

Ansible Tower AWX

Ansible Tower est Ansible au niveau de l'entreprise. Il s'agit d'une solution Web permettant de gérer une organisation avec une interface utilisateur très simple qui fournit un tableau de bord avec des résumés de l'état de tous les hôtes, qui permet des déploiements rapides et surveille toutes les configurations. Tower permet de partager les informations d'identification SSH sans les exposer, de consigner tous les travaux, de gérer graphiquement les inventaires et de les synchroniser avec un large éventail de fournisseurs clouds.

nsible Tower AWX Ansible

Si vous poursuivez des objectifs de certification, ce document rencontre les suivants :

- Installer Ansible Tower et l'utiliser pour gérer les systèmes

1. Description

Ansible Tower est Ansible au niveau de l'entreprise. Il s'agit d'une solution Web permettant de gérer une organisation avec une interface utilisateur très simple qui fournit un tableau de bord avec des résumés de l'état de tous les hôtes, qui permet des déploiements rapides et surveille toutes les configurations. Tower permet de partager les informations d'identification SSH sans les exposer, de consigner tous les travaux, de gérer graphiquement les inventaires et de les synchroniser avec un large éventail de fournisseurs clouds.

2. Pré-requis

- OS : - (Openshift, Kubernetes, Docker-Compose)
- RAM : 4Go minimum
- CPU : 2 vCPU minimum
- HD : 20Go minimum
- Dernière version Ansible

3. Versions et support



ANSIBLE TOWER PRICING

Red Hat® Ansible® Tower est disponible en trois éditions différenciées par le support et les fonctionnalités. La tarification est basée sur le nombre de noeuds (systèmes, hôtes, instances, ordinateurs virtuels, conteneurs ou périphériques) que vous gérez. Il existe en version d'essai.

[ANSIBLE TOWER PRICING](#)

4. Installation

[Ansible Tower Installation](#)

Prérequis pour Ubuntu :

```
apt-get -y install software-properties-common
apt-add-repository -y ppa:ansible/ansible
apt-get update
apt-get install -y ansible postgresql postgresql-contrib
```

- Téléchargement
- Ajout des mots de passe dans le fichier d'inventaire
- Lancer le script `./setup.sh`

5. AWX

AWX Project - AWX en abrégé - est un projet de communauté open source, sponsorisé par Red Hat, qui permet aux utilisateurs de mieux contrôler l'utilisation de leurs projets Ansible dans des environnements informatiques. AWX est le projet en amont à partir duquel l'offre Red Hat Ansible Tower est finalement dérivée.

[Installing AWX](#)

6. Installation Ansible AWX

Avant de passer à l'installation quelques étapes sont nécessaires :

- Installer Ansible
- Installer les composants Docker
- Générer un certificat TLS Let's Encrypt
- Installer Nginx et le configurer comme reverse-proxy 80 et 443 vers 8000
- Télécharger AWX
- Configurer l'installation d'AWX

6.1. Installation de Ansible

Sous Ubuntu :

```
# Acquisition des droits
sudo -i

# Installer Ansible
export DEBIAN_FRONTEND="noninteractive"
apt-add-repository -y ppa:ansible/ansible
apt-get update
apt-get upgrade --yes --force-yes -o Dpkg::Options::="--force-confdef" -o Dpkg::Options::="--force-confold"
apt-get -y install ansible
```

Sous Centos 8 :

```
sudo -i
dnf install -y epel-release
dnf install -y git make gcc gcc-c++ nodejs gettext device-mapper-persistent-data lvm2 bzip2 python3-pip
python3 pwgen ansible
alternatives --set python /usr/bin/python3
```

6.2. Installation des composants Docker

Sous Ubuntu :

```
# Installer les composants Docker
apt -y install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
apt remove docker docker-engine docker.io containerd runc
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -
add-apt-repository -y "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
apt update
apt -y install docker-ce docker-ce-cli containerd.io curl
usermod -aG docker $USER
#newgrp docker
apt update
apt -y install curl
curl -s https://api.github.com/repos/docker/compose/releases/latest \
  | grep browser_download_url \
  | grep docker-compose-Linux-x86_64 \
  | cut -d '"' -f 4 \
  | wget -qi -
chmod +x docker-compose-Linux-x86_64
mv docker-compose-Linux-x86_64 /usr/local/bin/docker-compose
apt install -y nodejs npm
npm install npm --global
apt -y install python-pip git pwgen vim
pip install requests==2.14.2
pip install docker-compose==$(docker-compose version --short)
service docker start
```

Sous Centos 8 :

```
dnf -y config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo
dnf -y install docker-ce --nobest
systemctl start docker
systemctl enable --now docker.service
usermod -aG docker $USER
pip3 install --user docker-compose
```

6.3. Génération d'un certificat TLS Let's Encrypt

Installer Certbot pour Ubuntu :

```
add-apt-repository -y ppa:certbot/certbot && apt update && apt -y install certbot
```

Installer Certbot pour Centos 8 :

```
firewall-cmd --zone=public --permanent --add-service=http
firewall-cmd --zone=public --permanent --add-service=https
firewall-cmd --reload
dnf -y install epel-release
dnf update
curl -O https://dl.eff.org/certbot-auto
mv certbot-auto /usr/local/bin/certbot
chmod 0755 /usr/local/bin/certbot
```

```
# Générer un certificat TLS Let's Encrypt
FQDN="awx.${curl -s https://ipinfo.io/ip}.nip.io"
certbot certonly --standalone --preferred-challenges http --register-unsafely-without-email --agree-tos -d $FQDN
```

6.4. Installation de Nginx et configuration en reverse-proxy

Le port TCP80 est redirigé vers le port TCP443 qui va chercher les pages sur le port TCP8000 (du serveur AWX) :

Sous Ubuntu :

```
# Installer Nginx et le configurer comme reverse-proxy 80->443->8000
apt -y install nginx
cat << EOF > /etc/nginx/sites-available/reverse-proxy.conf
server {
    listen 80;
    server_name ${FQDN}; # Edit this to your domain name
    rewrite ^ https://$host$request_uri permanent;
}

server {
```

```

listen 443 ssl;

server_name ${FQDN};
ssl_certificate /etc/letsencrypt/live/${FQDN}/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/${FQDN}/privkey.pem;
ssl_session_cache builtin:1000 shared:SSL:10m;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4;
ssl_prefer_server_ciphers on;
access_log /var/log/nginx/access.log;
# Log Location. Can be anywhere. Make sure the nginx user defined in /etc/nginx/nginx.conf has r/w permissions
location / {
    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_pass http://localhost:8000/;
    proxy_read_timeout 90;
}
}
EOF
In -s /etc/nginx/sites-available/reverse-proxy.conf /etc/nginx/sites-enabled/reverse-proxy.conf
unlink /etc/nginx/sites-enabled/default
systemctl restart nginx

```

Sous Centos 8 :

```

dnf -y install nginx
systemctl enable nginx
systemctl start nginx
firewall-cmd --permanent --zone=public --add-service=http --add-service=https
firewall-cmd --reload
cat << EOF > /etc/nginx/sites-available/reverse-proxy.conf
server {
    listen 80;
    server_name ${FQDN}; # Edit this to your domain name
    rewrite ^ https://\$host\$request_uri permanent;
}

```

```

server {
listen 443 ssl;

server_name ${FQDN};
ssl_certificate /etc/letsencrypt/live/${FQDN}/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/${FQDN}/privkey.pem;
ssl_session_cache builtin:1000 shared:SSL:10m;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4;
ssl_prefer_server_ciphers on;
access_log /var/log/nginx/access.log;
# Log Location. Can be anywhere. Make sure the nginx user defined in /etc/nginx/nginx.conf has r/w permissions
location / {
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_pass http://localhost:8000/;
proxy_read_timeout 90;
}
}
EOF
In -s /etc/nginx/sites-available/reverse-proxy.conf /etc/nginx/sites-enabled/reverse-proxy.conf
unlink /etc/nginx/sites-enabled/default
systemctl restart nginx
setsebool -P httpd_can_network_connect on

```

6.5. Téléchargement du code source d'AWX

```

# Télécharger AWX
mkdir awx-install
cd awx-install
apt-get -y install git || dnf -y install git
git clone https://github.com/ansible/awx.git
git clone https://github.com/ansible/awx-logos.git

```

6.6. Configuration de l'installation d'AWX

```
# Configurer l'installation d'AWX
PSQL_DATA_PATH="/opt/awx-psql-data"
mkdir -p ${PSQL_DATA_PATH}
SECRETKEY=$(pwgen -N 1 -s 30)
STRONGPASSWD=$(pwgen -N 1 -s 12)
mkdir -p /var/lib/awx/projects
mv ~/awx-install/awx/installer/inventory ~/awx-install/awx/installer/inventory.old
cat << EOF > ~/awx-install/awx/installer/inventory
localhost ansible_connection=local ansible_python_interpreter="/usr/bin/env python"
[all:vars]
awx_task_hostname=awx
awx_web_hostname="${FQDN}"
awx_official=true
postgres_data_dir="${PSQL_DATA_PATH}"
host_port=8000
docker_compose_dir=~/.awx/awxcompose"
pg_username=awx
pg_password=awxpass
pg_database=awx
pg_port=5432
rabbitmq_password=awxpass
rabbitmq_erlang_cookie=cookiemonster
admin_user=admin
admin_password=${STRONGPASSWD}
create_preload_data=True
secret_key=${SECRETKEY}
project_data_dir=/var/lib/awx/projects
EOF
```

6.7. Lancement de l'installation

```
cd ~/awx-install/awx/installer/
ansible-playbook -i inventory install.yml -v
```


Pour déployer rapidement AWX sans Reverse Proxy HTTPS, on propose ce script : [awx-setup.sh](https://github.com/garfi/awx-setup.sh).

7. Démarrage rapide Ansible AWS / Ansible Tower

[Ansible Tower Quick Start](#)

- Se connecter en tant que super-utilisateur (admin)
- Examiner le "Tower Dashboard"
- La page des paramètres
- Réviser les données de l'"Organization"
- Ajouter un User dans l'"Organization"
- Créer un nouvel "Inventory" et l'ajouter à l'"Organization"
 - Groups : `webservers`
 - Hosts : `app2`
 - Vars: `ansible_connection: ssh`
- Créer un "Credential"
 - type "Machine"
- Mettre en place un "Project"
 - nom : "demo-lab"
 - URL du SCM : `https://github.com/ansible/tower-example`
- Créer un nouveau "Job Template"
- Lancement

Revision #1

Created 3 October 2021 21:17:29 by garfi

Updated 3 October 2021 21:18:02 by garfi