



Université Paris Cité

Rapport Programmation Web

29/04/2024

HASNI Ziad
AHMED YAHIA Yacine

Introduction.....	2
Commencer par un seul service en local.....	2
Ajouter une gateway en local.....	5
Réalisation Google Labs.....	6

Introduction

Minikube est un outil qui permet d'exécuter Kubernetes localement. Il crée un cluster Kubernetes à nœud unique sur votre ordinateur personnel (y compris sur les PC Windows, macOS et Linux) afin que vous puissiez essayer Kubernetes ou effectuer un travail de développement quotidien.

Commencer par un seul service en local

- Coder une mini application dans le langage que vous voulez
- Créer une image Docker => faire un Dockerfile
- Publier l'image Docker sur le Docker Hub
- Créer un déploiement Kubernetes
- Créer un service Kubernetes

Pour notre application nous avons décidé d'utiliser Flask qui est un micro-framework de développement web en python.

Voici notre application:

```
flask > flask_app.py > ...
1  from flask import Flask, jsonify
2  app = Flask(__name__)
3
4  @app.route("/", methods=["GET"])
5  def bienvenue():
6      return jsonify({"msg": "Projet ProgWeb de Ziad HASNI et Yacine AHMED YAHIA"})
7
8
9  if __name__ == "__main__":
10     app.run(debug=True)
11
```

Pour créer une image docker de l'application nous créons tout d'abord un Dockerfile:

```
flask > Dockerfile > ...
1  # syntax=docker/dockerfile:1
2  FROM python:3.11.3-slim-buster
3
4  WORKDIR /app
5
6  COPY requirements.txt .
7
8  RUN pip install --upgrade pip
9  RUN pip install -r requirements.txt
10
11 COPY . .
12 ENV FLASK_APP=flask_app.py
13
14 CMD ["python3", "-m", "flask", "run", "--host=0.0.0.0"]
15 EXPOSE 5000
```

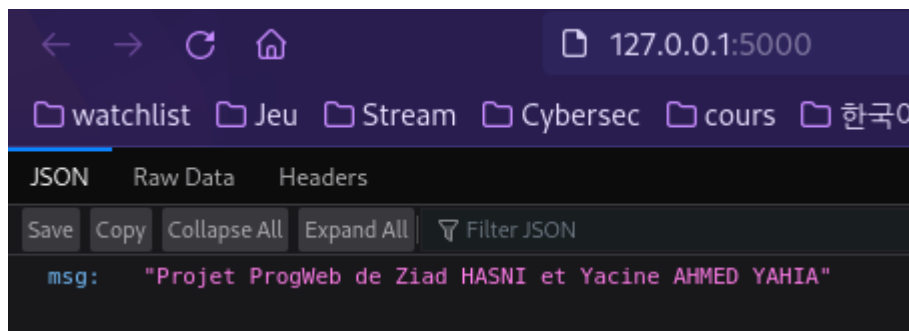
On précise la version de python qu'on utilise, WORKDIR indique le dossier courant dans lequel les commandes seront exécutées, on met à jour pip (le gestionnaire de paquets python) et on installe les prérequis qui auront été indiqués dans le fichier requirements.txt qu'on aura créé auparavant (ici, le seul prérequis est Flask), COPY . . copie les fichiers du répertoire dans lequel se trouve le fichier Dockerfile dans le conteneur Docker, on set ensuite la variable d'environnement FLASK_APP, dans le cas contraire on a eu erreur qui nous dit que l'interpréteur ne trouve pas d'application Flask à lancer. CMD nous permet de run la commande de notre choix, ici on lance l'application flask à l'aide de python en précisant l'host, pour finir on expose le port 5000, que flask utilise pour écouter.

Ensuite on build l'image:

`docker build -t flaskprojet .`

puis on essaye de docker run pour voir si tout fonctionne correctement:

```
(ziad@localhost)~[~/Desktop/ProjetProgWeb/flask]
$ docker run -p 5000:5000 -t flaskprojet
* Serving Flask app 'flask_app.py'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
```



On tag ensuite l'image et on l'upload sur Docker Hub: "docker push ziadhasni/flaskprojet:latest", elle est accessible à ce lien:

<https://hub.docker.com/repository/docker/ziadhasni/flaskprojet/general>

Pour créer un déploiement et un services Kubernetes on crée ensuite un fichier YAML:

```

flask > ! flask_app.yml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: flaskapp
5  spec:
6    replicas: 2
7    selector:
8      matchLabels:
9        app: flaskapp
10   template:
11     metadata:
12       labels:
13         app: flaskapp
14     spec:
15       containers:
16       - name: flaskapp
17         image: ziadhasni/flaskprojet:latest
18         ports:
19         - containerPort: 5000
20
21   ---
22   apiVersion: v1
23   kind: Service
24   metadata:
25     name: flaskapp
26   spec:
27     type: NodePort
28     ports:
29     - port: 5000
30       targetPort: 5000
31     selector:
32       app: flaskapp

```

Puis on utilise la commande `kubectl apply -f flask_app.yml` puis `kubectl get pods`:

```

(ziad@localhost)-[~/Desktop/ProjetProgWeb/flask]
$ kubectl get pods

```

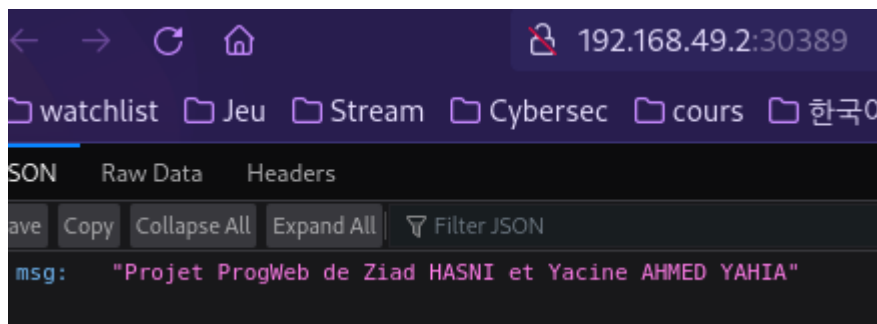
NAME	READY	STATUS	RESTARTS	AGE
flaskapp-6f88896c85-hgxmg	1/1	Running	0	3m13s
flaskapp-6f88896c85-n7scj	1/1	Running	0	3m1s

On peut ainsi si on le souhaite entrer dans le container, on y voit les fichiers (Dockerfile, flask_app.py, flask_app.yml et requirements.txt) qui ont été copiés lors du build de l'image:


```
(ziad@localhost)-[~/Desktop/ProjetProgWeb/flask]
$ kubectl exec -it flaskapp-6f88896c85-hgxmg -- /bin/bash
root@flaskapp-6f88896c85-hgxmg:/app# ls
Dockerfile  __pycache__ flask_app.py flask_app.yml requirements.txt
root@flaskapp-6f88896c85-hgxmg:/app#
```

On a grâce au fichier yaml et NodePort exposé le service:

```
(ziad@localhost)-[~/Desktop/ProjetProgWeb/flask]
$ minikube service flaskapp --url
http://192.168.49.2:30389
```



Réalisation Google Labs



Ziad HASNI
Date d'abonnement : 2024
250 points

Votre profil n'est pas public ni accessible. [Rendre le profil public](#)

Parcours de formation

Activités


Classement

Badges

CoursAtelierQuizJeu

En coursTerminée

Activité	Type	Date de début	Date de fin	Score	Réussie
Développement d'applications : déployer l'application dans Kubernetes Engine – Python	Atelier	19 mars 2024	19 mars 2024	Assessment: 100%	✓
Présentation des ateliers pratiques Google Cloud	Atelier	9 janv. 2024	9 janv. 2024	Assessment: 100%	✓



Yacine AHMED YAHIA
Member since 2024
250 points

Your profile is not public and accessible. [Make profile public](#)

Paths

Activities

Leaderboard

Badges

CourseLabQuizGame

In progressFinished

Activity	Type	Date started	Date finished	Score	Passed
App Dev: Deploying the Application into Kubernetes Engine - Python	Lab	Mar 19, 2024	Mar 19, 2024	Assessment: 100%	✓
A Tour of Google Cloud Hands-on Labs	Lab	Jan 9, 2024	Jan 9, 2024	Assessment: 100%	✓