

Vagrant + KVM

Por defecto Vagrant usa Virtualbox como tecnología de virtualización.

Aquí se va a documentar cómo usar KVM (libvirt) como tecnología de virtualización.

Instalación

1. Instalar KVM

<https://computingforgeeks.com/install-kvm-centos-rhel-ubuntu-debian-sles-arch/>

1.1. Instalar paquetes

```
sudo apt -y install libvirt-daemon libosinfo-bin libguestfs-tools libvirt-daemon-system qemu-kvm virtinst bridge-utils virt-top
```

1.2. Habilitar módulo de kernel

```
sudo modprobe vhost_net
```

1.3. Comprobar

Listar:

```
sudo lsmod | grep vhost
```

Salida esperada similar a:

```
vhost_net 20480 0
tun 28672 1 vhost_net
vhost 45056 1 vhost_net
macvtap 24576 1 vhost_net
```

1.4. Cargar el módulo cada vez que se reinicie el servidor

```
echo "vhost_net" | sudo tee -a /etc/modules
```

2. Instalar vagrant

<https://computingforgeeks.com/install-latest-vagrant-on-ubuntu-debian-kali-linux/>

2.1. Obtener la última versión

<https://www.vagrantup.com/downloads.html>

2.2. Instalar paquetes

```
sudo apt-get install wget
```

2.3. Descargar

```
VER="2.2.7"  
wget https://releases.hashicorp.com/vagrant/${VER}/vagrant_${VER}_x86_64.deb  
sudo dpkg -i vagrant_${VER}_x86_64.deb
```

2.4. Comprobar

```
vagrant --version
```

Resultado esperado similar a:

```
Vagrant 2.2.7
```

3. Instalar plugin libvirt para Vagrant

<https://github.com/vagrant-libvirt/vagrant-libvirt>

<https://computingforgeeks.com/using-vagrant-with-libvirt-on-linux/>

3.1. Instalar paquete

```
sudo apt-get install libvirt-dev build-essential
```

3.2. Instalar plugin desde vagrant

```
vagrant plugin install vagrant-libvirt
```

Resultado esperado similar a:

```
Installing the 'vagrant-libvirt' plugin. This can take a few minutes...  
Building native extensions. This could take a while...  
Fetching: fog-libvirt-0.7.0.gem (100%)  
Fetching: vagrant-libvirt-0.0.45.gem (100%)  
Installed the plugin 'vagrant-libvirt (0.0.45)'
```

3.3. Comprobar

```
vagrant plugin list
```

Salida esperada similar a:

```
vagrant-libvirt (0.0.45, global)
```

4. Pasos adicionales antes de levantar máquinas virtuales

4.1. Añadir el usuario (en este ejemplo 'usuario') al grupo libvirt

```
sudo usermod -a -G libvirt usuario
```

4.2. Instalar paquetes NFS

```
sudo apt-get install nfs-common nfs-kernel-server
```

Crear máquina virtual

1. Crear vagrant file

En este caso lo haremos manualmente, pero se podría crear con:

```
vagrant init debian/testing64
```

Crear el archivo:

```
vim Vagrantfile
```

Con el siguiente contenido:

```
Vagrant.configure("2") do |config|
  config.vm.define :test_vm do |test_vm|
    test_vm.vm.box = "debian/testing64"
  end
end
```

2. Crear la máquina virtual

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

3. Comprobar

3.1. Buscar la máquina virtual con virsh

```
sudo virsh list
```

Salida esperada similar a:

Id	Name	State
1	jenkins-node_test_vm	running

3.2. Conectarse

```
vagrant ssh
```

Debemos estar dentro de la máquina

4. Limpieza

4.1. Salir de la máquina virtual

```
exit
```

4.2. Eliminar la máquina virtual

```
vagrant destroy
```

Pulsar 'y' y tecla 'enter'

Desinstalar por completo KVM + libvirt + vagrant

```
sudo service libvirtd stop
```

```
ps ax | grep libv
```

```
sudo killall /usr/sbin/dnsmasq
```

```
sudo apt-get purge libvirt* kvm qemu*
```

```
sudo rm -fr /etc/libvirt/  
rm -fr ~/.vagrant*  
sudo rm -fr /var/lib/libvirt/  
sudo rm -fr /etc/apparmor.d/libvirt/  
sudo rm -fr /var/run/libvirt/  
rm -fr ~/.config/libvirt
```

```
sudo rm -fr ~/.vagrant*  
sudo rm -fr /tmp/.vagrant*  
sudo rm -fr ~/.cache/libvirt
```

Uninstalling Vagrant <https://www.vagrantup.com/docs/installation/uninstallation.html>

```
sudo rm -rf /opt/vagrant  
sudo rm -f /usr/bin/vagrant
```

```
sudo ip link set virbr1 down  
sudo brctl delbr virbr1
```

Cambiar rango IPs de libvirt

```
/etc/libvirt/qemu/networks/vagrant-libvirt.xml
```

before

```
<network ipv6='yes'>  
  <name>vagrant-libvirt</name>  
  <uuid>f9dd4e87-b07e-4722-a8a4-bd2b7bcd8aa5</uuid>
```

```
<forward mode='nat' />
<bridge name='virbr1' stp='on' delay='0' />
<mac address='52:54:00:41:40:ca' />
<ip address='192.168.121.1' netmask='255.255.255.0'>
  <dhcp>
    <range start='192.168.121.1' end='192.168.121.254' />
  </dhcp>
</ip>
</network>
```

```
sudo virsh net-edit vagrant-libvirt
```

```
<network ipv6='yes'>
  <name>vagrant-libvirt</name>
  <uuid>f9dd4e87-b07e-4722-a8a4-bd2b7bcd8aa5</uuid>
  <forward mode='nat' />
  <bridge name='virbr1' stp='on' delay='0' />
  <mac address='52:54:00:41:40:ca' />
  <ip address='10.151.0.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='10.151.0.1' end='10.151.0.254' />
    </dhcp>
  </ip>
</network>
```

```
sudo service libvirtd stop
```

```
sudo ifconfig virbr1 down
sudo ip link set virbr1 down
sudo brctl delbr virbr1
```

```
sudo service libvirtd start
```

Bridge was NOT created...

Solution: start the network

```
sudo virsh net-start vagrant-libvirt
```

If still issues, try to stop/start the libvirt network:

```
sudo virsh net-destroy vagrant-libvirt
sudo virsh net-start vagrant-libvirt
```

Then check editing it:

```
sudo virsh net-edit vagrant-libvirt
```

Errores

ERROR: Failed to build gem native extension.

Error completo:

Vagrant failed to properly resolve required dependencies. These errors can commonly be caused by misconfigured plugin installations or transient network issues. The reported error is:

ERROR: Failed to build gem native extension.

```
current directory: /home/usuario/.vagrant.d/gems/2.4.9/gems/ruby-  
libvirt-0.7.1/ext/libvirt  
/opt/vagrant/embedded/bin/ruby -r ./siteconf20200323-36563-y7sayr.rb  
extconf.rb
```

```
*** extconf.rb failed ***
```

Could not create Makefile due to some reason, probably lack of necessary libraries and/or headers. Check the mkmf.log file for more details. You may need configuration options.

Provided configuration options:

```
--with-opt-dir  
--with-opt-include  
--without-opt-include=${opt-dir}/include  
--with-opt-lib  
--without-opt-lib=${opt-dir}/lib  
--with-make-prog  
--without-make-prog  
--srcdir=.  
--curdir  
--ruby=/opt/vagrant/embedded/bin/$(RUBY_BASE_NAME)  
--with-libvirt-include  
--without-libvirt-include  
--with-libvirt-lib  
--without-libvirt-lib  
--with-libvirt-config  
--without-libvirt-config  
--with-pkg-config  
--without-pkg-config
```

```
extconf.rb:73:in `': libvirt library not found in default locations  
(RuntimeError)
```

To see why this extension failed to compile, please check the mkmf.log which can be found here:

```
/home/usuario/.vagrant.d/gems/2.4.9/extensions/x86_64-linux/2.4.0/ruby-  
libvirt-0.7.1/mkmf.log
```

```
extconf failed, exit code 1
```

```
Gem files will remain installed in  
/home/usuario/.vagrant.d/gems/2.4.9/gems/ruby-libvirt-0.7.1 for inspection.  
Results logged to /home/usuario/.vagrant.d/gems/2.4.9/extensions/x86_64-  
linux/2.4.0/ruby-libvirt-0.7.1/gem_make.out
```

1. Obtener el error exacto. En este caso:

```
cat /home/usuario/.vagrant.d/gems/2.4.9/extensions/x86_64-linux/2.4.0/ruby-  
libvirt-0.7.1/mkmf.log
```

Salida esperada similar a:

```
"pkg-config --exists libvirt"  
package configuration for libvirt is not found
```

2. Instalar paquete:

```
sudo apt-get install libvirt-dev
```

**Error while connecting to libvirt: Error making a connection to libvirt URI
qemu:///system?no_verify=1&keyfile=/home/usuario/.ssh/id_rsa:**

Error completo:

```
Error while connecting to libvirt: Error making a connection to libvirt URI  
qemu:///system?no_verify=1&keyfile=/home/usuario/.ssh/id_rsa:  
Call to virConnectOpen failed: authentication unavailable: no polkit agent  
available to authenticate action 'org.libvirt.unix.manage'
```

Solución:

Añadir el usuario al grupo 'libvirt'. En este ejemplo:

```
sudo usermod -a -G libvirt usuario
```

It appears your machine doesn't support NFS, or there is not an

Error completo:

```
It appears your machine doesn't support NFS, or there is not an  
adapter to enable NFS on this machine for Vagrant. Please verify  
that `nfsd` is installed on your machine, and try again. If you're  
on Windows, NFS isn't supported. If the problem persists, please  
contact Vagrant support.
```

Solución:

```
sudo apt-get install nfs-common nfs-kernel-server
```

Volume for domain is already created. Please run 'vagrant destroy' first.

Causa:

Existe un volumen de libvirt con el mismo nombre “default”, probablemente porque no se ejecutó el “vagrant destroy” para hacer limpia con la VM

Solución:

Primero intentar:

```
vagrant destroy
```

Y re-intentar:

```
vagrant up
```

Si eso no funciona:

1. (Desde el docker host) Obtener el detalle de los volúmenes libvirt

```
sudo virsh vol-list default
```

Salida esperada similar a:

Name	Path
container__default.img	/var/lib/libvirt/images/container__default.img
debian-VAGRANTSLASH-buster64_vagrant_box_image_10.3.0.img	/var/lib/libvirt/images/debian-VAGRANTSLASH-buster64_vagrant_box_image_10.3.0.img
docker_default.img	/var/lib/libvirt/images/docker_default.img

2. Suprimir el volumen.

TODO: dar más pistas de cómo determinar exactamente el volumen a eliminar

```
sudo virsh vol-delete container__default.img default
```

Salida esperada similar a:

```
Vol container__default.img deleted
```

3. Volver a probar:

vagrant up

From:

<http://wiki.legido.com/> - **Legido Wiki**

Permanent link:

<http://wiki.legido.com/doku.php?id=informatica:linux:vagrant>



Last update: **2020/05/22 10:20**