Création d'un cluster k3s mono-noeud sur Rocky Linux

Mise en place des pré-requis

Désactiver firewalld

```
systemctl stop firewalld
systemctl disable firewalld
```

Désactiver selinux en modifiant le fichier /etc/sysconfig/selinux

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
# See also:
# https://access.redhat.com/documentation/en-
us/red_hat_enterprise_linux/9/html/using_selinux/changing-selinux-states-and-modes_using-
selinux#changing-selinux-modes-at-boot-time_changing-selinux-states-and-modes
#
# NOTE: Up to RHEL 8 release included, SELINUX=disabled would also
# fully disable SELinux during boot. If you need a system with SELinux
# fully disabled instead of SELinux running with no policy loaded, you
# need to pass selinux=0 to the kernel command line. You can use grubby
# to persistently set the bootloader to boot with selinux=0:
#
#    grubby --update-kernel ALL --args selinux=0
#
# To revert back to SELinux enabled:
#
#    grubby --update-kernel ALL --remove-args selinux
#
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected,
```

```
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Installer les packages manquants suivants

```
dnf install -y git tar bash-completion
```

Modifier les paramètres kernel suivants en créant le fichier /etc/sysctl.d/99-tunning.conf

```
#Disable IP V6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1

#tunning fs
fs.aio-max-nr = 1048576
fs.file-max = 6815744

#Limit Swap
vm.swappiness = 10

vm.max_map_count = 262144
vm.overcommit_memory = 1

net.core.somaxconn=65535

#Tuning IPV4
net.ipv4.ip_local_port_range = 10000 65500
net.ipv4.tcp_keepalive_time=30
net.ipv4.tcp_keepalive_intvl=30
net.ipv4.tcp_keepalive_probes=10
```

Rebooter le server

Installation de kubectl et d'un kube k3s

Installation de kubectl

Télécharger le binaire kubectl et le déposer dans /usr/local/bin/

```
curl -LO https://dl.k8s.io/release/$(curl -Ls
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl
chmod +x ./kubectl
sudo mv ./kubectl /usr/local/bin/kubectl
```

Modifier le bashrc pour ajouter /usr/local/bin dans la variable $PATH

```
...
...
export PATH=$PATH:/usr/local/bin
...
```

Ajouter la completion sur kubectl

```
echo 'source <(kubectl completion bash)' >>~/.bashrc
```

Vérifier que le binaire est opérationnel

```
kubectl version --client
```

Installation de K3S sans le composant “Traeffik” et en choisissant un CIDR qui n'overlap pas un réseau existant

```
curl -sfL https://get.k3s.io | K3S_KUBECONFIG_MODE="644" INSTALL_K3S_EXEC="--cluster-
cidr=192.168.255.0/24 --disable=traefik" sh -
```

A la fin du déploiement, créer le fichier d'authentification / connection au cluster kubernetes pour le client kubectl

```
mkdir ~/.kube
cp /etc/rancher/k3s/k3s.yaml ~/.kube/config
```

Vérifier que le kube est correctement démarrer

```
kubectl get pods --all-namespaces
NAMESPACE     NAME                                      READY   STATUS    RESTARTS   AGE
kube-system   coredns-0587-fbcd5-sdfsf                  1/1     Running   0          84s
kube-system   local-path-provisioner-fdsfsfs-r6pbm      1/1     Running   0          84s
```

Created 2024-06-21 15:05:21 UTC by NoNo
Updated 2024-06-21 15:33:07 UTC by NoNo