**Juan Carlos Navidad García**

**Sistemas Operativos en Red**

GESTION DE LA VIRTUALIZACION CON

VAGRANT



## Instalación de Git en Windows:

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

## 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.

## Instalación para Windows y para Ubuntu de:

### VirtualBox:

* + 1. **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.
    2. **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.

### Vagrant:

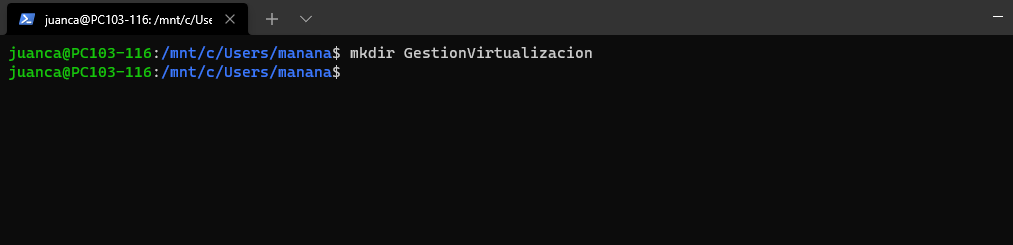
* + 1. **Windows:** el proceso es similar al de VirtualBox, únicamente hay que descargarse el ejecutable desde la página del desarrollador e instalarlo.
    2. **Ubuntu:** este es más complejo. Lo primero de todo es instalar Curl, Curl es una herramienta para transferir datos hacia o desde un servidor. 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

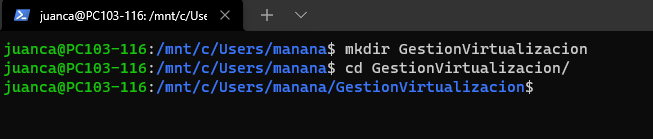
-” Una vez descargado, añadimos el repositorio “sudo apt-add- repository "deb [arch=amd64] https://apt.releases.hashicorp.com

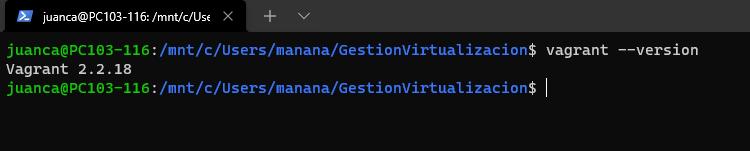
$(lsb\_release -cs) main ”” procedemos a instalarlo con: “sudo apt-get update && sudo apt-get install vagrant” y ya debería de estar instalado.

## Desde PowerShell:

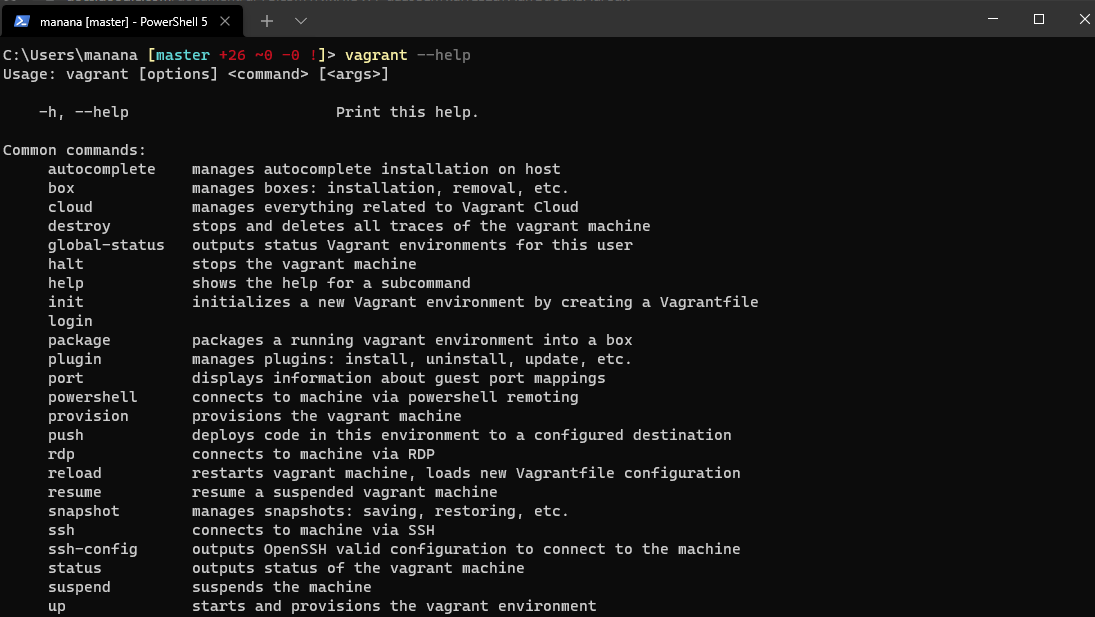
* 1. **Crear una carpeta dentro de tu usuario con el nombre de GestionVirtualizacion**



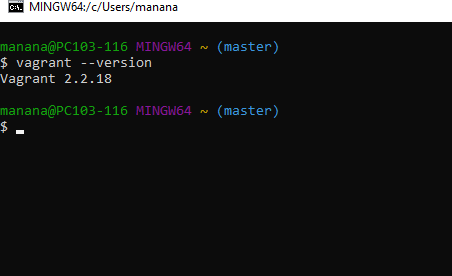
* 1. **Accede a ella**
  2. **Ejecuta los comandos de vagrant para**
     1. **Ver la versión de vagrant**



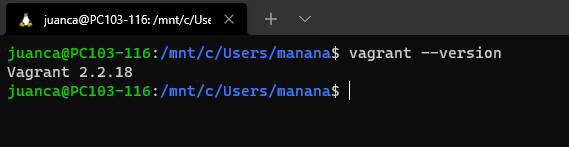
* + 1. **Mostrar conjunto de comandos de vagrant**



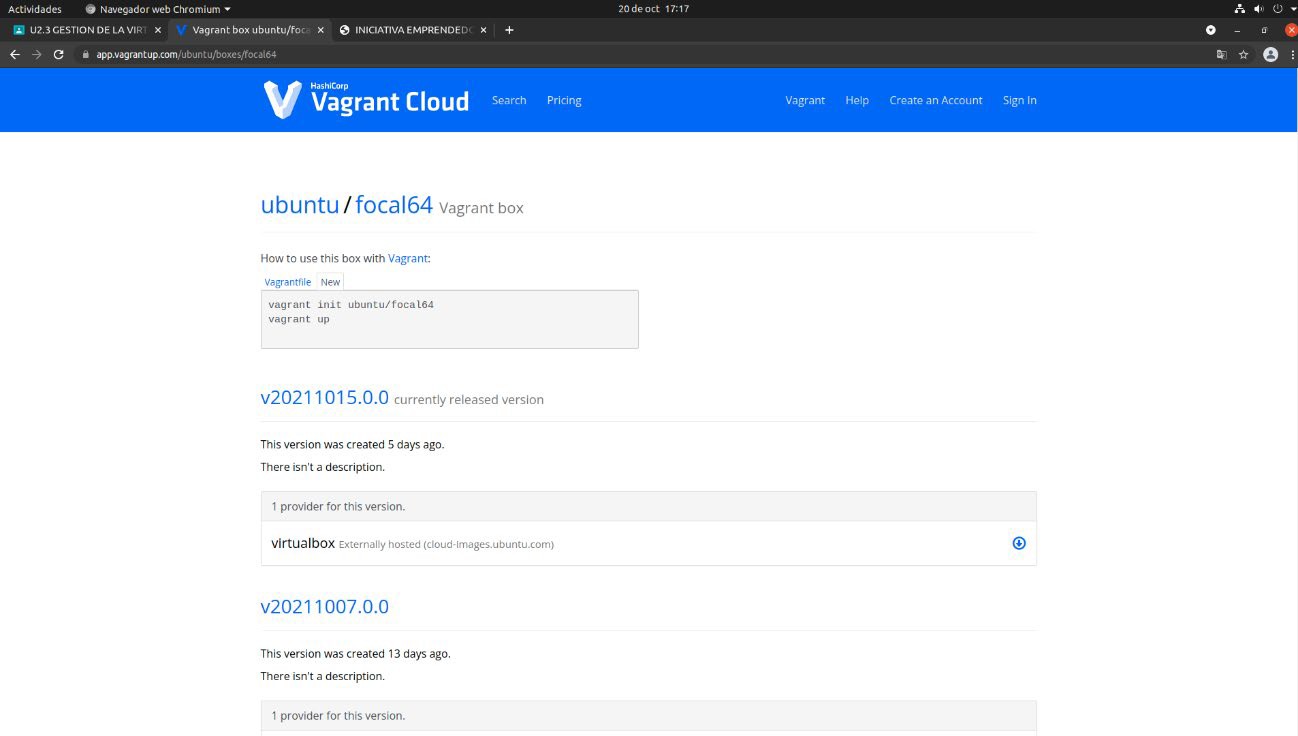
## Desde Git bash comprobar que puedo acceder a vagrant ejecutando un comando cualquiera.



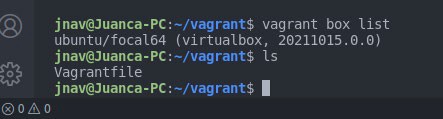
1. **Lo mismo desde Ubuntu en Windows:**



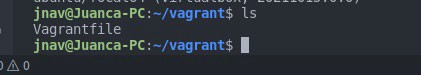
1. **Desde la página de vagrant, busca algunas boxes y añádelas al sistema.**



1. **Listas las boxes que tienes**



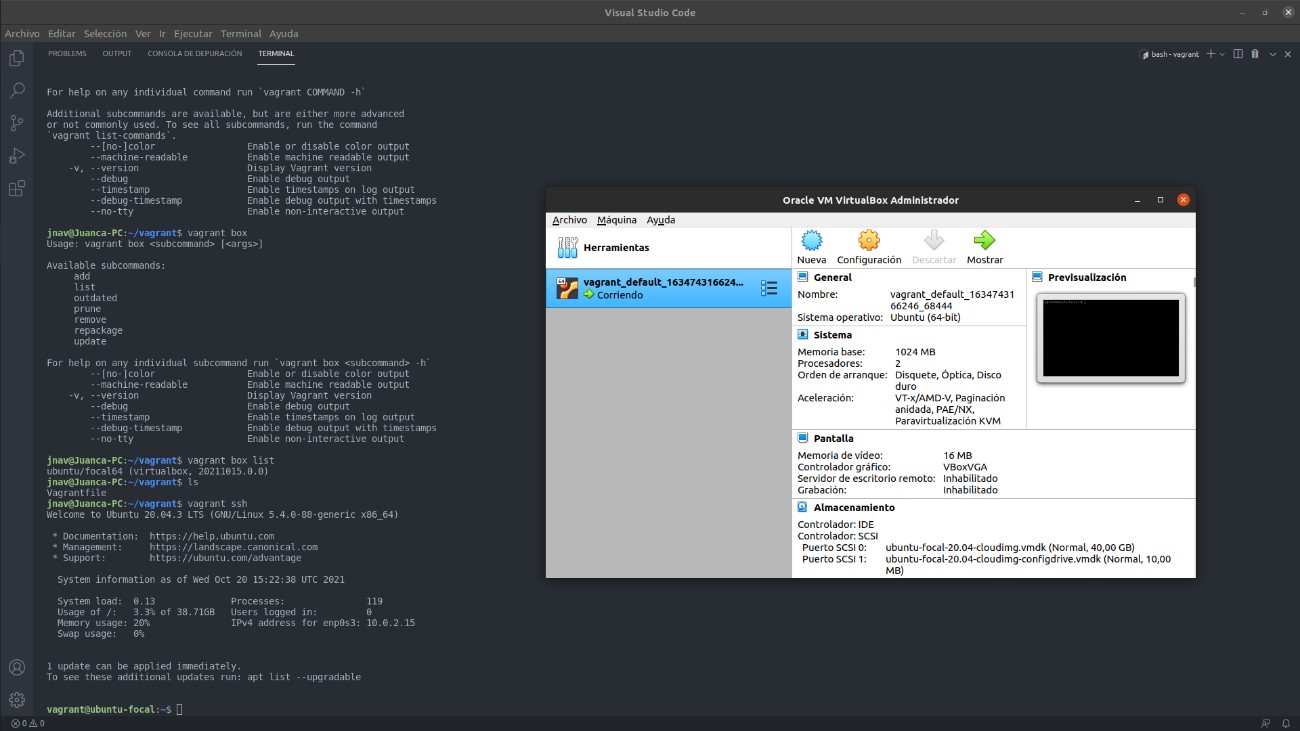
1. **Verificar las boxes descargadas mirando el directorio donde se almacenan las boxes**



1. **Verificad:**

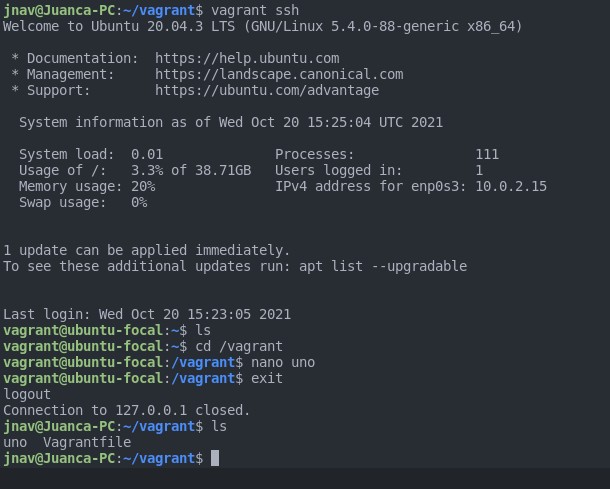
Tendremos dos escenarios compatibles.

Si accedemos a virtual box veríamos las maquinas.

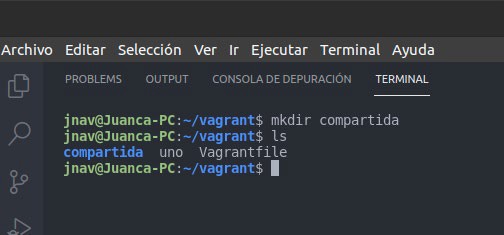


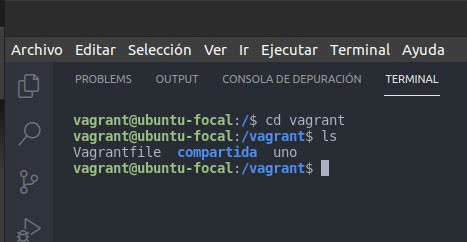
## Directorio Compartido:

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.



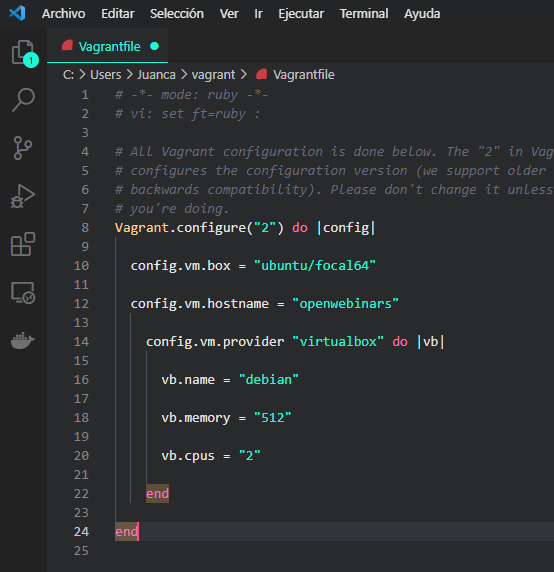
1. crea una carpeta llamada compartida dentro de nuestro anfitrión (Windows) y verifica que se puede acceder desde la máquina virtual.

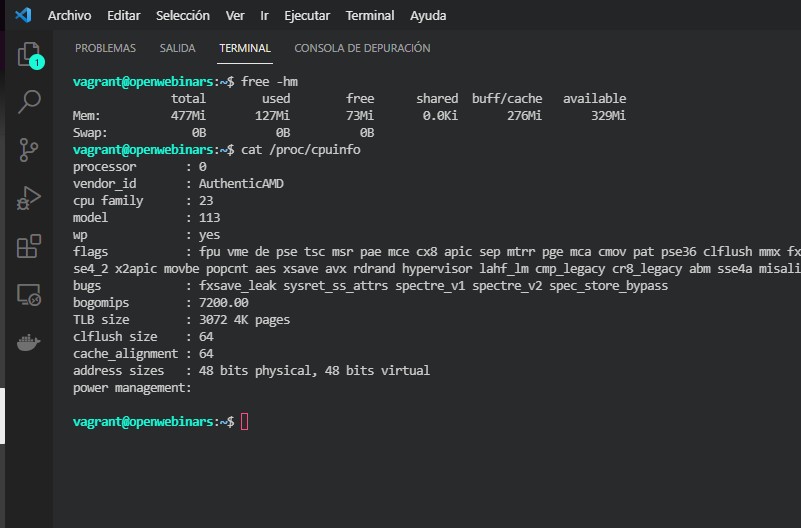




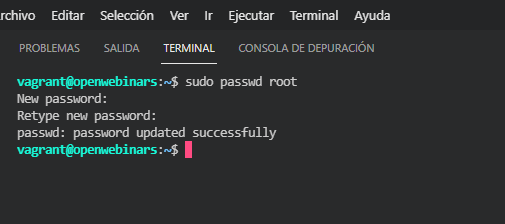
# MODIFICACION HARDWARE MAQUINA VIRTUAL:

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

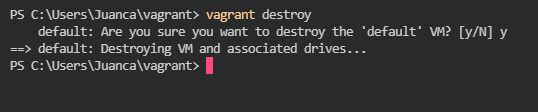


1. Verifico las características de la maquina:
2. Abriendo virtualbox
3. Entrando en la máquina virtual (vagrant ssh)
   1. Free -hm
   2. Cat /proc/cpuinfo
4. Activamos el usuario Root:

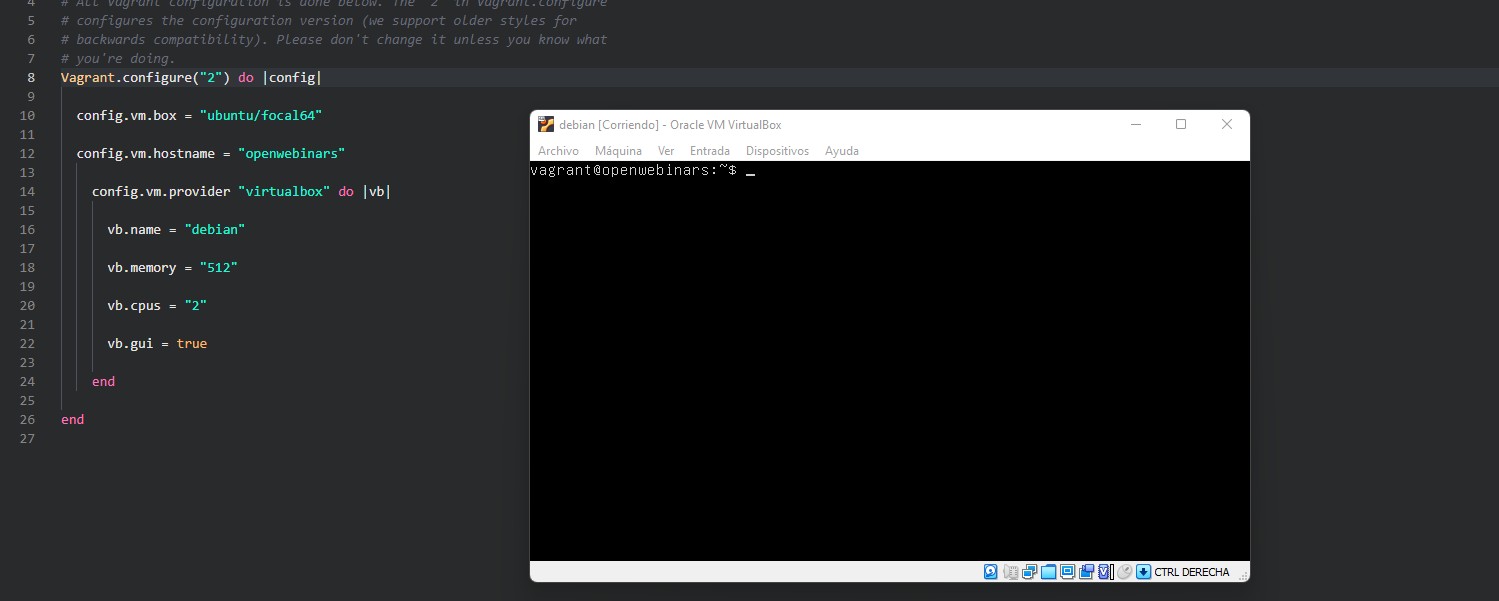
1. Sudo passwd root



1. Destruye la maquina (destroy)

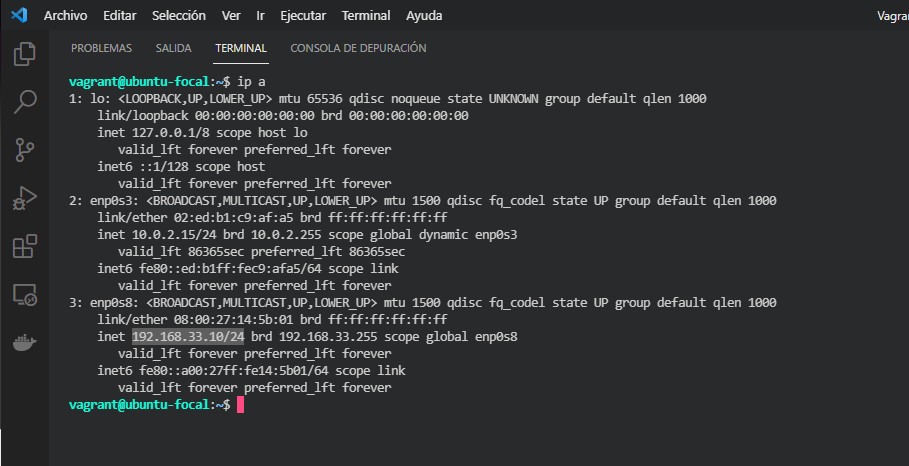
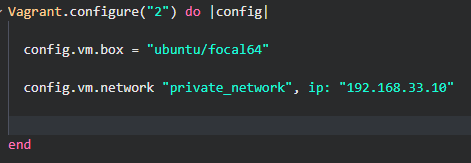


1. Créala de nuevo, pero con interfaz grafica

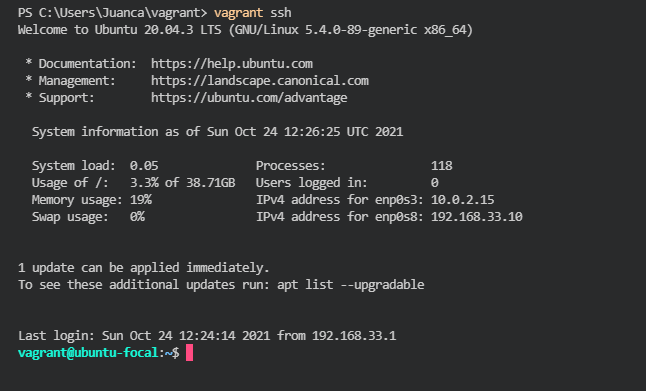


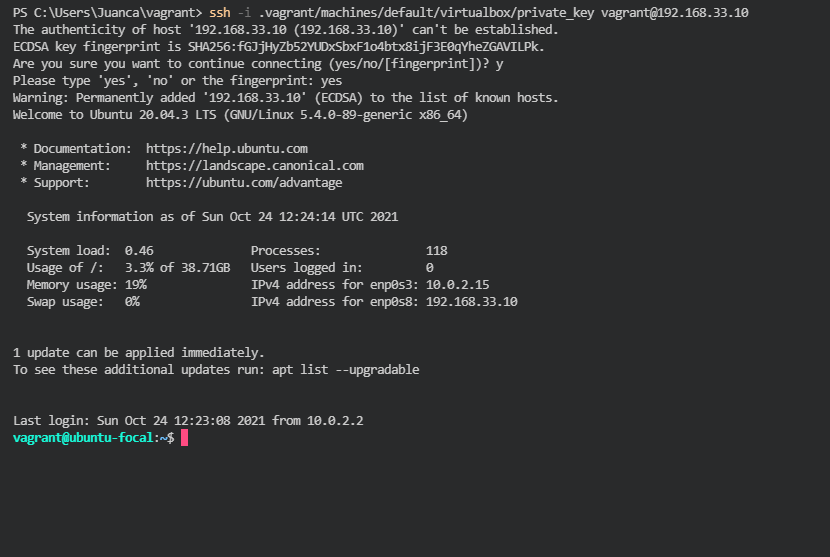
# Redes privadas:

1. Crea un nuevo Proyecto p3redes
   1. Ubuntu/trusty64
   2. Añade una red privada en 192.168.33.10



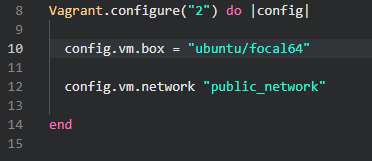
* 1. Levanta la maquina
  2. Accede mediante las dos interfaces



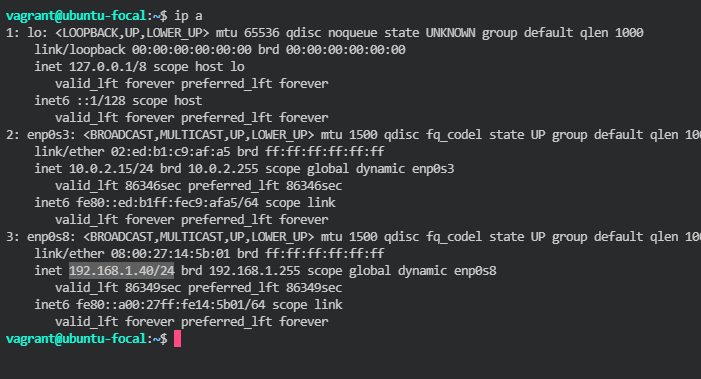


# 14. Redes públicas:

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

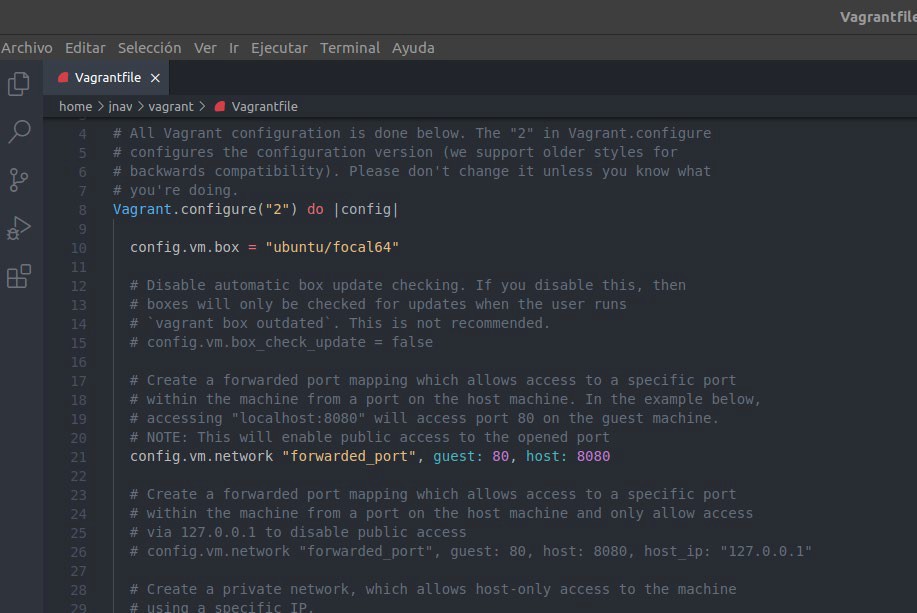


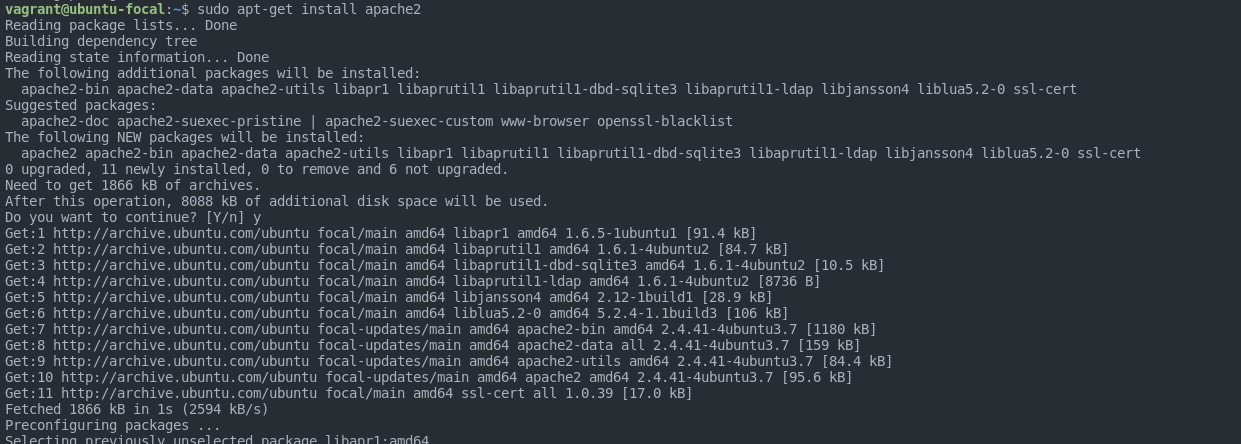
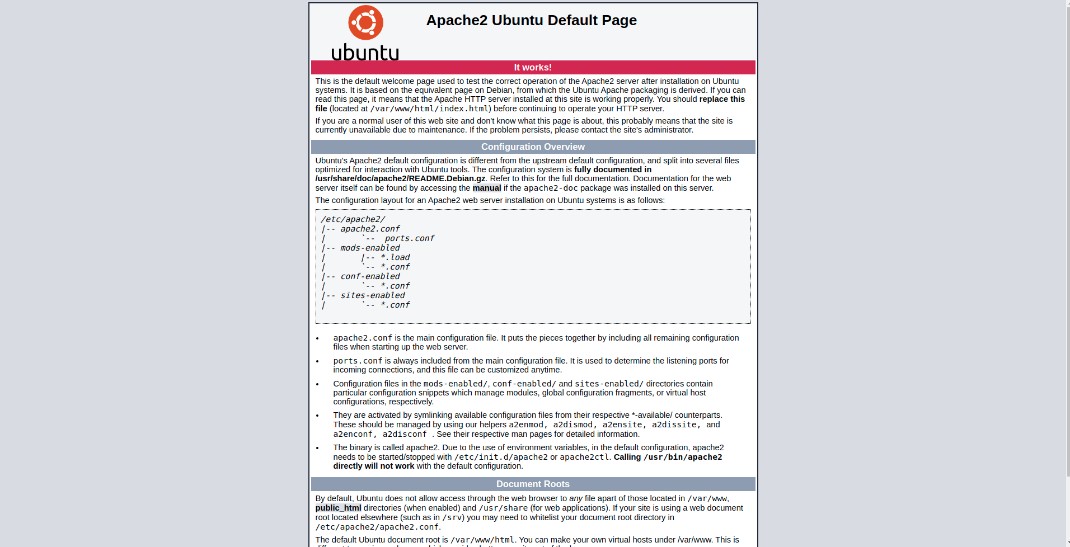
1. Levantamos la maquina vagrant up
2. Me conecto vagrant ssh
3. Veo el direccionamiento ip

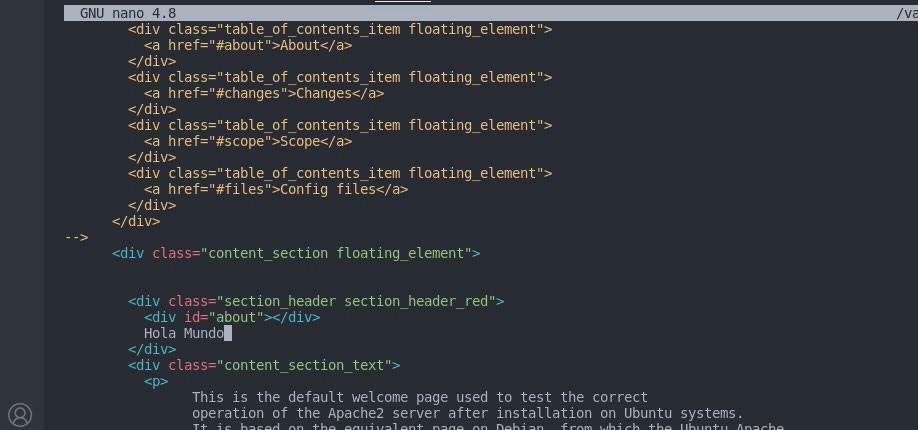


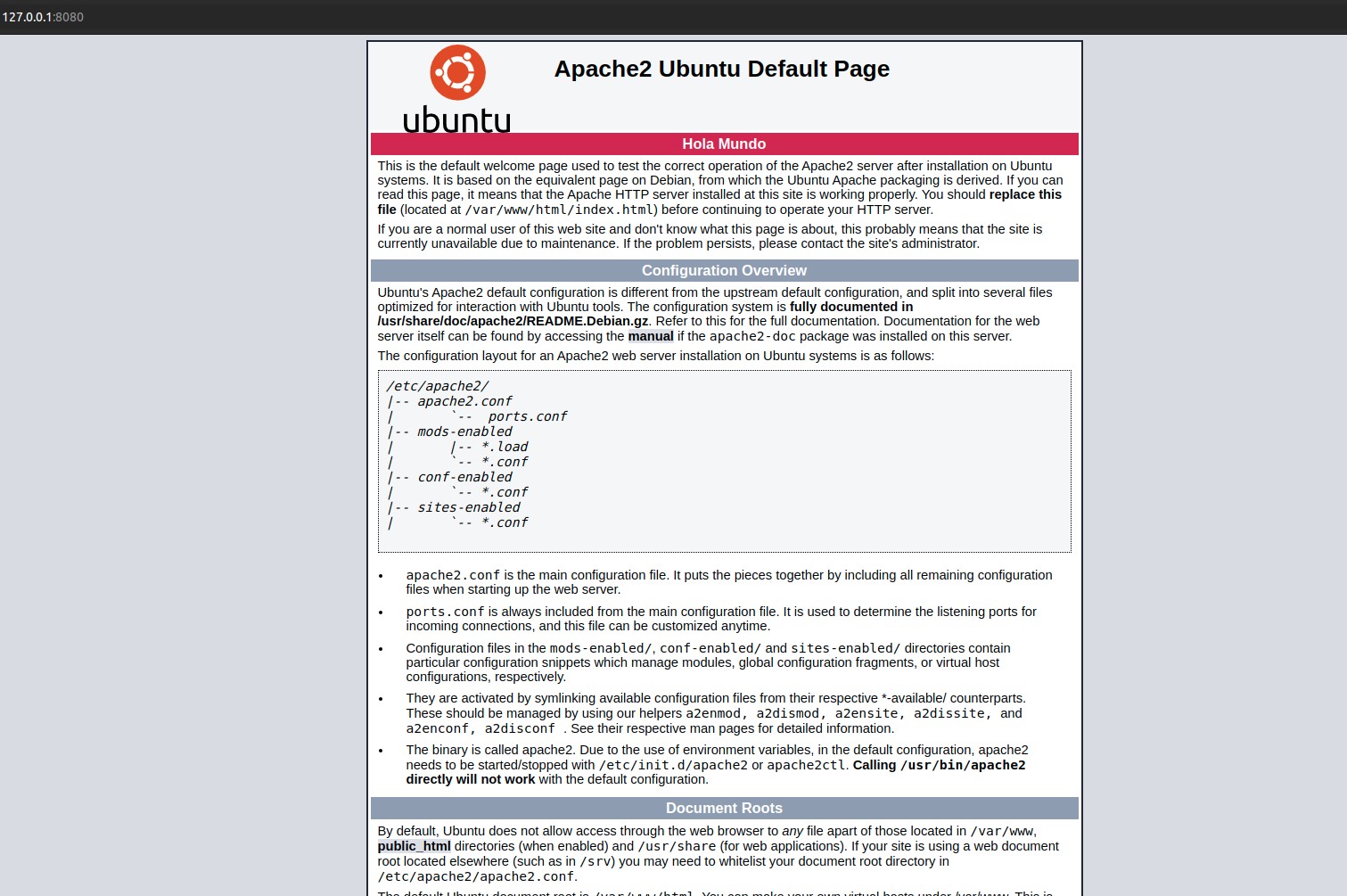
# Servidor Web:

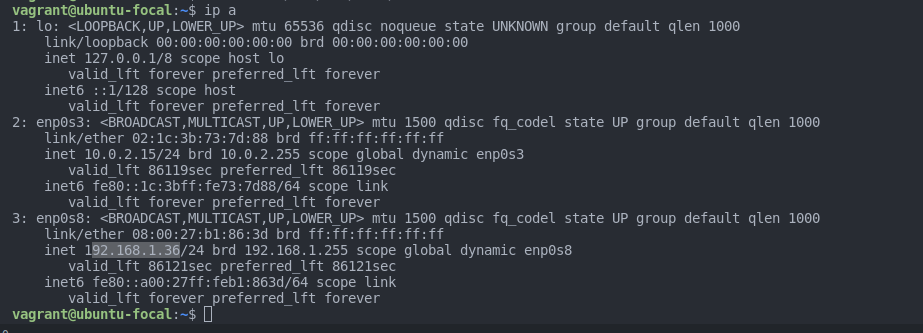
* 1. Modifica el fichero vagrant de la maquina debian para que se pueda acceder a un servidor web en la máquina virtual (guest) en el puerto 80 sabiendo que el puerto que se utiliza en la maquina anfitriona es el 8080

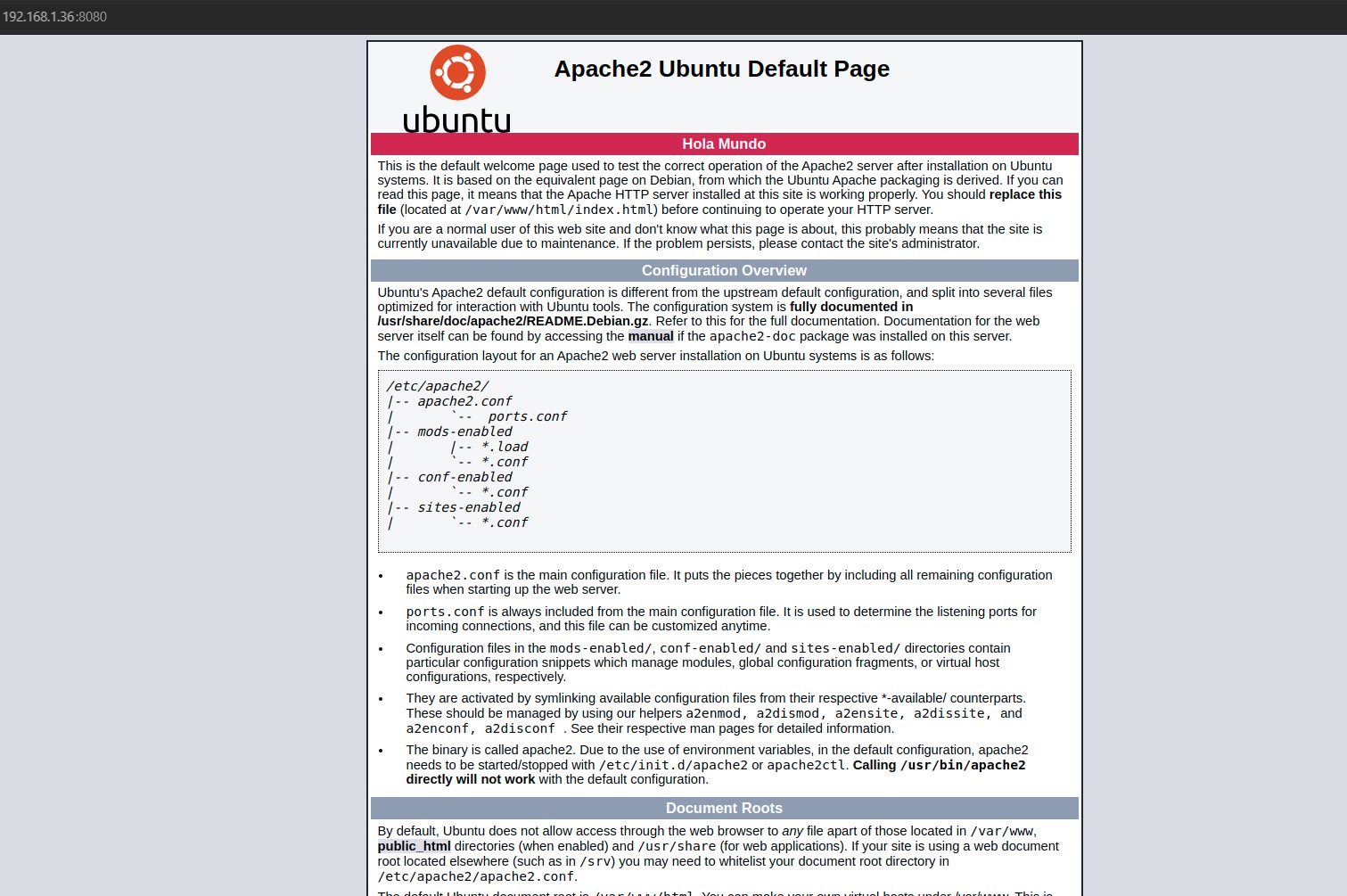


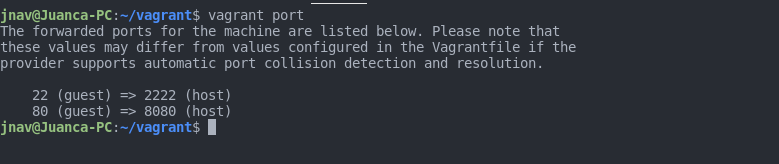
* 1. Levanta la maquina
  2. Vagrant up
  3. Vagrant ssh
  4. Actualizamos e Instalamos el servicio de apache
  5. Abrimos el navegador
  6. Accedemos al archivo index.html para cambiarlo y que aparezco hola mundo



* 1. Vuelvo a acceder al navegador para ver la modificación
  2. 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

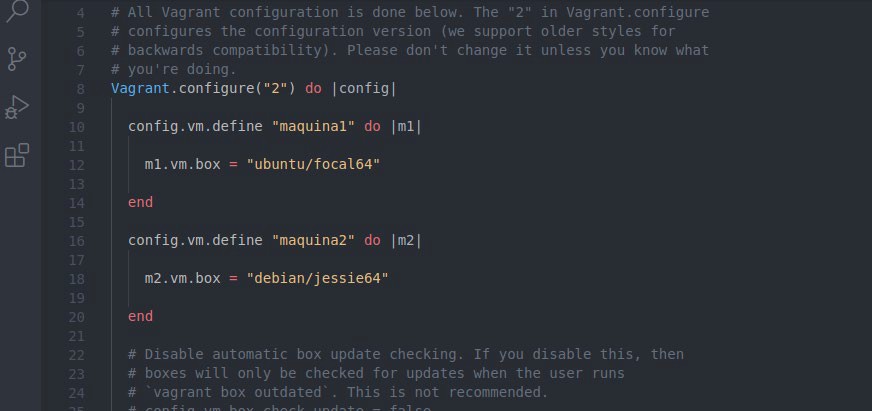


* 1. Accedo a internet desde esa ip
  2. Vagrant port

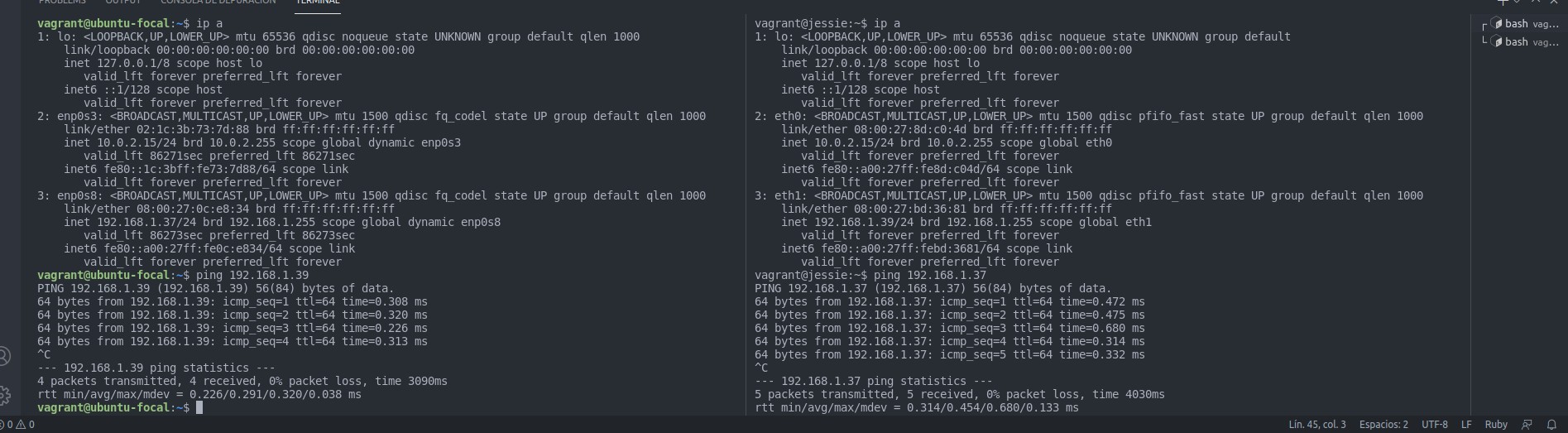


# Multientornos:

1. Configurar el Vagrantfile:

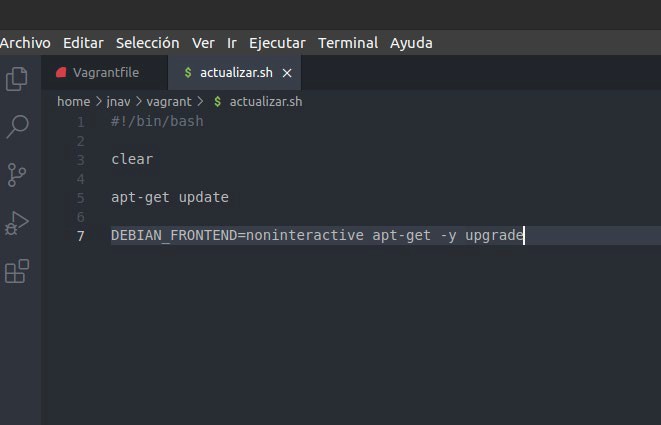


1. Vagrant reload
2. Vagrant up
3. Inicia las máquinas en terminales diferentes:
4. Haz un ping entre ellas:

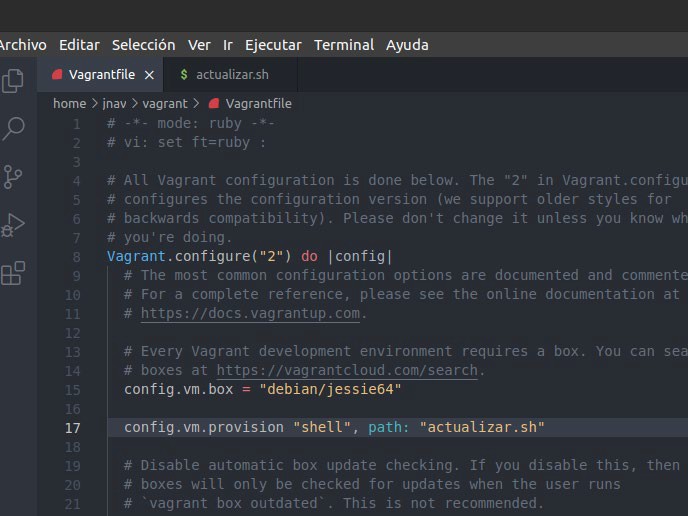


# 17. Actualización automática con script:

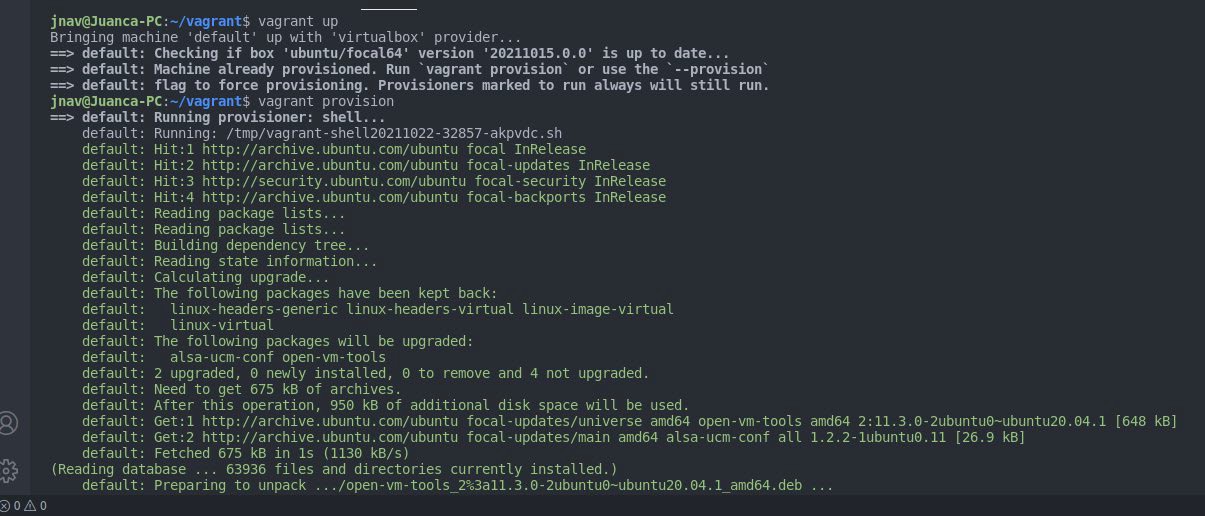
1. Creamos un script .sh llamado actualizar.sh:



1. Actualizamos el Vagrantfile:

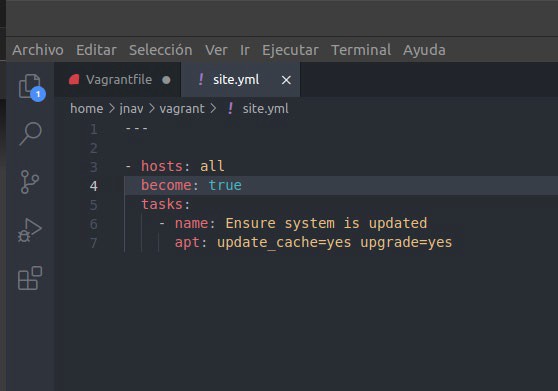


1. Levantamos la máquina
2. Hacemos un vagrant provision:

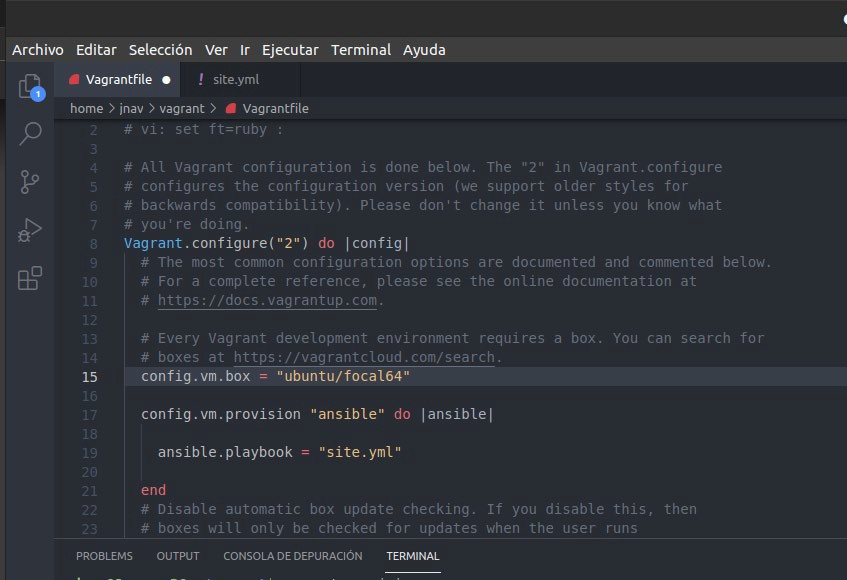


# 18. Actualización automática con Ansible:

1. Creamos un script en Yamal:



1. Actualizamos el Vagrantfile:



1. Levantamos la máquina
2. Hacemos un vagrant provision:

