

**PROJECT**

**HEALTHCARE DOMAIN**

**SUBMITTED BY – BALASASTHA P**

**SUBMITTED TO – STARAGILE**

**DATE OF SUBMISSION – 19-01-2025**

## **HEALTHCARE DOMAIN**

Go to your repository and create a new repository in the name of “***project-3***” → Now download the project into zip file using the following link

“<https://github.com/StarAgileDevOpsTraining/star-agile-health-care.git>” → Now extract the downloaded folder and paste the folder in the desktop → Folder name will be “***star-agile-health-care***” move the folder to desktop and rename it as “***Project-3***” → Now install the git bash → Now go inside the folder and right click and open git bash terminal → Now give the command “***git init***” to initialize the git in the current folder → Now add remote github using the command “***git remote add origin*** [\*\*\*https://github.com/Balasastha/Project-3\*\*\*](https://github.com/Balasastha/Project-3)” → Create a PAT personal access token in the github → Go to github profile → Settings → Developer settings → Personal Access tokens and generate the token → First add all the files using the command “***git add .***” and make commit using the command “***git commit -m "Initial commit"***” Now push the file from git to github using the command “***git push origin -u master***” → It asks for PAT just copy and paste it from github and sign in → This will push file to

github → Now create 2 EC2 ubuntu instance with all traffic enabled using terraform → First download terraform from the “***terraform.io***” website and create a folder in the desktop named as “***terraform-scripts***” now go to the downloaded folder and extract it and copy the application file and paste it in the “***terraform-scripts***” folder → Now download Visual Studio Code → Open the folder that we have created in desktop → Inside the folder create a file names as “***main.tf***” and type the following content inside it,

```
provider "aws" {  
    region = "ap-northeast-1"  
}
```

```
resource "aws_security_group" "allow_all" {  
    name      = "allow_all"  
    description = "Allow all inbound and outbound traffic"
```

```
    ingress {  
        from_port  = 0  
        to_port    = 65535  
        protocol   = "tcp"
```

```

  cidr_blocks = ["0.0.0.0/0"]

}

egress {
  from_port   = 0
  to_port     = 65535
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]

}

resource "aws_instance" "master_server" {
  ami           = "ami-0a290015b99140cd1" # Ubuntu AMI ID
  instance_type = "t2.medium"             # EC2 instance type
  security_groups = [aws_security_group.allow_all.name]
  key_name       = "Project-1"            # SSH key name

  root_block_device {
    volume_size = 15 # 15 GB storage
  }
}

```

```
volume_type = "gp2" # General Purpose SSD
}

tags = {
    Name = "Master-server"
}
}

resource "aws_instance" "k8s_worker" {
    ami = "ami-0a290015b99140cd1" # Ubuntu AMI ID
    instance_type = "t2.medium"
    security_groups = [aws_security_group.allow_all.name]
    key_name = "Project-1"

    root_block_device {
        volume_size = 15
        volume_type = "gp2"
    }
}
```

```
tags = {  
    Name = "k8s-worker"  
}  
}  
  
resource "aws_instance" "test_server" {  
    ami           = "ami-0a290015b99140cd1" # Ubuntu AMI ID  
    instance_type = "t2.medium"  
    security_groups = [aws_security_group.allow_all.name]  
    key_name       = "Project-1"  
  
    root_block_device {  
        volume_size = 15  
        volume_type = "gp2"  
    }  
}  
  
tags = {  
    Name = "Test-server"  
}
```

}

Now to access the aws account we have open the terminal in VS Code using the cmd “**ctrl + `**” and type “**aws configure**” there it will ask for access key and secret access key which will be available under profile section → Security credentials in aws account → There you can create keys and copy and paste it in the VS terminal → Now type the command “**terraform init**” to initialize and type the command “**terraform plan**” and type the command “**terraform apply**” and enter the value as “**yes**” this will create the instance accordingly → Now go to the aws account and check for the instance the instances will be created in which one instance is “**Master-server**” and another one is “**Test-server**” → Now connect the master-server machine → Go inside the master machine and type the following command “**sudo su**” to gain root privilege access and update all packages using the command “**apt update**” → Now install java using the following command “**sudo apt install openjdk-11-jdk -y**” → Now install Jenkins in it using the following commands,

“**wget <https://raw.githubusercontent.com/akshu20791/Deployment-script/main/jenkins.sh>**”

“**chmod +x jenkins.sh**”

“**./jenkins.sh**”

Now copy the public ip of the machine and paste it in the browser like “**<public-ip>:8080**” → It will asks for administrator password which is

in the machine copy and paste it in the Jenkins → Install suggested plugins → Install docker using the command “***sudo apt install docker.io -y***” → To add the jenkins user to the docker group use the command “***sudo usermod -aG docker jenkins***” → Now give root user access to jenkins for go to sudoers file by using the cmd “***visudo***” → Under user privilege specification add “***jenkins ALL=(ALL:ALL) NOPASSWD: ALL***” → Press “***ctrl + x***” and “***y***” to save and exit → Restart the jenkins using command “***service jenkins restart***” → Go to jenkins and refresh it and login again → Create new item → Give the project name and choose pipeline → Go to pipeline syntax to add docker credentials → Choose sample step as “***withCredentials: Bind Credentials to variables***” → Under bindings click “***Add***” → Secret text → Add → Jenkins → Under kind choose “***secret text***” → In secret file type the password of your docker account → In ID type as “***dockerhubpass***” → Under description type “***To hide docker hub password***” → Click on “***Generate Pipeline Script***” → Just copy the pipeline that is generated which will be used in jenkins pipeline to login into docker → Go to the master machine and install ansible using following commands,

“***sudo apt install software-properties-common***”

“***sudo add-apt-repository --yes --update ppa:ansible/ansible***”

“***sudo apt install ansible***”

Go to jenkins → Manage jenkins → Plugins → Available plugins → Search for “***docker***” plugin and install it and “***ansible***” plugin and install it → Go to manage jenkins → Tools → Under ansible installation “***Add ansible***” → Name it as “***ansible***” → Check the box “***Install automatically***” → Apply and save → Now go to master machine and type the command “***vi /etc/ansible/hosts***” → In that file type the following,

“***[ansiblegroup]***

***Private IP of test server***”

Now configure the ansible file in the local repo and push it into remote repo using the command “***vi ansible-playbook.yml***” → Inside type the following content,

“**- name :** Configure Docker on EC2 Instances

**hosts :** all

**become: true**

**connection :** ssh

**tasks :**

**- name:** updating apt

**command :** sudo apt-get update

**- name :** Install Docker

**command :** sudo apt-get install -y docker.io

**- name :** Start Docker Service

**command :** sudo systemctl start docker

**- name:** Deploy Docker Container

**command:** docker run -itd -p 8084:8081 balasastha/healthcare-domain:3”

Save and exit the file now add the file using the command “**git add ansible-playbook.yml**” → Now commit the changes using command “**git commit -m “Second commit”**” → Now push the command to the github repo “**git push origin -u master**” → Now go to jenkins → Go to the project → Pipeline → Pipeline syntax → Under Sample step choose “**ansiblePlaybook: Invoke an ansible playbook**” → Under ansible tool choose “**ansible**” → Specify playbook file path in workspace as “**ansible-playbook.yml**” → Specify inventory file path in workspace as “**/etc/ansible/hosts**” → Under SSH connection credentials → Click “**Add**” → Jenkins → Under kind choose “**SSH username with private key**” → Specify id as “**ansible**” → Specify description “**Private key file for test server connection**” → Specify username as “**ubuntu**” → Check

“***Enter directly***” option and paste the private key file → Under SSH connection credentials select “***ubuntu (Private key for test server connection)***” → Check the option ”***Use become***” → Check “***Disable the host SSH key check***” → Click on “***Generate pipeline***” → Just copy the pipeline that is generated which will be used in jenkins pipeline to establish connection to the test server → Now we will install Kubernetes in “***master-server***” and “***k8s-worker***” →

### **In Master machine**

“***wget https://raw.githubusercontent.com/akshu20791/Deployment-script/main/k8s-master.sh***”

“***chmod 777 k8s-master.sh***”

“***./k8s-master.sh***”

### **In k8s-worker Machine**

“***wget https://raw.githubusercontent.com/akshu20791/Deployment-script/main/k8s-nodes.sh***”

“***chmod 777 k8s-nodes.sh***”

“***./k8s-nodes.sh***”

“***modprobe br\_netfilter***”

**“echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables”**

**“echo 1 > /proc/sys/net/ipv4/ip\_forward”**

After k8s installation is done. We need to connect nodes with the master via tokens type the command “**kubeadm token create --print-join-command**” → Whatever token comes up ...copy the token and paste it in notepad and after that, in the command you will see 6443 written after that paste “**--cri-socket unix:///var/run/cri-dockerd.sock**” → Now paste the tokens in the “**k8s-worker**” → Go to the master server and type the command “**kubectl get nodes**” → Now create a YAML file in the local repo and push it to the remote repo using the command “**vi deployment.yaml**” → Paste the following content inside the file,

**“apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: healthcare**

**spec:**

**replicas: 1**

**selector:**

**matchLabels:**

*app: healthcare*

*template:*

*metadata:*

*labels:*

*app: healthcare*

*spec:*

*containers:*

*- name: healthcare*

*image: balasastha/healthcare-domain:3*

*ports:*

*- containerPort: 8086*

---

*apiVersion: v1*

*kind: Service*

*metadata:*

*name: healthcare*

*spec:*

*selector:*

```
app: healthcare
```

```
type: NodePort
```

```
ports:
```

```
- port: 80
```

```
targetPort: 8086
```

```
nodePort: 30007”
```

Save and quit the file → Add file using the command “**git add deployment.yaml**” → Now commit the changes using the command “**git commit -m “Third commit”**” → Now push the Jenkins file to the github repo using the command “**git push origin -u master**” → Now create a jenkins file in the local repo and push it into remote repo using the command “**vi Jenkinsfile**” → Paste the following content inside the file,

```
pipeline {
```

```
    agent any
```

```
    stages{
```

```
        stage('Git checkout'){
```

```
            steps
```

```
{
```

```
            echo 'Checking out the git repo...'
```

```
            git url:' https://github.com/Balasastha/Project-3.git'
```

```
}  
}  
stage('Package the code') {  
    steps  
    {  
        echo 'Packaging the code for you...'  
        sh 'mvn clean package'  
    }  
}  
stage ('Build docker image') {  
    steps  
    {  
        echo 'Building docker image for you..'  
        sh 'docker build -t balasastha/healthcare-domain:3 .'  
    }  
}  
stage ('Docker login') {  
    steps  
    {
```

```
echo 'Logging in into docker...'
withCredentials([string(credentialsId: 'dockerhubpass',
variable: 'dockerhubpass')]) {

    sh 'docker login -u balasastha -p ${dockerhubpass}'
}
}

stage ('Push image into docker') {
    steps
{
        echo 'Pushing the docker image for you'
        sh 'docker push balasastha/healthcare-domain:3'
}
}

stage ('Invoking ansible playbook') {
    steps
{
        echo 'Running ansible playbook in test server...'

```

```

    ansiblePlaybook become: true, credentialsId: 'ansible',
disableHostKeyChecking: true, installation: 'ansible', inventory:
'/etc/ansible/hosts', playbook: 'ansible-playbook.yml', vaultTmpPath:
"
}

}

stage ('Deployment of project on production server'){

steps

{
    echo 'Deploying using kubernetes..'

    sh 'sudo kubectl apply -f deployment.yaml'

    sh 'sudo kubectl get all'

}

}

}

}

```

Save and quit the file → Add file using the command “***git add Jenkinsfile***” → Now commit the changes using the command “***git commit -m “Fourth commit”***” → Now push the Jenkins file to the

github repo using the command “***git push origin -u master***” → Go to the jenkins and go to the pipeline of the created project → Under definition choose “***Pipeline script from SCM***” → In SCM choose “***Git***” → In repository URL give the link of the repo that we are using “[\*\*\*https://github.com/Balasastha/Project-3\*\*\*](https://github.com/Balasastha/Project-3)” → Now build the project by clicking the option “***Build now***” → The project will be build → Now whenever the developer pusher the code to github the jenkins job should be triggered automatically for that go the github repository of the project → Go to settings → Webhooks → Add Webhooks → In the “***Payload URL***” field, provide the URL of your Jenkins server followed by /github-webhook/. For example, [\*\*\*http://your-jenkins-server/github-webhook/\*\*\*](http://your-jenkins-server/github-webhook/) → Set the content type to “***application/json***” → Under “***Which events would you like to trigger this webhook?***”, select ***Just the push event*** → Save the webhook → Go to jenkins → Go to configure → Check “***Github Project***” and enter the github URL there and check “***GitHub hook trigger for GITScm polling***” → Now create a file in local repo and push it to the remote repo the job will be started automatically executed in the jenkins without any manual trigger → Now create a grafana machine with ubuntu ami to monitor “***test-server***” and “***k8s-***

**worker” →** For that you have to install node exporter in both k8s-worker and test server → To install node exporter connect the 2 machine k8s-worker and test server and create a .sh file using the command “**vi install\_node\_exporter.sh**” and type the following command,

“**#!/bin/bash**

*# Exit the script if any command fails*

*set -e*

*echo "Starting Node Exporter installation..."*

*# Step 1: Download Node Exporter*

*echo "Downloading Node Exporter..."*

*wget*

*https://github.com/prometheus/node\_exporter/releases/download/v1.6.1/node\_exporter-1.6.1.linux-amd64.tar.gz -O*

*/tmp/node\_exporter.tar.gz*

*# Step 2: Extract the archive*

*echo "Extracting Node Exporter..."*

*tar -xvzf /tmp/node\_exporter.tar.gz -C /tmp/*

*# Step 3: Move the binary*

*echo "Moving Node Exporter binary to /usr/local/bin..."*

*sudo mv /tmp/node\_exporter-1.6.1.linux-amd64/node\_exporter  
/usr/local/bin/*

*# Step 4: Create a systemd service file*

*echo "Creating systemd service file for Node Exporter..."*

*sudo tee /etc/systemd/system/node\_exporter.service > /dev/null  
<<EOL*

*[Unit]*

*Description=Node Exporter*

*After=network.target*

**[Service]**

**User=prometheus**

**ExecStart=/usr/local/bin/node\_exporter**

**[Install]**

**WantedBy=multi-user.target**

**EOL**

**# Step 5: Create a user for Node Exporter**

**echo "Creating a dedicated user for Node Exporter..."**

**sudo useradd --no-create-home --shell /bin/false prometheus**

**# Step 6: Reload systemd, start, and enable Node Exporter**

**echo "Starting and enabling Node Exporter service..."**

**sudo systemctl daemon-reload**

**sudo systemctl start node\_exporter**

```
sudo systemctl enable node_exporter
```

```
echo "Node Exporter installation and configuration completed  
successfully!""
```

Now save and quit the file → Now give the file permission to execute the by using the command “**chmod +x**

***install\_node\_exporter.sh***” → Now execute the file with the command “***./install\_node\_exporter.sh***” → Node Exporter metrics should now be available at

“***http://<server\_ip>:9100/metrics***” → Now we will setup

Prometheus in grafana server for that connect the instance and create a .sh file using the command “**vi** ***setup\_prometheus\_and\_node\_exporter.sh***” and type the following content inside it,

```
#!/bin/bash
```

```
# Exit script on any error
```

```
set -e
```

```
echo "Starting setup for Prometheus and Node Exporter..."
```

```
# --- Install Prometheus ---
```

```
echo "Downloading Prometheus..."
```

```
wget
```

```
https://github.com/prometheus/prometheus/releases/download/v2.47.0/prometheus-2.47.0.linux-amd64.tar.gz -O /tmp/prometheus.tar.gz
```

```
echo "Extracting Prometheus..."
```

```
tar xvzf /tmp/prometheus.tar.gz -C /tmp/
```

```
echo "Moving Prometheus binaries to /usr/local/bin..."
```

```
sudo mv /tmp/prometheus-2.47.0.linux-amd64/prometheus /usr/local/bin/
```

```
sudo mv /tmp/prometheus-2.47.0.linux-amd64/promtool /usr/local/bin/
```

*echo "Moving Prometheus configuration and data  
directories..."*

*sudo mkdir -p /etc/prometheus /var/lib/prometheus*

*sudo mv /tmp/prometheus-2.47.0.linux-amd64/consoles  
/etc/prometheus/*

*sudo mv /tmp/prometheus-2.47.0.linux-  
amd64/console\_libraries /etc/prometheus/*

*# --- Setup Prometheus configuration ---*

*echo "Creating Prometheus configuration file..."*

*sudo tee /etc/prometheus/prometheus.yml > /dev/null <<EOL*

*global:*

*scrape\_interval: 15s*

*scrape\_configs:*

*- job\_name: 'node\_exporter'*

*static\_configs:*

*- targets: ['18.183.176.119:9100', '54.95.66.217:9100']*

**EOL**

# --- *Create Prometheus systemd service* ---

*echo "Creating Prometheus systemd service..."*

*sudo tee /etc/systemd/system/prometheus.service > /dev/null*

*<<EOL*

*[Unit]*

*Description=Prometheus*

*Wants=network-online.target*

*After=network-online.target*

*[Service]*

*User=prometheus*

*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*

**[Install]**

**WantedBy=multi-user.target**

**EOL**

***echo "Creating Prometheus user and setting permissions..."***

***sudo useradd --no-create-home --shell /bin/false prometheus***

***sudo chown -R prometheus:prometheus /etc/prometheus  
/var/lib/prometheus***

**# Start Prometheus**

***echo "Starting Prometheus service..."***

***sudo systemctl daemon-reload***

***sudo systemctl enable prometheus***

***sudo systemctl start prometheus***

**# --- Install Node Exporter on Local Machine ---**

```
echo "Installing Node Exporter on this machine..."
wget
https://github.com/prometheus/node\_exporter/releases/download/v1.6.1/node\_exporter-1.6.1.linux-amd64.tar.gz -O
/tmp/node_exporter.tar.gz
tar -xvzf /tmp/node_exporter.tar.gz -C /tmp/
sudo mv /tmp/node_exporter-1.6.1.linux-amd64/node_exporter
/usr/local/bin/
```

*echo "Creating Node Exporter systemd service..."*

```
sudo tee /etc/systemd/system/node_exporter.service > /dev/null
<<EOL
```

*[Unit]*

*Description=Node Exporter*

*After=network.target*

*[Service]*

*User=prometheus*

*ExecStart=/usr/local/bin/node\_exporter*

*[Install]*

*WantedBy=multi-user.target*

*EOL*

*echo "Starting Node Exporter service..."*

*sudo systemctl daemon-reload*

*sudo systemctl enable node\_exporter*

*sudo systemctl start node\_exporter*

*# --- Completion ---*

*echo "Setup completed! Prometheus is running and Node Exporter is installed."*"

Now give execute permissions to the file using command

*“chmod +x setup\_prometheus\_and\_node\_exporter.sh” →*

Execute the file using the command

*“./setup\_prometheus\_and\_node\_exporter.sh” → Now the*

Prometheus is accessible at `http://<grafana-server-ip>:9090` →  
Go to Status → Target → To see the status of the targets → Now  
we will setup grafana for that create a .sh file using command “`vi  
install_grafana.sh`” and type the following content,

“`#!/bin/bash`

*# Update the package list*

*sudo apt-get update -y*

*# Install required software properties package*

*sudo apt-get install -y software-properties-common*

*# Add the Grafana repository*

*sudo add-apt-repository "deb [arch=amd64]*

*<https://packages.grafana.com/oss/deb> stable main"*

*# Add Grafana GPG key*

```
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
```

*# Update the package list after adding the Grafana repository*

```
sudo apt-get update -y
```

*# Install Grafana*

```
sudo apt-get install -y grafana
```

*# Start Grafana service*

```
sudo systemctl start grafana-server
```

*# Enable Grafana to start on boot*

```
sudo systemctl enable grafana-server
```

*# Print success message*

`echo "Grafana has been installed and started successfully." >`

Now give execute permissions to the file using command

`"chmod +x install_grafana.sh"` → Execute the file using the command `“./install_grafana.sh”` → Now the Grafana is accessible at `“http://<grafana-server-ip>:3000”`

Go to connections → Data sources → Prometheus → Specify Prometheus server URL as `“http://3.112.195.232:9090”` → Save and test →

### **Monitoring CPU Utilization**

Go to home and click on `“Create your first dashboard”` → Click on `“Add visualization”` → Select metric as `“process_cpu_seconds_total”` → In the label filters choose `“instance”` and choose `“master-server IP”` → Select add query and choose metric as `“process_cpu_seconds_total”` → In the label filters choose `“instance”` and choose `“test-server IP”` → Choose the graph type that you want → Specify pannel title as `“CPU Utilization”` → Add fieldoverride property to change the display name → Save dashboard → Specify the Dashboard title as `“Monitoring”` → Save

## Disk Space Utilization

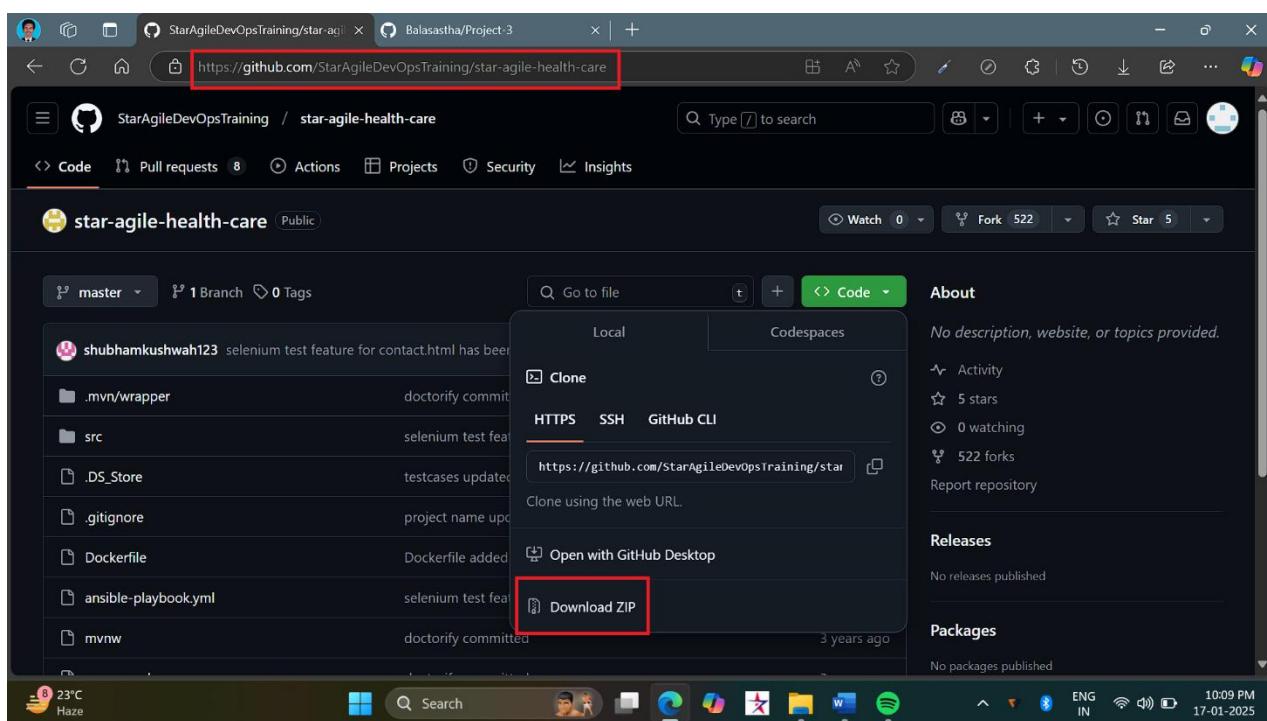
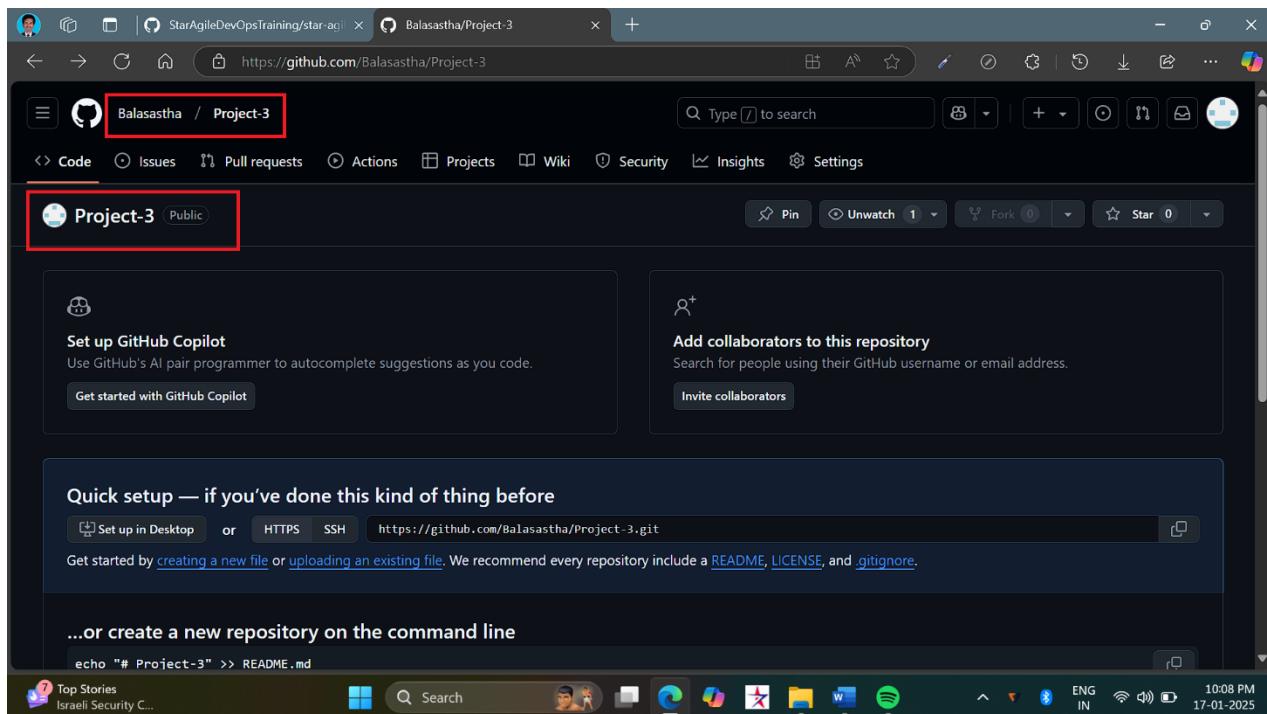
Click on “*Add*” → Choose “*Visualization*” → Type metric as  
“*100 - (node\_filesystem\_avail\_bytes{mountpoint="/" \* 100 / node\_filesystem\_size\_bytes{mountpoint="/"})*” → It calculate the disk space utilization for both instances → Specify panel title as “*Disk Space Utilization*” → Add threshold value as

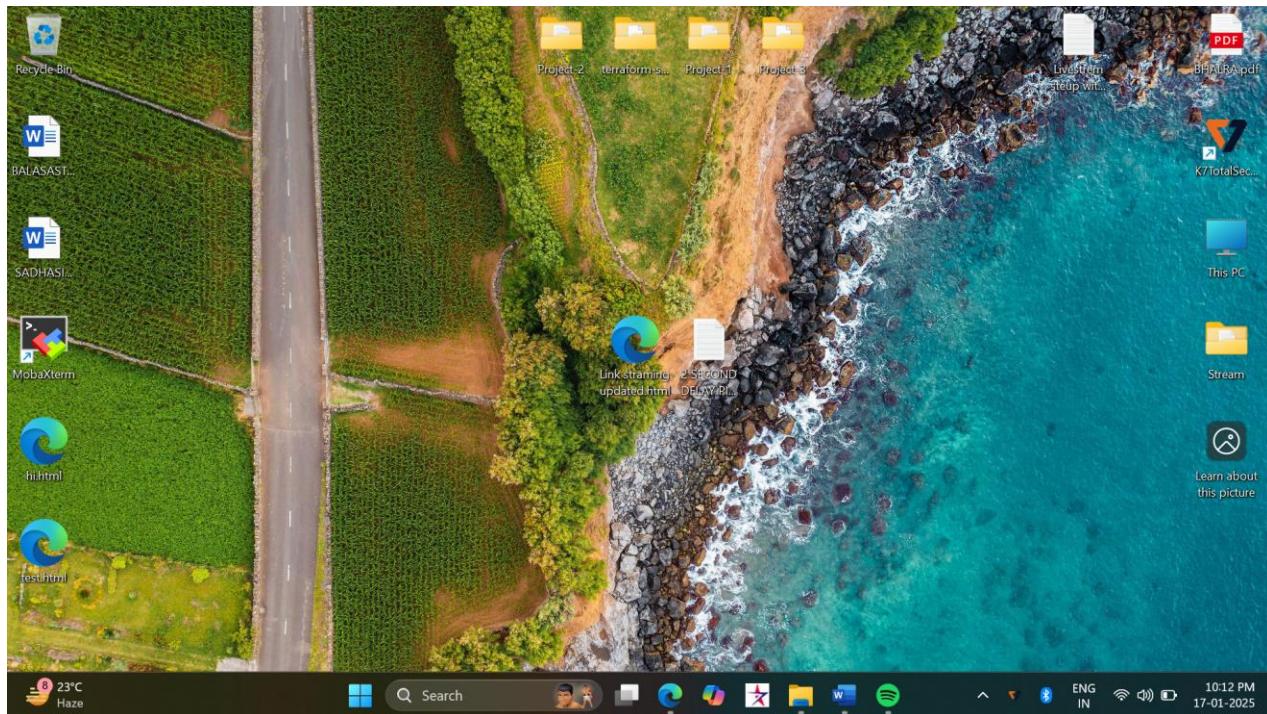


**Green** (0-80%): Normal usage.

**Yellow** (81-90%): Warning.

**Red** (91-100%): Critical.





```
MINGW64:/c/Users/Lenovo/Desktop/Project-3
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3
$ git init
Initialized empty Git repository in C:/Users/Lenovo/Desktop/Project-3/.git/
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git remote add origin https://github.com/Balasastha/Project-3
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git add .
warning: LF will be replaced by CRLF in .gitignore
warning: LF will be replaced by CRLF in .mvn/wrapper/maven-wrapper.properties
warning: LF will be replaced by CRLF in Dockerfile
warning: LF will be replaced by CRLF in ansible-playbook.yml
warning: LF will be replaced by CRLF in mvnw
warning: LF will be replaced by CRLF in mvnw.cmd
warning: LF will be replaced by CRLF in pom.xml
warning: LF will be replaced by CRLF in project/staragile/Doctor.java
warning: LF will be replaced by CRLF in project/staragile/MedicureApplication.java
warning: LF will be replaced by CRLF in project/staragile/MedicureController.java
warning: LF will be replaced by CRLF in project/staragile/MedicureRepository.java
warning: LF will be replaced by CRLF in project/staragile/MedicureService.java
warning: LF will be replaced by CRLF in project/resources/application.properties
warning: LF will be replaced by CRLF in project/resources/static/css/bootstrap.css
warning: LF will be replaced by CRLF in project/resources/static/css/font-awesome.min.css
warning: LF will be replaced by CRLF in project/resources/static/css/style.css
warning: LF will be replaced by CRLF in project/resources/static/css/style.css.map
warning: LF will be replaced by CRLF in project/resources/static/js/bootstrap.js
warning: LF will be replaced by CRLF in project/test/java/com/project/staragile/MedicureApplicationTests.java
warning: LF will be replaced by CRLF in project/test/java/com/project/staragile/MedicureApplicationTests.java
```

```
MINGW64/c/Users/Lenovo/Desktop/Project-3
warning: in the working copy of 'src/main/java/com/project/staragile/MedicureService.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/resources/application.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/resources/static/css/bootstrap.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/resources/static/css/font-awesome.min.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/resources/static/css/style.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/resources/static/css/style.css.map', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/resources/static/js/bootstrap.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/test/java/com/project/staragile/MedicureApplicationTests.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/test/java/com/project/staragile/TestMedicureService.java', LF will be replaced by CRLF the next time Git touches it
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git commit -m "initial commit"
[master (root-commit) has been modified] initial commit
48 files changed, 18863 insertions(+)
create mode 100644 .gitignore
create mode 100644 .mvn/wrapper/maven-wrapper.jar
create mode 100644 .mvn/wrapper/maven-wrapper.properties
create mode 100644 Dockerfile
create mode 100644 ansible-playbook.yml
create mode 100644 mvnw
create mode 100644 mvnw.cmd
create mode 100644 pom.xml
create mode 100644 src/main/java/com/project/staragile/Doctor.java
create mode 100644 src/main/java/com/project/staragile/MedicureApplication.java
create mode 100644 src/main/java/com/project/staragile/MedicureController.java
create mode 100644 src/main/java/com/project/staragile/MedicureRepository.java
create mode 100644 src/main/java/com/project/staragile/MedicureService.java
create mode 100644 src/main/resources/application.properties
create mode 100644 src/main/resources/static/about.html
create mode 100644 src/main/resources/static/contact.html
create mode 100644 src/main/resources/static/css/bootstrap.css
create mode 100644 src/main/resources/static/css/font-awesome.min.css
create mode 100644 src/main/resources/static/css/responsive.css
create mode 100644 src/main/resources/static/css/style.css
create mode 100644 src/main/resources/static/css/style.css.map
create mode 100644 src/main/resources/static/css/style.scss
create mode 100644 src/main/resources/static/departments.html
create mode 100644 src/main/resources/static/doctors.html
create mode 100644 src/main/resources/static/fonts/fontawesome-webfont.ttf

23°C Haze Search ENG IN 10:16 PM 17-01-2025
```

```
MINGW64/c/Users/Lenovo/Desktop/Project-3
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git push origin -u master
Enumerating objects: 69, done.
Counting objects: 100% (69/69), done.
Delta compression using up to 16 threads
Compressing objects: 100% (60/60), done.
Writing objects: 100% (69/69) 17.78 MiB | 1019.00 KiB/s, done.
Total 69 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3); done.
To https://github.com/Balasastha/Project-3
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ |
```

The screenshot shows the HashiCorp Terraform website. On the left, a sidebar lists 'Operating Systems' including macOS, Windows (selected), Linux, FreeBSD, OpenBSD, and Solaris. The main content area is divided into 'Windows' and 'Linux' sections. The 'Windows' section features a 'Binary download' section with '386' and 'AMD64' options, both labeled 'Version: 1.10.4' with 'Download' buttons. A red box highlights this section. The 'Linux' section includes a 'Package manager' tab for Ubuntu/Debian, CentOS/RHEL, Fedora, Amazon Linux, and Homebrew, followed by a terminal-like interface showing apt-get commands for installing Terraform. The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating the date and time.

The screenshot shows the Visual Studio Code interface with a dark theme. The left sidebar has 'EXPLORER', 'OPEN EDITORS' (with 'Welcome' and 'main.tf' listed), and 'TERRAFORM' (with 'main.tf' and 'terraform.exe' listed, the latter highlighted with a red box). The main editor pane displays a Terraform configuration file ('main.tf') with code for creating AWS instances and security groups. The bottom status bar shows file details like 'Ln 40, Col 36' and 'Spaces 4', along with tabs for 'BLACKBOX Chat', 'Add Logs', 'CyberCoder', 'Improve Code', and 'Share Code Link'. The bottom right shows system status including battery level, signal strength, and the date and time '17-01-2025'.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the top right corner of the main interface, there is a red box highlighting the 'Security credentials' link under the 'Software Images' section. The 'Software Images' section also includes links for 'Amazon Linux 2' and 'Security credentials'. Other visible sections include 'Summary', 'Number of instances' (set to 1), 'Virtual server type' (set to 't2.micro'), and 'Storage (volumes)'. At the bottom right of the main window, there is a large orange 'Launch instance' button.

The screenshot shows the AWS IAM 'Create access key' page. A green success message at the top states: 'Access key created. This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' Below this message, two access keys are listed: 'Access key' (checkbox checked) and 'Secret access key' (checkbox checked). The secret access key value is 'kRyw67/c3NzdUkxmXHgkKhVqFtx/PpkJmkxihHs'. There is a 'Hide' link next to the secret key. Below the keys, a section titled 'Access key best practices' lists several guidelines:

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

At the bottom of the page, there is a note: 'For more details about managing access keys, see the [best practices for managing AWS access keys](#)'. Below this note are two buttons: 'Download .csv file' and 'Done'.

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the output of the 'aws configure' command:

```
PS C:\Users\Lenovo\Desktop\terraform-scripts-1> aws configure
AWS Access Key ID [*****none]: AKIAS2VS4EQJLEHIDSPG
AWS Secret Access Key [*****none]: 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
Default region name [none]: ap-northeast-1
Default output format [none]: json
PS C:\Users\Lenovo\Desktop\terraform-scripts-1>
```

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the output of the 'terraform init' command:

```
PS C:\Users\Lenovo\Desktop\terraform-scripts-1> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.84.0...
```

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the output of the 'terraform plan' command. It includes the provider configuration, resource definitions, and the successful initialization message. The 'terraform plan' command is highlighted with a red box.

```
provider "aws" {
  region = "ap-northeast-1"
}

resource "aws_security_group" "allow_all" {
  name            = "allow_all"
  description     = "Allow all inbound and outbound traffic"

  ingress {
    from_port    = 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\Lenovo\Desktop\terraform-scripts-1> terraform plan
```

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the output of the 'terraform apply' command. It shows the creation of a new AWS Security Group named 'allow\_all'. The 'terraform apply' command is highlighted with a red box.

```
provider "aws" {
  region = "ap-northeast-1"
}

resource "aws_security_group" "allow_all" {
  name            = "allow_all"
  description     = "Allow all inbound and outbound traffic"

  ingress {
    from_port    = 0
  }
}

Plan: 4 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
PS C:\Users\Lenovo\Desktop\terraform-scripts-1> terraform apply
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files like `main.tf`, `terraform.exe`, `terraform.lock.hcl`, and `terraform.state`.
- Editor:** Displays the `main.tf` file containing Terraform configuration for three instances: `test_server`, `k8s_worker`, and `master_server`. The configuration includes specifying the AMI, instance type (t2.medium), security groups, and root block device.
- Terminal:** Shows the output of the Terraform command, indicating the creation of three resources: `aws_instance.master_server`, `aws_instance.test_server`, and `aws_instance.k8s_worker`. It also shows the completion message: `Apply complete! Resources: 3 added, 0 changed, 0 destroyed.`
- Bottom Status Bar:** Includes system icons for battery, signal, and temperature (23°C Haze), and a timestamp (17-01-2025).

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

- Left Navigation:** Shows categories like Dashboard, EC2 Global View, Events, Instances, Images, and Elastic Block Store.
- Instances Table:** Displays three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Test-server	i-001f2340bda339cdc	Running	t2.medium	Initializing	<a href="#">View alarms +</a>
k8s-worker	i-0535872a4bcce7418	Running	t2.medium	Initializing	<a href="#">View alarms +</a>
Master-server	i-0ff845f4341f3b1ae	Running	t2.medium	Initializing	<a href="#">View alarms +</a>
- Bottom Status Bar:** Includes system icons for battery, signal, and temperature (23°C Haze), and a timestamp (17-01-2025).

```
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.  
44bd1bd6cf564023a92cec1053fc2eb9  
root@ip-172-31-40-213:/home/ubuntu# history  
1 apt update  
2 sudo apt install openjdk-11-jdk -y  
3 wget https://raw.githubusercontent.com/akshu20791/Deployment-script/main/jenkins.sh  
4 chmod +x jenkins.sh  
5 ./jenkins.sh  
6 history  
root@ip-172-31-40-213:/home/ubuntu# java --version  
openjdk 17.0.13 2024-10-15  
OpenJDK Runtime Environment (build 17.0.13+11-Ubuntu-2ubuntu124.04)  
OpenJDK 64-Bit Server VM (build 17.0.13+11-Ubuntu-2ubuntu124.04, mixed mode, sharing)  
root@ip-172-31-40-213:/home/ubuntu# jenkins --version  
2.479.3  
root@ip-172-31-40-213:/home/ubuntu#  
  
i-085a313a279904c5e (Master-server)  
PublicIPs: 13.231.238.182 PrivateIPs: 172.31.40.213
```

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 10:48 PM 18-01-2025

## Getting Started

# Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

/var/lib/jenkins/secrets/initialAdminPassword

Please copy the password from either location and paste it below.

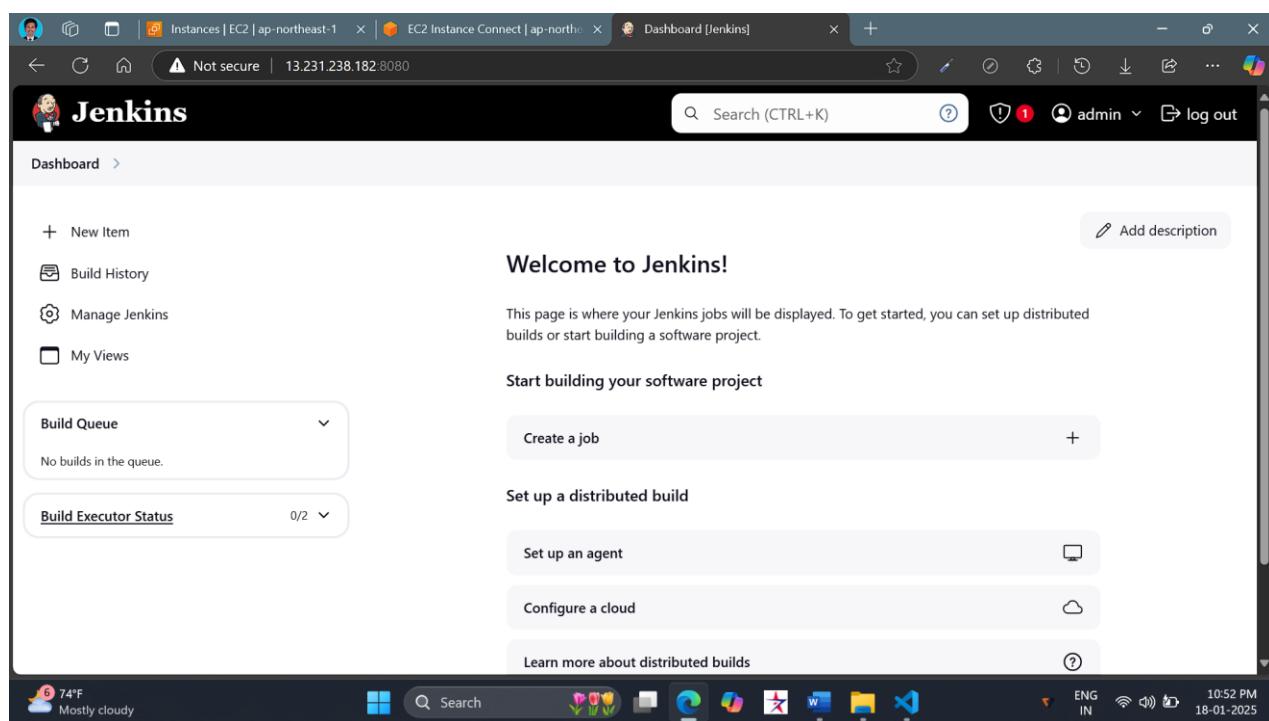
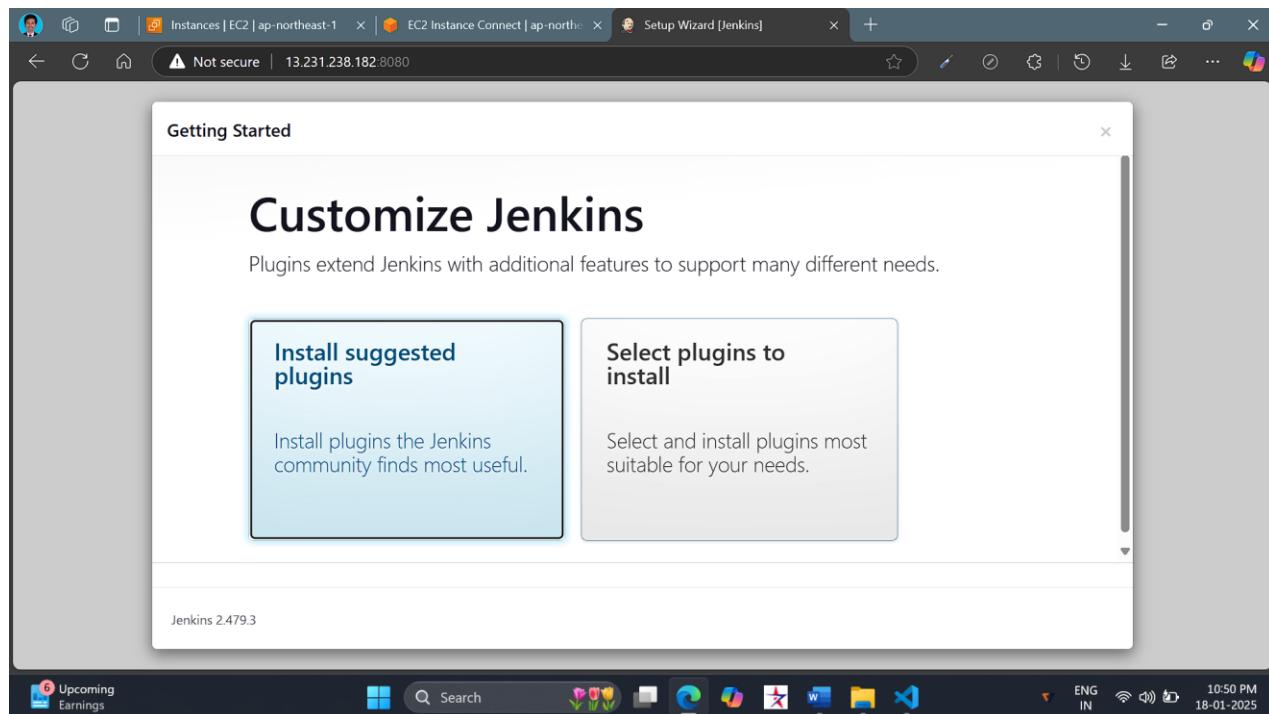
Administrator password

.....

Continue

Upcoming Earnings 6

CloudShell Search ENG IN 10:50 PM 18-01-2025



info: Selecting GID from range 100 to 999 ...  
info: Adding group `docker' (GID 114) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.  
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.  
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...  
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.  
root@ip-172-31-40-213:/home/ubuntu# docker --version  
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1  
root@ip-172-31-40-213:/home/ubuntu#

**i-085a313a279904c5e (Master-server)**

Public IPs: 13.231.238.182 Private IPs: 172.31.40.213

```
info: Adding group `docker` (GID 114) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...
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.
root@ip-172-31-40-213:/home/ubuntu# docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
root@ip-172-31-40-213:/home/ubuntu# sudo usermod -aG docker jenkins
root@ip-172-31-40-213:/home/ubuntu# visudo

i-085a313a279904c5e (Master-server)
PublicIPs: 13.231.238.182 PrivateIPs: 172.31.40.213
```

```
GNU nano 7.2          /etc/sudoers.tmp *
```

```
# User alias specification

# Cmnd alias specification

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

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

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

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

**i-085a313a279904c5e (Master-server)**

Public IPs: 13.231.238.182 Private IPs: 172.31.40.213

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Cloud Shell 74°F Mostly cloudy Search ENG IN 10:54 PM 18-01-2025

```
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...
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.
root@ip-172-31-40-213:/home/ubuntu# docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
root@ip-172-31-40-213:/home/ubuntu# sudo usermod -aG docker jenkins
root@ip-172-31-40-213:/home/ubuntu# visudo
root@ip-172-31-40-213:/home/ubuntu# service jenkins restart
```

**i-085a313a279904c5e (Master-server)**

Public IPs: 13.231.238.182 Private IPs: 172.31.40.213

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Cloud Shell 74°F Mostly cloudy Search ENG IN 10:55 PM 18-01-2025

The screenshot shows the Jenkins 'New Item' creation interface. At the top, there are tabs for 'Instances | EC2 | ap-northeast-1', 'EC2 Instance Connect | ap-northeast-1', and 'New Item [Jenkins]'. The main title is 'New Item'. Below it, the sub-navigation shows 'Dashboard > All > New Item'. The main content area is titled 'New Item' and contains a form to enter an item name, which is currently set to 'Project-3'. A section for selecting an item type is present, showing three options: 'Freestyle project', 'Pipeline', and 'Multi-configuration project'. The 'Pipeline' option is highlighted. At the bottom of the form is a blue 'OK' button. The status bar at the bottom of the browser window shows the date and time as '18-01-2025 10:59 PM'.

The screenshot shows the Jenkins Pipeline Syntax Snippet Generator. The URL in the address bar is '13.231.238.182:8080/job/Project-3/pipeline-syntax/'. The main title is 'Pipeline Syntax Snippet Generator'. The left sidebar has links for 'Online Documentation', 'Examples Reference', and 'IntelliJ IDEA GDSDL'. The main content area is titled 'Sample Step' and shows a code snippet: 'withCredentials: Bind credentials to variables'. This snippet is highlighted with a red rectangle. Below this, there is a detailed description of the 'withCredentials' step, mentioning that secret values are masked. A modal dialog is open over the description, showing a list of binding types under the heading 'Bindings'. One item, 'Secret text', is highlighted with a red rectangle. Other items listed include 'Certificate', 'Git Username and Password', 'SSH User Private Key', 'Secret ZIP file', and 'Secret file'. The status bar at the bottom shows the date and time as '18-01-2025 11:00 PM'.

The screenshot shows the Jenkins Pipeline Syntax Snippet General interface. In the 'Bindings' section, there is a 'Secret text' configuration. The 'Variable' field is empty. The 'Credentials' dropdown menu is open, showing a single item: 'Jenkins'. This 'Jenkins' item is highlighted with a red box. Below the dropdown is a button labeled 'Add'.

key file, are not masked. See the inline help for details and usage guidelines.

**Bindings**

☰ Secret text ?

Variable ?

Credentials ?

+ Add

Jenkins

Add ▾

Generate Pipeline Script

ADA -6.91%

Search

ENG IN 11:01 PM 18-01-2025

The screenshot shows the 'Jenkins Credentials Provider: Jenkins' configuration dialog. It has the following fields:

- Kind:** Secret text (highlighted with a red box)
- Scope:** Global (Jenkins, nodes, items, all child items, etc)
- Secret:** A password field containing '.....' (highlighted with a red box)
- ID:** dockerhubpass (highlighted with a red box)
- Description:** To hide docker hub password (highlighted with a blue box)

CAD/INR -0.59%

Search

ENG IN 11:04 PM 18-01-2025

The screenshot shows the Jenkins Pipeline Syntax Snippet General page. At the top, there are several tabs: Instances | EC2 | ap-northeast-1, EC2 Instance Connect | ap-north, Project-3 Config [Jenkins], and Pipeline Syntax Snippet General. The main content area is titled "Pipeline Syntax". It contains a "Global Variables" section with a note about unsupported features and a "Generate Pipeline Script" button. A code snippet is displayed in a text area, with the first few lines highlighted by a red box:

```
withCredentials([string(credentialsId: 'dockerhubpass', variable: '')]) {  
    // some block  
}
```

The screenshot shows the AWS CloudShell interface. The URL in the address bar is https://ap-northeast-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-northeast-1&... The terminal window displays a series of commands run on a root shell:

```
0 jenkins --version  
9 sudo apt install docker.io -y  
10 docker --version  
11 sudo usermod -aG docker jenkins  
12 visudo  
13 service jenkins restart  
14 sudo apt install software-properties-common  
15 sudo add-apt-repository --yes --update ppa:ansible/ansible  
16 sudo apt install ansible  
17 history  
root@ip-172-31-40-213:/home/ubuntu# ansible --version  
ansible [core 2.17.7]  
config file = /etc/ansible/ansible.cfg  
configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']  
ansible python module location = /usr/lib/python3/dist-packages/ansible  
ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections  
executable location = /usr/bin/ansible  
python version = 3.12.3 (main, Nov 6 2024, 18:32:19) [GCC 13.2.0] (/usr/bin/python3)  
jinja version = 3.1.2  
libyaml = True  
root@ip-172-31-40-213:/home/ubuntu#
```

Below the terminal, the text "i-085a313a279904c5e (Master-server)" is displayed, along with "Public IPs: 13.231.238.182 Private IPs: 172.31.40.213". The bottom of the screen shows the Windows taskbar with various icons and the date/time: 18-01-2025, 11:10 PM.

The screenshot shows the Jenkins plugin manager interface. The left sidebar has tabs for 'Updates', 'Available plugins' (which is selected and highlighted with a red box), 'Installed plugins', and 'Advanced settings'. The main area is titled 'Plugins' and contains a search bar and an 'Install' button (also highlighted with a red box). Below this is a table with columns for 'Name', 'Type', and 'Released'. The first row shows the 'Ansible' plugin, version 403.v8d0ca\_dcb\_b\_502, which is up for adoption. The second row shows the 'Docker' plugin, version 1.7.0, which integrates Jenkins with Docker. Other listed plugins include 'JavaMail API' and 'Command Agent Launcher'.

The screenshot shows the Jenkins plugin manager interface with the 'Download progress' tab selected (highlighted with a red box). The left sidebar includes 'Updates', 'Available plugins' (selected), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main area is titled 'Download progress' and shows a 'Preparation' section with a bulleted list: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. It also lists several plugins with their download status: Ansible (Success), Cloud Statistics (Success), Authentication Tokens API (Success), Docker Commons (Success), Apache HttpComponents Client 5.x API (Success), Commons Compress API (Success), Docker API (Success), Docker (Success), and Loading plugin extensions (Success). At the bottom, there are links to 'Go back to the top page' and 'Restart Jenkins when installation is complete and no jobs are running'.

The screenshot shows a web browser window with multiple tabs open. The active tab is 'Tools [Jenkins]' at the URL [13.231.238.182:8080/manage/configureTools/](https://13.231.238.182:8080/manage/configureTools/). The page displays the 'Ansible installations' configuration. A new entry is being created with the name 'ansible'. The 'Install automatically' checkbox is checked. Below the form are 'Save' and 'Apply' buttons. The browser's status bar at the bottom right shows the date as 18-01-2025.

The screenshot shows an AWS CloudShell session. The terminal window displays the output of the command `ansible --version`, which shows the version 2.17.7 and various configuration details. A red box highlights the command `vi /etc/ansible/hosts`. Below the terminal, a message box shows the public and private IP addresses: `i-085a313a279904c5e (Master-server)`, `Public IPs: 13.231.238.182 Private IPs: 172.31.40.213`. The browser's status bar at the bottom right shows the date as 18-01-2025.

```
MINGW64/c/Users/Lenovo/Desktop/Project-3
Lenovo@Balasastha_MINGW64 ~/Desktop/Project-3 (master)
$ vi ansible-playbook.yml
```



74°F Mostly cloudy Search ENG IN 11:20 PM 18-01-2025

```
MINGW64/c/Users/Lenovo/Desktop/Project-3
- name : Configure Docker on EC2 Instances
  hosts : all
  become: true
  connection : ssh
  tasks :
    - name: updating apt
      command : sudo apt-get update
    - name : Install Docker
      command : sudo apt-get install -y docker.io
    - name : Start Docker Service
      command : sudo systemctl start docker
    - name: Deploy Docker Container
      command: docker run -itd -p 8084:8081 balasastha/healthcare-domain:3
```



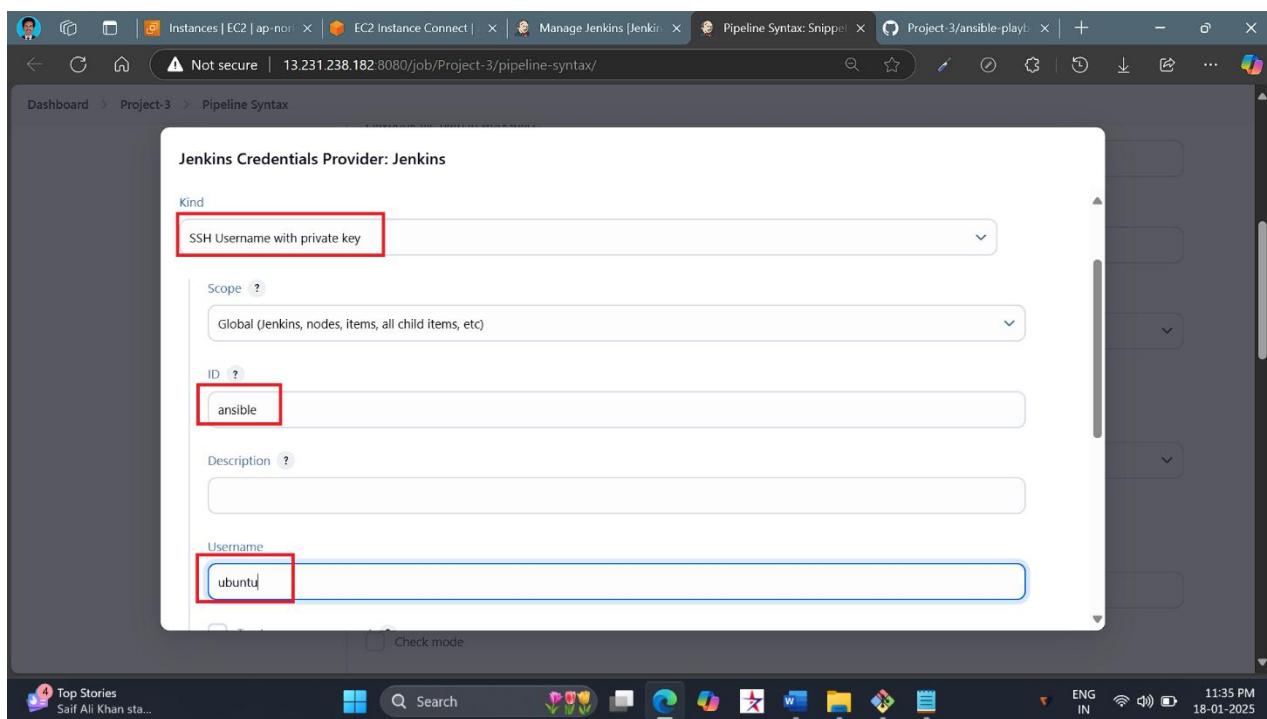
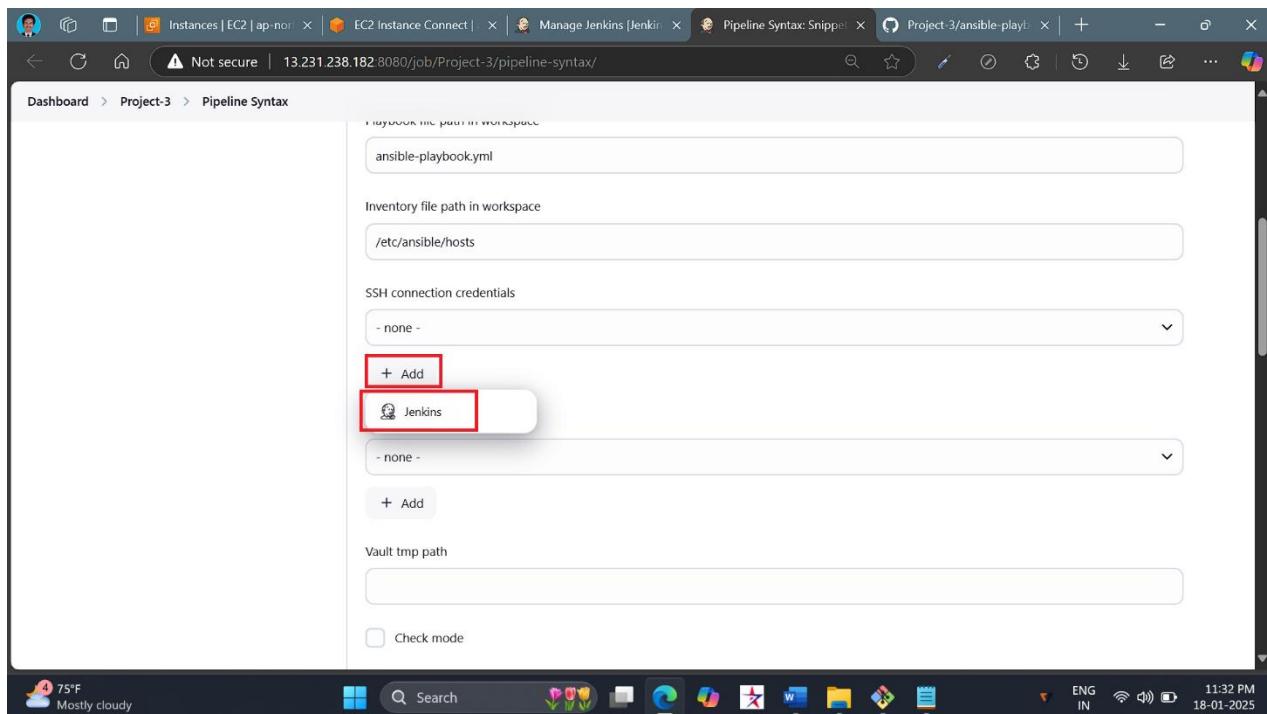
6 74°F Mostly cloudy Search ENG IN 11:22 PM 18-01-2025

```
MINGW64/c/Users/Lenovo/Desktop/Project-3
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ vi ansible-playbook.yml
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git add ansible-playbook.yml
warning: in the working copy of 'ansible-playbook.yml', LF will be replaced by CRLF the next time Git touches it
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git commit -m "second commit"
[master de25804] second commit
 1 file changed, 3 insertions(+), 6 deletions(-)
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git push origin -u master
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git push origin -u master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), done.
Total (delta 0, local 0) (delta 0, pack-reused 0 (from 0))
remote: resolving deltas: 100% (2/2); completed with 2 local objects.
To https://github.com/Balasastha/Project-3
 b354955..de25804 master -> master
branch 'master' set up to track 'origin/master'.
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ |
```

The screenshot shows a web browser window with the URL <http://13.231.238.182:8080/job/Project-3/pipeline-syntax/>. The page displays the Jenkins Pipeline Syntax configuration for a project named 'Project-3'. The left sidebar lists navigation links: Dashboard, Project-3, Pipeline Syntax, Global Variables Reference, Online Documentation, Examples Reference, and IntelliJ IDEA DSL. The main content area is titled 'Sample Step' and contains a dropdown menu with the option 'ansiblePlaybook: Invoke an ansible playbook' selected. Below this, there are input fields for 'Ansible tool' (set to 'ansible'), 'Playbook file path in workspace' (set to 'ansible-playbook.yml'), and 'Inventory file path in workspace' (set to '/etc/ansible/hosts'). Under 'SSH connection credentials', a dropdown menu shows '- none -'. At the bottom of the configuration form, there is a '+ Add' button. The browser's status bar at the bottom shows the date and time as 11:30 PM on 18-01-2025.



The screenshot shows the Jenkins Pipeline Syntax configuration page for a job named 'Project-3'. The 'Pipeline Syntax' tab is selected. The configuration includes:

- SSH connection credentials:** A dropdown menu set to 'ubuntu', highlighted with a red box.
- Vault credentials:** A dropdown menu set to '- none -'.
- Vault tmp path:** An empty input field.
- Check mode:** An unchecked checkbox.
- Use become:** A checked checkbox, highlighted with a red box.
- Become username:** An input field containing 'root'.

The status bar at the bottom shows battery level (ADA -5.04%), system tray icons, and the date/time (11:36 PM 18-01-2025).

The screenshot shows the Jenkins Pipeline Syntax configuration page for a job named 'Project-3'. The 'Pipeline Syntax' tab is selected. The configuration includes:

- Number of parallel processes to use:** An empty input field.
- Disable the host SSH key check:** A checked checkbox, highlighted with a red box.
- Colorized output:** An unchecked checkbox.
- Extra parameters:** An empty input field.
- Generate Pipeline Script:** A blue button.
- Pipeline Script Content:** A code block containing:

```
ansiblePlaybook become: true, credentialsId: 'ansible', disableHostKeyChecking: true, installation: 'ansible', inventory: '/etc/ansible/hosts', playbook: 'ansible-playbook.yml', vaultTmpPath: ''
```

The status bar at the bottom shows battery level (CAD/INR -0.59%), system tray icons, and the date/time (11:38 PM 18-01-2025).

```
Last login: Sat Jan 18 17:14:59 2025 from 3.112.23.4
ubuntu@ip-172-31-40-213:~$ sudo su
root@ip-172-31-40-213:/home/ubuntu# wget https://raw.githubusercontent.com/akshu20791/Deployment-script/main/k8s-master.sh
--2025-01-19 07:16:34-- https://raw.githubusercontent.com/akshu20791/Deployment-script/main/k8s-master.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1169 (1.1K) [text/plain]
Saving to: 'k8s-master.sh'

k8s-master.sh          100%[=====] 1.14K --.-KB/s   in 0s

2025-01-19 07:16:35 (108 MB/s) - 'k8s-master.sh' saved [1169/1169]

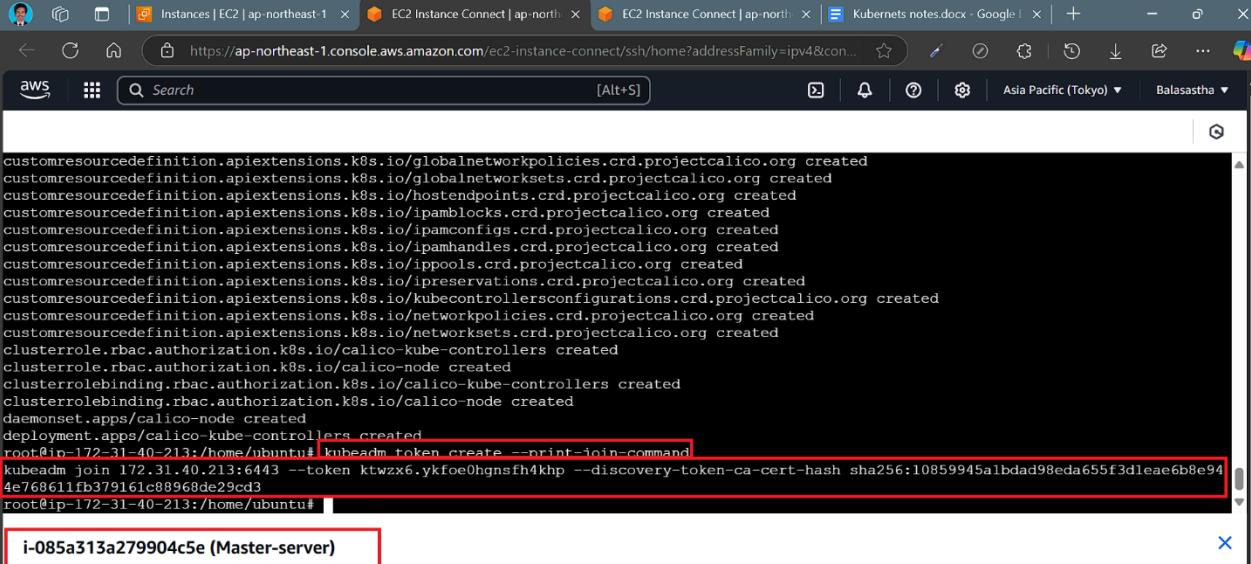
root@ip-172-31-40-213:/home/ubuntu# chmod 777 k8s-master.sh
root@ip-172-31-40-213:/home/ubuntu# ./k8s-master.sh
--2025-01-19 07:17:00-- https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installDocker.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.110.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response...
```

i-085a313a279904c5e (Master-server)  
PublicIPs: 13.231.238.182 PrivateIPs: 172.31.40.213

```
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 (gemu) binaries on this host.
kubelet, kubeadm & kubectl are successfully installed
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
root@ip-172-31-38-23:/home/ubuntu# history
1 wget https://raw.githubusercontent.com/akshu20791/Deployment-script/main/k8s-nodes.sh
2 chmod 777 k8s-nodes.sh
3 ./k8s-nodes.sh
4 history
root@ip-172-31-38-23:/home/ubuntu# modprobe br_netfilter
root@ip-172-31-38-23:/home/ubuntu# echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables
root@ip-172-31-38-23:/home/ubuntu# echo 1 > /proc/sys/net/ipv4/ip_forward
root@ip-172-31-38-23:/home/ubuntu#
```

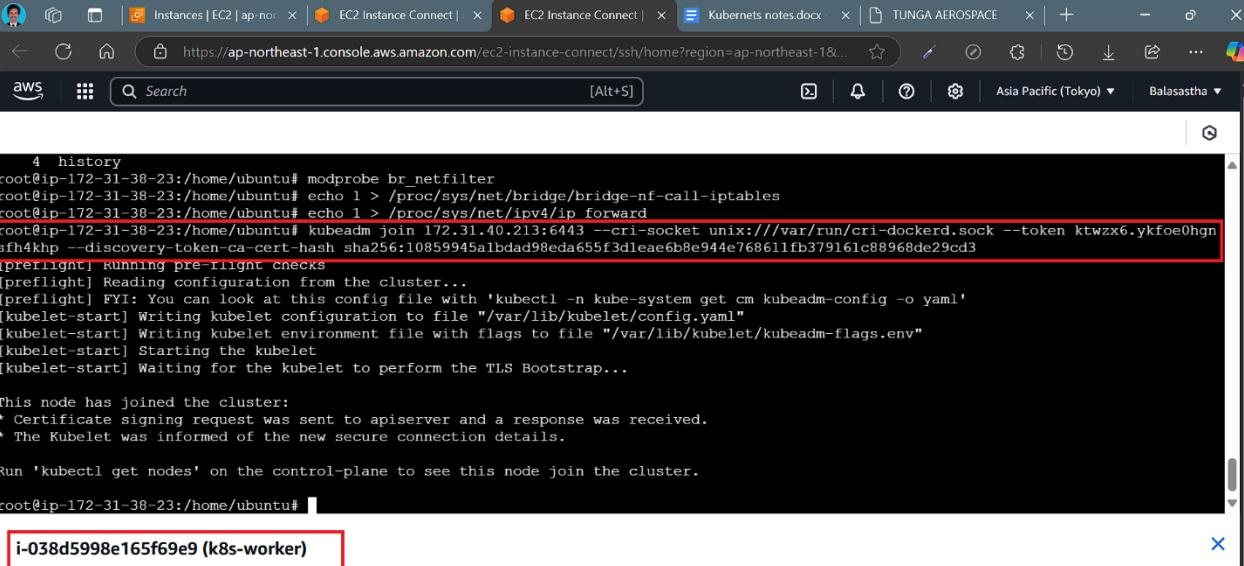
i-038d5998e165f69e9 (k8s-worker)  
PublicIPs: 54.95.66.217 PrivateIPs: 172.31.38.23



```
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/irpreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
root@ip-172-31-40-213:/home/ubuntu# kubeadm token create --print-join-command
kubeadm join 172.31.40.213:6443 --token ktwzx6.ykfoe0hgnsh4khp --discovery-token-ca-cert-hash sha256:10859945albdad98eda655f3d1eae6b8e94
4e768611fb379161c88968de29cd3
root@ip-172-31-40-213:/home/ubuntu#
```

i-085a313a279904c5e (Master-server)

PublicIPs: 13.231.238.182 PrivateIPs: 172.31.40.213



```
4 history
root@ip-172-31-38-23:/home/ubuntu# modprobe br_netfilter
root@ip-172-31-38-23:/home/ubuntu# echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables
root@ip-172-31-38-23:/home/ubuntu# echo 1 > /proc/sys/net/ipv4/ip_forward
root@ip-172-31-38-23:/home/ubuntu# kubeadm join 172.31.40.213:6443 --cri-socket unix:///var/run/cri-dockerd.sock --token ktwzx6.ykfoe0hgn
sh4khp --discovery-token-ca-cert-hash sha256:10859945albdad98eda655f3d1eae6b8e94e768611fb379161c88968de29cd3
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

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

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
root@ip-172-31-38-23:/home/ubuntu#
```

i-038d5998e165f69e9 (k8s-worker)

PublicIPs: 54.95.66.217 PrivateIPs: 172.31.38.23

root@ip-172-31-40-213:/home/ubuntu# kubectl get nodes

NAME	STATUS	ROLES	AGE	VERSION
ip-172-31-38-23	Ready	<none>	21m	v1.29.13
ip-172-31-40-213	Ready	control-plane	32m	v1.29.13

i-085a313a279904c5e (Master-server)  
PublicIPs: 13.231.238.182 PrivateIPs: 172.31.40.213

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 26°C Mostly cloudy 01:20 PM 19-01-2025

Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)

\$ vi deployment.yaml

CAD/INR -0.59% Search 01:54 PM 19-01-2025

```
MINGW64/c/Users/Lenovo/Desktop/Project-3
apiVersion: apps/v1
kind: Deployment
metadata:
  name: healthcare
spec:
  replicas: 1
  selector:
    matchLabels:
      app: healthcare
  template:
    metadata:
      labels:
        app: healthcare
    spec:
      containers:
        - name: healthcareproject
          image: balasastha/healthcare-domain:3
          ports:
            - containerPort: 8084

---
apiVersion: v1
kind: Service
metadata:
  name: healthcare
spec:
  selector:
    app: healthcare
  type: NodePort
  ports:
    - port: 80
      targetPort: 8084
      nodePort: 30004

deployment.yaml [+] [unix] (05:29 01/01/1970) 33,21 A11
```

```
MINGW64:/c/Users/Lenovo/Desktop/Project-3
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ vi deployment.yaml
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git add deployment.yaml
warning: in the working copy of 'deployment.yaml', LF will be replaced by CRLF the next time Git touches it

Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git commit -m "third commit"
[master 0f03a7e] third commit
 1 file changed, 33 insertions(+)
 create mode 100644 deployment.yaml

Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git push origin -u master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 509 bytes | 509.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Balasastha/Project-3
   de25804..8f215f2  master -> master
branch 'master' set up to track 'origin/master'.

Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ |
```

```
MINGW64/c/Users/Lenovo/Desktop/Project-3
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ vi Jenkinsfile
```

```
        sh "mvn clean package"
    }
stage ('Build docker image'){
    steps
    {
        echo 'Building docker image for you..'
        sh 'docker build -t balasastha/healthcare-domain:3 .'
    }
}
stage ('Docker login'){
    steps
    {
        echo 'Logging in into docker...'
        withCredentials([string(credentialsId: 'dockerhubpass', variable: 'dockerhubpass')]) {
            sh 'docker login -u balasastha -p ${dockerhubpass}'
        }
    }
}
stage ('Push image into docker'){
    steps
    {
        echo 'Pushing the docker image for you'
        sh 'docker push balasastha/healthcare-domain:3'
    }
}
stage ('Invoking ansible playbook'){
    steps
    {
        echo 'Running ansible playbook in test server...'
        ansiblePlaybook become: true, credentialsId: 'ansible', disableHostKeyChecking: true, installation: 'ansible', inventory: '/etc/ansible/hosts', playbook: 'ansible-playbook.yaml', vaultTmpPath: ''
    }
}
stage ('Deployment of project on production server'){
    steps
    {
        echo 'Deploying using kubernetes..'
        sh 'sudo kubectl apply -f deployment.yaml'
        sh 'sudo kubectl get all'
    }
}
```

Jenkinsfile[+] [unix] (05:29 01/01/1970) 59,0–1 Bot

```
MINGW64/c/Users/Lenovo/Desktop/Project-3
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ vi Jenkinsfile
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git add Jenkinsfile
warning: in the working copy of 'Jenkinsfile', LF will be replaced by CRLF the next time Git touches it
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git commit -m "Fourth commit"
[master ade35d5] Fourth commit
 1 file changed, 59 insertions(+)
 create mode 100644 Jenkinsfile
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git push origin -u master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 825 bytes | 825.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Balasastha/Project-3
 8f215f2..ade35d5 master -> master
branch 'master' set up to track 'origin/master'.
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ |
```

The screenshot shows a web browser window with the URL <https://13.231.238.182:8080/job/Project-3/configure>. The page is titled 'Project-3 Config [Jenkin...]' and displays the 'Configuration' section for the 'Project-3' pipeline. The 'Pipeline' tab is selected. The 'Definition' field contains 'Pipeline script from SCM'. Under the 'SCM' section, 'Git' is selected. The 'Repositories' section shows a single repository with the 'Repository URL' set to <https://github.com/Balasastha/Project-3>. The 'Credentials' dropdown is set to '- none -'. At the bottom of the form are 'Save' and 'Apply' buttons. The browser's address bar shows multiple tabs related to EC2 instances and Jenkins.

A screenshot of a Microsoft Edge browser window. The address bar shows a non-secure connection to 13.231.238.182:8080/job/Project-3/6/console. The main content area displays the terminal output of a deployment command:

```
+ sudo kubectl apply -f deployment.yaml
deployment.apps/healthcare unchanged
service/healthcare unchanged
[Pipeline] sh
+ sudo kubectl get all
NAME           READY   STATUS    RESTARTS   AGE
pod/healthcare-65894f649d-plft6  1/1     Running   0          29m

NAME             TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/healthcare   NodePort   10.100.158.103 <none>       80:30007/TCP  42m
service/kubernetes  ClusterIP  10.96.0.1    <none>       443/TCP   131m

NAME            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/healthcare  1/1      1           1          29m

NAME           DESIRED   CURRENT   READY   AGE
replicaset.apps/healthcare-65894f649d  1         1         1      29m
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

The taskbar at the bottom shows various pinned icons and the date/time: 03:40 PM 19-01-2025.

A screenshot of a Microsoft Edge browser window showing the GitHub settings page for a repository. The URL is https://github.com/Balasastha/Project-3/settings/hooks/new. The left sidebar has a 'Webhooks' section highlighted with a red box. The main form fields are outlined with red boxes:

- Payload URL \***: http://13.231.238.182:8080/github-webhook/
- Content type \***: application/json
- Secret**: (empty field)
- SSL verification**:
  - Enable SSL verification (radio button selected)
  - Disable (not recommended)
- Which events would you like to trigger this webhook?**
  - Just the push event (radio button selected)
  - Send me everything.
  - Let me select individual events

The taskbar at the bottom shows various pinned icons and the date/time: 04:20 PM 19-01-2025.

The screenshot shows the Jenkins configuration interface for a project named 'Project-3'. The 'General' tab is selected. In the 'GitHub project' section, the URL 'https://github.com/Balasastha/Project-3' is entered. Under the 'Triggers' section, the 'GitHub hook trigger for GITScm polling' checkbox is checked. Both sections are highlighted with red boxes.

```
MINGW64:/c/Users/Lenovo/Desktop/Project-3
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ vi readme.txt
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git add readme.txt
warning: in the working copy of 'readme.txt', LF will be replaced by CRLF the next time Git touches it
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git commit -m "fifth commit"
[master eee8c7] fifth commit
 1 file changed, 1 insertion(+), 1 deletion(-)
Lenovo@Balasastha MINGW64 ~/Desktop/Project-3 (master)
$ git push origin -u master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 282 bytes | 282.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Balasastha/Project-3
 f4aa6d0..eee8c7 master -> master
branch 'master' set up to track 'origin/master'.
```

The screenshot shows a web browser window with a URL of <https://13.231.238.182:8080/job/Project-3/>. The page title is "Project-3". On the left, there's a sidebar with various project management options like Status, Changes, Build Now, Configure, Delete Pipeline, GitHub, Stages, Rename, Pipeline Syntax, and GitHub Hook Log. Below this is a "Builds" section with a table showing two rows: #7 (10:56 AM) and #6 (9:29 AM). The row for build #7 is highlighted with a red box. The table includes columns for build number, timestamp, progress bar, and delete/refresh icons.

The screenshot shows a web browser window with a URL of <https://ap-northeast-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-northeast-1&instanceId=i-09b2d3b16af005414>. The terminal window displays the command `ubuntu@ip-172-31-33-141:~$ vi install_node_exporter.sh`. Below the terminal, the instance details are shown: **i-09b2d3b16af005414 (Test-server)**, Public IPs: 18.183.176.119, Private IPs: 172.31.33.141. At the bottom, there are links for CloudShell and Feedback, and standard browser navigation controls.

```
#!/bin/bash

# Exit the script if any command fails
set -e

echo "Starting Node Exporter installation..."

# Step 1: Download Node Exporter
echo "Downloading Node Exporter..."
wget https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux-amd64.tar.gz -O /tmp/node_exporter.tar.gz

# Step 2: Extract the archive
echo "Extracting Node Exporter..."
tar -xvzf /tmp/node_exporter.tar.gz -C /tmp/

# Step 3: Move the binary
echo "Moving Node Exporter binary to /usr/local/bin..."
sudo mv /tmp/node_exporter-1.6.1.linux-amd64/node_exporter /usr/local/bin/
-- INSERT --
```

i-09b2d3b16af005414 (Test-server)  
Public IPs: 18.183.176.119 Private IPs: 172.31.33.141

```
ubuntu@ip-172-31-33-141:~$ chmod +x install_node_exporter.sh
ubuntu@ip-172-31-33-141:~$ ./install_node_exporter.sh
Starting Node Exporter installation...
Downloading Node Exporter...
--2025-01-19 11:14:46-- https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 20.27.177.113
Connecting to github.com (github.com)|20.27.177.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/5509b569-5c34-471e-8598-c05c0733bb7f?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250119%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250119T111446Z&X-Amz-Expires=300&X-Amz-Signature=f1ad95da233370b26c6f06e224b61d95f2c3dbf7ea78d60df636c02ee932895&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.6.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2025-01-19 11:14:46-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/5509b569-5c34-471e-8598-c05c0733bb7f?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250119%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250119T111446Z&X-Amz-Expires=300&X-Amz-Signature=f1ad95da233370b26c6f06e224b61d95f2c3dbf7ea78d60df636c02ee932895&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.6.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.111.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
```

i-09b2d3b16af005414 (Test-server)  
Public IPs: 18.183.176.119 Private IPs: 172.31.33.141

https://ubuntu.com/aws/pro  
Expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status

```
Last login: Sun Jan 19 07:18:34 2025 from 3.112.23.4
ubuntu@ip-172-31-38-23:~$ sudo su
root@ip-172-31-38-23:/home/ubuntu# vi install_node_exporter.sh
root@ip-172-31-38-23:/home/ubuntu# chmod +x install_node_exporter.sh
root@ip-172-31-38-23:/home/ubuntu# ./install_node_exporter.sh
Starting Node Exporter installation...
Downloading Node Exporter...
--2025-01-19 11:19:52-- https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 20.27.177.113
Connecting to github.com (github.com) [20.27.177.113]:443... connected.
HTTP request sent, awaiting response...
```

i-038d5998e165f69e9 (k8s-worker)  
PublicIPs: 54.95.66.217 PrivateIPs: 172.31.38.23

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

26°C Mostly cloudy 04:49 PM 19-01-2025

Not secure | 54.95.66.217:9100/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_routines Number of goroutines that currently exist.
# TYPE go_routines gauge
go_routines 7
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.20.6"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 995216
# 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 995216
# 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 1.445232e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 735
# 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 7.233112e+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 995216
# 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 1.613824e+06
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
```

26°C Mostly cloudy 04:51 PM 19-01-2025

# 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 7  
# HELP go\_info Information about the Go environment.  
# TYPE go\_info gauge  
go\_info{version="go1.20.6"} 1  
# HELP go\_memstats\_alloc\_bytes Number of bytes allocated and still in use.  
# TYPE go\_memstats\_alloc\_bytes gauge  
go\_memstats\_alloc\_bytes 994640  
# HELP go\_memstats\_alloc\_bytes\_total Total number of bytes allocated, even if freed.  
# TYPE go\_memstats\_alloc\_bytes\_total 994640  
# HELP go\_memstats\_alloc\_bytes\_total 994640  
# HELP go\_memstats\_buck\_hash\_bytes Number of bytes used by the profiling bucket hash table.  
# TYPE go\_memstats\_buck\_hash\_bytes gauge  
go\_memstats\_buck\_hash\_bytes 1.445208e+06  
# HELP go\_memstats\_frees\_total Total number of frees.  
# TYPE go\_memstats\_frees\_total counter  
go\_memstats\_frees\_total 741  
# 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 7.233112e+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 994640  
# 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 1.613824e+06  
# HELP go\_memstats\_heap\_inuse\_bytes Number of heap bytes that are in use.  
# TYPE go\_memstats\_heap\_inuse\_bytes gauge

26°C Mostly cloudy      Search      CloudShell      Feedback      04:52 PM 19-01-2025

```
Get:44 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2940 B]
Get:45 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:46 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [356 B]
Fetched 30.5 MB in 6s (4862 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
root@ip-172-31-33-152:/home/ubuntu# vi setup_prometheus_and_node_exporter.sh
root@ip-172-31-33-152:/home/ubuntu# chmod +x setup_prometheus_and_node_exporter.sh
root@ip-172-31-33-152:/home/ubuntu# ./setup_prometheus_and_node_exporter.sh
Starting setup for Prometheus and Node Exporter...
Downloading Prometheus...
--2025-01-19 11:42:02-- https://github.com/prometheus/prometheus/releases/download/v2.47.0/prometheus-2.47.0.linux-amd64.tar.gz
Resolving github.com (github.com)... 20.27.177.113
Connecting to github.com (github.com) |20.27.177.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/c5680b71-f095-412f-8e17-4e410f346f4d?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250119%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250119T114202Z&X-Amz-Expires=300&X-Amz-Signature=5755d4972d47b6920c490f2c8df1cd315381ae8dbafcef718927e0d8d361575b&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.47.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [follow]
```

i-07214677de65234a2 (grafana)  
PublicIPs: 3.112.195.232 PrivateIPs: 172.31.33.152

CloudShell      Feedback      © 2025, Amazon Web Services, Inc. or its affiliates.      Privacy      Terms      Cookie preferences

26°C Mostly cloudy      Search      CloudShell      Feedback      05:13 PM 19-01-2025

The screenshot shows the Prometheus Targets page. At the top, there are tabs for 'All scrape pools' (selected), 'All', 'Unhealthy', and 'Collapse All'. A search bar says 'Filter by endpoint or labels' with filters for 'Unknown', 'Unhealthy' (checked), and 'Healthy'. Below this, a section titled 'node\_exporter (2/2 up)' shows two entries:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://18.183.176.119:9100/metrics	UP	instance="18.183.176.119:9100", job="node_exporter"	15.880s ago	19.564ms	
http://54.95.66.217:9100/metrics	UP	instance="54.95.66.217:9100", job="node_exporter"	4.991s ago	13.590ms	

At the bottom of the window, there is a status bar with a weather icon (26°C, Mostly cloudy), a search bar, and system icons.

The screenshot shows an AWS CloudShell session. The terminal window displays the following commands being run:

```
root@ip-172-31-33-152:/home/ubuntu# vi install_grafana.sh
root@ip-172-31-33-152:/home/ubuntu# chmod +x install_grafana.sh
root@ip-172-31-33-152:/home/ubuntu# ./install_grafana.sh
```

A red box highlights the command `./install_grafana.sh`. Below the terminal, a message box shows:

**i-07214677de65234a2 (grafana)**  
Public IPs: 3.112.195.232 Private IPs: 172.31.33.152

At the bottom of the window, there is a status bar with a weather icon (26°C, Mostly cloudy), a search bar, and system icons.

Welcome to Grafana

Need help? Documentation Tutorials Community Public Slack

**Basic**

The steps below will guide you to quickly finish setting up your Grafana installation.

**TUTORIAL**  
DATA SOURCE AND DASHBOARDS  
Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

**DATA SOURCES**  
Add your first data source

**DASHBOARDS**  
Create your first dashboard

Learn how in the docs [\[link\]](#) Learn how in the docs [\[link\]](#)

Panels: https://grafana.com/tutorials/grafana-fundamentals?utm\_source=grafana\_g... Latest from the blog

26°C Mostly cloudy

Search bar: Search ctrl+k

Bottom navigation: Home > Connections > Data sources > prometheus

Home > Connections > Data sources > prometheus

**prometheus**

Type: Prometheus

**Settings** **Dashboards**

Configure your Prometheus data source below  
Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources from Grafana Labs with the free-forever Grafana Cloud plan.

Name: prometheus Default:

Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#).

Fields marked with \* are required

26°C Mostly cloudy

The screenshot shows the 'Connections' section of Grafana, specifically the 'Data sources' configuration for 'prometheus'. The 'Name' field is set to 'prometheus', and the 'Default' toggle is turned on. A note at the top states: 'Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#)'. Below this, a message says 'Fields marked with \* are required'. The 'Connection' section contains a 'Prometheus server URL' input field containing 'http://3.112.195.232:9090', which is highlighted with a red rectangle. The 'Authentication' section is partially visible below.

The screenshot shows the 'Dashboards' section of Grafana, specifically the 'Edit panel' view for a new dashboard. The 'Panel Title' is set to 'No data'. There are two query panels: 'A' and 'B', both configured for the 'prometheus' data source. Panel A has a query: `process_cpu_seconds_total{instance="18.183.176.119:9100"}`. The 'Metric' dropdown is highlighted with a red rectangle. The right side of the screen displays various configuration options under 'Panel options' and 'Tooltip' sections, such as 'Title', 'Description', and 'Transparent background'. The bottom status bar shows system information including the date and time.

The screenshot shows the Grafana dashboard editor interface. A single panel titled "Panel Title" is displayed, showing the message "No data". The panel contains a query section for Prometheus. The query is:

```
process_cpu_seconds_total(instance="54.95.66.217:9100")
```

The "Label filters" section is highlighted with a red box. It shows a dropdown for "Metric" set to "process\_cpu\_seconds\_total", a dropdown for "instance" set to "54.95.66.217:9100", and an equals sign operator. There is also a "+" button to add more filters.

On the right side of the screen, the "Panel options" sidebar is open, showing fields for "Title" (set to "Panel Title"), "Description", "Transparent background" (disabled), "Panel links", "Repeat options", and "Tooltip" (disabled). Buttons for "Discard panel" and "Save dashboard" are at the top right.

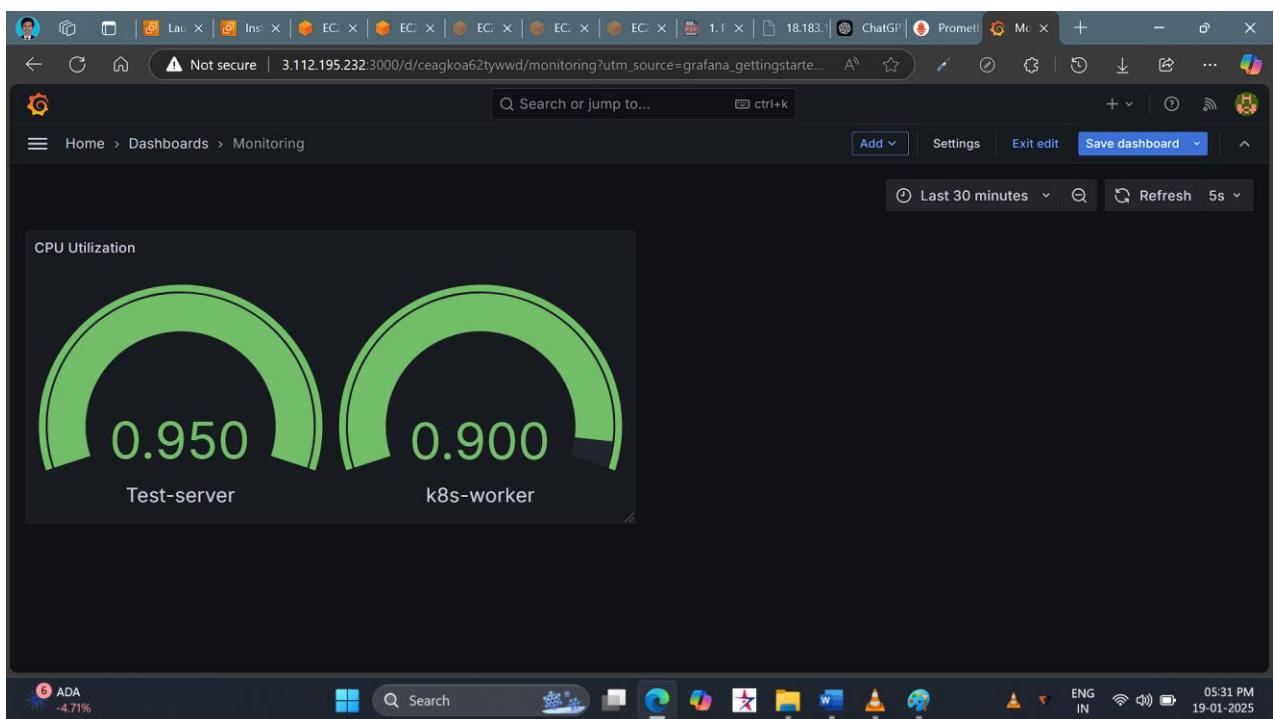
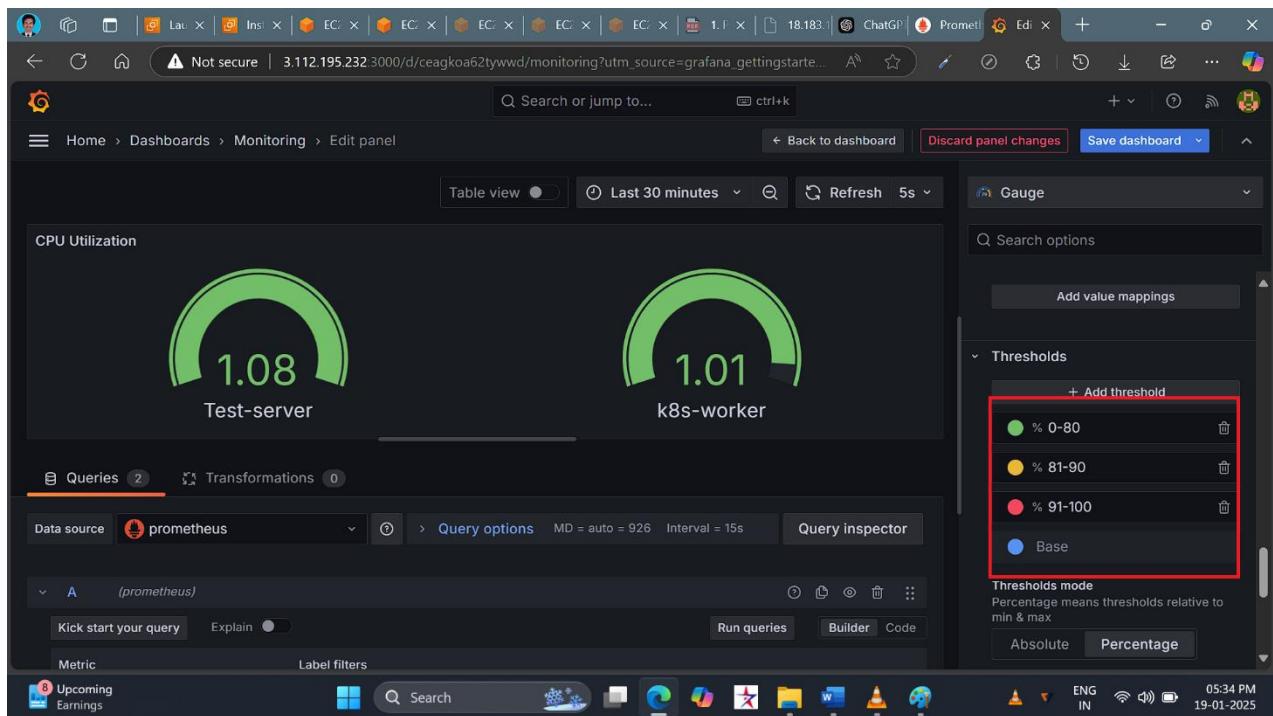
The screenshot shows the Grafana dashboard editor interface. A panel titled "Panel Title" displays two gauge charts. The left chart is for "Test-server" with a value of 0.880. The right chart is for "k8s-worker" with a value of 0.830. Both charts have a red border around them.

The panel contains a query section for Prometheus:

```
process_cpu_seconds_total(instance="18.183.176.119:9100")
```

The "Label filters" section is identical to the one in the previous screenshot, showing "process\_cpu\_seconds\_total" as the metric and "18.183.176.119:9100" as the instance.

The "Panel options" sidebar on the right is expanded, showing "Override 1" and "Override 2" sections. Each section has a "Fields with name" dropdown set to "process\_cpu\_seconds\_total" and a "Standard options > Display name" field with the value "Test-server" for the first and "k8s-worker" for the second. Buttons for "Discard panel" and "Save dashboard" are at the top right.



**HEALTHCARE DOMAIN PROJECT HAS BEEN  
SUCCESSFULLY COMPLETED**

**THANK YOU!!!**