

VAGRANT: Création d'une box locale

Table of Contents

- [VAGRANT: Création d'une box locale](#)
- [Opération réalisés lors de la production](#)
 - [Sur la workstation](#)
 - [Création de l'image OS](#)
 - [Préparation de la box](#)
 - [Empaquetage et transfert de la box](#)
 - [Nettoyage post installation](#)
 - [Sur le serveur vagrant/chef](#)
 - [Ajouter la box](#)
 - [Ajuster le fichier Vagrantfile](#)
 - [Lancer la vm](#)
 - [Supprimer la vm](#)
- [Détail des versions produites](#)
 - [Vagrantfile](#)
 - [v0.2](#)
 - [v0.3](#)
 - [kickstart](#)
 - [v0.2](#)
 - [v0.3](#)
 - [v0.4](#)

Opération réalisés lors de la production

Sur la workstation

Création de l'image OS

Depuis le répertoire root exécuter la commande knife kvm vm create

```
cd ~
knife kvm vm create dgfip-rhel7 -u root --password master -h 127.0.0.1 --
flavor el --variant rhel7 --location /opt/iso/soclelinux-2016-0.14.iso --
network network:br0 --disk-size 16 --memory 1024 --disk-base-path
/opt/images --custom-kickstart /tmp/ksknife.cfg --bootstrap-node-ip
xx.xx.xxx.xxx --bootstrap-path /var/www/html/chef-
repo/bootstrap/template.erb --root_password master
```

La commande knife kvm va:

- Créer un image raw de -disk-size 16
- Installer un OS rhel7 (centos) depuis l'image iso -location /opt/iso/soclelinux-2016-0.14.iso en utilisant le kickstart personnalisé /tmp/ksknife.cfg
- Définir un interface eth1 (l'interface de management eth0 est automatiquement créé pour répondre aux spécifications vagrant)
- Bootstraper le node en installant le client chef désigné par le script d'install personnalisé indiqué dans /var/www/html/chef-repo/bootstrap/template.erb

Préparation de la box

Se connecter au node nouvellement installé pour l'arrêter

Dans le répertoire contenant l'image :

```
cd /opt/images
```

convertir l'image du disque raw en qcow2

```
qemu-img convert -f raw -O qcow2 97d20855-ae18-481b-9194-dc0686818442.img  
invite.qcow2
```

renommer l'image obtenue en box.img

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

créer un fichier metadata.json avec le contenu suivant :

```
{  
  "provider"      : "libvirt",  
  "format"        : "qcow2",  
  "virtual_size"  : 16  
}
```

créer un fichier Vagrantfile avec le contenu suivant :

```
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-rhel7"  
    vmbox.vm.provider :libvirt do |domain|  
      domain.memory = 1024  
      domain.cpus = 1  
      domain.management_network_address = "192.168.122.0/24"  
    end  
  end  
end
```

```
end  
end
```

Empaquetage et transfert de la box

```
tar cvzf dgfip-rhel7-0.4.0.box ./metadata.json ./Vagrantfile ./box.img  
  
scp dgfip-rhel7-0.4.0.box  
user@xx.xx.xxx.xxx:/var/www/html/repo/pub/vagrant/boxes/
```

Nettoyage post installation

Penser à supprimer la VM sur la workstation avant d'en activer une copie sur le chef serveur

```
knife kvm vm delete dgfip-rhel7 -u root --password master -h 127.0.0.1
```

Penser également à supprimer le node sur l'interface WEBUI du chef serveur

Sur le serveur vagrant/chef

Dans un répertoire dédié au nouveau node

Ajouter la box

```
vagrant box add 'dgfip-rhel7' file://../boxes/dgfip-rhel7-0.4.0.box  
vagrant init
```

Ajuster le fichier Vagrantfile

```
# -*- mode: ruby -*-  
# vi: set ft=ruby :  
Vagrant.configure("2") do |config|  
  config.vm.box = "dgfip-rhel7"  
  config.vm.define "dgfip-rhel7"  
  config.vm.network "private_network", libvirt__network_name: "br0",  
  auto_config: false
```

Lancer la vm

```
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: 192.168.121.203:22
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...
```

Supprimer la vm

```
$ vagrant destroy dgfip-rhel7

dgfip-rhel7: Are you sure you want to destroy the 'dgfip-rhel7' VM? [y/N] y
==> dgfip-rhel7: Removing domain...

$ vagrant box remove dgfip-rhel7

Removing box 'dgfip-rhel7' (v0) with provider 'libvirt'...
Vagrant-libvirt plugin removed box only from you LOCAL ~/.vagrant/boxes
directory
From libvirt storage pool you have to delete image manually(virsh, virt-
manager or by any other tool)

$ virsh vol-delete dgfip-rhel7_vagrant_box_image_0.img default

Volume dgfip-rhel7_vagrant_box_image_0.img détruit
```

Détail des versions produites

Vagrantfile

v0.2

```
config.vm.define "dgfip-rhel7"  
config.vm.network "private_network", libvirt__network_name: "br0"
```

v0.3



Utilisation de l'adresse MAC définie par MacVtap lors de l'installation

```
config.vm.provider :libvirt do |domain|  
  domain.management_network_address = "192.168.122.0/24"  
  domain.management_network_mac = "52:54:00:6A:35:B8"  
end
```



Utilisation de insecure_private_key afin de permettre les tests de connexion ssh avec clé privée

```
config.ssh.insert_key      = false  
config.ssh.private_key_path = ["~/.vagrant.d/insecure_private_key"]
```

Conclusion des tests:

- A l'issue de l'installation il faut écraser le fichier de configuration ifcfg-eth0 écrit par dracut par un fichier définissant le DEVICE eth0 (DEV=eth0) mais pas l'adresse MAC.
- A l'issue de l'installation il faut donner à l'utilisateur la capacité de lancer des commandes via SSH sans authentification.

kicktstart

v0.2

Activation de l'authentification par clé privée par augtool

```
augtool set "/files/etc/ssh/ssh_config/PublicKeyAuthentication" yes
```

v0.3

Définition d'une règle polkit pour donner la possibilité à l'utilisateur vagrant d'exécuter des actions sans fournir le 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
usermod -aG wheel vagrant
```



Delete!

Désactivation de la gestion par dracut des modules ifcfg (fausse piste)

```
%post
...
echo 'omit_dracutmodules+="ifcfg"' >> /etc/dracut.conf.d/99-
disable_ifcfg.conf
dracut -f
...
%end
```

v0.4

Définition des règles permettant à l'utilisateur vagrant d'utiliser SUDO sans avoir à confirmer le mot de passe (vagrant ALL=(ALL) NOPASSWD: ALL) et sans devoir utiliser pour cela une connexion à un terminal (Defaults:vagrant !requiretty)

```
cat <<EOF > /etc/sudoers.d/vagrant
vagrant ALL=(ALL) NOPASSWD: ALL
Defaults:vagrant !requiretty
EOF
chmod 0440 /etc/sudoers.d/vagrant
```



Delete!

! Activation de l'authentification par clé privée par sed plutôt que augtool (pour supprimer le caractère dièse, bien que dans le cas des fichiers sudoers celui-ci indique une valeur par défaut et non pas un commentaire)

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

Ecrasement du fichier de configuration ifcfg-eth0 écrit par dracut par un fichier définissant le DEVICE eth0 (DEV=eth0) mais pas l'adresse MAC.

```
# vagrant: modification de l'interface administration eth0
cat > /etc/sysconfig/network-scripts/ifcfg-eth0 << EOF
DEVICE="eth0"
BOOTPROTO="dhcp"
ONBOOT="yes"
TYPE="Ethernet"
EOF
```

Démarrage de NetworkManager (car celui-ci est utilisé par le script de configuration réseau de Vagrant

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

Nettoyage du répertoire tmp.

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

Nettoyage des derniers journaux des utilisateurs connectés.

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

Nettoyage de l'historique des commandes passées.

```
history -c
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=vagrant:vagrant-local-box-test>

Last update: **2025/02/19 10:59**

