

## WEEK 10 – LAB ACTION PLAN

Working with Minikube and Nagios

### KUBERNETES BASICS

Kubernetes automates running and managing containerized applications.

Pods are the smallest deployable units.

Minikube provides a local single-node Kubernetes cluster.

### NAGIOS BASICS

Nagios monitors:

- Servers
- Network devices
- Applications & services

It alerts administrators when issues occur.

---

### MINIKUBE SETUP & PRACTICAL STEPS

---

#### STEP 1: INSTALL PREREQUISITES

1. Ensure virtualization support is enabled.

2. Install any one hypervisor:

- Hyper-V (Windows Pro/Enterprise)
- Docker Desktop
- VirtualBox

Commands to check virtualization:

Run in CMD:

`systeminfo | find "Virtualization"`

#### STEP 2: DOWNLOAD & INSTALL MINIKUBE

Run in PowerShell (Admin):

```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-installer.exe  
.\minikube-installer.exe
```

#### STEP 3: ADD MINIKUBE TO PATH (IF REQUIRED)

Add: C:\Program Files\Minikube to PATH environment variable.

#### STEP 4: START MINIKUBE

Choose hypervisor:

```
minikube start --driver=docker
```

OR

```
minikube start --driver=hyperv
```

Check status:

```
minikube status
```

#### STEP 5: USING KUBECTL (KUBERNETES CLI)

Check pods:

```
minikube kubectl -- get pods -A
```

Open dashboard:

```
minikube dashboard
```

Verify installation:

```
minikube version
```

```
kubectl version --client
```

---

## MINIKUBE AUTOMATION STEPS

---

#### STEP 1: START CLUSTER

```
minikube start
```

## STEP 2: CREATE A DEPLOYMENT

```
kubectl create deployment mynginx --image=nginx
```

Check deployment:

```
kubectl get deployments
```

```
kubectl get pods
```

```
kubectl describe pods
```

Update image (if already created):

```
kubectl set image deployment/mynginx nginx=nginx:latest
```

## STEP 3: EXPOSE DEPLOYMENT

```
kubectl expose deployment mynginx --type=NodePort --port=80 --target-port=80
```

Check service:

```
kubectl get service mynginx
```

## STEP 4: SCALE DEPLOYMENT

```
kubectl scale deployment mynginx --replicas=4
```

## STEP 5: ACCESS NGINX APP

Option 1 – Port Forward:

```
kubectl port-forward svc/mynginx 8081:80
```

Open: <http://localhost:8081>

Option 2 – Minikube Tunnel:

```
minikube tunnel
```

```
minikube service mynginx --url
```

## STEP 6: CLEAN UP

Delete app:

```
kubectl delete deployment mynginx
```

```
kubectl delete service mynginx
```

Stop minikube:

```
minikube stop
```

Delete minikube:

```
minikube delete
```

---

## NAGIOS AUTOMATION STEPS

---

### STEP 1: PULL NAGIOS DOCKER IMAGE

```
docker pull jasonrivers/nagios:latest
```

### STEP 2: RUN NAGIOS

```
docker run --name nagiosdemo -p 8888:80 jasonrivers/nagios:latest
```

Access dashboard:

<http://localhost:8888>

Login:

Username: nagiosadmin

Password: nagios

### STEP 3: STOP & REMOVE NAGIOS

```
docker stop nagiosdemo
```

```
docker rm nagiosdemo
```

List images:

```
docker images
```

Remove image:

```
docker rmi jasonrivers/nagios:latest
```

---

## AWS FREE TIER ACCOUNT CREATION STEPS

---

1. Open <https://aws.amazon.com>
  2. Click “Create AWS account”
  3. Enter email, verification code
  4. Add contact details
  5. Enter billing card details (INR 2 refundable)
  6. Verify mobile OTP
  7. Select Basic Plan
  8. Log in as Root user
- 

## VIVA QUESTIONS (SBQs)

1. Pod restarting repeatedly – check logs, describe pod, events.
2. Pod stuck in Pending – insufficient resources, volume issues.
3. Debug failed deployment – kubectl describe deployment, logs.
4. Pods failing health checks – check readiness & liveness probes.
5. Roll back faulty deployment – kubectl rollout undo.
6. Debug running pod – kubectl logs, kubectl exec -it.
7. Expose service externally – kubectl expose / NodePort / Ingress.
8. Start/Stop Nagios – docker run / docker stop.
9. Nagios web issue – restart container, check logs.
10. New host not appearing – verify config files, restart Nagios.
11. Check Nagios running – ps aux | grep nagios.
12. View logs in real time – tail -f /var/log/nagios/nagios.log.
13. Advantages of Nagios – monitoring, alerting, plugins, dashboards.

## CONCLUSION

In this lab we learned:

- Running Kubernetes pods via Minikube

- Scaling and exposing services
- Monitoring with Nagios
- Creating AWS Free Tier account