

Fuente: Bing Chat AI



BLOQUE 2. VAGRANT



MÁSTER EN INGENIERÍA WEB 2023 – 2024

Administración de Sistemas Operativos v1.2 ("L-62 Princesa de Asturias")

© José Manuel Redondo López. Universidad de Oviedo



Contenido

Nivel 1 (Cabo): Instalación y uso básico	6
¿Qué necesito para hacer este bloque de actividades?	6
Tarea de administración: Establecer una IP fija con <i>Netplan</i>	6
Tarea de administración: Hosts con múltiples IPs	7
🔧 N1VAG_INSTALAR: Instalación de Vagrant (0,1 puntos)	9
👤 Descripción de la actividad	9
🏆 Resultados esperados	9
📖 Información necesaria para su realización	9
🔧 N1VAG_ARRANCA_BOX: Arrancar boxes en Vagrant (0,1 puntos)	9
👤 Descripción de la actividad	9
🏆 Resultados esperados	9
📖 Información necesaria para su realización	9
🔧 N1VAG_INTERACCION: Interacción básica con Vagrant Boxes (0,1 puntos)	10
👤 Descripción de la actividad	10
🏆 Resultados esperados	10
📖 Información necesaria para su realización	10
🔧 N1VAG_BASE_VFILES: Uso básico de Vagrantfiles (0,2 puntos)	10
👤 Descripción de la actividad	10
🏆 Resultados esperados	11
📖 Información necesaria para su realización	11
Nivel 2 (Sargento): Resolución de problemas sencillos frecuentes	12
🔧 N2VAG_CONFPROV: Creación de configuraciones por proveedor de virtualización (0,1 puntos)	12
👤 Descripción de la actividad	12
🏆 Resultados esperados	12
📖 Información necesaria para su realización	12
🔧 N2VAG_VARS: Creación de variables en una Vagrantfile (0,1 puntos)	13
👤 Descripción de la actividad	13
🏆 Resultados esperados	13
📖 Información necesaria para su realización	13
🔧 N2VAG_BASICPARAM: Configuración de parámetros básicos de una MV (0,1 puntos)	13
👤 Descripción de la actividad	13
🏆 Resultados esperados	13



📁	Información necesaria para su realización	14
🔧	N2VAG_TAMHD: Configuración de tamaño del disco duro (0,1 puntos)	14
👤	Descripción de la actividad	14
🏆	Resultados esperados	14
📁	Información necesaria para su realización	14
🔧	N2VAG_AUTOWEB: Instalación automatizada de un servidor web (0,2 puntos)	14
👤	Descripción de la actividad	14
🏆	Resultados esperados	15
📁	Información necesaria para su realización	15
🔧	N2VAG_ALPINWEB: Instalación automatizada de un servidor web en una imagen de Alpine Linux (0,1 puntos)	15
👤	Descripción de la actividad	15
🏆	Resultados esperados	15
📁	Información necesaria para su realización	15
🔧	N2VAG_AUTOBD: Instalación automatizada de un servidor de bases de datos (0,2 puntos)	16
👤	Descripción de la actividad	16
🏆	Resultados esperados	16
📁	Información necesaria para su realización	16
🔧	N2VAG_CUSTOMBOX: Personalizar manualmente una box (0,2 puntos)	16
👤	Descripción de la actividad	16
🏆	Resultados esperados	16
📁	Información necesaria para su realización	16
	Nivel 3 (Teniente de navío): Resolución de situaciones más avanzadas	18
🔧	N3VAG_PROV: Uso y creación de provisioners más avanzados (0,3 puntos)	18
👤	Descripción de la actividad	18
🏆	Resultados esperados	18
📁	Información necesaria para su realización	18
🔧	N3VAG_VIRTEXT: Instalación de utilidades de virtualización (0,2 puntos)	19
👤	Descripción de la actividad	19
🏆	Resultados esperados	19
📁	Información necesaria para su realización	19
🔧	N3VAG_OPTPROV: Optimización de proveedores de virtualización (0,2 puntos)	20
👤	Descripción de la actividad	20
🏆	Resultados esperados	20



📁	Información necesaria para su realización	20
🔧	N3VAG_W10: Instalación de una máquina virtual optimizada partir de una box de Windows de Stefan Scherer (0,2 puntos)	22
👤	Descripción de la actividad	22
🏆	Resultados esperados	22
📁	Información necesaria para su realización	22
🔧	N3VAG_VAGGUI: Instalación de un GUI (0,2 puntos)	22
👤	Descripción de la actividad	22
🏆	Resultados esperados	23
📁	Información necesaria para su realización	23
🔧	N3VAG_MULTIPLEPROV: Provisionando con varios proveedores de virtualización (0,3 puntos)	23
👤	Descripción de la actividad	23
🏆	Resultados esperados	23
📁	Información necesaria para su realización	23
🔧	N3VAG_FRONTBACK: Creación de una infraestructura web (front+back) (0,3 puntos)	24
👤	Descripción de la actividad	24
🏆	Resultados esperados	24
📁	Información necesaria para su realización	24
Nivel 4 (Capitán de navío): Redes, volúmenes y despliegue de plataformas de desarrollo		25
🔧	N4VAG_SHAREDF: Carpetas compartidas con el host (0,2 puntos)	25
👤	Descripción de la actividad	25
🏆	Resultados esperados	25
📁	Información necesaria para su realización	25
🔧	N4VAG_DISKVOL: Creación de volúmenes de disco (0,2 puntos)	25
👤	Descripción de la actividad	25
🏆	Resultados esperados	26
📁	Información necesaria para su realización	26
🔧	N4VAG_REDES: Manejo de redes en Vagrant (0,3 puntos)	26
👤	Descripción de la actividad	26
🏆	Resultados esperados	26
📁	Información necesaria para su realización	26
🔧	N4VAG_PACKERWIN: Despliegue automatizado de Windows con Packer (0,2 puntos)	27
👤	Descripción de la actividad	27



🏆 Resultados esperados	27
📖 Información necesaria para su realización	27
🔧 N4VAG_PLATMIW: Instalación de plataformas especializadas de otras asignaturas (0,5 puntos)	28
👤 Descripción de la actividad	28
🏆 Resultados esperados	28
📖 Información necesaria para su realización	28

Nivel 5 (Almirante): Optimizaciones avanzadas 29

🔧 N5VAG_MINHD: Minimizar el tamaño de disco duro las <i>boxes</i> distribuidas (0,2 puntos).....	29
👤 Descripción de la actividad	29
🏆 Resultados esperados	29
📖 Información necesaria para su realización	29
🔧 N5VAG_LINKEDCLON: Uso de <i>linked clones</i> y de máquinas sin GUI. Ventajas/desventajas (0,3 puntos)	29
👤 Descripción de la actividad	29
🏆 Resultados esperados	30
📖 Información necesaria para su realización	30
🔧 N5VAG_MULTIHYPERVERSOR: Conversión de máquinas entre distintas plataformas de virtualización (0,2 puntos)	31
👤 Descripción de la actividad	31
🏆 Resultados esperados	31
📖 Información necesaria para su realización	31
🔧 N5VAG_MINMV: Creación de una máquina virtual mínima (0,4 puntos).....	31
👤 Descripción de la actividad	31
🏆 Resultados esperados	31
📖 Información necesaria para su realización	32
🔧 N5VAG_CUSTOMBOX: Preparación de una <i>base box</i> personalizada (0,3 puntos).....	32
👤 Descripción de la actividad	32
🏆 Resultados esperados	32
📖 Información necesaria para su realización	32

Insignias y Autoevaluación 33



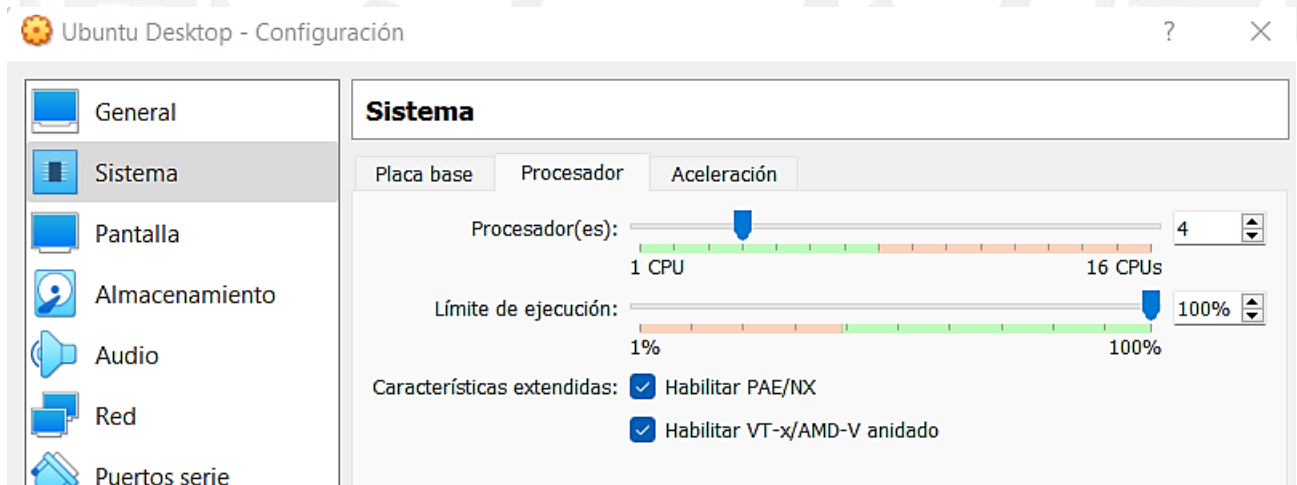
Nivel 1 (Cabo): Instalación y uso básico

¿Qué necesito para hacer este bloque de actividades?

Para realizar este bloque de actividades necesitas una máquina física compatible con *Vagrant*. Dado que este programa es compatible con *Windows*, *Linux* (varias distribuciones) y *MacOS*, no debería haber ningún problema al respecto. Las máquinas de los laboratorios de la escuela ya tienen *Vagrant* instalado.

Si no deseas instalar ningún software en ninguno de tus PCs, **este bloque también puede hacerse dentro de una máquina virtual Linux o Windows**, si bien habrá una pérdida lógica de rendimiento que puede ser inasumible si la máquina real o virtual tiene recursos limitados, ya que se tratará de un escenario de **virtualización anidada**. Si eliges este camino, necesitas:

- **Un software de virtualización instalado:** en principio sirve cualquiera que tenga la opción que se describirá en el último punto. Se recomienda *VirtualBox*.
- **Núcleos de CPU:** Mínimo 2, mejor 4
- **RAM:** Mínimo 8, mejor 12 o 16
- **SO:** Un *Linux* da mejor resultado que *Windows*, se ha probado sobre *Ubuntu 18.04* con éxito.
- **Activar el soporte de virtualización hardware (VT-X/AMD-V) anidado** tal y como establezca tu software de virtualización. En *VirtualBox* se trata de esta opción. **OJO:** No todas las CPUs soportan esta característica, por lo que, si te sale desactivada, **no podrás usar esta segunda opción**.



Se recomienda la opción de máquina física, si bien la segunda es perfectamente viable si tienes recursos suficientes.

Tarea de administración: Establecer una IP fija con *Netplan*

Esta tarea solo se menciona por si os hiciera falta para algo. *Ubuntu 17+* tiene una nueva forma de gestionar las redes mediante el programa **netplan** (en lugar del clásico **ifupdown**). *Netplan* simplifica mucho las configuraciones de red complejas y, aunque **ifupdown** sigue soportándose, hay que instalarlo



expresamente. Para modificar la configuración de **netplan**, debemos editar el fichero **/etc/netplan/50-cloud-init.yaml** (o uno similar, según el origen de la máquina virtual que estéis usando) y tener en cuenta estos aspectos. Podemos fijar una dirección IP en la red que necesitamos de la siguiente forma: <https://www.ostechnix.com/how-to-configure-ip-address-in-ubuntu-18-04-lts/>. Para ello debemos tener en cuenta varias cosas

- Es importante **NO** usar tabuladores en el fichero **.yaml**
- No hay que olvidarse de hacer **sudo netplan apply**
- Y finalmente comprobar que la dirección se ha cambiado con **ifconfig**

```
GNU nano 2.9.3 /etc/netplan/50-cloud-init.yaml

# This file is generated from information provided by
# the datasource.  Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses: [192.168.1.200/24]
      dhcp4: no
  version: 2
```

```
operario@server1804:~$ sudo netplan apply
operario@server1804:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:feea:bc6d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:ea:bc:6d txqueuelen 1000 (Ethernet)
    RX packets 4663 bytes 3806687 (3.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2201 bytes 144510 (144.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.200 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fe14:3a65 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:14:3a:65 txqueuelen 1000 (Ethernet)
    RX packets 208 bytes 21104 (21.1 KB)
    RX errors 0 dropped 129 overruns 0 frame 0
    TX packets 53 bytes 5808 (5.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Tarea de administración: Hosts con múltiples IPs

Esta tarea solo se menciona por si os hiciera falta para algo. Es posible asignar a una misma interfaz de red varias IPs siguiendo estas instrucciones y el procedimiento del apartado anterior: <https://www.serverlab.ca/tutorials/linux/administration-linux/how-to-configure-network-settings-in-ubuntu-18-04-bionic-beaver/>



GNU nano 2.9.3 /etc/netplan/00-installer-config.yaml

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.56.105/24, 192.168.56.106/24]_
  version: 2
```



1

2

3

4

5



N1VAG_INSTALAR: Instalación de *Vagrant* (0,1 puntos)

Descripción de la actividad

Esta actividad consiste en **hacer una instalación básica y funcional de Vagrant**

Resultados esperados

El comando **vagrant** está instalado en el *host* y es usable

Información necesaria para su realización

Hay que asegurarse de lo siguiente:

- **Instalarlo desde su página oficial** para tener la última versión de este. Seguir las instrucciones de la web oficial: <https://www.vagrantup.com/>
- **Asegurarse de tener instalado un proveedor de virtualización** adecuado instalado en la máquina que va a ejecutar *Vagrant*. Se recomienda el uso de *VirtualBox*: **sudo apt-get install virtualbox** (o descargarlo de su web oficial), al ser el proveedor para el que *Vagrant* se construyó originalmente.
- Si prefirieses usar otro proveedor compatible, no hay problema siempre que lo tengas en cuenta para el resto del boletín de ejercicios.

N1VAG_ARRANCA_BOX: Arrancar boxes en *Vagrant* (0,1 puntos)

Descripción de la actividad

Este ejercicio consiste en **instalar la primera máquina virtual operativa** con *Vagrant*

Resultados esperados

Has podido instalar una box de un sistema *Linux* de tu elección y entrar en sesión en la misma.

Información necesaria para su realización

Debes seguir un proceso en tres pasos:

- **vagrant init <nombre de la máquina>**: Esto crea un fichero *Vagrantfile* a nombre de la box que hayamos indicado. Recuerda que si el nombre de la *box* no existe en local (no lo hemos usado nunca), se busca en el repositorio central *Vagrant Cloud*. Como este será tú caso, por **favor selecciona una box base de algún proveedor oficial** del SO que elijas.
 - La selección del SO tiene bastante libertad. **Se recomienda instalar preferentemente una máquina virtual Ubuntu 18.04 LTS** con la siguiente configuración: 2 Gb de RAM y 80 Gb de HD. Si preferís otro sistema operativo *Linux* (*CentOS Stream*, *Debian*, etc.) también es posible, si tenéis la suficiente destreza con el mismo.



- **vagrant up**: Arranca la máquina especificada en la *Vagrantfile* del directorio actual, construyéndola primero
- **vagrant ssh**: Nos crea una terminal SSH con la máquina especificada en la *Vagrantfile* del directorio actual. Debe haberse arrancado previamente. Como alternativa, puede accederse a ella como una máquina “normal” desde la GUI del proveedor de virtualización elegido.

N1VAG_INTERACCION: Interacción básica con *Vagrant Boxes* (0,1 puntos)

Descripción de la actividad

Este ejercicio consiste en **probar operaciones de interacción básica** con una máquina *Vagrant*

Resultados esperados

Eres capaz de hacer lo siguiente:

- Puedes consultar el estado de una máquina virtual con *Vagrant*
- Entiendes las formas que existen de destruirla para que deje de ocupar espacio en su *host* cuando deja de ser necesaria
- Puedes contestar a las preguntas formuladas en el ejercicio.

Información necesaria para su realización

Debes probar las siguientes operaciones de interacción básica con una máquina *Vagrant*

- Se puede **consultar el estado** (*stopped*, *preparing*, *running*) de la máquina especificada por una *Vagrantfile* ejecutando **vagrant status** desde el mismo directorio donde está. Probarlo.
- **Conectarse, vía tu cliente SSH favorito**, al puerto **2222** del host (puerto por SSH por defecto si arrancamos una sola MV, que redirige al puerto 22 de la MV arrancada) y comprobar que podemos entrar en sesión.
 - La sesión SSH se puede finalizar con **CTRL+D**, aunque también podemos hacer **logout**
- Probar **vagrant destroy** para **eliminar la máquina construida** a partir de la *Vagrantfile*. ¿Qué pasa ahora si hago un **vagrant up**? ¿Para qué crees que sirve esto?
- Probar a **eliminar la imagen base** a partir de la cual la construimos con **vagrant box remove**. ¿Funciona? ¿Qué diferencia hay con **vagrant destroy**? ¿Para qué crees que sirve esto?

N1VAG_BASE_VFILES: Uso básico de *Vagrantfiles* (0,2 puntos)

Descripción de la actividad

Esta tarea consiste en volver a crear la máquina anterior, pero esta vez **con un provisioner**, que haga que pueda servir como base para futuros trabajos.



Resultados esperados

Tienes un fichero *Vagrantfile* que instala automáticamente una máquina virtual *Linux* y la provisiona para trabajar desde ella con consola y, además:

- El provisionamiento incluye actualizarla e instalar una serie de herramientas útiles para distintos usos
- Debe además reiniciarla al final

Información necesaria para su realización

El *provisioner* (preferiblemente de *shell* en un fichero externo en lugar de en línea) debe hacer las siguientes operaciones que dejen lista a la MV para poder trabajar con ella en otros escenarios. Añade también comentarios (con `#`) que documenten lo que vas haciendo:

- Actualizar todos los paquetes de la máquina virtual.
- Instalar un conjunto básico de herramientas de trabajo en línea de comandos: `apt-transport-https ca-certificates gnupg-agent software-properties-common unzip curl wget tmux preload gpm mc virt-what nano grep`. Estos paquetes son para *Linux* tipo *Debian*; si usas otro tipo de *Linux*, tendrás que buscar los equivalentes.
- Cambiar el teclado a Español. En sistemas *Ubuntu* puede hacerse así:

```
loadkeys es
sudo DEBIAN_FRONTEND=noninteractive dpkg-reconfigure console-setup
sudo sed -i 's/XKBLAYOUT="\w*"/XKBLAYOUT="es"/g' /etc/default/keyboard
```

- Añadir un usuario cuyo nombre sea tu **UO** a la máquina virtual adaptando, por ejemplo, este *script* (reemplaza `$1` por el nombre de usuario que quieras, y `"test123..."` por la *password* que desees):

```
#!/bin/bash
echo "Adding user '$1'"
sudo useradd -m -s /bin/bash -p $(openssl passwd -1 test123...) $1
sudo adduser $1 sudo
```

- Hacer que el *provisioner* al final reinicie la máquina para cargar correctamente el nuevo *software* instalado



Nivel 2 (Sargento): Resolución de problemas sencillos frecuentes

N2VAG_CONFPROV: Creación de configuraciones por proveedor de virtualización (0,1 puntos)

Descripción de la actividad

Crear una *Vagrantfile* con **secciones separadas para distintos proveedores de virtualización**.

Resultados esperados

Tienes un fichero *Vagrantfile* que permite asignar configuraciones independientes a una misma box en función del proveedor de virtualización y valida correctamente. Puedes contestar a estas preguntas:

- ¿Entiendes por qué esto es necesario?
- ¿Crees que todos los proveedores de virtualización usan los mismos nombres de atributos en sus objetos de configuración?

Información necesaria para su realización

Como hemos dicho en la teoría, es posible servir una misma *box* desde varios proveedores de virtualización, **siempre y cuando dicha box los soporte**. Esto implica que la *Vagrantfile* deberá tener **una sección diferente para cada proveedor admitido**, de forma que pueda usar sus atributos para configurar los aspectos de la máquina que desee. Vamos a hacer una prueba con nuestra máquina anterior para dos proveedores, *VirtualBox* y *Hyper-V*. Para ello, incorpora a la *Vagrantfile* anterior estas dos secciones de forma que **se asigne la misma cantidad de RAM a la máquina construida a través de ambos**, conservando el resto de las operaciones que se realizan en la misma. Usa este trozo de código como guía:

```
config.vm.provider "virtualbox" do |vb|
  vb.memory = 2
end
config.vm.provider "hyperv" do |hv|
  hv.memory = 2
end
```



N2VAG_VARS: Creación de variables en una *Vagrantfile* (0,1 puntos)

Descripción de la actividad

Crear una serie de **variables** en una *Vagrantfile*.

Resultados esperados

Tienes un fichero *Vagrantfile* que permite definir variables con valores y valida correctamente, aunque estas no se usen todavía

Información necesaria para su realización

El “**hardcodeo**” de valores es una mala idea, especialmente cuando varios proveedores tienen la misma configuración y los cambios nos obligan a ir viendo cuántas veces aparece el mismo valor en el código del fichero. Es mucho mejor tener los valores guardados en variables y usarlas desde donde haga falta, de manera que las modificaciones de valores consistan simplemente modificar estas variables.

Las variables en una *Vagrantfile* se crean en el contexto global del fichero, no dentro de un `config.vm.provider`. La sintaxis es sencilla y consiste en un nombre (normalmente en mayúscula) una asignación y un valor, que habitualmente puede ser un `string` (entre “”) o un número. Ejemplos:

```
BOX_NAME = "hashicorp/bionic64"
RAM = 2048
```

Sabiendo esto, añade a la *Vagrantfile* anterior variables que guarden los siguientes valores

- El nombre de la *box* de la *Vagrant Cloud* que se va a usar
- El nombre de la máquina a crear
- Una descripción de la máquina a crear
- Tamaño de la RAM
- Nº de cores
- Tamaño del disco (normalmente es un *string* con un nº y la unidad de medida, por ejemplo GB)

N2VAG_BASICPARAM: Configuración de parámetros básicos de una MV (0,1 puntos)

Descripción de la actividad

Configurar parámetros básicos de una MV según su proveedor de virtualización

Resultados esperados

Esta actividad se considera terminada cuando hayas conseguido parametrizar con variables determinadas características de las máquinas virtuales a crear



Información necesaria para su realización

Esta actividad te pide usar la documentación oficial de los proveedores de virtualización para asignar a los atributos de los objetos de configuración de cada proveedor que controlen las características de la máquina (RAM y Cores, **HD todavía no**) y otras características (Ej.: nombre de la máquina) con las variables correspondientes definidas anteriormente. Para ello, usa la siguiente documentación:

- Atributos del objeto de *VirtualBox*: <https://www.vagrantup.com/docs/virtualbox/>
- Atributos del objeto de *Hyper-V*: <https://www.vagrantup.com/docs/providers/hyperv>

N2VAG_TAMHD: Configuración de tamaño del disco duro (0,1 puntos)

Descripción de la actividad

Redimensionar el disco duro de una MV, para hacer que el espacio de este sea distinto que el que trae por defecto

Resultados esperados

Esta actividad se considera terminada cuando hayas conseguido redimensionar el disco duro de la máquina virtual creada, estableciendo el tamaño con una variable definida anteriormente

Información necesaria para su realización

En la actividad anterior hemos parametrizado una serie de valores de las máquinas virtuales, **pero hemos omitido a propósito el disco duro**. El motivo de ello es que redimensionar el disco de las MVs creadas es aún una **característica experimental**. El tamaño de disco inicial de la box creada depende del que tenga asignado la box de base que usemos. Si queremos cambiarlo, debemos hacer uso de esta opción global experimental (<https://www.vagrantup.com/docs/experimental>).

- Las **características experimentales** de *Vagrant* se pueden activar con variables de entorno. En este caso es necesario añadir al principio de la *Vagrantfile* `ENV[ 'VAGRANT_EXPERIMENTAL' ] = 'disks'`
- Hecho esto, **dentro del entorno de configuración global** (`vagrant.configure("2") do |config|`, no es necesario hacerlo por proveedor de virtualización) podemos hacer por ejemplo esto para redimensionar el disco duro a 120Gb: `config.vm.disk :disk, size: 120Gb, primary: true`

N2VAG_AUTOWEB: Instalación automatizada de un servidor web

(0,2 puntos)

Descripción de la actividad

Consiste en usar *Vagrant* para **configurar automáticamente una MV que sirva como servidor web básico**.



Resultados esperados

Esta actividad se considera terminada cuando hayas conseguido que una máquina virtual *Vagrant* pueda desplegar automáticamente un servidor web básico. Además, **debe ser accesible desde el exterior** usando un puerto del *host*.

Información necesaria para su realización

En esta actividad lo primero que debes hacer es copiar la *Vagrantfile* anterior a una nueva carpeta que crees y cambiar el nombre de la máquina virtual a crear en el fichero para que sea una nueva máquina. En esta nueva máquina, **debes instalar un servidor web de tu elección** (*Apache2*, *Nginx*...) modificando el *provisioner* existente para instalar todo lo necesario. En este caso basta con que el servidor web sirva su *web* por defecto. El servidor web debe atender peticiones cuando un navegador se conecte a un puerto concreto del *host* (se sugiere usar un puerto alto que no esté previamente ocupado).

N2VAG_ALPINEWEB: Instalación automatizada de un servidor web en una imagen de *Alpine Linux* (0,1 puntos)

Descripción de la actividad

Consiste en usar *Vagrant* para **configurar automáticamente una MV que sirva como servidor web básico**, pero **usando un *Alpine Linux*** como SO base.

Resultados esperados

Esta actividad se considera terminada cuando hayas conseguido que una máquina virtual *Vagrant* pueda desplegar automáticamente un servidor web básico que sea accesible desde el exterior usando un puerto del *host*, pero sobre un *Alpine Linux*. Puedes contestar a esta pregunta: *¿Qué diferencia de tamaño hay entre la versión con *Alpine* y la versión con *Ubuntu*?*

Información necesaria para su realización

En esta actividad lo primero que debes hacer es copiar la *Vagrantfile* anterior a una nueva carpeta que crees y cambiar el nombre de la máquina virtual a crear en el fichero para que sea una nueva máquina. En esta máquina, debes instalar un servidor web de tu elección (*Apache2*, *Nginx*...) **como en el caso anterior**, pero **usando una imagen de *Alpine Linux*** como base (<https://app.vagrantup.com/generic/boxes/alpine38>), en lugar de la *Ubuntu* anterior. Una vez construido el servidor, deben compararse los tamaños de ambas imágenes para ver la diferencia.



N2VAG_AUTOBD: Instalación automatizada de un servidor de bases de datos (0,2 puntos)

Descripción de la actividad

Consiste en **desplegar automáticamente un SGBD** gracias a *Vagrant*.

Resultados esperados

Esta actividad se considera terminada cuando hayas conseguido que una máquina virtual *Vagrant* pueda desplegar automáticamente un servidor de BBDD **que sea accesible desde el exterior** usando un puerto del host

Información necesaria para su realización

En esta actividad lo primero que debes hacer es **copiar la *Vagrantfile* anterior a una nueva carpeta** que crees y **cambiar el nombre de la máquina virtual** a crear en el fichero para que sea una nueva máquina. En esta nueva máquina, debes instalar un servidor de bases de datos **de tu elección** (*MariaDB, MySQL...*) modificando el *provisioner* existente para instalar todo lo necesario. Consulta como se hace una instalación desde el sistema de paquetes de tu SO.

En este caso basta con que el servidor tenga una **configuración por defecto**. El servidor debe atender peticiones cuando un cliente se conecte a un puerto concreto del *host* (se sugiere usar un **puerto alto** que no esté previamente ocupado).

N2VAG_CUSTOMBOX: Personalizar manualmente una box (0,2 puntos)

Descripción de la actividad

Consiste en usar una *box* arrancada existente para **crear una box nueva a partir de ella** que se pueda usar posteriormente.

Resultados esperados

Este ejercicio se considerará terminado cuando consigamos exportar una máquina virtual que tengamos a una *box* y reimportar dicha *box* desde otra *Vagrantfile* para formar una máquina virtual nueva.

Información necesaria para su realización

Una forma sencilla de crear una nueva *box* que sirva de base para futuras máquinas es hacerlo a partir de una preconstruída. **Implementar esto es el objetivo de esta tarea**, y se puede hacer con los siguientes pasos:

- **Arrancar** alguna máquina de las que ya tenemos



- **Entrar en sesión** en ella e instalar programas / ficheros / etc. que queramos como lo haríamos habitualmente (**apt** o **snap** si usamos la *Ubuntu* de referencia). Además, a la hora de crear una *box* de base nueva, tenemos que estar seguros de que **tiene todo lo necesario** (no sólo son importantes programas, sino usuarios, ficheros, servicios arrancados, puertos abiertos...).
- También es importante instalar las **extensiones de virtualización** del proveedor para el que esté pensado la máquina. Para *VirtualBox* se explica aquí: <https://www.vagrantup.com/docs/virtualbox/boxes.html>
- Una vez hayamos terminado, **salimos de la sesión SSH** con **exit** y desde el mismo directorio donde está la *Vagrantfile* hacemos **vagrant package**. Aquí están las posibles opciones de empaquetado que podamos necesitar: <https://www.vagrantup.com/docs/cli/package.html>
- La máquina exportada se escribirá en **package.box** (podemos renombrarla luego). Este fichero ahora es utilizable como base **especificándolo en nuevos Vagrantfile** que creamos. No hace falta hacerlo en este ejercicio.

```
operario@ubuntuserver:~/vbox$ vagrant package
==> default: Attempting graceful shutdown of VM...
==> default: Clearing any previously set forwarded ports...
==> default: Exporting VM...
==> default: Compressing package to: /home/operario/vbox/package.box
operario@ubuntuserver:~/vbox$ ls
package.box  Vagrantfile
operario@ubuntuserver:~/vbox$ ls -la package.box
-rw-rw-r-- 1 operario operario 288100406 sep 12 15:52 package.box
operario@ubuntuserver:~/vbox$
```



Nivel 3 (Teniente de navío): Resolución de situaciones más avanzadas

N3VAG_PROV: Uso y creación de *provisioners* más avanzados (0,3 puntos)

Descripción de la actividad

Consiste en refactorizar un *provisioners* en **varios ficheros más pequeños** y especializados.

Resultados esperados

Este ejercicio se considerará terminado cuando consigas refactorizar tu *provisioner* en varios ficheros independientes con una función concreta, y **servir la web que quieras** copiándola dentro de la máquina servidora anterior

Información necesaria para su realización

Cuando las máquinas virtuales se provisionan, es posible que necesitemos instalar o configurar un gran nº de cosas, así como incorporar ficheros que se encuentren en el *host* dentro de las máquinas creadas. Para tener un mayor control y orden en los ficheros de provisionamiento, es necesario seguir **buenas prácticas de programación** y usar nuevas funcionalidades. Eso es *Infrastructure as Code*, y por tanto seguir las buenas prácticas de programación tiene todo el sentido 😊. Para ello esta actividad te plantea mejorar tu *provisioner* anterior de *shell* con las siguientes técnicas:

- **Refactorizar el *provisioner* en varios ficheros** de manera que cada uno haga una cosa concreta. Por ejemplo, se puede crear un fichero que haga sólo la instalación del *software* base, otro que actualice la máquina, etc. **El fichero al que se llama desde la *Vagrantfile* contendrá solo llamadas para ejecutar estos ficheros refactorizados.** También puedes lanzarlos como *provisioners* independientes dentro de la *Vagrantfile* (tendrías por tanto **varios `config.vm.provision`**), pero debes recordar **hacer que sólo el último reinicie la máquina creada**).
- **Usar el *provisioner* `file`** para copiar ficheros dentro de la máquina virtual. Por ejemplo, puedes copiar la web dada como ejemplo dentro de la carpeta `/var/www/html` de la máquina, que es donde los servidores web leen normalmente el contenido a servir. Para ello tienes que usar el *provisioner* `file`, con la siguiente sintaxis: **`config.vm.provision "file", source: "<ruta del host>", destination: "<ruta de la máquina virtual>"`**



N3VAG_VIRTEXT: Instalación de utilidades de virtualización (0,2 puntos)

Descripción de la actividad

Consiste en instalar las **extensiones de virtualización de tu proveedor de virtualización** para poder mejorar el rendimiento y la experiencia de uso de tus MVs.

Resultados esperados

Este ejercicio se considerará terminado cuando consigas instalar y configurar correctamente las herramientas de virtualización del proveedor de virtualización que estés usando, de manera que puedan usarse tras un reinicio de la máquina

Información necesaria para su realización

Una de las mejoras de rendimiento más grandes que puede experimentar una máquina virtual es la **instalación de una serie de utilidades concretas de cara proveedor de virtualización**. Además de mejor rendimiento, al permitir acceder a *hardware* emulado más avanzado, también permite activar **utilidades de virtualización avanzadas**, como el portapapeles y las carpetas compartidas.

En esta actividad se trata de añadir un nuevo *script* a tu *provisioner* refactorizado que te añada dichas utilidades en función del proveedor de virtualización que estés usando. Para ello, puedes usar como guía los siguientes extractos:

- Si tu proveedor de virtualización es **VirtualBox**, debes incorporar esto

```
sudo apt-get -y --no-install-recommends install -y virtualbox-guest-dkms virtualbox-guest-
utils virtualbox-guest-x11
sudo VBoxClient --clipboard
sudo VBoxClient --draganddrop
sudo VBoxClient --vmsvga
sudo VBoxClient --checkhostversion
sudo VBoxClient --seamless
sudo adduser vagrant vboxsf
```

También debes añadir cualquier usuario que pueda usar las carpetas compartidas al grupo **vboxsf** con `sudo adduser <id de usuario> vboxsf`

- Si es **Hyper-V**, se incorporaría esto:

```
sudo sed -i '$a hv_vmbus' /etc/initramfs-tools/modules
sudo sed -i '$a hv_storvsc' /etc/initramfs-tools/modules
sudo sed -i '$a hv_blkvsc' /etc/initramfs-tools/modules
sudo sed -i '$a hv_netvsc' /etc/initramfs-tools/modules
# Replace default kernel with linux-virtual for the best LIS experience
sudo apt-get -y install linux-virtual linux-cloud-tools-virtual linux-tools-virtual
# Update Initramfs
sudo update-initramfs -u
```



N3VAG_OPTPROV: Optimización de proveedores de virtualización

(0,2 puntos)

Descripción de la actividad

Consiste en aprender a **exprimir las opciones más avanzadas que mejoran la virtualización** según el proveedor de esta que uses.

Resultados esperados

Este ejercicio se considerará terminado cuando consigas poner en marcha todas las optimizaciones disponibles del proveedor de virtualización para máquinas virtuales

Información necesaria para su realización

Otra de las cosas que afecta significativamente a los proveedores de virtualización **es el uso de sus características avanzadas para exprimir al máximo el rendimiento de su máquinas virtuales**. Si bien eso requiere un conocimiento profundo del proveedor, en este boletín os enumeramos la configuración concreta que optimiza los proveedores de virtualización que estamos usando hasta el momento.

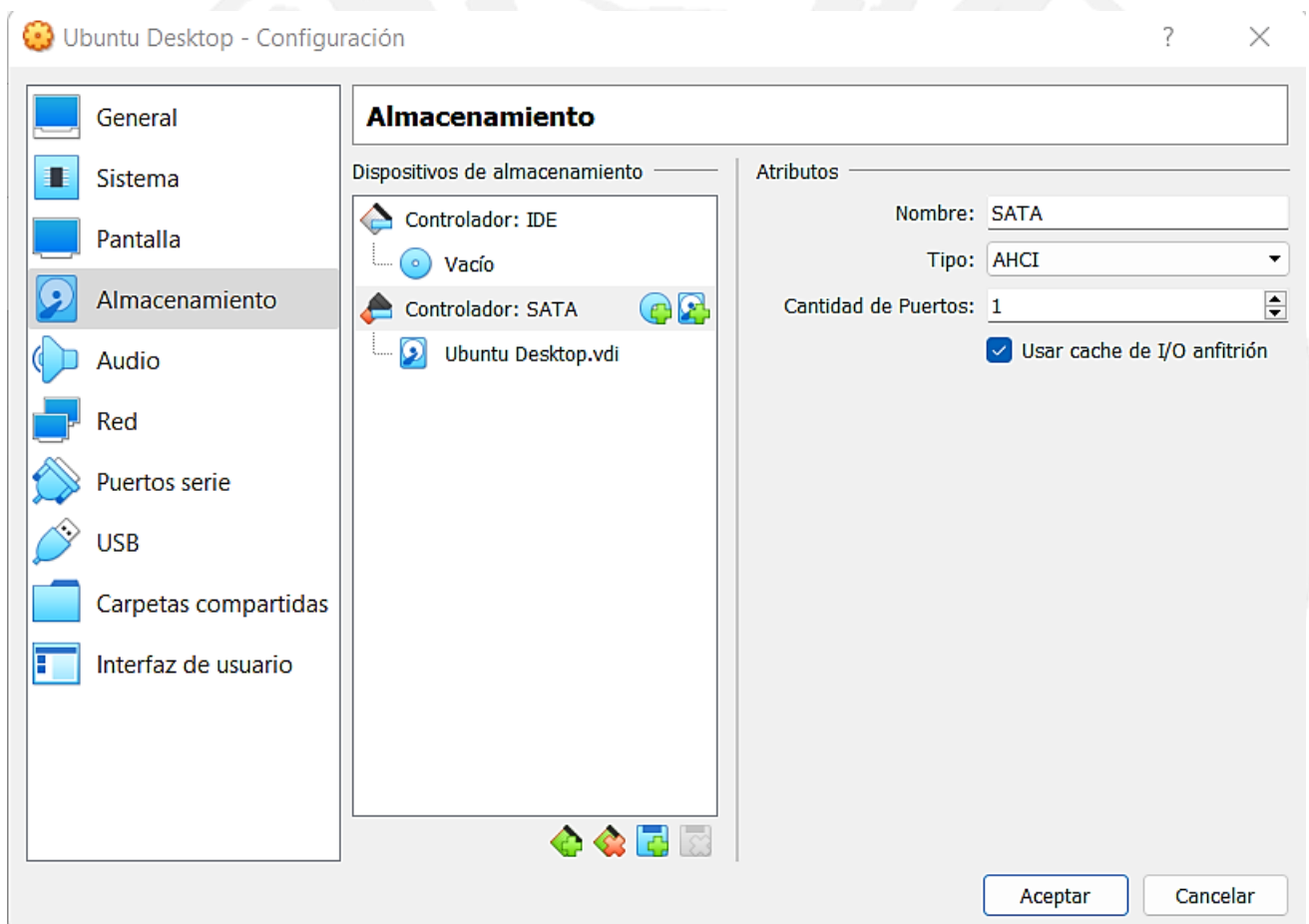
Si usas VirtualBox, esta es la combinación de opciones que da un mejor rendimiento que debes incorporar a tu *Vagrantfile*. Todas ellas usan la funcionalidad `customize` del objeto del proveedor de *VirtualBox*, que en este ejemplo llamamos `vb`. Algunas de las opciones presentadas **usan variables que debes crear previamente**, mirando sus valores admitidos en la documentación oficial del proveedor. Estas opciones son las que controlan el parámetro `modifyvm` del comando `VBoxManage`, que es el comando al que llama esta opción del proveedor en la *Vagrantfile*: <https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm>

```
vb.customize ["modifyvm", :id, "--acpi", "on"]
vb.customize ["modifyvm", :id, "--accelerate3d", VIDEO_3D]
vb.customize ["modifyvm", :id, "--apic", "on"]
vb.customize ["modifyvm", :id, "--biosapic", "x2apic"]
vb.customize ["modifyvm", :id, "--cpu-profile", "host"]
vb.customize ["modifyvm", :id, "--graphicscontroller", "vmsvga"]
vb.customize ["modifyvm", :id, "--hpet", "on"]
vb.customize ["modifyvm", :id, "--hwvirtex", "on"]
vb.customize ["modifyvm", :id, "--largepages", "on"]
vb.customize ["modifyvm", :id, "--longmode", "on"]
vb.customize ["modifyvm", :id, "--nestedpaging", "on"]
vb.customize ["modifyvm", :id, "--pae", "on"]
vb.customize ["modifyvm", :id, "--pagefusion", "on"]
vb.customize ["modifyvm", :id, "--rtcuseutc", "on"]
vb.customize ["modifyvm", :id, "--spec-ctrl", "off"]
vb.customize ["modifyvm", :id, "--x2apic", "on"]
vb.customize ["modifyvm", :id, "--vram", VRAM]
vb.customize ["modifyvm", :id, "--vtxux", "on"]
vb.customize ["modifyvm", :id, "--vtxvpid", "on"]
vb.customize ["modifyvm", :id, "--vrde", VRDE]
vb.customize ["storagectl", :id, "--name", HD_CONTROLLER, "--hostiocache", "on"]
```



En el caso del **HD_CONTROLLER**, se refiere al **nombre del controlador de disco** que tiene la máquina que se va a crear. Esto depende de cómo se haya creado la *box* base, y no podemos cambiarlo. En muchas ocasiones estaremos usando *box* prefabricadas, por lo que a priori no podremos saberlo directamente. En caso de la *Ubuntu* que estamos manejando suele ser “*SATA Controller*”, pero también puede ser “*IDE Controller*”, “*IDE*” o “*SCSI*”.

Si falla el arranque de la máquina y el error dice que **no se encuentra el manejador de disco**, para solucionarlo debemos **ir al interfaz de nuestro proveedor de virtualización** y consultar el tipo de controlador de disco que se ha creado en esta pantalla, para cambiarlo por el valor correcto en la *Vagrantfile*. Sí, aunque la máquina no haya arrancado se habrá creado una entrada para ella en nuestro proveedor de virtualización igualmente, y nos aprovechamos de eso para mirar el valor que nos haga falta



NOTA: La opción “**--pagefusion**” seguramente **no estará disponible si tu host es un MAC**, por lo que debes comentarla o borrarla.

Por otra parte, **si usas un proveedor Hyper-V** las opciones disponibles son muchas menos, y se pueden resumir en estas:

```
hv.enable_enhanced_session_mode = true
hv.vm_integration_services = {
  guest_service_interface: true,
  heartbeat: true,
  key_value_pair_exchange: true,
  shutdown: true,
  time_synchronization: true,
```



```
vss: true,  
}
```

N3VAG_W10: Instalación de una máquina virtual optimizada partir de una box de Windows de Stefan Scherer (0,2 puntos)

Descripción de la actividad

Consiste en **poner en marcha una MV Windows 10** automáticamente con *Vagrant*.

Resultados esperados

Este ejercicio se considerará terminado cuando tengas operativa una máquina virtual *Windows 10* a partir de las boxes proporcionadas por Stefan Scherer, con todas las optimizaciones a la virtualización que hemos visto.

Información necesaria para su realización

Esta actividad consiste en usar todas las opciones de optimización vistas en la actividad anterior para construir una máquina virtual *Windows 10* a partir de las imágenes ofrecidas por Stefan Scherer en la *Vagrant Cloud*: https://app.vagrantup.com/StefanScherer/boxes/windows_10. Para ello, hay que tener en cuenta los siguientes datos para construir la *Vagrantfile* concreta

- No se va a requerir usar *provisioners*. En caso de hacerlo tendríamos que usar *PowerShell*, y esto queda fuera del alcance de esta asignatura, ya que el uso de *PowerShell* es casi un curso en sí mismo 😊 (<https://learn.microsoft.com/es-es/powershell/>). El proveedor sería **el mismo que hemos usado hasta ahora (shell)**, y un ejemplo de cómo hacerlo sería: <https://gist.github.com/MattHodge/4df62673b575e546b9ff>.
- El nombre de la box base es **StefanScherer/windows_10**
- El tipo de sistema operativo que usa *VirtualBox* es **"Microsoft Windows"**
- El tipo de controlador de disco es **"IDE Controller"**
- Todas las opciones de optimización funcionan **SALVO ESTA**: **vb.customize ["modifyvm", :id, "--x2apic", "on"]**. Debe por tanto eliminarse de la *Vagrantfile*
- NOTA:** Si vuestra máquina virtual se colgase al poco de arrancar, quita las opciones de optimización de *VirtualBox* del ejercicio anterior. Esto ha ocurrido en unas pocas pruebas y no se ha podido averiguar la causa exacta, ya que parece que depende del hardware o la versión del proveedor o de *Vagrant* usada...

N3VAG_VAGGUI: Instalación de un GUI (0,2 puntos)

Descripción de la actividad

Consiste en **instalar un GUI** en una máquina virtual automáticamente si se dan ciertas condiciones.



Resultados esperados

Este ejercicio se considerará terminado cuando tengas control total de si se instala o no un GUI en una máquina virtual mediante una variable dentro de la *Vagrantfile*

Información necesaria para su realización

Instalar un GUI es una decisión que dependerá del propósito de la máquina virtual a instalar. Para un uso como servidor no tiene mucho sentido (salvo en entornos de aprendizaje), mientras que para un uso de usuario estándar sí. Es una decisión que depende del contexto y que debemos tomar de antemano.

En esta actividad **se trata de usar las capacidades de las *Vagrantfiles* de usar estructuras del lenguaje Ruby para definir una variable que controle si no se instala GUI**, se instala uno ligero (XFCE4) o uno completo (Gnome), usando la estructura **if else** del lenguaje *Ruby*. Para ello se pueden seguir las documentaciones de instalación de ambos GUI oficiales, introduciendo su instalación en un nuevo fichero que se pueda llamar desde el *provisioner*. Estos ejemplos son para *Ubuntu*:

- XFCE4: `sudo apt install xfce4`
- Gnome: `sudo apt-get install gnome-shell ubuntu-gnome-desktop`

N3VAG_MULTIPLEPROV: Provisionando con varios proveedores de virtualización (0,3 puntos)

Descripción de la actividad

Consiste en hacer **un *provisioner* que haga cosas diferentes en función del proveedor de virtualización** que uses.

Resultados esperados

Esta actividad se considerará completa cuando el *provisioner* que instala las extensiones de virtualización sea capaz de detectar el proveedor de virtualización concreto, y en función de ello decidir qué instala. Puedes contestar a las siguientes preguntas:

- ¿Entiendes ahora la importancia de poder discriminar entre proveedores de virtualización a la hora de decidir qué software instalar?
- ¿Para qué crees que serviría algo así en producción? (piensa en que los proveedores de virtualización tienen distinta configuración hardware por defecto en las MV)

Información necesaria para su realización

En una actividad anterior vimos cómo instalar las extensiones de virtualización para el proveedor de virtualización instalado, pero esto tiene el problema de que, si la *Vagrantfile* se mueve a otro *host*, entonces puede ser que **el proveedor en la nueva máquina no sea el mismo que en la anterior**, teniendo por tanto que editar la *Vagrantfile* para reemplazar estas líneas.



Una forma más elegante de hacer esto es intentar **que el *provisioner* examine el tipo de proveedor de virtualización** con el que se está trabajando y tome las decisiones oportunas. Para ello una técnica habitual es usar el paquete **virt-what**, disponible en la mayoría de las distribuciones *Linux* y fácilmente usable desde un *provisioner* de *shell*.

Esta actividad plantea usar dicho paquete para que el *provisioner* anterior, que instala las extensiones de virtualización, sea capaz de identificar el proveedor usado en la máquina virtual e instalar unas u otras en función de eso, haciendo la *Vagrantfile* y sus *provisioners* **portables**. La misma técnica se puede usar con todo aquello que dependa del proveedor de virtualización que sea necesario. Para ello se recomienda usar esta estructura, **instalando previamente el paquete *virt-what*** en la MV:

```
if [[ $(sudo virt-what | grep 'virtualbox') = *virtualbox* ]]; then
  <Código que instala las VirtualBox Guest Additions que vimos antes>
elif [[ $(sudo virt-what | grep 'hyperv') = *hyperv* ]]; then
  <Código que instala las mejoras de Hyper-V que vimos antes>
fi
```

N3VAG_FRONTBACK: Creación de una infraestructura web (*front+back*) (0,3 puntos)

Descripción de la actividad

Consiste en **instalar dos máquinas distintas** desde una misma *Vagrantfile*, emulando una estructura clásica de un servidor web.

Resultados esperados

Esta actividad se completará cuando tengas un despliegue completo de la web de ejemplo en dos máquinas creadas con *Vagrant* en un entorno multi-máquina

Información necesaria para su realización

Esta actividad quiere que uses las capacidades de **hacer una *Vagrantfile* multi-máquina** que vimos en teoría para que, usando los conocimientos que has adquirido hasta ahora, distribuyas la web de ejemplo suministrada entre dos máquinas, una que tenga **sólo el *front-end*** con *Apache* o *Nginx* y la otra que tenga el ***back-end* y el servidor de base de datos *mariadb***.

(**NOTA:** La web de ejemplo no usa bases de datos, solo se trata de hacer el despliegue de ambas máquinas y no de una aplicación que use BBDD necesariamente, salvo que tengas una a mano que lo haga).

Usar solo máquinas de la *Vagrant Cloud* oficiales para construir este entorno (¡no te instales un *malware* por intentar ahorrarte unos pasos! 😞) y arranca ambas máquinas a la vez. Recuerda la forma de **crear redes privadas entre máquinas** que se explicó en teoría para comunicar ambas. Ej.: `main.vm.network "private_network", ip: "<IP estática>", adapter: <nº de adaptador de la MV>, virtualbox____intnet: "<nombre de la red interna, que debe ser común a ambas máquinas para que puedan comunicarse>"`



Nivel 4 (Capitán de navío): Redes, volúmenes y despliegue de plataformas de desarrollo

N4VAG_SHAREDF: Carpetas compartidas con el host (0,2 puntos)

Descripción de la actividad

Consiste en poder **crear carpetas compartidas con el host** para una máquina virtual y así poder compartir fácilmente contenido entre el host y distintas MVs.

Resultados esperados

Esta actividad se completará cuando puedas servir una página web desde una carpeta del host en una máquina virtual. Por otro lado, también es necesario que **desactives la ruta compartida por defecto**. Si la desactivamos, ¿Recuerdas de módulos anteriores alguna otra forma de conseguir copiar ficheros del host a la máquina que vas a usar?

Información necesaria para su realización

Hemos visto que **por defecto Vagrant comparte la carpeta del host** donde está la *Vagrantfile* en una ruta **/vagrant** que se crea en la máquina virtual. No obstante, esto no sirve en muchos casos, y es mejor **personalizar esta ruta**. Para ello, hay que usar esta directiva en la sección general de la *Vagrantfile*: **config.vm.synced_folder "<ruta en el host>", "<ruta de la máquina virtual>"**. Ten en cuenta que una ruta relativa en el *host* tiene su raíz en la carpeta donde está la *Vagrantfile*. **Usa esta técnica para servir una página web que esté en alguna carpeta del host desde una MV Vagrant que tenga un servidor web.**

Por otro lado, el uso de carpetas compartidas activa el uso del protocolo **SMB**. Usar este **protocolo es un riesgo de seguridad** en función de dónde esté situada la máquina virtual, por lo que muchas veces conviene desactivar las carpetas compartidas y no usar esta funcionalidad en absoluto. Esto se puede hacer con la carpeta por defecto así: **config.vm.synced_folder ".", "/vagrant", disabled: true**

N4VAG_DISKVOL: Creación de volúmenes de disco (0,2 puntos)

Descripción de la actividad

Consiste en **añadir nuevos discos duros** a las MVs creadas con *Vagrant*.



Resultados esperados

Esta actividad se completará cuando puedas crear nuevos discos de los dos tipos vistos en la teoría en una de las máquinas virtuales.

Información necesaria para su realización

En la teoría vimos que a una máquina virtual **se le pueden añadir nuevos discos de distintos tipos**. En esta actividad se deben usar dichos conocimientos para **añadir a la máquina** un nuevo disco duro de 20Gb para hacer copias de seguridad y una **.iso** cualquiera de una programa que descarguéis de internet, para **simular** la instalación de este (no hace falta que lo hagáis) en la máquina virtual.

Se puede usar la opción por defecto de añadirlo al controlador de disco actual sin necesidad de especificar uno, por ejemplo: `config.vm.disk :disk, name: "backup", size: "10GB"`. Recuerda que **es obligatorio** la incluir en la *Vagrantfile*: `ENV[ 'VAGRANT_EXPERIMENTAL' ] = 'disks'`

N4VAG_REDES: Manejo de redes en Vagrant (0,3 puntos)

Descripción de la actividad

Consiste en usar de forma efectiva **todos los tipos de redes** admitidos por *Vagrant* habitualmente.

Resultados esperados

Esta actividad se completará cuando puedas crear satisfactoriamente ejemplos operativos de las tres operaciones de redes enumeradas (redes privadas, redes públicas y mapeo de puertos)

Información necesaria para su realización

En la teoría de la asignatura hemos visto los distintos tipos de redes existentes en *Vagrant*, y esta actividad pretende que se usen estos conocimientos para añadir a alguna de las máquinas virtuales anteriores lo siguiente:

- Una **red pública** que permita acceder a los servicios de una máquina virtual con una IP de la misma red que el *host*. Tu PC sería por tanto el cliente de esta MV.
- Una **red privada** que comunique dos de las máquinas creadas anteriormente entre ellas. Una de esas máquinas sería por tanto el cliente de la otra.
- Un **mapeo de puertos** que permita ofrecer un servicio existente en una de las máquinas virtuales por un **puerto** del *host* que se mapee contra el del servicio de la MV. Una máquina del laboratorio de prácticas de algún compañero/a podría ser el cliente de tu MV.



N4VAG_PACKERWIN: Despliegue automatizado de Windows con Packer (0,2 puntos)

Descripción de la actividad

Consiste en **desplegar una máquina Windows 10 con Packer** para Vagrant.

Resultados esperados

Puedes instalar automáticamente Windows 10 a partir de una ISO de una versión de esta, permitiendo así la **instalación desde cero** de una distribución oficial de Windows 10 de la que dispongáis, sin recurrir a box externas de terceros. Puedes responder a las siguientes preguntas:

- ¿Este proceso crees que es más rápido o lento que usar una box Windows 10 de la Vagrant Cloud?
- ¿Qué ventajas de seguridad tiene hacerlo así?

Información necesaria para su realización

A lo largo de este boletín de ejercicios, y en la teoría general de la asignatura, **hemos priorizado los sistemas Linux** para poder profundizar más en todos los conceptos que vamos a impartir sin diversificarnos demasiado en ambos sistemas operativos. De hacerlo, hubiéramos perdido mucho tiempo, y no hubiéramos conseguido ver los conceptos de la misma manera, contribuyendo a que la asignatura fuese menos clara y más compleja.

No obstante, **gracias al trabajo de la alumna María Flórez Miranda**, en esta actividad en el nivel 4 podréis crear un Windows 10 construido a partir de una ISO que os podáis descargar del repositorio de software de la Universidad.

*(NOTA: la dificultad de este ejercicio no reside en que sea especialmente difícil, puesto que el trabajo de María **hace el procedimiento muy sencillo**. El problema es que, en función de los recursos del PC del que dispongáis, es un trabajo que puede tardar mucho tiempo (es una instalación completa de Windows 10, aunque no la veas...). Por tanto, se recomienda hacerlo cuando estéis con otra cosa o lo dejéis ejecutar mientras estáis haciendo otras actividades fuera del ordenador)*

Para la realización de esta actividad necesitas:

- Descomprimir el fichero **vagrant-ansible-hardening-main.zip** descargándote este repositorio: <https://github.com/Yori1999/vagrant-ansible-hardening>
- Ir a la carpeta **/packer/windows** y copiar la ISO del sistema operativo Windows que quieras instalar. Puedes descargar una ISO desde esta dirección, recomendándose una Windows 10 Education: https://portal.azure.com/#view/Microsoft_Azure_Education/EducationMenuBlade/~/software
- Ahora debemos abrir el archivo **windows_10_es.json** de esa carpeta y cambiar el **path** de la ISO para que coincida con la que acabamos de añadir.
- Para la creación de la ISO necesitamos instalar **packer** siguiendo sus tutoriales oficiales: <https://learn.hashicorp.com/tutorials/packer/get-started-install-cli>
- Ejecutar **packer_build_windows_10_es_box.sh** o **packer_build_windows_10_es_box.bat** en función del sistema operativo de tu **host**. **Ten en cuenta que va a tardar mucho tiempo en crear un archivo .box.**



- Una vez se cree dicho archivo, ejecutar `vagrant_add_windows_10_es_box.sh` o `vagrant_add_windows_10_es_box.bat` en función del SO de tu *host*.

Una de las principales aportaciones del trabajo de María es **crear un fichero de respuestas por defecto** a la instalación de *Windows 10* que evita que la misma se pare, y así puedas hacerla automáticamente (recuerda que si se pregunta algo al usuario, el proceso de *Vagrant* se rompe). Si te resulta útil para el futuro, puedes examinar los ficheros de María para ver cómo se hacen estas operaciones, algo que te puede venir bien si despliegas una máquina personalizada en entornos de nube en el futuro.

N4VAG_PLATMIW: Instalación de plataformas especializadas de otras asignaturas (0,5 puntos)

Descripción de la actividad

Consiste en instalar **MVs que sean adecuadas para hacer el trabajo de otras asignaturas** del máster.

Resultados esperados

Esta actividad se completará cuando puedas crear satisfactoriamente las máquinas virtuales enumeradas que preparan automáticamente software que se usa en otras asignaturas del máster.

Información necesaria para su realización

En esta actividad se usarán todos los conocimientos anteriores para preparar máquinas virtuales **de la forma más optimizada posible**, tanto a nivel de proveedor de virtualización como del sistema operativo a instalar, de **la última versión estable** de estas herramientas. Estas máquinas virtuales corresponderán a entornos de desarrollo que se van a usar en otras asignaturas del máster, por lo que se recomienda que **se instalen con GUI y con las extensiones de virtualización** del proveedor de esta que estamos utilizando

- **Administración de Sistemas de Persistencia de Objetos:** *Oracle Enterprise Linux 7*
- **Arquitectura y Diseño de Sitios Web:** Un *Ubuntu 18* o superior con *Eclipse* y *Spring Tools for Eclipse*
- **Diseño y Programación de Interfaces de Usuario:** Un *Ubuntu 18* con *Visual Studio Code* y *NodeJS*
- **Gestores de Contenidos Web:** Un *Ubuntu 18* o superior con *XAMPP*
- **Programación Orientada a Objetos:** Un *Ubuntu 18* con *PyCharm*, *Python 3* y `virtualenv`
- **Sistemas de Persistencia de Objetos:** Un *Ubuntu 18* con *Hadoop*
- **Sistemas de Seguridad en la Web:** Una máquina *Kali Linux* (versión *Rolling Release*)



Nivel 5 (Almirante): Optimizaciones avanzadas

N5VAG_MINHD: Minimizar el tamaño de disco duro las *boxes* distribuidas (0,2 puntos)

Descripción de la actividad

Consiste en que sepas como **minimizar el tamaño de las MV** que empaquetes y distribuyas.

Resultados esperados

Esta actividad se completará cuando puedas exportar una máquina virtual a **.ova con el espacio libre inicializado a ceros**, y compruebes la diferencia entre hacer esta operación y no hacerla a la hora disminuir el tamaño del **.ova** exportado. Recuerda por tanto exportar la MV con la que vayas a probar inicialmente y guardar su **.ova** para poder así comparar fácilmente el resultado del antes y el después.

Información necesaria para su realización

Uno de los principales problemas a la hora de exportar máquinas virtualizadas **es el tamaño que ocupan**. Por mucho que se optimice el *software* instalado, **es muy fácil obtener ficheros de varios Gb**. Por ejemplo, en caso de que una máquina se exporte tras hacer mucho trabajo que implique borrar archivos, dado que esos archivos no se borran físicamente del disco duro, en la exportación siguen ocupando espacio.

Podemos en ese caso **disminuir el tamaño de la máquina exportada** (a veces drásticamente) simplemente **escribiendo ceros** en ese espacio libre. Para ello hay varios métodos, pero uno básico que suele no fallar es escribir un fichero lleno de ceros al disco duro (en el ejemplo a **/var/zeros**) con el comando **dd: dd if=/dev/zero of=/var/zeros bs=8126k**, para luego borrarlo justo después con **rm /var/zeros**. Hecho esto, la exportación de la máquina normal debería devolver un fichero **.ova** de tamaño significativamente inferior al que tendría si no hacemos esta operación.

N5VAG_LINKEDCLON: Uso de *linked clones* y de máquinas sin GUI. **Ventajas/desventajas** (0,3 puntos)

Descripción de la actividad

Consiste en entender **qué son los *linked clones* de Vagrant** y sus ventajas/desventajas.



🏆 Resultados esperados

Esta actividad se completará cuando puedas comparar una clonación normal con un *linked clone*. Puedes además responder a las siguientes preguntas:

- ¿Qué pasaría si se borrara la imagen base?
- ¿Qué pasaría si se combinase un *linked clone* con la opción *page fusion* vista en teoría?
- ¿Para qué crees que puede ser más adecuada una máquina arrancada sin GUI (*headless*)?

📄 Información necesaria para su realización

Como vimos en la teoría, el uso de *linked clones* en *VirtualBox* **puede disminuir drásticamente el tamaño en disco de cada máquina**. Esta actividad consiste en clonar una máquina normalmente (**full clone**) con *Vagrant* y compararla con una clonación usando **linked clone**, para comprobar la diferencia de tamaño entre ambas opciones. Además, debe comprobarse el comportamiento de arrancar una MV sin el GUI del software de virtualización (**headless**), y si crees que esto influye en el tiempo de respuesta. Puedes usar esta *Vagrantfile* como guía para crear infraestructuras compuestas de varios *linked clones* (entornos multi-máquina) ahora y en el futuro

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

VAGRANTFILE_API_VERSION = "2"

vagrant.configure("2") do |config|
  # Configuración general de las máquinas Vagrant.
  config.vm.box = "<Pon aquí la box que hayas usado en otros ejercicios"
  config.vm.synced_folder ".", "/vagrant", disabled: true
  config.vm.provider :virtualbox do |v|
    v.memory = 512
    v.linked_clone = true
  end

  # Servidor web 1.
  config.vm.define "web1" do |web|
    web.vm.hostname = "web1"
    web.vm.network :private_network, ip: "192.168.10.4"
  end

  # Servidor web 2.
  config.vm.define "web2" do |app|
    app.vm.hostname = "web2"
    app.vm.network :private_network, ip: "192.168.10.5"
  end

  # Servidor de base de datos.
  config.vm.define "db" do |db|
    db.vm.hostname = "db"
    db.vm.network :private_network, ip: "192.168.10.6"
  end
end
```



N5VAG_MULTIHYPERVERSOR: Conversión de máquinas entre distintas plataformas de virtualización (0,2 puntos)

Descripción de la actividad

Consiste en que aprendas a **migrar máquinas creadas de un proveedor de virtualización a otro**.

Resultados esperados

Esta actividad se completará cuando puedas convertir una máquina virtual creada con *Vagrant* para *VirtualBox* a una máquina funcional equivalente que funcione sobre *Hyper-V*

Información necesaria para su realización

Si bien *Vagrant* está preparado para trabajar principalmente con *VirtualBox*, y este es libre y gratuito, no en pocas ocasiones no podremos disponer de él porque ya existe previamente un proveedor de virtualización instalado y no podemos o no nos autorizan a instalar otro (políticas de empresa, por ejemplo).

Si la *box* base que usamos no tiene una versión para el proveedor de virtualización que está instalado, solo nos queda crear la máquina para *VirtualBox* y **convertir dicha máquina al proveedor que tengamos** con herramientas adaptadas para hacer esta operación. Un caso común ocurre con uno de los proveedores de virtualización que da problemas frecuentemente con *VirtualBox*: **Hyper-V** (*Microsoft, Windows*), dado el diferente carácter de ambos, como vimos en teoría. Se trata por tanto de usar alguna de las máquinas virtuales que ya has creado junto con la herramienta gratuita de *Starwind V2V Converter* (<https://www.starwindsoftware.com/starwind-v2v-converter>) y este procedimiento: <https://www.groovypost.com/howto/migrate-virtual-box-vms-windows-10-hyper-v/> para intentar conseguir un equivalente a una máquina que has creado que pueda ser usada sobre *Hyper-V*.

N5VAG_MINMV: Creación de una máquina virtual mínima (0,4 puntos)

Descripción de la actividad

Consiste en crear una **máquina Ubuntu con el mínimo de software y funcionalidad** para ser operativa, pero ocupando la menor cantidad de recursos posibles y por tanto disminuyendo su superficie de ataque.

Resultados esperados

Esta actividad se completará cuando puedas crear una máquina virtual *Ubuntu* con *Vagrant* para *VirtualBox* cuyo espacio ocupado al exportarla sea mucho menor que el de una instalación estándar. Puedes contestar a esta pregunta: *Si usar una distribución de Linux concreta no es realmente un requisito, ¿qué usarías para lograr un SO que ocupe lo mínimo?*



Información necesaria para su realización

En esta actividad se plantea **usar las instrucciones dadas en teoría para minimizar el espacio ocupado por una máquina virtual Ubuntu** cuando se exporte con *VirtualBox* a un fichero **.ova**, comparando el tamaño obtenido por la máquina original (sirve cualquiera que se haya usado hasta ahora) con la minimizada, obtenida de aplicar todas las opciones de reducción de espacio ocupado mencionadas en un *provisioner* de *Vagrant*.

N5VAG_CUSTOMBOX: Preparación de una base box personalizada

(0,3 puntos)

Descripción de la actividad

Consiste en aprender a convertir una MV que tengas en una que **se pueda usar como box base de Vagrant**.

Resultados esperados

Esta actividad se completará cuando puedas crear tu propia *base box* funcional con *Vagrant*, que funcione de la misma forma que cualquiera descargada de la *Vagrant Cloud*

Información necesaria para su realización

En la teoría vimos cómo era posible personalizar cualquier máquina virtual que tengamos para convertirla en una *box* base que pueda usarse para construir a partir de ella con todo el *software* que necesitemos de partida adaptado a nuestras necesidades. En esta actividad se trata de seguir las indicaciones de teoría para convertir una máquina de las que hemos creado en fases anteriores **en una base box utilizable por Vagrant**, haciendo los cambios pertinentes para que *Vagrant* pueda interactuar con ella.

También puedes usar este enlace como guía: <https://www.dbigcloud.com/virtualizacion/146-como-crear-un-base-box-en-vagrant-a-partir-de-una-maquina-virtual.html>

Insignias y Autoevaluación

Esta asignatura hace uso de insignias. Esto quiere decir que, cuando vayas desbloqueando niveles de maestría en la tecnología, podrás obtener una **insignia pública** que luego podrás usar en tu currículum gracias al soporte de estas del campus virtual. Este enlace te enseña a usar esta característica para poder unir las a tu currículum: <https://docs.moodle.org/all/es/Insignias>. Cada insignia pública **lleva adjuntos los textos de las habilidades que desbloquea** que puedes consultar en la siguiente tabla, para que cualquiera los vea si decides usarlas y sepa lo que has conseguido hacer, validado por mí. De esta manera, mi intención es ayudarte a aumentar tus posibilidades de contratación en un futuro, **exponiendo claramente lo que has podido hacer en esta asignatura** referente a cada tecnología sobre la que has decidido evaluarte.

Rango de Maestría en la Tecnología y Habilidades Demostradas		Insignia Pública Obtenida
Nivel 1: Cabo de Vagrant		
	Puede instalar <i>Vagrant</i> en un sistema operativo que lo admita	
	Es capaz de arrancar una <i>box</i> de <i>Vagrant</i> sacada de la <i>Vagrant Cloud</i>	
	Puede consultar el estado de una máquina <i>Vagrant</i> activa y destruirla completamente cuando ya no hace falta	
	Puede provisionar una máquina <i>Vagrant</i> por <i>shell</i> para prepararla para trabajar por la línea de comandos	
Nivel 2: Sargento de Vagrant		
	Puede crear configuraciones diferentes por proveedor de virtualización	
	Puede parametrizar con variables distintos valores que se usen en una <i>Vagrantfile</i> , incluyendo características de una máquina virtual particulares de cada proveedor de virtualización (2 ejercicios)	
	Puede usar la característica de redimensionado de discos duros	
	Puede instalar de forma automatizada máquinas virtuales que ejecuten un servidor web y un servidor de base de datos (3 ejercicios)	
	Puede crear una <i>box</i> personalizada a partir de “empaquetar” una existente previamente personalizada	
Nivel 3: Teniente de Navío Vagrant		
	Puede refactorizar sus <i>provisioners</i> y usar <i>provisioners</i> que incorporen ficheros del host a la máquina virtual para mejorar la automatización de operaciones	
	Puede instalar automáticamente utilidades de virtualización en una máquina virtual que mejoren su rendimiento, incluso haciendo que el <i>script</i> de provisionamiento detecte el proveedor y elija el más adecuado	
	Puede extraer el mejor rendimiento de virtualización posible con <i>VirtualBox</i> y <i>Hyper-V</i> gracias a las características de <i>Vagrant</i> , tanto para <i>Windows</i> como para <i>Linux</i> (2 ejercicios)	



	Puede decidir si no instalar un GUI, instalar uno ligero o uno completo con funciones de <i>Vagrant</i>	
	Puede crear infraestructuras de máquinas virtuales más complejas con proveedores multi-máquina (2 ejercicios)	
Nivel 4: Capitán de Navío Vagrant		
	Puede definir a voluntad carpetas compartidas con el <i>host</i> que pongan en las máquinas virtuales cualquier contenido del <i>host</i> en el lugar de la máquina que se desee	
	Puede crear nuevos tipos de disco en una máquina virtual	
	Es capaz de crear nuevas redes públicas o privadas y mapear los puertos de las máquinas virtuales creadas para la construcción de infraestructuras más complejas	
	Es capaz de instalar otros sistemas operativos que no son <i>Ubuntu</i> en máquinas <i>Vagrant</i> , como <i>Oracle Linux</i> o <i>Kali Linux</i>	
	Es capaz de personalizar imágenes <i>Ubuntu</i> con GUI y optimizadas para crear entornos de desarrollo usados en otros contextos	
Nivel 5: Almirante Vagrant		
	Es capaz de exportar ficheros .ova que contengan máquinas virtuales que ocupen el menor espacio posible (2 ejercicios)	
	Es capaz de crear múltiples máquinas que sean réplicas de una dada, pero que minimicen el espacio ocupado tanto en disco como en RAM. Además, es capaz de arrancarlas en modo <i>headless</i> para minimizar aún más los recursos consumidos	
	Puede transformar discos duros de máquinas virtuales entre distintos formatos admitidos por diversos proveedores de virtualización	
	Puede crear tus propias <i>base boxes</i> mediante la preparación para su uso con <i>Vagrant</i> , de acuerdo con las especificaciones de <i>Hashicorp</i> , de un sistema <i>Linux</i> instalado en una máquina virtual existente	
	Cruz con el distintivo rojo en Vagrant: Ha completado con éxito 4,7 puntos (aproximadamente el 80%) de las tareas correspondientes de los 5 niveles de la parte de <i>Vagrant</i> , demostrando así su maestría en esta tecnología y estar preparado para usarla en tu futuro trabajo de forma efectiva.	