

Lab01 – Créer un cluster avec Kubeadm

Dans ce Lab, vous apprendrez à créer un cluster à l'aide de kubeadm. Le cluster contiendra un seul nœud de plan de contrôle nommé « kube-control-plane » et deux nœuds de travail nommés « kube-node1 » et « kube-node2 ».

Vous pouvez trouver une description complète sur le déploiement du cluster kubernetes dans la documentation officielle de Kubernetes

(<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/>).

Initialisation du nœud du plan de contrôle

1. Git clone le dépôt <https://github.com/brahimhamdi/k8s-lab>, puis entrez dans le répertoire k8s-lab.

```
brahim@Training:~$ git clone https://github.com/brahimhamdi/k8s-lab
Clonage dans 'k8s-lab'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 29 (delta 8), reused 29 (delta 8), pack-reused 0
Réception d'objets: 100% (29/29), 10.64 Kio | 265.00 Kio/s, fait.
Résolution des deltas: 100% (8/8), fait.
brahim@Training:~$ cd k
bash: cd: k: Aucun fichier ou dossier de ce type
brahim@Training:~$ cd k8s-lab
brahim@Training:~/k8s-lab$
```

2. Déployez l'environnement vagrant à l'aide de la commande « vagrant up ».

```
brahim@Training:~/k8s-lab$ vagrant up
Bringing machine 'kube-control-plane' up with 'virtualbox' provider...
Bringing machine 'kube-node1' up with 'virtualbox' provider...
Bringing machine 'kube-node2' up with 'virtualbox' provider...
==> kube-control-plane: Importing base box 'generic/ubuntu2004'...
==> kube-control-plane: Matching MAC address for NAT networking...
==> kube-control-plane: Checking if box 'generic/ubuntu2004' version '4.2.14' is up to date...
==> kube-control-plane: A newer version of the box 'generic/ubuntu2004' for provider 'virtualbox' is
==> kube-control-plane: available! You currently have version '4.2.14'. The latest is version
==> kube-control-plane: '4.2.16'. Run 'vagrant box update' to update.
==> kube-control-plane: Setting the name of the VM: k8s-lab_kube-control-plane_1689795061604_38153
==> kube-control-plane: Clearing any previously set network interfaces...
==> kube-control-plane: Preparing network interfaces based on configuration...
    kube-control-plane: Adapter 1: nat
    kube-control-plane: Adapter 2: hostonly
==> kube-control-plane: Forwarding ports...
    kube-control-plane: 22 (guest) => 2222 (host) (adapter 1)
==> kube-control-plane: Running 'pre-boot' VM customizations...
==> kube-control-plane: Booting VM...
==> kube-control-plane: Waiting for machine to boot. This may take a few minutes...
    kube-control-plane: SSH address: 127.0.0.1:2222
    kube-control-plane: SSH username: vagrant
    kube-control-plane: SSH auth method: private key
    kube-control-plane: Warning: Connection reset. Retrying...
```

...

- Installez Flannel à l'aide de la commande `kubectl apply -f kube-flannel.yml`. Flannel mettra en place le réseau qui va interconnecter les pods à travers le cluster Kubernetes.

```
vagrant@kube-control-plane:~$ kubectl apply -f kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
vagrant@kube-control-plane:~$
vagrant@kube-control-plane:~$ _
```

- Vérifiez que le nœud de plan de contrôle est « Prêt » en exécutant la commande «`kubectl get nodes -owide` ».

```
vagrant@kube-control-plane:~$ kubectl get nodes -o wide
NAME                                STATUS    ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION   CONTAINER
kube-control-plane                 Ready    control-plane   11m   v1.27.4   192.168.56.10 <none>        Ubuntu 20.04.5 LTS  5.4.0-139-generic   containerd://1.6.21
vagrant@kube-control-plane:~$ _
```

- Copiez la commande `kubeadm join ...` générée par la commande `sudo kubeadm token create --print-join-command`, puis quittez la VM à l'aide de la commande `exit`.

```
vagrant@kube-control-plane:~$ sudo kubeadm token create --print-join-command
kubeadm join 192.168.56.10:6443 --token 871ibd.wc0x7dqpbrjj77a --discovery-token-ca-cert-hash sha256:8f97c9c18c7e0ad440d4a7e07536be4197ce19b7ab604e9bdf461a51ab7e56c9
vagrant@kube-control-plane:~$
vagrant@kube-control-plane:~$ exit
logout
brahim@Training:~/k8s-lab$ _
```

Rejoindre les nœuds au cluster

- Shell dans la VM kube-node1 à l'aide de la commande «`vagrant ssh kube-node1`». Rejoignez ce nœud au cluster à l'aide de la commande «`kubeadm join ...`» générée précédemment.

```
brahim@Training:~/k8s-lab$ vagrant ssh kube-node1
vagrant@kube-node1:~$ sudo kubeadm join 192.168.56.10:6443 --token 871ibd.wc0x7dqpbrjj77a --discovery-token-ca-cert-hash sha256:8f97c9c18c7e0ad440d4a7e07536be4197ce19b7ab604e9bdf461a51ab7e56c9
[preFlight] Running pre-flight checks
[preFlight] Reading configuration from the cluster...
[preFlight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
vagrant@kube-node1:~$ _
```

10. Vérifiez, avec la commande « `kubectl get nodes` », que le nœud `kube-node1` est à l'état « Prêt »

```
vagrant@kubernetes1:~$ exit
logout
brahm@Training:~/k8s-lab$ vagrant ssh kube-control-plane
Last login: Wed Jul 19 20:25:43 2023 from 10.0.2.2
vagrant@kubernetes1:~$ kubectl get nodes -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-ENGINE
kube-control-plane	Ready	control-plane	26m	v1.27.4	192.168.56.10	<none>	Ubuntu 20.04.5 LTS	5.4.0-139-generic	containerd
kube-node1	Ready	<none>	2m2s	v1.27.4	192.168.56.11	<none>	Ubuntu 20.04.5 LTS	5.4.0-139-generic	containerd

```
vagrant@kubernetes1:~$
```

11. Répétez les mêmes étapes sur le nœud « `kube-node2` ».

```
vagrant@kubernetes1:~$ exit
logout
brahm@Training:~/k8s-lab$ vagrant ssh kube-node2
vagrant@kubernetes2:~$ sudo kubeadm join 192.168.56.10:6443 --token 871lbd.wc0x7dqpbrjj77a --discovery-token-ca-cert-hash sha256:8f97c9c18c7e0ad440d4a7e07536be4197ce19b7ab604e9bdf461a51ab7e56c9
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiser and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

vagrant@kubernetes2:~$
vagrant@kubernetes2:~$ exit
logout
brahm@Training:~/k8s-lab$ vagrant ssh kube-control-plane
Last login: Wed Jul 19 20:26:32 2023 from 10.0.2.2
vagrant@kubernetes1:~$ kubectl get nodes -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-ENGINE
kube-control-plane	Ready	control-plane	29m	v1.27.4	192.168.56.10	<none>	Ubuntu 20.04.5 LTS	5.4.0-139-generic	containerd
kube-node1	Ready	<none>	5m10s	v1.27.4	192.168.56.11	<none>	Ubuntu 20.04.5 LTS	5.4.0-139-generic	containerd
kube-node2	NotReady	<none>	100s	v1.27.4	192.168.56.12	<none>	Ubuntu 20.04.5 LTS	5.4.0-139-generic	containerd

```
vagrant@kubernetes1:~$
```

Vérification de l'installation

12. Vérifiez que tous les nœuds ont été correctement enregistrés et sont dans l'état « Prêt ».

```
vagrant@kubernetes1:~$ kubectl get nodes -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-ENGINE
kube-control-plane	Ready	control-plane	30m	v1.27.4	192.168.56.10	<none>	Ubuntu 20.04.5 LTS	5.4.0-139-generic	containerd
kube-node1	Ready	<none>	6m39s	v1.27.4	192.168.56.11	<none>	Ubuntu 20.04.5 LTS	5.4.0-139-generic	containerd
kube-node2	Ready	<none>	3m9s	v1.27.4	192.168.56.12	<none>	Ubuntu 20.04.5 LTS	5.4.0-139-generic	containerd

```
vagrant@kubernetes1:~$
```

13. Vérifiez que le cluster fonctionne correctement et que tous les pods système s'exécutent dans l'espace de nom kube-system, ainsi que les pods du Flannel dans l'espace de noms kube-flannel.

```
vagrant@kube-control-plane:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.56.10:6443
CoreDNS is running at https://192.168.56.10:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
vagrant@kube-control-plane:~$ 
vagrant@kube-control-plane:~$ kubectl get pod --namespace kube-system
NAME                                READY    STATUS    RESTARTS   AGE
coredns-5d78c9869d-7s5rw            1/1     Running   0           32m
coredns-5d78c9869d-jq2zl            1/1     Running   0           32m
etcd-kube-control-plane             1/1     Running   1           32m
kube-apiserver-kube-control-plane    1/1     Running   1           32m
kube-controller-manager-kube-control-plane 1/1     Running   0           32m
kube-proxy-8n9tb                    1/1     Running   0           32m
kube-proxy-c9hrf                    1/1     Running   0           8m45s
kube-proxy-vlbcx                    1/1     Running   0           5m15s
kube-scheduler-kube-control-plane    1/1     Running   1           33m
vagrant@kube-control-plane:~$ 
vagrant@kube-control-plane:~$ kubectl get pod --namespace kube-flannel
NAME                                READY    STATUS    RESTARTS   AGE
kube-flannel-ds-f8mgc               1/1     Running   0           24m
kube-flannel-ds-pqcdr               1/1     Running   0           9m8s
kube-flannel-ds-vphv9               1/1     Running   0           5m38s
vagrant@kube-control-plane:~$ 
vagrant@kube-control-plane:~$
```