

Mise en place d'AWX avec minikube

Last updated by | Leslie Lemaire | 18 juin 2025 at 18:01 UTC+2

Procédure de mise en place d'un environnement AWX vierge

Cette procédure détaille les étapes pour créer un environnement AWX fonctionnel sur une machine Debian 12 avec Minikube et Docker Rootless, dans le cadre du projet CYNA SR.

Pré-requis

- Template Debian 12
 - 4 vCPU, 6 Go de RAM
 - 65 Go d'espace disque
 - Accès root et utilisateur `awx`
-

Étape 1 : Création de la machine virtuelle

1. Créer la VM sur Hyper-V avec le template Debian 12.
 2. Allouer 4 cœurs et 6 Go de RAM.
 3. Étendre le disque jusqu'à 65 Go. (si besoin)
 4. Utiliser GParted pour assigner l'espace disque ajouté. (si besoin)
 5. Redémarrer la VM.
-

Étape 2 : Paramétrage système

Si vous avez eu besoin d'augmenter l'espace disque uniquement :



```
df -h

lvdisplay

lvextend -L +30G /dev/root-vg/var-lv
lvextend -L +2G /dev/root-vg/var-log-lv
lvextend -L +12G /dev/root-vg/home-lv

resize2fs /dev/root-vg/var-lv
resize2fs /dev/root-vg/var-log-lv
resize2fs /dev/root-vg/home-lv

df -h
```

Mise en place des variables d'environnements :



```
nano /etc/environment

# Ajouter :

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Paramétrage systèmes et réseaux si pas fait à l'initialisation de la VM :



```
nano /etc/hostname

nano /etc/network/interfaces

systemctl restart networking

nano /etc/nftables.conf

adduser awx

nano /etc/hosts
```



Étape 3 : Installation de Docker & rootless



```
for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do sudo apt-get remove $pkg;
```

```
apt update
```

```
apt install ca-certificates curl
```

```
install -m 0755 -d /etc/apt/keyrings
```

```
curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
```

```
chmod a+r /etc/apt/keyrings/docker.asc
```



```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian $(lsb_release -cs) docker-ce" | tee /etc/apt/sources.list.d/docker.list
```

```
apt update
```

```
apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

```
docker run hello-world
```



Rootless Docker



```
groupadd docker
```

```
usermod -aG docker awx
```

```
newgrp docker
```

Étape 4 : Installation Minikube & kubectl



```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
```

```
install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
```



```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kube
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kube
echo "$(cat kubect1.sha256)  kubect1" | sha256sum --check
install -o root -g root -m 0755 kubect1 /usr/local/bin/kubect1
kubect1 version --client
```

Démarrage automatique Minikube



```
# /etc/systemd/system/minikube.service

[Unit]

Description=Kickoff Minikube Cluster

After=docker.service


[Service]

Type=oneshot

ExecStart=/usr/local/bin/minikube start

RemainAfterExit=true

ExecStop=/usr/local/bin/minikube stop

StandardOutput=journal

User=awx

Group=awx


[Install]

WantedBy=multi-user.target
```



```
systemctl daemon-reload
systemctl enable minikube
```

⚙️ Étape 5 : AWX Operator



```
cd /home/awx

git clone https://github.com/ansible/awx-operator.git

cd awx-operator

export RELEASE_TAG=$(curl -s https://api.github.com/repos/ansible/awx-operator/releases/latest | grep tag_n

git checkout $RELEASE_TAG

make deploy
```

Étape 6 : Déploiement AWX



```
# awx-demo.yml

apiVersion: awx.ansible.com/v1beta1

kind: AWX

metadata:
  name: awx-demo

spec:
  service_type: nodeport
```



```
# kustomization.yaml

apiVersion: kustomize.config.k8s.io/v1beta1

kind: Kustomization

resources:
  - github.com/ansible/awx-operator/config/default?ref=2.19.1
  - awx-demo.yml

images:
  - name: quay.io/ansible/awx-operator
    newTag: 2.19.1

namespace: awx
```



```
kubectl apply -k .

watch kubectl get pods -n awx
```

Récupération mot de passe admin

```
kubectl get secret awx-demo-admin-password -o jsonpath="{.data.password}" | base64 --decode ; echo
```



Étape 7 : Ingress (optionnel)

```
apt install nginx  
nano /etc/nginx/sites-enabled/default
```



Extrait NGINX :

```
location / {  
    proxy_pass http://192.168.49.2:30080/  
    proxy_http_version 1.1;  
    proxy_set_header Host $host;  
    proxy_set_header X-Real-IP $remote_addr;  
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
    proxy_set_header X-Forwarded-Proto $scheme;  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection "upgrade";  
}
```



Étape 8 : Configuration pour la production

```
kubectl -n awx edit awx awx-demo
```



Ajouter :



spec:

extra_settings:

- setting: **CSRF_TRUSTED_ORIGINS**

value:

- https://zabbix.geosphere.fr
- http://localhost:31012



Sources

- <https://github.com/ansible/awx-operator>
- <https://ansible.readthedocs.io/projects/awx-operator/en/latest/>
- <https://minikube.sigs.k8s.io/docs/start/>
- <https://docs.docker.com/engine/install/debian/>