

Modul Praktikum 7

Container Orchestration dengan Kubernetes

1. Maksud

Modul ini bertujuan untuk memberikan pemahaman dan keterampilan praktis dalam mengelola dan mengatur container menggunakan Kubernetes sebagai alat *orchestration*. Peserta akan mempelajari konsep dasar, arsitektur, serta penerapan Kubernetes untuk mengelola aplikasi yang berjalan dalam container secara efisien.

2. Tujuan

Setelah menyelesaikan modul ini, peserta diharapkan mampu:

- a) Memahami konsep dasar *container orchestration* dan peran Kubernetes.
- b) Menginstal dan mengkonfigurasi Kubernetes (*minikube/kubeadm*).
- c) Menjalankan aplikasi dalam *cluster* Kubernetes.
- d) Mengelola *deployment*, *scaling*, *service discovery*, dan *load balancing*.
- e) Memahami cara kerja *Pods*, *services*, *deployments*, dan *namespaces*.
- f) Menerapkan studi kasus sederhana menggunakan Kubernetes.

3. Dasar Teori

3.1 Container Orchestration

Container orchestration adalah proses mengotomatiskan penerapan, pengelolaan, penskalaan, dan jaringan container. Alat seperti Kubernetes membantu dalam mengelola aplikasi yang terdiri dari banyak container yang berjalan di beberapa host.

3.2 Kubernetes

Kubernetes (K8s) adalah platform *open-source* untuk mengotomatiskan operasi container. Beberapa komponen utama Kubernetes:

- a) **Cluster:** Kumpulan *node* (mesin fisik/virtual) yang menjalankan container.
- b) **Node:** *Worker machine* yang menjalankan aplikasi dalam container.
- c) **Pod:** Unit terkecil dalam Kubernetes, berisi satu atau lebih container.
- d) **Deployment:** Mengelola replika Pod dan memastikan aplikasi berjalan sesuai keinginan.
- e) **Service:** Menyediakan *load balancing* dan *service discovery* untuk Pod.
- f) **Namespace:** Membagi *cluster* menjadi beberapa bagian logis.

4. Prosedur Praktik

4.1 Persiapan

Instalasi Docker

1. Unduh dan instal Docker sesuai sistem operasi Anda:

- a. **Linux (Ubuntu/Debian):**

```
sudo apt update && sudo apt install docker.io -y  
sudo systemctl enable --now docker
```

- b. **Windows/macOS:**

Unduh dari [Docker Desktop](#) dan ikuti panduan instalasi.

2. Verifikasi instalasi:

```
docker --version
```

Instalasi kubectl (Kubernetes CLI)

1. **Linux:**

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s  
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"  
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

2. **Windows (PowerShell):**

```
curl.exe -LO "https://dl.k8s.io/release/v1.28.0/bin/windows/amd64/kubectl.exe"
```

Tambahkan ke PATH.

3. Verifikasi:

```
kubectl version --client
```

Instalasi Minikube

1. **Linux:**

```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-  
amd64  
sudo install minikube-linux-amd64 /usr/local/bin/minikube
```

2. **Windows (PowerShell):**

choco install minikube # Jika menggunakan Chocolatey

Atau unduh dari [Minikube Releases](#).

3. MacOS (Homebrew):

```
brew install minikube
```

4. Mulai Minikube dengan Docker Driver:

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

5. Verifikasi:

```
minikube
```

status

```
kubectl get nodes
```

4.2 Langkah-Langkah Praktik

1. Menjalankan Minikube

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

2. Mengecek Status Cluster

```
kubectl get nodes
```

3. Membuat Deployment

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

4. Mengekspos Service

```
kubectl expose deployment nginx --port=80 --type=NodePort
```

5. Mengakses Aplikasi

```
minikube service nginx
```

5. Latihan (Studi Kasus dan Penyelesaian)

Studi Kasus: Deploy Aplikasi Web Sederhana

Permasalahan:

Anda diminta untuk mendeploy aplikasi web berbasis Nginx dan memastikan aplikasi dapat diakses melalui browser.

Penyelesaian:

1. Buat *deployment* Nginx:

```
kubectl create deployment web-app --image=nginx
```

2. Ekspos sebagai *service*:

```
kubectl expose deployment web-app --port=80 --type=NodePort
```

3. Verifikasi:

```
kubectl get pods,svc
```

4. Akses aplikasi:

```
minikube service web-app
```

6. Tugas

1. Instalasi Kubernetes

- Instal Minikube dan kubectl di mesin lokal Anda.
- Jalankan *cluster* dan verifikasi statusnya.

2. Deploy Aplikasi

- Buat *deployment* menggunakan image httpd.
- Ekspos aplikasi tersebut dan akses melalui browser.

3. Scaling

- Scale *deployment* menjadi 3 replika.
- Verifikasi bahwa semua Pod berjalan.

4. Persistent Storage

- Buat *PersistentVolume* dan *PersistentVolumeClaim*.
- Mount ke dalam Pod dan simpan data di dalamnya.

5. Analisis

- Jelaskan perbedaan antara Deployment, Service, dan Pod.
- Apa manfaat menggunakan *Namespace* dalam Kubernetes?

Catatan: Kumpulkan dokumentasi langkah-langkah yang telah dilakukan beserta screenshot hasilnya.

Referensi:

- Dokumentasi Resmi Kubernetes: <https://kubernetes.io/docs>

- Buku "Kubernetes Up & Running" oleh Kelsey Hightower.