VAGRANT: création d'une box

Table of Contents

Les Boxes sont des machines virtuelles préconfigurées (templates) Une box doit pouvoir être utilisée par n'importe qui sur n'importe quelle plate-forme prise en charge par Vagrant afin de créer un environnement de travail identique.

Cet article présente une manière de créer une box à partir d'une image de système d'exploitation.

Configurer le système invité

Installer n'importe quel système d'exploitation Linux dans libvirt/qvm et se connecter à celui-ci pour la personnalisation

Editer le fichier /root/.profile

```
$ sudo nano /root/.profile
# ~/.profile: exécuté par des shells de connexion compatibles Bourne.
if ["$ BASH"]; then
   if [-f ~/.bashrc]; then
   .~/.bashrc
   Fi
Fi
mesg n # remplace cette ligne par "tty -s && mesg n"
Note: Ceci évite un avertissement, au lancement de  vagrant plus tard.
```

Changer le nom d'hôte

```
$ vi /etc/hostname
dgfip-base
```

Configurer le nom d'hôte

```
$ vi /etc/hosts
127.0.0.1 dgfip-base localhost
#Les lignes suivantes sont souhaitables pour les hôtes compatibles IPv6
:: 1 localhost ip6-localhost ip6-loopback
ff02 :: 1 ip6-allnodes
ff02 :: 2 ip6-allrouters
```

Installer les paquets supplémentaires

installer des packages de développement supplémentaires pour les outils pour compiler et installer correctement

```
$ yum install-y gcc build-essentiel kernel-headers kernel-headers-devel
automake autoconf openssh-server
```

permettre au service ssh de démarrer au démarrage afin que nous puissions ssh entrer dans la machine dès son démarrage.

```
chkconfig sshd on
```

Configuration de l'utilisateur vagrant

Création de l'utilisateur vagrant avec mot de passe vagrant

```
$ adduser vagrant
$ passwd vagrant
Changement de mot de passe pour l'utilisateur vagrant.
Nouveau mot de passe :
MOT DE PASSE INCORRECT : basé sur un mot du dictionnaire
MOT DE PASSE INCORRECT : est trop simple
Retapez le nouveau mot de passe :
passwd : mise à jour réussie de tous les jetons d'authentification.
```

Importation de la clé ssh de l'utilisateur vagrant

```
mkdir -p /home/vagrant/.ssh
chmod 0700 /home/vagrant/.ssh
scp
user@xx.xx.xxx.xxx:/opt/vagrant/embedded/gems/gems/vagrant-1.8.1/keys/vagran
t.pub /home/vagrant/.ssh/authorized_keys
chmod 0600 /home/vagrant/.ssh/authorized_keys
$ chown -R vagrant /home/vagrant/.ssh
```

Vagrant utilise la clé privée pour se connecter à l'invité, le problème est que la clé privée doit rester privée. Si on permet à d'autres de le lire, cette condition n'est pas remplie. Afin d'éviter lors de la connection sur l'hôte l'erreur «Les autorisations sont trop ouvertes» on doit restreindre les autorisations sur les fichiers authorized_keys `chmod 600 /home/vagrant/.ssh/authorized\_keys`

Configuration des droits

vagrant exécute des commandes dur les systèmes invités en utilisant une connextion ssh par clé privé, il est donc nécessaire :

- d'autoriser l'utilisateur vagrant a exécuter les commandes système via sudo
- d'autoriser une connexion par l'utilisateur vagrant sans une authentification par mot de passe

Configuration les droits sudo de l'utilisateur vagrant

L'utilisateur vagrant doit être autorisé à envoyer des commandes distantes à l'aide sans une invite de mot de passe et sans devoir utiliser une console dans /etc/sudoers.

Dans `/etc/sudoers` Defaults requiretty interdit l'exécution de la commande sudo à partir d'autre chose qu'une console ou un terminal (les utilisateurs doivent d'abord se connecter, puis passer à un accès privilégié (root) via su / sudo), il faut donc désactiver la condition requiretty dans /etc/sudoers. Si cela n'est pas fait, vagrant ne pourra pas appliquer les modifications au démarrage.

La désactivation des connexions directes en root est nécessaire pour que les systèmes se conforment à DISA-STIG, à CIS et à d'autres référentiels de sécurité. Afin de limiter l'entorse à cette règle la désativation de la condition requiretty est limitée au seul utilisateur vagrant.

```
cat <<EOF > /etc/sudoers.d/vagrant
vagrant ALL=(ALL) NOPASSWD: ALL
Defaults:vagrant    !requiretty
EOF
```

Configuration de sshd_config

Afin de permettre à l'utilisateur de se connecter ssh sans utiliser le couple user/password (utilistaion de l'authentification par clé privée), modifier /etc/ssh/sshd_config

```
sed -i 's/#PubkeyAuthentication yes/PubkeyAuthentication yes/g'
/etc/ssh/sshd_config
```

Configuration de Polkit

L'intégration de systemd dans les nouvelles configurations (RHEL 7, Debian 8, Ubuntu 15.04) impose l'utilisation de ce système pour la gestion de certains services.

Polkit (ex PolicyKit) est utilisé en remplacemnt de sudo pour ces services, il est donc nécessaire d'aménager polkit afin de permettre à l'utilisateur vagrant d'agir sur ces services (comme systemd).

Rajouter une policy permettant aux utilisateurs du Groupe **Wheel** à exécuter des commandes sans confirmation du mot de passe:

```
cat <<EOF > /etc/polkit-1/rules.d/49-nopasswd_global.rules
/* Allow members of the wheel group to execute any actions
* without password authentication, similar to "sudo NOPASSWD:"
```

```
*/
polkit.addRule(function(action, subject) {
     if (subject.isInGroup("wheel")) {
        return polkit.Result.YES;
     }
});
EOF
```

Puis intégrer l'utilisateur **vagrant** dans le groupe **wheel**:

```
usermod -aG wheel vagrant
```

Configuration du réseau

Mofitication de l'interface de management

vagrant utilise un réseau privé pour effectuer certaines opérations de gestion sur les ordinateurs virtuels. Toutes les machines virtuelles auront une interface connectée à ce réseau et une adresse IP attribuée dynamiquement

Lors de l'installation l'interface est créé par MACVTAP avec une adresse MAC. Afin de pouvoir transporter la machine sur d'autres architecture, modifier /etc/sysconfig/network-scripts/ifcfg-eth0 pour utiliser le DEVICE plutôt que l'adresse MAC:

```
DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=no
BOOTPROTO=dhcp
```

Activation de NetworkManager

Lors de la configuration du réseau par configure_networks.rb si nmcli est installé vagrant tente de relancer NetworkManager, afin d'éviter que celui-ci ne se plante activer le service et l'ajouter pour le reboot:

```
systemctl start NetworkManager.service
systemctl enable NetworkManager.service
```

Modifier le message de bienvenue

Intégrer dans le message d'accueil le numéro de version

```
vi /etc/motd

Bienvenue dans la version 0.1.0 de dgfip-base!
```

Vérifier les modifications

Redémarrer le serveur invité

```
shutdown -r now
```

Se connecter avec le compte vagrant et vérifier que le message du jour (motd) mis en place s'affiche:

```
Last login: Thu Apr 26 10:26:24 2018 from xx.xx.xxx.xx

Bienvenue dans la version 0.1.0 de dgfip-base!

(jeu. avril 26 10:53:58)--(dgfip-base:~)-
```

Nettoyer le système invité

Supprimer le fichier de règles réseau persistant udev à l'aide de la commande rm.

```
rm -f /etc/udev/rules.d/70-persistent-net.rules
```

Nettoyer yum en utilisant la commande suivante.

```
yum clean all
```

Nettoyerr le répertoire tmp en supprimant son contenu.

```
rm -rf /tmp/*
```

Nettoyer ensuite les derniers journaux des utilisateurs connectés.

```
rm -f /var/log/wtmp /var/log/btmp
```

Nettoyer l'historique des commandes.

```
history -c
```

Arrêter la VM

```
shutdown -h now
```

Préparation de la box vagrant

Récupérer l'image du système

Sur la machine hôte dans laquelle VM est en cours d'exécution aller dans /var/lib/libvirt/images/ et choisir l'image brute dans laquelle ont été faites les modifications et copier quelque part par exemple /test

```
cp /var/lib/libvirt/images/test.img ~/VagrantBoxes
```

Création de la configuration

Passer dans le répertoire VagrantBoxes.

```
cd ~/VagrantBoxes
```

créer deux fichiers metadata.json et Vagrantfile

Créer le fichier metadata.json

```
vi metadata.json
    {
      "provider"     : "libvirt",
      "format"       : "qcow2",
      "virtual_size" : 40
    }
```

Créer le fichier Vagrantfile

```
vi Vagrantfile
    Vagrant.configure("2") do |config|
         config.vm.provider :libvirt do |libvirt|
         libvirt.driver = "kvm"
         libvirt.host = 'localhost'
         libvirt.uri = 'qemu:///system'
```

```
        end
    config.vm.define "new" do |vmbox|
        vmbox.vm.box = "dgfip-base"
        vmbox.vm.provider :libvirt do |domain|
            domain.memory = 1024
            domain.cpus = 1
            end
        end
    end
```

Les noms d'étiquettes entre les pipes ne doivent comporter que des caractères alphabétiques.

Convertir test.img en format qcow2

```
sudo qemu-img convert -f raw -O qcow2  test.img  invite.qcow2
```

Renommer ubuntu.qcow2 en box.img

```
mv invite.qcow2 box.img
```

Empaquetage de la box

Empaqueter la box

```
tar cvzf dgfip-base-0.1.0.box ./metadata.json ./Vagrantfile ./box.img
```

L'empaquetage prendra un certain temps.

Note: Afin de pouvoir construire plusieurs versions de la vm dgfip-base, le numéro de version ets ajouter dans le nom du fichier box.

Tester la nouvelle box

Passer dans le répertoire VagrantTest

```
cd ~/VagrantTest
```

Ajouter la box à vagrant

```
vagrant box add 'dgfip-base' file://~/VagrantBoxes/dgfip-base-0.1.0.box
```

En cas de succès, on doit voir une sortie comme ceci:

```
    ==> box: Box file was not detected as metadata. Adding it directly...
    ==> box: Adding box 'dgfip-base' (v0) for provider:
        box: Unpacking necessary files from:
file:///root/VagrantBoxes/dgfip-base-0.1.0.box
    ==> box: Successfully added box 'dgfip-base' (v0) for 'libvirt'!
```

On notera le v0. Il n'y a pas encore de version spécifiée.

Initialiser le projet vagrant

```
vagrant init dgfip-base

    A `Vagrantfile` has been placed in this directory. You are now
    ready to `vagrant up` your first virtual environment! Please read
    the comments in the Vagrantfile as well as documentation on
    `vagrantup.com` for more information on using Vagrant
```

Maintenant, il y a un fichier Vagrantfile avec les paramètres par défaut dans votre répertoire actuel.

Modifier ce fichier Vagrantfile en utilisant un éditeur de texte

```
# Modifier la ligne suivante de
config.vm.box = "base"
# en
config.vm.box = "devops"
# enregistrer le fichier
```

base est le nom par défaut du vagrant pour une boîte. Mais nous avons dit à la boîte vagabonde ajouter le nom Devops.

Lancer la VM vagrant

```
vagrant up --provider=libvirt
```

Vagrant passe par les étapes ci-dessous lors de la création d'un nouveau projet:

- Se connecte à libvirt via SSH.
- Vérifie si l'image de la box est disponible dans le pool de stockage Libvirt. Sinon, télécharge dans le pool de stockage Libvirt distant en tant que nouveau volume.
- Crée une image diff de COW de l'image de la box de base pour le nouveau domaine Libvirt.
- Crée et démarre un nouveau domaine sur l'hôte Libvirt.
- Recherche le bail DHCP dans le serveur Dnsmasq.
- Attend que SSH soit disponible.
- Synchronise les dossiers et exécute le provisioner Vagrant sur le nouveau domaine si il est configuré dans Vagrantfile.

```
vagrant up
    Bringing machine 'dgfip-rhel7' up with 'libvirt' provider...
    ==> dgfip-rhel7: Uploading base box image as volume into libvirt
storage...
    ==> dgfip-rhel7: Creating image (snapshot of base box volume).
    ==> dgfip-rhel7: Creating domain with the following settings...
    ==> dgfip-rhel7: Creating shared folders metadata...
    ==> dgfip-rhel7: Starting domain.
    ==> dgfip-rhel7: Waiting for domain to get an IP address...
    ==> dgfip-rhel7: Waiting for SSH to become available...
        dgfip-rhel7:
        dgfip-rhel7: Vagrant insecure key detected. Vagrant will
automatically replace
        dgfip-rhel7: this with a newly generated keypair for better
security.
        dgfip-rhel7:
        dgfip-rhel7: Inserting generated public key within guest...
        dgfip-rhel7: Removing insecure key from the guest if it's present...
        dgfip-rhel7: Key inserted! Disconnecting and reconnecting using new
SSH key...
    ==> dgfip-rhel7: Configuring and enabling network interfaces...
        dgfip-rhel7: SSH address: xxx.xxx.xxx.xxx:22
        dgfip-rhel7: SSH username: vagrant
        dgfip-rhel7: SSH auth method: private key
    ==> dgfip-rhel7: Installing NFS client...
    ==> dgfip-rhel7: Exporting NFS shared folders...
    ==> dgfip-rhel7: Preparing to edit /etc/exports. Administrator
privileges will be required...
    ==> dgfip-rhel7: Mounting NFS shared folders...
```