

# Compte Rendu - TP1 Kubernetes

Étudiant : Jules DEBÉE

## Introduction

Ce document présente les étapes suivies lors du TP Kubernetes, comprenant la création de namespaces, le déploiement d'un service NGINX en tant que reverse proxy, et la vérification des événements et des statuts des pods.

## Création des Namespaces

### Commandes Utilisées

Pour créer les namespaces, des fichiers YAML ont été utilisés, suivis des commandes :

1. Créer le namespace **front** :  
`kubectl apply -f front/namespace.yml`
2. Créer le namespace **middle** :  
`kubectl apply -f middle/namespace.yml`
3. Créer le namespace **database** :  
`kubectl apply -f database/namespace.yml`

### Vérification des Namespaces

Pour visualiser la liste des namespaces :

```
kubectl get namespaces
```

```
jules@LEGIONJULES:~$ kubectl get namespace
NAME                STATUS    AGE
database            Active    45m
default             Active    77m
front               Active    45m
kube-node-lease     Active    77m
kube-public         Active    77m
kube-system         Active    77m
middle              Active    45m
```

# Déploiement de NGINX

## Commandes Utilisées pour le Déploiement

Déploiement des ressources NGINX :

```
kubectl apply -f front/nginx-deployment.yml  
kubectl apply -f front/nginx-service.yaml
```

## Vérification du Déploiement

Pour vérifier le bon démarrage des pods et leurs conteneurs :

**Vérifier les événements :**

```
- kubectl get events -n front
```

**Vérifier le statut des pods :**

```
- kubectl get pods -n front
```

## Port Forwarding

Pour accéder au service NGINX, le port-forwarding a été configuré :

```
nohup kubectl port-forward -n front svc/nginx-svc 8080:8383 &
```

## Vérification des Processus Actifs

Pour observer les processus actifs :

```
ps aux | grep kubectl
```

## Arrêt du Processus

Pour fermer le processus, récupérer le PID et utiliser :

```
kill <PID>
```

Exemple avec photo si dessous : "kill 41248"

```
jules@LEGIONJULES:~$ nohup kubectl port-forward -n front svc/nginx-svc 8080:8383 &  
[1] 41248  
jules@LEGIONJULES:~$ nohup: ignoring input and appending output to 'nohup.out'  
  
jules@LEGIONJULES:~$ ps aux | grep kubectl  
jules      41248  1.8  0.2 5473840 44492 pts/4    Sl   11:24   0:00 kubectl port-forward -n front svc/nginx-svc 8080:8383  
jules      41287  0.0  0.0  4028   1976 pts/4    S+   11:24   0:00 grep --color=auto kubectl  
jules@LEGIONJULES:~$ kill 41248
```

# Réponses aux Questions avec images

## Question Image 1

Les pointillés violets dans l'image représentent une politique de réseau dans Kubernetes. Cette politique contrôle le flux de trafic autorisé entre différents pods au sein d'un cluster Kubernetes, notamment entre les pods 1 et 2.

## Question Image 2

Les pointillés dans l'image correspondent aux limites des Pods dans une architecture Kubernetes. Un Pod est la plus petite unité déployable dans Kubernetes et peut contenir un ou plusieurs conteneurs. Dans cette image, il y a deux Pods, chacun contenant un conteneur NGINX version 1.27.1-alpine.

---