

## DevOps Day-5

**\*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/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_simple-web-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'
            }
        }
    }
}
}}
```

## Edit Config File:

```
pcubuntu@PraneshPC: ~/kut
pcubuntu@PraneshPC:~$ cat ~/.kube/config
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURCaNDOQWU2Z0F3SUJBZ0lCQWRBTKJna3Foa2lHOXcwQkRrc0ZBREFWTVJnd0RVRURWUVRFE
  RxdwdGFXXAKYTNMaVpVtkJNQjRYRFRJMU1ETXhPREEYTXpBd00xb1hEVE0xTURNeE56QTJNekF3ITTFvd0ZURVRNQkVHQTFVRQpBeE1LYlZsdWFXdDFZbVZEUVRDQ0FTSXdEUVlKS29aSWh2
  Y09BUUVCQlFBRGdnRvB8BRENDQVfVq2dnRUJBTUfKCLZMVMvVWmY5ZER0ZlZ0TlPcGt4aStvT3YrU2hLQkd3T2dIcnJEek1yUktnd3c0dVYzdjBxMmRMa0NQUVUWYl3gKOG5ibGU1Yk5IbDlm
  ZW5GVjVDMm1OU1J6K5SEW5SZet6WFFqCvJlBkFahZVQW083MVVEDzU3UEVIEUdZeTzPAPBdk54QVJnMwJYOHQ3SJ2SG9Y0XY3am56SnL6QZRXUnZ6MjB1N05RMGPXSEdIcngwRTlSSVox
  akLSN1F0dFbZCLpHtmJibk1pNkNZM3ZNamdyMwo2ZkP4VVLfN2VHQy8ZV1pEeW4ySGsrQnLhTFRYUdvMkI4bkLjaWdyGmVbm8KazRiSnRKM1p8NDZ4L1l5d2V0cWFLTjQvUfDkQnBMT2J5
  MitLVLfIZE9SVhBdXVEbjJpSzVqKqjZTTMzandRwpRbTRKbDN0dG9ZS1p4bUloc1I4Q0F3RUFYU5oTUV4d0RnWURWUjBQQVFIL0JBURBZ0trTUIwR0E5VWRKUVFXck1CUUdDQ3NHQVfV
  RkJ3TJUNZ2dyQmdFRkRyR08BVEFQmdOVkhS1UJBJzhFQlRBREFRSC9NQjBHQTFVZERnUvcKQjRVHRAhRFob01CV3FRbkJOUmQ2VGRhRmNZhcm56QU5CZ2tXaGtpRzL3MEJBUXNGOUFPQ0FR
  RUFNOXhLdThEYQpJZEVRAxRzcXBtejk1MzV2EDRWpV0GpsY1JITGVheUY20XlXSDJPMhWNjLjZVfjUUFFLeGcxSDhMT3l6dGNDcItjcDdVSctvb1ZtVjQ1N1kS0xVdHpaT04mQXFPBUW5r
  cy81czVMW00xeXp1TnZydp0b0ZUfJpQVhna01SZHIKek1KQVYyRENJTVZVEMydvNMekYxeDhFqY2l0wCs2SEVQD0V0aXovaVNZM29QZ1FCNhpjekx5d1VrbVZFUeFQNWmVncyQWls
  ZEhEK0VChERSYn0k3cYtVdUlhDbxBNnc9PQotLS0tLUVORCBERVJUSUZJQ0FURSB0tLS0tCg==
  extensions:
  - extension:
    last-update: Fri, 21 Mar 2025 04:49:52 UTC
    provider: minikube.sigs.k8s.io
    version: v1.35.0
    name: cluster_info
    server: https://127.0.0.1:32769
  name: minikube
contexts:
- context:
  cluster: minikube
  extensions:
  - extension:
    last-update: Fri, 21 Mar 2025 04:49:52 UTC
    provider: minikube.sigs.k8s.io
    version: v1.35.0
    name: context_info
    namespace: default
    user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
```

```
pcubuntu@PraneshPC: ~/kut
name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
  client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURJVENDQWdtZ0F3SUJBZ0lCQWpBTKJna3Foa2lHOXcwQkRrc0ZBREFWTVJnd0RVRURWUVRFE
  RxdwdGFXXAKYTNMaVpVtkJNQjRYRFRJMU1ETXhPREEYTXpBd00xb1hEVE0xTURNeE9EQTJNekF3ITTFvd0ZURVRNQkVHQTFVRQpDaE1PYzNsemRHVnRpbTfoYzNSbGnuTXhGakFVQmd0VkhBTBVR
  EVzFwYmVscmRXSxMwFZGWIhJd2dnRUJBTUfKCLZMVMvVWmY5ZER0ZlZ0TlPcGt4aStvT3YrU2hLQkd3T2dIcnJEek1yUktnd3c0dVYzdjBxMmRMa0NQUVUWYl3gKOG5ibGU1Yk5IbDlm
  ZW5GVjVDMm1OU1J6K5SEW5SZet6WFFqCvJlBkFahZVQW083MVVEDzU3UEVIEUdZeTzPAPBdk54QVJnMwJYOHQ3SJ2SG9Y0XY3am56SnL6QZRXUnZ6MjB1N05RMGPXSEdIcngwRTlSSVox
  akLSN1F0dFbZCLpHtmJibk1pNkNZM3ZNamdyMwo2ZkP4VVLfN2VHQy8ZV1pEeW4ySGsrQnLhTFRYUdvMkI4bkLjaWdyGmVbm8KazRiSnRKM1p8NDZ4L1l5d2V0cWFLTjQvUfDkQnBMT2J5
  MitLVLfIZE9SVhBdXVEbjJpSzVqKqjZTTMzandRwpRbTRKbDN0dG9ZS1p4bUloc1I4Q0F3RUFYU5oTUV4d0RnWURWUjBQQVFIL0JBURBZ0trTUIwR0E5VWRKUVFXck1CUUdDQ3NHQVfV
  RkJ3TJUNZ2dyQmdFRkRyR08BVEFQmdOVkhS1UJBJzhFQlRBREFRSC9NQjBHQTFVZERnUvcKQjRVHRAhRFob01CV3FRbkJOUmQ2VGRhRmNZhcm56QU5CZ2tXaGtpRzL3MEJBUXNGOUFPQ0FR
  RUFNOXhLdThEYQpJZEVRAxRzcXBtejk1MzV2EDRWpV0GpsY1JITGVheUY20XlXSDJPMhWNjLjZVfjUUFFLeGcxSDhMT3l6dGNDcItjcDdVSctvb1ZtVjQ1N1kS0xVdHpaT04mQXFPBUW5r
  cy81czVMW00xeXp1TnZydp0b0ZUfJpQVhna01SZHIKek1KQVYyRENJTVZVEMydvNMekYxeDhFqY2l0wCs2SEVQD0V0aXovaVNZM29QZ1FCNhpjekx5d1VrbVZFUeFQNWmVncyQWls
  ZEhEK0VChERSYn0k3cYtVdUlhDbxBNnc9PQotLS0tLUVORCBERVJUSUZJQ0FURSB0tLS0tCg==
  client-key-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURJVENDQWdtZ0F3SUJBZ0lCQWpBTKJna3Foa2lHOXcwQkRrc0ZBREFWTVJnd0RVRURWUVRFE
  RxdwdGFXXAKYTNMaVpVtkJNQjRYRFRJMU1ETXhPREEYTXpBd00xb1hEVE0xTURNeE9EQTJNekF3ITTFvd0ZURVRNQkVHQTFVRQpDaE1PYzNsemRHVnRpbTfoYzNSbGnuTXhGakFVQmd0VkhBTBVR
  EVzFwYmVscmRXSxMwFZGWIhJd2dnRUJBTUfKCLZMVMvVWmY5ZER0ZlZ0TlPcGt4aStvT3YrU2hLQkd3T2dIcnJEek1yUktnd3c0dVYzdjBxMmRMa0NQUVUWYl3gKOG5ibGU1Yk5IbDlm
  ZW5GVjVDMm1OU1J6K5SEW5SZet6WFFqCvJlBkFahZVQW083MVVEDzU3UEVIEUdZeTzPAPBdk54QVJnMwJYOHQ3SJ2SG9Y0XY3am56SnL6QZRXUnZ6MjB1N05RMGPXSEdIcngwRTlSSVox
  akLSN1F0dFbZCLpHtmJibk1pNkNZM3ZNamdyMwo2ZkP4VVLfN2VHQy8ZV1pEeW4ySGsrQnLhTFRYUdvMkI4bkLjaWdyGmVbm8KazRiSnRKM1p8NDZ4L1l5d2V0cWFLTjQvUfDkQnBMT2J5
  MitLVLfIZE9SVhBdXVEbjJpSzVqKqjZTTMzandRwpRbTRKbDN0dG9ZS1p4bUloc1I4Q0F3RUFYU5oTUV4d0RnWURWUjBQQVFIL0JBURBZ0trTUIwR0E5VWRKUVFXck1CUUdDQ3NHQVfV
  RkJ3TJUNZ2dyQmdFRkRyR08BVEFQmdOVkhS1UJBJzhFQlRBREFRSC9NQjBHQTFVZERnUvcKQjRVHRAhRFob01CV3FRbkJOUmQ2VGRhRmNZhcm56QU5CZ2tXaGtpRzL3MEJBUXNGOUFPQ0FR
  RUFNOXhLdThEYQpJZEVRAxRzcXBtejk1MzV2EDRWpV0GpsY1JITGVheUY20XlXSDJPMhWNjLjZVfjUUFFLeGcxSDhMT3l6dGNDcItjcDdVSctvb1ZtVjQ1N1kS0xVdHpaT04mQXFPBUW5r
  cy81czVMW00xeXp1TnZydp0b0ZUfJpQVhna01SZHIKek1KQVYyRENJTVZVEMydvNMekYxeDhFqY2l0wCs2SEVQD0V0aXovaVNZM29QZ1FCNhpjekx5d1VrbVZFUeFQNWmVncyQWls
  ZEhEK0VChERSYn0k3cYtVdUlhDbxBNnc9PQotLS0tLUVORCBERVJUSUZJQ0FURSB0tLS0tCg==
  pcubuntu@PraneshPC:~$ cd ~/.kube
pcubuntu@PraneshPC:~/kube$ sudo vi config
[sudo] password for pcubuntu:
pcubuntu@PraneshPC:~/kube$ kubectl grt node
error: unknown command "grt" for "kubectl"
```

Development.yml:

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-deploy

labels:

name: my-deploy

spec:

replicas: 1

selector:

matchLabels:

apptype: web-backend

strategy:

type: RollingUpdate

template:

metadata:

labels:

apptype: web-backend

spec:

containers:

- name: my-app

image: praneshc/webapplication

ports:

- containerPort: 9000

---

apiVersion: v1

kind: Service

metadata:

name: my-service

labels:

app: my-service

spec:

type: NodePort

ports:

- port: 9000

targetPort: 8080

nodePort: 30002

selector:

apptype: web-backend

## Cloud Testing:



The screenshot shows the Jenkins Cloud Testing configuration interface. At the top is a large empty text input field. Below it is a checkbox labeled "Inject restricted PSS security context in agent container definition" with a help icon. The "Credentials" section features a dropdown menu currently showing "config (kube-cred)" and a "+ Add" button. Below the credentials is a status indicator "Connected to Kubernetes v1.32.0" and a "Test Connection" button. Underneath are two checkboxes: "WebSocket" and "Direct Connection", both with help icons. An orange warning message states: "TCP port for inbound agents' is disabled in Global Security settings. Connecting Kubernetes agents will not work without this or WebSocket model". At the bottom is a "Jenkins URL" field with a help icon.

Create a new application in java ,a pipeline project



**Jenkins** Dashboard

REST API Jenkins 2.492.2

Build Queue: No builds in the queue.

Build Executor Status: 0/2

S	W	Name	Last Success	Last Failure	Last Duration
✓	☁	java-application	31 min #6	1 hr 8 min #4	1 min 16 sec
✓	☀	maven	2 days 1 hr #1	N/A	53 sec
✓	☔	simpletomapp	1 day 23 hr #6	1 day 23 hr #5	58 sec

**Jenkins** Dashboard > java-application

Stage View

Average stage times: (full run time: ~53s)

	SCM	Build-clean	Build-validate	Build-compile	Build-test	Build-package	build to images	docker push hub	Deploy App
#6 15:17 No Changes	6s	3s	2s	3s	3s	3s	741ms	50s	979ms
#5 14:43 No Changes	1s	2s	1s	2s	2s	3s	508ms	29s	337ms
#4 14:40 No Changes	1s	2s	1s	2s	3s	4s	484ms	37s	533ms failed
#3 14:36 No Changes	1s	3s	2s	3s	3s	3s	508ms	30s	1s failed
#2 13:37 No Changes	1s	1s	848ms	1s	1s	3s	336ms	28s	
#1 13:33 No Changes	1s	1s	848ms	1s	1s	1s	422ms	75ms	

Hello World Output:

```
pcubuntu@PraneshPC: ~/kut
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 gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: default-storageclass, storage-provisioner
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
pcubuntu@PraneshPC: ~/kube$ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
my-app        1/1     Running   1 (127m ago)  5h8m
my-deploy-795d5b5d8c-tpm7h  1/1     Running   0           58m
my-pod        1/1     Running   1 (127m ago)  5h19m
pcubuntu@PraneshPC: ~/kube$ minikube service my-service


| NAMESPACE | NAME       | TARGET PORT | URL                       |
|-----------|------------|-------------|---------------------------|
| default   | my-service | 9000        | http://192.168.49.2:30002 |


Starting tunnel for service my-service.


| NAMESPACE | NAME       | TARGET PORT | URL                    |
|-----------|------------|-------------|------------------------|
| default   | my-service |             | http://127.0.0.1:34781 |


Opening service default/my-service in default browser...
http://127.0.0.1:34781
Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^C Stopping tunnel for service my-service.
pcubuntu@PraneshPC: ~/kube$ curl http://192.168.49.2:30002
<!doctype html><html lang="en"><head><title>HTTP Status 404 - Not Found</title><style type="text/css">body {font-family:Tahoma,Arial,sans-serif;
} h1, h2, h3, b {color:white;background-color:#525D76;border:none;} h1 {font-size:22px;} h2 {font-size:16px;} h3 {font-size:14px;} p {font-size:12px;} a {co
lor:black;} .line {height:1px;background-color:#525D76;border:none;}</style></head><body><h1>HTTP Status 404 - Not Found</h1><hr class="line" />
<p><b>Type</b> Status Report</p><p><b>Description</b> The origin server did not find a current representation for the target resource or is not
willing to disclose that one exists.</p><hr class="line" /><h3>Apache Tomcat/9.0.102</h3></body></html>pcubuntu@PraneshPC: ~/kube$ ^C
pcubuntu@PraneshPC: ~/kube$ curl http://192.168.49.2:30002/my-app/
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
```