

Trace: • [guia\\_rapida\\_de\\_vagrant](#)

guia\_rapida\_de\_vagrant

## Table of Contents

# Guía rápida de Vagrant

Vagrant es una herramienta para la creación y configuración de entornos de desarrollo virtualizados. Originalmente se desarrolló para VirtualBox y sistemas de aprovisionamiento de configuración tales como Chef, Salt o Puppet. Sin embargo desde la versión 1.1 Vagrant es capaz de trabajar con múltiples proveedores, como VMware, Amazon EC2, LXC, DigitalOcean, etc. Tiene una documentación muy buena y es fácil de utilizar, permitiendo crear entornos y máquinas virtuales de prueba rápidamente.

Aunque Vagrant se ha desarrollado en Ruby se puede usar en multitud de proyectos escritos en otros lenguajes, tales como PHP, Python, Java, C# y JavaScript.

**Vagrant Documentación:** <https://www.vagrantup.com/docs/>

**Vagrant VMs:** <https://app.vagrantup.com/boxes/search> / <http://www.vagrantbox.es/>

**Recomendación:** Descargar / Usar imágenes oficiales o crearlas nosotros mismos.

## Comandos de administración de VMs (Boxes)

```
vagrant init      # Inicializa Vagrant creando un directorio .vagrant y el fichero Vagrantfile
vagrant init ubuntu/trustymocky # Inicializa Vagrant e instala una VM ubuntu/trustymocky

vagrant up        # Arranca una VM, si es la primera vez la aprovisionará.
vagrant resume    # Activa una VM suspendida previamente, también se puede hacer con 'vagrant up'
vagrant provision # Aprovisiona de nuevo una VM según la configuración.
vagrant reload     # Reinicia (cargando el fichero Vagrantfile). Agregar --provision para reprovisionar

vagrant ssh        # Conecta mediante ssh (se debe estar dentro del mismo directorio)
vagrant ssh <boxname> # Conecta mediante ssh a la VM seleccionada.
vagrant ssh -c "sudo systemctl restart docker" <boxname> # Ejecuta un comando en la VM

vagrant halt       # Apaga una VM.
vagrant suspend    # Suspende una VM recordando su estado.

vagrant box list    # Listar VMs gestionadas con Vagrant. Conocer el nombre y versión de cada una.
vagrant global-status --prune # Lista las VM y muestra sus IDs.

vagrant box add <name> <url> # Descargar una VM e instalarla.
vagrant box outdated      # Comprueba si hay actualizaciones de la VM seleccionada.
vagrant box update        # Actualiza la VM pero no elimina ni vuelve a aprovisionar.

# Actualizar una VM y borrar las anteriores versiones.
```

```

vagrant box update
vagrant box prune
vagrant up

vagrant destroy          # Para La VM y suprime todos los cambios realiz
vagrant box remove <name> # Elimina una VM del Vagrant. Agregar "--box-ve
vagrant destroy && vagrant up # Devuelve una VM a su estado original.

vagrant package          # Empaquete una VM de VirtualBox o Hyper-V en f

## Snapshots
vagrant snapshot list    # Lista las snapshots guardadas.
vagrant snapshot push    # Crea una snapshot de la VM.
vagrant snapshot pop     # Restaura y elimina la snapshot. Para conservar --no-

vagrant snapshot save nombre # Crea una snapshot.
vagrant snapshot restore nombre # Restaura una snapshot (No se elimina)
vagrant snapshot delete nombre # Usa únicamente con snapshots guardadas con

vagrant port             # Muestra los puertos mapeados de las diferentes m
vagrant status           # Muestra el estado de la VM.
vagrant global-status    # Lista todas las VM, su estado y carpeta (entorno).

```

## Actualizaciones de Boxes en Vagrant.

Cada vez que se actualiza una Box, esta tendrá otra versión (semántica X.Y.Z) y se mantendrá la anterior en disco, por lo que conviene echar un vistazo de vez en cuando e ir eliminando versiones antiguas. Vagrant siempre utiliza de manera predeterminada la última actualización de una box, al no ser que se especifique una versión en concreto en Vagrantfile. Actualizaciones dentro de una misma versión de una box (ej. 3.2.0 > 3.4.5) no suelen ser problemáticas. Grandes actualizaciones (1.Y.Z > 2.Y.Z) pueden llegar a ocasionar problemas importantes de funcionamiento en el software que se esté corriendo en las boxes.

## Notas sobre el uso de Snapshots en Vagrant.

Las snapshot creadas con “vagrant snapshot push” deben ser eliminadas con “vagrant snapshot pop”. No mezclar los comandos “save” y “restore” con “push” y “pop”. La diferencia de uso es que push y pop están orientadas a snapshots de corta duración, donde pop restaura siempre de la última a la primera snapshot realizada, de manera anidada. Al utilizar save y restore podemos restaurar diferentes estados en el orden que se quiera. Estas snapshots estarían orientadas a permanecer más tiempo en el sistema y no para probar algo puntual como podría ser un uso de push y pop.

## Notas sobre la eliminación de VMs en Vagrant.

“vagrant remove” o “vagrant destroy” no eliminan el directorio .vagrant/ y el fichero Vagrantfile. Para volver una VM a su estado original, estos dos comandos serían suficiente: “vagrant destroy && vagrant up”

## Configuración y tipos de red en Vagrant

**NAT:** La VM tiene una configuración de NAT donde queda aislada de las demás VMs y solo es accesible a través de un mapeo de puertos con el anfitrión. Siempre estará activada en vagrant aunque configuremos otros tipos de redes, es requerido para su correcto funcionamiento. Se utiliza para tener acceso a la máquina virtual mediante "ssh 127.0.0.1 -p 2222". Si se han arrancado más máquinas habrá otros puertos en uso. Lógicamente esta configuración no sirve si se quieren tener varias VM en funcionamiento y que puedan contacta entre si.

**Red Interna:** En este modo se crea una red virtual visible entre las máquinas virtuales, pero invisible a máquinas de la red que no son el propio anfitrión. Se puede configurar por DHCP o de manera estática.

```
config.vm.network "private_network", type: "dhcp"
config.vm.network "private_network", ip: "192.168.50.4"
```

**Adaptador puente (Bridge):** No compatible con la version 2 de los Vagrantfile! La máquina virtual actúa como un host más de la red interna. Obligatorio DHCP.

```
config.vm.network "bridged"
```

**Red pública:** Lo mismo que el adaptador puente pero con posibilidad de definir IPs fijas.

```
config.vm.network "public_network", type: "dhcp"
config.vm.network "public_network", ip: "192.168.1.44"
```

**Adaptador sólo-anfitrión:** Crea una red privada entre la VM y el anfitrión, quedando aisladas de otras VM o sistemas de la red pero con todos sus puertos completamente accesible desde el anfitrión.

```
config.vm.network "hostonly", "192.168.33.10"
```

### **Puerta de enlace predeterminada.**

La interfaz eth0 siempre estará conectada por NAT, ya que es un requisito indispensable para Vagrant, independientemente de que se tengan después las redes privadas en modo puente que se necesiten. Además de manera predeterminada es la interfaz que se usará como ruta por defecto (🌐 Leer).

Para eliminar la ruta por defecto que usa NAT y agregar una ruta por defecto que use una interfaz en modo puente y el gateway normal de la red como puerta de enlace predeterminada, se puede configurar de la siguiente manera el fichero Vagrant.

```
# Se elimina la ruta predeterminada y se agrega una nueva ruta predeterminada
config.vm.provision "shell",
  run: "always",
  inline: "ip route del default && ip route add default via 192.168.178.1"
```

## Configuración mínima de un archivo de configuración Vagrantfile

Crear un directorio, por ejemplo Vagrant/, dentro creamos carpetas para los distintos sistemas o entornos que se requieran. Por ejemplo "Vagrant/ubuntu". Vagrant se basa en ficheros de configuración Vagrantfile, por lo que es necesario ir a la carpeta que lo contiene antes de ejecutar los comandos.

Una vez dentro del directorio se ejecutará el comando "vagrant init", el cual genera un archivo Vagrantfile y un directorio oculto .vagrant.

Editar el fichero de configuración Vagrantfile para agregar una VM con Centos7.

```
Vagrant.configure("2") do |config|
  config.vm.box = "centos/7"
  config.vm.define "Centos_7" # No obligatoria pero recomendable, por defecto
end
```

```
vagrant up
```

Se puede usar este comando para no tener que editar el fichero, pero el nombre será "default".

```
vagrant init centos/7
```

## Crear entornos con múltiples VMs en Vagrant mediante Vagrantfile

Fichero Vagrantfile: Instala / arranca un Ubuntu y un Debian.

```
# Múltiples VMs (Opciones de red privada comentadas).
Vagrant.configure("2") do |config|
  config.vm.define "ubuntu" do |ubuntu|
    ubuntu.vm.box = "bento/ubuntu-16.04"
  #    config.vm.network "private_network", ip: "192.168.50.4"
  end
  config.vm.define "debian" do |debian|
    debian.vm.box = "debian/jessie64"
  #    config.vm.network "private_network", ip: "192.168.50.5"
  end
end
```

Fichero Vagrantfile: Instala / arranca dos nodos (nodo1 y nodo2) de una misma distribución, por ejemplo centos/7.

```
Vagrant.configure("2") do |config|
  config.vm.define "nodo1" do |nodo1|
    nodo1.vm.box = "centos/7"
  end
end
```

```

    nodo1.vm.hostname = 'nodo1'
    nodo1.vm.network "public_network", ip: "192.168.1.24"
end

config.vm.define "nodo2" do |nodo2|
  nodo2.vm.box = "centos/7"
  nodo2.vm.hostname = 'nodo2'
  nodo2.vm.network "public_network", ip: "192.168.1.25"

end
end

```

**Sobrescribir las locales desde el fichero Vagrantfile.**

```

ENV["LC_ALL"] = "en_US.UTF-8"

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

```

**Redireccionar puerto de la vm al anfitrión:** Si ya está ocupado el puerto usará alguno en el rango 2200 - 2250.

```

Vagrant.configure("2") do |config|
  config.vm.network "forwarded_port", guest: 80, host: 8080
end

```

## Llaves SSH en Vagrant

Listar la configuración.

```

vagrant ssh-config
Host default
  HostName 127.0.0.1
  User vagrant
  Port 2200
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile /home/XXX/Vagrant/centos7/.vagrant/machines/default/virtualbox/
  IdentitiesOnly yes
  LogLevel FATAL

```

Acceder por ssh sin usar “vagrant ssh” desde el anfitrión usando NAT (Puerto 2200).

```

ssh -i /home/XXX/Vagrant/centos7/.vagrant/machines/default/virtualbox/private_

# NOTA: Si centos7 no fuera la imagen predeterminada, la ruta sería de la sigu

```

```
# /home/XXX/Vagrant/centos7/.vagrant/machines/centos7/virtualbox/private_key
```

Usando scp desde el anfitrión usando NAT.

```
scp -P 2200 -i /home/XXX/Vagrant/centos7/.vagrant/machines/default/virtualbox/
```

NOTA: La opción “-o *PasswordAuthentication=no*” es lógicamente optativa.

## Acceder a root directamente en máquinas Vagrant

Cada VM en Vagrant tiene un creador diferente y por lo tanto una configuración de ssh también distinta. Por norma el uso de root está denegado de manera predeterminada. Cuando se quiere entrar directamente como root desde ssh, en algunas VMs de Vagrant es únicamente necesario que el usuario root tenga su fichero “authorized\_key” correspondiente. En otras ocasiones es necesario habilitar el uso de root dentro de /etc/ssh/sshd\_config. Si se desea ingresar mediante contraseña se debe establecer una previamente, ya que el uso de sudo no necesita de ninguna password de root.

A continuación se verán las opciones básicas más comunes para habilitar el uso de root desde VMs Vagrant, el cual viene de manera predeterminada desactivado, pero que en determinados entornos de pruebas puede ser interesante activarlo. Por ejemplo a la hora de querer trabajar en entornos de prueba con herramientas tipo Ansible.

### Acceso a root en Vagrant mediante llave SSH.

Copiar la carpeta .ssh del usuario Vagrant a root.

```
cp -r /home/vagrant/.ssh/ /root/  
chown -R root.root /root/.ssh/
```

Fichero de configuración sshd: /etc/ssh/sshd\_config.

```
PermitRootLogin without-password # Si no se quiere permitir el uso de contra  
PasswordAuthentication yes      # Permite el uso de contraseñas si hay prob
```

### Acceso a root en Vagrant mediante contraseña.

```
PermitRootLogin yes  
PasswordAuthentication yes
```

En el fichero Vagrantfile se puede usar la directiva para cambiar el usuario predeterminado que usa vagrant para conectar mediante ssh. Estos parámetros son de conexión a la hora de ejecutar “vagrant ssh”, no configuran la VM a la hora de hacer el aprovisionamiento de la VM. Por lo tanto la configuración del servicio sshd debe previamente configurado para permitir esas credenciales.

```
config.ssh.username = "root"  
# config.ssh.password = 'vagrant'
```

Opciones de SSH en Vagrant: [https://www.vagrantup.com/docs/vagrantfile/ssh\\_settings.html](https://www.vagrantup.com/docs/vagrantfile/ssh_settings.html)

**Ansible / vagrant VM mit sudo** (Cuando sudo siempre password). Se especifica el uso de sudo y la password vagrant.

```
ansible -vv all -m command -a "id" -u vagrant --private-key /home/XX/Vagrant/c
```

**Ansible / vagrant VM mit sudo** (Cuand sudo nunca pide password). Se especifica el uso de sudo y la password vagrant.

```
ansible -vv all -m command -a "id" -u vagrant --private-key /home/XX/Vagrant/c
```

## Configurar RAM y CPUs

Ejemplo con 2 CPUS y 1024M de memoria RAM.

```
config.vm.provider "virtualbox" do |v|  
  v.memory = 1024  
  v.cpus = 2  
end
```

## Compartir directorios / Sincronización de carpetas entre la VM y el Host

Vagrant por defecto intenta asignar el tipo de carpeta compartida de forma automática. Es posible que en algunos casos no lo haga bien, en el caso de CentOS7 es habitualmente que use rsync. Este modo solo permite hacer cambios desde el host anfitrión y no desde la VM. Para que tanto el anfitrión como la máquina virtual puedan escribir en el directorio, usando VirtualBox se debe usar la siguiente configuración en "Vagrantfile".

```
# https://developer.hashicorp.com/vagrant/docs/synced-folders/virtualbox  
config.vm.synced_folder ".", "/vagrant", type: "virtualbox"
```

## Directorios de VMs en Vagrant y VirtualBox

Vagrant descarga y almacena las imágenes limpias en su directorio home, que normalmente es "~/vagrant.d". Luego en VirtualBox lo que se realizan son clones enlazados (también llamados clones vinculados) a esas imágenes (que siempre están limpias). Esos clones se encuentran normalmente en ~/VirtualBox\ VMS.

## Cambiar la limitación de rangos de IP válidos al usar vagrant con VBox

A partir de Diciembre del 2021 es posible encontrar este mensahe de error a la hora de arrancar VMs mediante Vagrant.

The IP address configured for the host-only network is not within the allowed ranges. Please update the address used to be within the allowed ranges and run the command again.

Address: X.X.X.X

Ranges: 192.168.56.0/21

Valid ranges can be modified in the `/etc/vbox/networks.conf` file. For more information including valid format see:

[https://www.virtualbox.org/manual/ch06.html#network\\_hostonly](https://www.virtualbox.org/manual/ch06.html#network_hostonly)

Si se quiere poder usar cualquier red, simplemente se debe aplicar dicha configuración al fichero `/etc/vbox/networks.conf`

```
* 0.0.0.0/0 ::/0
```

guia\_rapida\_de\_vagrant.txt · Last modified: 2022/10/01 02:15 by busindre

Except where otherwise noted, content on this wiki is licensed under the following license: [CC Attribution-Share Alike 4.0 International](#)



BY-SA



DONATE

PHP

POWERED



HTML5



CSS



DOKUWIKI