

TP – Openshift – Déploiement d'une application Flask

<https://serval-concept.com/>

Objectif:

Temps:

Partie 1 : Code python & Dockerfile

- 1- Pour commencer, sur votre machine, commencez par créer la structure suivante dans un dossier (static et templates seront donc des sous-dossier) :

```
Dockerfile
main.py

static
templates
    home.html
```

- 2- Dans main.py, ajoutez le code de base pour utiliser une application flask

```
from flask import Flask, render_template, request, send_file, redirect, url_for, Response, redirect

app = Flask(__name__)

import sys
import os

@app.route('/', methods=['GET', 'POST'])

def Menu():
    return render_template("home.html")

if __name__ == "__main__":
    app.run(debug=True, host='0.0.0.0', port=8001)
```

- 3- Il faut désormais créer le Dockerfile

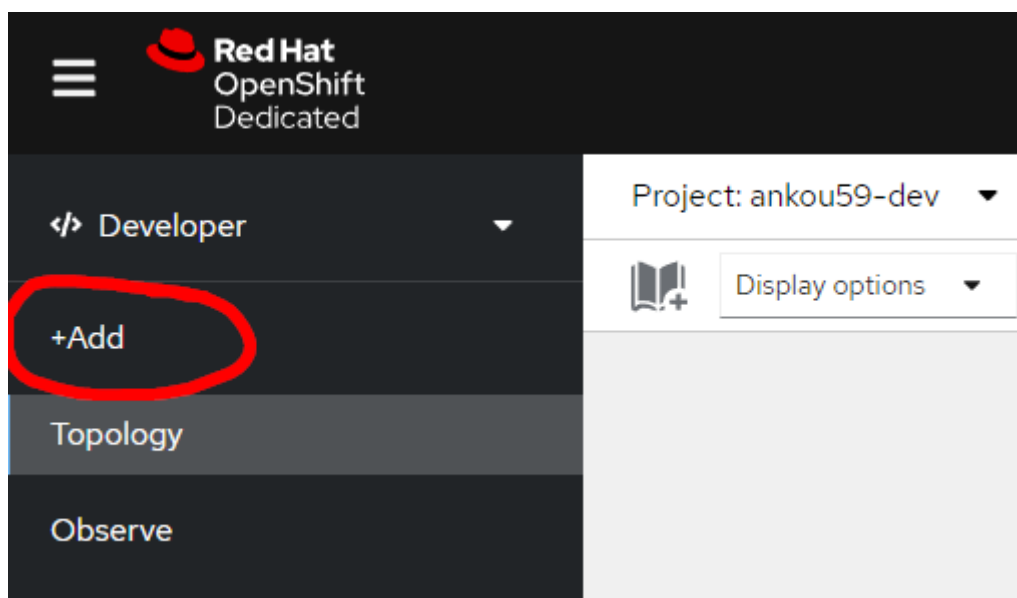
```
1 FROM python:3.9
2 COPY . /app
3 WORKDIR /app
4
5 RUN pip install flask
6
7 ENV PYTHONUNBUFFERED 1
8
9 EXPOSE 8001
10 CMD python3 ./main.py
```

4- Vous allez ajouter du contenu a votre page home.html
Pour ma manip de test, j'ai simplement ajouté : `<h1>Coucou les DO2</h1>`
Mais vous avez le droit d'être plus créatif

5- Allez ensuite sur github et créer un repository.
Placez-y l'ensemble de vos fichiers directement dans la branche main

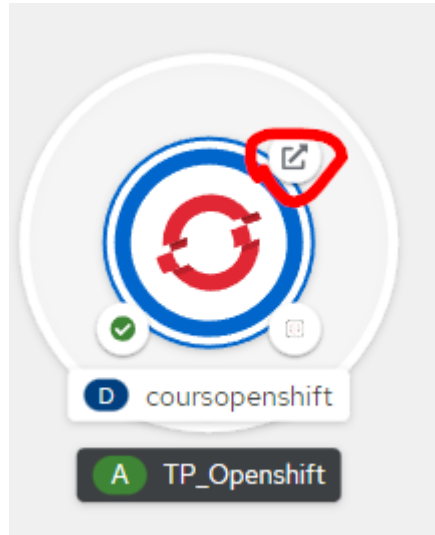
Partie 2 : Déploiement de l'application

- 1) Rendez-vous désormais sur Openshift de redhat.
Connectez-vous a votre sandbox, placez-vous en tant que Developer, et,
après avoir créé un projet, faite +Add



- 2) Sélectionner « Import from git » et ajoutez-y l'URL de votre dépôt
Vous pouvez renommer l'application ici dans la configuration, mais le plus important sera de modifier le Target port par celui utilisé par notre image docker
Une fois tout renseigné correctement, il vous suffira de faire « Create »

- 3) Vous allez arriver sur la vue « topology » avec votre application de visible. Il faudra patienter le temps du build et du déploiement de votre application. Lorsque tout sera ok, vous aurez votre application qui ressemblera à ça et vous pourrez donc accéder à votre page web en cliquant sur le bouton supérieur droit :



- 4) Ouvrez le CLI depuis le navigateur.

A l'aide des commandes du cours, afficher le détail concernant le buildConfig puis le pod

- 5) A l'aide de la commande `oc new-app` et `oc expose`, créer à nouveau un pod avec votre application flask.
`oc expose` sera ici utilisé afin de créer une route. Je vous invite donc à taper la commande afin d'afficher les routes avant et après l'avoir tapé afin de vérifier que tout c'est bien déroulé.