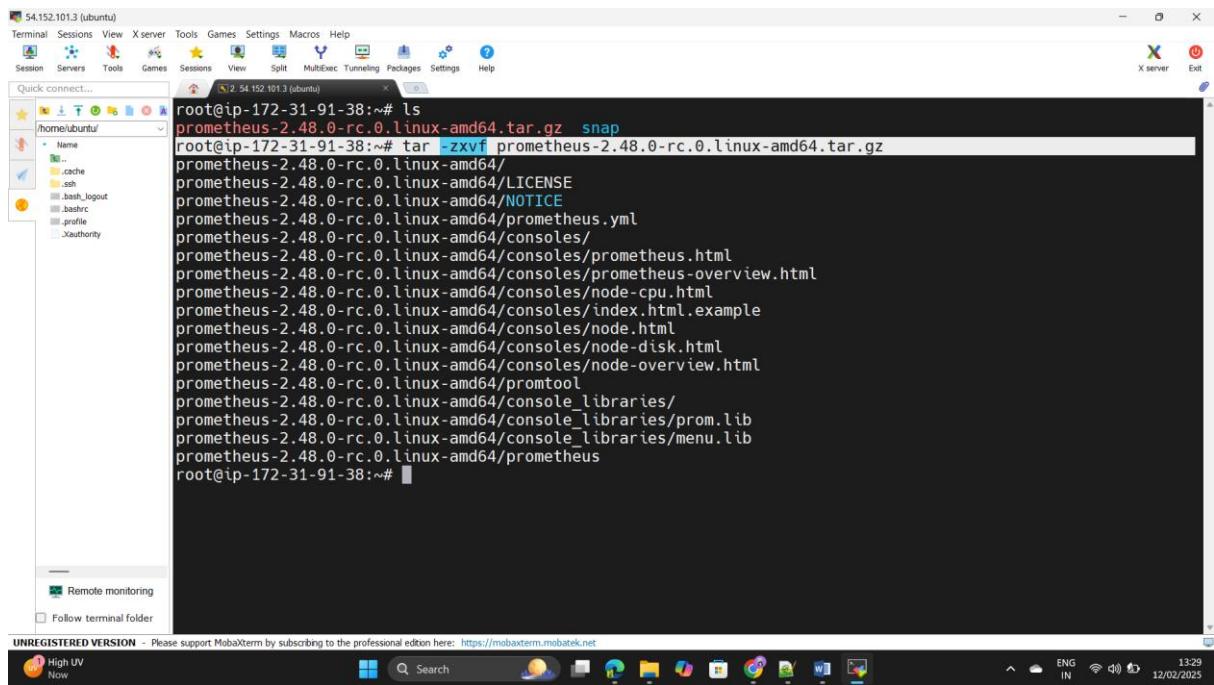
- prometheus
- alertmanager
- blackbox_exporter
- consul_exporter
- graphite_exporter
- memcached_exporter
- mysqld_exporter
- node_exporter
- promlens
- pushgateway
- statsd_exporter

Operating system	popular	Architecture	popular	
prometheus				
The Prometheus monitoring system and time series database. prometheus/prometheus				
3.2.0-rc.1 / 2025-01-29 Pre-release Release notes				
File name	OS	Arch	Size	SHA256 Checksum
prometheus-3.2.0-rc.1.darwin-amd64.tar.gz	darwin	amd64	110.48 MiB	6a1b4d1481c9a7991058fd8134cfcb3d4116f13806abe329c19ce697bd08304
prometheus-3.2.0-rc.1.darwin-arm64.tar.gz	darwin	arm64	106.85 MiB	5f8db2724caeac4d90a198416cceaa1921f39e9fe0d54d7edf73919f117ae81
prometheus-3.2.0-rc.1.linux-amd64.tar.gz	linux	amd64	108.89 MiB	21fb63a6b0dd004c8623ec130d687ca6546ee04a45f0e01095c1fe504e558207

Step84: - Copt the link address and paste in the instance as use the command as – wget.

```
root@ip-172-31-91-38:~# wget https://github.com/prometheus/prometheus/releases/download/v2.48.0-rc.0/prometheus-2.48.0-rc.0.linux-amd64.tar.gz
```

Step85: - Extract the file by using the tar -zxvf <file name>.

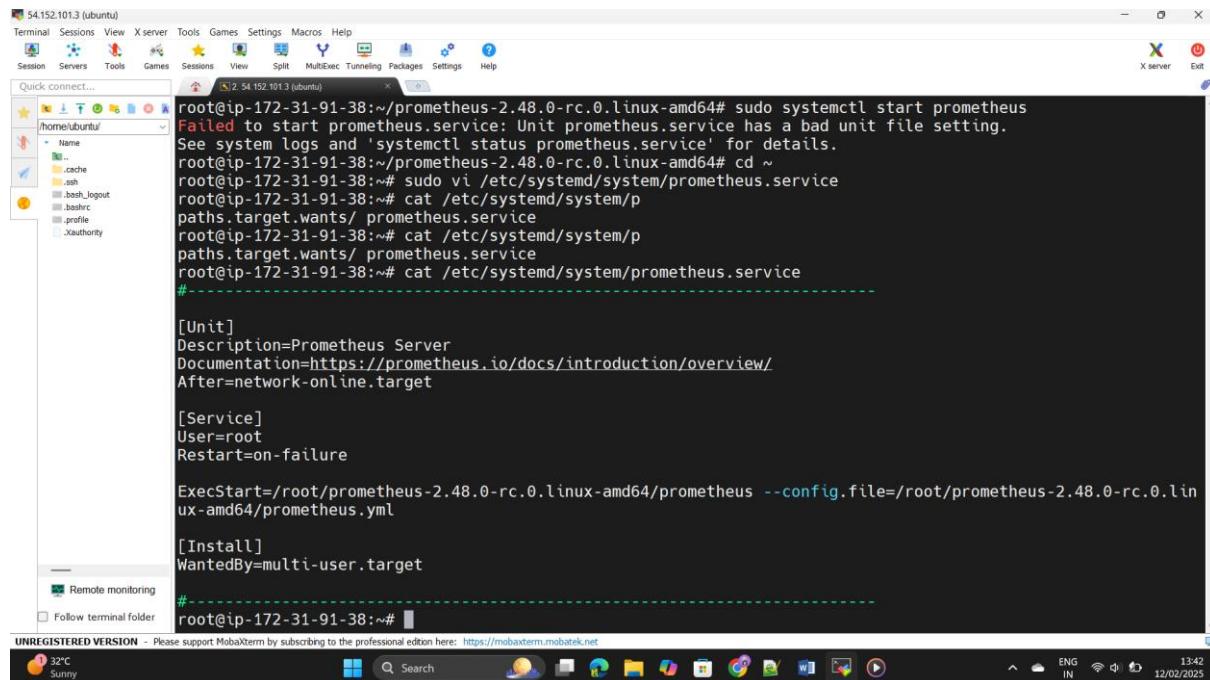


The screenshot shows a terminal window in MobaXterm connected to IP 54.152.101.3 (ubuntu). The user is root. They have extracted the Prometheus tar.gz file into the current directory:

```
root@ip-172-31-91-38:~# ls
prometheus-2.48.0-rc.0.linux-amd64.tar.gz  snap
root@ip-172-31-91-38:~# tar -zxf prometheus-2.48.0-rc.0.linux-amd64.tar.gz
prometheus-2.48.0-rc.0.linux-amd64/
prometheus-2.48.0-rc.0.linux-amd64/LICENSE
prometheus-2.48.0-rc.0.linux-amd64/NOTICE
prometheus-2.48.0-rc.0.linux-amd64/prometheus.yml
prometheus-2.48.0-rc.0.linux-amd64/consoles/
prometheus-2.48.0-rc.0.linux-amd64/consoles/prometheus.html
prometheus-2.48.0-rc.0.linux-amd64/consoles/prometheus-overview.html
prometheus-2.48.0-rc.0.linux-amd64/consoles/node-cpu.html
prometheus-2.48.0-rc.0.linux-amd64/consoles/index.html.example
prometheus-2.48.0-rc.0.linux-amd64/consoles/node.html
prometheus-2.48.0-rc.0.linux-amd64/consoles/node-disk.html
prometheus-2.48.0-rc.0.linux-amd64/consoles/node-overview.html
prometheus-2.48.0-rc.0.linux-amd64/promtool
prometheus-2.48.0-rc.0.linux-amd64/console_libraries/
prometheus-2.48.0-rc.0.linux-amd64/console_libraries/prom.lib
prometheus-2.48.0-rc.0.linux-amd64/console_libraries/menu.lib
prometheus-2.48.0-rc.0.linux-amd64/prometheus
root@ip-172-31-91-38:~#
```

Step86: - Create the file for the Prometheus activation as a service use the command as –

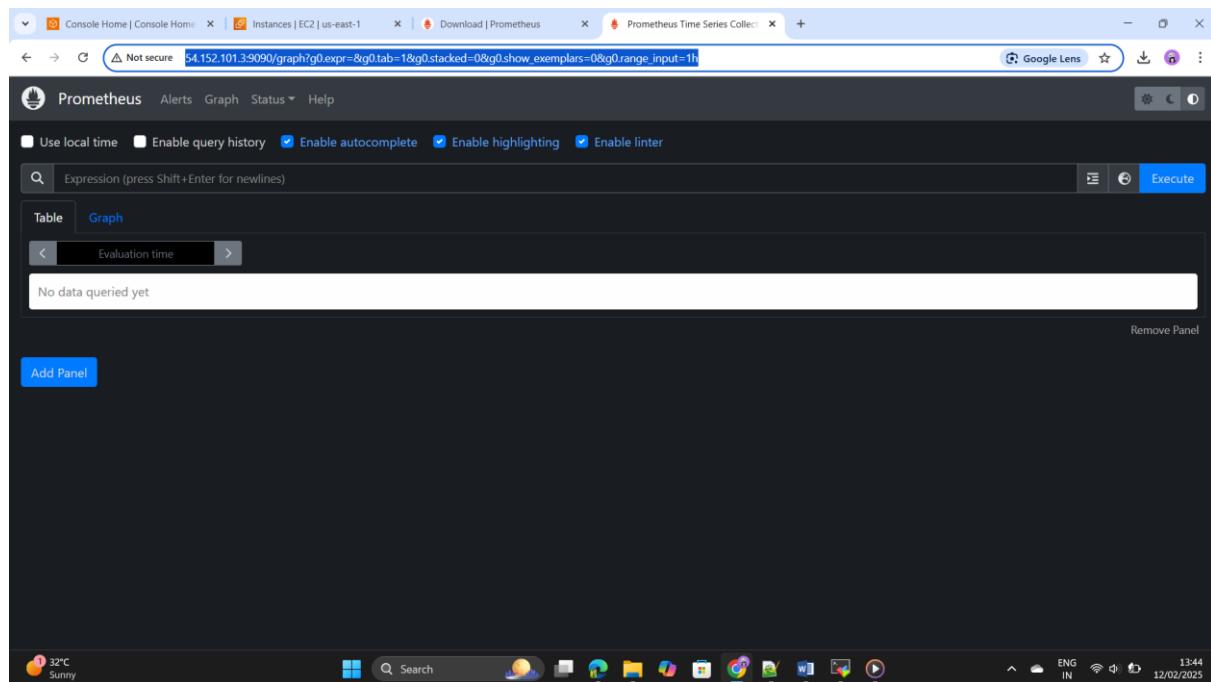
vi /etc/systemd/system/prometheus.service.



The screenshot shows a terminal window in MobaXterm connected to IP 54.152.101.3 (ubuntu). The user is root. They are creating a new systemd service file for Prometheus:

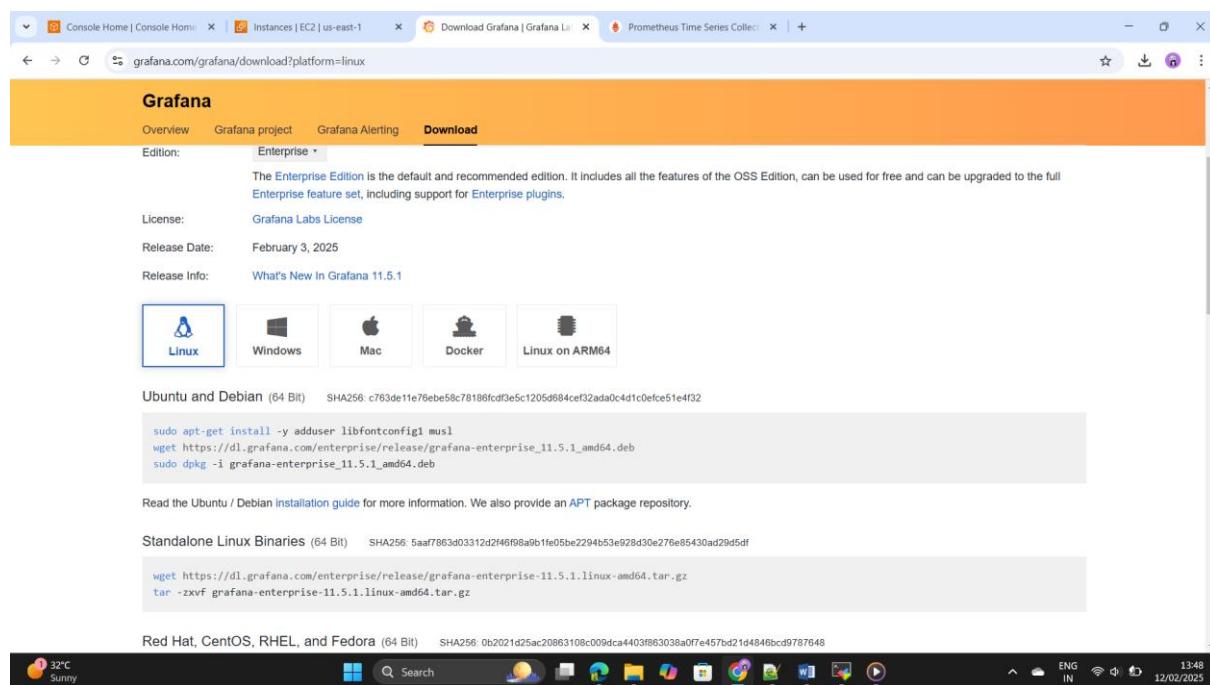
```
root@ip-172-31-91-38:~/prometheus-2.48.0-rc.0.linux-amd64# sudo systemctl start prometheus
Failed to start prometheus.service: Unit prometheus.service has a bad unit file setting.
See system logs and 'systemctl status prometheus.service' for details.
root@ip-172-31-91-38:~/prometheus-2.48.0-rc.0.linux-amd64# cd ~
root@ip-172-31-91-38:~# sudo vi /etc/systemd/system/prometheus.service
root@ip-172-31-91-38:~# cat /etc/systemd/system/p
paths.target.wants/ prometheus.service
root@ip-172-31-91-38:~# cat /etc/systemd/system/p
paths.target.wants/ prometheus.service
root@ip-172-31-91-38:~# cat /etc/systemd/system/prometheus.service
#-----[Unit]-----[Unit]
Description=Prometheus Server
Documentation=https://prometheus.io/docs/introduction/overview/
After=network-online.target
[Service]
User=root
Restart=on-failure
ExecStart=/root/prometheus-2.48.0-rc.0.linux-amd64/prometheus --config.file=/root/prometheus-2.48.0-rc.0.linux-amd64/prometheus.yml
[Install]
WantedBy=multi-user.target
#-----[Unit]-----[Unit]
root@ip-172-31-91-38:~#
```

Step87: - Add the unit file in that and run the Prometheus by using port number 9090.



Step88: - Install grafana in the same instance,click on the below url

<https://grafana.com/grafana/download.>



Step89: - Install the grafana by using command as wget<url>.

```
root@ip-172-31-91-38:~# wget https://dl.grafana.com/oss/release/grafana-9.1.2-1.x86_64.rpm
--2025-02-12 08:19:03-- https://dl.grafana.com/oss/release/grafana-9.1.2-1.x86_64.rpm
Resolving dl.grafana.com (dl.grafana.com)... 146.75.34.217, 2a04:4e42:78::729
Connecting to dl.grafana.com (dl.grafana.com)|146.75.34.217|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 80551209 (77M) [application/octet-stream]
Saving to: 'grafana-9.1.2-1.x86_64.rpm'

grafana-9.1.2-1.x86_64.rpm 100%[=====] 76.82M 100MB/s in 0.8s
2025-02-12 08:19:04 (100 MB/s) - 'grafana-9.1.2-1.x86_64.rpm' saved [80551209/80551209]
root@ip-172-31-91-38:~#
```

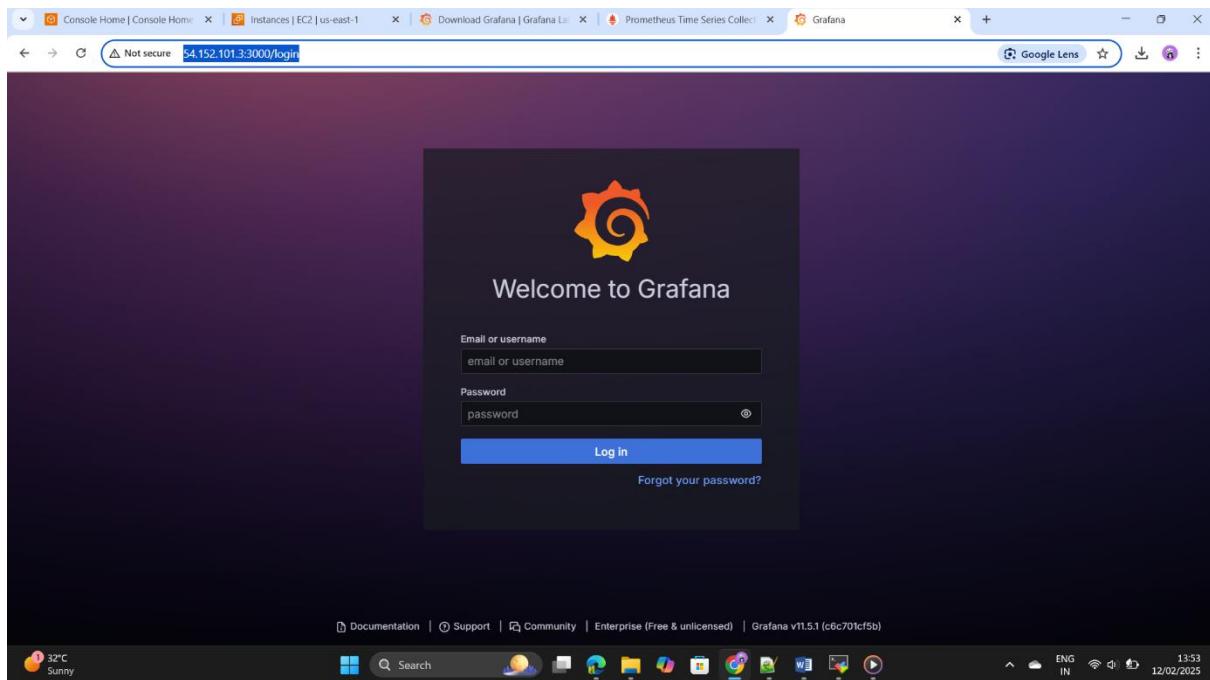
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

```
Synchronizing state of grafana-server.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable grafana-server
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service → /lib/systemd/system/grafana-server.service.
root@ip-172-31-91-38:~# sudo /bin/systemctl start grafana-server.service
root@ip-172-31-91-38:~# sudo /bin/systemctl status grafana-server.service
● grafana-server.service - Grafana instance
    Loaded: loaded (/lib/systemd/system/grafana-server.service; enabled; vendor preset: enabled)
      Active: active (running) since Wed 2025-02-12 08:20:50 UTC; 4s ago
        Docs: http://docs.grafana.org
          Main PID: 5332 (grafana)
            Tasks: 5 (limit: 1130)
           Memory: 86.8M
             CPU: 897ms
           CGroup: /system.slice/grafana-server.service
                   └─5332 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/gra

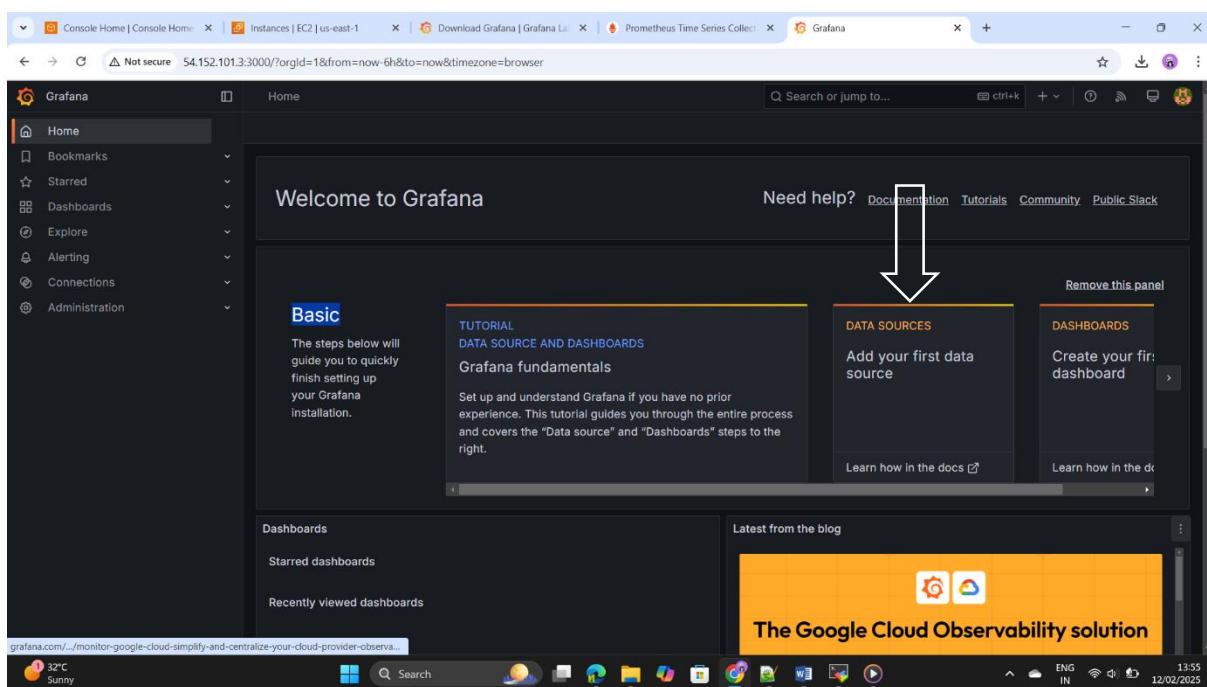
Feb 12 08:20:55 ip-172-31-91-38 grafana[5332]: logger=migrator t=2025-02-12T08:20:55.36223567Z level=info m>
Feb 12 08:20:55 ip-172-31-91-38 grafana[5332]: logger=migrator t=2025-02-12T08:20:55.363072991Z level=info >
Feb 12 08:20:55 ip-172-31-91-38 grafana[5332]: logger=migrator t=2025-02-12T08:20:55.371740592Z level=info >
Feb 12 08:20:55 ip-172-31-91-38 grafana[5332]: logger=migrator t=2025-02-12T08:20:55.372302449Z level=info >
Feb 12 08:20:55 ip-172-31-91-38 grafana[5332]: logger=migrator t=2025-02-12T08:20:55.380521878Z level=info >
Feb 12 08:20:55 ip-172-31-91-38 grafana[5332]: logger=migrator t=2025-02-12T08:20:55.404322395Z level=info >
Feb 12 08:20:55 ip-172-31-91-38 grafana[5332]: logger=migrator t=2025-02-12T08:20:55.41381983Z level=info >
Feb 12 08:20:55 ip-172-31-91-38 grafana[5332]: logger=migrator t=2025-02-12T08:20:55.422667034Z level=info >
Feb 12 08:20:55 ip-172-31-91-38 grafana[5332]: logger=migrator t=2025-02-12T08:20:55.435605117Z level=info >
Feb 12 08:20:55 ip-172-31-91-38 grafana[5332]: logger=migrator t=2025-02-12T08:20:55.435714531Z level=info >
[Lines 1-21/21 (END)]
```

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

Step90: - Copy paste the Public Ip address of the instance as port number is 3000 we can see grafana is started.



Step91: - Integrate both Prometheus in the grafana click on the data source and add the Prometheus url and save it.



Step92: - Prometheus and Grafana integration is done successfully.

Step93: - ADD the target servers in the Prometheus for the monitoring.

The screenshot shows a browser window with two tabs: 'Instances | EC2 | us-east-1' and 'Download | Prometheus'. The 'Download | Prometheus' tab is active, displaying a table of available exporters. An arrow points to the 'node_exporter-1.8.2.darwin-amd64.tar.gz' link in the table.

	OS	Arch	Size	SHA256 Checksum
mysql_exporter-0.16.0.linux-amd64.tar.gz	linux	amd64	9.02 MIB	32fe0b59ef3f52624a1918aaef6b8855f27c2d492a7026d2a9750bd251be209d
mysql_exporter-0.16.0.windows-amd64.zip	windows	amd64	9.17 MIB	cdb0e31fe3f79076e086ade34a7e79f89c532d31ea9d3792969f46354c222c08
node_exporter				
Exporter for machine metrics prometheus/node_exporter				
1.8.2 / 2024-06-19 Release notes				
File name	OS	Arch	Size	SHA256 Checksum
node_exporter-1.8.2.darwin-amd64.tar.gz	darwin	amd64	4.83 MIB	97ad998fe48094a8b085f2c5adc6a1eed05309f5fc29baed2e94ee2e73cf1728
node_exporter-1.8.2.darwin-arm64.tar.gz	darwin	arm64	4.44 MIB	1b56711d389116d086dd414240e6fbefc6bdec6a710fer16771c113a1f7e83c
node_exporter-1.8.2.linux-amd64.tar.gz	linux	amd64	10.18 MIB	6809ad00b3e45f0de992c19871d6b5253eed5ead7rf9686885s51d85c6643c66

promlens

PromLens – The query builder, analyzer, and explainer for PromQL [prometheus/promlens](#)

File name	OS	Arch	Size	SHA256 Checksum
promlens-0.3.0.darwin-amd64.tar.gz	darwin	amd64	16.90 MIB	a90fe174d57fe4ecf0456db2a7161de3d49f72ce8cdc312855497c89f442b02c0
promlens-0.3.0.darwin-arm64.tar.gz	darwin	arm64	16.25 MIB	bff5696e41d11dab37edac8941c070cc879ae0e22a484483685b7fc53e2108b229a
promlens-0.3.0.linux-amd64.tar.gz	linux	amd64	17.15 MIB	8fdcc621cf559b7e55c0e3cf33408662ae0f53cf999cf5d7283d2841f62e9f

Windows taskbar at the bottom showing various icons and system status.

Step94: - Add the link in the production servers as kubernets Workernodes by using wget command& extract the file by using tar command.

```
root@worker-node2:~# wget https://github.com/prometheus/node_exporter/releases/download/v1.4.0-rc.0/node_exporter-1.4.0-rc.0.linux-amd64.tar.gz
```

Step95: - Edit the following file and add the service file in that and save it.

```
root@worker-node1:~# cat /etc/systemd/system/node_exporter.service
[Unit]
Description=Prometheus Server
Documentation=https://prometheus.io/docs/introduction/overview/
After=network-online.target

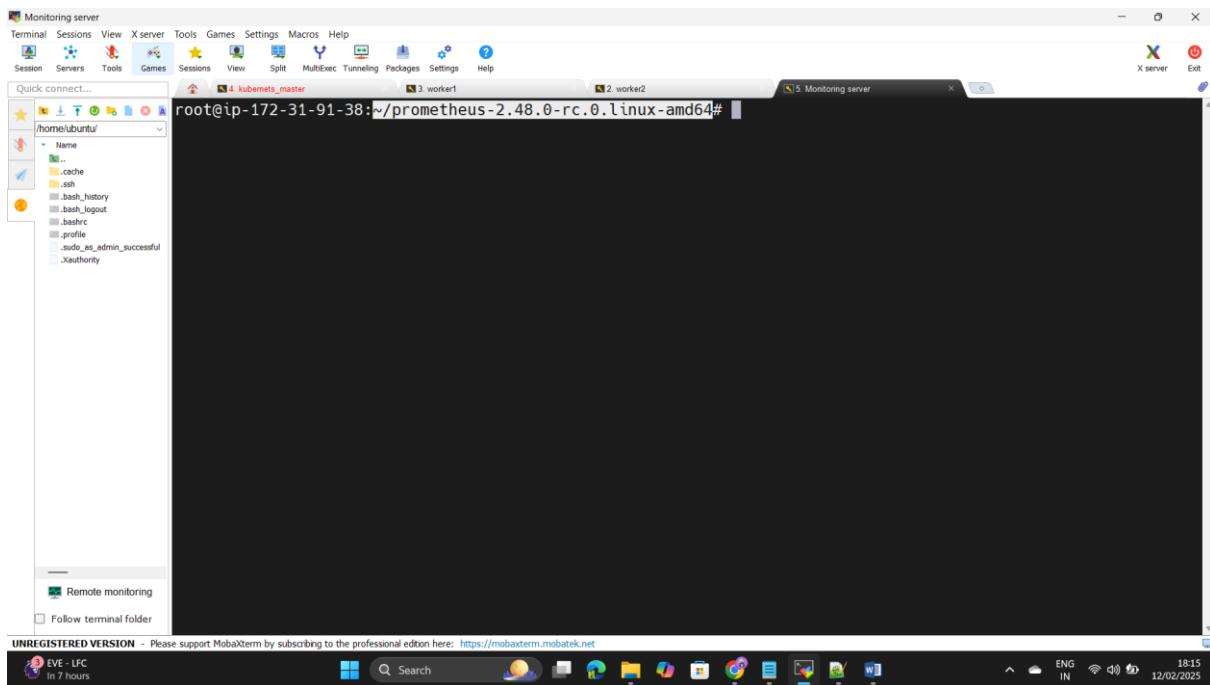
[Service]
User=root
Restart=on-failure

ExecStart=/root/node_exporter-1.4.0-rc.0.linux-amd64/node_exporter

[Install]
WantedBy=multi-user.target
#-----
```

Step96: - Go to the monitoring server and go to the directory

```
cd /root/prometheus-2.38.0.linux-amd64.
```



Step97: - Edit the vi prometheus.yml this file and add the private ip address of the 2 target servers and save it.

```
# scrape_timeout is set to the global default (10s).
# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
            # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

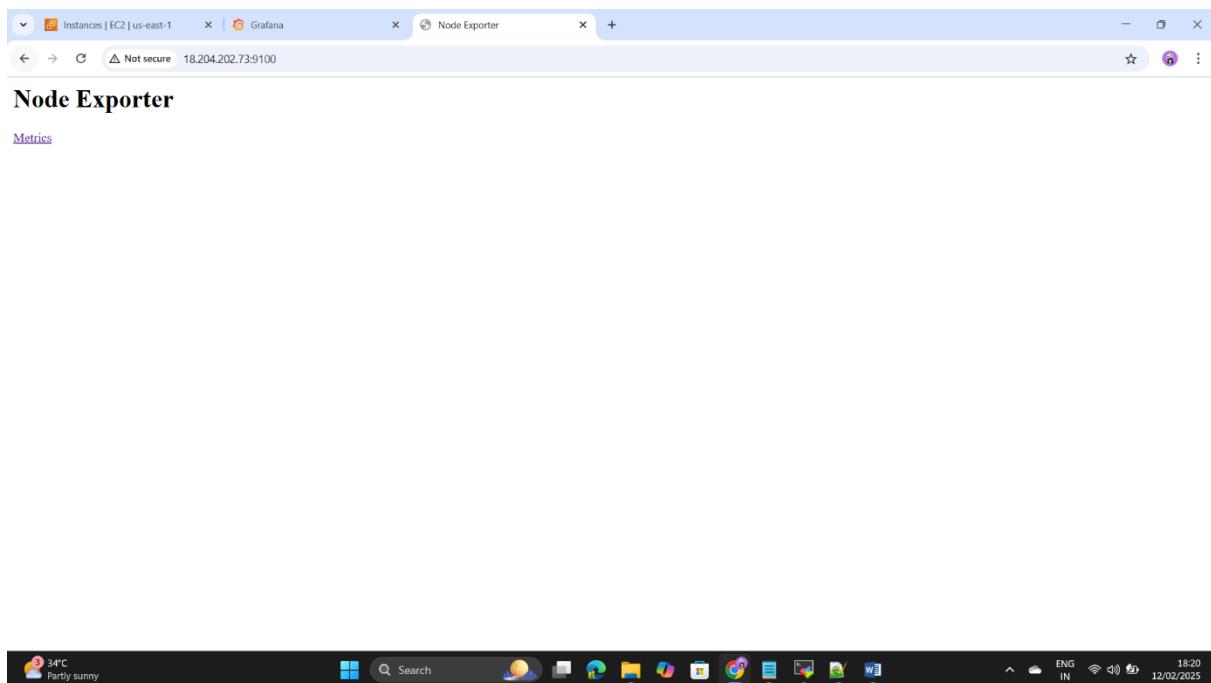
# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

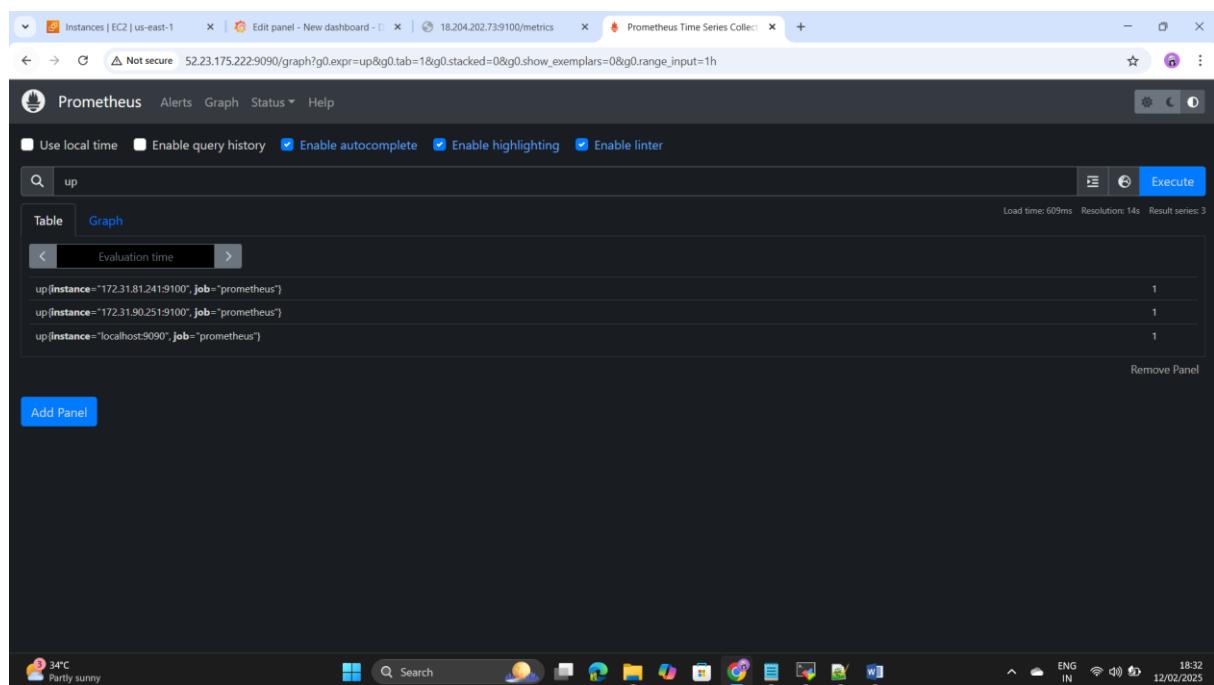
    static_configs:
      - targets: ["localhost:9090"]
      - targets: ["172.31.90.251:9100"]
      - targets: ["172.31.81.241:9100"]

-- INSERT --
```

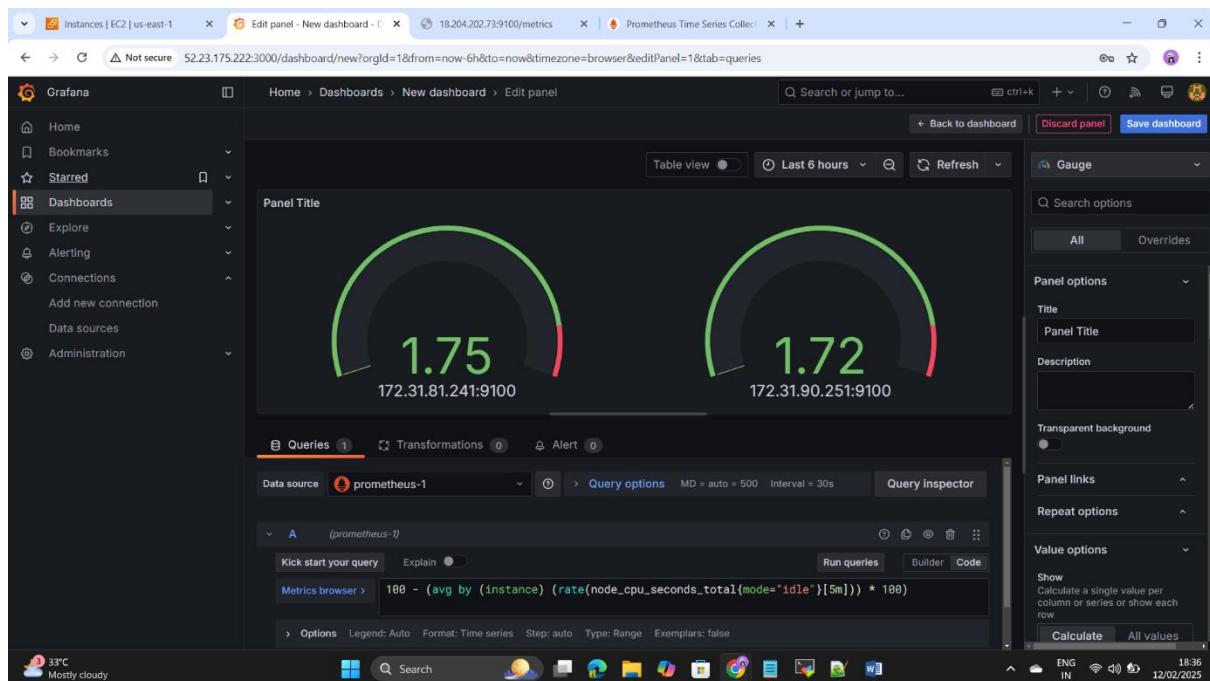
Step98: - copy paste the Target server ip address with port as 9100.



Step99: - We can see the target servers in the Prometheus.

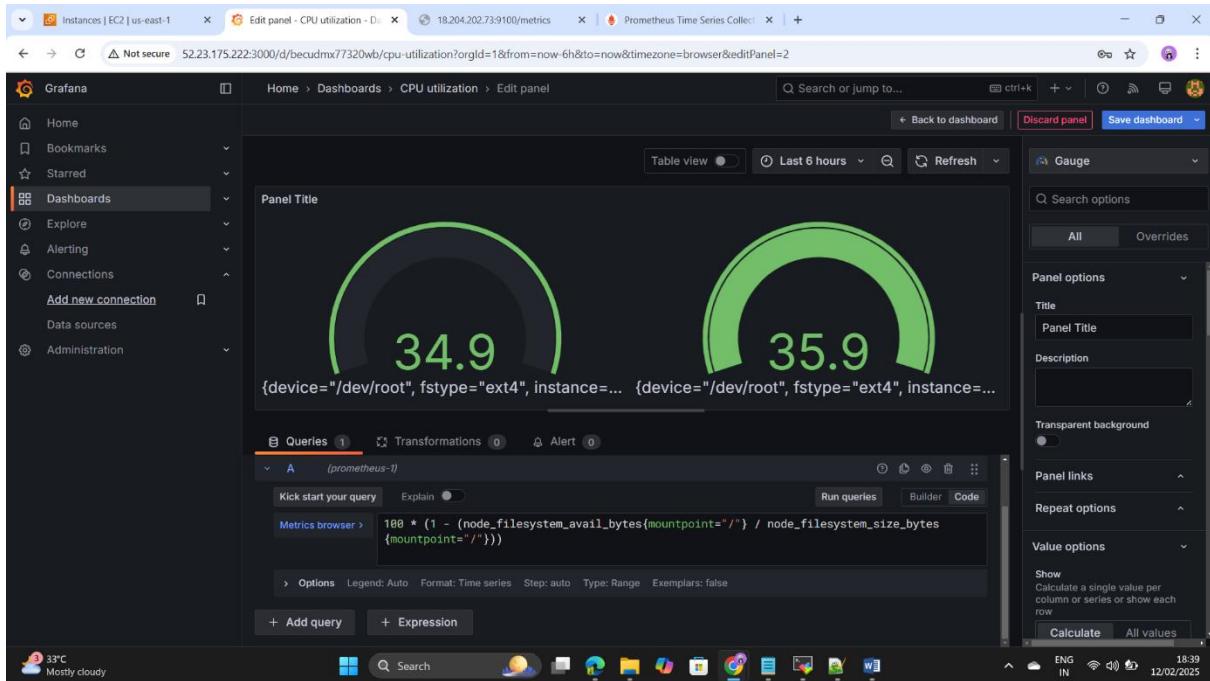


Step100: - Go to the grafana and add this code < 100 - (avg by (instance) (rate(node_cpu_seconds_total{mode="idle"}[5m])) * 100)> for the CPU utilization.



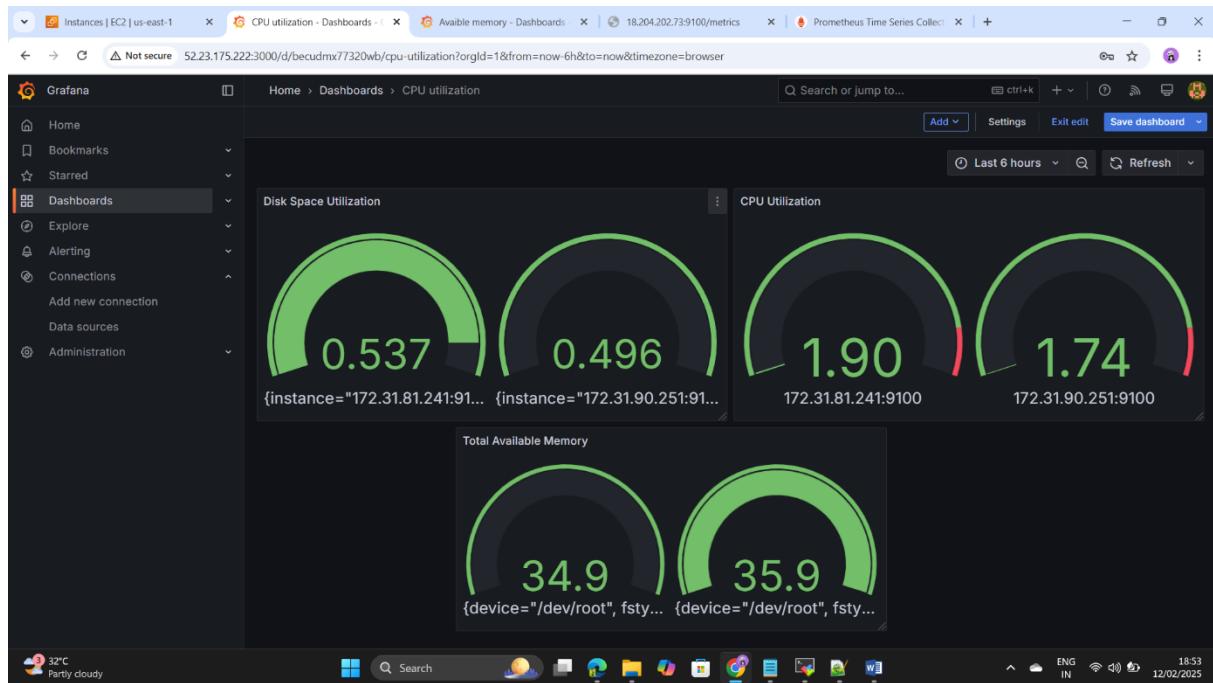
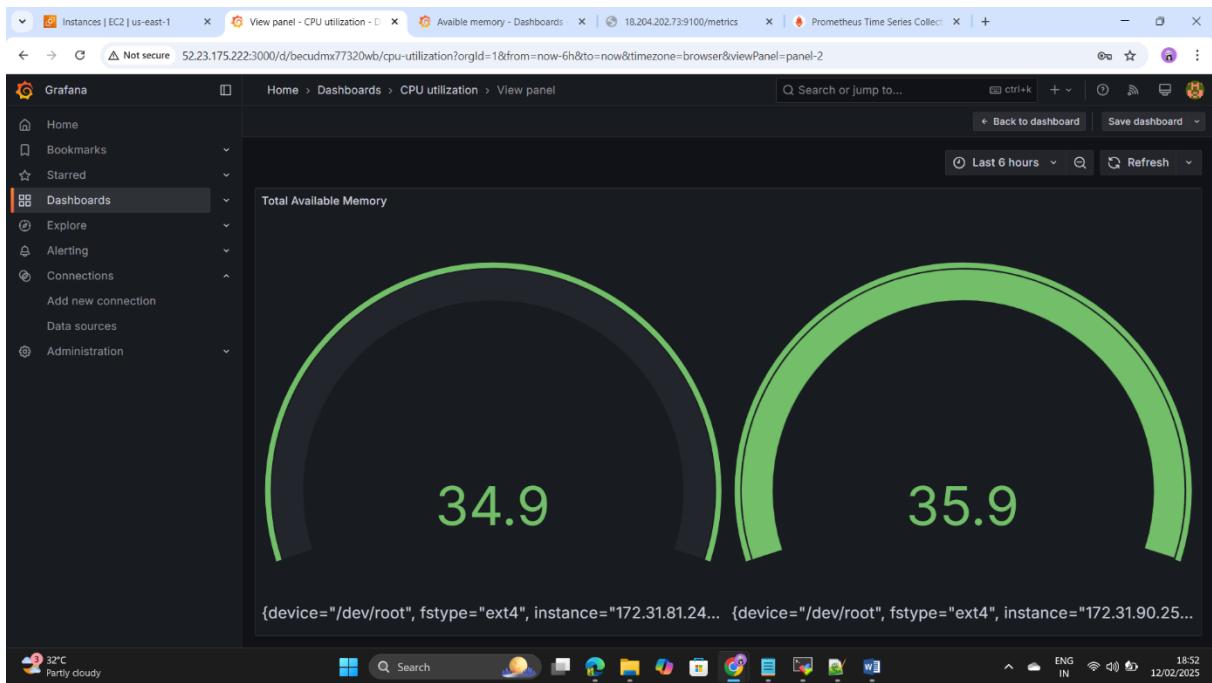
Step101: - Add the below code for the disk space utilization < 100 *

```
(1 - (node_filesystem_avail_bytes{mountpoint="/"}/
node_filesystem_size_bytes{mountpoint="/"}))>.
```



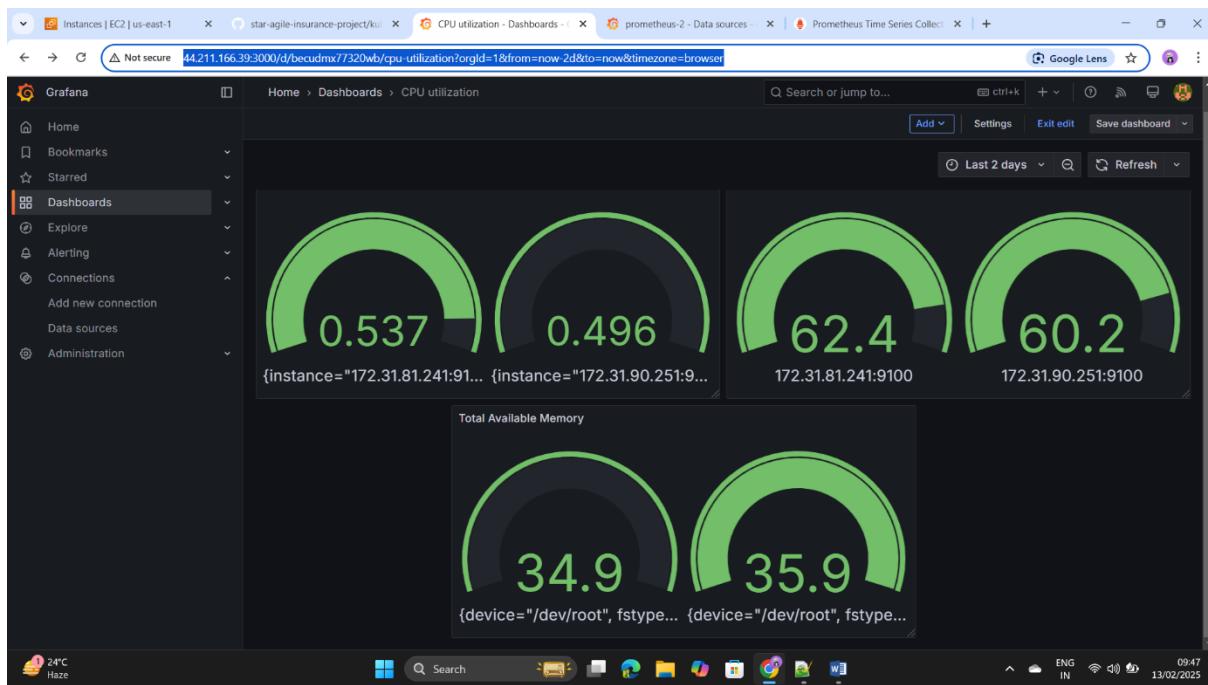
Step102: - Add the below code for the Total available memory.

```
100 - (avg by (instance)
(rate(node_cpu_seconds_total{mode="idle"}[5m])) * 100)
```



Step103:- Click on Refresh the dashboard we can see the

- 1.Disk Space utilization
- 2.CPU Utilization
3. Total Available Memory



Step104:- Monitoring of the test of the production servers ,creating the dashboards has been done successfully.

Step105:- Insurance Me project has been done Successfully.

