**Environment DOFFINder**

**(Martin/Tommy prosjekt)**

**(** Kubernetes **Minikube**)

**Oppgaver for å trene på Kubernetes**

**1. Forbered Kubernetes-miljøet**

* **Læringsmål**: Sett opp et Kubernetes-miljø for å teste applikasjonen.
* **Handlinger**:
  1. Installer **Minikube** (for lokalt Kubernetes-kluster) eller bruk en skyløsning som Google Kubernetes Engine (GKE), Amazon EKS eller Azure AKS.
  2. Installer **kubectl** (Kubernetes CLI) for å administrere klusteren.
  3. Start Minikube og sørg for at det kjører:

bash

Kopier kode

minikube start

kubectl get nodes

**2. Opprett Docker-image og last opp til container registry**

* **Læringsmål**: Gjøre Docker-image tilgjengelig for Kubernetes-klusteret.
* **Handlinger**:
  1. Bygg og tag Docker-imaget:

bash

Kopier kode

docker build -t doffinder:latest .

* 1. Push Docker-imaget til en container registry, som Docker Hub:

bash

Kopier kode

docker tag doffinder:latest <your-dockerhub-username>/doffinder:latest

docker push <your-dockerhub-username>/doffinder:latest

**3. Opprett Kubernetes Deployment**

* **Læringsmål**: Distribuer DOFFINder som en Kubernetes-arbeidsbelastning.
* **Handlinger**:
  1. Lag en fil deployment.yaml for backend:

yaml

Kopier kode

apiVersion: apps/v1

kind: Deployment

metadata:

name: doffinder-backend

spec:

replicas: 3

selector:

matchLabels:

app: doffinder-backend

template:

metadata:

labels:

app: doffinder-backend

spec:

containers:

- name: doffinder-backend

image: <your-dockerhub-username>/doffinder:latest

ports:

- containerPort: 8000

env:

- name: OPENAI\_API\_KEY

value: "your-openai-api-key"

- name: GOOGLE\_API\_KEY

value: "your-google-api-key"

* 1. Apply deployment til klusteret:

bash

Kopier kode

kubectl apply -f deployment.yaml

**4. Opprett en Kubernetes Service**

* **Læringsmål**: Eksponer applikasjonen for tilgang utenfor klusteret.
* **Handlinger**:
  1. Lag en fil service.yaml for backend:

yaml

Kopier kode

apiVersion: v1

kind: Service

metadata:

name: doffinder-backend-service

spec:

selector:

app: doffinder-backend

ports:

- protocol: TCP

port: 80

targetPort: 8000

type: LoadBalancer

* 1. Apply service til klusteret:

bash

Kopier kode

kubectl apply -f service.yaml

* 1. Sjekk LoadBalancer-adressen for tilgang:

bash

Kopier kode

kubectl get services

**5. Konfigurer Secrets**

* **Læringsmål**: Håndter API-nøkler og sensitiv informasjon sikkert.
* **Handlinger**:
  1. Opprett en Secret for API-nøklene:

bash

Kopier kode

kubectl create secret generic doffinder-secrets \

--from-literal=OPENAI\_API\_KEY=your-openai-api-key \

--from-literal=GOOGLE\_API\_KEY=your-google-api-key

* 1. Oppdater deployment for å bruke Secret:

yaml

Kopier kode

env:

- name: OPENAI\_API\_KEY

valueFrom:

secretKeyRef:

name: doffinder-secrets

key: OPENAI\_API\_KEY

- name: GOOGLE\_API\_KEY

valueFrom:

secretKeyRef:

name: doffinder-secrets

key: GOOGLE\_API\_KEY

**6. Opprett ConfigMaps for konfigurasjon**

* **Læringsmål**: Administrer ikke-sensitiv konfigurasjon.
* **Handlinger**:
  1. Lag en ConfigMap for generell konfigurasjon:

bash

Kopier kode

kubectl create configmap doffinder-config --from-literal=APP\_ENV=production

* 1. Legg til ConfigMap i deployment:

yaml

Kopier kode

env:

- name: APP\_ENV

valueFrom:

configMapKeyRef:

name: doffinder-config

key: APP\_ENV

**7. Oppsett for horisontal skalering**

* **Læringsmål**: Lær hvordan Kubernetes skalerer applikasjoner.
* **Handlinger**:
  1. Legg til en Horizontal Pod Autoscaler (HPA):

bash

Kopier kode

kubectl autoscale deployment doffinder-backend --cpu-percent=50 --min=1 --max=5

**8. Monitorering med Kubernetes**

* **Læringsmål**: Overvåk arbeidsbelastningene.
* **Handlinger**:
  1. Sjekk status for pods:

bash

Kopier kode

kubectl get pods

* 1. Sjekk logger for en pod:

bash

Kopier kode

kubectl logs <pod-name>

* 1. Installer et monitoreringsverktøy som Prometheus eller Grafana i klusteret.

**9. Bruk Ingress for avansert routing**

* **Læringsmål**: Lær hvordan du eksponerer tjenester med Ingress.
* **Handlinger**:
  1. Installer en Ingress-kontroller (som NGINX).
  2. Lag en ingress.yaml-fil for å definere routingregler.

**10. Ekstra oppgaver**

* Opprett et Helm-chart for prosjektet for å gjøre det enklere å installere.
* Utforsk CI/CD-integrasjon ved å bruke GitHub Actions eller Jenkins for å automatisk deploye til Kubernetes.

1. Dokumentasjon/kode for DOFFINder:

<https://github.com/ItemConsulting/DOFFINder>

1. Installasjon ved GitHub:

<https://github.com/ItemConsulting/DOFFINder>

Ignacio Tejera 15.01.2025

redigert