

# TP1 - Cours de virtualisation avancée - *Kubernetes*, introduction

## Attention:

Comme pour le TP de la semaine dernière, vous allez travailler dans une machine virtuelle VDN.

N'oubliez pas de lancer

```
export no_proxy="127.0.0.1,.vdn,localhost,192.168.49.1/24,10.96.0.0/12"
```

à chaque connexion ssh !

Pour lancer *Minikube*, lancez la commande `./minikube start` (attention vous aviez probablement laissé les binaires *minikube* et *kubectl* dans le dossier du TP2).

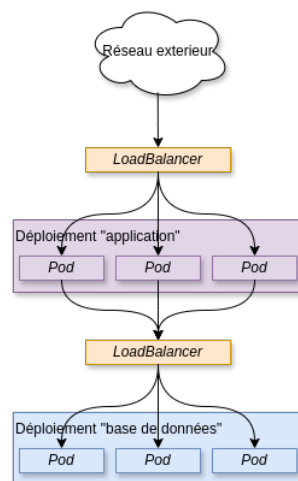
N'oubliez pas aussi de laisser un autre terminal tourner avec la commande `minikube tunnel ;)`

L'objectif de ce TP est de mettre en place une architecture plus compliquée que la semaine dernière. Cette architecture mettra en place les éléments vus la semaine dernière ainsi que les éléments vus cette semaine:

- *Deployment*
- *Service*
- *ConfigMap*
- *PersistentVolume*
- *PersistentVolumeClaim*

Utilisations des ressources:

- Les variables d'environnement des conteneurs doivent être stockées dans une *ConfigMap*
- La base de données doit utiliser un volume pour sauvegarder ses fichiers



# 1 Images à utiliser

- `evanespen/database` -> `https://hub.docker.com/r/evanespen/database`
- `evanespen/application` -> `https://hub.docker.com/r/evanespen/application`

## Attention:

Lisez bien les description des images !

# 2 Notes

Pour faire communiquer deux *Pods* via un service, vous pouvez utiliser le nom du service en tant que nom de domaine. Par exemple, si j'ai un service "backend-svc" qui expose en HTTP sur le port 8080, je peux y accéder depuis les *Pods* via l'URL `http://backend-svc:8080`.

# 3 Aide

Référez vous au cours et à ces documentations pour y parvenir (ce sont bien des liens):

- *Service*
- *Pod*
- *Deployment*
- *PV et PVC*

Amusez vous bien ;)