Project - Medicure

Submitted by - **Maulik DEVANI**
Date of submission -13-07-2024
Submitted to - Mr.Vikul

● Problem Statement :

- The Medicure would centrally like to manage all the doctor's and patient's data across the Medicure hospitals in various cities. They have developed an microservice, which offers these services. In order to reduce unnecessary maintenance cost and manual labor, they would like to automate their application build and deployment process using DevOps. They are fine to use any one of the (AWS, Azure, GCP) cloud platform as their primary cloud service provider.
- The company's primary goal is to deliver the product updates frequently to production with High quality & Reliability.
- Kubernetes cluster must contain at least 2 servers and must be monitored continuously using Prometheus and dashboard must be visualized using Grafana.

Following are the problems:

✓ **Building Complex builds is difficult**
✓ **Manual efforts to test various components/modules of the project**
✓ **Incremental builds are difficult to manage, test and deploy**
✓ **Creation of infrastructure and configure it manually is very time consuming**
✓ **Continuous manual monitoring the application is quite challenging.**

● **Aim:**

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.

Tools:

✓ 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
✓ Terraform - For creation of infrastructure.
✓ Kubernetes – for running containerized application in managed cluster.

- **Infrastructure:**

Jenkins Master: #Create and Schedule the job
Jenkins Slave: #To perform build app. Docker & Build

K8s Master: #To deploy the pods
K8s Worker Node-1:
K8s Worker Node-2:

- **Tools:**

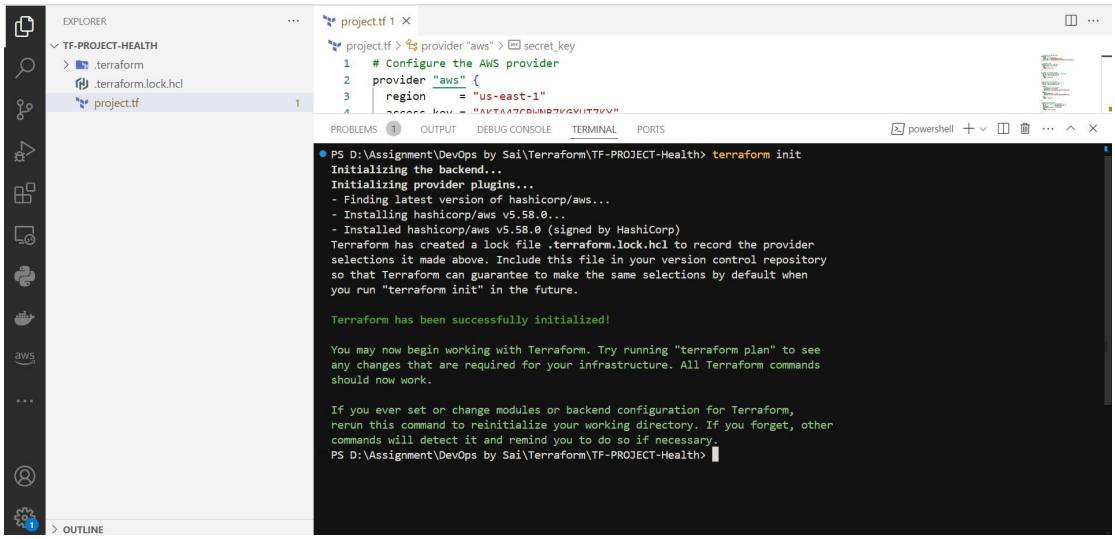
Jenkins Master: #JDK, Jenkins, Git, Ansible
Jenkins Slave(Build-Server): #JDK, Git, Maven, Docker

K8s Master: #cri, kubelet, kubectl, kubeadm,….
K8s Worker Node-1:
K8s Worker Node-2:

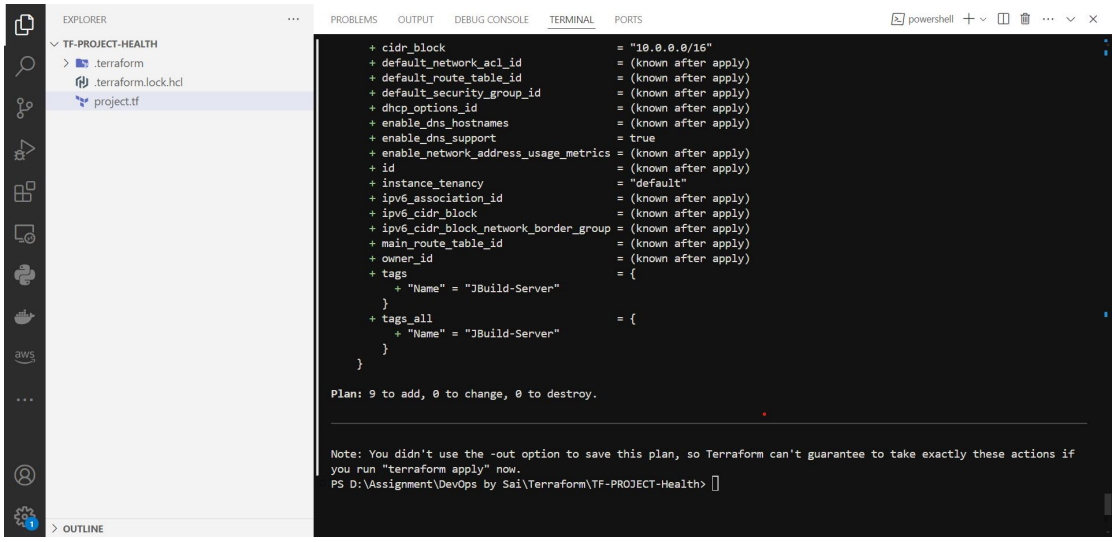
- **Installation and Integration of server:**

With terraform:
Providing build server:

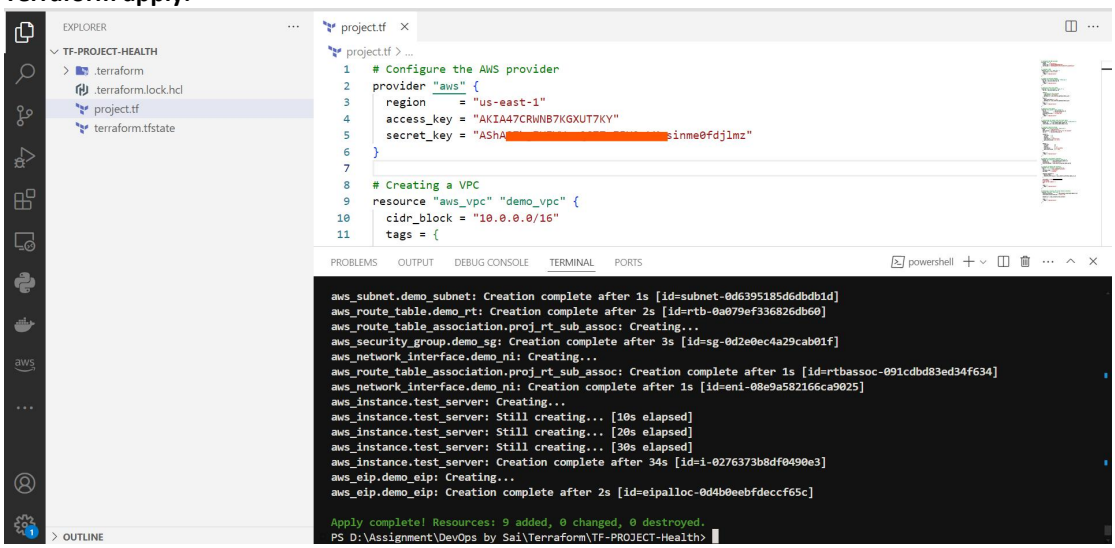
➢ **Initialize terraform**

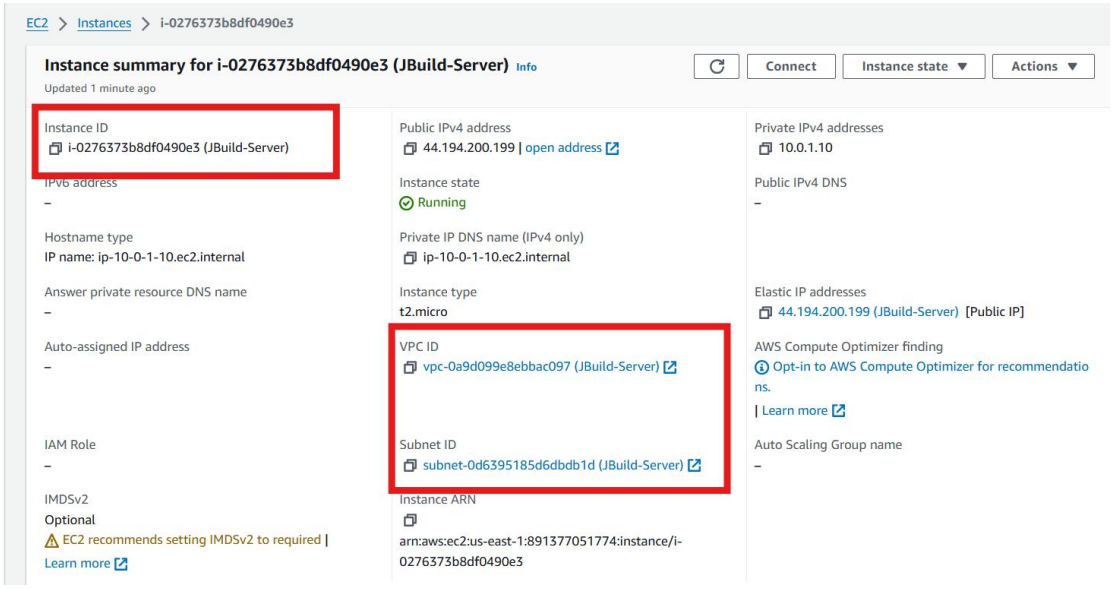


➢ Terraform Plan:



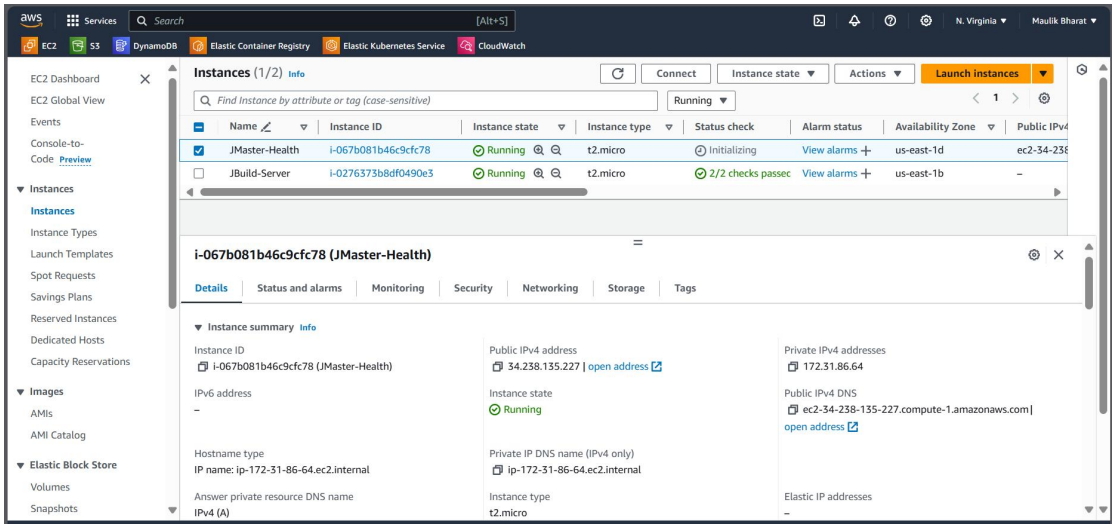
Terraform apply:





● Installation and configuration of Jenkins Master:

❖ **Jenkins Master:** **#JDK, Jenkins, Git**



➢ **Installation jdk:**

```
root@ip-172-31-86-64:~# apt-get install openjdk-17-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

```
root@ip-172-31-86-64:~# java --version
openjdk 17.0.11 2024-04-16
OpenJDK Runtime Environment (build 17.0.11+9-Ubuntu-122.04.1)
OpenJDK 64-Bit Server VM (build 17.0.11+9-Ubuntu-122.04.1, mixed mode, sharing)
root@ip-172-31-86-64:~# git --version
git version 2.34.1
```

➢ **Installation jenkins:**

https://www.jenkins.io/doc/book/installing/linux/#debianubuntu

```
root@ip-172-31-86-64:~# cat jenkins.sh
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
root@ip-172-31-86-64:~# sh jenkins.sh
```

```
root@ip-172-31-86-64:~# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-07-13 11:15:57 UTC; 5s ago
   Main PID: 5410 (java)
      Tasks: 44 (limit: 1120)
     Memory: 329.5M
        CPU: 43.064s
     CGroup: /system.slice/jenkins.service
             └─5410 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
```

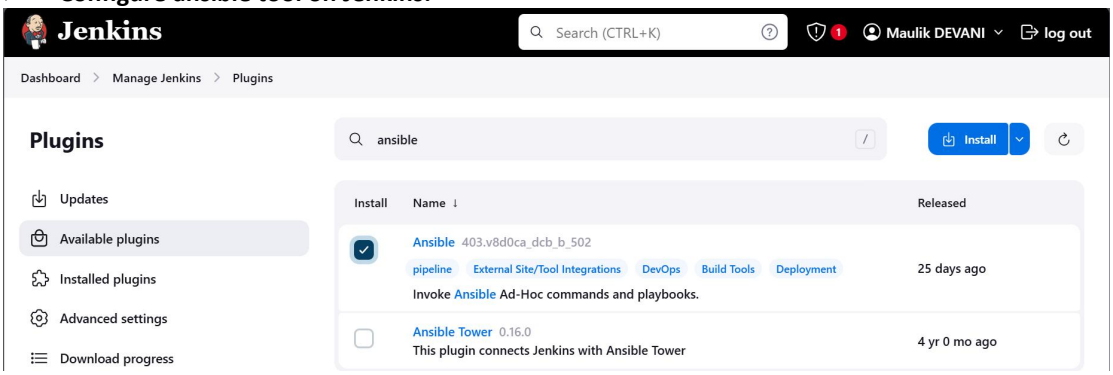
➢ Installing Ansible on JMaster server for configure JSlave:

```
root@ip-172-31-86-64:~# cat ansible.sh
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible -y
root@ip-172-31-86-64:~# sh ansible.sh
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:5 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:6 https://pkg.jenkins.io/debian-stable binary/ Release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
11 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
```

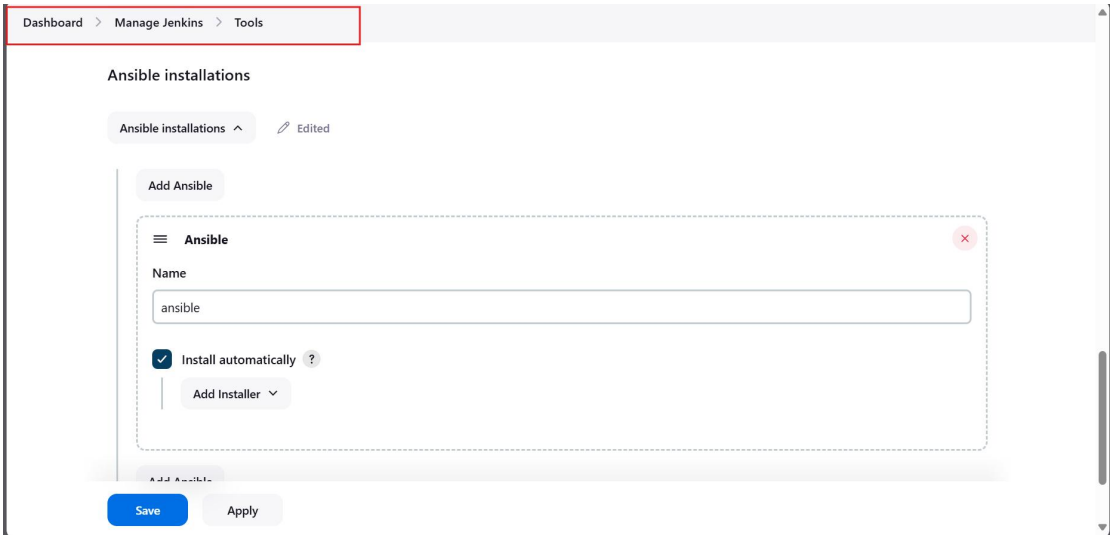
➢ **Provide host(Target server):**

```
root@ip-172-31-86-64:/etc/ansible# cat hosts
[testnodes]
44.194.200.199
```

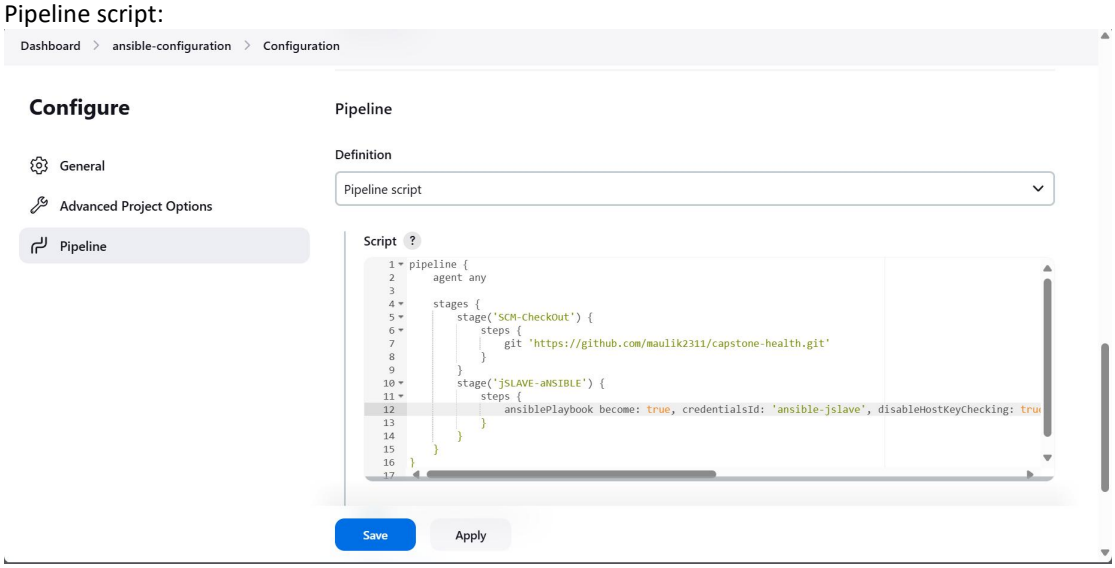
➢ **Configure ansible tool on Jenkins:**



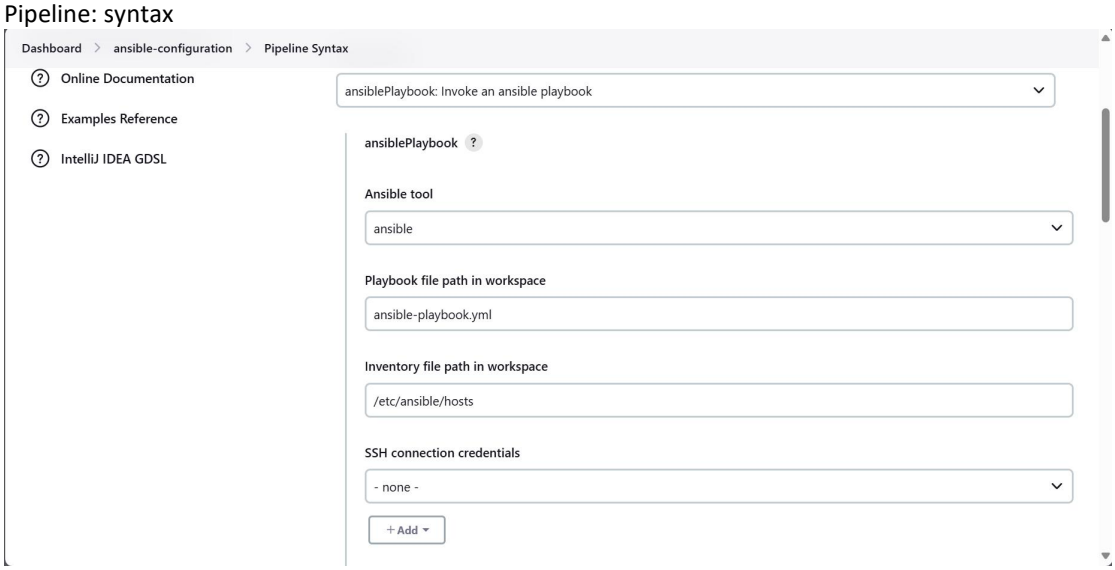
Tool

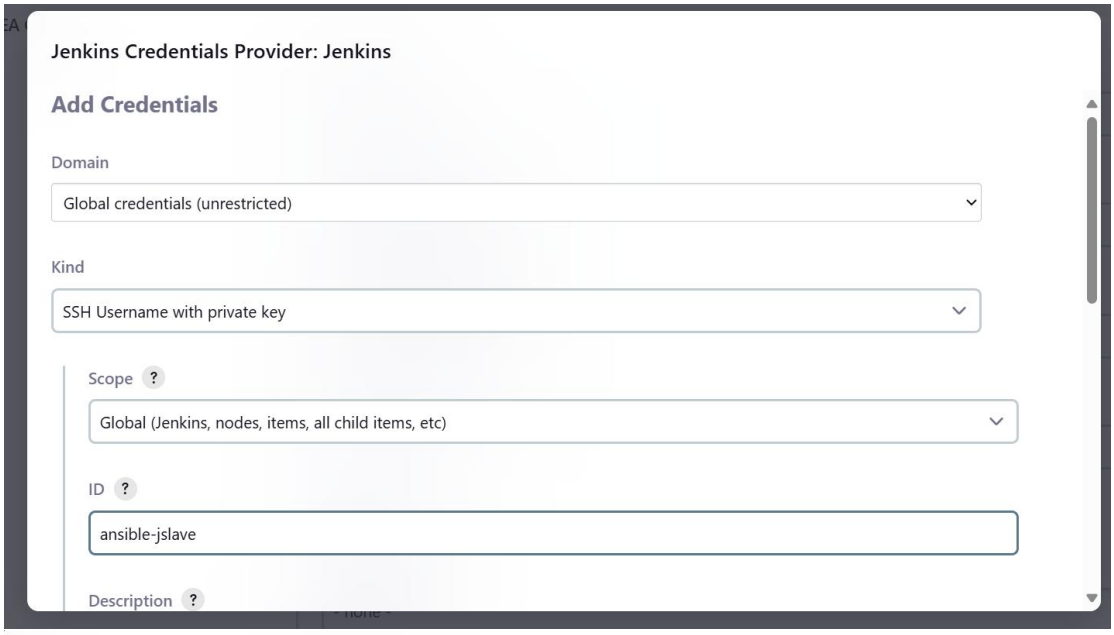


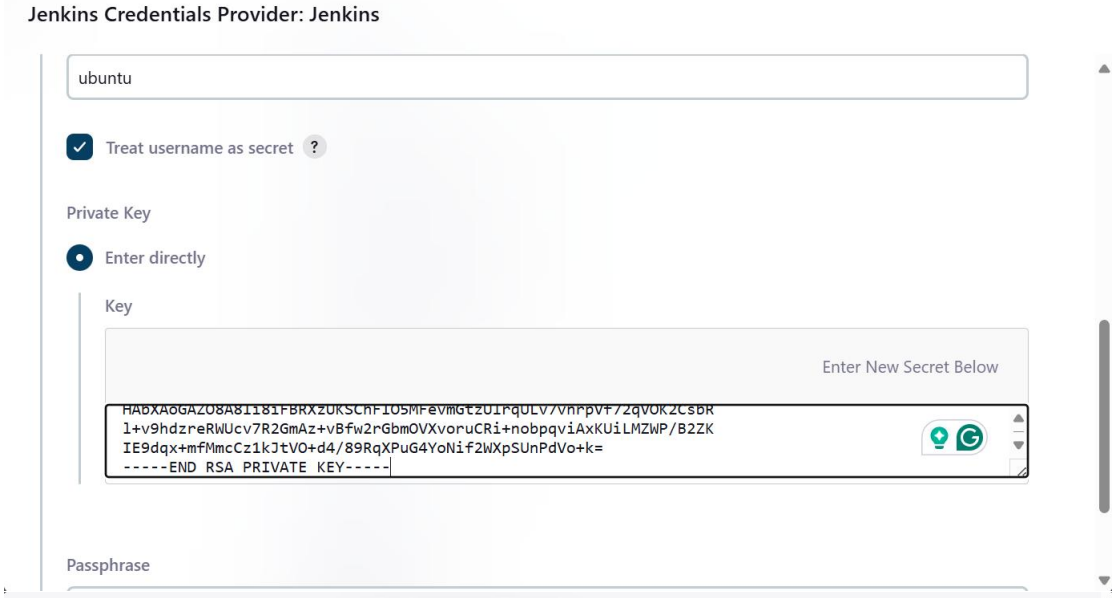
Pipeline script:

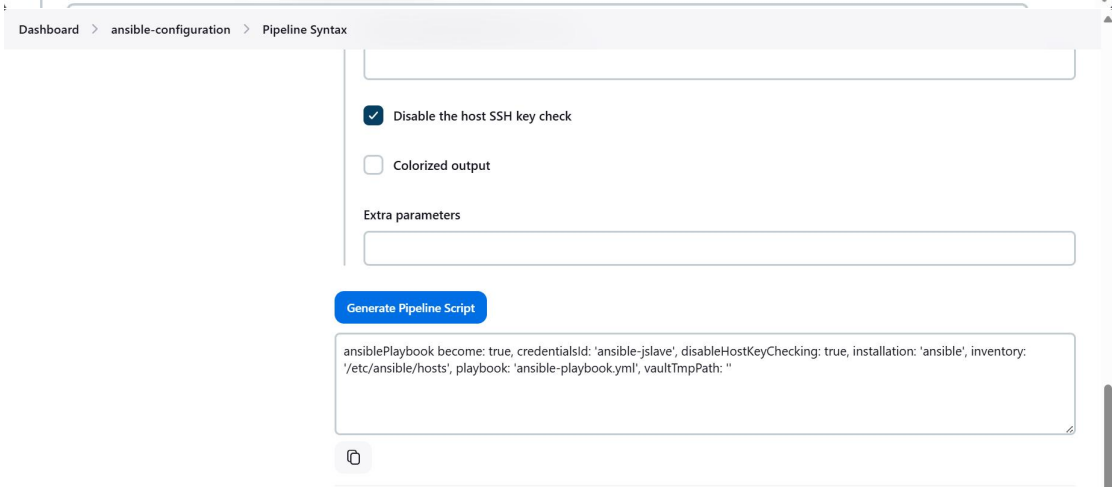


Pipeline: syntax




Jenkins Credentials Provider: Jenkins
Add Credentials
Domain
Global credentials (unrestricted)
Kind
SSH Username with private key
Scope
Global (Jenkins, nodes, items, all child items, etc)
ID
ansible-jslave
Description


Jenkins Credentials Provider: Jenkins
ubuntu
Treat username as secret
Private Key
Enter directly
Key
Enter New Secret Below
l+v9hdzreRWUcv7R2GmAz+vBfw2rGbmOVXvoruCRi+nobpqviAxKUiLMZWP/B2ZK
IE9dqx+mfMmcCz1kJtVO+d4/89RqXPuG4YoNif2WXpSUnPdVo+k=
-----END RSA PRIVATE KEY-----
Passphrase


Dashboard
ansible-configuration
Pipeline Syntax
Disable the host SSH key check
Colorized output
Extra parameters
Generate Pipeline Script
ansiblePlaybook become: true, credentialsId: 'ansible-jslave', disableHostKeyChecking: true, installation: 'ansible', inventory: '/etc/ansible/hosts', playbook: 'ansible-playbook.yml', vaultTmpPath: ''

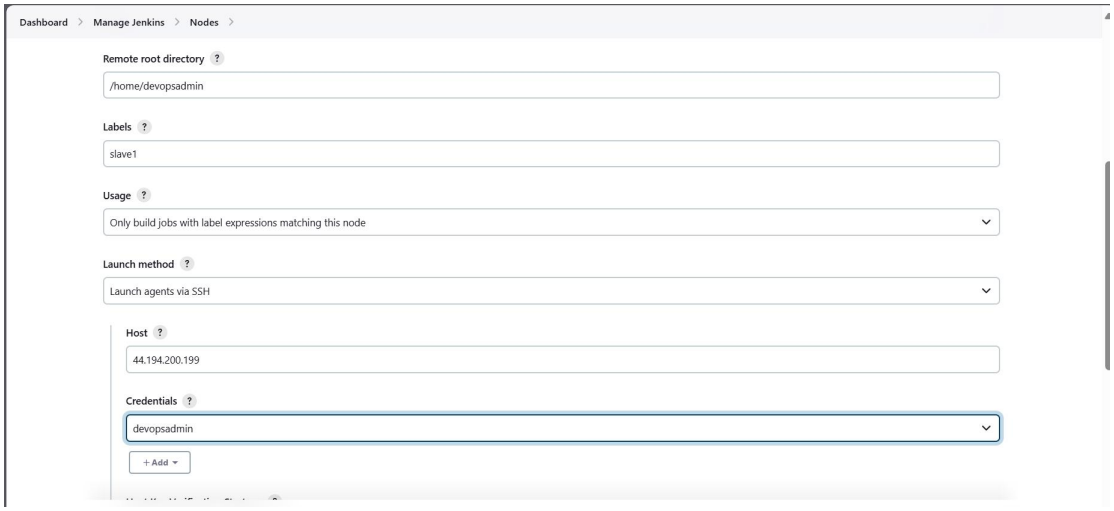
On Jslave server:

```
/var/lib/jenkins/workspace/ansible-configuration/ssh2743017449552478781.key -u ubuntu

PLAY [Configure Docker on EC2 Instances] ***************************************

TASK [Gathering Facts] *********************************************************
ok: [44.194.200.199]

TASK [updating apt] ************************************************************
changed: [44.194.200.199]

TASK [Install Java] ************************************************************
changed: [44.194.200.199]

TASK [Install Git] *************************************************************
changed: [44.194.200.199]

TASK [Install Git] *************************************************************
changed: [44.194.200.199]

TASK [Install Docker] **********************************************************
changed: [44.194.200.199]

TASK [Start Docker Service] ****************************************************
changed: [44.194.200.199]

PLAY RECAP *********************************************************************
44.194.200.199             : ok=7    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[Pipeline] }
[Pipeline] // stage
[Pipeline] }
```

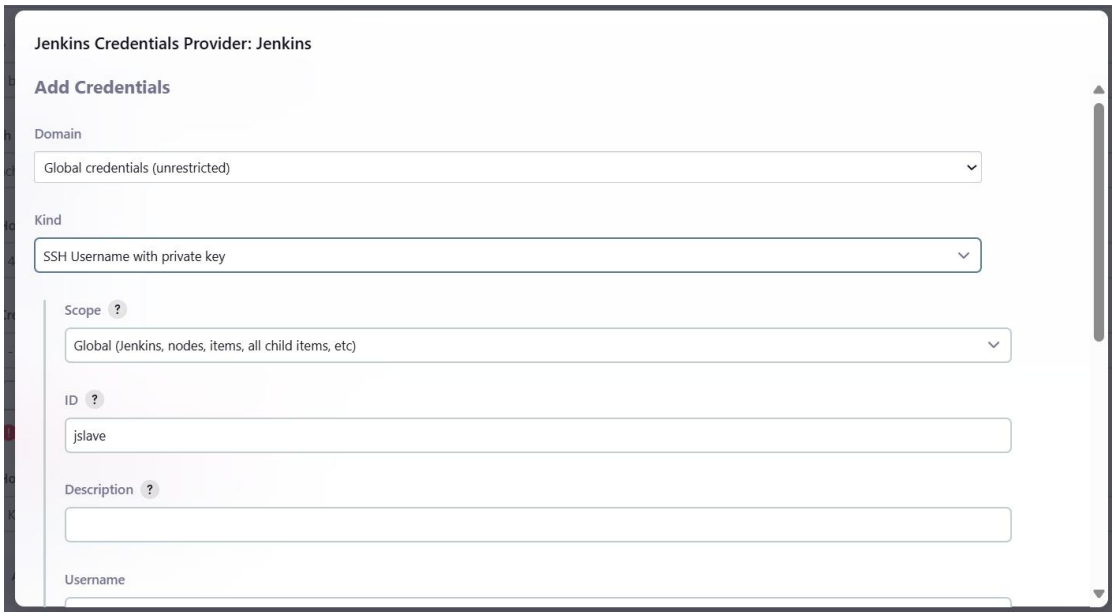
```
root@ip-10-0-1-10:~# docker --version
Command 'docker' not found, but can be installed with:
snap install docker          # version 24.0.5, or
apt  install docker.io       # version 24.0.7-0ubuntu2~22.04.1
apt  install podman-docker   # version 3.4.4+ds1-1ubuntu1.22.04.2
See 'snap info docker' for additional versions.
root@ip-10-0-1-10:~# docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu2~22.04.1
root@ip-10-0-1-10:~#
```

Integration Jenkins master Jenkins slave:

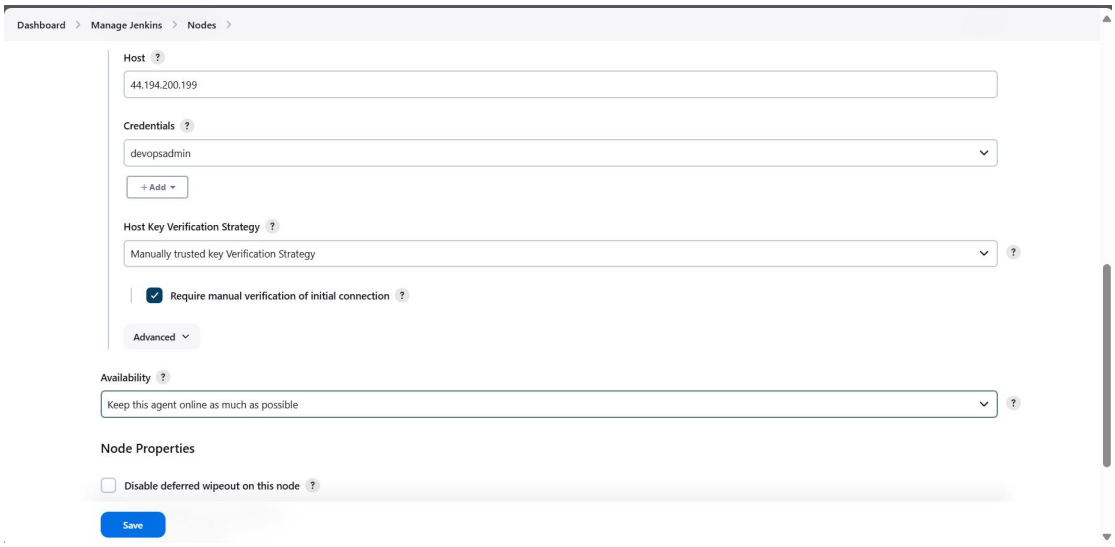
- **On Jenkins Server**



Ssh credentials:



Save



Agent Connected

```
[07/13/24 12:23:21] [SSH] Checking java version of java
[07/13/24 12:23:21] [SSH] java -version returned 17.0.11.
[07/13/24 12:23:21] [SSH] Starting sftp client.
[07/13/24 12:23:21] [SSH] Copying latest remoting.jar...
[07/13/24 12:23:21] [SSH] Copied 1,369,595 bytes.
Expanded the channel window size to 4MB
[07/13/24 12:23:21] [SSH] Starting agent process: cd "/home/devopsadmin" && java  -jar remoting.jar -workDir /home/devopsadmin -jar-cache
/home/devopsadmin/remoting/jarCache
Jul 13, 2024 12:23:21 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/devopsadmin/remoting as a remoting work directory
Jul 13, 2024 12:23:21 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/devopsadmin/remoting
<===[JENKINS REMOTING CAPACITY]===>channel started
Remoting version: 3206.vb_15dcf73f6a_9
Launcher: SSHLauncher
Communication Protocol: Standard in/out
This is a Unix agent
Agent successfully connected and online
```

Writing pipeline script for app build, containerized and deployment :

CICD :: git/jenkins/docker/kubernetes

==> CICD Pipeline - to automate the build and deployment.

1. Developers Create the Source Code

2. Commit the Source_Code to Source_Code Repository(Github)

3. Jenkins_pipeline ---> for Monolith Application Architecture

- SCM_Checkout - Download the source_code to build server

- Application_Build - Process of compiling the source code and create artifacts(Binaries - *.war/*.jar)

- Deploy the artifacts to Target Server(QA/UAT/PROD)

3.1 Jenkins Pipeline ---> for Containerized - Micro-Service Based Application Architecture

- SCM_Checkout

- Application_Build using Maven

- Application Image Build using Docker

- Published to Container Registry(Dockerhub)

- Deploy the Container Image in the Target Server and run the Application using Container.

Stage 1: "SCM checkout"

Script ?

```
pipeline {
    agent {label 'slave1'}

    stages {
        stage('SCM_Checkout') {
            steps {
                git 'https://github.com/maulik2311/capstone-health.git'
            }
        }
    }
}
```

Console Output

```
Started by user Maulik DEVANI
[Pipeline] Start of Pipeline
[Pipeline] node
Running on slave1 in /home/devopsadmin/workspace/Health_care
[Pipeline] {
[Pipeline] stage
[Pipeline] { (SCM_Checkout)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/maulik2311/capstone-health.git
 > git init /home/devopsadmin/workspace/Health_care # timeout=10
```

```
devopsadmin@ip-10-0-1-10:~/workspace/Health_care$ pwd
/home/devopsadmin/workspace/Health_care
devopsadmin@ip-10-0-1-10:~/workspace/Health_care$ ls
Dockerfile  ansible-playbook.yml  k8sdeploy.yaml  mvnw  mvnw.cmd  pom.xml  src
devopsadmin@ip-10-0-1-10:~/workspace/Health_care$
```

Stage-2; "Application Build"

Script ?

```
pipeline {
    agent {label 'slave1'}

    stages {
        stage('SCM_Checkout') {
            steps {
                git 'https://github.com/maulik2311/capstone-health.git'
            }
        }
        stage('App_Build_mvn') {
            steps {
                sh 'mvn clean package'
            }
        }
    }
}
```

```
completed.
[⍰[1;34mINFO⍰[m]
[⍰[1;34mINFO⍰[m] Results:
[⍰[1;34mINFO⍰[m]
[⍰[1;34mINFO⍰[m] ⍰[1;32mTests run: 3, Failures: 0, Errors: 0, Skipped: 0⍰[m
[⍰[1;34mINFO⍰[m]
[⍰[1;34mINFO⍰[m]
[⍰[1;34mINFO⍰[m] ⍰[1m--- ⍰[0;32mmaven-jar-plugin:3.2.2:jar⍰[m ⍰[1m(default-jar)⍰[m @ ⍰[36mmedicure⍰[0;1m ---⍰[m
[⍰[1;34mINFO⍰[m] Building jar: /home/devopsadmin/workspace/Health_care/target/medicure-0.0.1-SNAPSHOT.jar
[⍰[1;34mINFO⍰[m]
[⍰[1;34mINFO⍰[m] ⍰[1m--- ⍰[0;32mspring-boot-maven-plugin:2.7.4:repackage⍰[m ⍰[1m(repackage)⍰[m @ ⍰[36mmedicure⍰[0;1m ---⍰[m
[⍰[1;34mINFO⍰[m] Replacing main artifact with repackaged archive
[⍰[1;34mINFO⍰[m] ⍰[1m------------------------------------------------------------------------⍰[m
[⍰[1;34mINFO⍰[m] ⍰[1;32mBUILD SUCCESS⍰[m
[⍰[1;34mINFO⍰[m] ⍰[1m------------------------------------------------------------------------⍰[m
[⍰[1;34mINFO⍰[m] Total time:  21.863 s
[⍰[1;34mINFO⍰[m] Finished at: 2024-07-13T12:38:44Z
[⍰[1;34mINFO⍰[m] ⍰[1m------------------------------------------------------------------------⍰[m
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

```
devopsadmin@ip-10-0-1-10:~/workspace/Health_care/target$ ls
classes                 generated-test-sources  maven-status                        medicure-0.0.1-SNAPSHOT.jar.original  test-classes
generated-sources       maven-archiver          medicure-0.0.1-SNAPSHOT.jar         surefire-reports
```

Stage-3: "Application image build-Docker"

Dockerfile:

```
FROM openjdk:11
ARG JAR_FILE=target/*.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

Script ?

```
pipeline {
    agent {label 'slave1'}

    stages {
        stage('SCM_Checkout') {
            steps {
                git 'https://github.com/maulik2311/capstone-health.git'
            }
        }
        stage('App_Build_mvn') {
            steps {
                sh 'mvn clean package'
            }
        }
        stage('Docker_Image_build') {
            steps {
                sh "docker build -t maulikd2397/health:${BUILD_NUMBER} ."
                sh "docker tag maulikd2397/health:${BUILD_NUMBER} maulikd2397/health:latest "
                sh 'docker images'
            }
        }
    }
}
```

Image build failed:

```
+ docker build -t maulikd2397/health:7 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post
"http://%2Fvar%2Frun%2Fdocker.sock/v1.24/build?
buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Dockerfi
le&labels=%7B%7D&memory=0&memswap=0&networkmode=default&rm=1&shmsize=0&t=maulikd2397%2Fhealth%3A7&target=&ulimits=null&version
=1": dial unix /var/run/docker.sock: connect: permission denied
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
```

Reason:

```
docker:x:122:
devopsadmin:x:1001:
devopsadmin@ip-10-0-1-10:~/workspace/Health_care/target$
```

Add user to docker group

```
root@ip-10-0-1-10:~# usermod -aG docker devopsadmin
```

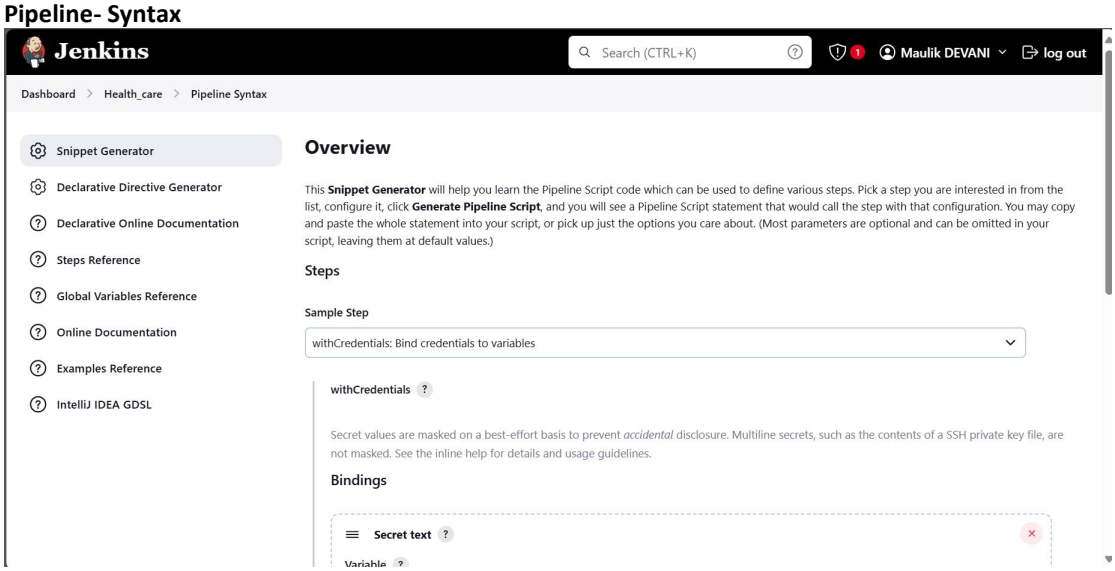
```
docker:x:122:devopsadmin
devopsadmin:x:1001:
```

Build Success:

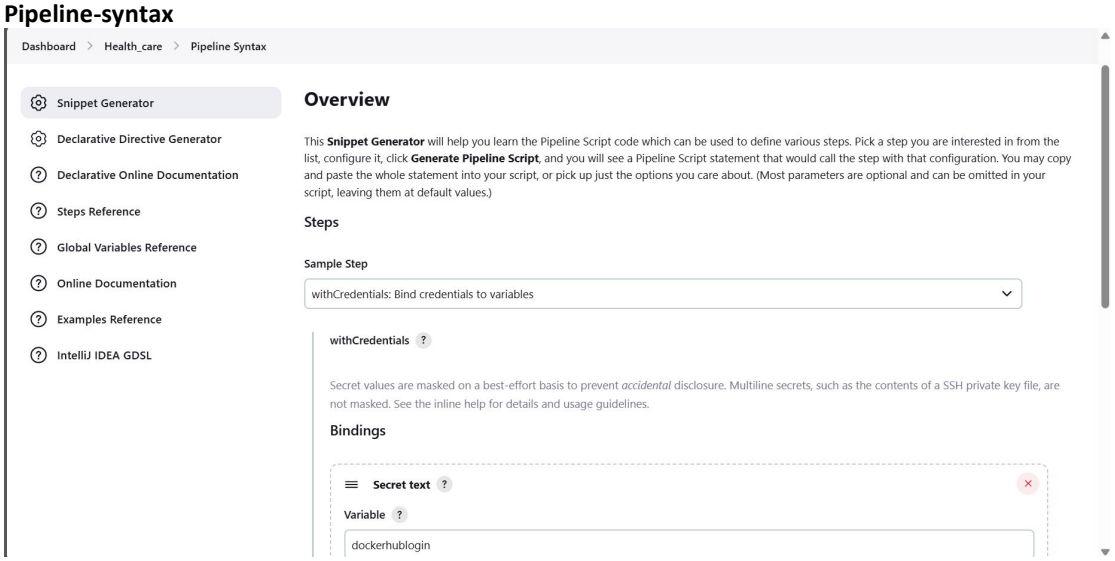
```
Successfully tagged maulikd2397/health:11
[Pipeline] sh
+ docker tag maulikd2397/health:11 maulikd2397/health:latest
[Pipeline] sh
+ docker images
REPOSITORY           TAG       IMAGE ID       CREATED         SIZE
maulikd2397/health   11        9f85a618b030   1 second ago    695MB
maulikd2397/health   latest    9f85a618b030   1 second ago    695MB
openjdk              11        47a932d998b7   23 months ago   654MB
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Stage-4: Login2Dockerhub

Pipeline- Syntax

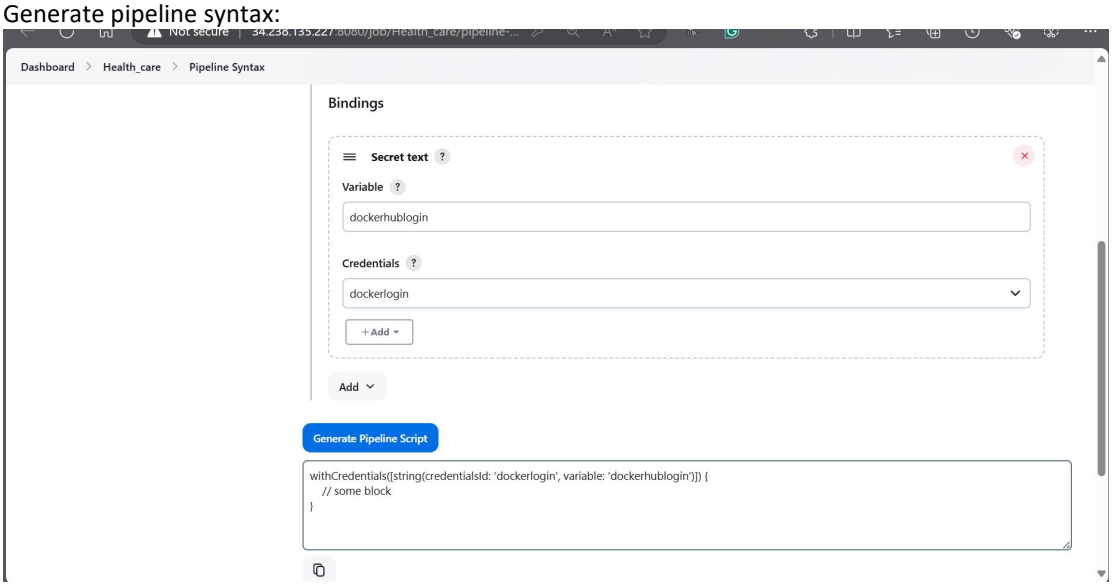


Pipeline-syntax



Credentials: Secret=Dockerhub password

Generate pipeline syntax:



Copy to pipeline script:

Script ?

```
pipeline {
    agent {label 'slave1'}

    stages {
        stage('SCM_Checkout') {
            steps {
                git 'https://github.com/maulik2311/capstone-health.git'
            }
        }
        stage('App_Build_mvn') {
            steps {
                sh 'mvn clean package'
            }
        }
        stage('Docker_Image_build') {
            steps {
                sh "docker build -t maulikd2397/health:${BUILD_NUMBER} ."
                sh "docker tag maulikd2397/health:${BUILD_NUMBER} maulikd2397/health:latest "
                sh 'docker images'
            }
        }
        stage('Login2Dockerhub') {
            steps {
                withCredentials([string(credentialsId: 'dockerlogin', variable: 'dockerhublogin')]) {
                sh "docker login -u maulikd2397 -p ${dockerhublogin}"
                }
            }
        }
    }
}
```

```
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Stage-5: "Push2Dockerhub"

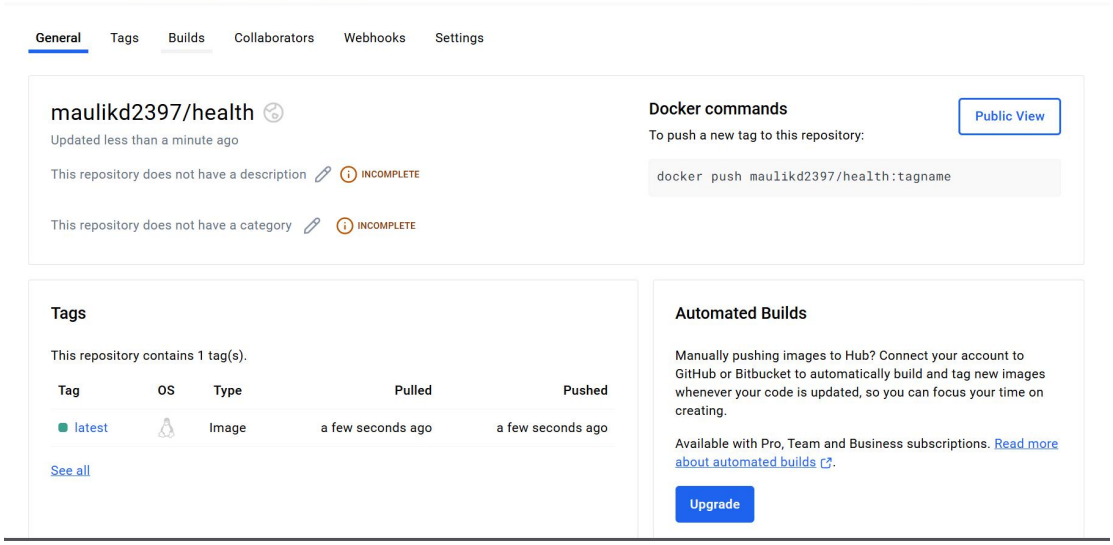
Script ?

```
            sh 'docker images'
        }
    }
    stage('Login2Dockerhub') {
        steps {
            withCredentials([string(credentialsId: 'dockerlogin', variable: 'dockerhublogin')]) {
            sh "docker login -u maulikd2397 -p ${dockerhublogin}"
            }
        }
    }
    stage('Push2Dockerhub') {
        steps {
            sh 'docker push maulikd2397/health:latest '
        }
    }
}
}
```

Pushed

```
9db07f618750: Pushed
latest: digest: sha256:5b4bd101880fe39d062272d927e0c9afbd495b1ab1a91799b9ad9020a2c7e631 size: 2007
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Dockerhub:



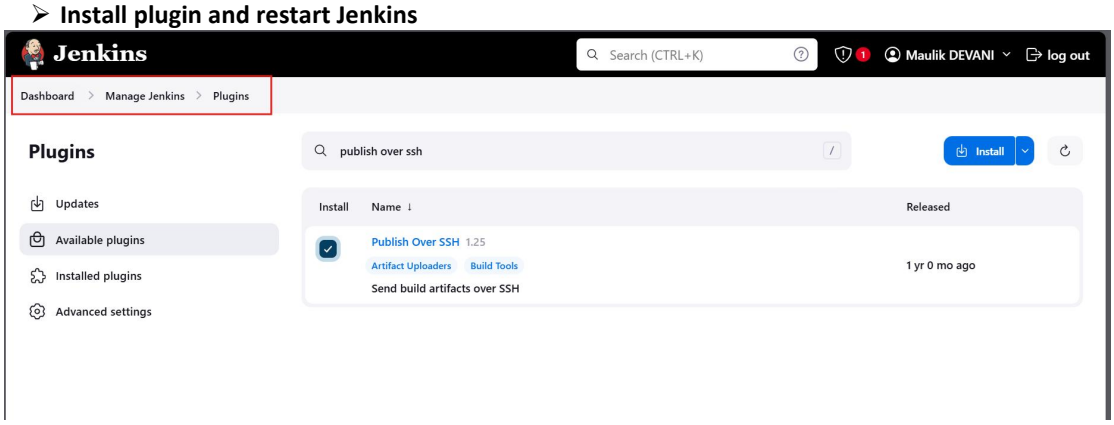
Stage-6: "k8sdeployment"

For Automated Deployments on k8s cluster, k8s master should be integrate with Jenkins master:

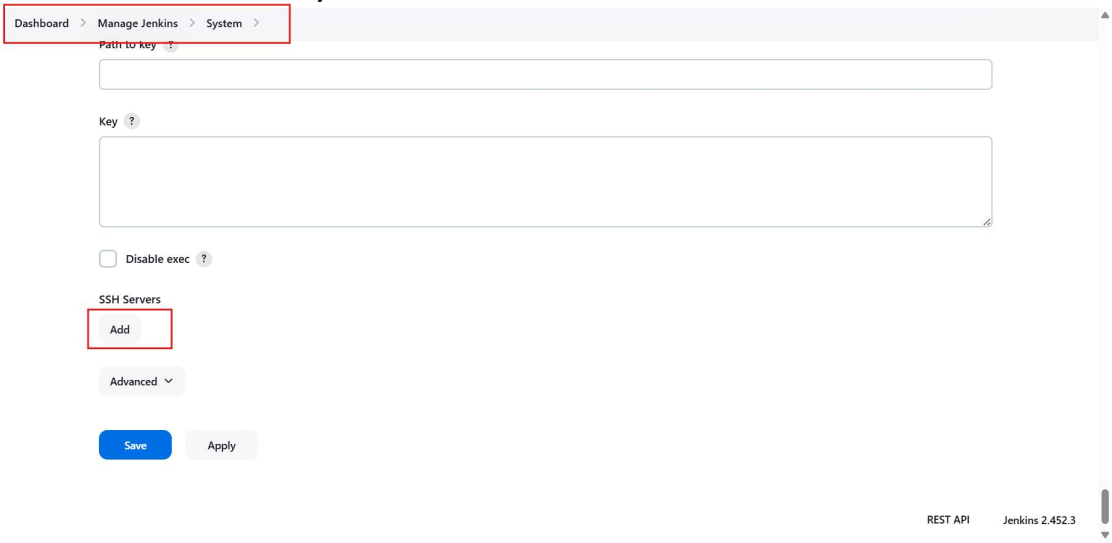
Steps:
ssh connection.
Kubernetes cluster already configured with user and user .kube configuration

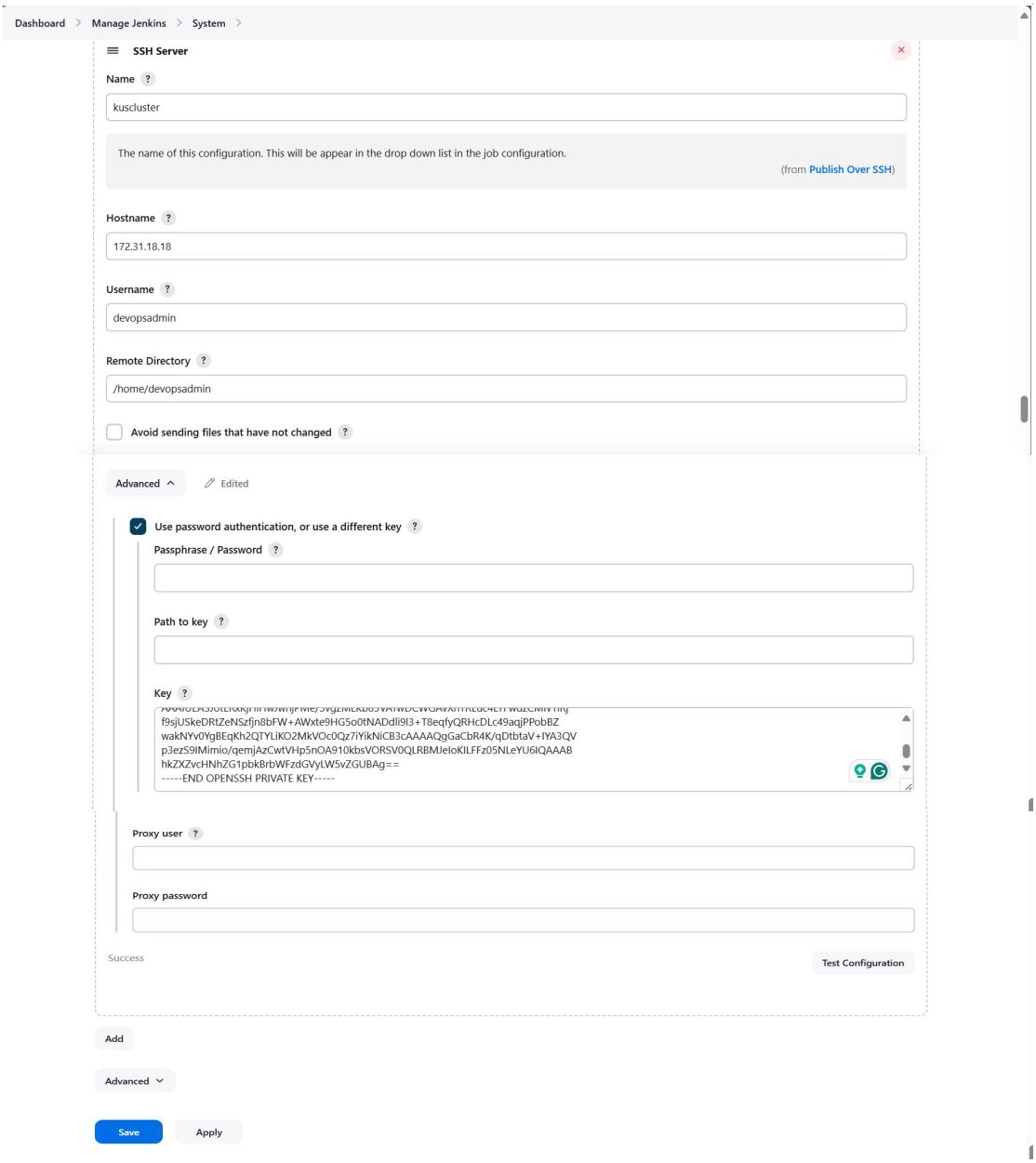
```
devopsadmin@kmaster-node:~/.ssh$ kubectl get node
NAME           STATUS   ROLES           AGE   VERSION
kmaster-node   Ready    control-plane   24d   v1.29.6
worker-node1   Ready    <none>          24d   v1.29.6
worker-node2   Ready    <none>          24d   v1.29.6
devopsadmin@kmaster-node:~/.ssh$ ls
authorized_keys  id_ecdsa  id_ecdsa.pub
devopsadmin@kmaster-node:~/.ssh$ ls
authorized_keys  id_ecdsa  id_ecdsa.pub
devopsadmin@kmaster-node:~/.ssh$
```

➢ **Install plugin and restart Jenkins**



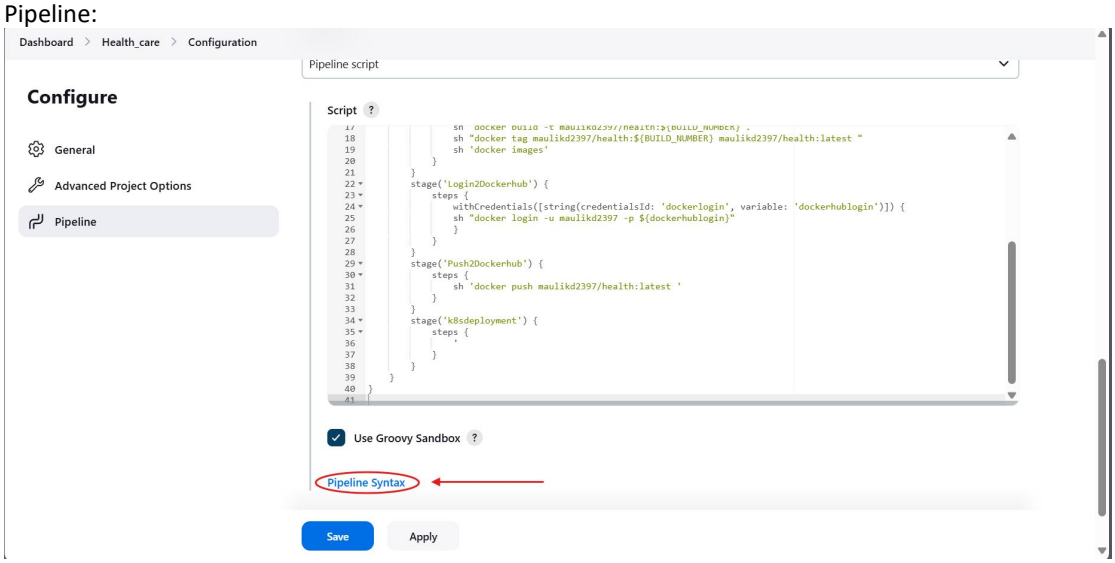
➢ **Add ssh servers from System.**



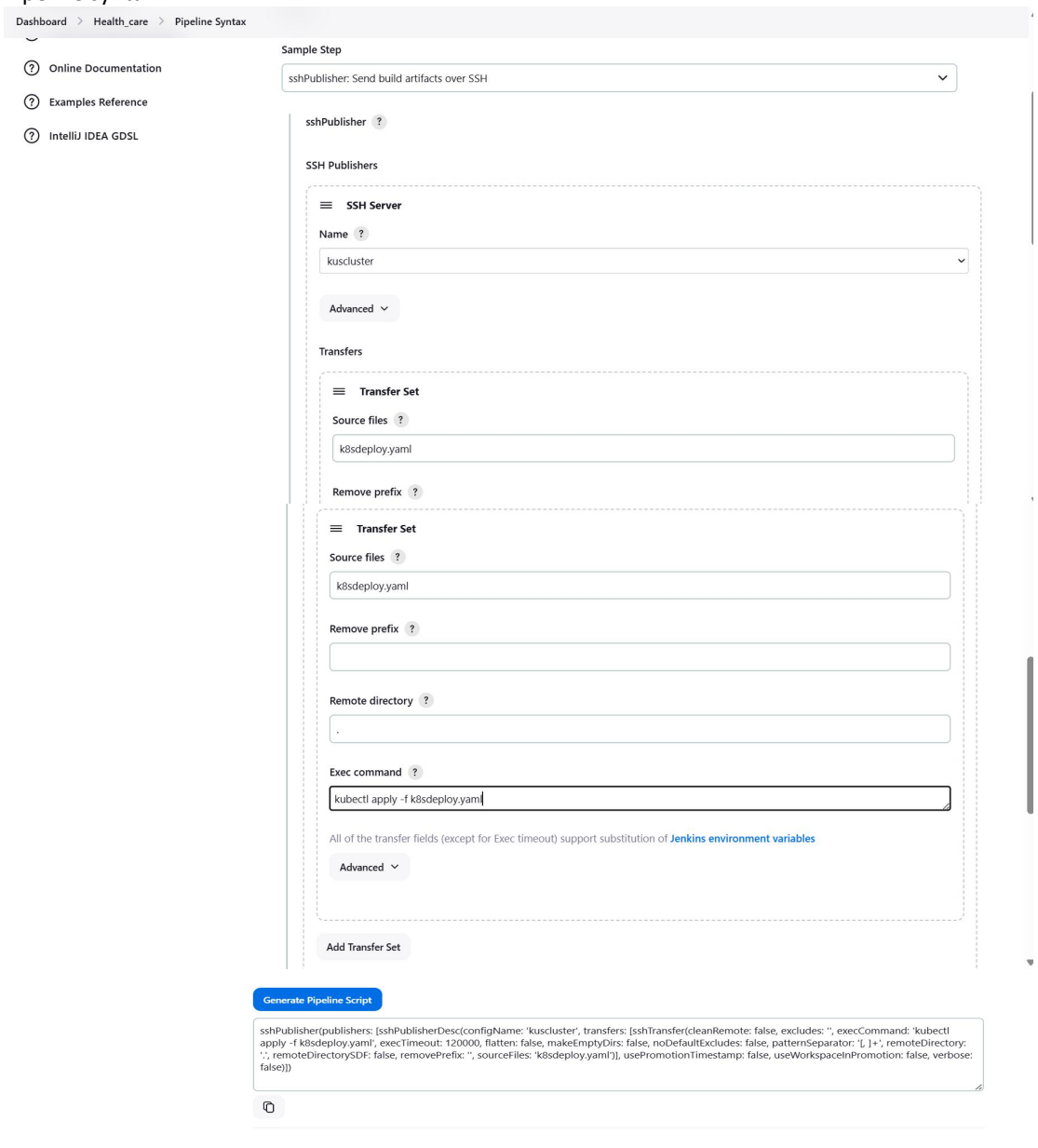


Test= Success

Pipeline:



Pipeline syntax:



Pipeline:
Copy pipeline syntax and tun the job.

Script ?

```
        sh 'mvn clean package'
        }
    }
    stage('Docker_Image_build') {
        steps {
            sh "docker build -t maulikd2397/health:${BUILD_NUMBER} ."
            sh "docker tag maulikd2397/health:${BUILD_NUMBER} maulikd2397/health:latest "
            sh 'docker images'
        }
    }
    stage('Login2Dockerhub') {
        steps {
            withCredentials([string(credentialsId: 'dockerlogin', variable: 'dockerhublogin')]) {
            sh "docker login -u maulikd2397 -p ${dockerhublogin}"
            }
        }
    }
    stage('Push2Dockerhub') {
        steps {
            sh 'docker push maulikd2397/health:latest '
        }
    }
    stage('k8sdeployment') {
        steps {
            sshPublisher(publishers: [sshPublisherDesc(configName: 'kuscluster', transfers: [sshTransfer(cleanRemote: fa
        }
    }
}
```

Deployment file:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: health-deployment
  labels:
    app: app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: app
  template:
    metadata:
      labels:
        app: app
    spec:
      containers:
        - name: c001
          image: maulikd2397/health
          ports:
            - containerPort: 8082


---

apiVersion: v1
kind: Service
metadata:
  name: health-service
  labels:
    app: app
spec:
  selector:
    app: app
  type: NodePort
  ports:
    - nodePort: 31231
      port: 80
      targetPort: 8082
```

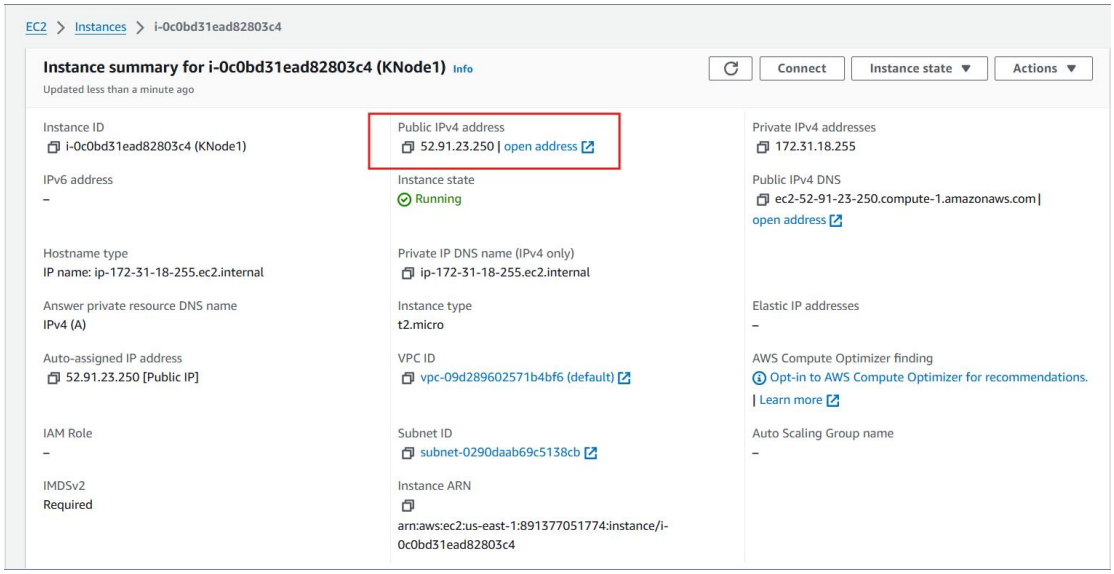
Deployment success

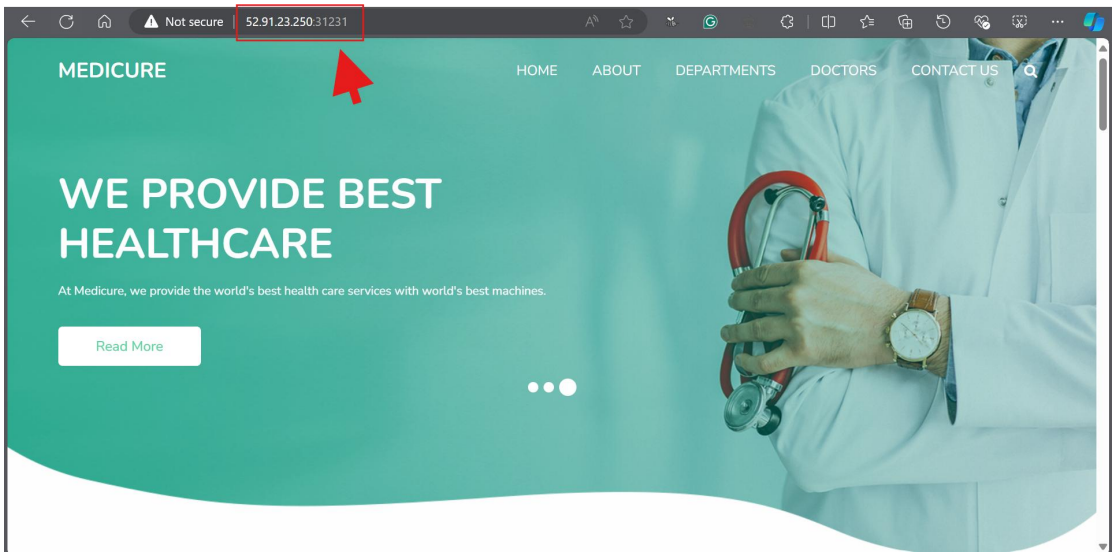
```
devopsadmin@kmaster-node:~/.ssh$ kubectl get all
NAME                                    READY   STATUS    RESTARTS   AGE
pod/health-deployment-677f4db968-hdv2g   1/1     Running   0          41s
pod/health-deployment-677f4db968-rfh45   1/1     Running   0          41s
pod/health-deployment-677f4db968-t85v9   1/1     Running   0          41s

NAME                              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/finance-svc-nodeport      NodePort    10.111.112.91   <none>        8081:31434/TCP   5d1h
service/health-service            NodePort    10.97.6.71      <none>        80:31231/TCP     41s
service/kubernetes                ClusterIP   10.96.0.1       <none>        443/TCP          24d

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/health-deployment   3/3     3            3           41s

NAME                                          DESIRED   CURRENT   READY   AGE
replicaset.apps/health-deployment-677f4db968   3         3         3       41s
```

Access the application.

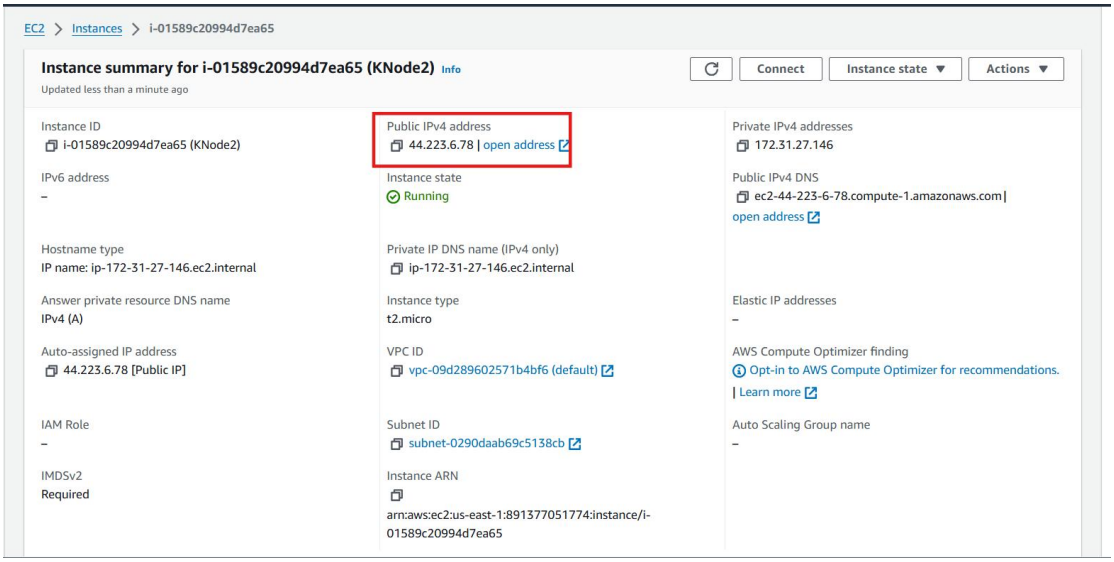
```
devopsadmin@kmaster-node:~/.ssh$ kubectl get pods -o wide
NAME                                  READY   STATUS    RESTARTS   AGE     IP            NODE           NOMINATED NODE   READINESS GATES
health-deployment-677f4db968-hdv2g    1/1     Running   0          2m16s   10.244.1.87   worker-node1   <none>           <none>
health-deployment-677f4db968-rfh45    1/1     Running   0          2m16s   10.244.2.89   worker-node2   <none>           <none>
health-deployment-677f4db968-t85v9    1/1     Running   0          2m16s   10.244.2.88   worker-node2   <none>           <none>
devopsadmin@kmaster-node:~/.ssh$
```

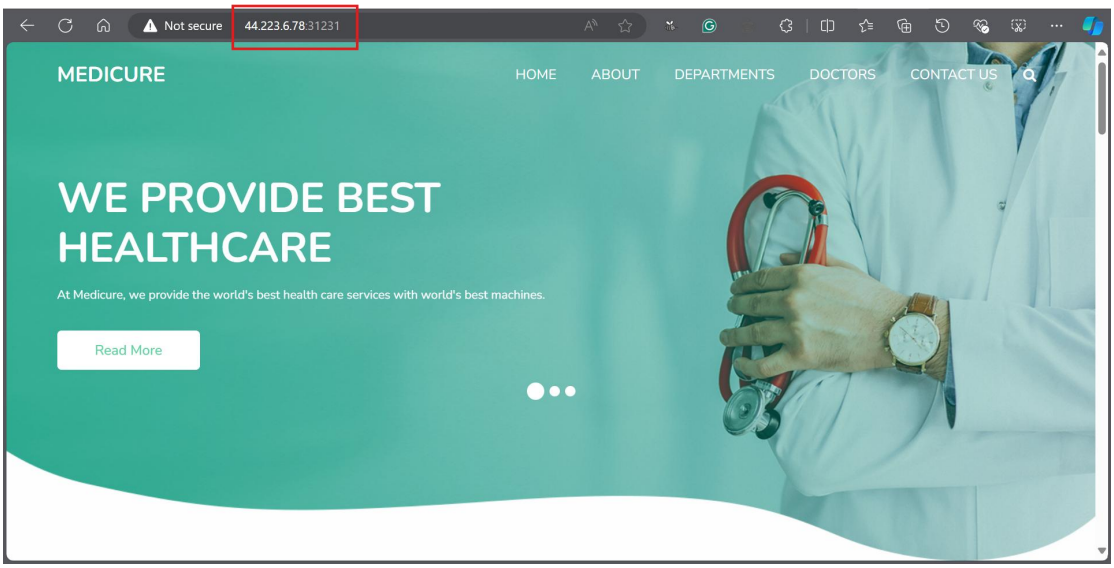
Worker-Node1





Worker Node-2





High availability

Adding server to the target of prometheus: Node Exporter

```
root@ip-172-31-46-14:~/prometheus-2.38.0.linux-amd64# ls
LICENSE  NOTICE  console_libraries  consoles  prometheus  prometheus.yml  promtool
```

```
# 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"]
      - targets: ["172.31.18.255:9100"]
~
~
~
-- INSERT --                                                                    30,40         All
```

Installing NodeExporter on target server:

```
root@worker-node1:~# wget https://github.com/prometheus/node_exporter/releases/download/v1.4.0
rc.0/^Cde_exporter-1.4.0-rc.0.linux-amd64.tar.gz
root@worker-node1:~# wget https://github.com/prometheus/node_exporter/releases/download/v1.8.1/node_exporter-1.8.1.linux-amd64.tar.gz
--2024-07-13 16:06:28--  https://github.com/prometheus/node_exporter/releases/download/v1.8.1/node_exporter-1.8.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/e07e4ee4-e4b0-48dc-9c04-eaad890c81b3?X-Amz-Algorithm=AWS4-
HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20240713%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240713T160628Z&X-Amz-Expires=300&X-Amz-Signat
ure=e78a708085751f110b16760b02393201ed39cd426f9960360e8a9576b71bada8&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=9524057&response-content-dispos
ition=attachment%3B%20filename%3Dnode_exporter-1.8.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2024-07-13 16:06:28--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/e07e4ee4-e4b0-48dc-9c04-eaad890c81b3?X-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20240713%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240713T160628Z&X-Amz-Expires=3
00&X-Amz-Signature=e78a708085751f110b16760b02393201ed39cd426f9960360e8a9576b71bada8&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=9524057&response
-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.8.1.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: 10672684 (10M) [application/octet-stream]
Saving to: 'node_exporter-1.8.1.linux-amd64.tar.gz'

node_exporter-1.8.1.linux-amd64.tar.gz 100%[=========================================================================>]  10.18M  --.-KB/s    in 0.07s

2024-07-13 16:06:28 (152 MB/s) - 'node_exporter-1.8.1.linux-amd64.tar.gz' saved [10672684/10672684]

root@worker-node1:~# ls
node_exporter-1.8.1.linux-amd64.tar.gz  snap
root@worker-node1:~# tar -xvf node_exporter-1.8.1.linux-amd64.tar.gz
node_exporter-1.8.1.linux-amd64/
node_exporter-1.8.1.linux-amd64/NOTICE
node_exporter-1.8.1.linux-amd64/LICENSE
node_exporter-1.8.1.linux-amd64/node_exporter
```

Adding service file and start node exporter:

vi /etc/systemd/system/node_exporter.service

[Unit]
Description=Prometheus Server
Documentation=https://prometheus.io/docs/introduction/overview/
After=network-online.target

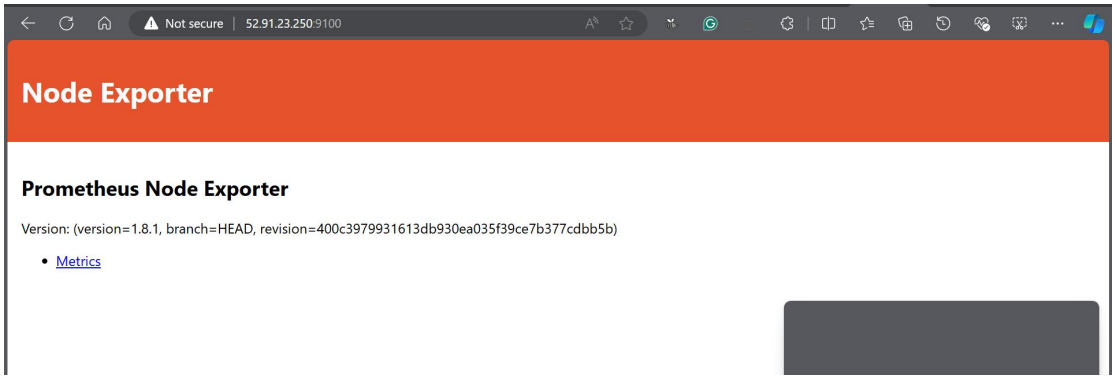
[Service]
User=root
Restart=on-failure

ExecStart=/root/node_exporter-1.8.1.linux-amd64/node_exporter

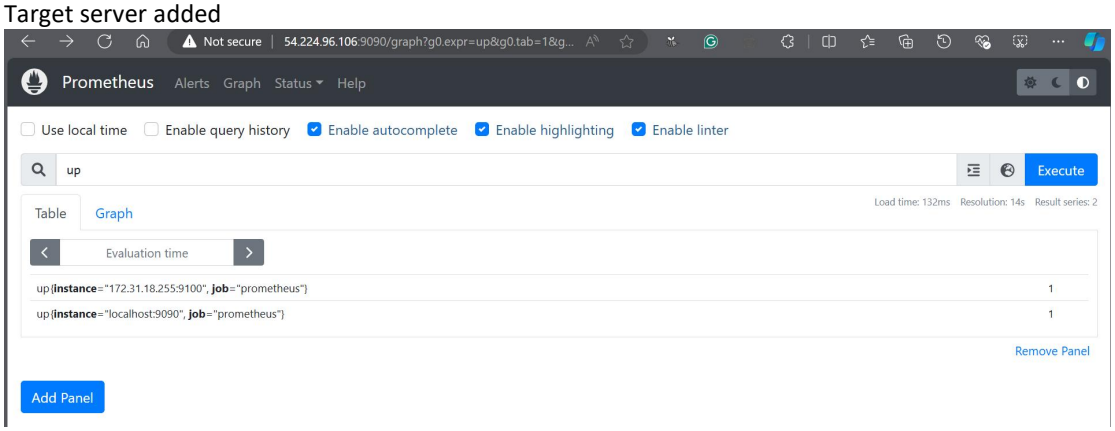
[Install]
WantedBy=multi-user.target

Restart:

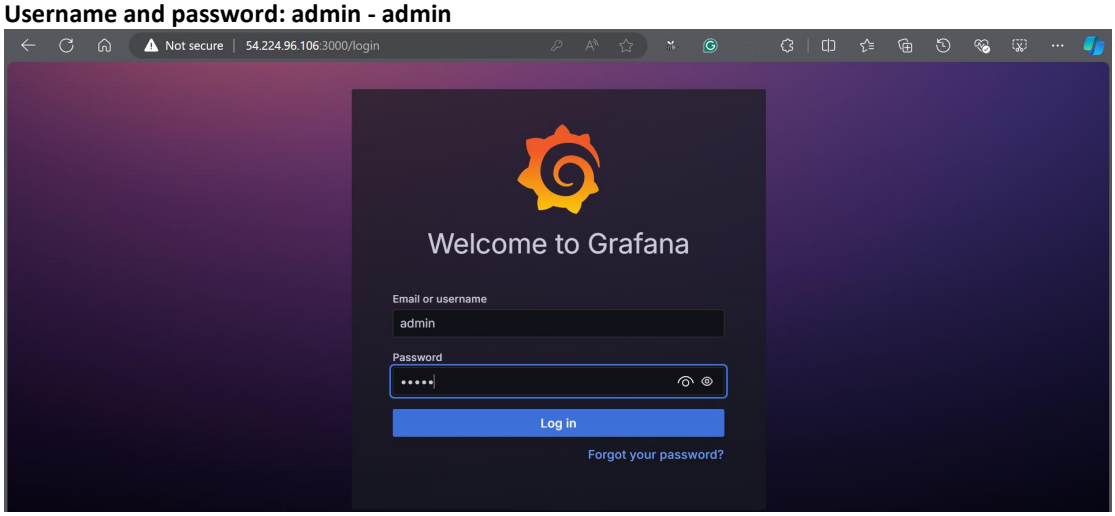
```
root@worker-node1:~/node_exporter-1.8.1.linux-amd64# vi /etc/systemd/system/node_exporter.service
root@worker-node1:~/node_exporter-1.8.1.linux-amd64# systemctl daemon-reload
root@worker-node1:~/node_exporter-1.8.1.linux-amd64# systemctl start node_exporter
root@worker-node1:~/node_exporter-1.8.1.linux-amd64# systemctl status node_exporter
● node_exporter.service - Prometheus Server
     Loaded: loaded (/etc/systemd/system/node_exporter.service; disabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-07-13 16:10:25 UTC; 5s ago
       Docs: https://prometheus.io/docs/introduction/overview/
   Main PID: 36403 (node_exporter)
      Tasks: 3 (limit: 1120)
     Memory: 2.0M
        CPU: 7ms
     CGroup: /system.slice/node_exporter.service
             └─36403 /root/node_exporter-1.8.1.linux-amd64/node_exporter

Jul 13 16:10:25 worker-node1 node_exporter[36403]: ts=2024-07-13T16:10:25.129Z caller=node_exporter.go:118 level=info collector=time
Jul 13 16:10:25 worker-node1 node_exporter[36403]: ts=2024-07-13T16:10:25.129Z caller=node_exporter.go:118 level=info collector=timex
Jul 13 16:10:25 worker-node1 node_exporter[36403]: ts=2024-07-13T16:10:25.129Z caller=node_exporter.go:118 level=info collector=udp_queues
Jul 13 16:10:25 worker-node1 node_exporter[36403]: ts=2024-07-13T16:10:25.129Z caller=node_exporter.go:118 level=info collector=uname
Jul 13 16:10:25 worker-node1 node_exporter[36403]: ts=2024-07-13T16:10:25.129Z caller=node_exporter.go:118 level=info collector=vmstat
Jul 13 16:10:25 worker-node1 node_exporter[36403]: ts=2024-07-13T16:10:25.129Z caller=node_exporter.go:118 level=info collector=watchdog
Jul 13 16:10:25 worker-node1 node_exporter[36403]: ts=2024-07-13T16:10:25.129Z caller=node_exporter.go:118 level=info collector=xfs
Jul 13 16:10:25 worker-node1 node_exporter[36403]: ts=2024-07-13T16:10:25.129Z caller=node_exporter.go:118 level=info collector=zfs
Jul 13 16:10:25 worker-node1 node_exporter[36403]: ts=2024-07-13T16:10:25.130Z caller=tls_config.go:313 level=info msg="Listening on" address=[::]:9100
Jul 13 16:10:25 worker-node1 node_exporter[36403]: ts=2024-07-13T16:10:25.130Z caller=tls_config.go:316 level=info msg="TLS is disabled." http2=false ad
lines 1-21/21 (END)
```



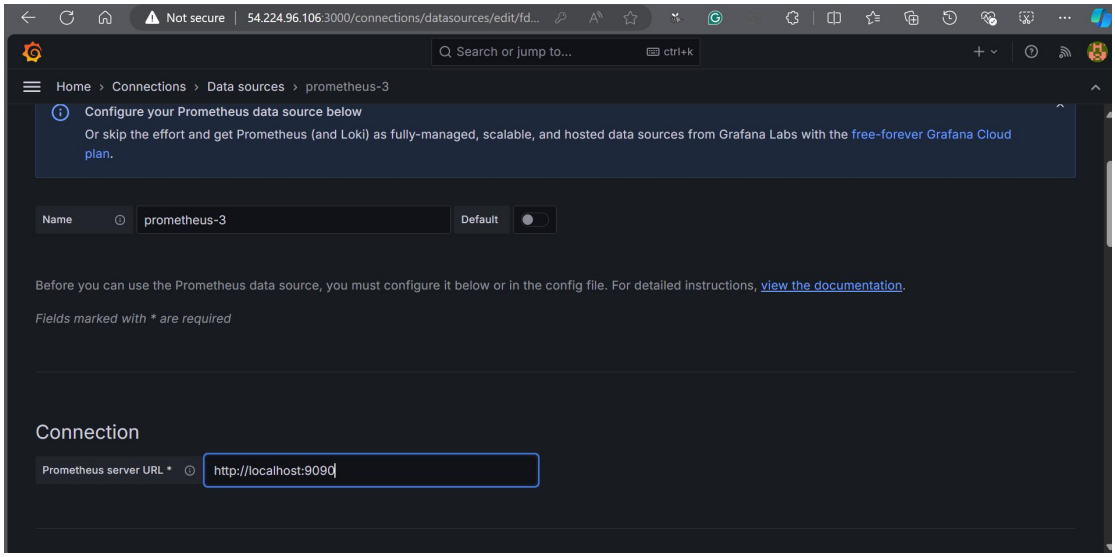
Target server added



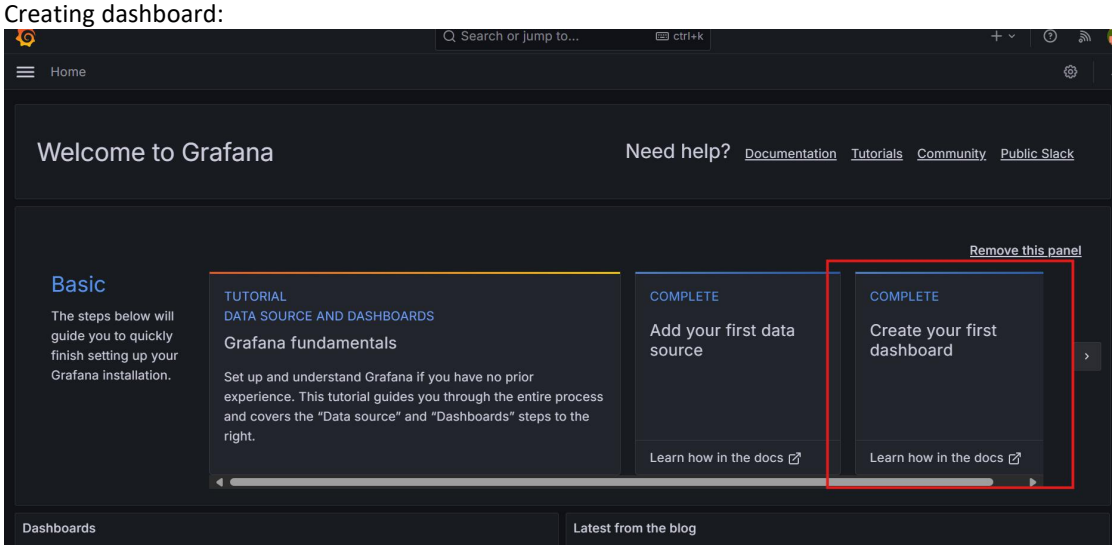
Login to Grafana:
Username and password: admin - admin



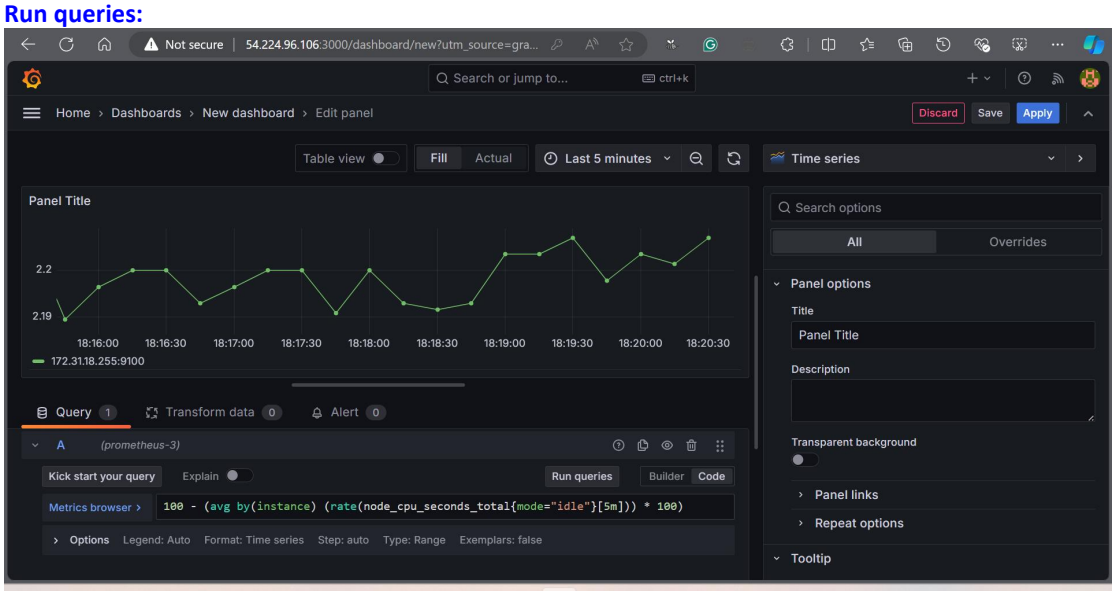
Prometheus and Grafana run on the same server:

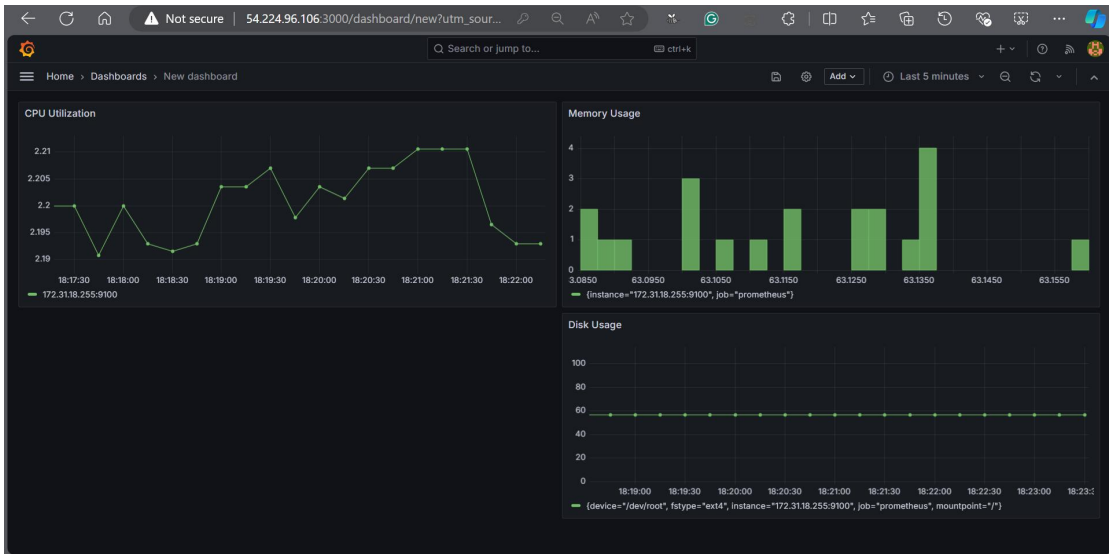


Creating dashboard:



Run queries:


Not secure | 54.224.96.106:3000/dashboard/new?utm_source=gra...
Search or jump to... ctrl+k
Home › Dashboards › New dashboard › Edit panel
Discard
Save
Apply
Table view
Fill
Actual
Last 5 minutes
Time series
Panel Title
2.2
2.19
18:16:00
18:16:30
18:17:00
18:17:30
18:18:00
18:18:30
18:19:00
18:19:30
18:20:00
18:20:30
172.31.18.255:9100
Search options
All
Overrides
Panel options
Title
Panel Title
Description
Transparent background
Panel links
Repeat options
Tooltip
Query 1
Transform data 0
Alert 0
A
(prometheus-3)
Kick start your query
Explain
Run queries
Builder
Code
Metrics browser ›
100 - (avg by(instance) (rate(node_cpu_seconds_total{mode="idle"}[5m])) * 100)
Options
Legend: Auto
Format: Time series
Step: auto
Type: Range
Exemplars: false


Not secure | 54.224.96.106:3000/dashboard/new?utm_sour...
Search or jump to... ctrl+k
Home › Dashboards › New dashboard
Add
Last 5 minutes
CPU Utilization
2.21
2.205
2.2
2.195
2.19
18:17:30
18:18:00
18:18:30
18:19:00
18:19:30
18:20:00
18:20:30
18:21:00
18:21:30
18:22:00
172.31.18.255:9100
Memory Usage
4
3
2
1
0
3.0850
63.0950
63.1050
63.1150
63.1250
63.1350
63.1450
63.1550
{instance="172.31.18.255:9100", job="prometheus"}
Disk Usage
100
80
60
40
20
0
18:19:00
18:19:30
18:20:00
18:20:30
18:21:00
18:21:30
18:22:00
18:22:30
18:23:00
{device="/dev/root", fstype="ext4", instance="172.31.18.255:9100", job="prometheus", mountpoint="/"}

Jenkins pipeline script:

```
pipeline {
   agent {label 'slave1'}

   stages {
      stage('SCM_Checkout') {
         steps {
            git 'https://github.com/maulik2311/capstone-health.git'
         }
      }
      stage('App_Build_mvn') {
         steps {
            sh 'mvn clean package'
         }
      }
      stage('Docker_Image_build') {
         steps {
            sh "docker build -t maulikd2397/health:${BUILD_NUMBER} ."
            sh "docker tag maulikd2397/health:${BUILD_NUMBER} maulikd2397/health:latest "
            sh 'docker images'
         }
      }
      stage('Login2Dockerhub') {
         steps {
            withCredentials([string(credentialsId: 'dockerlogin', variable: 'dockerhublogin')]) {
            sh "docker login -u maulikd2397 -p ${dockerhublogin}"
            }
         }
      }
      stage('Push2Dockerhub') {
         steps {
            sh 'docker push maulikd2397/health:latest '
         }
      }
      stage('k8sdeployment') {
         steps {
            sshPublisher(publishers: [sshPublisherDesc(configName: 'kuscluster', transfers: [sshTransfer(cleanRemote: false, excludes: '', execCommand: 'kubectl apply -f k8sdeploy.yaml', execTimeout: 120000, flatten: false, makeEmptyDirs: false, noDefaultExcludes: false, patternSeparator: '[, ]+', remoteDirectory: '.', remoteDirectorySDF: false, removePrefix: '', sourceFiles: 'k8sdeploy.yaml')], usePromotionTimestamp: false, useWorkspaceInPromotion: false, verbose: false)])
         }
      }
   }
}
```

%%%%%%Thank you%%%%%%