GESTION DE LA VIRTUALIZACION CON VAGRANT

Juan Carlos Navidad García

Sistemas Operativos en Red

1. Instalación de Git en Windows:

He instalado **Git** mediante interfaz gráfica, descargando Git desde la página del creador.

2. Instalación de Ubuntu en Windows:

Para instalar Ubuntu en Windows, primero he tenido que activar **Hyper-V** y **WSL 2.0**, después de reiniciar he descargado **Ubuntu 20.04** desde la Microsoft Store, posteriormente lo he configurado y al arrancar WSL desde Powershell, ya puedo utilizar la shell de Ubuntu en Windows.

3. Instalación para Windows y para Ubuntu de:

a. VirtualBox:

i. **Windows:** La Instalación de VirtualBox en Windows es muy simple, te vas a la página del fabricante y te descargas el ejecutable, lo instalas y listo.

ii. **Ubuntu:** Primero de todo hay que actualizar la lista de repositorios con "**apt update**" siempre con derechos de administrador, si no pones "**sudo**" delante. Una vez los repositorios actualizados, escribes "**apt-get install virtualbox**" y se nos empezará a instalar VirtualBox en Ubuntu.

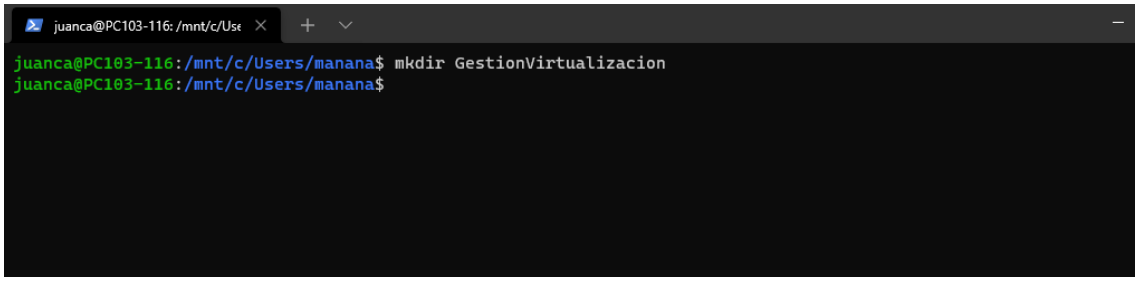
b. Vagrant:

i. **Windows:** el proceso es similar al de VirtualBox, únicamente hay que descargarse el ejecutable desde la página del desarrollador e instalarlo.

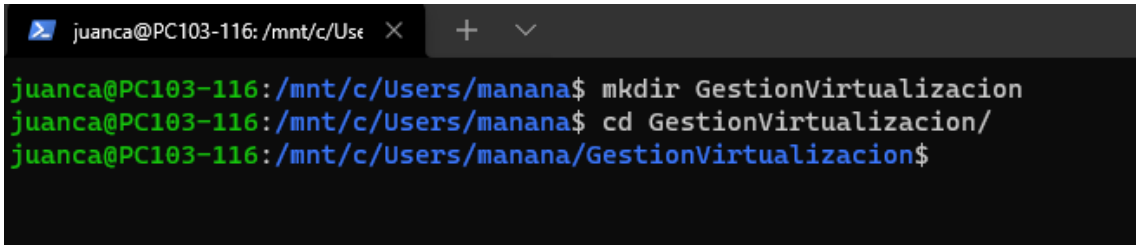
ii. **Ubuntu:** este es más complejo. Lo primero de todo es instalar Curl, Curl es una herramienta para transferir datos hacia o desde un servidor. Se instala con "**apt install curl**". Una vez instalado Curl, vamos a descargar el paquete de Vagrant desde su repositorio Web poniendo: "**curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -**" Una vez descargado, añadimos el repositorio "**sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb_release -cs) main**" procedemos a instalarlo con: "**sudo apt-get update && sudo apt-get install vagrant**" y ya debería de estar instalado.

4. Desde PowerShell:

a. Crear una carpeta dentro de tu usuario con el nombre de GestionVirtualizacion

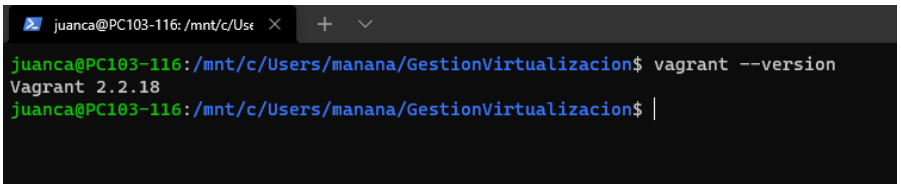


b. Accede a ella

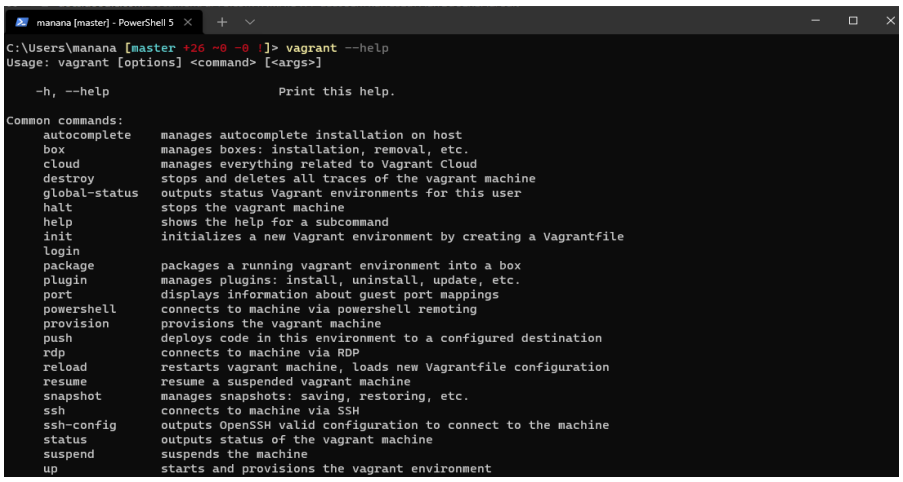


c. Ejecuta los comandos de vagrant para

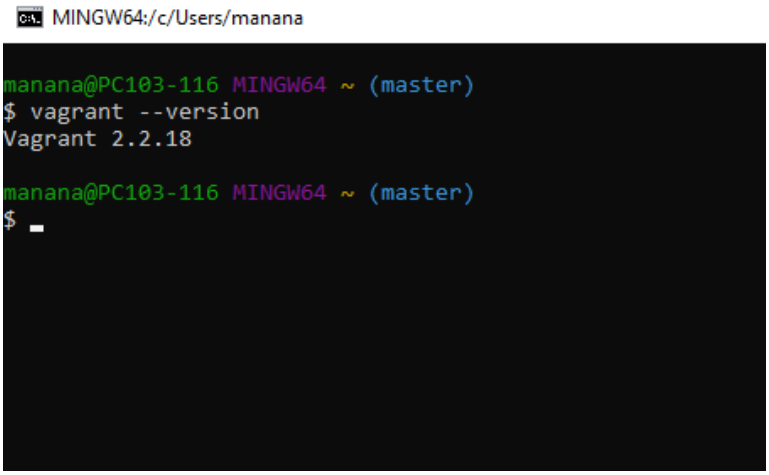
i. Ver la versión de vagrant



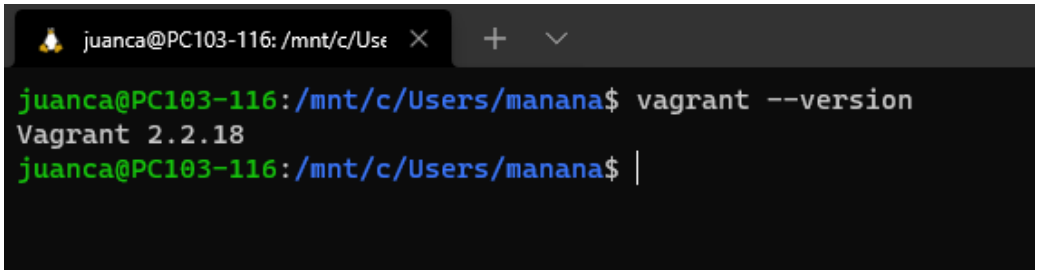
ii. Mostrar conjunto de comandos de vagrant



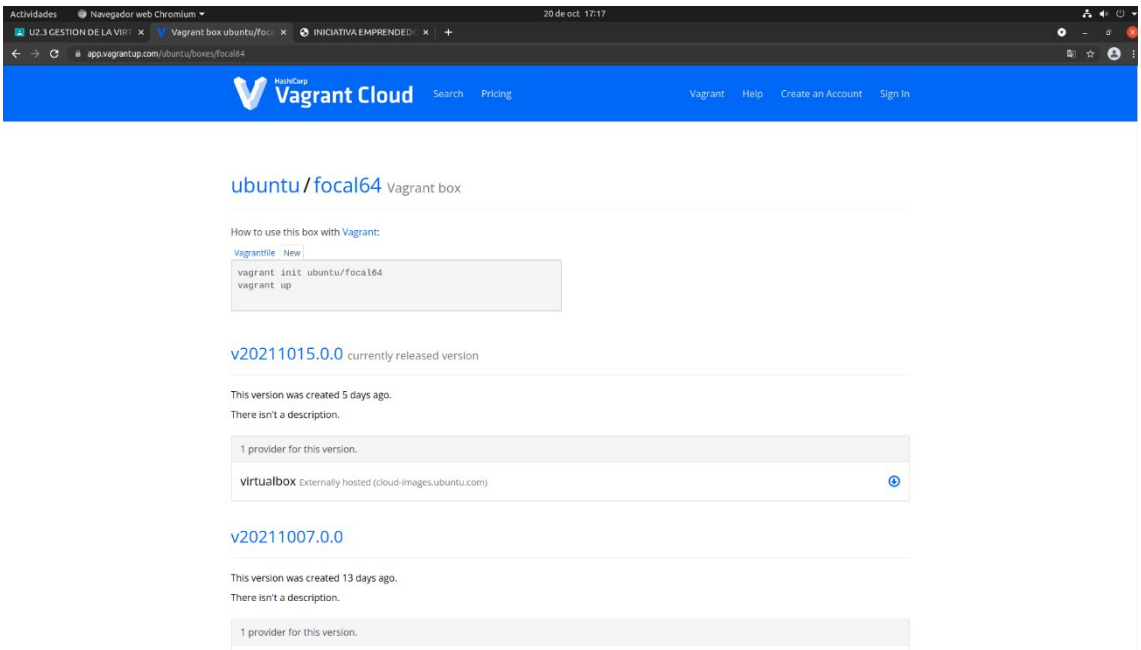
5. Desde Git bash comprobar que puedo acceder a vagrant ejecutando un comando cualquiera.



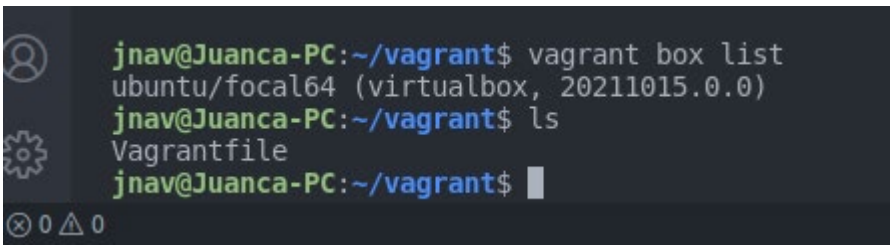
6. Lo mismo desde Ubuntu en Windows:



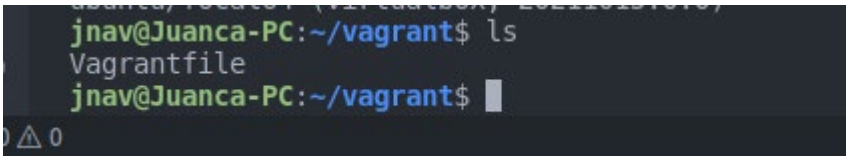
7. Desde la página de vagrant, busca algunas boxes y añádelas al sistema.



8. Listas las boxes que tienes



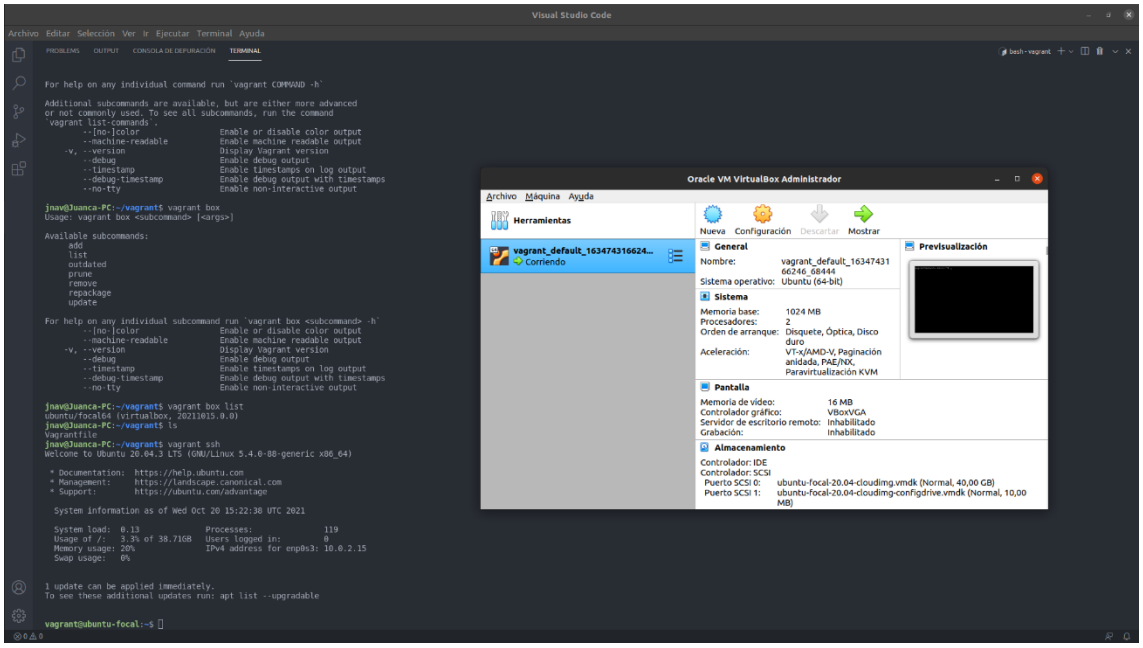
9. Verificar las boxes descargadas mirando el directorio donde se almacenan las boxes



10. Verificad:

Tendremos dos escenarios compatibles.

Si accedemos a virtual box veríamos las maquinas.

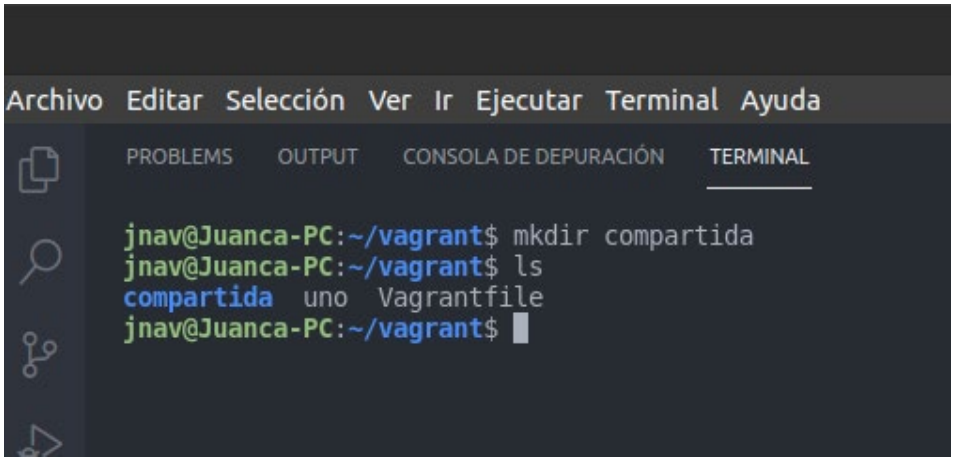


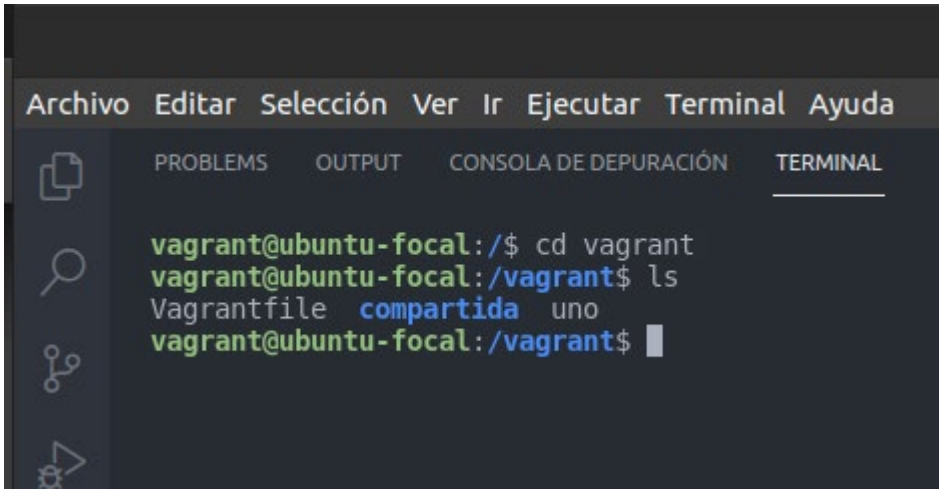
11. Directorio Compartido:

1. Accede a la maquina;
2. Comprueba que puedes ver el fichero vagrant en /vagrant desde la máquina virtual
3. Crea un fichero de texto en /vagrant
4. Sal de la maquina
5. Verifica desde el fichero sincronizado de la maquina anfitrión que puedes ver y modificar los ficheros anteriores.

```
jnav@Juanca-PC:~/vagrant$ vagrant ssh
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Oct 20 15:25:04 UTC 2021

  System load:  0.01              Processes:               111
  Usage of /:   3.3% of 38.71GB   Users logged in:         1
  Memory usage: 20%               IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%


1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Wed Oct 20 15:23:05 2021
vagrant@ubuntu-focal:~$ ls
vagrant@ubuntu-focal:~$ cd /vagrant
vagrant@ubuntu-focal:/vagrant$ nano uno
vagrant@ubuntu-focal:/vagrant$ exit
logout
Connection to 127.0.0.1 closed.
jnav@Juanca-PC:~/vagrant$ ls
uno  Vagrantfile
jnav@Juanca-PC:~/vagrant$
```

6. crea una carpeta llamada compartida dentro de nuestro anfitrión (Windows) y verifica que se puede acceder desde la máquina virtual.

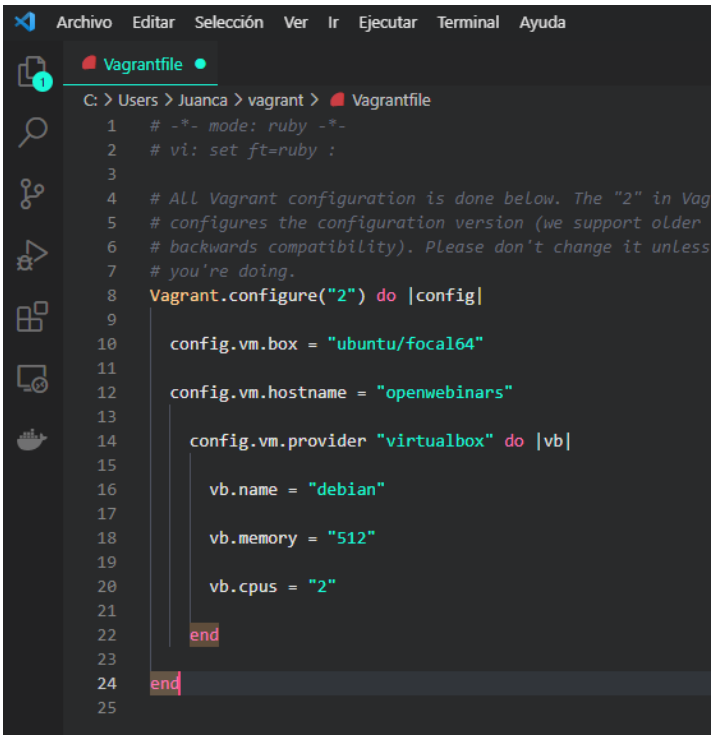




12. MODIFICACION HARDWARE MAQUINA VIRTUAL:

1. Crea un nuevo proyecto llamado p1 para recrear la configuración de la maquina descrita anteriormente:

- Debian/jessie64
- Hostname: openwebinars
- Provedor virtualbox:
- Nombre: debian
- Menoria 512
- 2 cpus



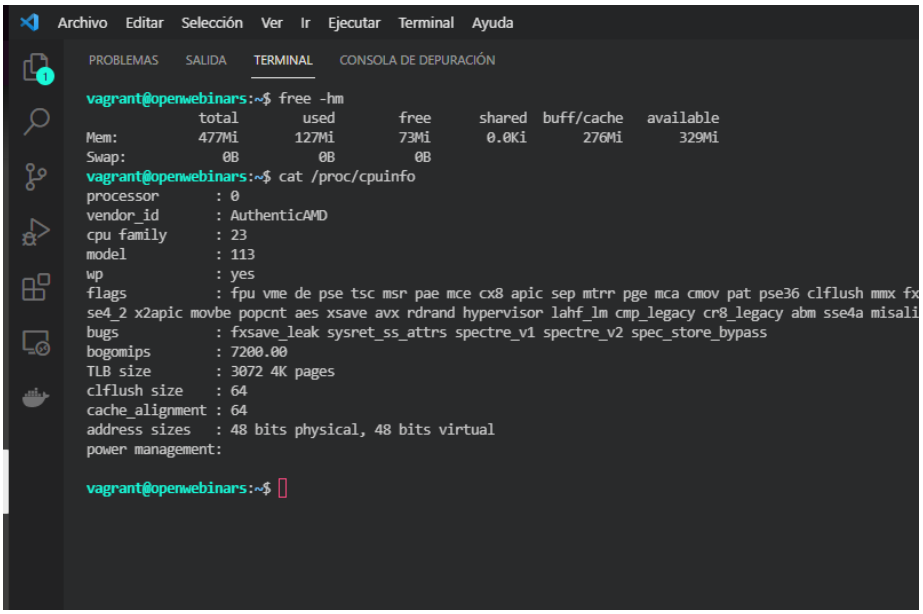
2. Verifico las características de la maquina:

1. Abriendo virtualbox

2. Entrando en la máquina virtual (vagrant ssh)

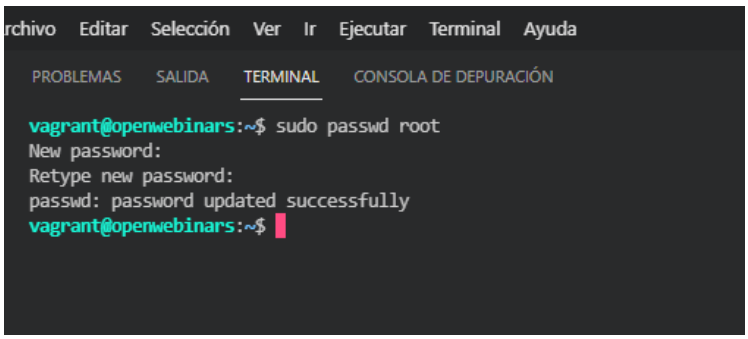
i. Free -hm

ii. Cat /proc/cpuinfo



3. Activa el usuario root

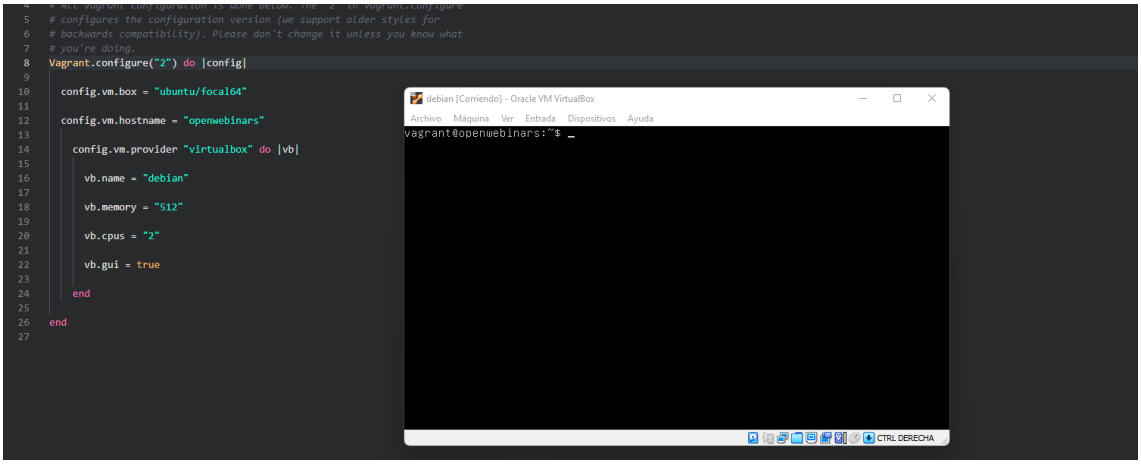
1. Sudo passwd root



4. Destruye la maquina (destroy)

```
PS C:\Users\Juanca\vagrant> vagrant destroy
    default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Destroying VM and associated drives...
PS C:\Users\Juanca\vagrant>
```

5. Créala de nuevo, pero con interfaz grafica



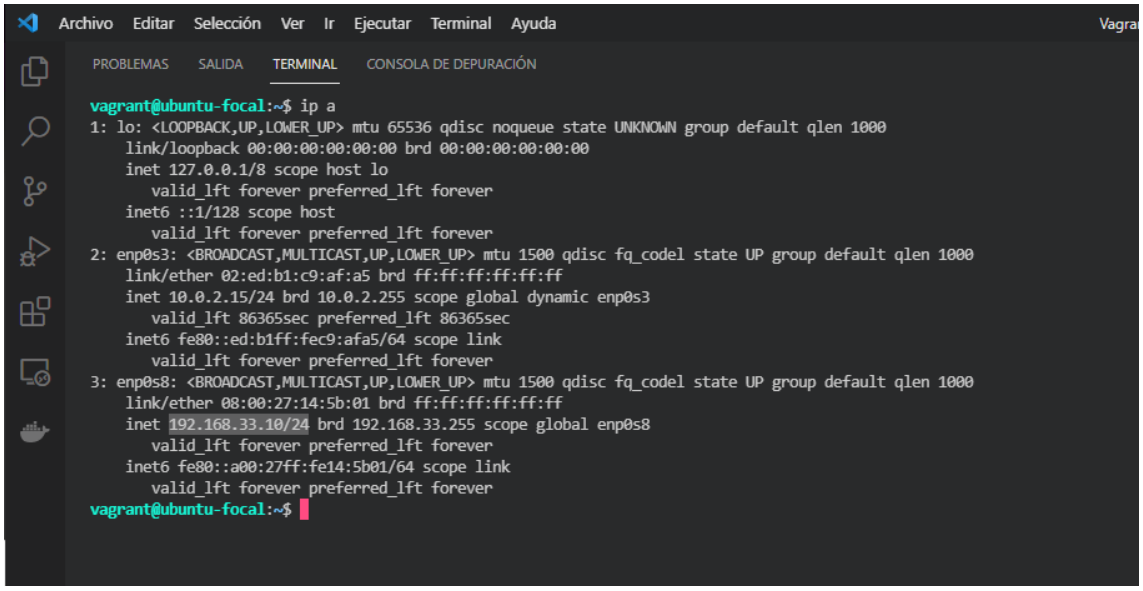
13. Redes privadas:

1. Crea un nuevo Proyecto p3redes

   a. Ubuntu/trusty64

   b. Añade una red privada en 192.168.33.10

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

  config.vm.box = "ubuntu/focal64"

  config.vm.network "private_network", ip: "192.168.33.10"


end
```



c. Levanta la maquina

d. Accede mediante las dos interfaces

```
PS C:\Users\Juanca\vagrant> vagrant ssh
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-89-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Oct 24 12:26:25 UTC 2021

  System load:  0.05              Processes:              118
  Usage of /:   3.3% of 38.71GB   Users logged in:        0
  Memory usage: 19%               IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%                IPv4 address for enp0s8: 192.168.33.10


1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Sun Oct 24 12:24:14 2021 from 192.168.33.1
vagrant@ubuntu-focal:~$
```

```
PS C:\Users\Juanca\vagrant> ssh -i .vagrant/machines/default/virtualbox/private_key vagrant@192.168.33.10
The authenticity of host '192.168.33.10 (192.168.33.10)' can't be established.
ECDSA key fingerprint is SHA256:fGJjHyZb52YUDxSbxF1o4btx8ijF3E0qYheZGAVILPk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.33.10' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-89-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Oct 24 12:24:14 UTC 2021

  System load:  0.46              Processes:              118
  Usage of /:   3.3% of 38.71GB   Users logged in:        0
  Memory usage: 19%               IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%                IPv4 address for enp0s8: 192.168.33.10


1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Sun Oct 24 12:23:08 2021 from 10.0.2.2
vagrant@ubuntu-focal:~$
```

14. Redes públicas:

1. Modifica el proyecto anterior para que utilice una red publica

```
8   Vagrant.configure("2") do |config|
9
10    config.vm.box = "ubuntu/focal64"
11
12    config.vm.network "public_network"
13
14  end
15
```

2. Levantamos la maquina vagrant up

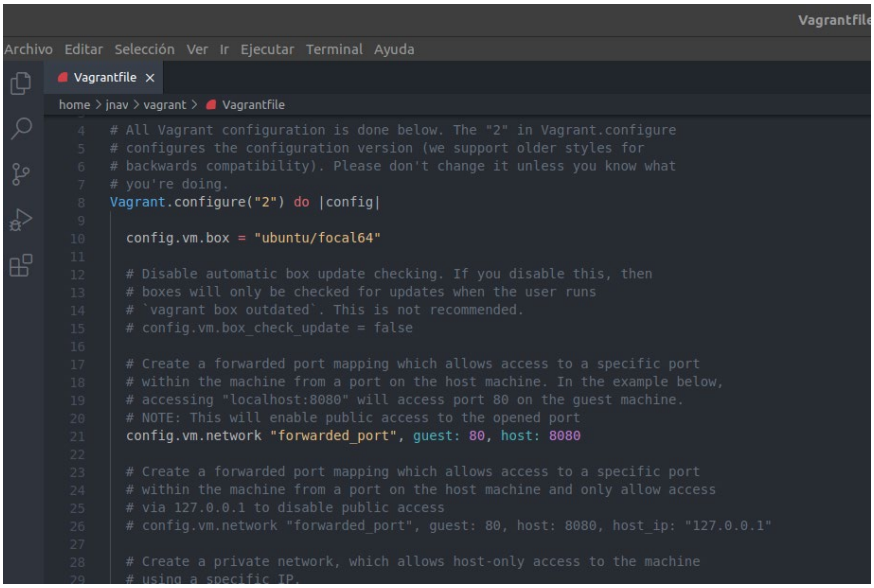
3. Me conecto vagrant ssh

4. Veo el direccionamiento ip

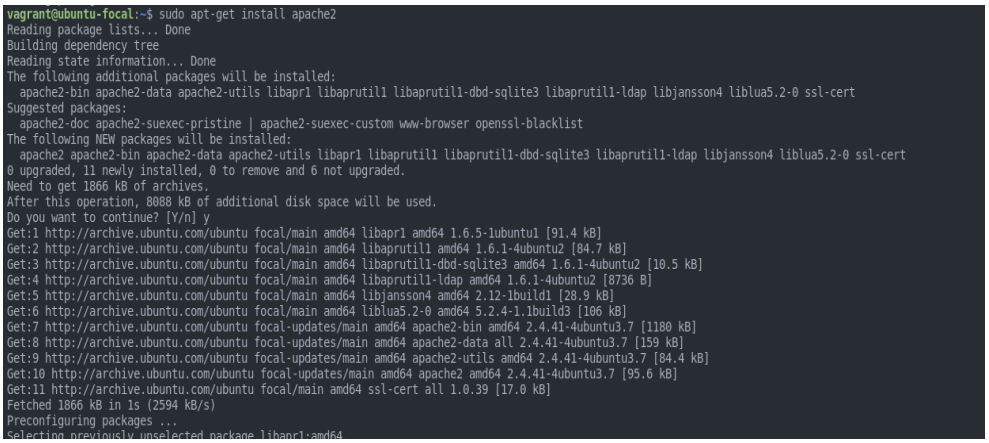
```
vagrant@ubuntu-focal:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 10
    link/ether 02:ed:b1:c9:af:a5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 86346sec preferred_lft 86346sec
    inet6 fe80::ed:b1ff:fec9:afa5/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 10
    link/ether 08:00:27:14:5b:01 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.40/24 brd 192.168.1.255 scope global dynamic enp0s8
       valid_lft 86349sec preferred_lft 86349sec
    inet6 fe80::a00:27ff:fe14:5b01/64 scope link
       valid_lft forever preferred_lft forever
vagrant@ubuntu-focal:~$
```

15. Servidor Web:

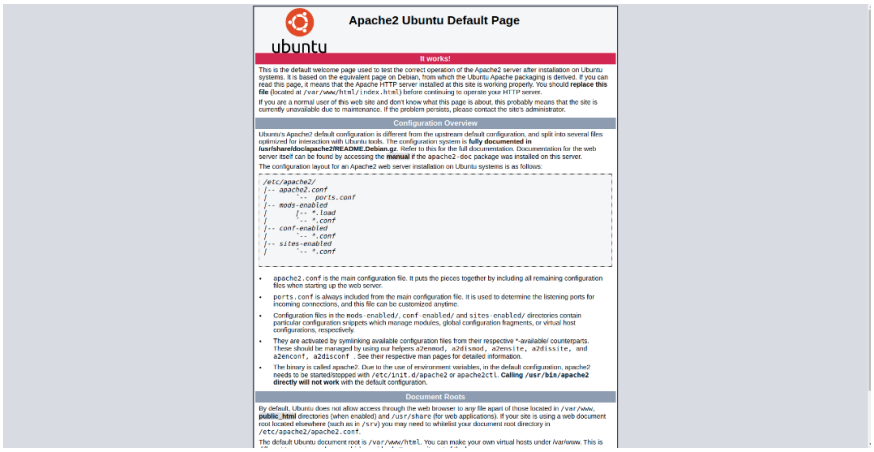
1. Modifica el fichero vagrant de la maquina debian para que se pueda acceder a un servidor web en la máquina virtual(guest) en el puerto 80 sabiendo que el puerto que se utiliza en la maquina anfitriona es el 8080



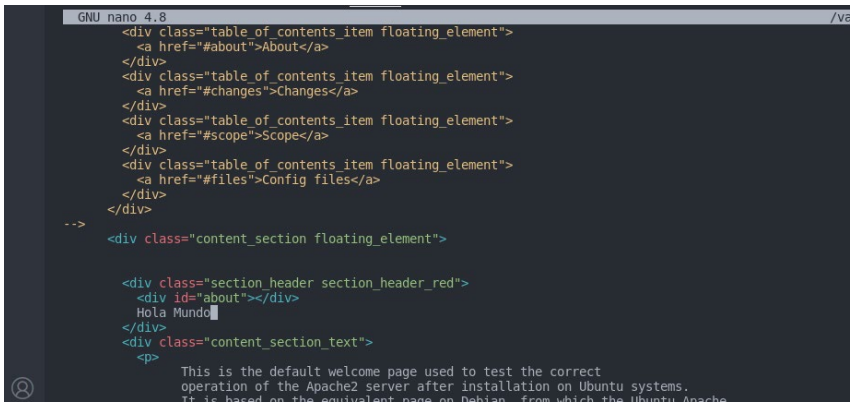
2. Levanta la maquina
3. Vagrant up
4. Vagrant ssh
5. Actualizamos e Instalamos el servicio de apache



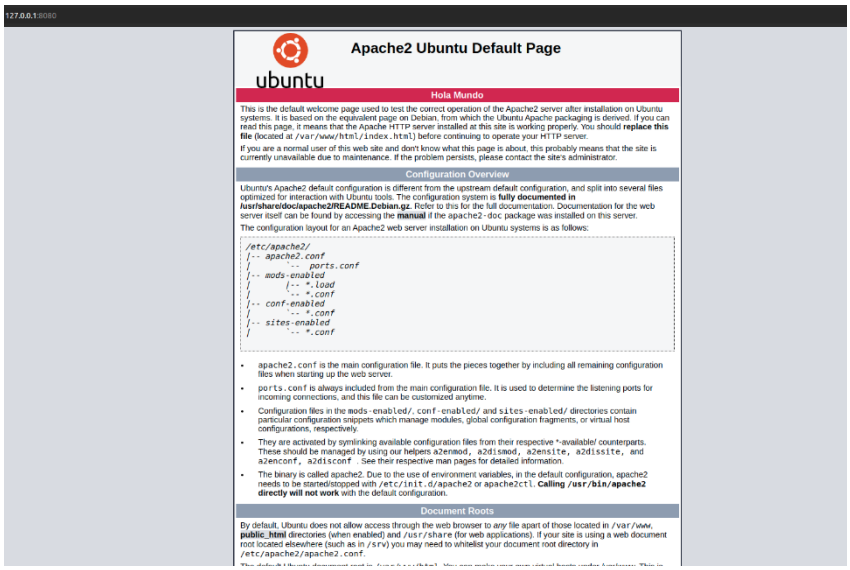
6. Abrimos el navegador



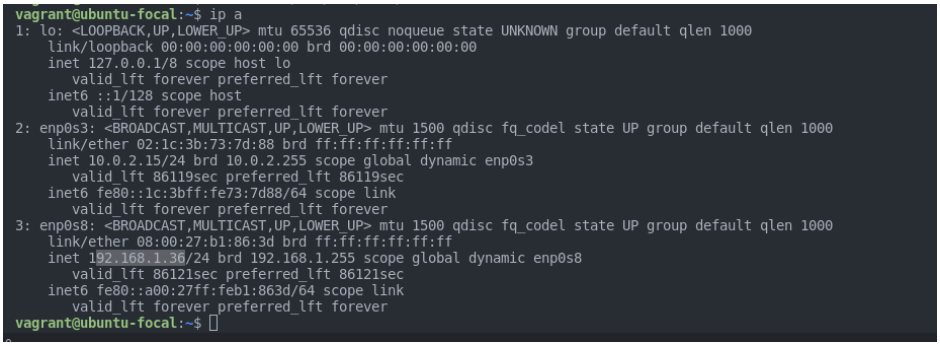
7. Accedemos al archivo index.html para cambiarlo y que aparezco hola mundo



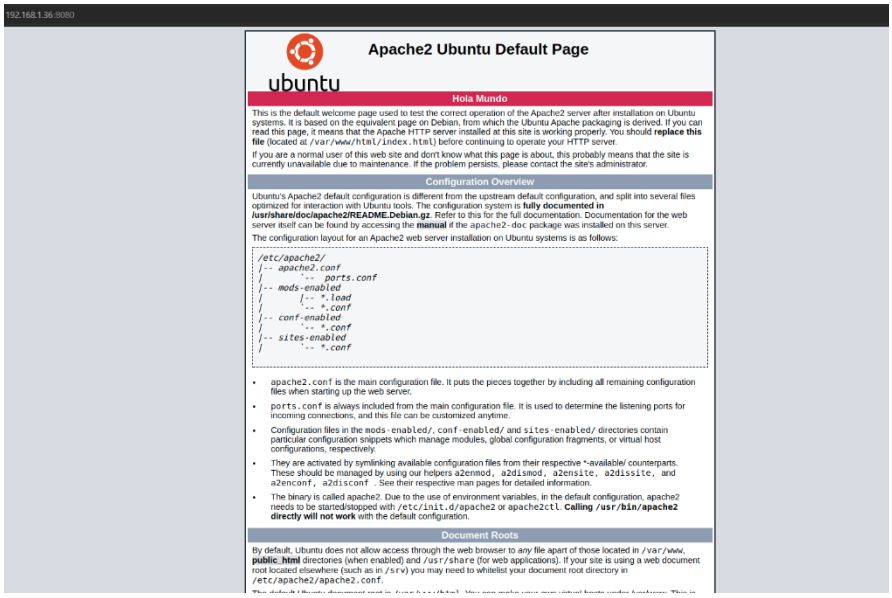
8. Vuelvo a acceder al navegador para ver la modificación



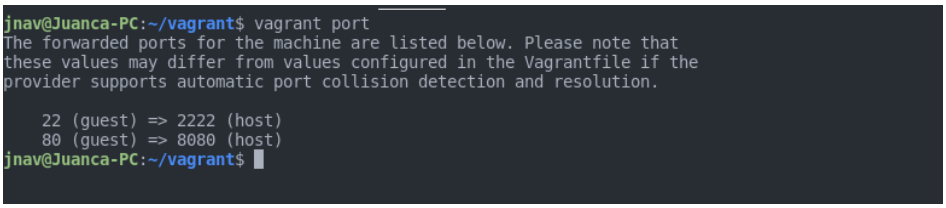
9. Salgo de la máquina para ver la ip de mi red y Compruebo en la maquina está dentro de mi red local. Miro la ip de la maquina



10. Accedo a internet desde esa ip



11. Vagrant port



16. Multientornos:

1. Configurar el Vagrantfile:

```
# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|

  config.vm.define "maquina1" do |m1|

    m1.vm.box = "ubuntu/focal64"

  end

  config.vm.define "maquina2" do |m2|

    m2.vm.box = "debian/jessie64"

  end

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
```

2. Vagrant reload

3. Vagrant up

4. Inicia las máquinas en terminales diferentes:

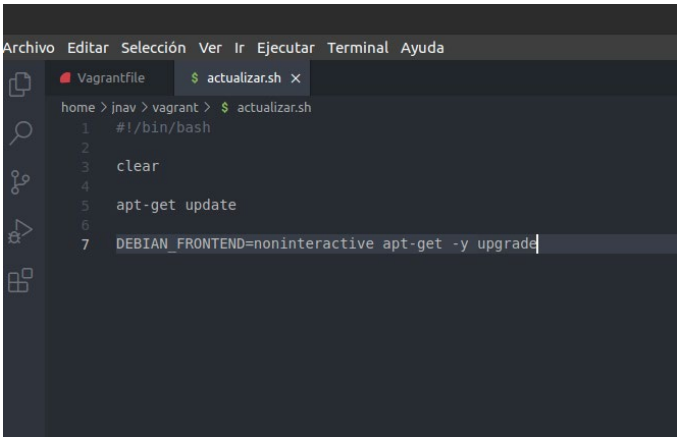
5. Haz un ping entre ellas:

```
vagrant@ubuntu-focal:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:1c:3b:73:7d:88 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 86271sec preferred_lft 86271sec
    inet6 fe80::1c:3bff:fe73:7d88/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:0c:e8:34 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.37/24 brd 192.168.1.255 scope global dynamic enp0s8
       valid_lft 86273sec preferred_lft 86273sec
    inet6 fe80::a00:27ff:fe0c:e834/64 scope link
       valid_lft forever preferred_lft forever
vagrant@ubuntu-focal:~$ ping 192.168.1.39
PING 192.168.1.39 (192.168.1.39) 56(84) bytes of data.
64 bytes from 192.168.1.39: icmp_seq=1 ttl=64 time=0.308 ms
64 bytes from 192.168.1.39: icmp_seq=2 ttl=64 time=0.320 ms
64 bytes from 192.168.1.39: icmp_seq=3 ttl=64 time=0.226 ms
64 bytes from 192.168.1.39: icmp_seq=4 ttl=64 time=0.313 ms
^C
--- 192.168.1.39 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3090ms
rtt min/avg/max/mdev = 0.226/0.291/0.320/0.038 ms
vagrant@ubuntu-focal:~$
```

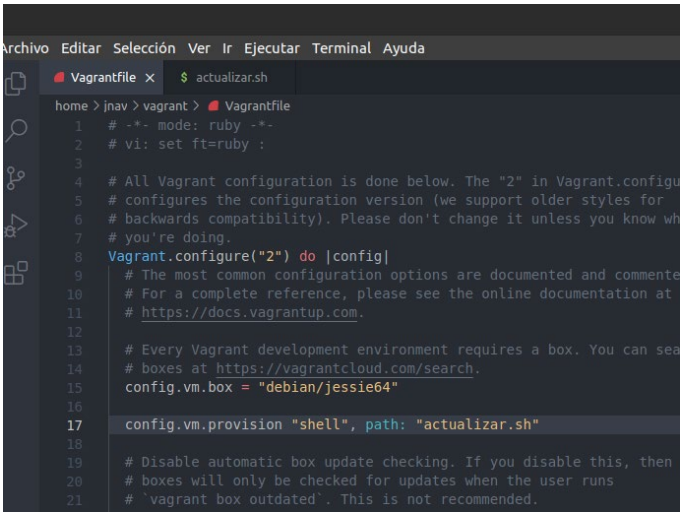
```
vagrant@jessie:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:8d:c0:4d brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe8d:c04d/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:bd:36:81 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.39/24 brd 192.168.1.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:febd:3681/64 scope link
       valid_lft forever preferred_lft forever
vagrant@jessie:~$ ping 192.168.1.37
PING 192.168.1.37 (192.168.1.37) 56(84) bytes of data.
64 bytes from 192.168.1.37: icmp_seq=1 ttl=64 time=0.472 ms
64 bytes from 192.168.1.37: icmp_seq=2 ttl=64 time=0.475 ms
64 bytes from 192.168.1.37: icmp_seq=3 ttl=64 time=0.680 ms
64 bytes from 192.168.1.37: icmp_seq=4 ttl=64 time=0.314 ms
64 bytes from 192.168.1.37: icmp_seq=5 ttl=64 time=0.332 ms
^C
--- 192.168.1.37 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4030ms
rtt min/avg/max/mdev = 0.314/0.454/0.680/0.133 ms
```

17. Actualización automática con script:

1. Creamos un script .sh llamado actualizar.sh:

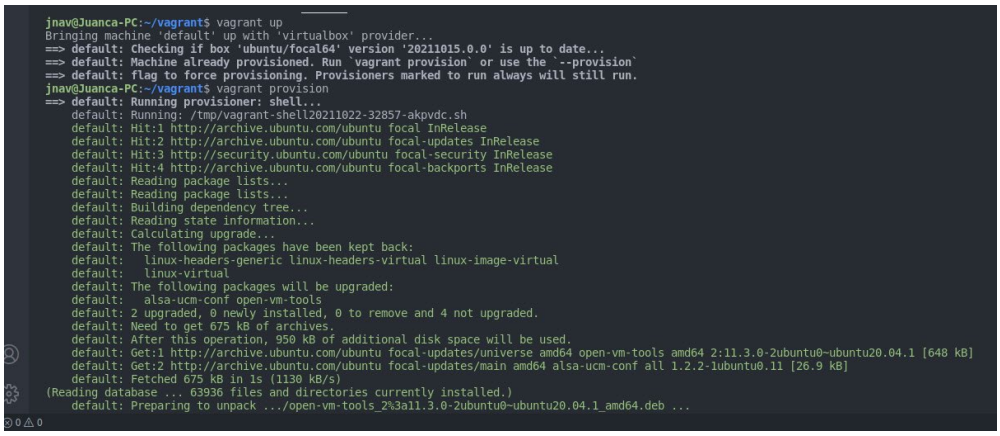


2. Actualizamos el Vagrantfile:



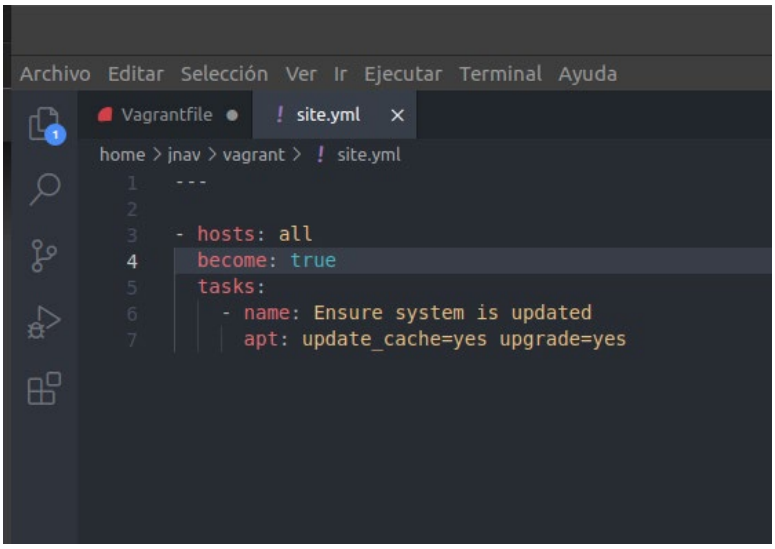
3. Levantamos la máquina

4. Hacemos un vagrant provision:

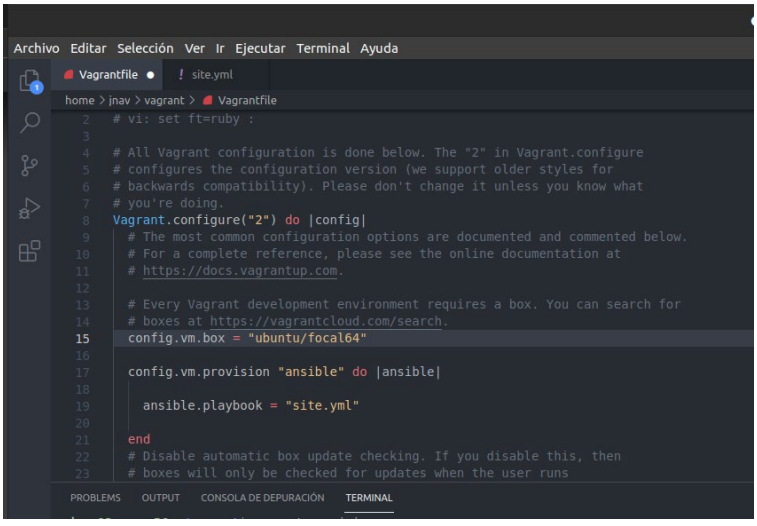


18. Actualización automática con Ansible:

1. Creamos un script en Yamal:



2. Actualizamos el Vagrantfile:



3. Levantamos la máquina

4. Hacemos un vagrant provision: