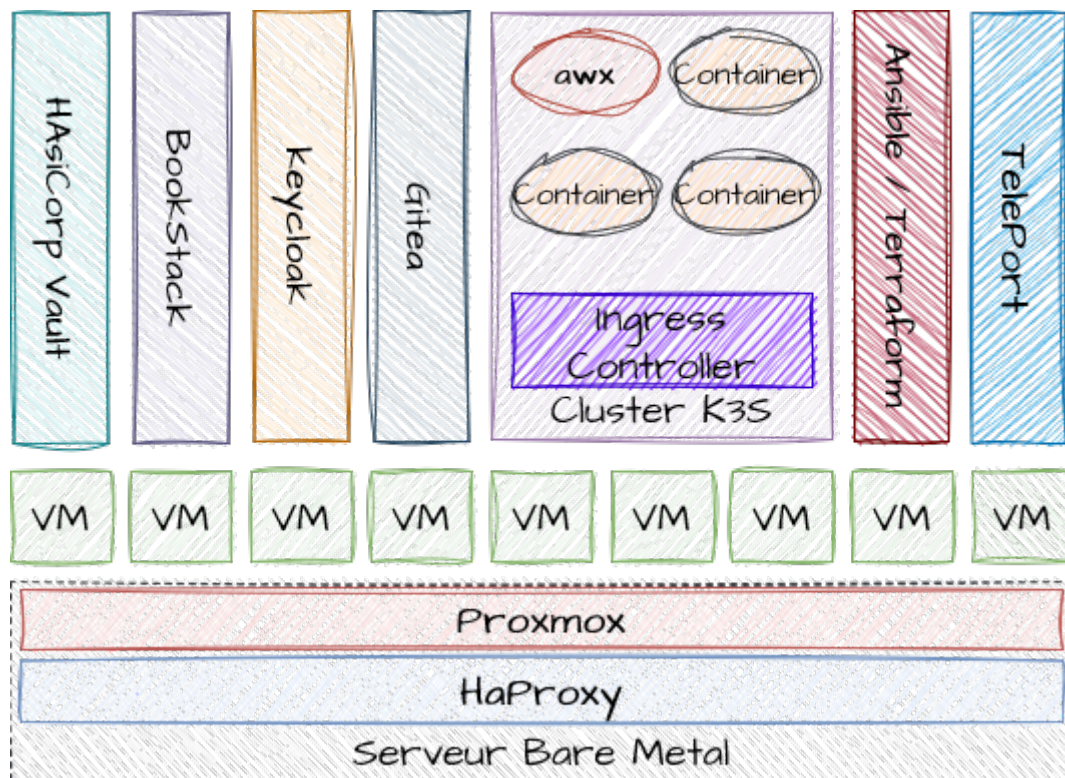


HomeLab

- [Architecture](#)
- [Création et configuration du serveur Proxmox](#)
- [Création d'un cluster K8S sur proxmox](#)

Architecture

Architecture cible envisagée



Principe du Homelab Ops

Le principe du homelab que je souhaite mettre en place est la mise en place d'un outillage standard utilisé par une équipe Ops pour la gestion du "Build" et du "Run" d'un SI.

Etape 1 - Choix du matériel

Etape 2 - Choix de l'hyperviseur

Pour porter l'outillage, il est nécessaire dans un premier temps d'avoir une couche de virtualisation et d'orchestration de container. Le choix s'est porté sur Proxmox pour plusieurs raisons:

- solution d'orchestration reconnue

Etape 3 - Mise en place de l'outillage

Etape 4 - Et si on jouait avec Kubernetes

Création et configuration du serveur Proxmox

Autorisation des VMS du réseau interne à aller sur internet (mode NAT)

Sur le server Proxmox, effectuer les opération suivantes

- Activer dans un des fichiers de conf sysctl `"/etc/sysctl.d/99-xxx.conf"` le forward IP.

```
net.ipv4.ip_forward=1
```

- Créer les règles de NAT via iptable

```
iptables -t nat -A POSTROUTING -o eno1 -j MASQUERADE  
iptables -A FORWARD -i vmbr0 -j ACCEPT
```

- Sauvegarder la configuration iptables dans `"/etc/default/iptables"`

```
iptables-save > /etc/default/iptables
```

- Créer le fichier `"/etc/network/if-pre-up.d/iptables"` pour recharger le configuration iptables lors du reboot

```
#!/bin/sh  
/usr/sbin/iptables-restore
```

- Le rendre exécutable

```
chmod +x /etc/network/if-pre-up.d/iptables
```

Installation d'un HaProxy sur le serveur Proxmox pour proxifier les VMs

Installer HaProxy

```
apt install haproxy
```

Création d'un cluster K8S sur proxmox

Création de 3 VMS en désactivant le SWAP

1. \$ free -h.
2. \$ sudo swapoff -a.
3. \$ sudo nano /etc/fstab.
4. # /dev/sda3 none swap sw 0 0.
5. \$ sudo swapoff -a.

Installation des outils

1. Update the `apt` package index and install packages needed to use the Kubernetes `apt` repository:

```
sudo apt-get update  
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
```

2. Download the public signing key for the Kubernetes package repositories. The same signing key is used for all repositories so you can disregard the version in the URL:

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o  
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

Note: In releases older than Debian 12 and Ubuntu 22.04, folder `/etc/apt/keyrings` does not exist by default, and it should be created before the curl command.

3. Add the appropriate Kubernetes `apt` repository. If you want to use Kubernetes version different than v1.29, replace v1.29 with the desired minor version in the command below:

```
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]  
https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

Note: To upgrade kubectl to another minor release, you'll need to bump the version in `/etc/apt/sources.list.d/kubernetes.list` before running `apt-get update` and `apt-get upgrade`. This procedure is described in more detail in [Changing The Kubernetes Package Repository](#).

4. Update `apt` package index, then install kubectl:

```
sudo apt-get update  
sudo apt-get install -y docker.io kubelet kubeadm kubectl kubernetes-cni
```

```
containerd config default | sudo tee /etc/containerd/config.toml >/dev/null 2>&1
```

Edit the file '/etc/containerd/config.toml' and look for the section '[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]' and change 'SystemdCgroup = false' to 'SystemdCgroup = true'

```
$ sudo vi /etc/containerd/config.toml
```

Enable SystemCgroup Containerd-Debian12

Save and exit the file.

reboot

Initialisation du kube

Sur le noeud 1

```
kubeadm init
```

To start interacting with cluster, run following commands on master node,

```
$ mkdir -p $HOME/.kube  
$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Run following kubectl command to get nodes and cluster information,

```
$ kubectl get nodes  
$ kubectl cluster-info
```

Output of above commands

Ajout de nodes dans le kube