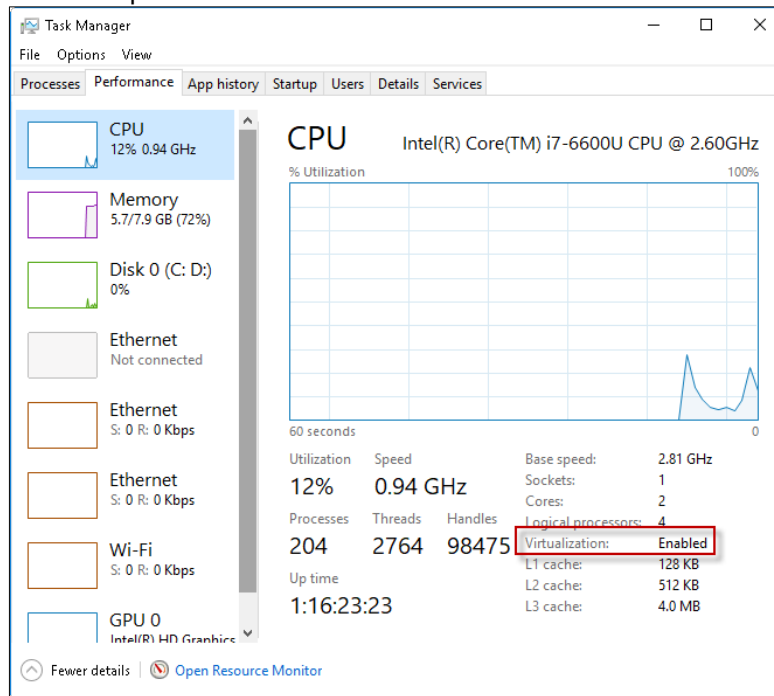


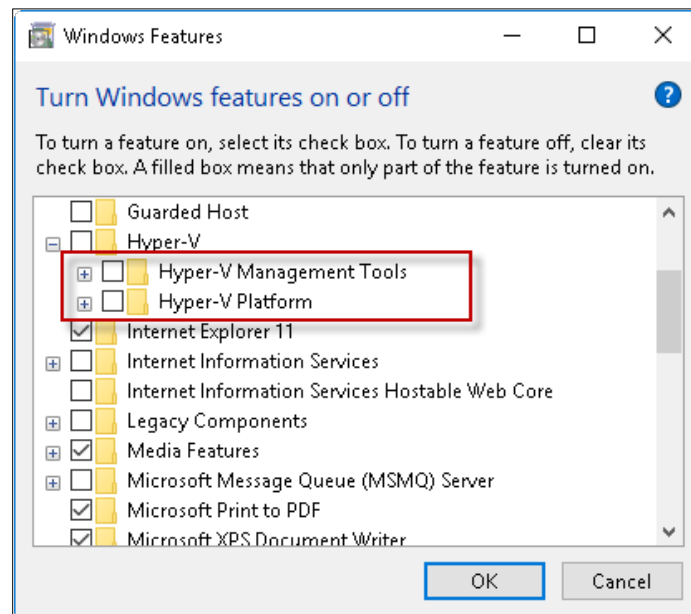
Guía técnica de referencia de Vagrant

I. Instalación y configuración

- Ir a <https://www.vagrantup.com/downloads.html> y descargar la versión apropiada para su sistema operativo.
- Tener en cuenta estas consideraciones adicionales:
 - Tener habilitado el soporte VTx:



- Deshabilitar las funcionalidades de Hyper-V, si estuviesen estas habilitadas:



- Probar desde la shell que ya esté el comando vagrant disponible:

```
vagrant --version
```

II. Primeros pasos

- Inicializar una configuración de vagrant con un box específico del catálogo:

```
vagrant init ubuntu/bionic64
```

- Levantar el box antes inicializado:

```
vagrant up
```

- Ver el estado del box creado:

```
vagrant status
```

- Conectarse por ssh al box creado:

```
vagrant ssh
```

- Cerrar sesión en la shell Linux del box ingresado, y luego destruirlo:

```
vagrant destroy
```

III. Tareas adicionales comunes

- En nuestro equipo, crear un directorio de trabajo exclusivo para vagrant. En dicha ruta, se deben crear subdirectorios específicos para cada proyecto de vagrant que deseemos utilizar. Ejm:

```
mkdir vagrant
mkdir vagrant/SAP4Hana
mkdir vagrant/pruebas-ubuntu
```

- Dentro de cada subdirectorio (o directorio de proyecto) se realiza tanto la inicialización como la ejecución de comandos deseados de vagrant:

```
cd vagrant/SAP4Hana
vagrant init centos/6
vagrant up
```

```
cd ..
cd vagrant/pruebas-ubuntu
vagrant init ubuntu/xenial64
vagrant up
vagrant ssh
```

- Dentro de un box (cuando ya nos conectamos a él, por ssh por ejemplo) se puede acceder al contenido del directorio del proyecto de vagrant bajo la ruta /vagrant. Este contenido se monta como una carpeta compartida desde el host al box (guest).

IV. Manejo de boxes y proveedores

- Consultar el catálogo completo de boxes de vagrant en <https://app.vagrantup.com/boxes/search>
- Descargar un box (Ejm: centos/7) del proveedor predeterminado (virtualbox):

```
vagrant box add centos/7
```

- El proveedor predeterminado se puede definir en la variable de entorno `VAGRANT_DEFAULT_PROVIDER`
- Descargar un box (Ejm: centos/7) del proveedor libvirt específico.

```
vagrant box add centos/7 --provider=libvirt
```

- Listar los boxes disponibles localmente

```
vagrant box list
```

- Conectarse por ssh a nuestro box Linux:

```
vagrant ssh
```

- Apagar un box en ejecución:

```
vagrant halt
```

- Encender un box:

```
vagrant up
```

- Encender un box de un proveedor específico (en caso tengamos más de uno instalado):

```
vagrant up --provider=aws
```

- Destruir un box:

```
vagrant destroy
```

- Más información sobre boxes en <https://www.vagrantup.com/docs/boxes.html>

V. Configuración de Vagrantfile

- En cada subdirectorio de proyecto de vagrant, se crea un archivo de nombre `Vagrantfile` luego que hacemos la inicialización. Ahí podemos ajustar diversos parámetros como la cantidad de cores, memoria o disco del box a levantar. También, interfaces de red adicionales con configuraciones particulares e incluso la integración con rutinas de aprovisionamiento.
- A continuación, algunas opciones generales de configuración en el archivo `Vagrantfile`. Agregar lo siguiente justo debajo de la directiva `config.vm.box` ubicado en la parte superior:

```
# Definir el hostname que tendrá el sistema operativo del box
config.vm.hostname = "servweb01.mydomain.com"

# Habilitar una interfaz de red para comunicación con el box desde
# el host a través de una IP privada
config.vm.network "private_network", ip: "192.168.33.10"
```

```
# Compartir dos directorios adicionales con el box
# Compartir la ruta ../docs montado como /documentos en el box
config.vm.synced_folder = "../docs", "/documentos"
# Compartir la ruta /data montado como /opt/data en el box
config.vm.synced_folder = "/data", "/opt/data"
```

- A continuación, algunas opciones de configuración en el archivo `Vagrantfile` aplicales solo para el proveedor `virtualbox`. Agregar lo siguiente justo antes de la directiva `end` ubicada al final:

```
config.vm.provider "virtualbox" do |v|
  # Memoria en MB
  v.memory = 1024

  # Numero de vCPUs
  v.cpus = 2

  # Nombre de la VM
  v.name = "vm-vbox-ubuntu18"
end
```

- También es posible configurar opciones de aprovisionamiento de archivos y ejecución de scripts o herramientas de automatización. Para ello, agregar lo siguiente justo antes de la directiva `end` ubicada al final:

```
# Copiar el script local /data/backups.sh al subdirectorio bin/ debajo
# del directorio personal del usuario vagrant dentro del box
config.vm.provision "file", source: "/data/backups.sh", destination: "$HOME/bin"

# Copiar el script local users.txt a /tmp/ dentro del box
config.vm.provision "file", source: "users.txt", destination: "/tmp"

# Ejecutar una serie de sentencias de comandos Linux una vez que el box levante
config.vm.provision "shell", inline: <<-MIMARCA
  apt-get update
  apt-get install -y apache2
MIMARCA

# Ejecutar un playbook de ansible (wordpress.yml) como aprovisionamiento del box
config.vm.provision "ansible" do |ansible|
  ansible.playbook = "wordpress.yml"
end
```

- Las sentencias de aprovisionamiento `config.vm.provision` se ejecutan solo la primera vez que levanta el box (`vagrant up`). Si se desea volver a ejecutar manualmente las sentencias de aprovisionamiento, ejecutar esto:

```
vagrant provision
```

- Más opciones de configuración en <https://www.vagrantup.com/docs/vagrantfile/> y <https://www.vagrantup.com/docs/virtualbox/configuration.html> (específico para el proveedor `virtualbox`).

VI. Plugins de Vagrant

- Repositorio de plugins
 - <https://github.com/hashicorp/vagrant/tree/master/plugins> (base)
 - <https://github.com/hashicorp/vagrant/wiki/Available-Vagrant-Plugins> (adicionales)
- También, se puede buscar gemas de Ruby en <https://rubygems.org/> que coincidan con “vagrant” para ser instalados como plugins.
- Instalar un plugin de vagrant a partir de una gema disponible en <https://rubygems.org/>:

```
vagrant plugin install <nombre-gema>
```

- O si se descargó localmente la gema, proceder así:

```
vagrant plugin install <ruta-local-fichero.gem>
```

- Listar plugins instalados:

```
vagrant plugin list
```

- Más información sobre plugins en <https://www.vagrantup.com/docs/cli/plugin.html>

VII. Próximos pasos

¿Qué más se puede hacer con Vagrant? Vendría bien que se investigue, pruebe y documente aspectos como:

- ✓ Usar vagrant con otros proveedores como VMware Workstation, vSphere, Hyper-V, AWS, Azure u otros.
- ✓ Probar algunos otros plugins de vagrant para añadirle funcionalidades extra.
- ✓ Descargar y usar boxes de Windows, probar su configuración en Vagrantfile para conectarse por WinRM y RDP, así como integrarlo con Ansible o scripts de PowerShell.
- ✓ Cómo crear boxes propios personalizados con el apoyo de la herramienta Packer.