

LAB ACTION PLAN FOR WEEK 10

Working with minikube and Nagios

1. Hands-on practice of creating, running and scaling pods in minikube.
2. Running Nginx server on specified port number by explaining the Nginx monitoring tool
3. Running Nagios server and Understanding the Monitoring tool using Docker.
4. AWS-free Trier account Creation steps
5. Upload the screenshots for the tasks

Kubernetes:

Kubernetes is a tool that automates how we run and manage applications inside the container.

Dockers will only run containers, if in any case the container fails/stopped/killed, the docker will not help us, here is where Kubernetes plays an important role, Kubernetes cluster will be responsible in creating a new container and managing various containers.

POD: In Kubernetes, a Pod is the smallest deployable unit that you can create and manage.

Minikube:

Minikube creates a VM on your local machine and deploys a simple Kubernetes cluster with one node. It's a lightweight implementation. Minikube is a version of Kubernetes.

Nagios:

Nagios is an **open-source IT infrastructure monitoring tool**. It monitors

- Servers
- Network devices
- Applications and services

It **alerts administrators** when issues occur and notifies when they are resolved.

Step 1: Install Prerequisites

Before installing Minikube, ensure the following are installed:

1. **Virtualization Support:**
 - Verify virtualization is enabled:

2. Hypervisor:

- Minikube supports multiple hypervisors (e.g., **Hyper-V**, **VirtualBox**, or **Docker** as a driver).
 - Install one of the following:
 - **Hyper-V** (pre-installed on Windows 10/11 Pro or Enterprise).
 - **Docker Desktop** (if you want to use Docker as the driver).
-

Step 2: Download Minikube

1. Open a PowerShell or Command Prompt with administrator privileges.

2. Download the latest Minikube executable using this command:

3. curl -LO
<https://storage.googleapis.com/minikube/releases/latest/minikube-installer.exe>

4. Install Minikube by running the installer:

5. .\minikube-installer.exe

Step 3: Add Minikube to PATH

If Minikube is not automatically added to your PATH during installation:

1. Open **System Properties** → **Environment Variables**.

2. Add the directory where Minikube is installed (e.g., C:\Program Files\Minikube) to your PATH variable.

Step 4: Start Minikube

1. Open a terminal (PowerShell or CMD). Do the following commands

2. Start Minikube with a specified driver (e.g., Hyper-V, Docker, or VirtualBox). For example:

- **Hyper-V:**
 - minikube start --driver=hyperv
- **Docker:**
 - minikube start --driver=docker

3. Verify Minikube is running:

4. minikube status

Step 5: Interact with Minikube

kubectl is a command-line tool used in Kubernetes to interact with and manage Kubernetes clusters.

Once Minikube is running:

1. Use kubectl to interact with the cluster.
 - o Install kubectl if not already installed:
 - o minikube kubectl -- get pods -A
 - o Or download it separately from the [official Kubernetes site](#).
 2. Open the Minikube dashboard (optional):
 3. minikube dashboard
-

Optional: Check Your Installation

Run the following to verify the installation:

Minikube version

```
kubectl version --client
```

Troubleshooting

1. **If Minikube fails to start:**
 - o Ensure your hypervisor (Hyper-V/Docker/VirtualBox) is installed and running.
 - o Check the Minikube logs:
 - o minikube logs
2. **Updating Minikube:**
 3. minikube update-check
 4. minikube update

Minikube Automation Steps

Step 1: Start Minikube Cluster

- Open your terminal and run the command:

```
minikube start
```

Step 2: Create and Manage Deployment

1. **Create an application in Kubernetes:**

- o Command:

```
kubectl create deployment mynginx --image=nginx  
  
if already created then  
  
kubectl set image deployment/myngnix nginx=nginx:latest
```

- Verify the deployment using: Kubernetes responds by showing you a list that includes the names of your deployment groups

```
kubectl get deployments
```

- Ensure `mynginx` appears in the output.

Check the following commands:

- `kubectl get pods`
- `kubectl describe pods`

2. Expose Deployment as a Service:

- Command:

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

Step 3: Scale the Deployment

Command:Scales the Nginx deployment to 4 replicas (pods).

```
kubectl scale deployment mynginx --replicas=4  
  
kubectl get service myngnix
```

Step 4: Access the Nginx App

1. Using Port Forwarding:

- Command:

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

- Access the app via <http://localhost:8081>.

If Error, use this option, **Using Minikube Tunnel**:

- Start the tunnel:

```
minikube tunnel
```

- Retrieve the service URL:

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

- Open the provided URL in your browser.
- Open the kubernets dashboard

- Open the minikube dashboard

Minikube dashboard

Step 5: Stop and Clean Up

1. Stop Nginx Deployment:

- Commands:

```
kubectl delete deployment mynginx  
kubectl delete service mynginx
```

2. Stop Minikube (Optional):

- Command:

```
minikube stop
```

3. Delete Minikube Cluster (Optional):

- Command:
`minikube delete`

Nagios Automation Steps

Step 1: Pull the Nagios Docker Image

- Open a terminal and run:

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

Step 2: Run Nagios

- Command:

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

Step 3: Access Nagios Dashboard

- Open your browser and navigate to:

<http://localhost:8888>

- **Login Credentials:**
 - Username: nagiosadmin
 - Password: nagios
- Once logged in, explore:
 - Hosts: View systems being monitored.
 - Services: Check tasks being monitored (e.g., CPU usage).
 - Alerts: Access recent notifications.

Step 4: Monitoring Host Details

1. **Navigate to the Host Information Page:**
 - Select a host from the **Hosts** menu.
2. **Key Details:**
 - Host Status: Indicates if the system is UP or DOWN.
 - Metrics: View CPU usage, memory status, and network activity.

- o Actions: Reschedule checks, disable notifications, or schedule downtime.

Step 5: Stop and Remove Nagios

1. Stop the Container:

- o Command:

```
docker stop nagiosdemo
```

2. Delete the Container:

- o Command:

```
docker rm nagiosdemo
```

3. Remove the Image (Optional):

- o List images:

```
docker images
```

- o Delete the Nagios image:

```
docker rmijasonrivers/nagios:latest
```

4. Observe the docker containers in DockerHub, we can see the latest Nagios Installed running on port:8888

Steps for AWS-free trier account creation

- open <https://aws.amazon.com>
- click on create AWS account
- Provide email id and name details for AWS account creation then a screen will appear as below
- Enter verification code received in your mail
- Provide your contact details and agree Terms and conditions, then click on create

- Provide billing details, click on verify and continue
- Amazon charges 2 rs and it will be credit back into your account once verification is over
- Once payment success a screen appear as below, provide your working contact number after selecting country, and click on send SMS after entering the details
- Provide verification code received in your mobile
- Now select basic plan and click on complete sign up
- Now a screen appear as follows then click on AWS management console
- You will navigating to the aws page ..where you need to Select role as Academic/Researcher and interest as DevOps and click on submit
- Sign into AWS console as Root user
- A console screen appears aws will appear

SBQ's

1. Your Pod keeps restarting repeatedly. What will you do?
2. A Kubernetes pod is stuck in a "Pending" state. What could be the possible reasons, and how would you troubleshoot it?
3. How would you debug a failed deployment in Kubernetes?
4. You have a Kubernetes Deployment with multiple replicas, and some pods are failing health checks. How would you identify the root cause and fix it?
5. How do you roll back a faulty deployment?
6. How do you debug a running Pod?
7. You need to expose a local service externally. How to do?
8. How to Start and stop Nagios
9. You installed Nagios but the web interface shows "*Unable to connect to Nagios process*". how to resolve this?
10. You added a new host in Nagios, but it's not appearing on the web interface. how to check?
11. How can you check whether Nagios is running properly?
12. How do you view Nagios logs in real-time
13. What are the advantages of using Nagios?

Conclusion: In this week we learnt how to deploy pods in minikube and how to monitor our local system using Nagios tool and will know how to create the aws free tier account.