

Lab 1 : Installer un cluster K3d

COMPTE RENDU

YOUSSEF BEN GHORBEL

Table des matières

Introduction.....	2
Objectifs	2
Étapes à suivre	2
1. Installation de kubectl	2
2. Installation de K3d	2
3. Création du cluster K3d.....	3
4. Vérification des conteneurs Docker.....	3
5. Configuration de l'accès au cluster	3
6. Configuration de kubectl.....	3
7. Installation de K9s (facultatif)	4

Introduction

Ce document décrit la mise en place d'un cluster Kubernetes à l'aide de K3d, qui utilise la distribution légère K3s. K3d permet de déployer un cluster multi-nœuds sous Docker, offrant des performances optimales pour le développement.

Objectifs

1. Installer les outils de gestion **kubectl** et **K9s**.
2. Créer un **cluster K3s**

Étapes à suivre

1. Installation de kubectl

Installation via le gestionnaire de packages

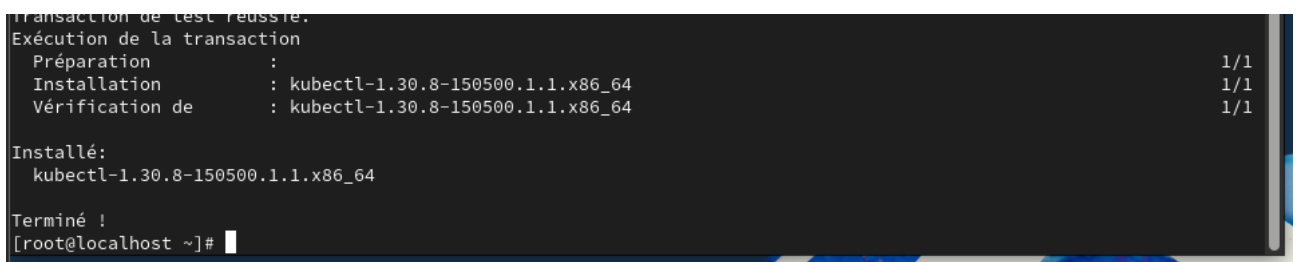
```
# cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]

name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/ enabled=1

gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key

EOF
```

yum install -y kubectl



```
Transaction de test réussie.
Exécution de la transaction
  Préparation      : kubectl-1.30.8-150500.1.1.x86_64      1/1
  Installation     : kubectl-1.30.8-150500.1.1.x86_64      1/1
  Vérification de  : kubectl-1.30.8-150500.1.1.x86_64      1/1

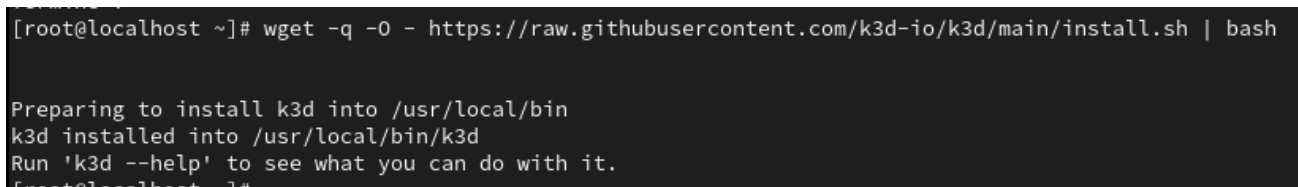
Installé:
  kubectl-1.30.8-150500.1.1.x86_64

Terminé !
[root@localhost ~]#
```

2. Installation de K3d

Pour installer K3d, exécutez la commande suivante :

```
wget -q -O - https://raw.githubusercontent.com/k3d-io/k3d/main/install.sh | bash
```



```
[root@localhost ~]# wget -q -O - https://raw.githubusercontent.com/k3d-io/k3d/main/install.sh | bash

Preparing to install k3d into /usr/local/bin
k3d installed into /usr/local/bin/k3d
Run 'k3d --help' to see what you can do with it.
[root@localhost ~]#
```

Vérifiez l'installation :

```
k3d version
```

```
[root@localhost ~]# k3d version
k3d version v5.7.5
k3s version v1.30.6-k3s1 (default)
[root@localhost ~]#
```

3. Création du cluster K3d

Créer un cluster Kubernetes nommé mycluster avec un nœud maître et deux nœuds de travail :

```
k3d cluster create mycluster --agents 2 --servers 1
```

4. Vérification des conteneurs Docker

Pour consulter les conteneurs créés : **docker ps**

```
[root@localhost ~]# docker ps
CONTAINER ID   IMAGE
f40451b586b8   ghcr.io/k3d-io/k3d-proxy:5.7.5
be4380d61b0f   rancher/k3s:v1.30.6-k3s1
86be70b5d804   rancher/k3s:v1.30.6-k3s1
0000778cf100   rancher/k3s:v1.30.6-k3s1
[root@localhost ~]#
```

5. Configuration de l'accès au cluster

Récupérez le fichier d'accès au cluster :

```
k3d kubeconfig get mycluster > k3s.yaml
```

6. Configuration de kubectl

Configurez kubectl pour utiliser le fichier d'accès :

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

```
kubectl version --client
```

Vérifiez l'état des nœuds :

```
kubectl top nodes
```

```
[root@localhost ~]# kubectl top nodes
NAME                                CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
k3d-mycluster-agent-0              88m          4%     238Mi           6%
k3d-mycluster-agent-1              72m          3%     206Mi           5%
k3d-mycluster-server-0             46m          2%     588Mi           16%
[root@localhost ~]#
```

7. Installation de K9s (facultatif)

Pour installer K9s sur Linux :

```
wget https://github.com/derailed/k9s/releases/download/v0.25.15/k9s_Linux_x86_64.tar.gz
```

```
tar xzf k9s_Linux_x86_64.tar.gz
```

Testez l'installation de K9s :

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

```
k9s
```