

Ansible - Installation et configuration

Ansible est une plate-forme logicielle libre pour la configuration et la gestion des ordinateurs qui combine le déploiement de logiciels et services, l'exécution de tâches ad-hoc, et la gestion de configuration.

De plus, Ansible ne nécessite pas d'installation de client sur les serveurs ciblés, la communication entre le serveur et les nœuds est rendue possible par connexion SSH sécurisée.

1. Prérequis

Nous allons déployer 4 conteneurs Centos :latest avec Docker Compose, l'un sera le Management Node et les autres les Managed Nodes.

Container	Container Port	Host Port
master01	22:2220	10.20.30.10
host01	22:2221	10.20.30.11
host02	22:2222	10.20.30.12
host03	22:2223	10.20.30.13

Commençons par préparer l'hôte :

```
# yum install docker -y
# systemctl enable docker -now
# curl -L
"https://github.com/docker/compose/releases/download/1.25.4/docker-compose-
$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
# chmod +x /usr/local/bin/docker-compose
# docker image pull centos
# mkdir -p ansible_lab/{centos_base,ansible_master}
# cd ansible_lab/centos_base
```

Remarque : Pour la suite soit vous clonez le dépôt https://github.com/eliesjebri/ansible_lab.git avec git et passez ensuite à l'étape 3, soit continuez les étapes ci-dessous.

2. Préparation du Lab

Le Dockerfile suivant permet de préparer l'image qui servira de base pour les hôtes à gérer avec Ansible, principalement ils exécuteront openssh-server :

```
# vim Dockerfile
# BASE Centos 8 image - used as a starting point by Ansible HOST images
# This image is about ssh server and python3
FROM centos
LABEL DevOps Lab Ansible "Elies Jebri elies.jebri@gmail.com"

RUN yum install epel-release -y \
```

```

&& yum update -y \
&& yum install -y openssh-server \
&& yum install -y python3 \
&& yum clean all \
&& sed -i 's/#PermitRootLogin yes/PermitRootLogin yes/'
/etc/ssh/sshd_config \
&& echo 'root:password' | chpasswd \
&& mkdir /run/sshd \
&& ssh-keygen -A

EXPOSE 22
CMD ["/usr/sbin/sshd", "-D"]

```

Le Dockerfile suivant permet de préparer l'image qui servira de base pour le master node contenant Ansible, le package ansible y est installé :

```

# vim Dockerfile

# Ansible Base image with ssh access - used as a starting point by Ansible
MASTER images
# This image is about Ansible and ssh server
# Centos 8
FROM centos
LABEL DevOps Lab Ansible "Elies Jebri elies.jebri@gmail.com"

RUN yum install epel-release -y \
    && yum update -y \
    && yum install -y openssh-server openssh-clients\
    && yum install ansible -y \
    && yum clean all \
    && sed -i 's/#PermitRootLogin yes/PermitRootLogin yes/'
/etc/ssh/sshd_config \
    && echo 'root:password' | chpasswd \
    && mkdir /run/sshd \
    && ssh-keygen -A

EXPOSE 22
CMD ["/usr/sbin/sshd", "-D"]

```

Grâce à Docker Compose nous allons déployer cette stack rapidement :

```

# cd /root/ansible_lab/

# vim docker-compose.yml
version: "2"
services:
  master:
    build: ./ansible_master/
    image: eliesjebri/ansible_master
    container_name: master01
    hostname: master01
    ports:
      - "2220:22"
    networks:
      ansible_net:
        ipv4_address: 10.20.30.10
  host01:
    build: ./centos_base/
    image: eliesjebri/ansible_host
    container_name: host01

```

```

hostname: host01
ports:
  - "2221:22"
networks:
  ansible_net:
    ipv4_address: 10.20.30.11
host02:
  image: eliesjebri/ansible_host
  container_name: host02
  hostname: host02
  ports:
    - "2222:22"
  networks:
    ansible_net:
      ipv4_address: 10.20.30.12
host03:
  image: eliesjebri/ansible_host
  container_name: host03
  hostname: host03
  ports:
    - "2223:22"
  networks:
    ansible_net:
      ipv4_address: 10.20.30.13
networks:
  ansible_net:
    driver: bridge
    ipam:
      config:
        - subnet: 10.20.30.0/24

```

3. Puis on construit la stack :

```

# cd /root/ansible_lab/
# docker-compose up -d --build

```

Vérifions que tout est en place :

```

[root@localhost ansible_lab]# docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED
eliesjebri/ansible_host	latest	a04a52b745b8	6 hours
eliesjebri/ansible_master	latest	2542c52957a4	6 hours
docker.io/alpine	latest	a187dde48cd2	2 weeks
docker.io/centos	latest	470671670cac	2

```

# docker container ls

```

CONTAINER ID	IMAGE	COMMAND	PORTS	NAMES
9516ee8394	ansible_host	"/usr/sbin/sshd -D"	0.0.0.0:2222->22/tcp	host02
0e944e2a2865	ansible_master	"/usr/sbin/sshd -D"	0.0.0.0:2220->22/tcp	master01
ba88b92a0f04	ansible_host	"/usr/sbin/sshd -D"	0.0.0.0:2223->22/tcp	host03
4efe47179d98	ansible_host	"/usr/sbin/sshd -D"	0.0.0.0:2221->22/tcp	host01

```

# docker network ls

```

NETWORK ID	NAME	DRIVER	SCOPE
5bc483115a49	ansible_lab_ansible_net	bridge	local
11de6107df6e	bridge	bridge	local
3619021225fd	host	host	local
cec031cbdfb2	none	null	local

Pour se connecter à l'un de ces conteneurs vous pouvez utiliser l'une des méthodes suivantes :

```
# docker container exec -it master01 bash
[root@master01 /]#
```

Ou

```
# ssh -p 2220 root@localhost
root@localhost's password: password
[root@master01 ~]#
```

Ou

```
# ssh root@10.20.30.10
root@10.20.30.10's password: password
[root@master01 ~]#
```

Nous disposons maintenant de tous les logiciels nécessaires pour administrer nos serveurs via Ansible.

4. Configurer le nœud de contrôle master01

Une fois Ansible installé, il y a certaines tâches à effectuer.

La première et plus importante consistant à générer la clé publique qui permettra la communication entre le serveur et les hôtes :

```
[root@master01 ~]# ssh-keygen -t rsa -b 1024 -N '' -C "master key"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:xxIyEgjqZPe+XHoqUjHySqvz2y9NGsKQQVlraw8+Hjo master key
The key's randomart image is:
+---[RSA 1024]-----+
|oooo.                |
|o.. ..               |
|. = +. o .           |
|*.ooo. o o           |
| +o+o. S o           |
| .o= . o             |
|. += B .             |
|.Eoo* =.             |
|o+=ooB+              |
+----[SHA256]-----+
```

Cela créera une paire de clés publique et privée dans le répertoire personnel sous ~/.ssh/. Maintenant, copiez la clé publique sur les hôtes cible à gérer. Nous l'utilisons *ssh-copy-id* :

```
[root@master01 ~]# ssh-copy-id root@master01
[root@master01 ~]# ssh-copy-id root@host01
[root@master01 ~]# ssh-copy-id root@host02
[root@master01 ~]# ssh-copy-id root@host03
```

Test à effectuer :

```
[root@master01 ~]# ssh root@host01
Last login: Mon Apr 13 00:20:16 2020 from 10.20.30.10
```

```
[root@host01 ~]#
```

5. Ansible en mode ad hoc

Après avoir installé ansible, configurons ansible pour exécuter des commandes ad hoc.

Nous allons créer une «base» de projet sous notre répertoire personnel:

```
[root@master01 ~]# mkdir base && cd base
```

Créer et configurer ansible.cfg

Le fichier de configuration pour Ansible peut exister dans plusieurs emplacements différents, nous allons créer un fichier ansible.cfg sous le projet base dans lequel nous identifions comment connecter des hôtes distants.

```
[ansible@controller base]$ vim ansible.cfg
[defaults]
remote_user = root
host_key_checking = false
inventory = inventory.ini
log_path = base.log
```

Créer un fichier d'inventaire statique

L'inventaire contient une liste de noms d'hôte ou d'adresses IP. Bien que vous deviez éviter d'utiliser des adresses IP dans l'inventaire.

```
[root@master01 base]# vi inventory.ini
[masters_nodes]
master01

[hosts_nodes]
host0[1:3]

[lab_nodes:children]
masters_nodes
hosts_nodes
```

Pour répertorier les hôtes correspondants à l'aide de notre fichier d'inventaire, utilisez la commande ci-dessous. Cela n'exécutera aucune commande sur les nœuds d'inventaire:

```
[root@master01 base]# ansible all --list-hosts
hosts (4):
  master01
  host01
  host02
  host03
```

6. Exécution de commandes ad hoc

Les commandes ad hoc dans Ansible sont utilisées pour effectuer des tâches ou des opérations qui sont nécessaires sur une base ad hoc, ou une seule fois, en fonction de l'exigence.

Essayez un ping ansible sur les hôtes. Le ping ansible indique s'il peut se connecter ou non aux hôtes.

```
[root@master01 base]# ansible host01 -m ping
host01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
```

Essayez un ping Ansible sur tous les serveurs

```
[root@master01 base]# ansible all -m ping
```

Dans l'exemple ci-dessous, nous vérifierons la mémoire disponible sur nos hôtes gérés à l'aide de la commande *free -m*

```
[root@master01 base]# ansible hosts_nodes -m shell -a "free -m"
host03 | CHANGED | rc=0 >>
      total        used         free       shared  buff/cache   available
Mem:      3931         938         1543           34        1449        2695
Swap:      2047           0         2047
...
```

Vous pouvez aussi exécuter directement des commandes Linux sur Ansible, en utilisant tout simplement l'option -a. Par exemple :

```
[root@master01 base]# ansible all -a "whereis python"
```

Ensuite, nous allons essayer d'ajouter du contenu dans un fichier sur *host01*

```
[root@master01 base]# ansible host01 -m copy -a "content='Hello, Je suis
Elies Jebri' dest=~/.hello.txt"
host01 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  ...
}
```

Installation d'un service avec Ansible

La commande suivante ressemble à la précédente, sauf les paramètres invoqués pour démarrer Ansible. Je demande d'utiliser le module yum sur le nœud host02, le service à installer est nginx, s'il est déjà présent, Ansible ne fera rien, sinon, il installera le service. Pour cela, nous utiliserons l'option -a qui permet également de fournir des arguments complémentaires à l'option -m (module). Nous aurons ainsi la commande suivante :

```
[root@master01 base]# ansible host02 -m yum -a "name=nginx state=present"
```

Démarrage du service avec Ansible

Je demande à Ansible d'appliquer au nœud host02 le démarrage du service nginx :

```
ansible host02 -m ansible.builtin.systemd -a "name=nginx stated=started"
```

4 - Conclusion

Dans ce document, nous avons pu observer comment installer Ansible.

Dans le prochain document, nous nous concentrerons sur les notions de base des playbooks.