



1) Le Dockerfile	1
1.1) Contenu du fichier Dockerfile	1
1.2) Build & Run de l'image.....	2
1.3) Envoi de l'image sur le Docker HUB.....	2
2) Le fichier de déploiement	2

1) Le Dockerfile

1.1) Contenu du fichier Dockerfile

Voici le contenu de mon Dockerfile :

```
# Utilise l'image officielle Apache comme base
FROM httpd:latest

# Installe les outils nécessaires pour mettre à jour la page web
RUN apt-get update && apt-get install -y nano

# Copie notre fichier de page web modifié dans le répertoire racine du serveur
Apache
COPY index.html /usr/local/apache2/htdocs/index.html

# Expose le port 80 du conteneur pour permettre aux requêtes HTTP d'arriver jus-
qu'au serveur Apache
EXPOSE 80

# Démarre le serveur Apache lorsque le conteneur est lancé
CMD ["httpd-foreground"]
```

Ce Dockerfile me permet d'installer un serveur Apache qui sera exposé sur le port 80 avec une page web « index.html » qui sera copié depuis ma machine locale vers le conteneur afin d'avoir une page web à mon nom. Voici donc le contenu de ma page web :

```
<!DOCTYPE html>
<html>

<head>
  <title>Bienvenue sur le serveur web de Samuel</title>
  <style>
    body {
      text-align: center;
    }
  </style>
</head>
```

```
<body>
  <h1>Bienvenue sur le serveur web de Samuel</h1>
</body>

</html>
```

1.2) Build & Run de l'image

Je peux maintenant lancer la création de mon image :

```
docker build -t apache-sam .
```

Une fois mon image créée je peux la tester localement dans un conteneur sans passer par kubernetes :

```
docker run -d -p 8080:80 apache-sam
```

Si je test maintenant d'aller sur un navigateur sur le port 8080 je peux visualiser ma page web :

192.168.1.30:8080



Bienvenue sur le serveur web de Samuel

1.3) Envoi de l'image sur le Docker HUB

Avant de pouvoir envoyer mon image sur le docker hub je dois lui créer un tag :

```
docker tag apache-sam:latest cookiies/apache-sam:latest
```

Une fois le tag effectué je peux me connecter à mon compte docker :

```
docker login
```

Pour finir je peux envoyer mon image vers le docker hub :

```
docker push cookiies/apache-sam
```

Si je regarde sur le hub je vois bien mon image :



2) Le fichier de déploiement

Avant de me créer un fichier de déploiement il faut que je fasse un cluster kubernetes avec minikube :

```
minikube start --driver=virtualbox --nodes 2
```

Maintenant que mon image est disponible sur le HUB et que mon cluster k3s est fait je peux me créer un fichier de déploiement qui va me permettre de déployer mon application dans un cluster kubernetes :

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: apache-server
spec:
  replicas: 1
  selector:
    matchLabels:
      app: apache-server
  template:
    metadata:
      labels:
        app: apache-server
    spec:
      containers:
        - name: apache-server
          image: cookiies/apache-sam
          ports:
            - containerPort: 80
```

Ce fichier me déploie une application « apache-server » qui se base sur l'image que j'ai envoyé précédemment sur le docker hub

Je peux maintenant le déployer avec kubectl :

```
kubectl apply -f deploy-apache
```

Pour continuer je peux exposer le port 80 de mon conteneur sur ma machine afin de pouvoir y accéder sans avoir besoin de faire « minikube ssh » :

```
kubectl expose deployment apache-server --type=LoadBalancer --port=80
```

Une fois le port exposé je peux utiliser la commande suivante pour lancer un navigateur sur la page de mon application :

```
minikube service apache-server
```

NAMESPACE	NAME	TARGET PORT	URL
default	apache-server	80	http://192.168.59.117:30451

* Ouverture du service default/apache-server dans le navigateur par défaut...



Bienvenue sur le serveur web de Samuel

Si je veux automatiser l'exposition du port à l'extérieur du conteneur je peux ajouter le contenu suivant dans mon fichier de déploiement :

```
apiVersion: v1
kind: Service
metadata:
  name: apache-service
  labels:
    app: apache-server
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30080
  selector:
    app: apache-server
```

Puis relancer le déploiement :

```
kubectl apply -f deploy-apache
```

Tout fonctionne bien jusqu'à cette étape. L'exposition du port se fait bien mais lorsque j'essaye d'accéder à la page web le chargement se fait à l'infini