

Le premier exercice porte sur docker compose et compte pour 70% de la note. Le second exercice porte sur kubernetes et compte pour 30% de la note. Vous devrez créer un répertoire dont le nom est VOTRE_NOM-VOTRE_PRENOM. vous y placerez le répertoire `store` contenant le code que je vous fournis. Quand je noterai votre travail, je veux pouvoir me placer dans VOTRE_NOM-VOTRE_PRENOM et exécuter `docker compose build`, suivi de `docker compose up -d`, et pouvoir me connecter à votre application sur le port 5000 de mon localhost.

Le rendu: doit être une archive appelée VOTRE_NOM-VOTRE_PRENOM.tgz contenant le répertoire VOTRE_NOM-VOTRE_PRENOM.

Exercice 1. Docker compose Pour cet exercice je vous donne le code d'une application web en python. Cette application à besoin des modules `flask` et `flask-sqlalchemy` que l'on peut installer avec `pip`. Elle a besoin d'une base de données mysql et utilise les variables d'environnement suivantes pour s'y connecter:

- STORE_USER
- STORE_PASSWORD
- STORE_SERVER: l'ip ou nom DNS du server mysql
- STORE_DATABASE: le nom de la base de données dans ce serveur

Pour lancer l'application, placez vous dans le répertoire VOTRE_NOM-VOTRE_PRENOM (donc le parent du répertoire `store`) et lancez la commande suivante:

```
| flask --app store run --host "0.0.0.0"
```

Pour cet exercice, vous pourrez mettre les valeurs des variables d'environnement en clair dans les fichiers `yaml`. Vous ne devez pas modifier le code que je vous fournis. Vous devrez écrire un fichier `Dockerfile` pour créer une image pour l'appli web que je vous donne, et un fichier `compose.yml` pour créer une app à 2 services (l'appli web et la bd mysql dont elle a besoin)

Exercice 2. Kubernetes Donnez les fichiers `yaml` nécessaires pour déployer l'app dans un cluster kubernetes. Pour cet exercice, vous suivrez les bonnes pratiques et utiliserez donc `configMap` et `Secret` où nécessaire.