

## TP Kubernetes — Déploiement de l'application **PetClinic**

### Objectif :

Déployer une application Java PetClinic composée d'un backend Spring Boot et d'une base de données PostgreSQL sur un cluster Kubernetes. Vous utiliserez des objets `Deployment`, `Service`, `ConfigMap` et `Secret` pour organiser et sécuriser la configuration.



## Contexte :

Vous êtes en charge de déployer l'application **PetClinic** dans un environnement Kubernetes. Cette application utilise une base de données PostgreSQL. L'application récupère sa configuration via une `ConfigMap`, tandis que la base de données utilise un `Secret` pour conserver ses informations sensibles.

## Livrables attendus :

Vous devez fournir les **manifestes YAML** nécessaires pour faire fonctionner :

- Un **Dockerfile** pour générer l'image de votre projet.
- Une **base de données PostgreSQL** configurée avec un `Secret`
- Un **Service** pour exposer cette base de données
- Une application **PetClinic** en conteneur (image déjà existante)
- Une `ConfigMap` utilisée pour injecter les variables d'environnement dans l'application
- Un `Service` de type `NodePort` pour accéder à l'application via un navigateur

## Contraintes :

- Le nom de la base de données est `petclinic`
- L'utilisateur est `user` et le mot de passe est `pass`
- Le pod de la base de données doit être nommé `demo-db`
- Le pod de l'application doit être nommé `petclinic`
- L'image de l'application est : `<votre_registry>/petclinic:v1`
- Le conteneur écoute sur le port `8080` , mais l'application doit être exposée sur le port `80` depuis l'extérieur

## **Recommandations :**

- Utilisez des `Secret` pour toutes les données sensibles côté base de données
- Utilisez une `ConfigMap` pour injecter les paramètres de l'application
- Vérifiez bien les labels et selectors pour assurer la bonne communication entre les services et les pods



**Pour tester :**

- Accédez à l'application via le **NodePort** exposé