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](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/Santhosh-P-2005/DevOps.git>'

}

}

stage('Build') { steps {

sh "mvn clean" sh "mvn install"

}

}

stage('Build Docker Image') { steps {

script {

sh 'docker build -t santhosh9405/new-image .'

}

}

}

stage('Push to Docker Hub') { steps {

script {

docker.withRegistry('<https://index.docker.io/v1/>', 'docker-hub-cred') {

sh 'docker push santhosh9405/new-image'

}

}

}

}

}

}

pipeline { agent any

stages {

stage('SCM') { steps {

git branch: 'main', url: '<https://github.com/Santhosh-P-2005/DevOps>'

}

}

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 santhosh9405/new-image ."

}

}

}

stage('docker push hub') { steps {

script{

withDockerRegistry(credentialsId: 'cred-2', url: '<https://index.docker.io/v1/>') {

sh 'docker push santhosh9405/new-image

}

}

}

}

}}

