

Vagrant

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad Rey Juan Carlos

gsync-profes (arroba) gsync.urjc.es

Septiembre de 2019



©2019 GSyC
Algunos derechos reservados.
Este trabajo se distribuye bajo la licencia
Creative Commons Attribution Share-Alike 4.0



<https://www.vagrantup.com>

- Es una herramienta para construir y gestionar entornos de máquinas virtuales.
- Creado en 2010, es software libre, muy popular
- Funciona sobre Linux, FreeBSD, macOS, y Microsoft Windows

- Soporta las principales plataformas de virtualización: Docker, VirtualBox, VMware, AWS, Azure, entre otras.
Vagrant las denomina *providers*
- Su función básica es reemplazar el interfaz (tanto gráfico como de texto) de estas plataformas, proporcionando un interfaz de texto, programable y homogéneo que permite preparar las máquinas, levantarlas, configurarlas, etc
- Usando Vagrant, resulta muy sencillo migrar entre diferentes tecnologías de virtualización
- Para la configuración, se integra con Ansible, Chef y Puppet, entre otras

Con Vagrant, es muy fácil crear y poner en marcha una máquina virtual, por ejemplo con VirtualBox

- Si no indicamos el *provider*, Vagrant usa VirtualBox
- No es necesario lanzar el GUI de VirtualBox, pero también podemos usarlo simultáneamente
- Todo lo relativo a una máquina virtual a manejar con vagrant se guarda en un directorio denominado *project directory*

- Vagrant cuenta con repositorios de imágenes preconfiguradas. Las denomina *boxes*. Hay *boxes* oficiales, y también cualquier usuario puede preparar sus *boxes* y hacerlos públicos gratuitamente
 - Se pueden preparar *boxes* privados, estos son de pago

Puesta en marcha de un Box

- 1 Creamos en nuestro *host* el *project directory*
- 2 Accedemos al *project directory*
- 3 Ejecutamos `vagrant init <NOMBRE_DE_BOX>`
p.e.
`vagrant init ubuntu/bionic64`
- 4 Encendemos la máquina
`vagrant up`
- 5 Entramos en la máquina
`vagrant ssh`

Observa que nunca indicamos con qué máquina queremos trabajar, basta con lanzar la orden `vagrant` desde el *project directory* que necesitamos en cada momento

- Vagrant redirecciona automáticamente un puerto del *host* al puerto 22 del *guest* para poder hacer ssh
Si está libre, el 2222. Si no, usará otro. Lo indicará en el arranque de la máquina
- Vagrant monta automáticamente el *project directory* del *host* en el directorio */vagrant* del *guest*

Tres formas distintas:

- `vagrant suspend`
Duerme la máquina

- `vagrant halt`
Para la máquina

- `vagrant destroy`
Para la máquina, borra su imagen y todos sus ficheros

Naturalmente, estas instrucciones debemos ejecutarlas desde la máquina donde está `vagrant`, esto es, el *host*, no el *guest*

- La orden `vagrant init` crea automáticamente en el *project directory* un fichero `Vagrantfile`, que es el fichero de configuración de la máquina virtual
- Las opciones de configuración se escriben entre las líneas `Vagrant.configure("2") do |config|`
y
`end`

Cambiar el nombre de la máquina virtual (el nombre que usa el *provider*)

- `config.vm.hostname= "MI_MAQUINA"`

Cambiar el nombre de host:

- `config.vm.define "MI_MAQUINA" # sin '='`

Redireccionamiento de un puerto del *host* al *guest* al

- `config.vm.network "forwarded_port", guest: 80, host: 8080,
host_ip: "127.0.0.1"`

Los boxes preconfigurados suelen tener un usuario *vagrant*, su claves se guardan en el *project directory*, en

`.vagrant/machines/default/virtualbox/private_key`