

TP INTRO DOCKER ET MINIKUBE

OBJECTIFS

- Exécuter une application Node.js conteneurisée avec Docker sans avoir Node.js sur votre machine.
- Déployer cette application dans un cluster local Kubernetes avec Minikube.
- Comprendre l'intérêt de Docker et Minikube pour l'infrastructure as code.

CREER ET EXECUTER UNE APPLICATION AVEC DOCKER

Préparation

- 1- Créer un répertoire pour le projet :

```
mkdir node-docker && cd node-docker
```

- 2- Créer le fichier server.js :

```
const http = require('http');


const hostname = '0.0.0.0';
const port = 3000;

const server = http.createServer( { requestListener: (
  req : InstanceType<IncomingMessage> ,
  res : InstanceType<ServerResponse> & {req: InstanceType<IncomingMessage>} ) => {
    res.statusCode = 200;
    res.setHeader( name: 'Content-Type', value: 'text/plain' );
    res.end('Hello, Docker and Minikube from Node.js!');
  }
});

server.listen(port, hostname, backlog: () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

- 3- Créer l'image Docker, créer un fichier Dockerfile :

Ce fichier indique à Docker comment exécuter le serveur Node.js.

```
FROM node:22-alpine AS  build
WORKDIR /app
COPY package*.json ./
RUN npm ci --only=production
COPY . .

FROM node:22-alpine
WORKDIR /app
COPY --from=build /app ./
EXPOSE 8080
ENV NODE_ENV=production
CMD ["node", "server.js"]
```

Vous pouvez ajouter ce .dockerignore :

```
node_modules
npm-debug.log
.DS_Store
.git
```

- 4- Construisez l'image Docker (sans avoir Node.js installé !) :

```
docker build -t hello-node:1.0 .
```

Le point (.) à la fin de la commande indique à Docker que le contexte de construction (le répertoire contenant les fichiers nécessaires à la création de l'image) est le répertoire

courant.

- 5- Exécuter le conteneur :

```
docker run -d -p 8080:3000 node-app
```

Accédez à l'application, ouvrez un navigateur sur <http://localhost:8080>

Vous verrez : « Hello, Docker and Minikube from Node.js ! »

- 6- Vérifier le conteneur actif :

```
docker ps
```

Déploiement dans Kubernetes avec Minikube

- 1- Démarrer Minikube :

Lancez Minikube :

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

Configurez Minikube pour utiliser Docker localement :

```
minikube image build -t hello-node:1.0 .
```

Recréez l'image Docker pour Minikube :

```
docker build -t node-app .
```

- 2- Créer le déploiement Kubernetes :

Créez un fichier deployment.yaml :

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-node
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello-node
  template:
    metadata:
      labels:
        app: hello-node
    spec:
      containers:
        - name: app
          image: hello-node:1.0
          imagePullPolicy: IfNotPresent
          ports:
            - containerPort: 8080
---
apiVersion: v1
kind: Service
metadata:
  name: hello-node
spec:
  type: NodePort
  selector:
    app: hello-node
  ports:
    - port: 8080
      targetPort: 8080
      nodePort: 30007

```

Les tirets dans votre fichier yaml servent à séparer plusieurs documents yaml dans un seul fichier.

3- Déployer sur Minikube :

Appliquez le fichier de déploiement :

```
kubectl apply -f deployment.yaml
kubectl get pods
minikube service hello-node --url
# OU (pour type LoadBalancer) :
# minikube tunnel
```

Vérifier le statut des pods :

```
kubectl get pods
```

Accéder à l'application via Minikube :

Ouvrez le service dans le navigateur :

```
minikube service node-app-service
```

Ou accédez directement à l'URL : `http://<minikube-ip>:30007`

Mener l'analyse :

Pourquoi Docker a permis d'exécuter le serveur Node.js sans Node.js installé localement ?

Quel est l'intérêt d'utiliser Minikube/Kubernetes pour déployer un conteneur ?

Quelles sont les différences entre lancer un conteneur directement avec Docker et l'orchestrer avec Kubernetes ?