DevOps-Day 05:

SAHASRAN M 22CSR167- III CSE

Devops class guvi (DAY-5)

21 March 2025

Granting Jenkins Sudo Privileges – The jenkins ALL=(ALL) NOPASSWD: ALL entry in the sudoers file allows the Jenkins user to run any command without a password prompt.

Restarting SSH Services – Commands like sudo systemctl restart ssh.service and sudo systemctl restart sshd.service restart the SSH service, ensuring remote login functionality.

Installing OpenSSH Server – The commands sudo apt update and sudo apt install openssh-server update package lists and install the OpenSSH server for secure remote access.

Checking SSH Service Status – sudo systemctl status ssh checks if the SSH service is running and displays its current status.

Systemd Service File Lookup – ls /etc/systemd/system/sshd.service or ls /usr/lib/systemd/system/sshd.service helps locate the SSH daemon’s systemd service file.

Reloading Systemd Daemon – sudo systemctl daemon-reload ensures that systemd picks up changes in service configurations without requiring a reboot.

Encoding Minikube Certificate – cat /home/david/.minikube/ca.crt | base64 w 0; echo encodes the Minikube CA certificate in base64 format, likely for authentication.

Changing Docker Socket Permissions – sudo chmod 666

/var/run/docker.sock grants read and write access to all users for Docker’s Unix socket, allowing non-root users to interact with Docker.

Deploying Kubernetes Resources – sh 'kubectl apply -f deployment.yml -validate=false' applies a Kubernetes deployment file, ignoring validation errors.

Accessing Minikube Service – minikube service my-service --url | xargs curl retrieves the Minikube service URL and sends an HTTP request to test its accessibility.

Commands:

jenkins ALL=(ALL) NOPASSWD: ALL

sudo systemctl restart ssh.service

sudo systemctl restart sshd.service

sudo apt update sudo apt install

openssh-server sudo systemctl

restart ssh sudo systemctl status

ssh

ls /etc/systemd/system/sshd.service or ls /usr/lib/systemd/system/sshd.service

sudo systemctl daemon-reload sudo systemctl status ssh sudo systemctl

restart ssh.service cat /home/david/.minikube/ca.crt | base64 -w 0; echo sudo

chmod 666 /var/run/docker.sock

https://192.168.39.226:8443

sh 'kubectl apply -f deployment.yml --validate=false' minikube

service my-service --url | xargs curl

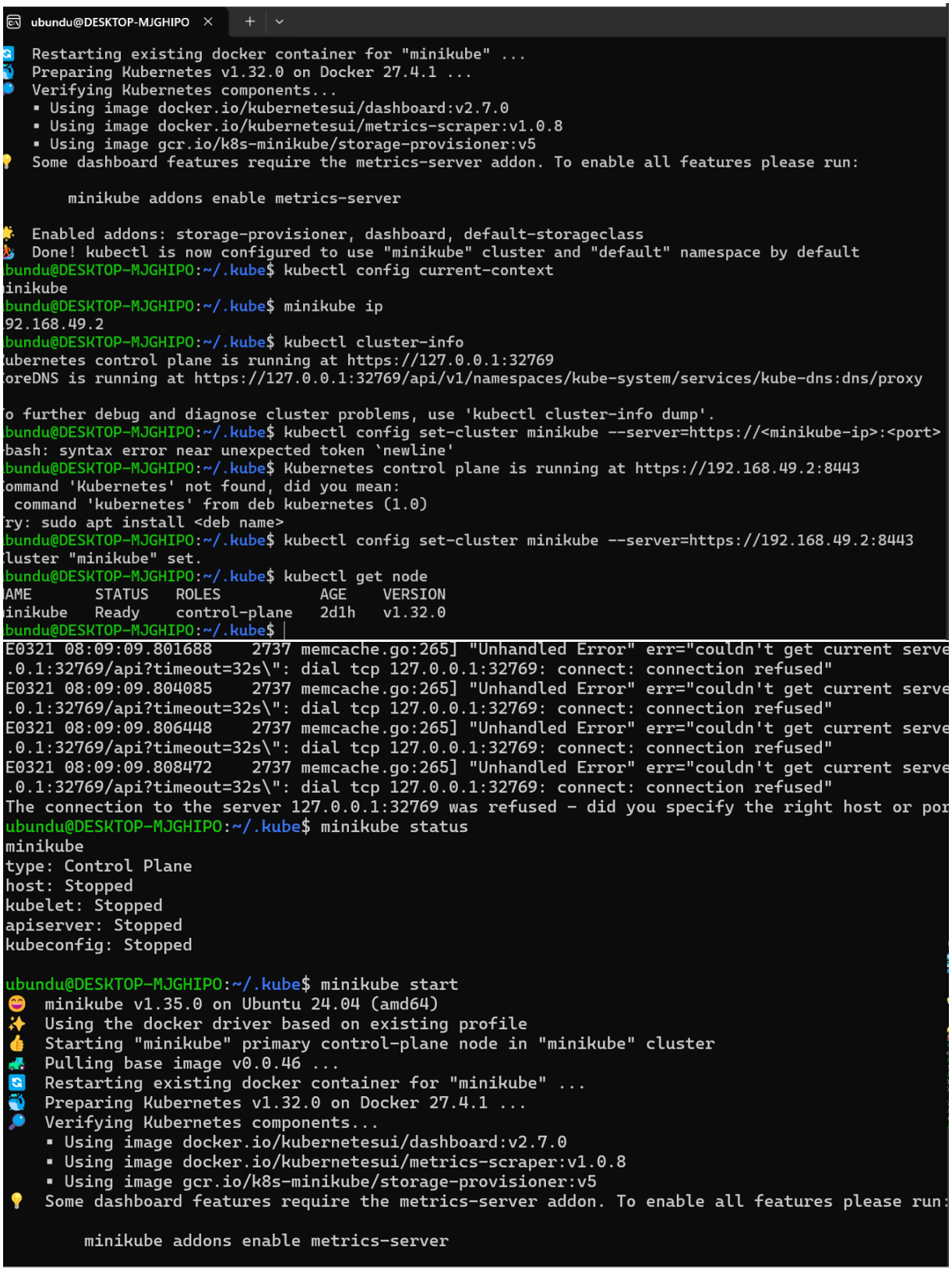
Pipeline codes:

```
pipeline { agent

any


environment {

   DOCKER_CREDENTIALS = credentials('docker-hub-cred')  // Docker Hub
Credentials ID

}


stages {        stage('SCM') {              steps {                   git branch: 'main', url:

'<https://github.com/MugeshS-04/guvidevopsday1.git>'

       }

   }


   stage('Build')          {

steps {            sh "mvn

clean"             sh "mvn

install"

       }

   }


   stage('Build Docker Image') {

       steps {

script {

            sh 'docker build -t mugeshs04/guvidevopsday1 .'

          }

       }

   }
```

```
    stage('Push to Docker Hub') {
        steps       {                                                       script       {
docker.withRegistry('<https://index.docker.io/v1/>',          'docker-hub-cred')         {
sh 'docker push mugeshs04/guvidevopsday1'
                }
            }
        }
    }
}


}

pipeline { agent
any


stages              {
stage('SCM')     {
steps {
            git branch: 'main', url:
'<https://github.com/PraneshC2005/DevOps_simpleweb-app.git>'
        }
    }
    stage('Build-clean') {
steps{              sh 'mvn
clean'
        }
```

```
    }
stage('Build-validate') {
steps{
          sh 'mvn validate'
       }
    }
stage('Build-compile') {
steps{
          sh 'mvn compile'
       }
    }
stage('Build-test') {
steps{
          sh 'mvn test'
       }
    }
stage('Build-package') {
       steps{
          sh 'mvn package'
       }
    }
    stage('build to images') {          steps {          script{
sh "docker build -t praneshc/webapplication ."
       }
       }
    }
    stage('docker push hub') {
```

```
        steps {             script{                 withDockerRegistry(credentialsId: 'cred-2', url:
'<https://index.docker.io/v1/>') {             sh 'docker push praneshc/webapplication'
        }
        }
        }
    }
}}
```


ubundu@DESKTOP-MJGHIPO
Restarting existing docker container for "minikube" ...
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
Verifying Kubernetes components...
▪ Using image docker.io/kubernetesui/dashboard:v2.7.0
▪ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
Some dashboard features require the metrics-server addon. To enable all features please run:
minikube addons enable metrics-server
Enabled addons: storage-provisioner, dashboard, default-storageclass
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
bundu@DESKTOP-MJGHIPO:~/.kube$ kubectl config current-context
inikube
bundu@DESKTOP-MJGHIPO:~/.kube$ minikube ip
92.168.49.2
bundu@DESKTOP-MJGHIPO:~/.kube$ kubectl cluster-info
ubernetes control plane is running at https://127.0.0.1:32769
oreDNS is running at https://127.0.0.1:32769/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
o further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
bundu@DESKTOP-MJGHIPO:~/.kube$ kubectl config set-cluster minikube --server=https://<minikube-ip>:<port>
bash: syntax error near unexpected token `newline'
bundu@DESKTOP-MJGHIPO:~/.kube$ Kubernetes control plane is running at https://192.168.49.2:8443
ommand 'Kubernetes' not found, did you mean:
command 'kubernetes' from deb kubernetes (1.0)
ry: sudo apt install <deb name>
bundu@DESKTOP-MJGHIPO:~/.kube$ kubectl config set-cluster minikube --server=https://192.168.49.2:8443
luster "minikube" set.
bundu@DESKTOP-MJGHIPO:~/.kube$ kubectl get node
AME STATUS ROLES AGE VERSION
inikube Ready control-plane 2d1h v1.32.0
bundu@DESKTOP-MJGHIPO:~/.kube$
E0321 08:09:09.801688 2737 memcache.go:265] "Unhandled Error" err="couldn't get current serve
.0.1:32769/api?timeout=32s\": dial tcp 127.0.0.1:32769: connect: connection refused"
E0321 08:09:09.804085 2737 memcache.go:265] "Unhandled Error" err="couldn't get current serve
.0.1:32769/api?timeout=32s\": dial tcp 127.0.0.1:32769: connect: connection refused"
E0321 08:09:09.806448 2737 memcache.go:265] "Unhandled Error" err="couldn't get current serve
.0.1:32769/api?timeout=32s\": dial tcp 127.0.0.1:32769: connect: connection refused"
E0321 08:09:09.808472 2737 memcache.go:265] "Unhandled Error" err="couldn't get current serve
.0.1:32769/api?timeout=32s\": dial tcp 127.0.0.1:32769: connect: connection refused"
The connection to the server 127.0.0.1:32769 was refused - did you specify the right host or por
ubundu@DESKTOP-MJGHIPO:~/.kube$ minikube status
minikube
type: Control Plane
host: Stopped
kubelet: Stopped
apiserver: Stopped
kubeconfig: Stopped
ubundu@DESKTOP-MJGHIPO:~/.kube$ minikube start
minikube v1.35.0 on Ubuntu 24.04 (amd64)
Using the docker driver based on existing profile
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
Restarting existing docker container for "minikube" ...
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
Verifying Kubernetes components...
▪ Using image docker.io/kubernetesui/dashboard:v2.7.0
▪ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
Some dashboard features require the metrics-server addon. To enable all features please run:
minikube addons enable metrics-server