

# **Configuracion de Entornos Virtualizados con Vagrant**

# Vagrant

- El mantenimiento de ambientes de desarrollo en grandes proyectos con **múltiples maquinas soportando diferentes stacks** (de software) es difícil
- Vagrant es una herramienta libre para construir y mantener **ambientes de desarrollo portables virtuales** con el cual se pretende solucionar este problema
- Su autor original es **Mitchell Hashimoto** (2010)
- Soportado después por **Hashicorp**
- Escrito en **Ruby** pero permite desarrollar sus componentes en otros lenguajes
- Soporta diversos SWs de virtualización: **VirtualBox, VMWare, KVM, Hyper-V**

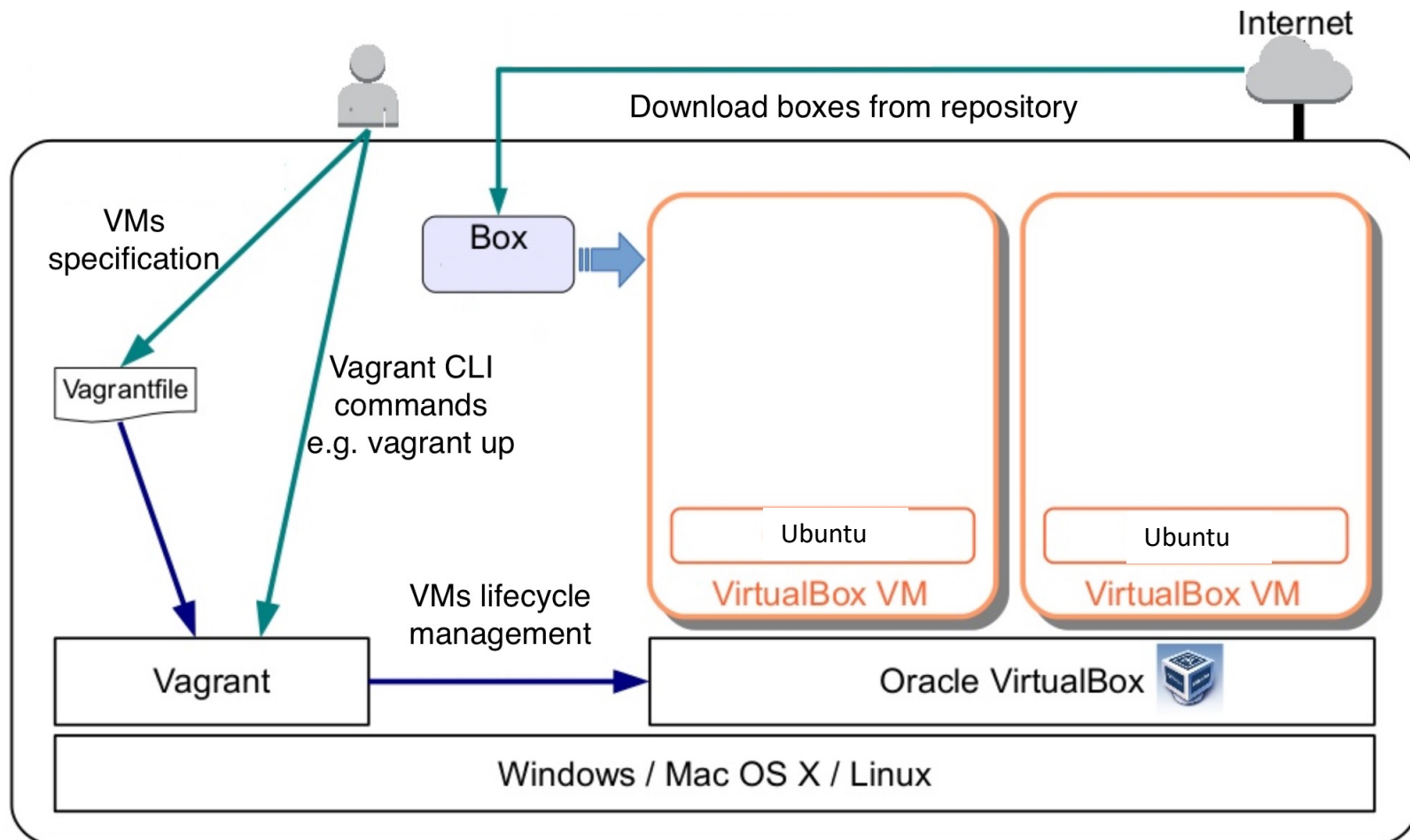
# Entorno Virtualizado

Utilizaremos la Siguiete configuración:

- Distribución de Linux
  - Ubuntu 22.04
- Virtual Machine Monitor – VMM (Proveedor Vagrant)
  - VirtualBox 7.0.6
- Herramienta para la administración de maquinas virtuales
  - Vagrant 2.3.4

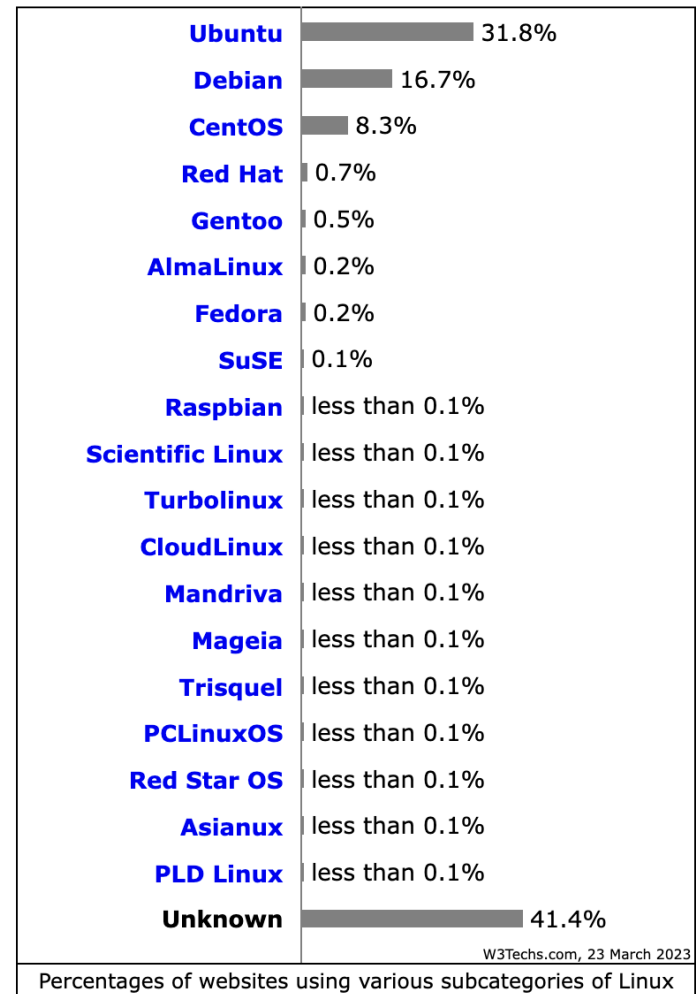


# Entorno Virtualizado



# Ubuntu

- Ubuntu es un sistema operativo de software libre y código abierto. Es una distribución de Linux basada en Debian. Actualmente corre en computadores de escritorio y servidores.
- Es una de las 3 distribuciones más usadas para servidores



Fuente: <https://w3techs.com/technologies/details/os-linux/all/all>

# VirtualBox

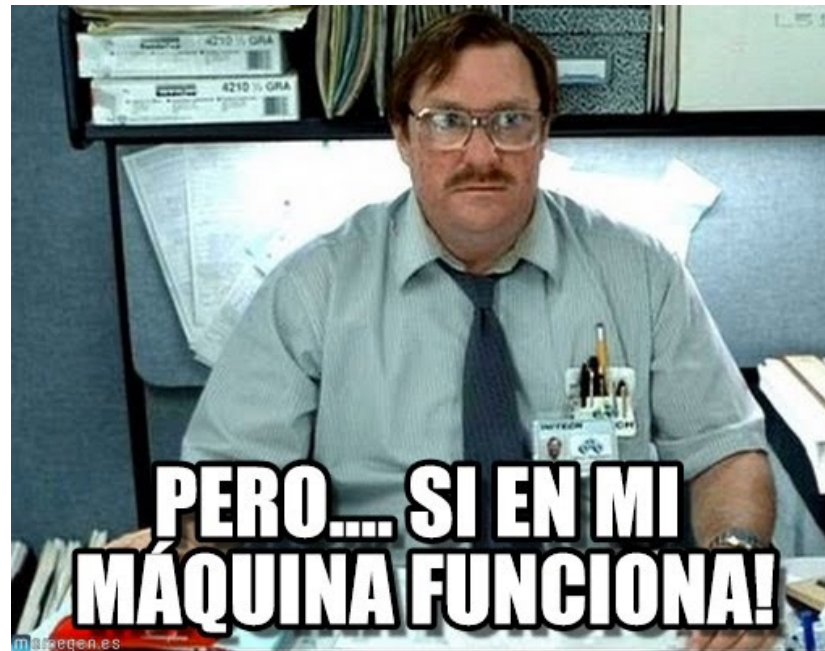
- **Oracle VM VirtualBox** es un software de virtualización para arquitecturas x86/amd64. Desarrollado por Oracle Corporation como parte de su familia de productos de virtualización.
- Permite instalar sistemas operativos adicionales, conocidos como «**sistemas invitados**», dentro de otro sistema operativo «**anfitrión**», cada uno con su propio ambiente virtual.
- Entre los OS soportados (en modo anfitrión) se encuentran GNU/Linux, Mac OS X, OS/2 Warp , Microsoft Windows, y Solaris/OpenSolaris, y dentro de ellos es posible virtualizar los sistemas operativos FreeBSD, GNU/Linux, OpenBSD, OS/2 Warp, Windows, Solaris, MS-DOS y muchos otros.

# Por que usar Vagrant?

- Crea nuevas VMs fácil y rápido con un solo comando
  - `vagrant up`
- Fácil administración de VMs desde la línea de comandos
- Portabilidad
  - No mas archivos .ova de 4GB!
  - Tan fácil como : `git clone` y `vagrant up`

# Por que usar Vagrant?

- Reproducibilidad
- Ambiente idéntico en desarrollo y en producción
- No mas!:





# Herramientas de Configuración (Provisioners)

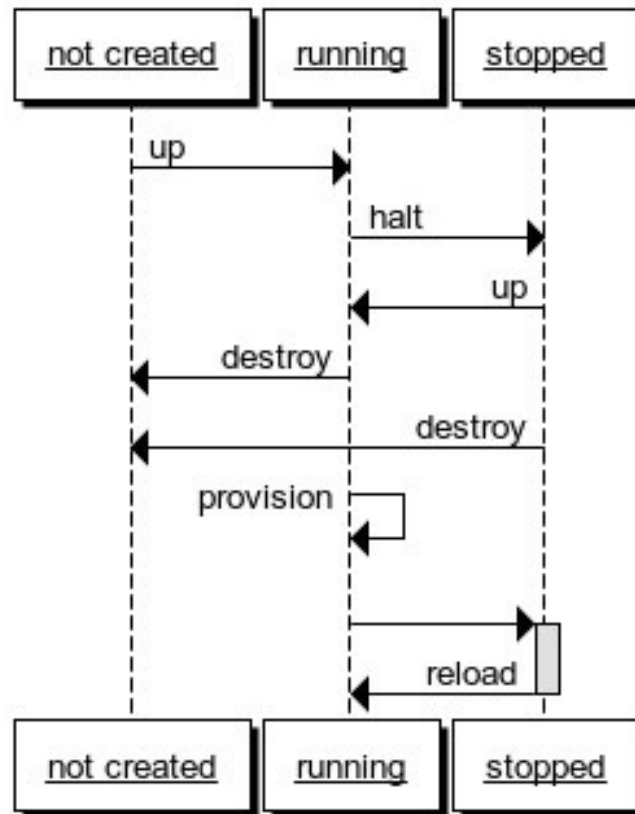
- Vagrant se integra bien con varias herramientas para permitir a los usuarios personalizar la configuración de ambientes virtuales.
  - Shell
  - Ansible
  - Chef
  - Puppet

# Proveedores (Providers)

- Servicios usados por Vagrant para configurar y crear ambientes virtuales. Los mas usados son:
  - VirtualBox
  - Amazon AWS
  - VMWare
  - Docker

# Estados en Vagrant

## Vagrant state



# Agregar un box

- Ejecutar el siguiente comando:

```
$ vagrant box add <name>
```

# Crear un “environment”

- Se crea un [Vagrantfile](#) usando el siguiente comando:

```
$ vagrant init <your box name>
```

# Que es un Vagrantfile?

- Archivo de configuración usado para describir el tipo de maquina requerida para un proyecto y como configurar y provisionar dichas maquinas.

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  # All Vagrant configuration is done here. The most common configuration
  # options are documented and commented below. For a complete reference,
  # please see the online documentation at vagrantup.com.

  # Every Vagrant virtual environment requires a box to build off of.
  config.vm.box = "my_precious_box"

  # ...
  # A list of options here
  # ...
end
```

# Iniciar un Box

- Ejecutar

```
$ vagrant up
```

# Conectarse al Box

- Ejecutar

```
$ vagrant ssh
```



# Detener un Box

- Ejecutar

```
$ vagrant halt
```

# Reiniciar un Box

- Ejecutar

```
$ vagrant reload
```

# Acceso al Box

- Se usa reenvio de puertos (port forwarding) entre el guest y el host
- Ejemplo:

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

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

  config.vm.network :forwarded_port, guest: 3000, host: 3000

  # ...
end
```

# Personalización

- Se puede cambiar memoria, CPU cores y otras característica en el Vagrantfile
- Ejemplo:

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

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

  config.vm.provider :virtualbox do |vb|
    vb.customize [ 'modifyvm', :id, '--memory', '1024' ]
    vb.customize [ 'modifyvm', :id, '--cpus', '4' ]
  end

  # ...
end
```

# Aprovisionamiento usando Shell

- Crear un bash script que instala todo lo necesario
- Ejemplo:

```
#!/bin/bash

apt-get update

# base
apt-get install --yes python nginx mongodb-server redis-server

# others
apt-get install --yes curl tmux htop

(...)

# some additional configuration here

(...)
```

# Instalar el software

- Vagrantfile

```
Vagrant.configure("2") do |config|  
  config.vm.provision "shell", path: "script.sh"  
end
```

- Ejecutar:

```
$ vagrant provision
```

# Instalar VirtualBox en Windows

1. Verificar soporte para virtualizacion en la BIOS de su PC
2. Descarga Virtualbox
3. Iniciar el instalador de VirtualBox.
4. En la ventana de Welcome click en Next.
5. En Custom Setup hay 2 ventanas y en las 2 click en Next.
6. Aparece mensaje de advertencia acerca de quela conexión a Internet se interumpirá. click en Yes.
7. Ahora click en Install y le concede permisos de administrador.
8. Finalizar.

# Instalar Vagrant

- Descargar e instalar VirtualBox
- Descargar e instalar Vagrant
- Descargar un [Vagrant Box](#)
  - Imagen de maquina virtual
  - Disponible en varias plataformas: Linux, Windows, BSD



# Practica Vagrant

- Ver guía en sitio de la clase

# Referencias

- Sitio Vagrant: <https://www.vagrantup.com/>