

MODULE NAME
CAPSTONE PROJECT

PROJECT NAME

HEALTHCARE-DOMAIN

SUBMITTED BY : SAIKUMAR NASIGAR

DATE OF SUBMISSION : 26-05-2025

Business challenge/requirement

As soon as the developer pushes the updated code on the GIT master branch, the Jenkins pipeline should be triggered and code should be checkout, compiled, tested, packaged and containerized. A new test-cluster should be provisioned and configured automatically with all the required software's and as soon as the cluster is healthy and available, the application must be deployed to the test-server automatically using Kubernetes.

The deployment should then be tested using a test automation tool, and if the build is successful, it should be deployed to the prod server/cluster using Kubernetes.

All this should happen automatically and should be triggered from a push to the GitHub master branch. Kubernetes cluster must contain at least 2 servers and must be monitored continuously using Prometheus and dashboard must be visualized using Grafana.

Tech stack

- ✓ Git - For version control for tracking changes in the code files
- ✓ Jenkins - For continuous integration and continuous deployment
- ✓ Docker - For containerizing applications
- ✓ Ansible - Configuration management tools
- ✓ Selenium - For automating tests on the deployed web application
- ✓ Terraform - For creation of infrastructure.
- ✓ Kubernetes – for running containerized application in managed cluster

PROCEDURE:

Provision 5 EC2 Ubuntu instances with Terraform:

- host-master (t3.medium) – Jenkins, Ansible, K8s Master
- production-1, production-2 (t3.medium) – K8s Prod Nodes
- test-1 (t2.medium) – K8s Test Node
- monitoring (t3.medium) – Prometheus & Grafana

Create a project folder like Healthcare-Infra in VS Code and inside it:

- authenticate.tf – for AWS provider and credentials
- variables.tf – to define instance names, AMI, type, etc.
- main.tf – to define actual EC2 resources
- output.tf – to print public/private Ips

The screenshot shows the VS Code interface with the following details:

- Project Structure:** The left sidebar shows a project named "MEDICURE-PROJECT" containing several sub-folders and files:
 - star-agile-health-care
 - Healthcare-Infra
 - terraform
 - variables.tf
 - main.tf
 - output.tf
 - terraform.state
 - terraform.state.backup
 - src
 - .gitignore
 - ansible-playbook.yml
 - Dockerfile
 - mmw
 - mmw.cmd
 - pom.xml
- Code Editor:** The main editor area displays the "main.tf" file content, which defines five AWS instances named "Medicure" with specific configurations like instance type, volume size, and tags.
- Terminal:** The bottom right terminal window shows the command-line output of running Terraform commands to provision the instances, including AWS provider configuration and instance creation logs.

```

variable "aws_region" {
  default = "us-east-1"
}

variable "instance_count" {
  default = 5
}

variable "instance_type" {
  default = "t3.medium"
}

variable "key_name" {
  default = "keypair1"
}

variable "security_group_id" {
  default = "sg-0eff05ddc0ad8664f"
}

```

```

output "instance_name_to_public_ip" {
  description = "Map of instance names to their public IP addresses"
  value = {
    for instance in aws_instance.medicure :
      instance.tags["Name"] => instance.public_ip
  }
}

output "instance_ids" {
  description = "IDs of all Medicure instances"
  value = [for instance in aws_instance.medicure : instance.id]
}

```

- Open a new CMD terminal and initialize the terraform using “terraform init” command

```

star-agile-health-care > Healthcare-Infra > output.tf > output "instance_ids"
1   output "instance_name_to_public_ip" {
2     description = "Map of instance names to their public IP addresses"
3     value = [
4       for instance in aws_instance.medicure :
5         instance.tags["Name"] -> instance.public_ip
6     ]
7   }
8
9   output "instance_ids" [
10    description = "IDs of all Medicure instances"
11    value = [for instance in aws_instance.medicure : instance.id]
12  ]

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\saiku\OneDrive\Desktop\Medicure-project\star-agile-health-care\Healthcare-Infra> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.98.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

PS C:\Users\saiku\OneDrive\Desktop\Medicure-project\star-agile-health-care\Healthcare-Infra>

Ln 12, Col 2 Spaces: 2 UTF-8 CRLF {} Terraform

31°C Haze

- Use “terraform plan” command to check and verify it the resources are matching as required then use command “terraform apply” to create the resources

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\saiku\OneDrive\Desktop\Medicure-project\star-agile-health-care\Healthcare-Infra> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

```

# aws_instance.medicure[0] will be created
+ resource "aws_instance" "medicure" {
    + ami                                = "ami-084568db4383264d4"
    + arn                                = (known after apply)
    + associate_public_ip_address        = (known after apply)
    + availability_zone                  = (known after apply)
    + cpu_core_count                     = (known after apply)
    + cpu_threads_per_core              = (known after apply)
    + disable_api_stop                  = (known after apply)
    + disable_api_termination           = (known after apply)
    + ebs_optimized                      = (known after apply)
    + enable_primary_ipv6                = (known after apply)
    + get_password_data                 = false
    + host_id                            = (known after apply)
    + host_resource_group_arn            = (known after apply)
    + iam_instance_profile               = (known after apply)
    + id                                 = (known after apply)
    + instance_initiated_shutdown_behavior = (known after apply)
    + instance_lifecycle                = (known after apply)
    + instance_state                     = (known after apply)
    + instance_type                      = "t3.medium"
    + ipv6_address_count                = (known after apply)
    + ipv6_addresses                     = (known after apply)
    + key_name                           = "keypair1"
    + monitoring                         = (known after apply)
    + outpost_arn                        = (known after apply)
    + password_data                      = (known after apply)
    + placement_group                   = (known after apply)
    + placement_partition_number         = (known after apply)
    + primary_network_interface_id      = (known after apply)
    + private_dns                         = (known after apply)
    + private_ip                          = (known after apply)
    + public_dns                          = (known after apply)
    + public_ip                           = (known after apply)
}
```

Ln 12, Col 2 Spaces: 2 UTF-8 CRLF {} Terraform

31°C Haze

File Edit Selection View Go Run Terminal Help ← → ⌘ Medicure-project

EXPLORER

MEDICURE-PROJECT

- star-agile-health-care
 - .mvn
- Healthcare-Infra
 - .terraform
 - main.tf
 - output.tf
 - terraform.tfstate
 - terraform.state.backup
 - variables.tf
- src
- .gitignore
- ansible-playbook.yml
- Dockerfile
- mvnw
- mvnw.cmd
- pom.xml

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

```

aws_instance.medicure[4]: Creating...
aws_instance.medicure[3]: Creating...
aws_instance.medicure[6]: Creating...
aws_instance.medicure[2]: Creating...
aws_instance.medicure[1]: Creating...
aws_instance.medicure[4]: Still creating... [00m08s elapsed]
aws_instance.medicure[3]: Still creating... [00m08s elapsed]
aws_instance.medicure[0]: Still creating... [00m08s elapsed]
aws_instance.medicure[2]: Still creating... [00m08s elapsed]
aws_instance.medicure[1]: Still creating... [00m08s elapsed]
aws_instance.medicure[0]: Creation complete after 19s [id: i-0795d510e8dcea3ca]
aws_instance.medicure[4]: Creation complete after 19s [id: i-039f21fe4c73e3cd]
aws_instance.medicure[1]: Creation complete after 19s [id: i-045ac2b0de5ee4e62]
aws_instance.medicure[2]: Creation complete after 19s [id: i-03c75d4964d3b467d]
aws_instance.medicure[3]: Creation complete after 19s [id: i-05224f923fb9c19b3]

```

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.

Outputs:

```

instance_ids = [
  "i-0795d510e8dcea3ca",
  "i-045ac2b0de5ee4e62",
  "i-03c75d4964d3b467d",
  "i-05224f923fb9c19b3",
  "i-039f21fe4c73e3cd"
]

instance_name_to_public_ip = {
  "master-host" = "44.203.95.187"
  "monitoring" = "44.202.1.240"
  "prod-1" = "18.234.227.196"
  "prod-2" = "3.95.132.249"
  "test-1" = "3.82.175.106"
}

```

PS C:\Users\salkum\OneDrive\Desktop\Medicure-project\star-agile-health-care\Healthcare-Infra>

Ln 12 Col 2 Spaces: 2 UTF-8 CRLF {} Terraform

31°C Haze

star-agile-health-care.mvn\wrap | Instances | EC2 us-east-1

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#instances:

EC2 > Instances

Instances (1/5) Info

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 |
|-------------|---------------------|----------------|---------------|-------------------|---------------|-------------------|-------------|
| test-1 | i-05224f923fb9c19b3 | Running | t3.medium | 3/3 checks passed | View alarms + | us-east-1b | ec2-3-82-1 |
| master-host | i-0795d510e8dcea3ca | Running | t3.medium | 3/3 checks passed | View alarms + | us-east-1b | ec2-44-20 |
| prod-1 | i-045ac2b0de5ee4e62 | Running | t3.medium | 3/3 checks passed | View alarms + | us-east-1b | ec2-18-23 |
| prod-2 | i-03c75d4964d3b467d | Running | t3.medium | 3/3 checks passed | View alarms + | us-east-1b | ec2-3-95-1 |
| monitoring | i-039f21fe4c73e3cd | Running | t3.medium | 3/3 checks passed | View alarms + | us-east-1b | ec2-44-20 |

i-0795d510e8dcea3ca (master-host)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary

Instance ID: i-0795d510e8dcea3ca

Public IPv4 address: 44.203.95.187

Private IPv4 addresses: 172.31.95.11

CloudShell Feedback

31°C Haze

- Connect the all the instances through MobaXterm using the key pair attached in terraform script
Install all the required packages in the master machine (Jenkins, Java, maven,git, ansible, docker, kubernetes) which is the Jenkins host and ansible controller and the Kubernetes master node .

For Java,Git,Maven and Jenkins:

```
#!/bin/bash
# USE UBUNTU20.04 - INSTANCE: 2GB RAM + 2VCPU MIN - WILL ONLY WORK
sudo apt update -y
sudo apt install openjdk-17-jdk -y
sudo apt update -y
sudo apt install maven -y
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
    /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
    https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
    /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt update -y
sudo apt install jenkins -y
service jenkins start
cat /var/lib/jenkins/secrets/initialAdminPassword
#chmod 777 jenkins.sh
./jenkins.sh
```

For Docker:

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
# Add the repository to Apt sources:
echo \
"deb      [arch=$(dpkg --print-architecture)      signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-
plugin -y
```

For ansible:

```
sudo su
apt update -y
apt-get install -y software-properties-common
apt-add-repository ppa:ansible/ansible
apt-get update
apt-get install -y ansible
```

For Kubernetes:

```
# As root user
```

```
sudo su
```

Create k8s-master.sh and paste the following :

```
## Install Docker
```

```
sudo wget
```

```
https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installDocker.sh -P /tmp
sudo chmod 755 /tmp/installDocker.sh
sudo bash /tmp/installDocker.sh
sudo systemctl restart docker.service
```

```
## Install CRI-Docker
```

```
sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installCRIDockerd.sh -P
/tmp
sudo chmod 755 /tmp/installCRIDockerd.sh
sudo bash /tmp/installCRIDockerd.sh
sudo systemctl restart cri-docker.service
```

```
## Install kubernetes
```

```
sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installK8S.sh -P /tmp
sudo chmod 755 /tmp/installK8S.sh
sudo bash /tmp/installK8S.sh
```

```
## Initialize kubernetes Master Node
```

```
sudo kubeadm init --cri-socket unix:///var/run/cri-dockerd.sock --ignore-preflight-errors=all
```

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

```
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
## install networking driver -- Weave/flannel/canal/calico etc...
```

```
## below installs calico networking driver
```

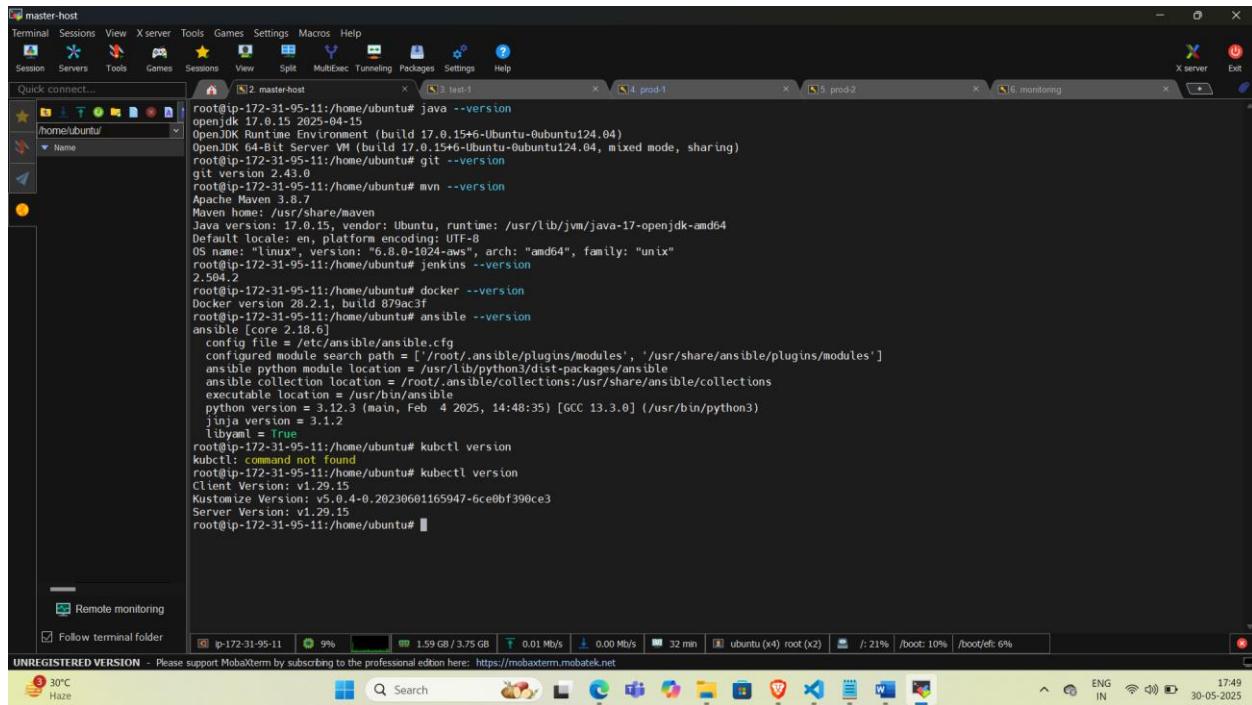
```
kubectl apply -f
```

<https://raw.githubusercontent.com/projectcalico/calico/v3.24.1/manifests/calico.yaml>

Then give permission and execute the k8s-master.sh:

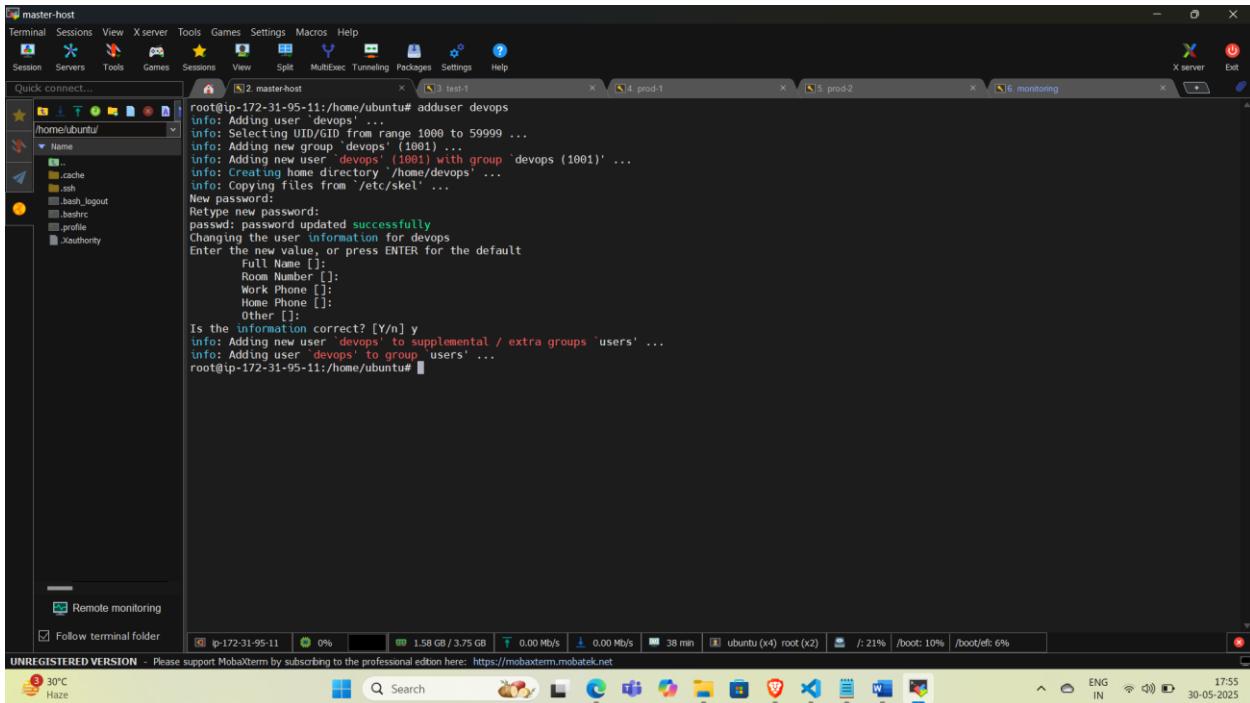
```
chmod +x k8s-master.sh
```

```
./k8s-master.sh
```



```
root@ip-172-31-95-11:/home/ubuntu# java --version
OpenJDK Runtime Environment (build 17.0.15+6-Ubuntu-Ubuntu124.04)
OpenJDK 64-Bit Server VM (build 17.0.15+6-Ubuntu-Ubuntu124.04, mixed mode, sharing)
root@ip-172-31-95-11:/home/ubuntu# git --version
git version 2.43.0
root@ip-172-31-95-11:/home/ubuntu# mvn --version
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 17.0.15, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1024-aws", arch: "amd64", family: "unix"
root@ip-172-31-95-11:/home/ubuntu# jenkins --version
2.580.2
root@ip-172-31-95-11:/home/ubuntu# docker --version
Docker version 28.2.1, build 879ac3f
root@ip-172-31-95-11:/home/ubuntu# ansible --version
ansible [core 2.18.6]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['~/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /root/.ansible/bin
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Feb 4 2025, 14:48:35) [GCC 13.3.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
root@ip-172-31-95-11:/home/ubuntu# kubectl version
Client Version: v1.29.15
Kustomize Version: v5.0.4-4.0.20230601165947-6ce0bf390ce3
Server Version: v1.29.15
root@ip-172-31-95-11:/home/ubuntu#
```

- Now do Ansible configuration from master node to connect all other nodes as user “devops” with password .Do it in master and all other nodes

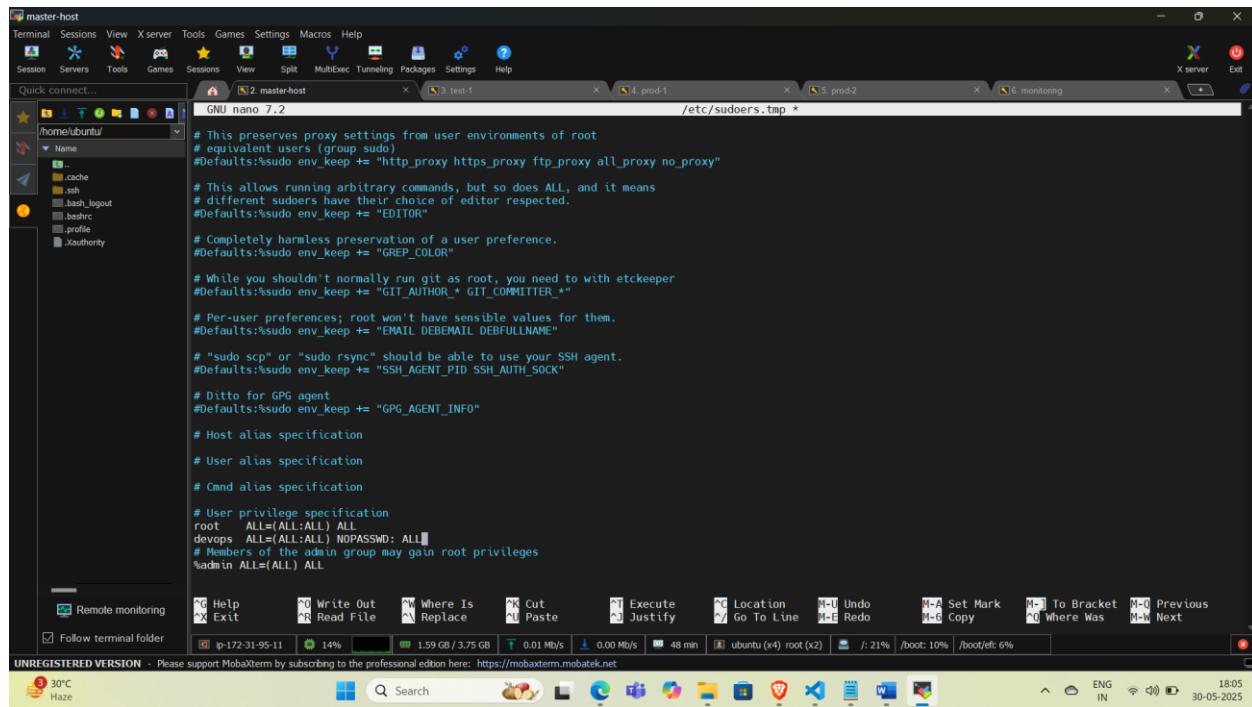


```
root@ip-172-31-95-11:/home/ubuntu# adduser devops
info: Adding user 'devops' ...
info: Selecting UID/GID from range 1000 to 50999 ...
info: Adding new group 'devops' (1001) ...
info: Adding new user 'devops' (1001) with group 'devops' (1001) ...
info: Creating home directory '/home/devops' ...
info: Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing user information for devops
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
info: Adding new user 'devops' to supplemental / extra groups  users' ...
info: Adding user 'devops' to group 'users' ...
root@ip-172-31-95-11:/home/ubuntu#
```

- Now add the user to the sudoers file with the permissions by “visudo”.Perform the

same in all the other nodes

devops ALL=(ALL:ALL) NOPASSWD: ALL



```
# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
Defaults:env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
Defaults:env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
Defaults:env_keep += "GREP_COLOR"

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

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

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

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

# Host alias specification

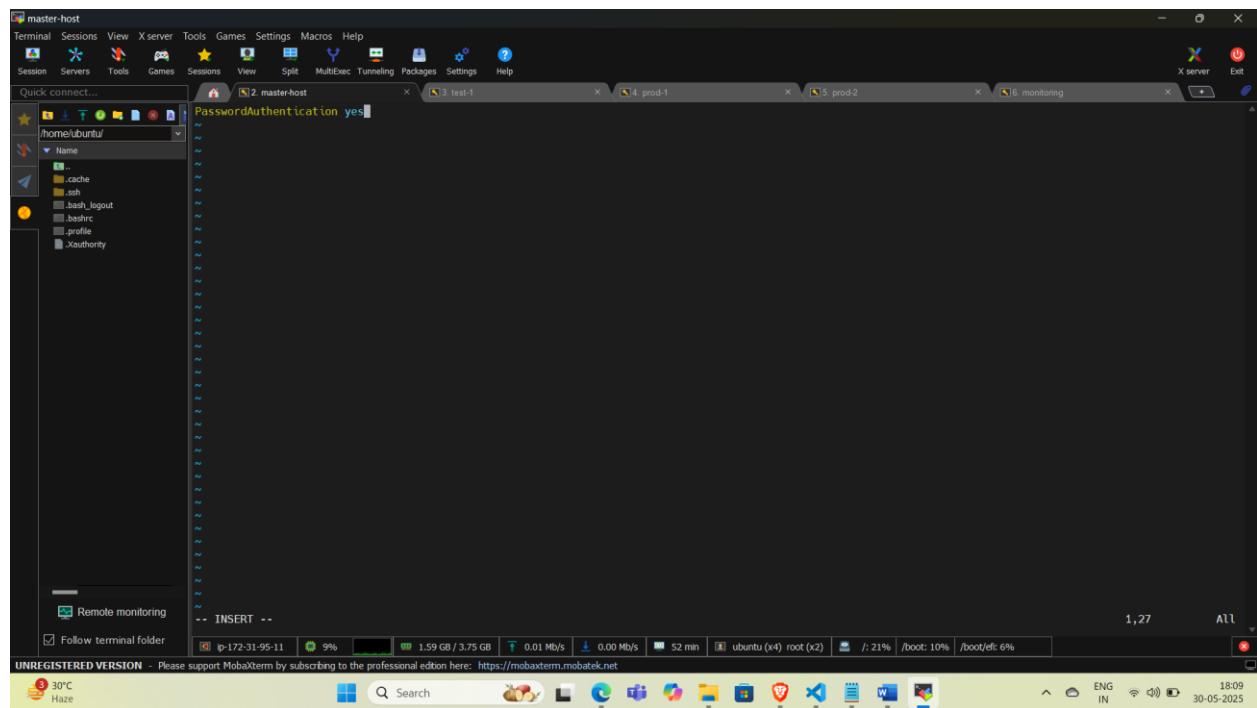
# User alias specification

# Cmnd alias specification

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


```

- Now configure sshd configuration in both master and other nodes Get into the “vi /etc/ssh/sshd_config.d/60-cloudimg-settings.conf “and change passwordauthentication to yes



```
passwordAuthentication yes
```

- Also change the configuration file in using command “vi /etc/ssh/sshd_config” change Permitlogin to yes Pubkeyauthentication to yes And passwordauthentication to yes

- Perform the same activity in nodes as well

```
# RekeyLimit default none
# Logging
#syslogFacility AUTH
#LogLevel INFO

# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
#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
#WPA
```

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

30°C Haze

ip-172-31-95-11 5% 1.58 GB / 3.75 GB 0.01 Mb/s 0.00 Mb/s 57 min ubuntu (x2) 21% /boot: 10% /boot/efi: 6%

ENG IN 18:14 30-05-2025

- Now run command “`sudo systemctl restart ssh`” in all the nodes after all the above configuration

```
root@ip-172-31-95-11:/home/ubuntu# adduser devops
info: Adding user 'devops' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group 'devops' (1001) ...
info: Adding new user 'devops' (1001) with group 'devops' (1001) ...
info: Creating home directory '/home/devops' ...
info: Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for devops
Enter the new value, or press ENTER for the default
Full Name []:
Room Number []:
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n] y
info: Adding new user 'devops' to supplemental / extra groups 'users' ...
info: Adding user 'devops' to group 'users' ...
root@ip-172-31-95-11:/home/ubuntu# vi /etc/ssh/sshd_config.d/60-cloudimg-settings.conf
root@ip-172-31-95-11:/home/ubuntu# vi /etc/ssh/sshd_config
root@ip-172-31-95-11:/home/ubuntu# sudo systemctl restart ssh
root@ip-172-31-95-11:/home/ubuntu#
```

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

30°C Haze

ip-172-31-95-11 5% 1.58 GB / 3.75 GB 0.01 Mb/s 0.00 Mb/s 61 min ubuntu (x2) 21% /boot: 10% /boot/efi: 6%

ENG IN 18:17 30-05-2025

- In master-host node, add the other nodes into the hosts file by “`vi /etc/ansible/hosts`” file with the private ip address:

[test]

test-1 ansible_host=<private-ip of test-1>

[production]

prod-1 ansible_host=<private-ip of prod-1>

prod-2 ansible_host=<private-ip of prod-2>

[monitoring-group]

monitoring ansible_host=<private-ip of monitoring>

The screenshot shows the MobaXterm interface with several windows open:

- A terminal window titled "master-host" containing Ansible host definitions.
- A file viewer window titled "monitoring" showing a configuration file for monitoring hosts.
- Other terminal windows labeled "test-1", "prod-1", and "prod-2".
- A system tray at the bottom showing battery status, network, and date/time.

```
# If you have multiple hosts following a pattern, you can specify
# them like this:
## www[001:006].example.com
# You can also use ranges for multiple hosts:
## db-[99:101]-node.example.com
# Ex 3: A collection of database servers in the 'dbservers' group:
## [dbservers]
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Ex4: Multiple hosts arranged into groups such as 'Debian' and 'openSUSE':
## [Debian]
## alpha.example.org
## beta.example.org

## [openSUSE]
## green.example.com
## blue.example.com
[test]
test-1 ansible_host=172.31.91.6

[production]
prod-1 ansible_host=172.31.83.48
prod-2 ansible_host=172.31.85.26

[monitoring-node]
monitoring ansible_host=172.31.87.101
:wd
```

- Now get into the user devops using command “su – devops” and generate a ssh key using “ssh-keygen” command in master-host.

master-host

```
root@ip-172-31-95-11:/home/ubuntu# su - devops
devops@ip-172-31-95-11:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/devops/.ssh/id_ed25519):
Created directory '/home/devops/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/devops/.ssh/id_ed25519
Your public key has been saved in /home/devops/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:V8dRqTy4/hu9+TzQ0aJm8I8862NAU6PN1zrRkqhvw devops@ip-172-31-95-11
The key's randomart image is:
+---[ED25519 256]---

o o o .
. * B o o
= + B oSo.
. * o + =o .
o o o .oo
.oB B ..+ .o
|o+E +oo +
+---[SHA256]-----
devops@ip-172-31-95-11:~$
```

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

Upcoming Earnings Search

ENG IN 18:29 30-05-2025

- And now we have to copy the ssh into other nodes using ssh-copy-id devops@<private-ip>
Perform this to every node from master host

master-host

```
root@ip-172-31-95-11:/home/ubuntu# su - devops
devops@ip-172-31-95-11:~$ cd .ssh/
devops@ip-172-31-95-11:~/ssh$ ls
id_ed25519 id_ed25519.pub known_hosts
devops@ip-172-31-95-11:~/ssh$ ssh-copy-id devops@172.31.86.165
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
devops@172.31.86.165's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'devops@172.31.86.165'"
and check to make sure that only the key(s) you wanted were added.

devops@ip-172-31-95-11:~/ssh$ ssh-copy-id devops@172.31.83.48
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_ed25519.pub"
The authenticity of host '172.31.83.48 (172.31.83.48)' can't be established.
ED25519 key fingerprint is SHA256:H4mkzgSHZtWrs0zjsrIOIDUNZLXGjLsxt2ou8jGmiQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
devops@172.31.83.48's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'devops@172.31.83.48'"
and check to make sure that only the key(s) you wanted were added.

devops@ip-172-31-95-11:~/ssh$ ssh-copy-id devops@172.31.85.26
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_ed25519.pub"
The authenticity of host '172.31.85.26 (172.31.85.26)' can't be established.
ED25519 key fingerprint is SHA256:ONzANB0+b1u4qUN6d4ThjhNW3BndSRL1KBIws3MELs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
devops@172.31.85.26's password:
```

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

29°C Sunny Search

ENG IN 19:02 30-05-2025

```

master-host
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
[2 master-host /home/ubuntu/ ] [3 test-1 ] [4 prod-1 ] [5 prod-2 ] [6 monitoring ]
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'devops@172.31.83.48'" and check to make sure that only the key(s) you wanted were added.
devops@ip-172-31-95-11:~/.ssh$ ssh-copy-id devops@172.31.85.26
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_ed25519.pub"
The authenticity of host '172.31.85.26' (172.31.85.26) can't be established.
ED25519 key fingerprint is SHA256:0NzAN0Gb1u4qUN6d4tjhNW3b0nSRL1K8lws3MEls.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
devops@172.31.85.26's password:

Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'devops@172.31.85.26'" and check to make sure that only the key(s) you wanted were added.
devops@ip-172-31-95-11:~/.ssh$ ssh-copy-id devops@172.31.87.101
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_ed25519.pub"
The authenticity of host '172.31.87.101' (172.31.87.101) can't be established.
ED25519 key fingerprint is SHA256:uDhQVR84k7+StXrWhfkcAvakcdQn5io4YGB2zQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
devops@172.31.87.101's password:

Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'devops@172.31.87.101'" and check to make sure that only the key(s) you wanted were added.
devops@ip-172-31-95-11:~/.ssh$ 

```

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

29°C Sunny

- Now generate a Kubernetes join key in the master node using command “ kubeadm token create –print-join command” and copy the token generated for further use.

```

master-host
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
[2 master-host /home/ubuntu/ ] [3 test-1 ] [4 prod-1 ] [5 prod-2 ] [6 monitoring ]
devops@ip-172-31-95-11:~/.ssh$ cd ..
devops@ip-172-31-95-11:~$ sudo su
root@ip-172-31-95-11:/home/devops# kubeadm token create --print-join-command
kubeadm join 172.31.95.11:6443 --token pzm9xr.662zwuctpvh8dw --discovery-token-ca-cert-hash sha256:9e59dc82bc81b137458b5d6d21e1bd1f1d19f874245dde94f90f4538a3
dfa2
root@ip-172-31-95-11:/home/devops# 

```

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

29°C Sunny

- whatever token comes up ...copy the token and paste it in notepad and then in the command you will see 6443 written ...after that paste --cri-socket unix:///var/run/cri-dockerd.sock

A screenshot of a Windows terminal window titled "kubeadm join 172.31.95.11:6443 --cri-socket unix:///var/run/cri-dockerd.sock --token pzm9xr.602zwuctvh8diwe --discovery-token-ca-cert-hash sha256:9e59dc82bc81b137458b5d6fd21e1bd1f1d19f874245dde94f90f4538a3dfa2". The terminal shows the command being run and its output. The window has a standard Windows title bar and taskbar at the bottom.

- Now write a ansible playbook to connect the nodes with the kubernetes cluster master and also set the hostname of each instance accordingly to the hosts file which has been written previously.
- This book will install the kubernetes slave configuration , network bridging and Kubernetes join token with cri socket:
[vi k8s-nodes-setup.yml](#):

```
- name: Set hostname and verify
hosts: all-nodes
become: true
vars:
  desired_node_name: "{{ inventory_hostname }}"
tasks:
  - name: Set custom hostname
    ansible.builtin.hostname:
      name: "{{ desired_node_name }}"
  - name: Print the set hostname for verification
    ansible.builtin.command: hostname
    register: current_hostname
  - name: Show the set hostname
    ansible.builtin.debug:
      msg: "Hostname set to: {{ current_hostname.stdout }}"
```

```
- name: Confirm and proceed (skipping prompt)
  ansible.builtin.debug:
    msg: "Proceeding with install on {{ inventory_hostname }}"

- name: Install Kubernetes and configure network
  hosts: all-nodes
  become: true

  tasks:
    - name: Download k8s-nodes.sh script
      ansible.builtin.get_url:
        url: https://raw.githubusercontent.com/akshu20791/Deployment-script/main/k8s-nodes.sh
        dest: /tmp/k8s-nodes.sh
        mode: '0755'

    - name: Ensure k8s-nodes.sh is executable
      ansible.builtin.file:
        path: /tmp/k8s-nodes.sh
        mode: '0755'

    - name: Run k8s-nodes.sh script
      ansible.builtin.shell: /tmp/k8s-nodes.sh

    - name: Enable networking modules
      ansible.builtin.shell: |
        modprobe br_netfilter
        echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables
        echo 1 > /proc/sys/net/ipv4/ip_forward

    - name: Join Kubernetes cluster
      ansible.builtin.shell: |
        kubeadm join 172.31.95.11:6443 --cri-socket unix:///var/run/cri-dockerd.sock \
          --token pzm9xr.602zwuctpvh8diwe \
          --discovery-token-ca-cert-hash
sha256:9e59dc82bc81b137458b5d6fd21e1bd1f1d19f874245dde94f90f4538a3dfa2
```

In the same directory as your playbook (/home/devops):

nano ansible.cfg

Paste this:

```
[defaults]
inventory = /etc/ansible/hosts
host_key_checking = False
retry_files_enabled = False
```

Save and exit (Ctrl + O, Enter, Ctrl + X)

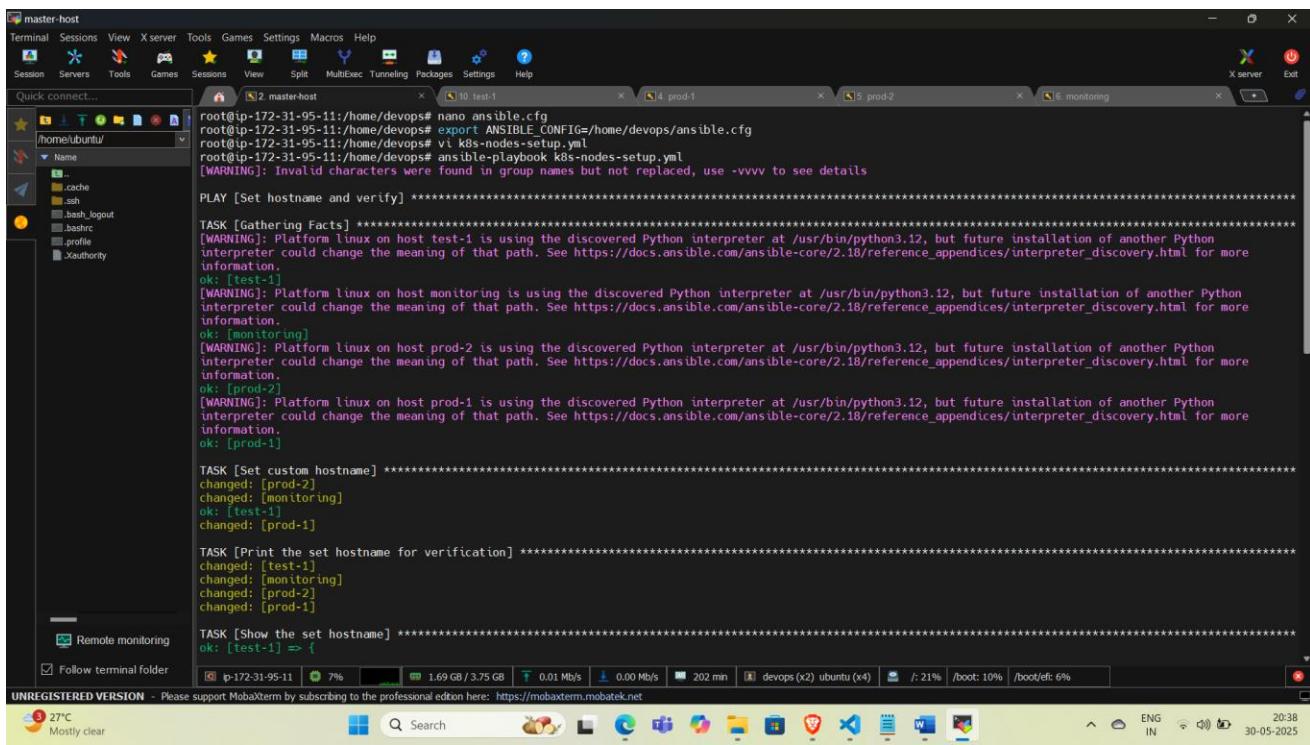
Export config (optional but recommended):

In the same terminal session, run:

```
export ANSIBLE_CONFIG=/home/devops/ansible.cfg
```

Then run the playbook:

```
ansible-playbook k8s-nodes-setup.yml
```



```
root@ip-172-31-95-11:/home/devops# nano ansible.cfg
root@ip-172-31-95-11:/home/devops# export ANSIBLE_CONFIG=/home/devops/ansible.cfg
root@ip-172-31-95-11:/home/devops# vi k8s-nodes-setup.yml
root@ip-172-31-95-11:/home/devops# ansible-playbook k8s-nodes-setup.yml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [Set hostname and verify] *****
[WARNING]: Platform linux on host test-1 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [test-1]
[WARNING]: Platform linux on host monitoring is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [monitoring]
[WARNING]: Platform linux on host prod-2 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [prod-2]
[WARNING]: Platform linux on host prod-1 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [prod-1]

TASK [Set custom hostname] *****
changed: [prod-2]
changed: [monitoring]
ok: [test-1]
changed: [prod-1]

TASK [Print the set hostname for verification] *****
changed: [test-1]
changed: [monitoring]
changed: [prod-2]
changed: [prod-1]

TASK [Show the set hostname] *****
ok: [test-1] => {
```

- After successful run of ansible playbook check if the nodes got connected successfully by running the command “kubectl get nodes”

```

root@ip-172-31-95-11:/home/devops# kubeadm token create --print-join-command
kubeadm join 172.31.95.11:6443 --token c12av1.qhv44bjp52w6ty --discovery-token-ca-cert-hash sha256:9e59dc82bc81b137458b5d6fd21e1bd1f1d19f874245dde94f90f4538a3
dfa2@ip-172-31-95-11:/home/devops# kubectl get nodes
NAME      STATUS   ROLES    AGE     VERSION
ip-172-31-95-11   Ready    control-plane   3h12m   v1.29.15
test-1     Ready    <none>   21m    v1.29.15
root@ip-172-31-95-11:/home/devops# kubectl get nodes
NAME      STATUS   ROLES    AGE     VERSION
ip-172-31-95-11   Ready    control-plane   3h14m   v1.29.15
prod-1     Ready    <none>   68s    v1.29.15
prod-2     Ready    <none>   60s    v1.29.15
test-1     Ready    <none>   23m    v1.29.15
root@ip-172-31-95-11:/home/devops#

```

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

- Create kubernetes namespace in order to separate the nodes using the namespace using command “kubectl create namespace <namespace-name> ” here I have created 2 namespace test, production and verify them by command “kubectl get namespace”

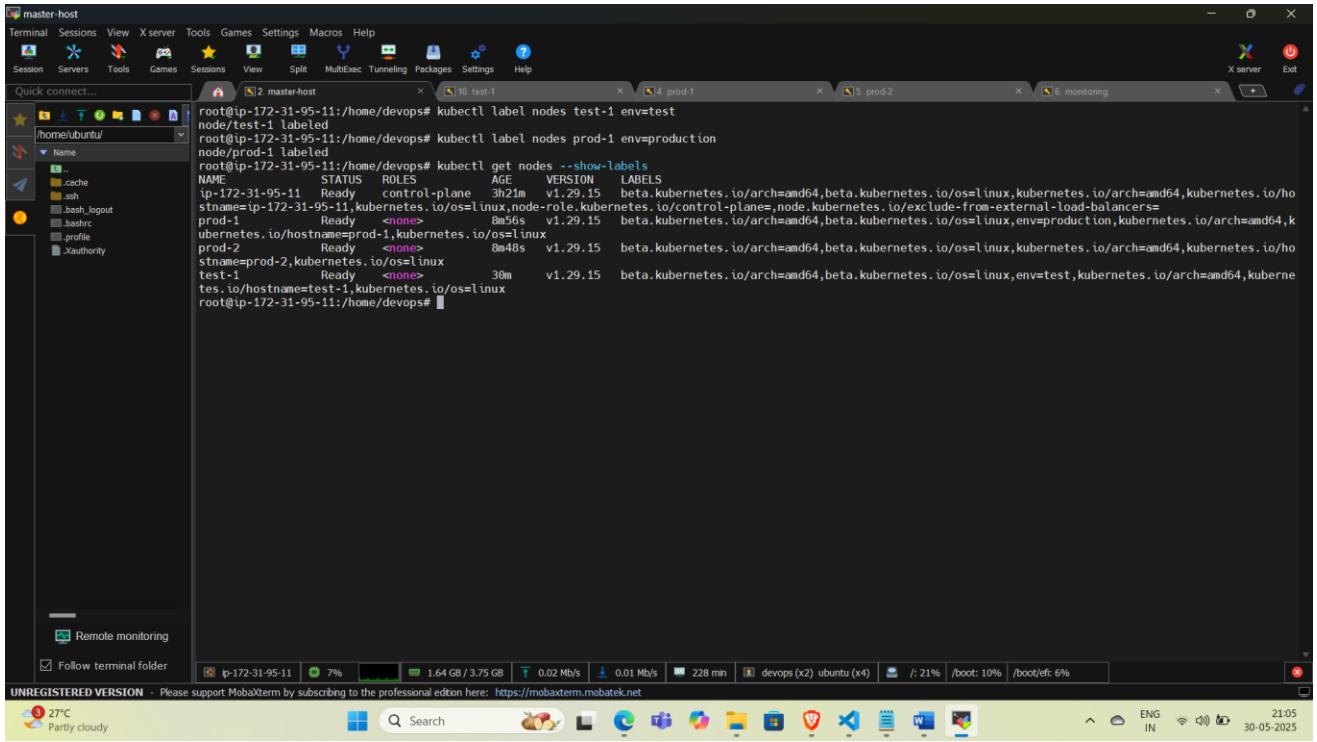
```

root@ip-172-31-95-11:/home/devops# kubeadm token create --print-join-command
kubeadm join 172.31.95.11:6443 --token c12av1.qhv44bjp52w6ty --discovery-token-ca-cert-hash sha256:9e59dc82bc81b137458b5d6fd21e1bd1f1d19f874245dde94f90f4538a3
dfa2@ip-172-31-95-11:/home/devops# kubectl get nodes
NAME      STATUS   ROLES    AGE     VERSION
ip-172-31-95-11   Ready    control-plane   3h12m   v1.29.15
test-1     Ready    <none>   21m    v1.29.15
root@ip-172-31-95-11:/home/devops# kubectl get nodes
NAME      STATUS   ROLES    AGE     VERSION
ip-172-31-95-11   Ready    control-plane   3h14m   v1.29.15
prod-1     Ready    <none>   68s    v1.29.15
prod-2     Ready    <none>   60s    v1.29.15
test-1     Ready    <none>   23m    v1.29.15
root@ip-172-31-95-11:/home/devops# kubectl create namespace test
namespace/test created
root@ip-172-31-95-11:/home/devops# kubectl create namespace production
namespace/production created
root@ip-172-31-95-11:/home/devops# kubectl get namespace
NAME        STATUS   AGE
default     Active   3h17m
kube-node-lease Active  3h17m
kube-public  Active   3h17m
kube-system  Active   3h17m
production   Active   18s
test        Active   26s
root@ip-172-31-95-11:/home/devops#

```

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

- Now set the labels each node using env tag using command “kubectl label nodes <node-name> env=<name> ” here I have set 2 env i.e prod, test. Check the labels using command “kubectl get nodes –show-labels”

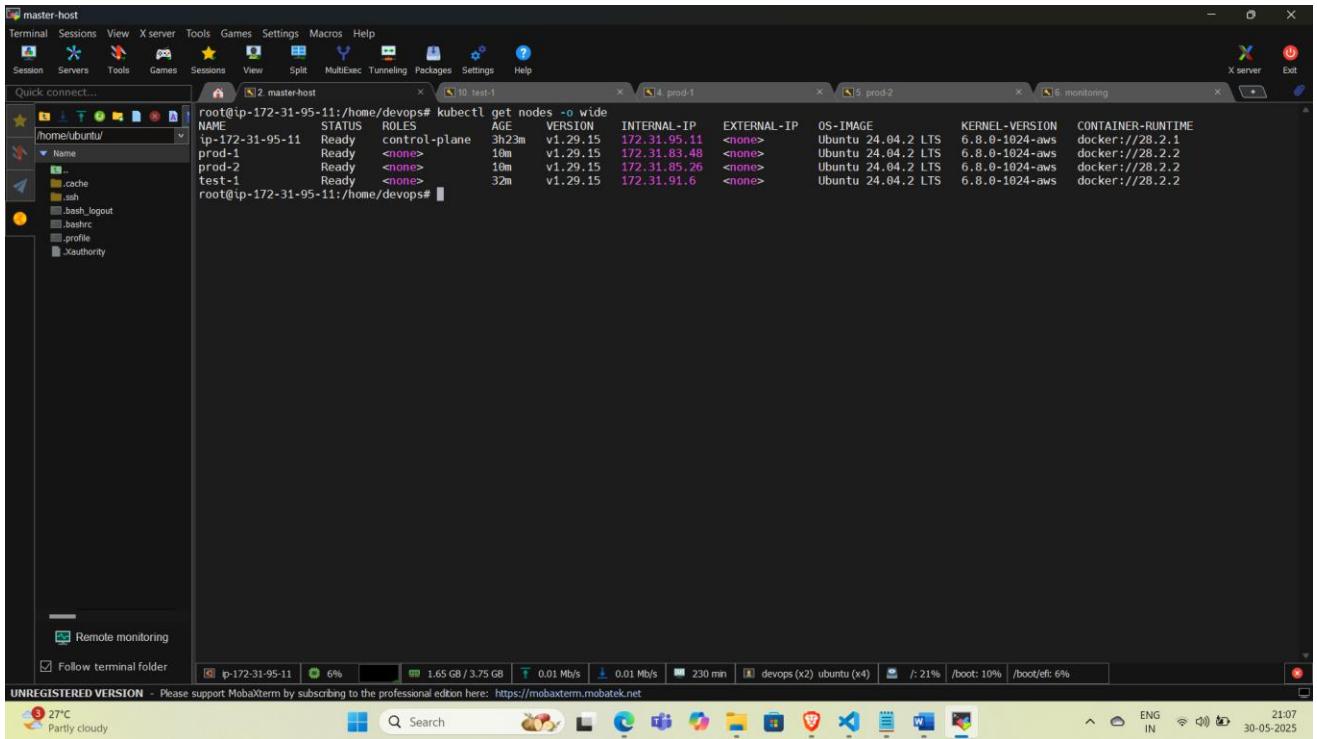


```

root@ip-172-31-95-11:/home/devops# kubectl get nodes test-1 env=test
node/test-1 labeled
root@ip-172-31-95-11:/home/devops# kubectl label nodes prod-1 env=production
node/prod-1 labeled
root@ip-172-31-95-11:/home/devops# kubectl get nodes --show-labels
NAME           STATUS   ROLES      AGE    VERSION   LABELS
ip-172-31-95-11 Ready    control-plane   3h21m   v1.29.15   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-172-31-95-11,kubernetes.io/os=linux-node-role,kubernetes.io/control-plane-node,kubernetes.io/exclude-from-external-load-balancers=
prod-1         Ready    <none>     8m56s   v1.29.15   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,env=production,kubernetes.io/arch=amd64,kubernetes.io/hostname=prod-1,kubernetes.io/os=linux
prod-2         Ready    <none>     8m48s   v1.29.15   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=prod-2,kubernetes.io/os=linux
test-1         Ready    <none>     30m    v1.29.15   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,env=test,kubernetes.io/arch=amd64,kubernetes.io/hostname=test-1,kubernetes.io/os=linux
root@ip-172-31-95-11:/home/devops#

```

- Now verify the nodes status using command “kubectl get nodes -o wide”

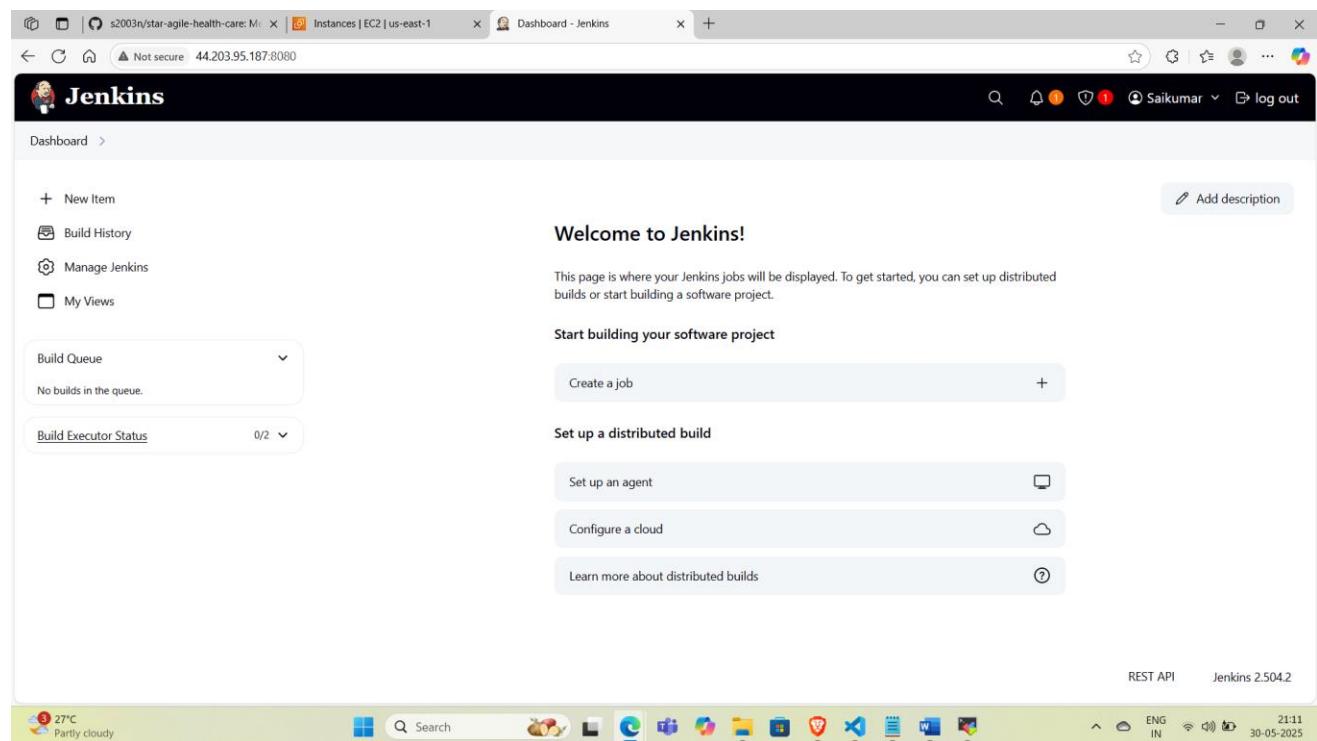
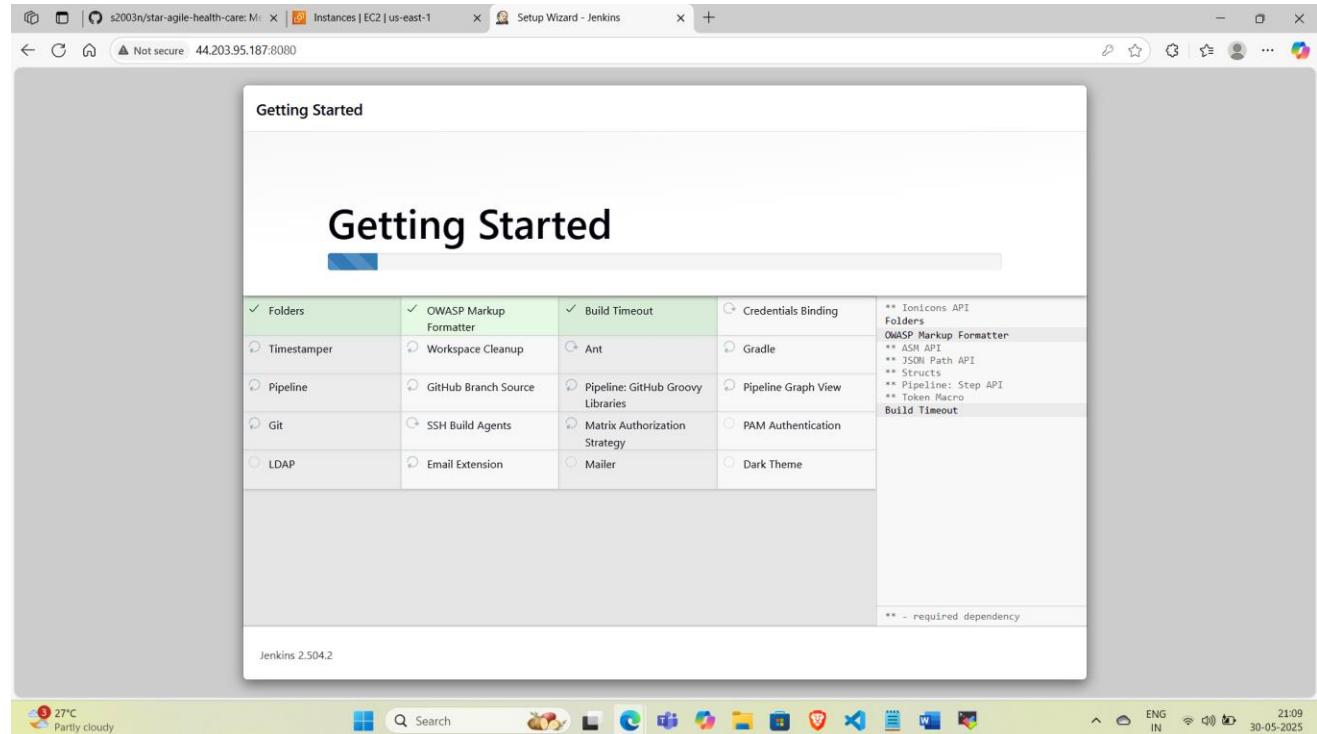


```

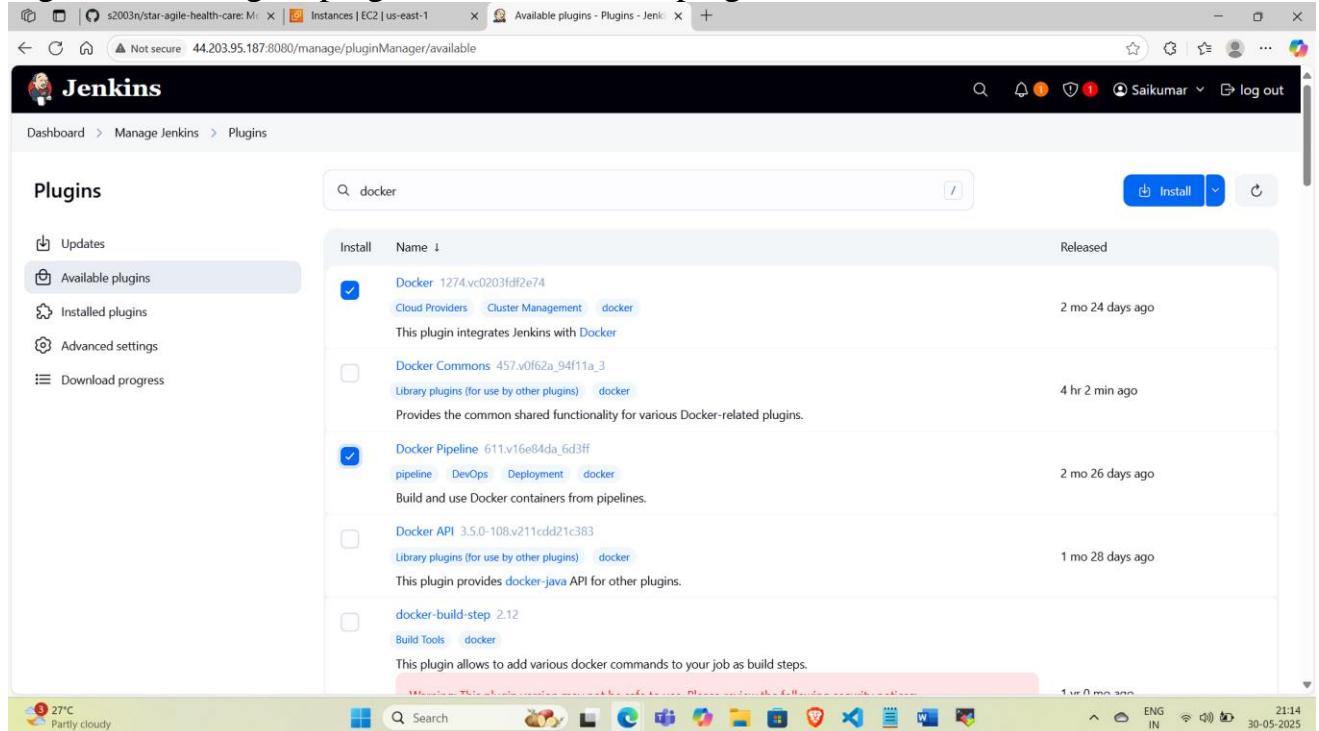
root@ip-172-31-95-11:/home/devops# kubectl get nodes -o wide
NAME           STATUS   ROLES      AGE    VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE          KERNEL-VERSION   CONTAINER-RUNTIME
ip-172-31-95-11 Ready    control-plane   3h23m   v1.29.15   172.31.95.11  <none>        Ubuntu 24.04.2 LTS   6.8.0-1024-aws  docker://28.2.1
prod-1         Ready    <none>       10m    v1.29.15   172.31.83.48  <none>        Ubuntu 24.04.2 LTS   6.8.0-1024-aws  docker://28.2.2
prod-2         Ready    <none>       10m    v1.29.15   172.31.85.26  <none>        Ubuntu 24.04.2 LTS   6.8.0-1024-aws  docker://28.2.2
test-1         Ready    <none>       32m    v1.29.15   172.31.91.6   <none>        Ubuntu 24.04.2 LTS   6.8.0-1024-aws  docker://28.2.2
root@ip-172-31-95-11:/home/devops#

```

- Now access the Jenkins service using the public ip on port 8080 and Login into the Jenkins using initial password which will get at path /var/jenkins_home/secrets/initialAdminPassword



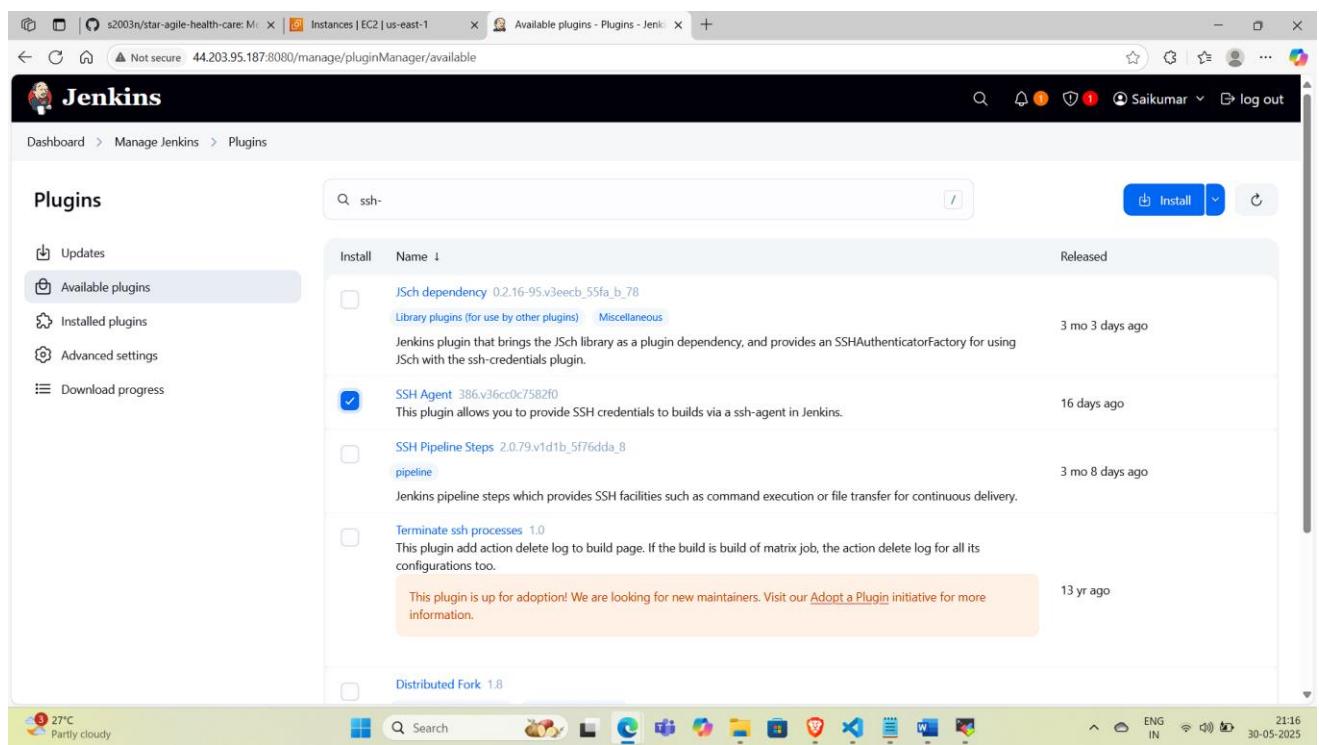
- Navigate to manage Jenkins -> plugins ->available plugins -> docker install docker plugins and ssh agent plugin and Kubernetes plugin



The screenshot shows the Jenkins Plugins management interface. A search bar at the top contains the query "docker". On the left sidebar, the "Available plugins" tab is selected. The main table lists several Docker-related plugins:

| Install | Name | Released |
|-------------------------------------|------------------------------------|------------------|
| <input checked="" type="checkbox"/> | Docker 1274.vc0203fdf2e74 | 2 mo 24 days ago |
| <input type="checkbox"/> | Docker Commons 457.v0f62a_94f11a_3 | 4 hr 2 min ago |
| <input checked="" type="checkbox"/> | Docker Pipeline 611.v16e84da_6d3ff | 2 mo 26 days ago |
| <input type="checkbox"/> | Docker API 3.5.0-108.v211cd21c383 | 1 mo 28 days ago |
| <input type="checkbox"/> | docker-build-step 2.12 | |

An "Install" button is visible in the top right corner of the table header.



The screenshot shows the Jenkins Plugins management interface. A search bar at the top contains the query "ssh-". On the left sidebar, the "Available plugins" tab is selected. The main table lists several SSH-related plugins:

| Install | Name | Released |
|-------------------------------------|--|-----------------|
| <input type="checkbox"/> | JSch dependency 0.2.16-95.v3eebc_b_55fa_b_78 | 3 mo 3 days ago |
| <input checked="" type="checkbox"/> | SSH Agent 386.v36cc0c7582f0 | 16 days ago |
| <input type="checkbox"/> | SSH Pipeline Steps 2.0.79.v1d1b_5f76dda_8 | 3 mo 8 days ago |
| <input type="checkbox"/> | Terminate ssh processes 1.0 | 13 yr ago |
| <input type="checkbox"/> | Distributed Fork 1.8 | |

An "Install" button is visible in the top right corner of the table header.

- Download Kubernetes plugin the HPI file manually from the repository “<https://github.com/akshu20791/cicdakshat>” and load it manually into the jenkins
Navigate to manage Jenkins -> plugins ->advanced setting -> then upload the hpi file and install it or just install kubernetes plugin

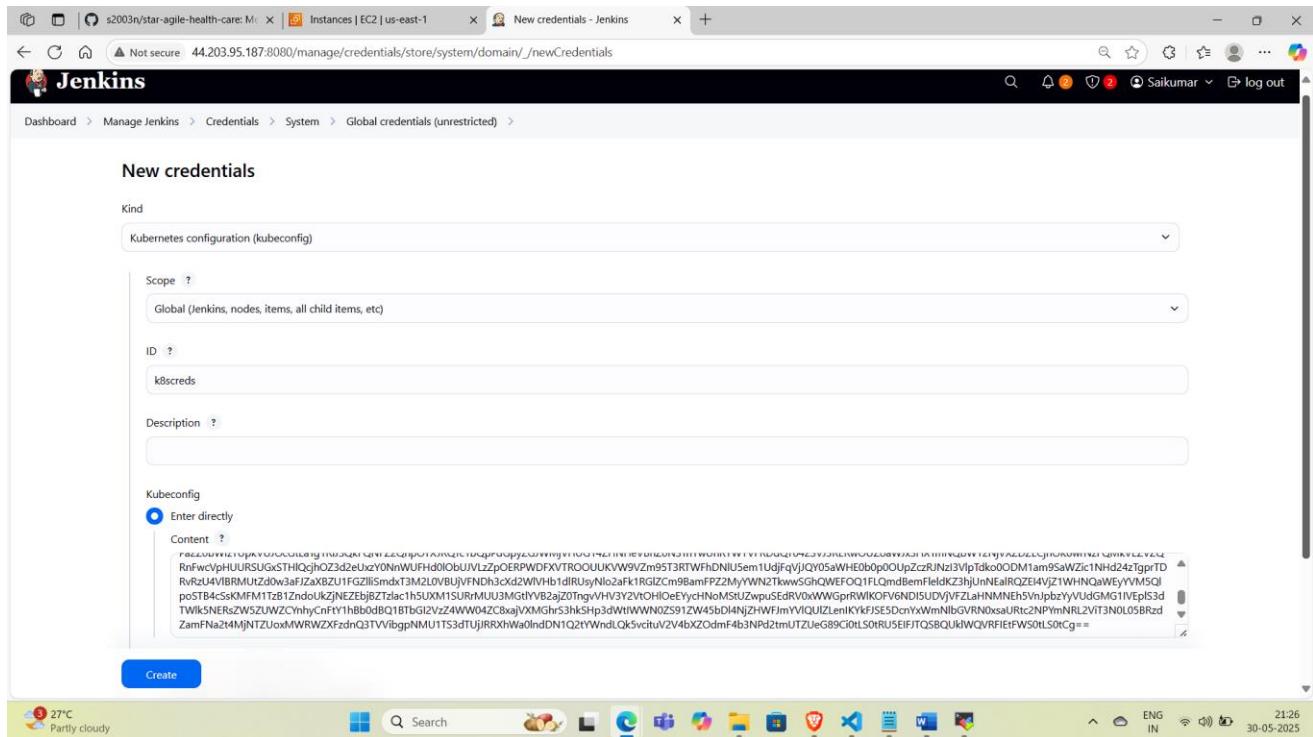
The screenshot shows the Jenkins Plugins management interface. The left sidebar has 'Available plugins' selected. A search bar at the top right contains 'kubernetes'. The main area lists several Kubernetes-related plugins:

| Plugin | Description | Released |
|--|---|------------------|
| Kubernetes 4350.va_0283de0d6d6 | This plugin integrates Jenkins with Kubernetes | 2 days 7 hr ago |
| Kubernetes Client API 6.10.0-251.v556f5f100500 | kubernetes Library plugins (for use by other plugins) | 2 mo 27 days ago |
| Kubernetes Client API plugin for use by other Jenkins plugins. | | |
| Kubernetes Credentials 192.v4d5b_1c429d17 | kubernetes credentials | 2 mo 27 days ago |
| Kubernetes CLI 1.364.vadef8cb8b823 | kubernetes Configure kubectl for Kubernetes | 2 mo 9 days ago |
| Kubernetes Credentials Provider 1.281.v331e3f5a_05a_9 | kubernetes credentials | 3 days 4 hr ago |
| Provides a read only credentials store backed by Kubernetes. | | |

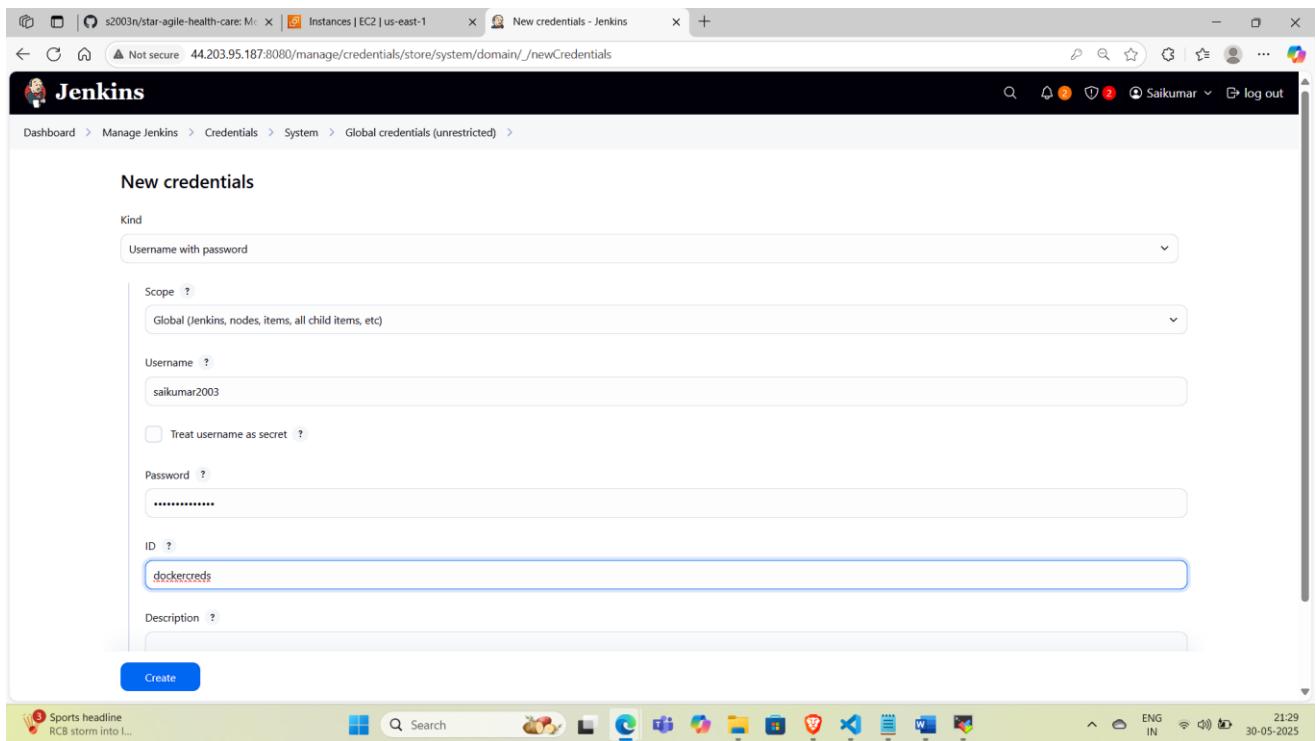
The screenshot shows the Jenkins Advanced settings page for plugin deployment. The left sidebar has 'Advanced settings' selected. A message at the top states: 'The Proxy configuration form has been moved to [Configure System](#) page'. The main area is titled 'Deploy Plugin' and includes fields for 'File' (with a 'Choose File' button) and 'URL'. A 'Deploy' button is located below these fields. At the bottom, there's a 'Update Site' section.

- Run this commands in the master machine and get the configid:
cd ~
ls -a
cd .kube
ls cat config

copy the whole config id which has came up for furhter adding it to jenkins credentails



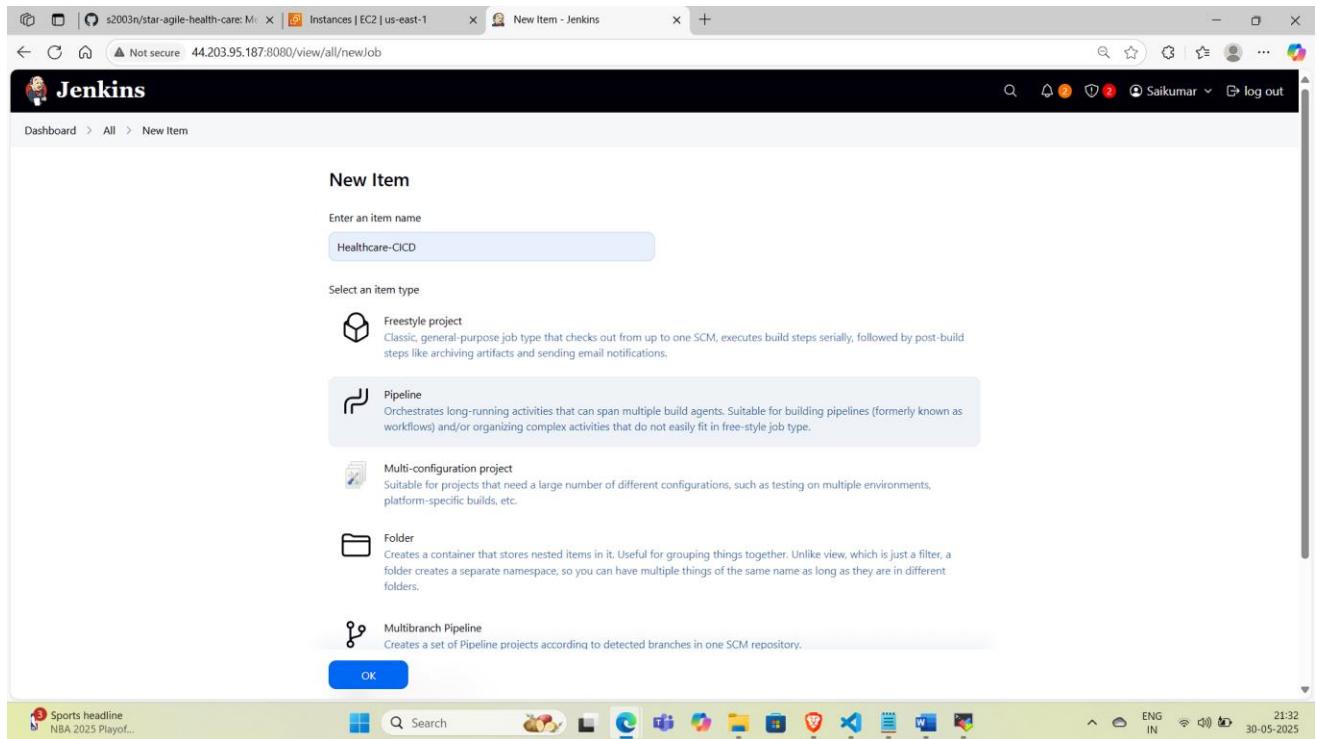
- In the credentials also add the docker credentials to access dockerhub for image push in docker hub (dockercreds)



- Now we have both kubernetes credentials and docker hub credentials:

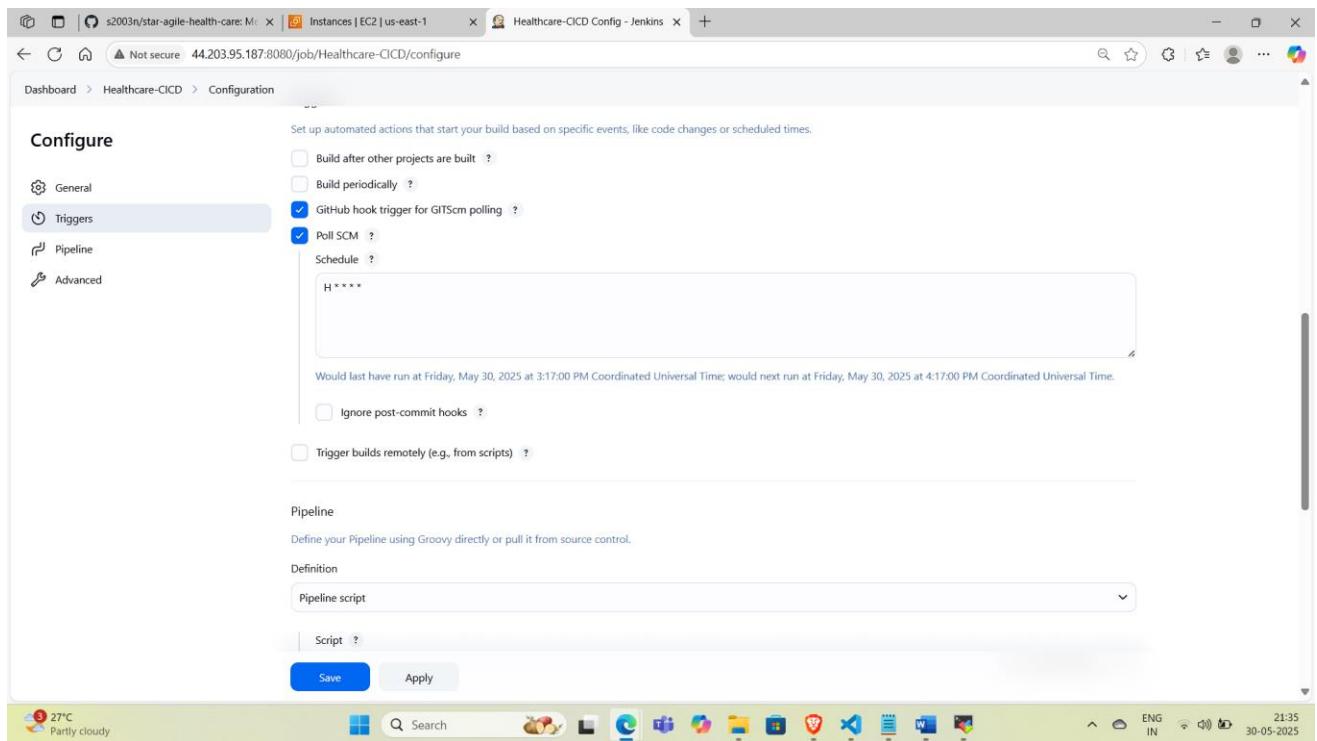
| ID | Name | Kind | Description |
|-------------|--------------------|---------------------------------------|-------------|
| k8screds | k8screds | Kubernetes configuration (kubeconfig) | |
| dockercreds | saikumar2003/***** | Username with password | |

- Now create a new item with name Healthcare-CICD and select the item type as pipeline:

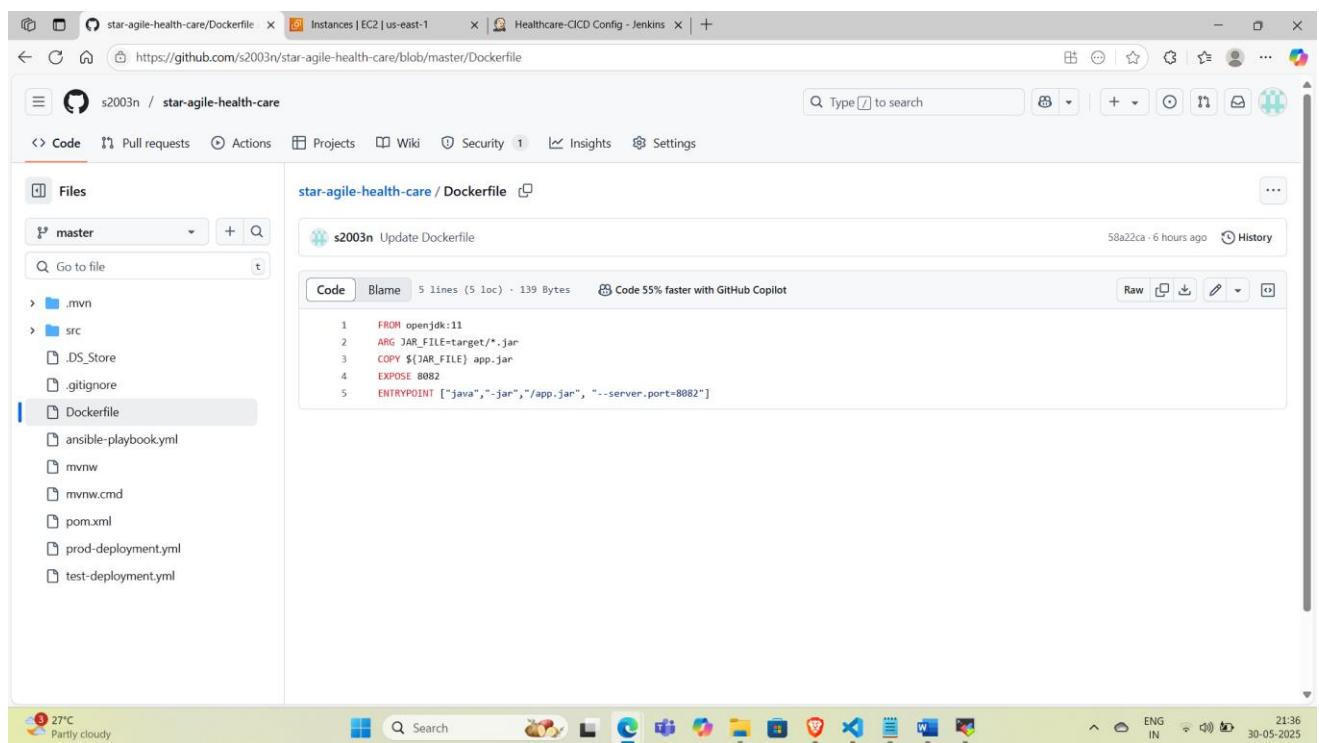


- Add the github repository in the general configurations add the forked repository:

- Add git triggers:



- Write the Dockerfile:



- Create a k8s files and write the deployment scripts of both production and test deployment with namespace production and tag prod This will ensure that the applications are deployed on the assigned tags and namespace and app server port :8082

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
  namespace: test
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      nodeSelector:
        env: test # This targets the test-labeled node
      containers:
        - name: my-app
          image: saikumar2003/medicure:latest
      ports:
        - containerPort: 8082
      ...
apiVersion: v1
kind: Service
metadata:
  name: my-app-service
  namespace: test
spec:
  type: NodePort
  selector:
    app: my-app
  ports:
    - port: 80
      targetPort: 8082
      nodePort: 30090 # Test app accessible via http://<test-node-ip>:30090

```

Also write the test deployment script with the namespace test and tag env and server port of app as 8082,of test this will ensure that the application is deployed on the specified nodes only

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
  namespace: production
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      nodeSelector:
        env: production # This targets nodes labeled as production
      containers:
        - name: my-app
          image: saikumar2003/medicure:latest
      ports:
        - containerPort: 8082
      ...
apiVersion: v1
kind: Service
metadata:
  name: my-app-service
  namespace: production
spec:
  type: NodePort
  selector:
    app: my-app
  ports:
    - port: 80
      targetPort: 8082
      nodePort: 30091 # Prod app accessible via http://<prod-node-ip>:30091

```

- Write the Jenkins script to deploy the application:

```

pipeline {
    agent any

    stages {

        stage('Checkout Code') {
            steps {
                git url: 'https://github.com/s2003n/star-agile-health-care.git', branch: 'master'
            }
        }

        stage('Build with Maven') {
            steps {
                sh './mvnw clean package'
            }
        }

        stage('Build Docker Image') {
            steps {
                sh 'docker build -t saikumar2003/medicure:latest .'
            }
        }

        stage('Push Docker Image') {
            steps {
                withCredentials([usernamePassword(credentialsId: 'dockercreds',
usernameVariable: 'USERNAME', passwordVariable: 'PASSWORD')]) {
                    sh """
                        echo $PASSWORD | docker login -u $USERNAME --password-stdin
                        docker push saikumar2003/medicure:latest
                    """
                }
            }
        }

        stage('Deploy to Test') {
            steps {
                kubernetesDeploy(
                    kubeconfigId: 'k8screds',
                    configs: 'test-deployment.yml'
                )
            }
        }
    }
}

```

```
        )
    }
}

stage('Wait for Test App to Be Ready') {
    steps {
        sh 'sleep 5'
    }
}

stage('Run Unit Tests') {
    steps {
        sh './mvnw test'
    }
}

stage('Check Build Status') {
    steps {
        script {
            echo "Current build result: ${currentBuild.result}"
        }
    }
}

stage('Publish JUnit Test Results') {
    steps {
        junit 'target/surefire-reports/TEST-*.xml'
    }
}

stage('Deploy to Production') {
    when {
        expression { currentBuild.result == null || currentBuild.result == 'SUCCESS' }
    }
    steps {
        kubernetesDeploy(
            kubeconfigId: 'k8screds',
            configs: 'prod-deployment.yml'
        )
    }
}

}
```

- Add github webhook to the repository from settings and add the Jenkins ip address link

The screenshot shows the GitHub repository settings page for 'star-agile-health-care'. The left sidebar is expanded to show the 'Webhooks' section. The main area is titled 'Webhooks / Add webhook'. It contains fields for 'Payload URL' (set to 'http://44.203.95.187:8080/github-webhook'), 'Content type' (set to 'application/json'), and a 'Secret' field. Under 'SSL verification', the 'Enable SSL verification' option is selected. In the 'Which events would you like to trigger this webhook?' section, the 'Just the push event.' option is selected. The status bar at the bottom indicates it's 21:52 on 30-05-2025.

The screenshot shows the GitHub repository settings page for 'star-agile-health-care'. The left sidebar is expanded to show the 'Webhooks' section. The main area is titled 'Webhooks' and displays a single webhook entry with the URL 'http://44.203.95.187:8080/github-w... (push)'. A warning message states 'Last delivery was not successful. Invalid HTTP Response: 302.' with 'Edit' and 'Delete' buttons. The status bar at the bottom indicates it's 21:54 on 30-05-2025.

Before running the pipeline make sure **Jenkins user doesn't lacks Docker access**

- If the Jenkins user does not have permission to execute Docker commands.
- **Fix:** Add the Jenkins user to the Docker group:
`sudo usermod -aG docker jenkins`
`sudo systemctl restart Jenkins`

- Now, Run the Jenkins pipeline

The screenshot shows a Jenkins pipeline console output. The pipeline starts by checking out code from a GitHub repository. It then performs several git operations: fetching upstream changes, checking out Revision 5262248d88f6df051d3228254dde51a6e820319, and finally committing and pushing changes. The commit message is "Delete src/k8s directory". The pipeline concludes with a success message and a timestamp of 2023-05-05T10:23:19Z.

```
Started by user Saikumar
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Healthcare-CI_CD
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout Code)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Healthcare-CI_CD/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/s2003n/star-agile-health-care.git # timeout=10
Fetching upstream changes from https://github.com/s2003n/star-agile-health-care.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/s2003n/star-agile-health-care.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 5262248d88f6df051d3228254dde51a6e820319 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 5262248d88f6df051d3228254dde51a6e820319 # timeout=10
> git branch -v --no-abbrev # timeout=10
> git branch -D master # timeout=10
> git checkout -b master 5262248d88f6df051d3228254dde51a6e820319 # timeout=10
Commit message: "Delete src/k8s directory"
> git rev-list --no-walk 5262248d88f6df051d3228254dde51a6e820319 # timeout=10
[Pipeline] }
```

Dashboard > Healthcare-CI_CD > #2

```
Kind=null, StrVal=25%, additionalProperties={}), maxAvailable=Integer(String(IntVal=null, Kind=null, StrVal=25%, additionalProperties={}), additionalProperties={}), additionalProperties={})), type=RollingUpdate, additionalProperties={})), template=PodTemplateSpec(metadata=ObjectMeta(annotations=null, clusterName=null, creationTimestamp=null, deletionGracePeriodSeconds=null, deletionTimestamp=null, finalizers=[], generateName=null, generation=null, initializers=null, labels={app=my-app}, name=null, namespace=null, ownerReferences=[], resourceVersion=null, selfLink=null, uid=null, additionalProperties={})), spec=PodSpec(activeDeadlineSeconds=null, affinity=null, automountServiceAccountToken=null, containers=[Container(args=[], command=[], env=[], envFrom=[], image=sai-kumar/2003/medicure:latest, imagePullPolicyAlways, lifecycle=null, livenessProbe=null, name=my-app, ports=[ContainerPort(containerPort=8082, hostIP=null, hostPort=null, name=null, protocol=TCP, additionalProperties={})], readinessProbe=null, resources=ResourceRequirements(limits=null, requests=null, additionalProperties={})), securityContext=null, stdIn=null, stdInOnce=null, terminationMessagePath=/dev/termination-log, terminationMessagePolicy=File, tty=null, volumeDevices=[]), volumeMounts=[], workingDir=null, additionalProperties={})), dnsConfig=null, dnsPolicy=ClusterFirst, hostAliases=[], hostIP=null, hostNetwork=null, hostPID=null, hostname=null, imagePullSecrets[], initContainers[], nodeName=null, nodeSelector={env=prod}, priority=null, priorityClassName=null, restartPolicy=Always, schedulerName=default-scheduler, securityContext=PodSecurityContext(fsgroup=null, runAsNonRoot=null, runAsUser=null, selinuxOptions=null, supplementalGroups=[], additionalProperties={})), serviceAccount=null, serviceAccountName=null, subdomain=null, terminationGracePeriodSeconds=30, tolerations=[], volumes=[]), additionalProperties={})), additionalProperties={})), status=DeploymentStatus(availableReplicas=null, collisionCount=null, conditions=[], observedGeneration=null, readyReplicas=null, replicas=null, unavailableReplicas=null, updatedReplicas=null, additionalProperties={})), additionalProperties={})))
```

Created Service: Service(apiVersion=v1, kind=Service, metadata=ObjectMeta(annotations=null, clusterName=null, creationTimestamp=2025-05-30T16:34:46Z, deletionGracePeriodSeconds=null, deletionTimestamp=null, finalizers=[], generateName=null, generation=null, initializers=null, labels=null, name=my-app-service, namespace=production, ownerReferences=[], resourceVersion=22895, selfLink=null, uid=ee738e7-a-f08-43cd-84e9-1cd108ebb57d, additionalProperties={managedFields=[{manager=okhttp, operation=Update, apiVersion=v1, time=2025-05-30T16:34:46Z, fieldsType=FieldsV1, fieldsV1={f:spec={f:externalTrafficPolicy={}, f:internalTrafficPolicy={}, f:ports={(.).k:{port=:80, protocol:"TCP"}=(.){.f:.nodePort=null, f:.port=null, f:.targetPort=null, f:.selector=null, f:.sessionAffinity=null, f:.type=(.))}}}}, spec=ServiceSpec(clusterIP=10.111.9.200, externalIPs=[], externalName=null, externalTrafficPolicy=Cluster, healthCheckNodePort=null, loadBalancerIP=null, loadBalancerSourceRanges=[], ports=[ServicePort(name=null, nodePort=30091, port=80, protocol=TCP, targetPort=Integer(IntVal=8082, Kind=null, StrVal=null, additionalProperties={})), additionalProperties={}]), publishNotReadyAddresses=null, selector=(app=my-app), sessionAffinity=None, sessionAffinityConfig=null, type=NodePort, additionalProperties={clusterIPs=[10.111.9.200], ipFamilies=[IPv4], ipFamilyPolicy=SingleStack, internalTrafficPolicy=Cluster}), status=ServiceStatus(loadBalancer=LoadBalancerStatus(ingress=[], additionalProperties={})), additionalProperties={})))

Finished Kubernetes deployment

[Pipeline] }

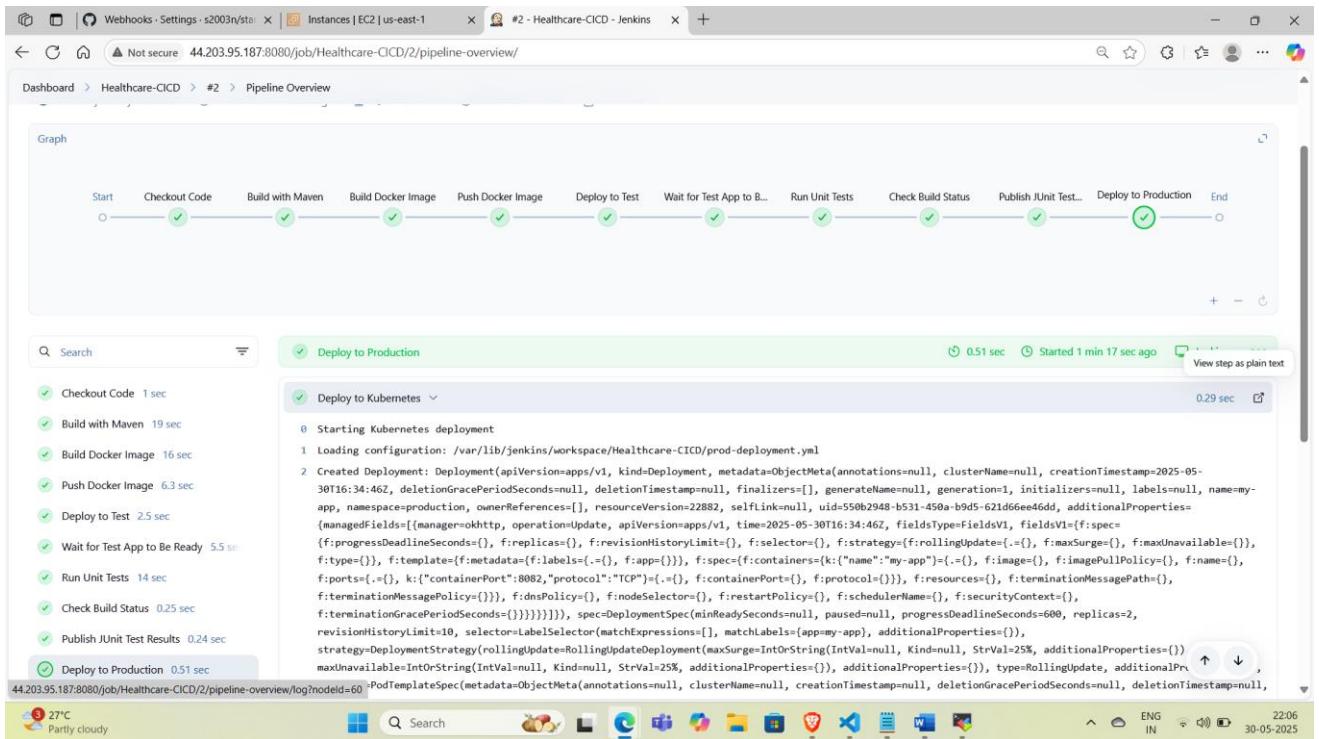
[Pipeline] // stage

[Pipeline] }

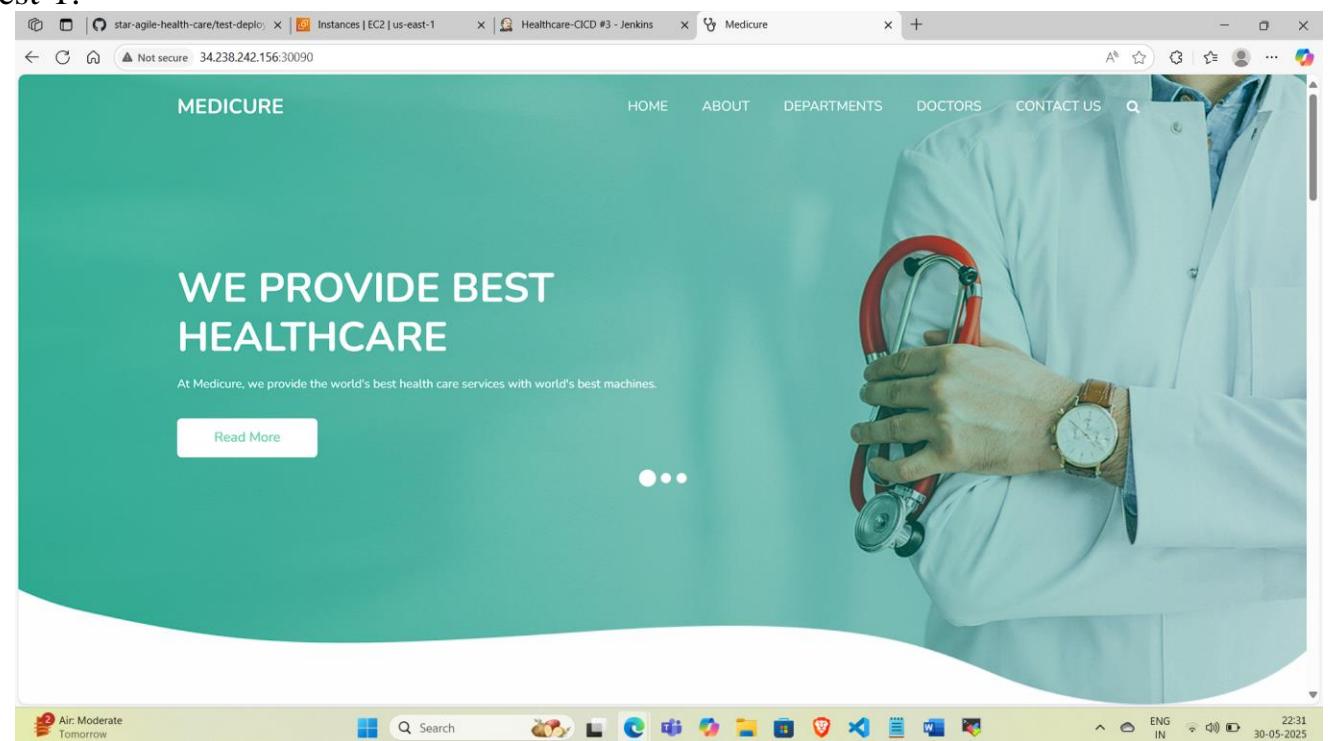
[Pipeline] // node

[Pipeline] End of Pipeline

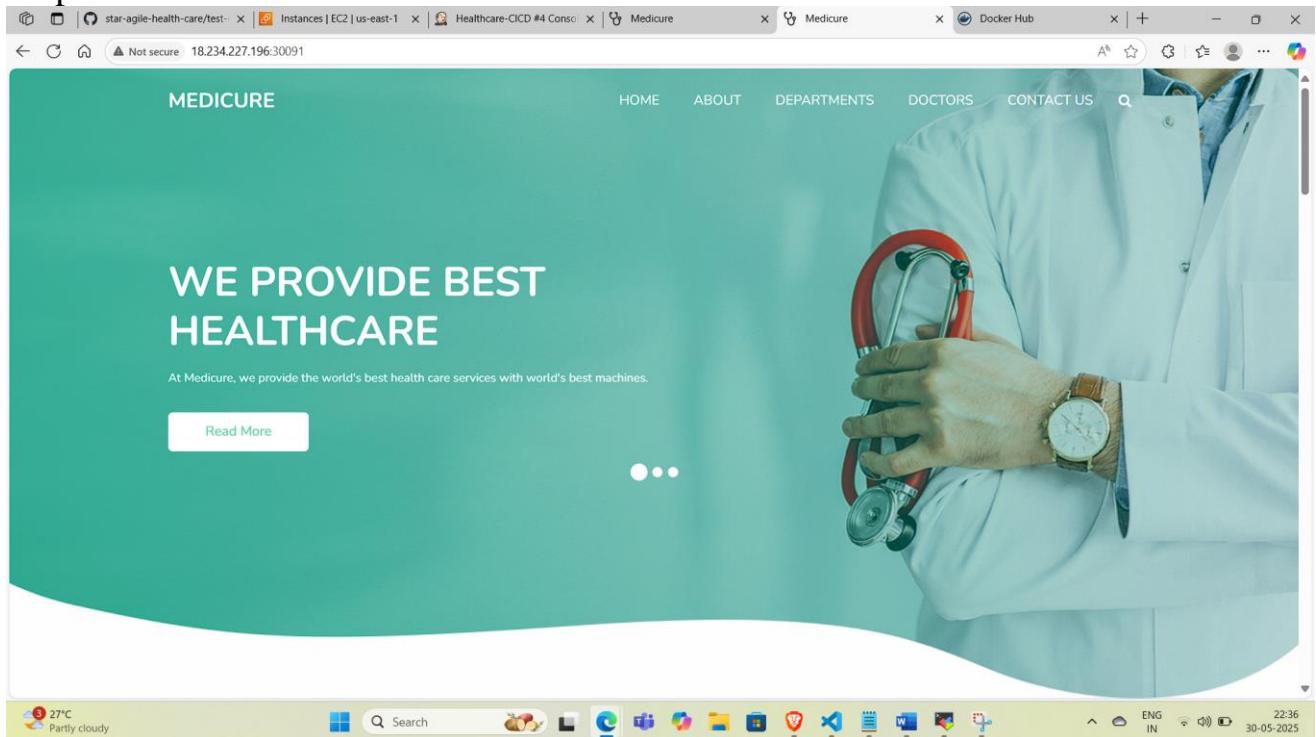
Finished: SUCCESS



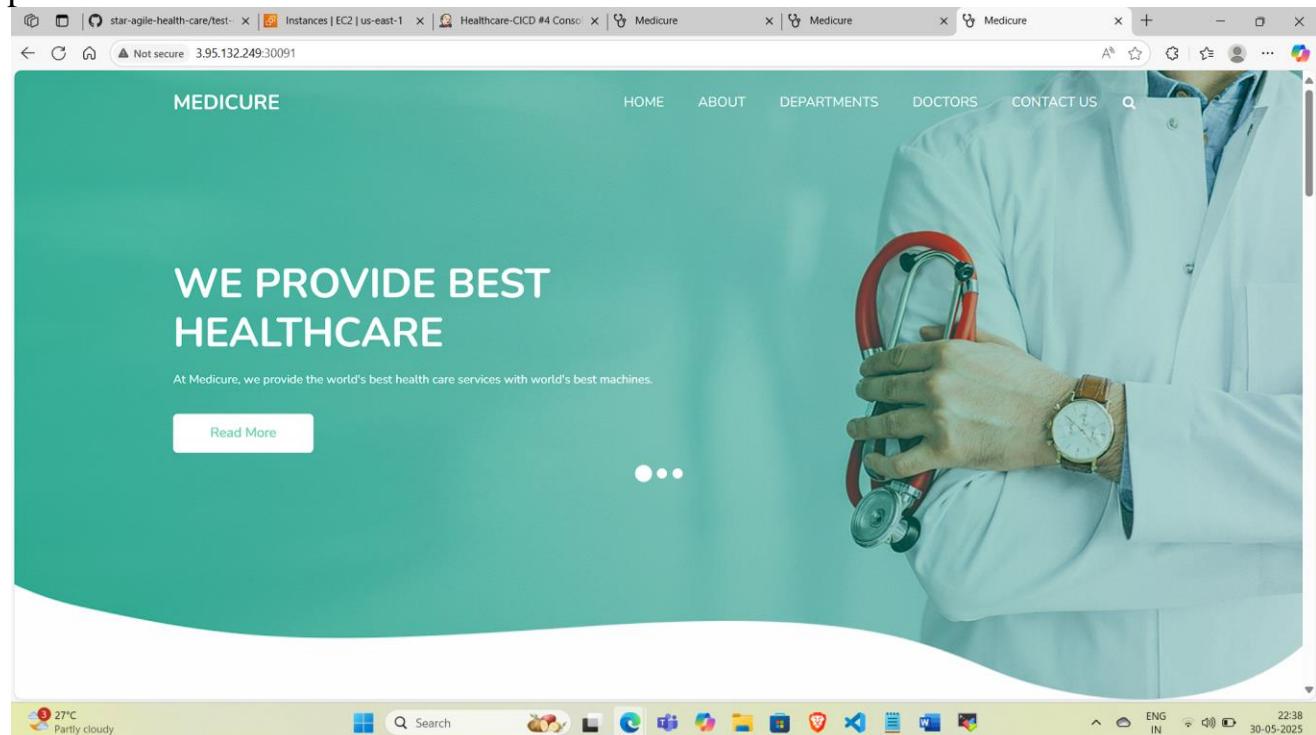
- After that access the application on the test server using the public ip address on node port number 30090 as mentioned in test-deployment.yml script:



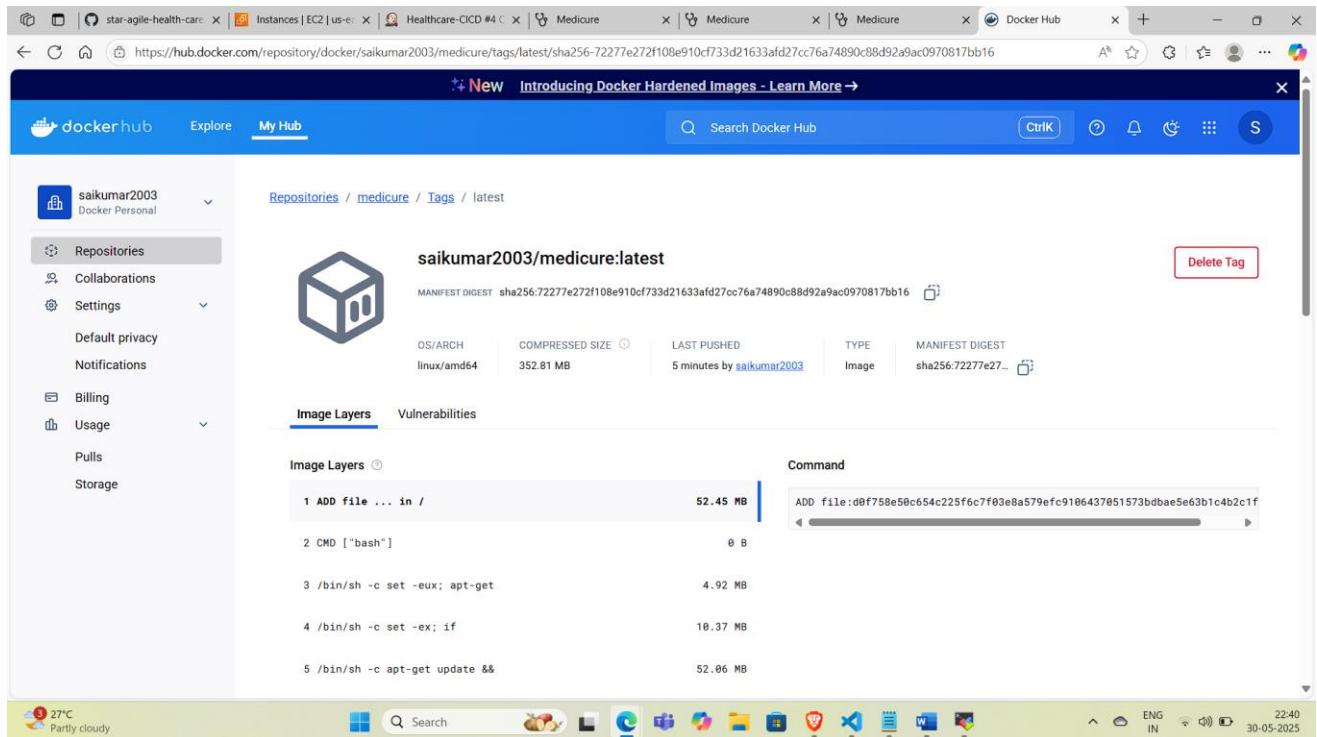
- After that access the application on both the production servers using the public ip address on port number 30091 as mentioned in the prod-deployment.yml script: prod-1:



- Access the application on the other production server(prod-2) also: prod-2:



- Check the docker hub if the image got created correctly:



Prometheus and Grafana setup on monitoring server:

Now in both production machines

Step 1: Update your system packages

```
sudo apt-get update && sudo apt-get upgrade -y
sudo yum update -y
```

Step 2: Create a dedicated user for Node Exporter

```
sudo useradd --no-create-home --shell /bin/false node_exporter
```

```

prod-1
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
[home/ubuntu/] 1 master-host 2 test-1 3 prod-1 4 prod-2 5 monitoring 6 monitoring
/home/ubuntu/
Processing triggers for install-info (7.1-3build2) ...
Processing triggers for unitramfs-tools (0.14ubuntu25.5) ...
update-initramfs: Generating /boot/initrd.img-6.8.0-1024-aws
Processing triggers for libc-bin (2.39-0ubuntu0.4) ...
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning candidates...
Scanning linux images...
Running kernel seems to be up-to-date.

Restarting services...
systemctl restart irqbalance.service multipathd.service packagekit.service polkit.service udisks2.service

Service restarts being deferred:
systemctl restart ModemManager.service
/etc/needrestart/restart.d/dbus.service
systemctl restart docker.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #35: bash[8751], sshd[4027]
ubuntu @ session #36: sshd[4029,4142]
ubuntu @ user manager service: systemd[1131]

No VM guests are running outdated hypervisor (qemu) binaries on this host.

sudo: yum: command not found
root@prod-1:/home/devops# sudo useradd --no-create-home --shell /bin/false node_exporter
root@prod-1:/home/devops# exit
exit
devops@prod-1:$ exit
logout
root@prod-1:/home/ubuntu# sudo useradd --no-create-home --shell /bin/false node_exporter
useradd: user 'node_exporter' already exists
root@prod-1:/home/ubuntu# 

```

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

27°C Party cloudy 22:47 ENG IN 30-05-2025

Step 3: Download the latest Node Exporter binary

Change to a temporary directory and download Node Exporter (replace version with latest if needed):

```

cd /usr/src
sudo wget
https://github.com/prometheus/node\_exporter/releases/download/v1.8.0/node\_exporter-1.8.0.linux-amd64.tar.gz

```

```

prod-1
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
[home/ubuntu/] 1 master-host 2 test-1 3 prod-1 4 prod-2 5 monitoring 6 monitoring
/home/ubuntu/
Ubuntu sessions running outdated binaries:
ubuntu @ session #35: bash[8751], sshd[4027]
ubuntu @ session #36: sshd[4029,4142]
ubuntu @ user manager service: systemd[1131]

No VM guests are running outdated hypervisor (qemu) binaries on this host.

sudo: yum: command not found
root@prod-1:/home/devops# sudo useradd --no-create-home --shell /bin/false node_exporter
root@prod-1:/home/devops# exit
exit
devops@prod-1:$ exit
logout
root@prod-1:/home/ubuntu# sudo useradd --no-create-home --shell /bin/false node_exporter
useradd: user 'node_exporter' already exists
root@prod-1:/home/ubuntu# cd /usr/src
root@prod-1:/home/ubuntu# sudo wget https://github.com/prometheus/node_exporter/releases/download/v1.8.0/node_exporter-1.8.0.linux-amd64.tar.gz
--2025-05-30 17:18:12 - https://github.com/prometheus/node_exporter/releases/download/v1.8.0/node_exporter-1.8.0.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10675730 [application/octet-stream]
Saving to: 'node_exporter-1.8.0.linux-amd64.tar.gz'

node_exporter-1.8.0.linux-amd64.tar.gz 100%[=====] 10.18M 0.03s
2025-05-30 17:18:12 (325 MB/s) - 'node_exporter-1.8.0.linux-amd64.tar.gz' saved [10675730/10675730]

root@prod-1:/usr/src# 

```

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

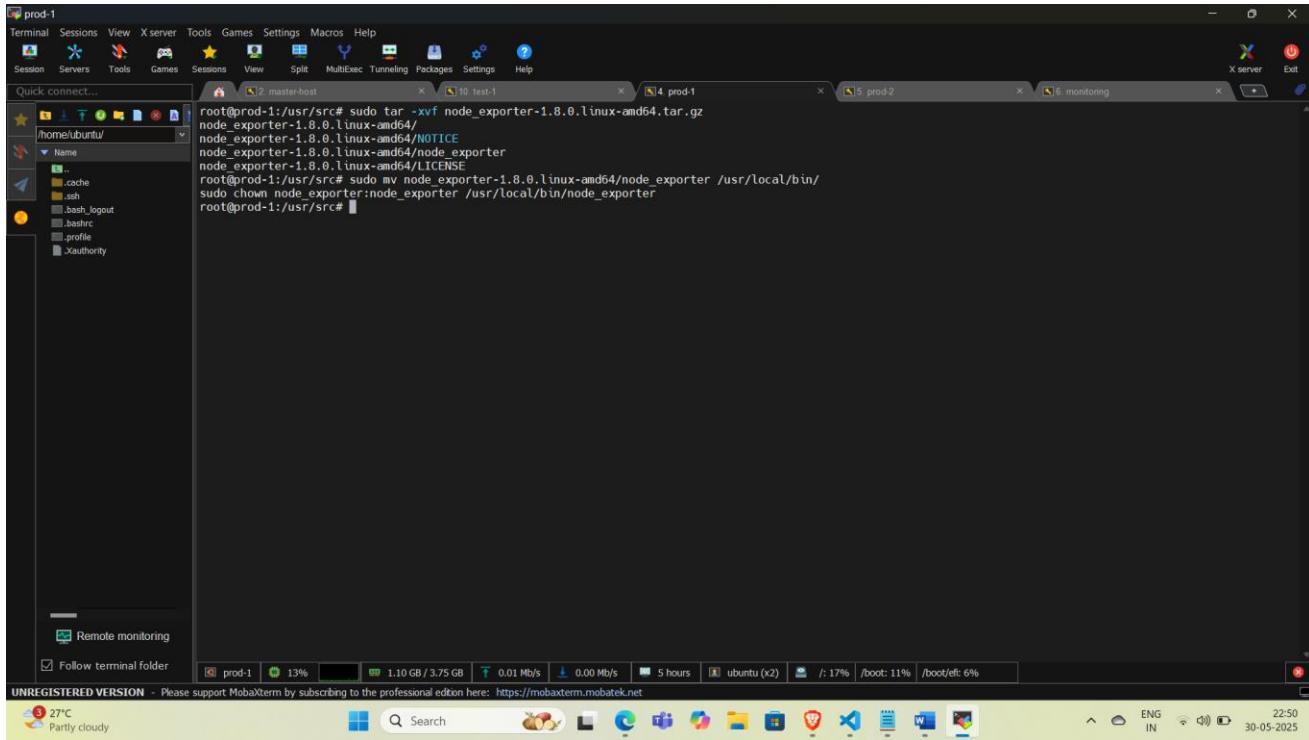
27°C Party cloudy 22:49 ENG IN 30-05-2025

Step 4: Extract the downloaded archive

```
sudo tar -xvf node_exporter-1.8.0.linux-amd64.tar.gz
```

Step 5: Move the Node Exporter binary to /usr/local/bin and set ownership

```
sudo mv node_exporter-1.8.0.linux-amd64/node_exporter /usr/local/bin/  
sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter
```



Step 6: Create a systemd service file for Node Exporter

Open the service file for editing:

```
sudo nano /etc/systemd/system/node_exporter.service
```

Paste the following content:

[Unit]

Description=Prometheus Node Exporter

Wants=network-online.target

After=network-online.target

[Service]

User=node_exporter

Group=node_exporter

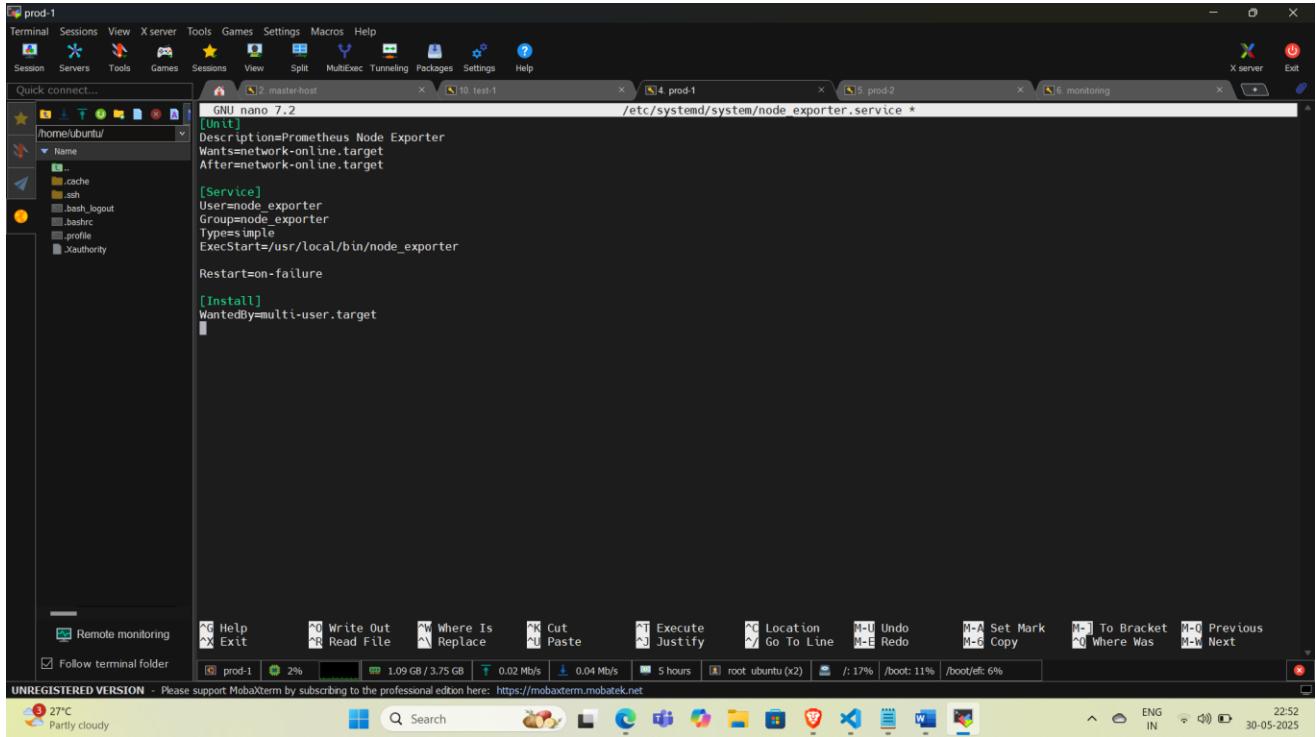
Type=simple

ExecStart=/usr/local/bin/node_exporter

Restart=on-failure

[Install]

WantedBy=multi-user.target



```
[Unit]
Description=Prometheus Node Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=node_exporter
Group=node_exporter
Type=immediate
ExecStart=/usr/local/bin/node_exporter

Restart=on-failure

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

Save and exit (Ctrl+O, Enter, Ctrl+X).

Step 7: Reload systemd, enable and start Node Exporter service

sudo systemctl daemon-reload

sudo systemctl enable node_exporter

sudo systemctl start node_exporter

Step 8: Verify Node Exporter is running

sudo systemctl status node_exporter

You should see the service active and running.

```

prod-1
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ubuntu/
Name
.. .cache .ssh .bash_logout .bashrc .profile .xauthority
root@prod-1:~/src# sudo tar -xvf node_exporter-1.8.0.linux-amd64.tar.gz
node_exporter-1.8.0.linux-amd64/NOTICE
node_exporter-1.8.0.linux-amd64/node_exporter
node_exporter-1.8.0.linux-amd64/LICENSE
root@prod-1:~/src# sudo mv node_exporter-1.8.0.linux-amd64/node_exporter /usr/local/bin/
sudo chmod 755 node_exporter
root@prod-1:~/src# sudo nano /etc/systemd/system/node_exporter.service
root@prod-1:~/src# sudo systemctl daemon-reload
sudo systemctl enable node_exporter
sudo systemctl start node_exporter
Created symlink /etc/systemd/system/multi-user.target.wants/node_exporter.service → /etc/systemd/system/node_exporter.service.
root@prod-1:~/src# sudo systemctl status node_exporter
● node_exporter.service - Prometheus Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-05-30 17:24:03 UTC; 16s ago
       Main PID: 91280 (node exporter)
         Tasks: 4 (limit: 4584)
        Memory: 2.6M (peak: 2.8M)
          CPU: 12ms
        CGroup: /system.slice/node_exporter.service
           └─ 91280 /usr/local/bin/node_exporter

May 30 17:24:03 prod-1 node_exporter[91280]: ts=2025-05-30T17:24:03.122Z caller=node_exporter.go:118 level=info collector=ttime
May 30 17:24:03 prod-1 node_exporter[91280]: ts=2025-05-30T17:24:03.122Z caller=node_exporter.go:118 level=info collector=timex
May 30 17:24:03 prod-1 node_exporter[91280]: ts=2025-05-30T17:24:03.122Z caller=node_exporter.go:118 level=info collector=udp_queues
May 30 17:24:03 prod-1 node_exporter[91280]: ts=2025-05-30T17:24:03.122Z caller=node_exporter.go:118 level=info collector=username
May 30 17:24:03 prod-1 node_exporter[91280]: ts=2025-05-30T17:24:03.122Z caller=node_exporter.go:118 level=info collector=vstat
May 30 17:24:03 prod-1 node_exporter[91280]: ts=2025-05-30T17:24:03.122Z caller=node_exporter.go:118 level=info collector=watchdog
May 30 17:24:03 prod-1 node_exporter[91280]: ts=2025-05-30T17:24:03.122Z caller=node_exporter.go:118 level=info collector=xfs
May 30 17:24:03 prod-1 node_exporter[91280]: ts=2025-05-30T17:24:03.122Z caller=node_exporter.go:118 level=info collector=zfs
May 30 17:24:03 prod-1 node_exporter[91280]: ts=2025-05-30T17:24:03.122Z caller=tls_config.go:313 level=info msg="Listening on" address=[::]:9100
May 30 17:24:03 prod-1 node_exporter[91280]: ts=2025-05-30T17:24:03.122Z caller=tls_config.go:316 level=info msg="TLS is disabled." http2=false address=[::]:9100
root@prod-1:~/src#

```

Remote monitoring

Follow terminal folder

prod-1 3% 1.09 GB / 3.75 GB 0.01 Mb/s 0.00 Mb/s 5 hours ubuntu (x2) 17% /boot: 11% /boot/ef: 6%

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

27°C Partly cloudy 22:54 ENG IN 30-05-2025

```

prod-2
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ubuntu/
Name
.. .cache .ssh .bash_logout .bashrc .profile .xauthority
root@prod-2:~/src# sudo tar -xvf node_exporter-1.8.0.linux-amd64.tar.gz
Saving to: 'node_exporter-1.8.0.linux-amd64.tar.gz'
2025-05-30 17:28:26 (214 MB/s) - 'node_exporter-1.8.0.linux-amd64.tar.gz' saved [10675730/10675730]

root@prod-2:~/src# sudo tar -xvf node_exporter-1.8.0.linux-amd64.tar.gz
node_exporter-1.8.0.linux-amd64/NOTICE
node_exporter-1.8.0.linux-amd64/node_exporter
node_exporter-1.8.0.linux-amd64/LICENSE
root@prod-2:~/src# sudo mv node_exporter /usr/local/bin/node_exporter
sudo chmod 755 node_exporter
root@prod-2:~/src# sudo nano /etc/systemd/system/node_exporter.service
root@prod-2:~/src# sudo systemctl daemon-reload
sudo systemctl start node_exporter
Created symlink /etc/systemd/system/multi-user.target.wants/node_exporter.service → /etc/systemd/system/node_exporter.service.
root@prod-2:~/src# sudo systemctl status node_exporter
● node_exporter.service - Prometheus Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-05-30 17:29:51 UTC, 45s ago
       Main PID: 92670 (node exporter)
         Tasks: 5 (limit: 4584)
        Memory: 2.4M (peak: 2.6M)
          CPU: 11ms
        CGroup: /system.slice/node_exporter.service
           └─ 92670 /usr/local/bin/node_exporter

May 30 17:29:51 prod-2 node_exporter[92670]: ts=2025-05-30T17:29:51.546Z caller=node_exporter.go:118 level=info collector=ttime
May 30 17:29:51 prod-2 node_exporter[92670]: ts=2025-05-30T17:29:51.546Z caller=node_exporter.go:118 level=info collector=timex
May 30 17:29:51 prod-2 node_exporter[92670]: ts=2025-05-30T17:29:51.546Z caller=node_exporter.go:118 level=info collector=udp_queues
May 30 17:29:51 prod-2 node_exporter[92670]: ts=2025-05-30T17:29:51.546Z caller=node_exporter.go:118 level=info collector=username
May 30 17:29:51 prod-2 node_exporter[92670]: ts=2025-05-30T17:29:51.546Z caller=node_exporter.go:118 level=info collector=vstat
May 30 17:29:51 prod-2 node_exporter[92670]: ts=2025-05-30T17:29:51.546Z caller=node_exporter.go:118 level=info collector=watchdog
May 30 17:29:51 prod-2 node_exporter[92670]: ts=2025-05-30T17:29:51.546Z caller=node_exporter.go:118 level=info collector=xfs
May 30 17:29:51 prod-2 node_exporter[92670]: ts=2025-05-30T17:29:51.546Z caller=node_exporter.go:118 level=info collector=zfs
May 30 17:29:51 prod-2 node_exporter[92670]: ts=2025-05-30T17:29:51.547Z caller=tls_config.go:313 level=info msg="Listening on" address=[::]:9100
May 30 17:29:51 prod-2 node_exporter[92670]: ts=2025-05-30T17:29:51.547Z caller=tls_config.go:316 level=info msg="TLS is disabled." http2=false address=[::]:9100
root@prod-2:~/src#

```

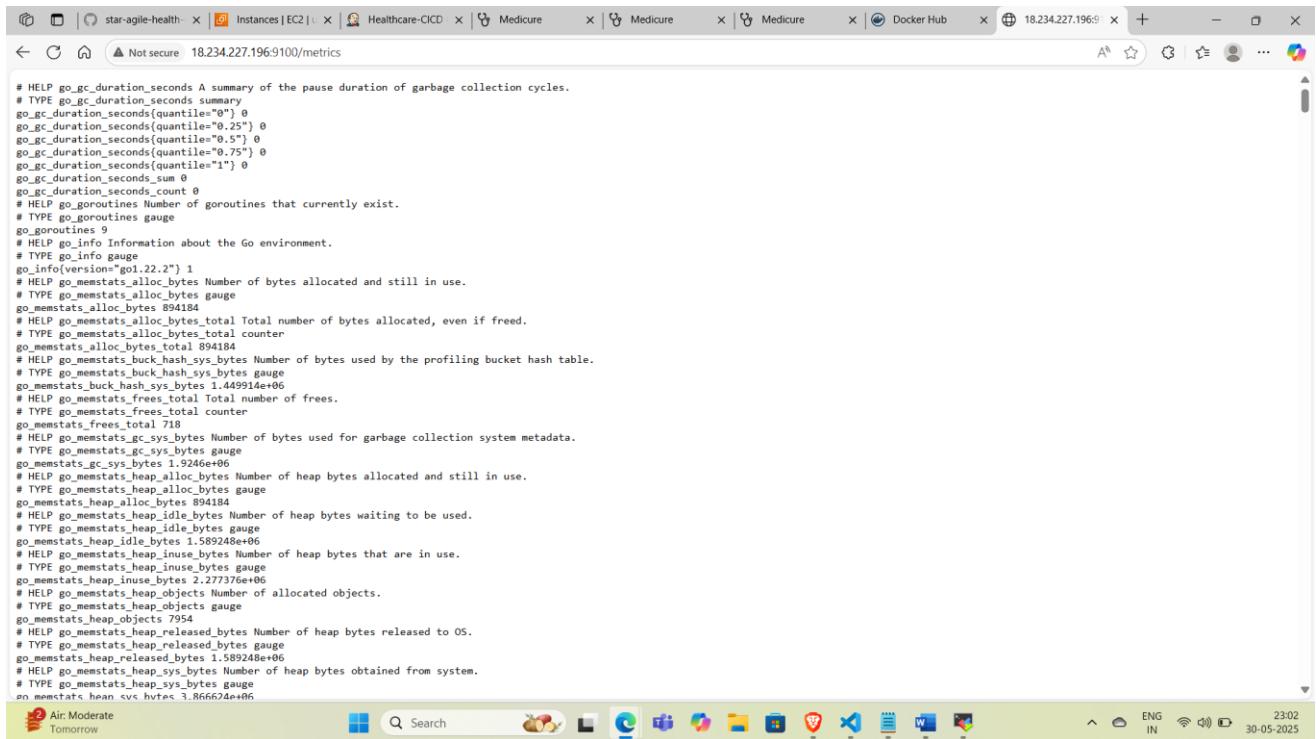
Air: Moderate Tomorrow 23:01 ENG IN 30-05-2025

Step 9: Confirm Node Exporter metrics are accessible

Open a browser or use curl to check metrics endpoint:

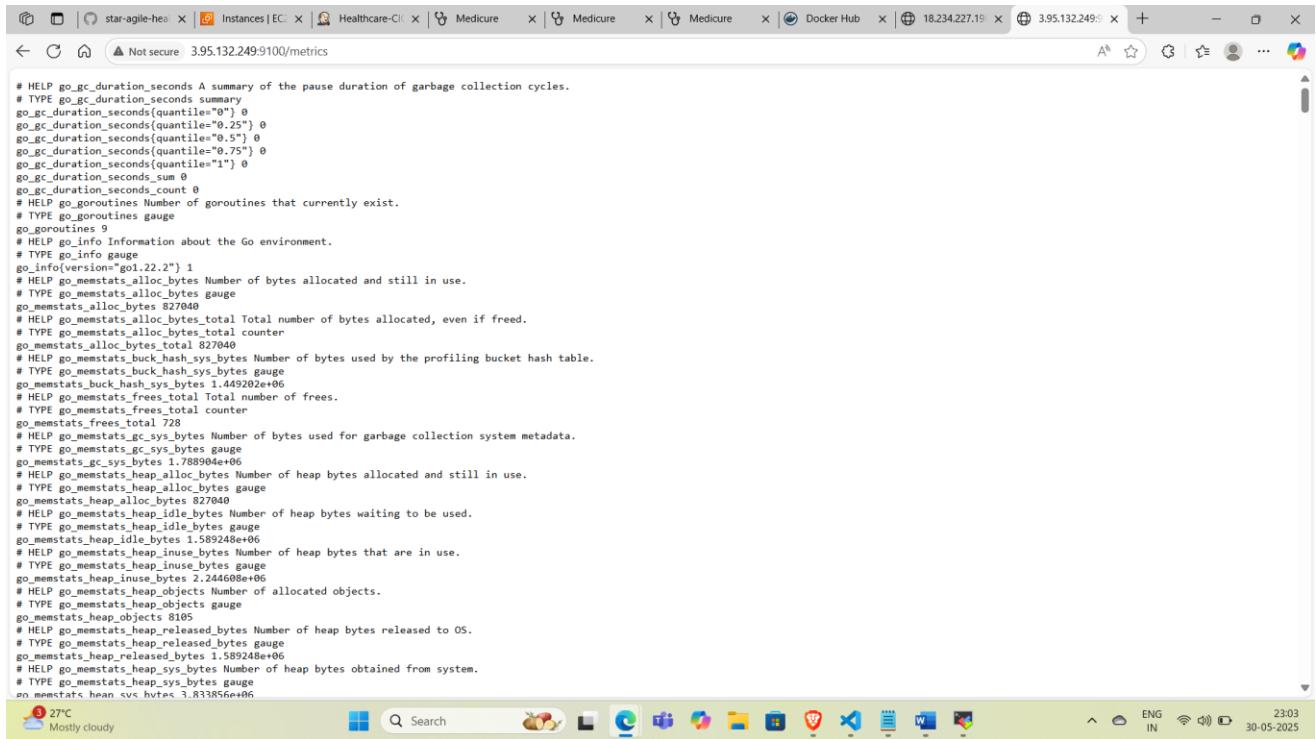
prod1-public-ip>:9100/metrics

prod2-public-ip>:9100/metrics



```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 9
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.22.2"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 894184
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 894184
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.449914e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 718
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 1.5246e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 894184
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 1.589248e+06
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 2.277376e+06
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 8105
# HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
# TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 1.589248e+06
# HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from system.
# TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 3.866624e+06

```



```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 9
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.22.2"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 827040
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 827040
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.449202e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 728
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 1.788904e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 827040
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 1.589248e+06
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 2.244608e+06
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 8105
# HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
# TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 1.589248e+06
# HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from system.
# TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 3.833856e+06

```

You should see a list of metrics output, confirming Node Exporter is working correctly.

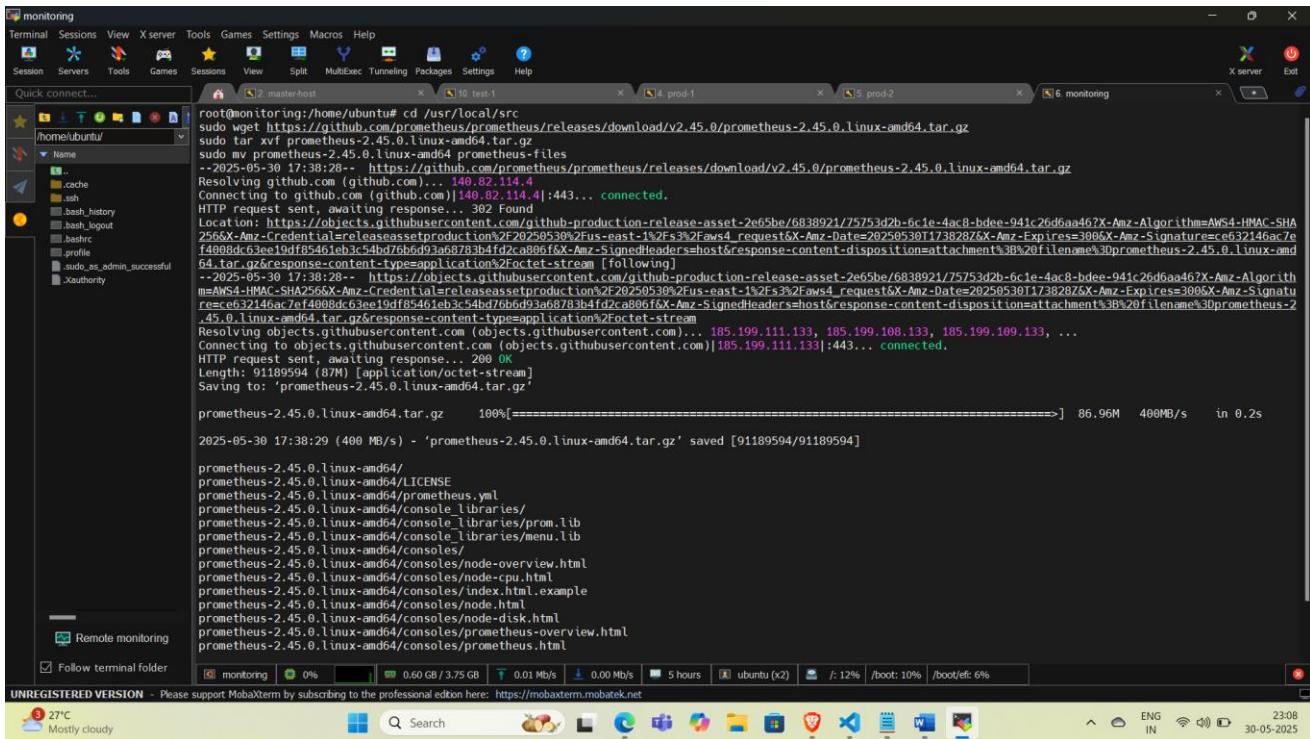
This procedure installs Node Exporter as a system service running under a dedicated user, ensuring proper permissions and automatic startup on boot.

The default port is 9100.

Install and Configure Prometheus:

Download and extract Prometheus

```
cd /usr/local/src  
sudo wget https://github.com/prometheus/prometheus/releases/download/v2.45.0/prometheus-2.45.0.linux-amd64.tar.gz  
sudo tar xvf prometheus-2.45.0.linux-amd64.tar.gz  
sudo mv prometheus-2.45.0.linux-amd64 prometheus-files
```



The screenshot shows a MobaXterm window titled 'monitoring' with several tabs open. The current tab displays a terminal session on 'master-host'. The user has run the command to download the Prometheus binary and extract it. The terminal output shows the download progress and the extraction of files into the 'prometheus' directory. The status bar at the bottom provides system information like CPU usage, memory, and network.

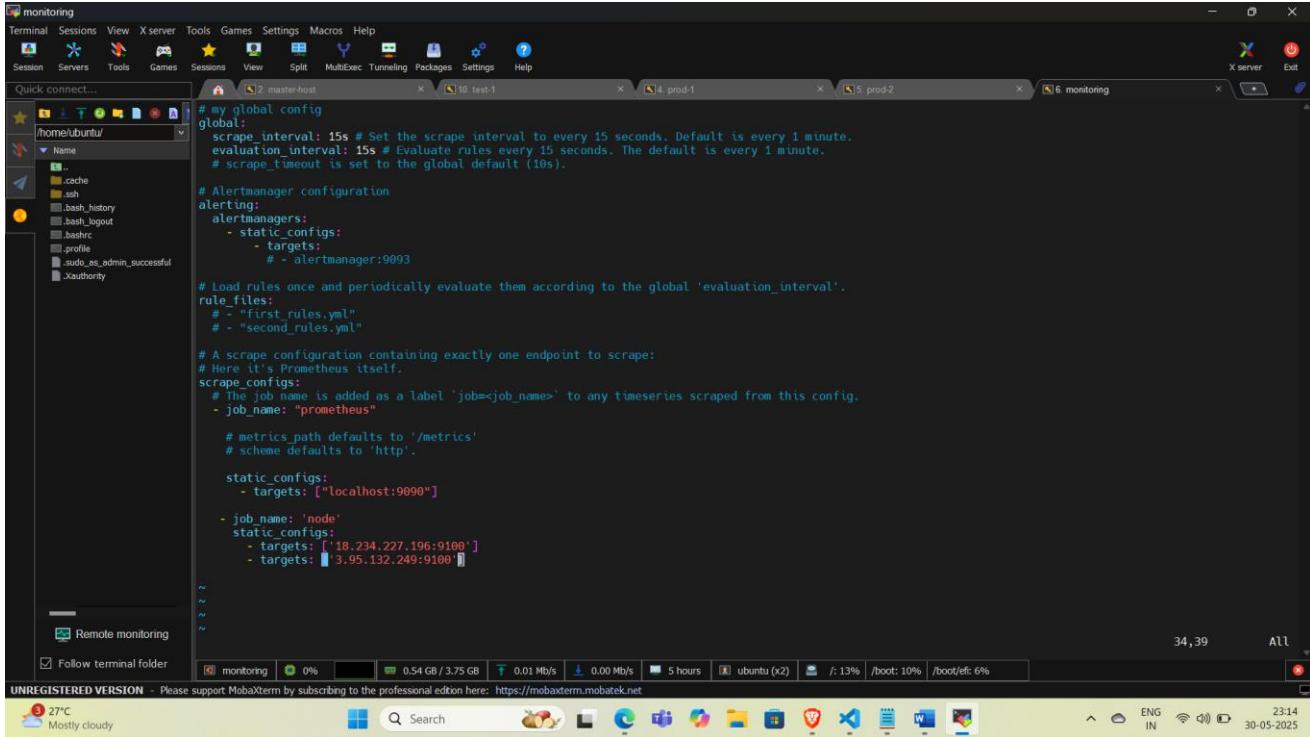
```
root@monitoring:/home/ubuntu# cd /usr/local/src  
root@monitoring:/home/ubuntu# sudo wget https://github.com/prometheus/prometheus/releases/download/v2.45.0/prometheus-2.45.0.linux-amd64.tar.gz  
root@monitoring:/home/ubuntu# sudo tar xvf prometheus-2.45.0.linux-amd64.tar.gz  
root@monitoring:/home/ubuntu# sudo mv prometheus-2.45.0.linux-amd64 prometheus-files  
--2025-05-30 17:38:28 - https://github.com/prometheus/prometheus/releases/download/v2.45.0/prometheus-2.45.0.linux-amd64.tar.gz  
Resolving github.com (github.com)... 140.82.114.4:443... connected.  
HTTP request sent, awaiting response... 302 Found  
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/75753d2b-6c1e-4ac8-bdee-941c26d6aa467X-Amz-Algorithm=AWS4-HMAC-SHA256X-Amz-Credential=releaseassetprod@F20250530T173828ZX-Amz-Expires=3086X-Amz-Signature=c632146ac7ef4008dc63ee19df8346eb3c54bd766d93a687834fd2ca806f6X-Amz-SignedHeaders=host&response-content-disposition=attachment%3Bfilename%3Dprometheus-2.45.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]  
--2025-05-30 17:38:28 - https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/75753d2b-6c1e-4ac8-bdee-941c26d6aa467X-Amz-Algorithm=AWS4-HMAC-SHA256X-Amz-Credential=releaseassetprod@F20250530T173828ZX-Amz-Expires=3086X-Amz-Signature=c632146ac7ef4008dc63ee19df8346eb3c54bd766d93a687834fd2ca806f6X-Amz-SignedHeaders=host&response-content-disposition=attachment%3Bfilename%3Dprometheus-2.45.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream  
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133,...  
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.111.133|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 91189594 (87M) [application/octet-stream]  
Saving to: 'prometheus-2.45.0.linux-amd64.tar.gz'  
prometheus-2.45.0.linux-amd64.tar.gz 100%[=====] 86.96M 400MB/s in 0.2s  
2025-05-30 17:38:29 (400 MB/s) - 'prometheus-2.45.0.linux-amd64.tar.gz' saved [91189594/91189594]  
prometheus-2.45.0.linux-amd64/  
prometheus-2.45.0.linux-amd64/LICENSE  
prometheus-2.45.0.linux-amd64/prometheus.yml  
prometheus-2.45.0.linux-amd64/console_libraries/  
prometheus-2.45.0.linux-amd64/console_libraries/promql  
prometheus-2.45.0.linux-amd64/console_libraries/menu.lib  
prometheus-2.45.0.linux-amd64/consoles/  
prometheus-2.45.0.linux-amd64/consoles/node-overview.html  
prometheus-2.45.0.linux-amd64/consoles/node-cpu.html  
prometheus-2.45.0.linux-amd64/consoles/index.html.example  
prometheus-2.45.0.linux-amd64/consoles/node.html  
prometheus-2.45.0.linux-amd64/consoles/node-disk.html  
prometheus-2.45.0.linux-amd64/consoles/prometheus-overview.html  
prometheus-2.45.0.linux-amd64/consoles/prometheus.html  
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```

Edit /etc/prometheus/prometheus.yml and add the following scrape config under scrape_configs:

```
- job_name: 'node'  
  static_configs:  
    - targets: ['<prod1-public-ip>:9100']  
    - targets: ['<prod2-public-ip>:9100']
```

You can edit the file using:
sudo nano /etc/prometheus/prometheus.yml

Then save and exit.



The screenshot shows a MobaXterm window with multiple tabs open. The current tab displays a Prometheus configuration file named 'my_global_config.yml'. The file contains various sections like 'global', 'Alertmanager configuration', 'scrape_configs', and 'static_configs'. It specifies scrape intervals, alertmanager endpoints, and targets for Prometheus itself and external nodes. The terminal interface includes a sidebar with session management, a status bar at the bottom, and a system tray at the bottom right.

```
# my_global_config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # 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"]

    - job_name: 'node'
      static_configs:
        - targets: ['18.234.227.196:9100']
        - targets: [3.95.132.249:9100]

  ~
  ~
  ~

  34,39   All
  27°C  Mostly cloudy
  Search
  ENG IN 22:14 30-05-2025
```

Create Prometheus systemd service

Create the file /etc/systemd/system/prometheus.service:

sudo nano /etc/systemd/system/prometheus.service

Paste the following content:

[Unit]

Description=Prometheus Monitoring

Wants=network-online.target

After=network-online.target

[Service]

User=prometheus

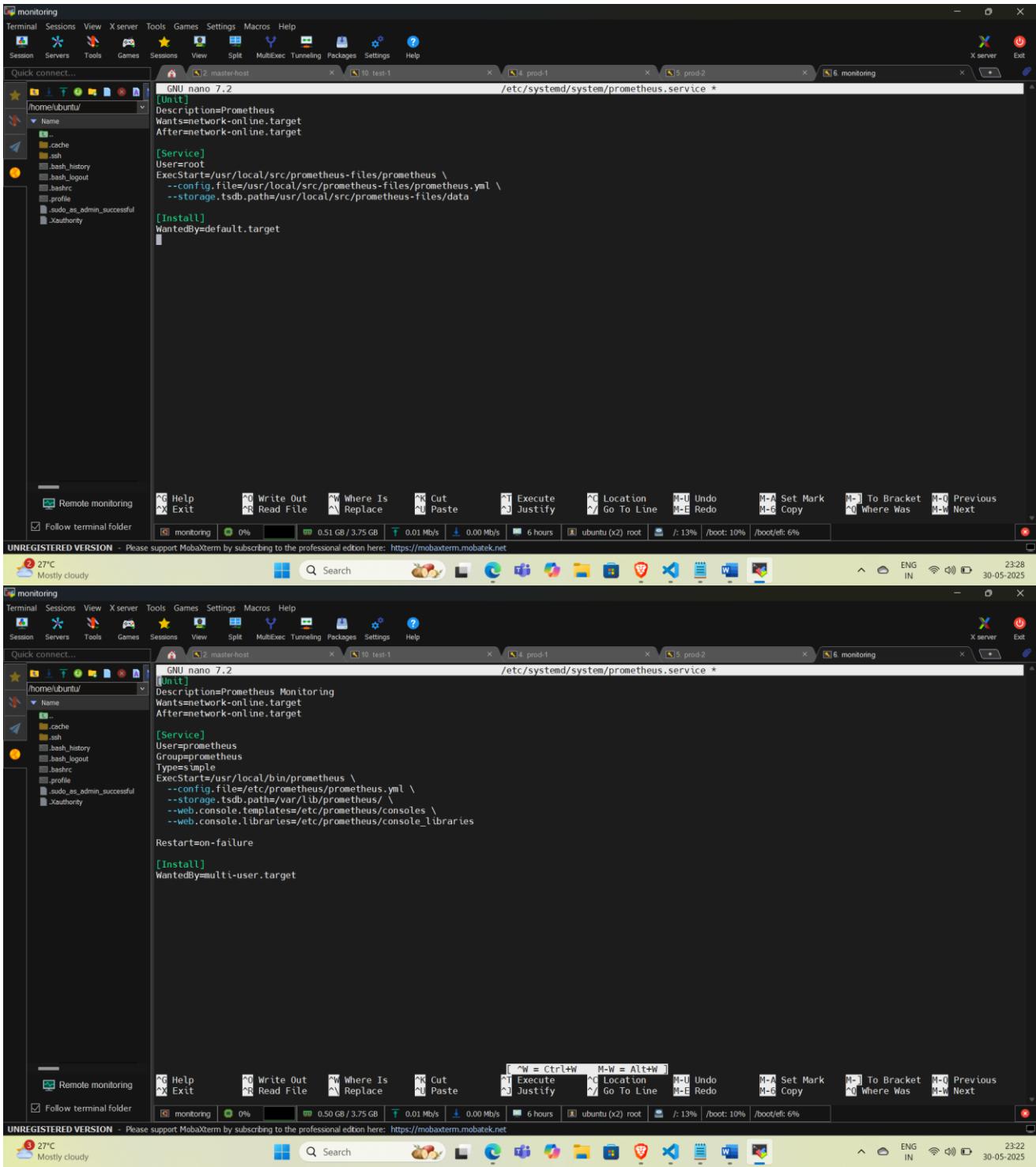
Group=prometheus

Type=simple

ExecStart=/usr/local/bin/prometheus \
--config.file=/etc/prometheus/prometheus.yml \
--storage.tsdb.path=/var/lib/prometheus/ \
--web.console.templates=/etc/prometheus/consoles \
--web.console.libraries=/etc/prometheus/console_libraries

Restart=on-failure

[Install] WantedBy=multi-user.target



```
monitoring
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
[2 master-host] /etc/systemd/system/prometheus.service *
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
User=root
ExecStart=/usr/local/src/prometheus-files/prometheus \
--config.file=/usr/local/src/prometheus-files/prometheus.yml \
--storage.tsdb.path=/usr/local/src/prometheus-files/data

[Install]
WantedBy=default.target

monitoring 0% 0.51 GB / 3.75 GB 0.01 Mb/s 0.00 Mb/s 6 hours ubuntu (x2) root /: 13% /boot: 10% /boot/efi: 6%
Help Write Out Where Is Cut Paste Execute Location Undo Set Mark To Bracket Where Was Previous
Exit Read File Replace Justify Go To Line Redo Copy Next
27°C Mostly cloudy Search 23:28 ENG IN 30-05-2025

monitoring
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
[2 master-host] /etc/systemd/system/prometheus.service *
[Unit]
Description=Prometheus Monitoring
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=impmle
ExecStart=/usr/local/bin/prometheus \
--config.file=/etc/prometheus/prometheus.yml \
--storage.tsdb.path=/var/lib/prometheus/ \
--web.console.templates=/etc/prometheus/consoles \
--web.console.libraries=/etc/prometheus/console_libraries

Restart=on-failure

[Install]
WantedBy=multi-user.target

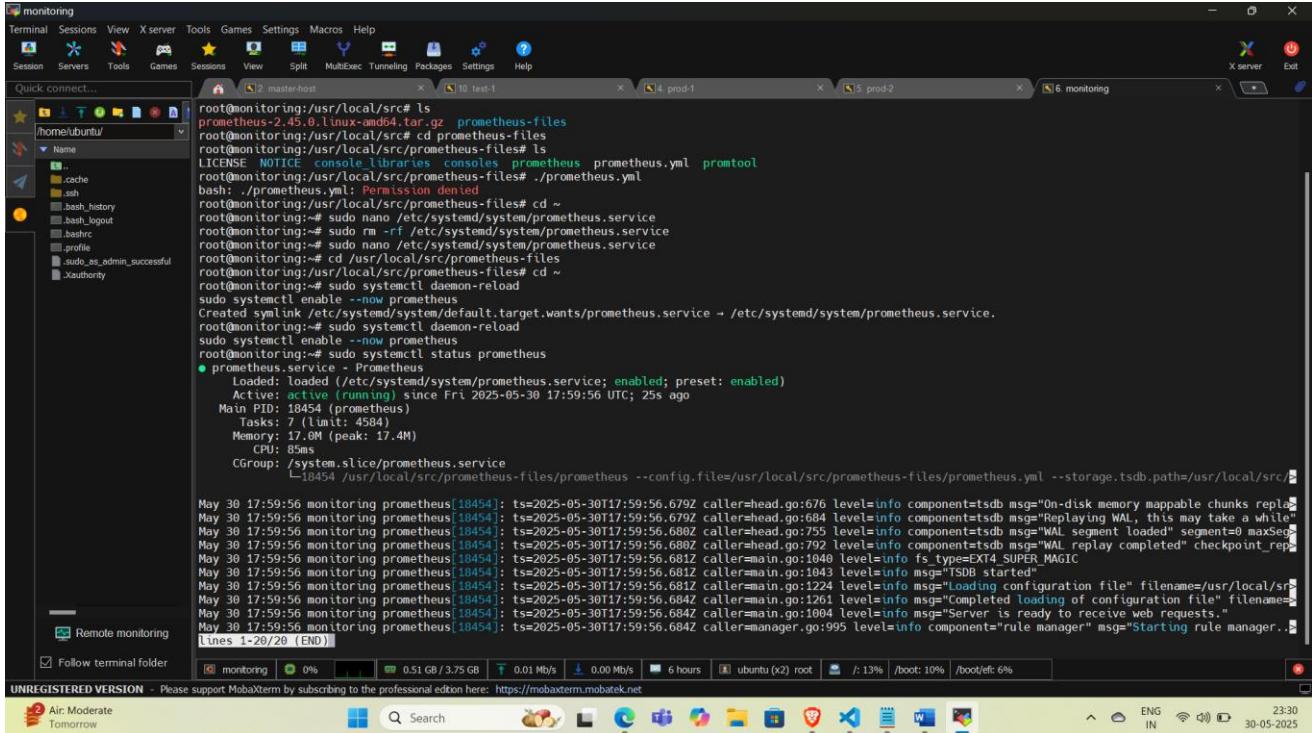
monitoring 0% 0.50 GB / 3.75 GB 0.01 Mb/s 0.00 Mb/s 6 hours ubuntu (x2) root /: 13% /boot: 10% /boot/efi: 6%
Help Write Out Where Is Cut Paste Execute Location Undo Set Mark To Bracket Where Was Previous
Exit Read File Replace Justify Go To Line Redo Copy Next
27°C Mostly cloudy Search 23:22 ENG IN 30-05-2025
```

Save and exit.

Start and enable Prometheus service

sudo systemctl daemon-reload

sudo systemctl enable --now prometheus
 sudo systemctl status Prometheus



```

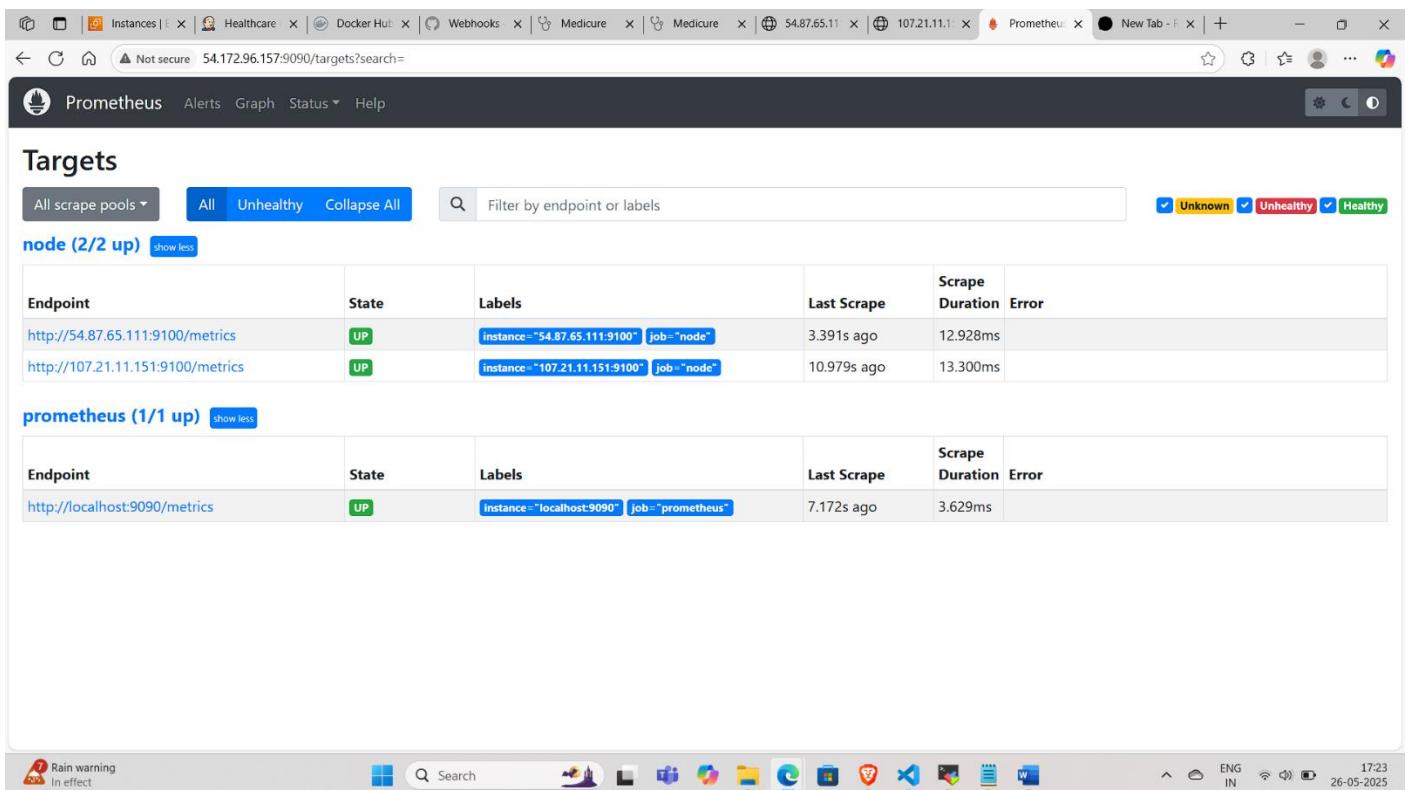
root@monitoring:/usr/local/src# ls
prometheus-2.45.0.linux-amd64.tar.gz  prometheus-files
root@monitoring:/usr/local/src# cd prometheus-files
root@monitoring:/usr/local/src/prometheus# ls
LICENSE NOTICE console_libraries consoles prometheus prometheus.yml promtool
root@monitoring:/usr/local/src/prometheus# ./prometheus.yml
bash: ./prometheus.yml: Permission denied
root@monitoring:/usr/local/src/prometheus# cd ~
root@monitoring:/# sudo nano /etc/systemd/system/prometheus.service
root@monitoring:/# sudo rm -rf /etc/systemd/system/prometheus.service
root@monitoring:/# cd /usr/local/src/prometheus-files
root@monitoring:/# sudo systemctl daemon-reload
sudo systemctl enable --now prometheus
Created symlink /etc/systemd/system/default.target.wants/prometheus.service → /etc/systemd/system/prometheus.service.
root@monitoring:/# sudo systemctl daemon-reload
sudo systemctl enable --now prometheus
root@monitoring:/# sudo systemctl status prometheus
● prometheus.service - Prometheus
  Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; preset: enabled)
  Active: active (running) since Fri 2025-05-30 17:59:56 UTC; 25s ago
    Main PID: 18454 (prometheus)
      Tasks: 7 (limit: 4584)
        Memory: 17.0M (peak: 17.4M)
          CPU: 85ms
        Group: /system.slice/prometheus.service
           ▾ 18454 /usr/local/src/prometheus/files/prometheus --config.file=/usr/local/src/prometheus-files/prometheus.yml --storage.tsdb.path=/usr/local/src/prometheus-files/prometheus

May 30 17:59:56 monitoring prometheus[18454]: ts=2025-05-30T17:59:56.679Z caller=head.go:676 level=info component=tedb msg="On-disk memory mappable chunks replicated"
May 30 17:59:56 monitoring prometheus[18454]: ts=2025-05-30T17:59:56.679Z caller=head.go:684 level=info component=tedb msg="Replaying WAL, this may take a while"
May 30 17:59:56 monitoring prometheus[18454]: ts=2025-05-30T17:59:56.688Z caller=head.go:755 level=info component=tedb msg="WAL segment loaded" segment=0 maxSeq=1
May 30 17:59:56 monitoring prometheus[18454]: ts=2025-05-30T17:59:56.688Z caller=head.go:792 level=info component=tedb msg="WAL replay completed" checkpoint_replay=1
May 30 17:59:56 monitoring prometheus[18454]: ts=2025-05-30T17:59:56.681Z caller=main.go:1040 level=info fs_type=EXT4_SUPER_MAGIC
May 30 17:59:56 monitoring prometheus[18454]: ts=2025-05-30T17:59:56.681Z caller=main.go:1043 level=info msg="TSDB started"
May 30 17:59:56 monitoring prometheus[18454]: ts=2025-05-30T17:59:56.681Z caller=main.go:1224 level=info msg="Loading configuration file" filename=/usr/local/src/prometheus-files/prometheus.yml
May 30 17:59:56 monitoring prometheus[18454]: ts=2025-05-30T17:59:56.684Z caller=main.go:1261 level=info msg="Completed loading of configuration file" filename=/usr/local/src/prometheus-files/prometheus.yml
May 30 17:59:56 monitoring prometheus[18454]: ts=2025-05-30T17:59:56.684Z caller=main.go:1004 level=info msg="Server is ready to receive web requests."
May 30 17:59:56 monitoring prometheus[18454]: ts=2025-05-30T17:59:56.684Z caller=manager.go:995 level=info component="rule manager" msg="Starting rule manager.."
Lines 1-20/20 (END)

[...]
```

Verify Prometheus is running by opening in a browser:

text
<http://<monitor-public-ip>:9090>



| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|---|-------|--|-------------|-----------------|-------|
| http://54.87.65.111:9100/metrics | UP | instance="54.87.65.111:9100" job="node" | 3.391s ago | 12.928ms | |
| http://107.21.11.151:9100/metrics | UP | instance="107.21.11.151:9100" job="node" | 10.979s ago | 13.300ms | |

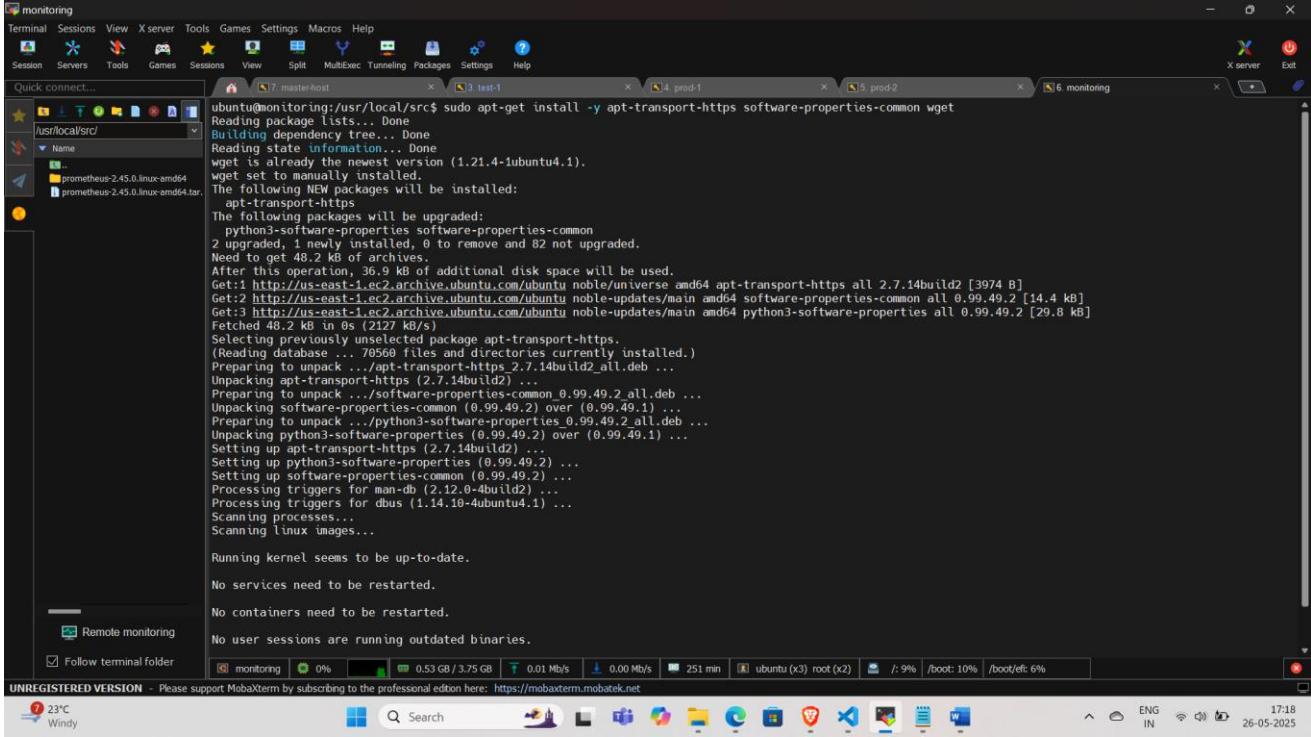
| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|---|-------|--|-------------|-----------------|-------|
| http://localhost:9090/metrics | UP | instance="localhost:9090" job="prometheus" | 7.172s ago | 3.629ms | |

Rain warning in effect

Part 2: Install and Configure Grafana

Step 1: Install required dependencies

```
sudo apt-get install -y apt-transport-https software-properties-common wget
```



The screenshot shows a MobaXterm window with multiple tabs open. The current tab, labeled '7. master-host', contains the command: `ubuntu@monitoring:/usr/local/src$ sudo apt-get install -y apt-transport-https software-properties-common wget`. The terminal output shows the process of installing the required packages, including apt-transport-https, software-properties-common, and wget. The output indicates that wget is already the newest version (1.21.4-1ubuntu4.1) and is set to manually installed. It also lists other packages being upgraded or installed from various Ubuntu repositories. The terminal concludes by stating that the kernel seems to be up-to-date and no services need to be restarted.

Step 2: Add Grafana GPG key and repository

```
sudo wget -q -O /usr/share/keyrings/grafana.key
https://apt.grafana.com/gpg.key
echo "deb [signed-by=/usr/share/keyrings/grafana.key]
https://packages.grafana.com/oss/deb stable main" | sudo tee
/etc/apt/sources.list.d/grafana.list
```

Step 3: Update package list and install Grafana

```
sudo apt-get update
sudo apt-get install -y Grafana
```

```

grafana musl
  upgraded, 2 newly installed, 0 to remove and 82 not upgraded.
Need to get 176 MB of archives.
After this operation, 650 MB of additional disk space will be used.
Get: http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/universe amd64 musl:amd64 1.2.4-2 [416 kB]
Get: https://packages.grafana.com/oss/deb/stable/main amd64 grafana:amd64 12.0.1 [175 MB]
Fetched 176 MB in 25 (71.2 MB/s)
Selecting previously unselected package musl:amd64.
(Reading database ... 70564 files and directories currently installed.)
Preparing to unpack .../grafana_12.0.1_amd64.deb ...
Unpacking grafana (12.0.1) ...
Setting up musl:amd64 (1.2.4-2) ...
Setting up grafana (12.0.1) ...
Setting up grafana (12.0.1) ...
info: Selecting UID from range 100 to 999 ...
info: Adding system user 'grafana' (UID 111) ...
info: Adding new user 'grafana' (UID 111) with group 'grafana' ...
info: Not creating home directory '/usr/share/grafana'.
## NOT starting on installation, please execute the following statements to configure grafana to start automatically using systemd
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable grafana-server
## You can start grafana-server by executing
sudo /bin/systemctl start grafana-server
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@monitoring:/usr/local/src$ 

```

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

22°C Windy

Step 4: Enable and start Grafana service

```

sudo systemctl enable --now grafana-server
sudo systemctl status grafana-server

```

```

ubuntu@monitoring:/usr/local/src$ sudo systemctl enable --now grafana-server
sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-05-26 11:51:50 UTC; 32ms ago
     Docs: http://docs.grafana.org
 Main PID: 13396 ((grafana))
   Tasks: 1 (limit: 4584)
  Memory: 412.0K (peak: 412.0K)
    CPU: 26ms
   CGroup: /system.slice/grafana-server.service
           └─13396 "/usr/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid --packaging=deb cfg:de

May 26 11:51:50 monitoring systemd[1]: Started grafana-server.service - Grafana instance.
Synchronizing state of grafana-server.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-05-26 11:51:50 UTC; 1s ago
     Docs: http://docs.grafana.org
 Main PID: 13396 ((grafana))
   Tasks: 7 (limit: 4584)
  Memory: 45.4M (peak: 46.0M)
    CPU: 99ms
   CGroup: /system.slice/grafana-server.service
           └─13396 "/usr/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid --packaging=deb cfg:de

May 26 11:51:51 monitoring grafana[13396]: logger=igrator t=2025-05-26T11:51:51.915164477Z level=info msg="Executing migration" id="drop index IDX_temp_user"
May 26 11:51:51 monitoring grafana[13396]: logger=igrator t=2025-05-26T11:51:51.916132303Z level=info msg="Migration successfully executed" id="drop index IDX_temp_user"
May 26 11:51:51 monitoring grafana[13396]: logger=igrator t=2025-05-26T11:51:51.923010011Z level=info msg="Executing migration" id="drop index IDX_temp_user"
May 26 11:51:51 monitoring grafana[13396]: logger=igrator t=2025-05-26T11:51:51.924002836Z level=info msg="Migration successfully executed" id="drop index IDX_temp_user"
May 26 11:51:51 monitoring grafana[13396]: logger=igrator t=2025-05-26T11:51:51.934057928Z level=info msg="Executing migration" id="Rename table temp_user to v2"
May 26 11:51:51 monitoring grafana[13396]: logger=igrator t=2025-05-26T11:51:51.944548172Z level=info msg="Migration successfully executed" id="Rename table temp_user to v2"
May 26 11:51:51 monitoring grafana[13396]: logger=igrator t=2025-05-26T11:51:51.959257162Z level=info msg="Creating temp_user v2"
May 26 11:51:51 monitoring grafana[13396]: logger=igrator t=2025-05-26T11:51:51.960423429Z level=info msg="Migration successfully executed" id="create temp_user"

```

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

Rain warning
In effect

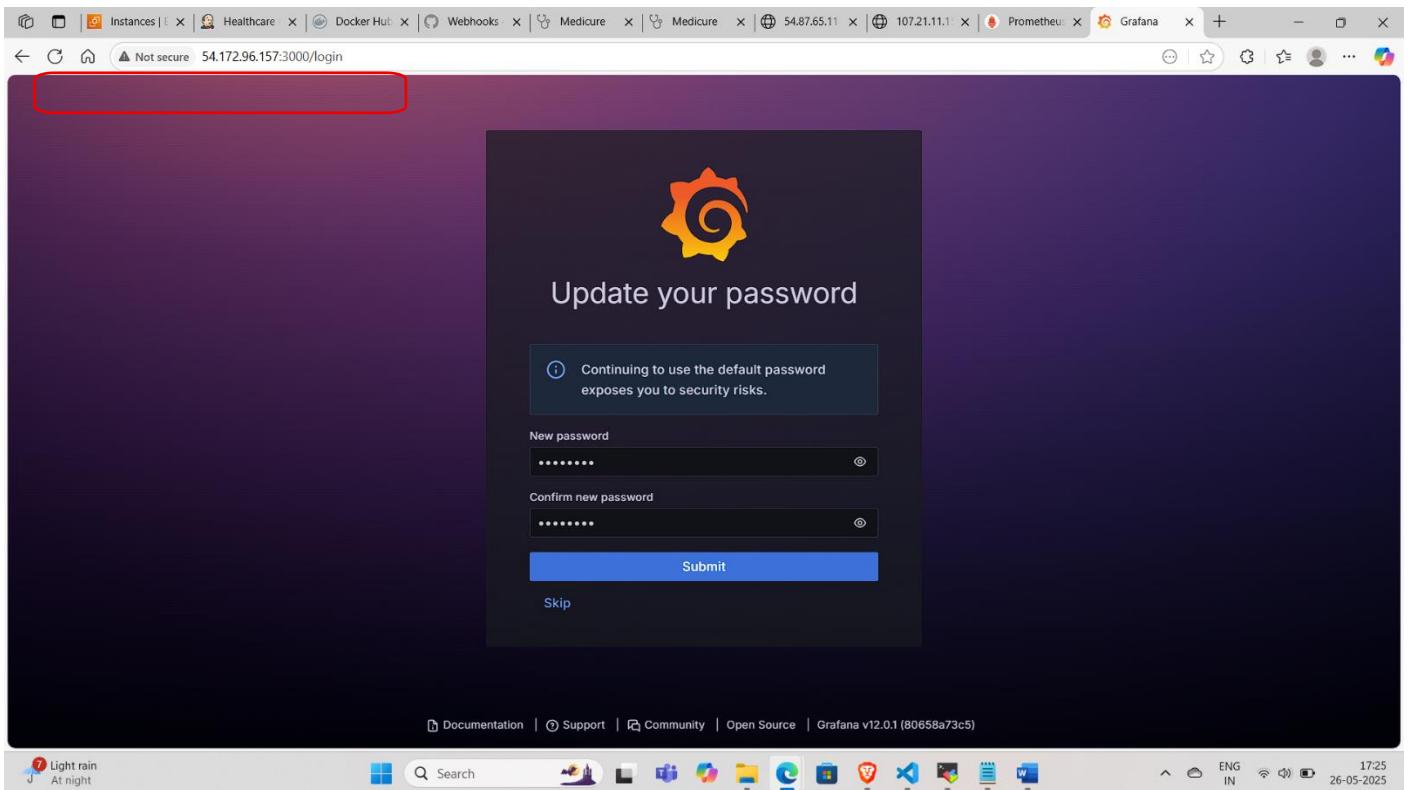
Step 5: Access Grafana web UI

Open your browser and go to:

<http://localhost:3000>

Default login credentials:

- Username: admin
- Password: admin

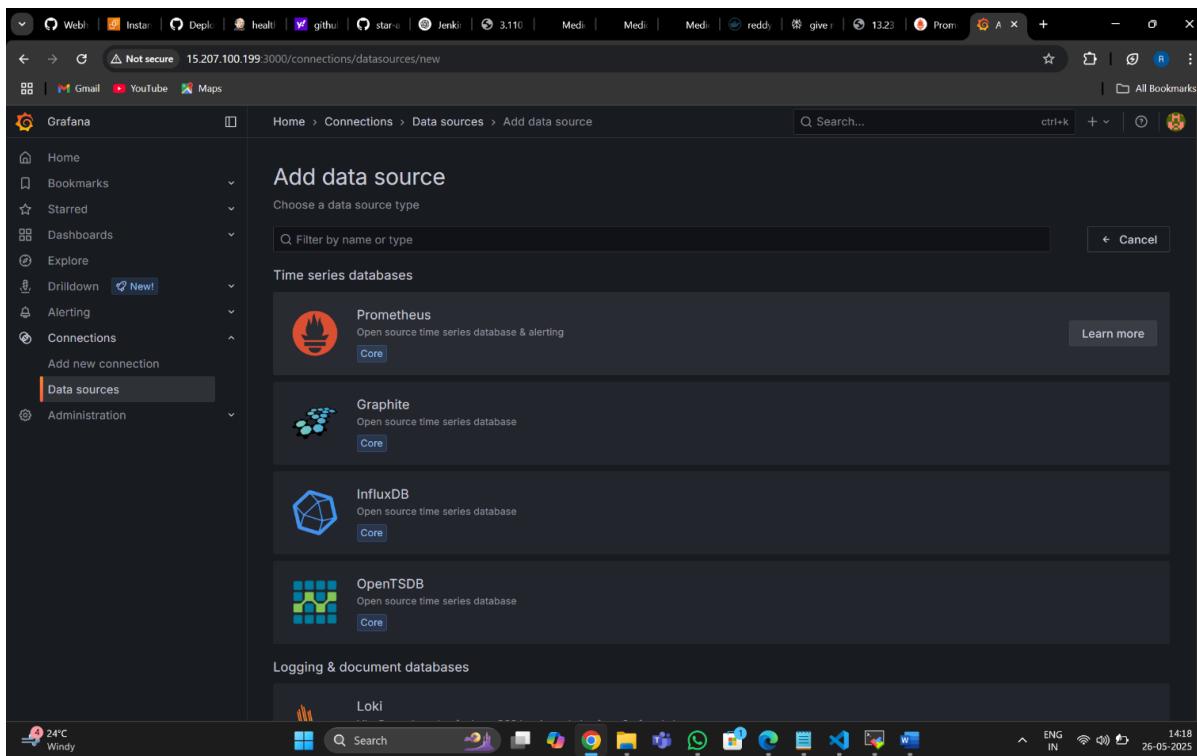


You will be prompted to change the password on first login.

Part 3: Configure Grafana to use Prometheus as a Data Source

1. Log in to Grafana UI.
2. Go to **Configuration** (gear icon) > **Data Sources** > **Add data source**.
3. Select **Prometheus**.
4. Set the URL to:

<http://<monitor-public-ip>:9090>



5. Click **Save & Test**. You should see **Data source is working**.

To monitor **CPU utilization**, **Memory utilization**, and **Disk utilization** in Grafana using Prometheus as the data source (with Node Exporter metrics), you can use the following PromQL queries and panel configurations.

1. CPU Utilization PromQL Query (Percentage CPU Usage)

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

- This calculates the percentage of CPU time spent **not idle** (i.e., CPU utilization) averaged over 5 minutes per instance.

2. Memory Utilization PromQL Query (Percentage Memory Usage)

**100 * (1 - ((node_memory_MemAvailable_bytes) /
(node_memory_MemTotal_bytes)))**

- This calculates the percentage of memory used by subtracting available memory from total memory.

3. Disk Utilization

PromQL Query (Percentage Disk Space Used)

```
100 * (1 - (node_filesystem_avail_bytes{fstype=~"ext4|xfs"} / node_filesystem_size_bytes{fstype=~"ext4|xfs"}))
```

- This calculates disk usage percentage for filesystems of type ext4 or xfs (common Linux filesystems).
- You can adjust or remove the fstype filter based on your environment.

