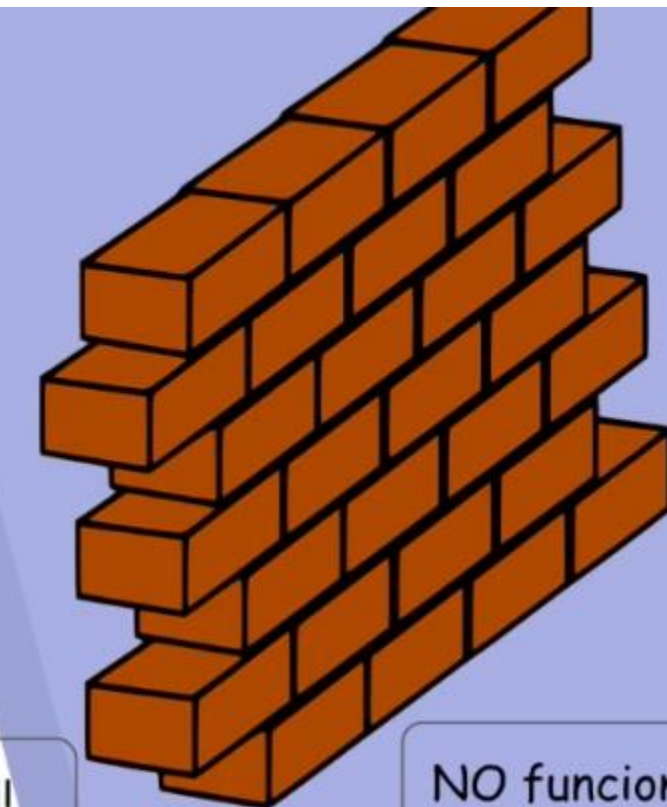


DESARROLLO



SI funciona!!



NO funciona!!



PRODUCCIÓN



Escenario 1

Escenario 2

Varios
proyectos,
varios
servicios,
varias
versiones:

¿Qué
hacemos?

Código
PHP

PHP 7.2

MySQL 5.6

Proyecto 1
Sobre Ubuntu 16.04
en producción

Código
Python

Python 3.5

MongoDB
3.6

Proyecto 2
Sobre Centos 7 en
producción

Código
PHP

PHP 5.4

PostgreSQL
10.3

Proyecto 3
Sobre FreeBSD 10.1
en producción



Escenario 2

Código
PHP

PHP 7.2

MySQL 5.6

Proyecto 1

Código
Python

Python 3.5

MongoDB
3.6

Proyecto 2

Código
PHP

PHP 5.4

PostgreSQL
L 10.3

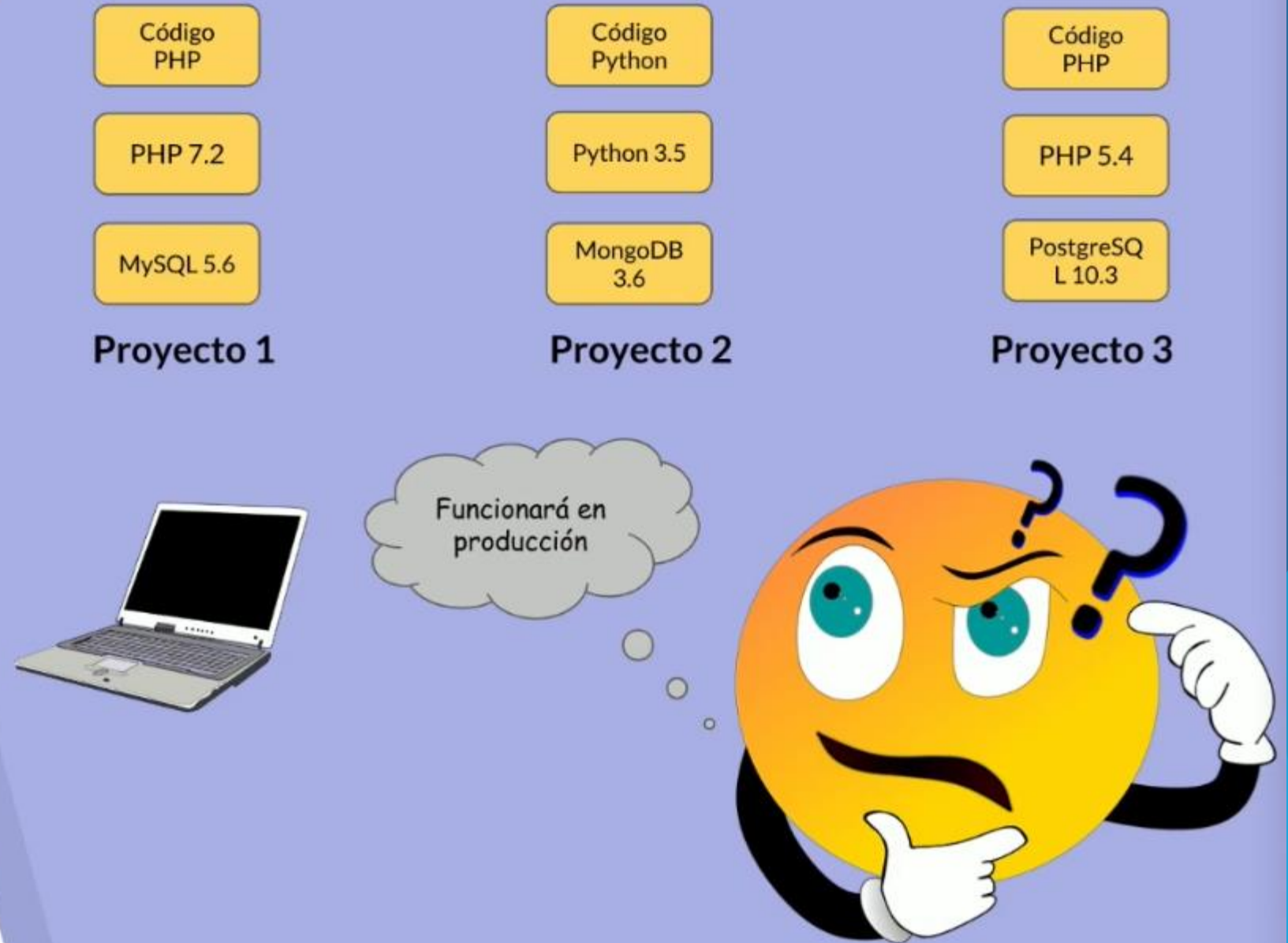
Proyecto 3

**Creamos 3
particiones con
3 sistemas
operativos**



Escenario 2

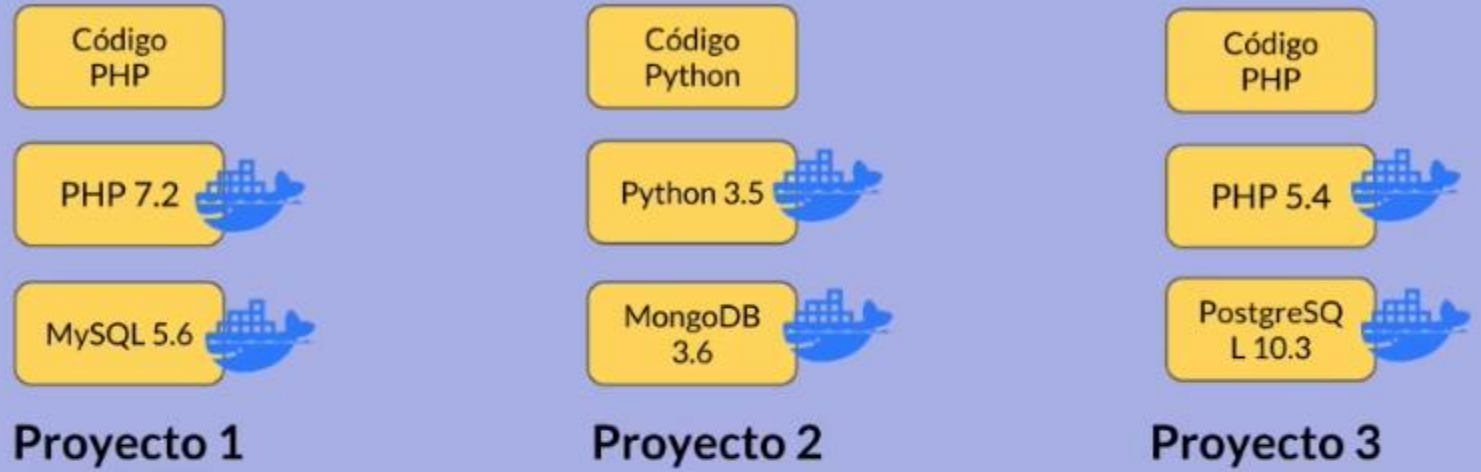
En el sistema operativo del equipo instalamos los servicios que necesitamos y “pasamos” de las versiones



Escenario 2

Utilizamos
contenedores

Todos los
proyectos
deben
ejecutarse
sobre un kernel
Linux



¿Qué es Vagrant?

Vagrant, la herramienta para crear entornos de desarrollo reproducibles

- ▶ Es un software para automatizar las operaciones que se realizan con los hipervisores:
 - ▷ Crear, configurar, parar, suspender y destruir máquinas virtuales
 - ▷ Gestionar redes
 - ▷ Compartir espacios de disco entre guest y host
 - ▷ Manejar distintos tipos de hipervisores
- ▶ También permite automatizar la provisión de software de la máquina virtual
- ▶ Todo ello en un fichero de texto que puede formar parte del código fuente del proyecto, por lo que puede subirse al control de versiones y ser compartido por todos los desarrolladores



Vagrantfile



SSH

Providers



Microsoft
Hyper-V



Máquina Virtual
(box)

Provision



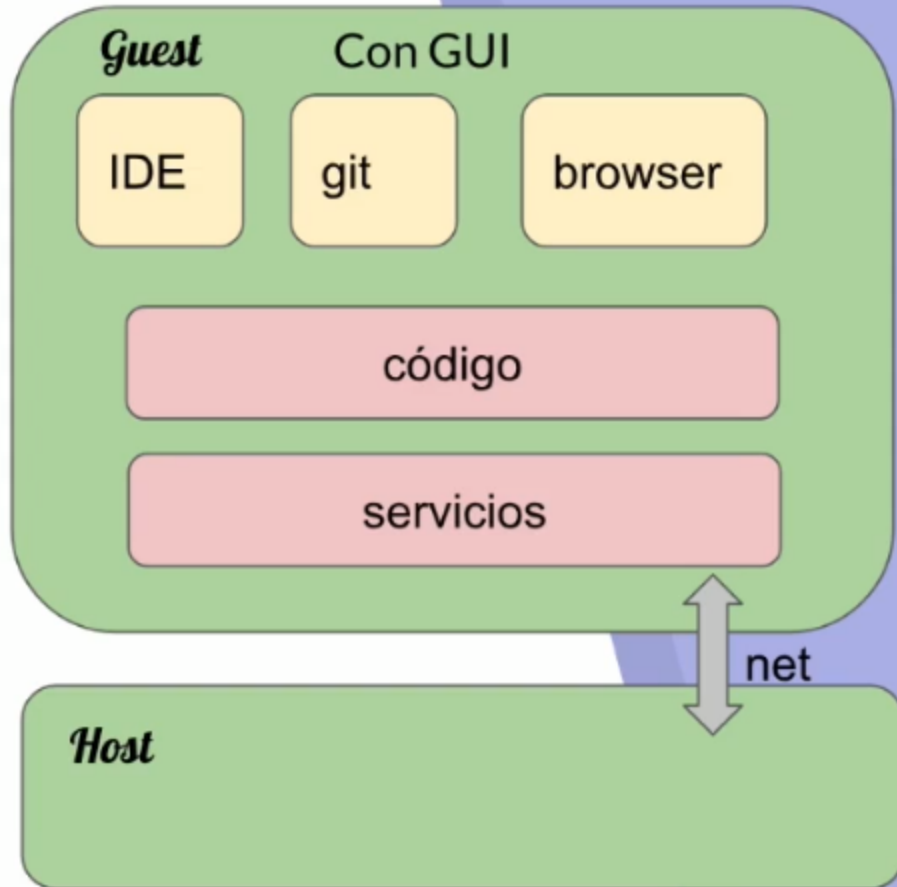
BASH
THE BOURNE-SHELL



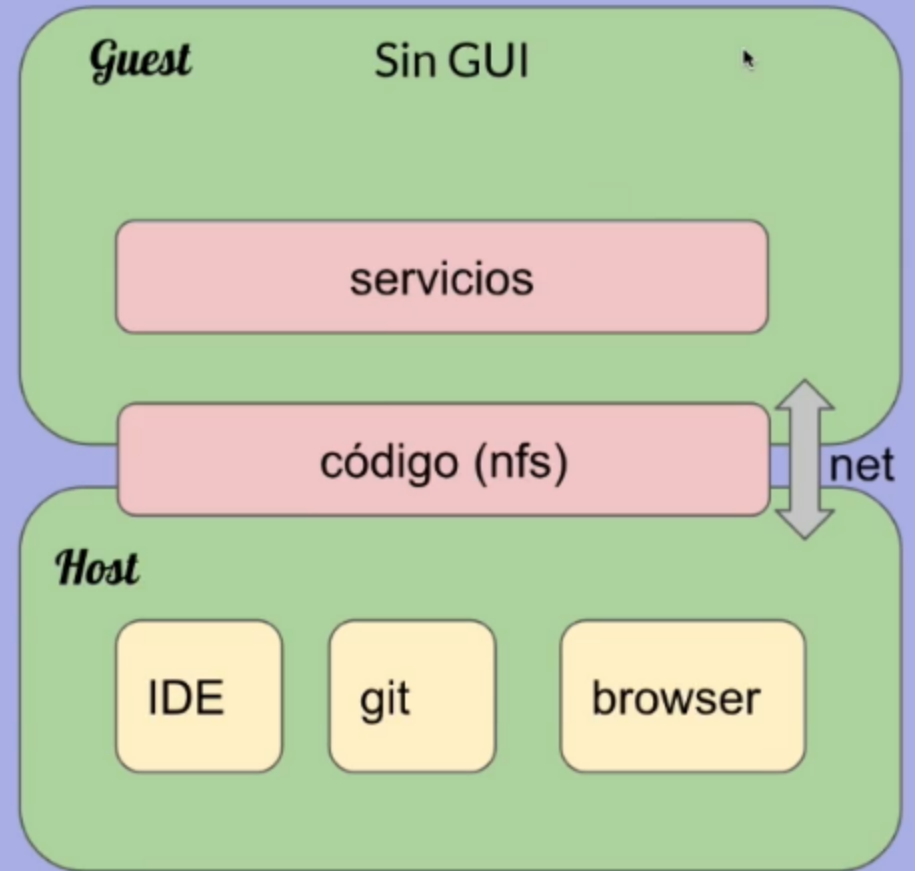
ANSIBLE

Arquitectura

Primera aproximación



Más productivo



Agilizando: creación de máquinas virtuales con vagrant

Con vagrant se agilizan enormemente todas las operaciones relativas a la creación y configuración de máquinas virtuales, además:

- ▶ Se automatiza la instalación y configuración de servicios y software (provisión)
- ▶ Todo lo relativo a la gestión de la máquina virtual queda reflejado en un ***simple archivo de texto***
 - ▷ Sistema operativo base
 - ▷ Configuración (cpu, memoria, disco)
 - ▷ Tipos de redes
 - ▷ Carpetas compartidas
 - ▷ Provisión



Flujo básico de trabajo con vagrant

- ▶ Creamos la piedra angular de vagrant: el fichero Vagrantfile:
 - ▶ `vagrant init ubuntu/xenial64`
- ▶ Levantamos la máquina
 - ▶ `vagrant up`
 - ▶ Si la box no existe, se intenta descargar
- ▶ Accedemos a la máquina
 - ▶ `vagrant ssh`
- ▶ Apagamos la máquina
 - ▶ `vagrant halt`
- ▶ O la suspendemos
 - ▶ `vagrant suspend`

```
Vagrant.configure("2") do |config|  
  # config.vm.box = "base" config es una  
  config.vm.box = "ubuntu/trusty64"  
end
```

Vagrant funciona en maquinas virtuales

Utilizamos VirtualBox

[Oracle VM VirtualBox](#)

[Vagrant by HashiCorp \(vagrantup.com\)](#)



[Intro](#) [Docs](#) [Community](#)



Development Environments Made Easy

[Get Started](#)

[Download 2.2.18](#)

[Find Boxes](#)

Instalación de maquinas virtuales

Download Vagrant

macOS

Windows

Linux

Debian

Centos

Arch Linux

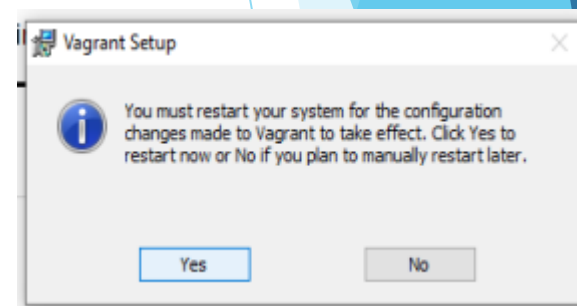
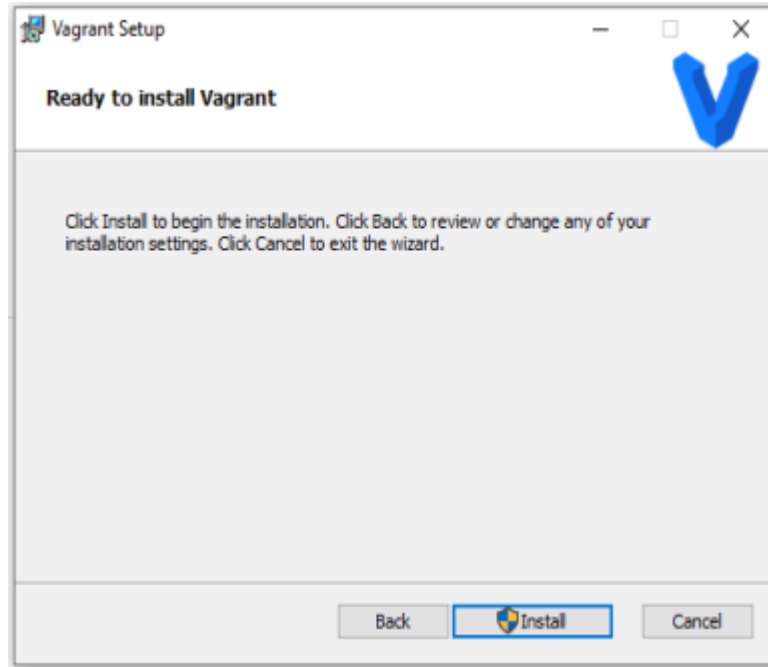
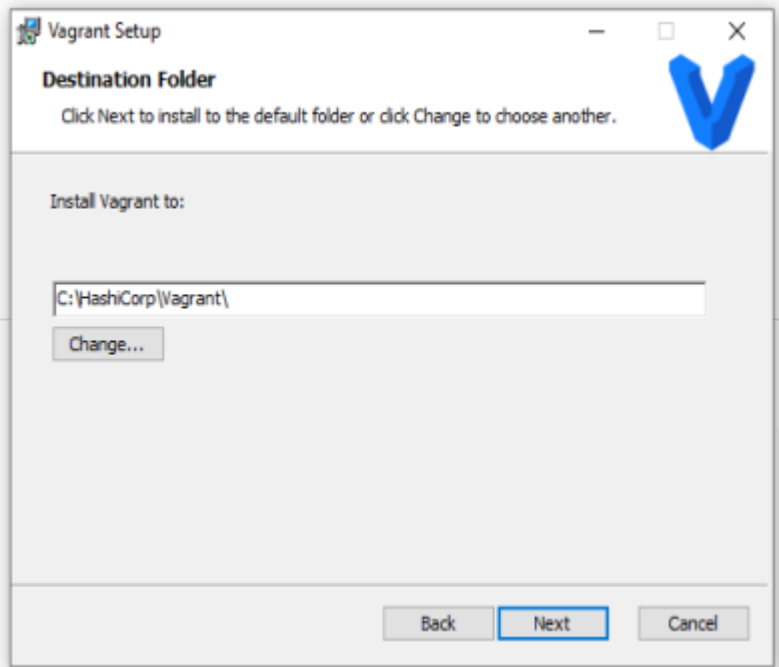
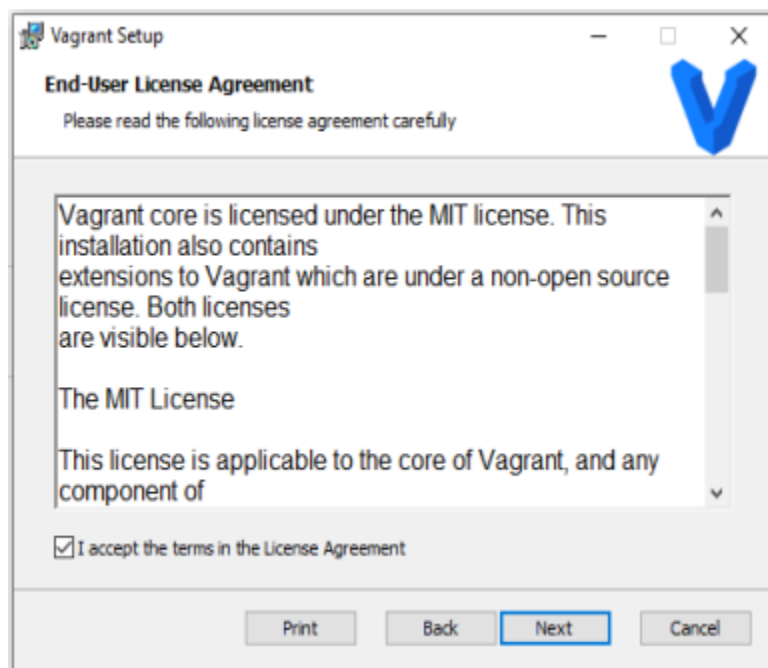
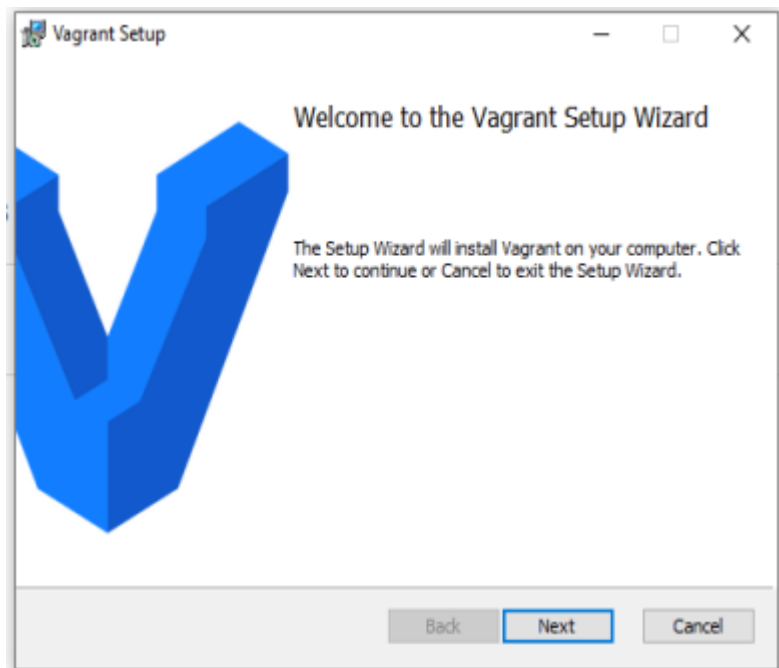
WINDOWS BINARY DOWNLOAD

 **Vagrant** 2.2.19

[32-bit](#) [64-bit](#)

Bandwidth courtesy of

fastly



Para comprobar si se ha instalado correctamente

1.- Abrimos una terminal

2.- Tecleamos “vagrant”

Deberíamos obtener un listado de las opciones

Vagrant version

```
C:\Users\Rafa>vagrant version
Installed Version: 2.2.19
Latest Version: 2.2.19

You're running an up-to-date version of Vagrant!

C:\Users\Rafa>
```

```
C:\Users\Rafa>vagrant
Usage: vagrant [options] <command> [<args>]

    -h, --help                Print this help.

Common commands:
    autocomplete    manages autocomplete installation on host
    box              manages boxes: installation, removal, etc.
    cloud            manages everything related to Vagrant Cloud
    destroy          stops and deletes all traces of the vagrant machine
    global-status    outputs status Vagrant environments for this user
    halt             stops the vagrant machine
    help             shows the help for a subcommand
    init             initializes a new Vagrant environment by creating a Vagrantfile
    login            login
    package          packages a running vagrant environment into a box
    plugin           manages plugins: install, uninstall, update, etc.
    port             displays information about guest port mappings
    powershell       connects to machine via powershell remoting
    provision         provisions the vagrant machine
    push             deploys code in this environment to a configured destination
    rdp              connects to machine via RDP
    reload           restarts vagrant machine, loads new Vagrantfile configuration
    resume           resume a suspended vagrant machine
    snapshot         manages snapshots: saving, restoring, etc.
    ssh              connects to machine via SSH
    ssh-config       outputs OpenSSH valid configuration to connect to the machine
    status           outputs status of the vagrant machine
    suspend          suspends the machine
    up               starts and provisions the vagrant environment
    upload           upload to machine via communicator
    validate         validates the Vagrantfile
    version          prints current and latest Vagrant version
    winrm            executes commands on a machine via WinRM
    winrm-config     outputs WinRM configuration to connect to the machine

For help on any individual command run `vagrant COMMAND -h`

Additional subcommands are available, but are either more advanced
or not commonly used. To see all subcommands, run the command
`vagrant list-commands`.

    --[no-]color            Enable or disable color output
    --machine-readable      Enable machine readable output
    -v, --version           Display Vagrant version
    --debug                 Enable debug output
    --timestamp             Enable timestamps on log output
    --debug-timestamp       Enable debug output with timestamps
    --no-tty                Enable non-interactive output

C:\Users\Rafa>
```


¿Qué es una box y como las obtenemos?

Imágenes base que utilizamos en vagrant para crear nuestras maquinas virtuales

[Discover Vagrant Boxes - Vagrant Cloud \(vagrantup.com\)](https://vagrantup.com/boxes)

[A list of base boxes for Vagrant - Vagrantbox.es](https://vagrantbox.es/)

[Máquinas virtuales - Microsoft Edge Developer](https://docs.microsoft.com/en-us/edge/develop/virtual-machines)

Vagrant box list .- Nos permite ver un listado de los boxes instalados

```
C:\Users\Rafa>vagrant box list
There are no installed boxes! Use `vagrant box add` to add some.
C:\Users\Rafa>
```

Usaremos “vagrant box add” para añadir los boxes que vamos a utilizar

Automáticamente tenemos la carpeta de nuestro proyecto enlazada en la maquina

```
vagrant@vagrant-ubuntu-trusty-64:~$ ls
vagrant@vagrant-ubuntu-trusty-64:~$ cd /
vagrant@vagrant-ubuntu-trusty-64:/$ ls
bin  dev  home  initrd.img.old  lib64  media  opt  root  sbin  sys  usr  var  vmlinuz.old
boot  etc  initrd.img  lib  lost+found  mnt  proc  run  srv  tmp  vagrant  vmlinuz
vagrant@vagrant-ubuntu-trusty-64:/$ cd vagrant
vagrant@vagrant-ubuntu-trusty-64:/vagrant$ ls
Config  proyecto2  Vagrantfile
```

```
C:\Users\Rafa>vagrant box add ubuntu/trusty64
==> box: Loading metadata for box 'ubuntu/trusty64'
    box: URL: https://vagrantcloud.com/ubuntu/trusty64
==> box: Adding box 'ubuntu/trusty64' (v20190514.0.0) for provider: virtualbox
    box: Downloading: https://vagrantcloud.com/ubuntu/boxes/trusty64/versions/20190514.0.0/providers/virtualbox.box
Download redirected to host: cloud-images.ubuntu.com
    box:
==> box: Successfully added box 'ubuntu/trusty64' (v20190514.0.0) for 'virtualbox'!
C:\Users\Rafa>
```

Vagrant init Ubuntu/xenial64

```
C:\Users\Rafa>vagrant box list
ubuntu/trusty64 (virtualbox, 20190514.0.0)

C:\Users\Rafa>_
```

Para crear nuestro fichero de configuración - vagrant init

```
C:\Users\Rafa\Desktop\vagrant>vagrant init
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
```

```
Vagrant.configure("2") do |config|
  # config.vm.box = "base" config es una variable que guarda todos los valores de configuración
  config.vm.box = "ubuntu/trusty64"
end
```

Áqui colocaríamos el box que nos hemos descargado

Donde se almacenan las box de vagrant

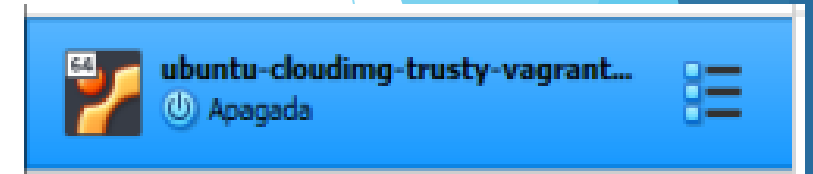
C:\Users\rafac\.vagrant.d

Como se arranca la box



Vagrant up

Si ejecutamos virtualbox podriamos observar



Como podemos acceder a la maquina

Vagrant ssh

Para salir de la maquina

exit

```
vagrant@vagrant-ubuntu-trusty-64:~$ exit
logout
Connection to 127.0.0.1 closed.
```

Vagrant halt - para apagar la máquina

Vagrant destroy - para eliminar cualquier maquina

```
C:\Users\rafac\OneDrive\Escritorio\vagrant>vagrant ssh
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 3.13.0-170-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information disabled due to load higher than 1.0

UA Infrastructure Extended Security Maintenance (ESM) is not enabled.
0 updates can be installed immediately.
0 of these updates are security updates.

Enable UA Infrastructure ESM to receive 64 additional security updates.
See https://ubuntu.com/advantage or run: sudo ua status

New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

vagrant@vagrant-ubuntu-trusty-64:~$
```