

# Lab 2 : Créer un Premier Pod et Manipuler son Environnement d'Orchestration K8s

COMPTE RENDU

YOUSSEF BEN GHORBEL

# Table des matières

<b>Objectifs .....</b>	<b>2</b>
<b>Pré-requis .....</b>	<b>2</b>
<b>Étapes.....</b>	<b>2</b>
<b>1. Configuration de l'environnement.....</b>	<b>2</b>
<b>2. Afficher les Namespaces.....</b>	<b>2</b>
<b>3. Lister les Pods :.....</b>	<b>3</b>
<b>4. Création d'un Pod .....</b>	<b>3</b>
<b>5. Exposition du Pod.....</b>	<b>3</b>
<b>6. Exécution de commandes dans un Pod .....</b>	<b>4</b>
<b>7. Vérification des modifications .....</b>	<b>4</b>
<b>8. Suppression du Pod.....</b>	<b>4</b>
<b>9. Utilisation d'un fichier de configuration YAML.....</b>	<b>4</b>
<b>10. Création et gestion des Namespaces .....</b>	<b>5</b>

## Objectifs

- Manipuler un Pod (créer, se connecter, supprimer)
- Écrire un fichier de configuration pour décrire un Pod
- Accéder à un Pod via le réseau
- Organiser les Pods via les Namespaces

## Pré-requis

- Un fichier k3s.yaml à la racine du dossier kubernetes-tutorial/
- Initialiser la variable d'environnement :

```
export KUBECONFIG=$PWD/k3s.yaml
```

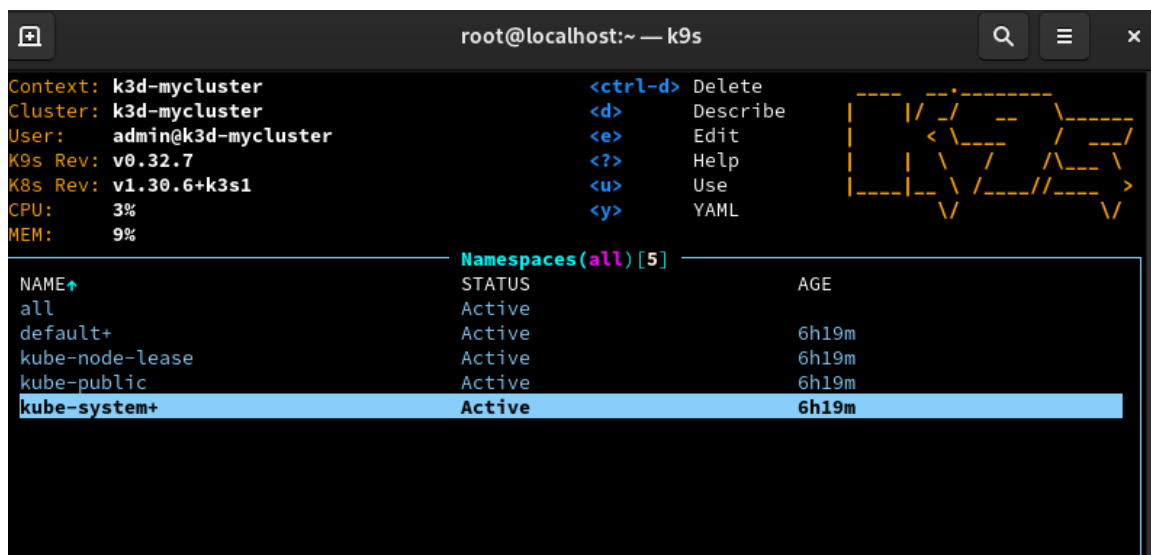
## Étapes

### 1. Configuration de l'environnement

- Ouvrir deux terminaux : un pour K9s et un pour kubectl.
- Dans le terminal K9s, exécuter :

### 2. Afficher les Namespaces

- Dans K9s, utilisez la commande `:namespaces` pour voir les Namespaces existants.



```
root@localhost:~ — k9s
Context: k3d-mycluster
Cluster: k3d-mycluster
User: admin@k3d-mycluster
K9s Rev: v0.32.7
K8s Rev: v1.30.6+k3s1
CPU: 3%
MEM: 9%

<ctrl-d> Delete
<d> Describe
<e> Edit
<?> Help
<u> Use
<y> YAML

Namespaces(all) [5]
NAME+ STATUS AGE
all Active 6h19m
default+ Active 6h19m
kube-node-lease Active 6h19m
kube-public Active 6h19m
kube-system+ Active 6h19m
```

### 3. Lister les Pods :

- Dans le terminal kubectl, exécuter :

```
kubectl get pods --all-namespaces
```

```
[root@localhost ~]# kubectl get pods --all-namespaces
NAMESPACE   NAME                                     READY   STATUS    RESTARTS   AGE
kube-system  coredns-7b98449c4-ctkkw               1/1     Running   0           62m
kube-system  helm-install-traefik-6s5d2             0/1     Completed 2           6h32m
kube-system  local-path-provisioner-595dcfc56f-dh4jr 1/1     Running   1 (61m ago) 6h32m
kube-system  metrics-server-cdcc87586-kt28w         1/1     Running   1 (61m ago) 6h32m
kube-system  svclb-traefik-c5e3a029-fsnsg           2/2     Running   2 (61m ago) 6h25m
kube-system  svclb-traefik-c5e3a029-q7tjv           2/2     Running   2 (61m ago) 6h25m
kube-system  svclb-traefik-c5e3a029-ztp56           2/2     Running   2 (61m ago) 6h25m
kube-system  traefik-d7c9c5778-824f8                1/1     Running   1 (61m ago) 6h25m
[root@localhost ~]#
```

### 4. Création d'un Pod

- Créer un Pod avec Nginx :

```
kubectl run myfirstpod --image=nginx:latest
```

```
[root@localhost ~]# kubectl run myfirstpod --image=nginx:latest
pod/myfirstpod created
```

- Vérifier la création :

```
kubectl get pods
```

```
[root@localhost ~]# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
myfirstpod    1/1     Running   0           3m10s
[root@localhost ~]#
```

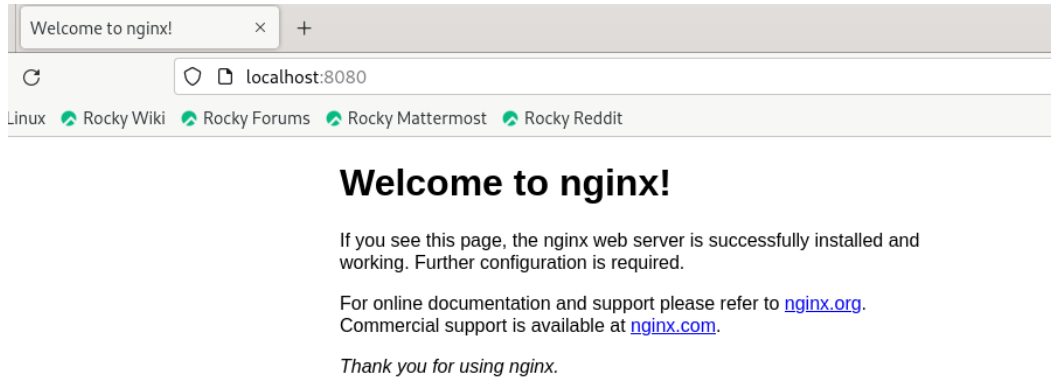
### 5. Exposition du Pod

- Rediriger les ports pour accéder au Pod :

```
kubectl port-forward myfirstpod 8080:80
```

```
[root@localhost ~]# kubectl port-forward myfirstpod 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
Handling connection for 8080
```

- Accéder à Nginx via un navigateur à `http://localhost:8080`.



## 6. Exécution de commandes dans un Pod

- Connexion au conteneur Nginx :

```
kubectl exec -it myfirstpod -- /bin/bash
```

```
root@myfirstpod:/# ls
bin    dev          docker-entrypoint.sh  home  lib64  mnt  proc  run  srv  tmp  var
boot  docker-entrypoint.d  etc          lib   media  opt  root  sbin sys  usr
```

- Modifier la page HTML :

```
echo "Modification de la page web par défaut" >
/usr/share/nginx/html/index.html
```

## 7. Vérification des modifications

- Utiliser cURL pour vérifier :

```
curl http://localhost:8080
```

```
[root@localhost ~]# curl http://localhost:8080
Modification de la page web par défaut
[root@localhost ~]#
```

## 8. Suppression du Pod

- Supprimer le Pod :

```
kubectl delete pods myfirstpod
```

## 9. Utilisation d'un fichier de configuration YAML

- Créer le fichier mypod.yaml :

```

apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: mycontainer-1
    image: nginx:latest
    ports:
    - containerPort: 80
  - name: mycontainer-2
    image: alpine:latest
    command: ["watch", "wget", "-qO-", "localhost"]

```

- Appliquer le fichier :

```
kubectl apply -f mypod.yaml
```

```

[root@localhost exercice1-pod-tools]# kubectl apply -f mypod.yaml
pod/mypod created
[root@localhost exercice1-pod-tools]#

```

## 10. Création et gestion des Namespaces

- Supprimer le Pod créé précédemment :

```
kubectl delete pods mypod
```

```

[root@localhost exercice1-pod-tools]# kubectl delete pods mypod
pod "mypod" deleted

```

- Créer un Namespace

```

apiVersion: v1
kind: Namespace
metadata:
  name: mynamespaceexercice1

```

- Appliquer le Namespace :

```
kubectl apply -f mynamespaceexercice1.yaml
```

```

error: the path "exercice1-pod-tools/mynamespaceexercice1.yaml" does not exist
[root@localhost exercice1-pod-tools]# kubectl apply -f mynamespaceexercice1.yaml
namespace/mynamespaceexercice1 created
[root@localhost exercice1-pod-tools]#

```

- Recréer le Pod dans le Namespace

```
kubectl apply -f mypod.yaml -n mynamespaceexercice1
```

- Supprimer le Namespace :

```
kubectl delete namespace mynamespaceexercice1
```