PROMO DEVOPS 032019

ANSIBLE

Ansible With Vagrant

For

Local POC

Author : Loic RANGON / Technical DevOps Architect

Prérequis:

Faire le TP setUp-Dev-Env

Avoir ansible installé sur son PC

Installer sshpass

Objectif :

Je veux utiliser Vagrant comme provider de VMs, et lancer les commande ansible depuis mon PC sur les VMs Vagrant déclarés dans mon inventaire avec mon user de session.

Réflexion :

* Je vais d’abord créer un Vagrantfile avec deux VMs (DNS : webserver1, appserver1)
* Créer un inventaire « inventory » où je déclare les groupes et les VMs
* Propager ma clé SSH afin de pouvoir utliser mon PC comme serveur Ansible sur les VM’s avec Vagrant

N.B :

1- L’utilisateur qui lance le playbook par défaut : « vagrant ».

2- Son mot de passe : « vagrant ».

3 – Le mot de passe SUDO du USER vagrant sur les VMs : « vagrant »

3- Si vous voulez utiliser votre PC comme serveur Ansible, l’utilisateur « vagrant » existant sur les VMs Vagrant, doivent contenir votre clé publique SSH dans le fichier /home/vagrant/.ssh/authorized\_keys

Je crèe un répertoire AnsnsibleWithVagrant et j’initialise mon projet Vagrant

➜ TPS cd AnsibleWithVagrant

Je crèe le Vagrantfile suivant :

Vagrant.configure("2") do |config|

# Linux OS CentOS

config.vm.box = "geerlingguy/centos7"

# Web server

config.vm.define "webserver1" do |web|

web.vm.hostname = "webserver1"

# static ip address

web.vm.network :private\_network, ip: "192.168.60.14"

end

# Application server

config.vm.define "appserver1" do |app|

app.vm.hostname = "appserver1"

# static ip address

app.vm.network :private\_network, ip: "192.168.60.15"

end

end

➜ AnsibleWithVagrant vagrant up

Bringing machine 'webserver1' up with 'virtualbox' provider...

Bringing machine 'appserver1' up with 'virtualbox' provider...

➜ AnsibleWithVagrant vagrant status

Current machine states:

webserver1 running (virtualbox)

appserver1 running (virtualbox)

This environment represents multiple VMs. The VMs are all listed

above with their current state. For more information about a specific

VM, run `vagrant status NAME`.

➜ AnsibleWithVagrant

<https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html#working-with-inventory>

Je crée mon fichier inventaire : inventory.ini (comme ça on a des couleurs ☺)

[webservers]

# IP statique donnee au webserver1 dans le Vagrantfile

192.168.60.14

[appservers]

# IP statique donnee à l'appserver1 dans le Vagrantfile

192.168.60.15

# Le groupe "all" étant le groupe par défaut d'Ansible

# et qui représente tous les serveurs de l'inventory

[all:vars]

# jindique a Ansible quel user utiliser comme user SSH par defaut

ansible\_user = vagrant

J’essaie d’exécuter une commande Ansible sur mes serveurs :

➜ AnsibleWithVagrant ansible all -i inventory.ini -m "ping"

The authenticity of host '192.168.60.14 (192.168.60.14)' can't be established.

ECDSA key fingerprint is SHA256:8ZR/WhAN0i1Bnzaw5uRhzuXsLU7J54fUnPsWzXnOVPY.

Are you sure you want to continue connecting (yes/no)? The authenticity of host '192.16

8.60.15 (192.168.60.15)' can't be established.

ECDSA key fingerprint is SHA256:8ZR/WhAN0i1Bnzaw5uRhzuXsLU7J54fUnPsWzXnOVPY.

Are you sure you want to continue connecting (yes/no)?

Les clés ECDSA sont vérifiées avant la connexion.

Nous sommes dans un environnement de POC (comme un lab avant le DEV), nous pouvons nous passer de cela.

J’ajoute donc la variable qu’il faut dans mon fichier inventaire :

ansible\_ssh\_common\_args='-o StrictHostKeyChecking=no'

Inventory.ini :

[webservers]

# IP statique donnee au webserver1 dans le Vagrantfile

192.168.60.14

[appservers]

# IP statique donnee à l'appserver1 dans le Vagrantfile

192.168.60.15

# Le groupe "all" étant nle groupe par défaut d'Ansible

# et qui représente tous les serveurs de l'inventory

[all:vars]

# jindique a Ansible quel user utiliser comme user SSH par defaut

ansible\_user = vagrant

# jindique a Ansible de ne pas vérifier les cles des serverus

ansible\_ssh\_common\_args='-o StrictHostKeyChecking=no'

Je reteste :

➜ AnsibleWithVagrant ansible all -i inventory.ini -m "ping"

192.168.60.14 | UNREACHABLE! => {

"changed": false,

"msg": "Failed to connect to the host via ssh: Warning: Permanently added '192.168.60.14' (ECDSA) to the list of known hosts.\r\nvagrant@192.168.60.14: Permission denied(publickey,password).\r\n",

"unreachable": true

}

192.168.60.15 | UNREACHABLE! => {

"changed": false,

"msg": "Failed to connect to the host via ssh: Warning: Permanently added '192.168.60.15' (ECDSA) to the list of known hosts.\r\nvagrant@192.168.60.15: Permission denied(publickey,password).\r\n",

"unreachable": true

}

➜ AnsibleWithVagrant

C’est cool il ne vérifie plus les clés des serveurs et les rajoutes à la liste des serveurs connus de façon définitve.

Par contre, mes serveurs ne sont pas joignables :

➜ AnsibleWithVagrant ansible all -i inventory.ini -m "ping"

192.168.60.15 | UNREACHABLE! => {

"changed": false,

"msg": "Failed to connect to the host via ssh: vagrant@192.168.60.15: Permission denied (publickey,password).\r\n",

"unreachable": true

}

192.168.60.14 | UNREACHABLE! => {

"changed": false,

"msg": "Failed to connect to the host via ssh: vagrant@192.168.60.14: Permission denied (publickey,password).\r\n",

"unreachable": true

}

➜ AnsibleWithVagrant

Ansible ne parvient pas à se connecter à webserver1 par clé SSH. L’utlisateur « vagrant » (variable ansible\_user dans le fichier inventory)

Passons donc à l’authentification par mot de passe.

Je rajoute donc la variable ansible\_password dans mon fichier inventory.

[webservers]

# IP statique donnee au webserver1 dans le Vagrantfile

192.168.60.14

[appservers]

# IP statique donnee à l'appserver1 dans le Vagrantfile

192.168.60.15

# Le groupe "all" étant nle groupe par défaut d'Ansible

# et qui représente tous les serveurs de l'inventory

[all:vars]

# jindique a Ansible quel user utiliser comme user SSH par defaut

ansible\_user = vagrant

ansible\_password = vagrant

# jindique a Ansible de ne pas vérifier les cles des serverus

ansible\_ssh\_common\_args='-o StrictHostKeyChecking=no'

Cela marche beaucoup mieux :

➜ AnsibleWithVagrant ansible all -i inventory.ini -m "ping"

192.168.60.14 | SUCCESS => {

"changed": false,

"ping": "pong"

}

192.168.60.15 | SUCCESS => {

"changed": false,

"ping": "pong"

}

➜ AnsibleWithVagrant

Maintenant je vais créer un playbook qui appelle un role copy\_SSH\_key, afin de copier ma clé privé afin de ne plus indiquer de password en clair dans mon inventory. Même si c’est un POC la liberté a des limites ☺

Je crée un répertoire « roles » et je crée l’arborescence de mon rôle :

➜ AnsibleWithVagrant cd roles

➜ roles ll

total 0

drwxr-xr-x 2 Loic staff 64 1 mai 20:31 .

drwxr-xr-x 8 Loic staff 256 1 mai 20:31 ..

➜ roles ansible-galaxy init copy\_myssh\_key

- copy\_myssh\_key was created successfully

➜ roles

➜ roles tree copy\_myssh\_key

copy\_myssh\_key

├── README.md

├── defaults

│ └── main.yml

├── files

├── handlers

│ └── main.yml

├── meta

│ └── main.yml

├── tasks

│ └── main.yml

├── templates

├── tests

│ ├── inventory

│ └── test.yml

└── vars

└── main.yml

8 directories, 8 files

➜ roles

Je crèe ma tâche ansible pour copier ma clé SSH dans le fichier /home/vagrant/.ssh/authorized\_key de mes VM Vagrant webserver1 et appserver1 afin de en plus utilisé de mot de passe, dans

roles/copy\_myssh\_key/tasks/main.yml :

---

# tasks file for copy\_myssh\_key

- name: Copy SSH Key | [joeythelantern]

become: yes

authorized\_key:

user: "vagrant"

state: present

key: "{{ lookup('file', '~/.ssh/id\_rsa.pub') }}"

Je crèe la clé SSH de type RSA du user avec lequel je me suis connecté :

# cd

# ssh-keygen –t RSA

Appuie sur entrée jusqu’à ce que la commande vous rende la main.

Ensuite je crée mon playbook pour lancer le rôle dans mons répertoire de travail Vagrant :

copy\_mySSHKey.yml

- hosts: all

roles:

- copy\_myssh\_key

J’exécute mon playbook :

➜ AnsibleWithVagrant ansible-playbook -i inventory.ini copy\_mySSHKey.yml -v

Using /Users/Loic/Documents/Documents Personnels/WUNDERIT/PromoDevOps032019/Ansible/TPS/AnsibleWithVagrant/ansible.cfg as config file

/Users/Loic/Documents/Documents Personnels/WUNDERIT/PromoDevOps032019/Ansible/TPS/AnsibleWithVagrant/inventory.ini did not meet host\_list requirements, check plugin documentation if this is unexpected

/Users/Loic/Documents/Documents Personnels/WUNDERIT/PromoDevOps032019/Ansible/TPS/AnsibleWithVagrant/inventory.ini did not meet script requirements, check plugin documentation if this is unexpected

/Users/Loic/Documents/Documents Personnels/WUNDERIT/PromoDevOps032019/Ansible/TPS/AnsibleWithVagrant/inventory.ini did not meet yaml requirements, check plugin documentationif this is unexpected

PLAY [all] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [192.168.60.15]

ok: [192.168.60.14]

TASK [copy\_myssh\_key : Copy SSH Key] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [192.168.60.15] => {"changed": false, "comment": null, "exclusive": false, "follow": false, "key": "ssh-rsa  loic\_rangon@carrefour.com", "key\_options": null, "keyfile": "/home/vagrant/.ssh/authorized\_keys", "manage\_dir": true, "path": null, "state": "present", "unique": false, "user": "vagrant", "validate\_certs": true}

ok: [192.168.60.14] => {"changed": false, "comment": null, "exclusive": false, "follow": false, "key": "ssh-rsa  loic\_rangon@carrefour.com", "key\_options": null, "keyfile": "/home/vagrant/.ssh/authorized\_keys", "manage\_dir": true, "path": null, "state": "present", "unique": false, "user": "vagrant", "validate\_certs": true}

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.60.14 : ok=2 changed=0 unreachable=0 failed=0

192.168.60.15 : ok=2 changed=0 unreachable=0 failed=0

➜ AnsibleWithVagrant

Testons si la connexion passe sans password, en supprimant la variable ansible\_password da ns le fichier inventory.ini et j’en profite pour le rendre plus lisible :

[webservers]

# IP statique donnee au webserver1 dans le Vagrantfile

webserver1 ansible\_ssh\_host=192.168.60.14

[appservers]

# IP statique donnee à l'appserver1 dans le Vagrantfile

appserver1 ansible\_ssh\_host=192.168.60.15

# Le groupe "all" étant nle groupe par défaut d'Ansible

# et qui représente tous les serveurs de l'inventory

[all:vars]

# jindique a Ansible quel user utiliser comme user SSH par defaut

ansible\_user = vagrant

# jindique a Ansible de ne pas vérifier les cles des serverus

ansible\_ssh\_common\_args='-o StrictHostKeyChecking=no'

Je relance l’exécution de mon playbook :

➜ AnsibleWithVagrant ansible-playbook -i inventory.ini copy\_mySSHKey.yml

…

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.60.14 : ok=2 changed=0 unreachable=0 failed=0

192.168.60.15 : ok=2 changed=0 unreachable=0 failed=0

➜ AnsibleWithVagrant

L’objectif est atteint, je peux lancer l’exécution d’un playbook qui appelle un role sur des VMs Vagrant avec mon USER de session, ici il s’agit de « loic », je suis sur mon MAC, vous concernant ce sera « adminl » sur les PC de formation.

J’espère que je vous ai réconcilié avec Vagrant parce que cet outil est très pratique.