Santiago de Cali Febrero 26 del 2020
Mateo Castro 1633561-2711
Manuel Díaz 1741652-2711
Universidad del Valle

Práctica Vagrant

Punto 1

Instale un servidor web y haga accesible este servicio de red desde un navegador el cual deberá estar corriendo en su estación de trabajo.

Primero realizaremos los pasos que se mencionan en la guia

1. probamos que vagrant este corriendo con el comando

   ```
   vagrant -v
   ```

2. luego para mirar qué box (distribuciones que se usan para las máquinas virtuales) tenemos descargadas usamos el siguiente comando:

   ```
   vagrant box list
   ```

3. Luego agregamos la box que necesitamos, en este caso ubuntu/xenial64 con el siguiente comando:

   ```
   vagrant box add ubuntu/xenial64
   ```

4. luego vamos a crear un insumo con el box que acabamos de descargar para poder iniciar la maquina virtual con el comando:

   ```
   vagrant init ubuntu/xenial64 -m
   ```

5. Con el comando anterior creamos el Vagrantfile que contiene la configuración de la máquina que vamos a levantar con el comando:

   ```
   vagrant up
   ```

6. luego para entrar a la máquina utilizamos el comando:

```
vagrant ssh
```

7. cuando necesitemos destruir la máquina usamos:

```
vagrant destroy -f
```

8. Ya estando aquí vamos al Vagrantfile (está en el equipo host, no dentro de la máquina virtual) y lo modificamos, entramos con el comando:

```
nano Vagrantfile
```

   agregamos las siguientes líneas para hacer el port forwarding

```
config.vm.network :forwarded_port, guest: 80, host: 8000
config.vm.network "private_network", ip: "192.168.33.90"
config.vm.network "public_network", ip: "192.168.33.90"
```

   Grabamos el archivo y toca apagar la máquina y volverla a iniciar para que se vean los cambios

9. Una vez hecho esto e iniciado la maquina virtual con los nuevos cambios entramos en ella con el comando

```
vagrant ssh
```

   después de esto instalamos un servidor web, nosotros utilizaremos apache2 y lo instalamos con el comando

```
sudo apt-get install apache2
```

10. Ya hecho esto está corriendo y podremos acceder a nuestro servidor desde el navegador del host entrando a un navegador web y digitando las direcciones:

```
http://localhost:8000 o 192.168.33.90
```

    y listo, hemos accedido a nuestro servidor web

    VIDEO https://asciinema.org/a/qhwm7yJTtx6y9GkcXI4x8RMwn

Punto 2

- Use el comando VBoxManage showvminfo para obtener la información de la máquina virtual creada con Vagrant. Guarde esa información en un archivo llamado ‘showvminfo.orig’
- Ahora, identifique la sintaxis de los comandos de VBoxManage que permiten ajustar las siguientes características de una máquina virtual cuando esta está apagada
    - Nombre de la máquina virtual a demo-00
    - Número de CPUs asignadas a la máquina virtual a 1 CPU
    - Tamaño en RAM de la máquina virtual a 720 MB
    - Modificación del porcentaje de CPU que se le asigna a una máquina virtual[1]
    - Habilitar/deshabilitar el servicio de *port forwarding* en la primera interfaz de red de modo que el puerto de red 8080 de la máquina física este asociado con el puerto de red 80 de la máquina virtual
- ¿Cuál de las tareas anteriormente mencionadas se pueden ejecutar usando el subcomando controlvm?

1. Salimos de la máquina virtual con el comando exit y desde aquí listamos las máquinas corriendo con el comando

    VBoxManage list vms

    en mi caso se llama "manuel_default_1582689014245_86178"
    ahora usamos el comando

    VBoxManage showvminfo manuel_default_1582689014245_86178

    y nos sale la información de la máquina virtual y la guardamos en un archivo llamado showvminfo.orig usando el siguiente comando

    nano showvminfo.orig

1

copiamos y pegamos la información y guardamos con ctrl+o y luego ctrl+x para salir.

2. - Para cambiar el nombre de la máquina virtual usamos el comando

```
VBoxManage modifyvm manuel_default_1582689014245_86178 --name demo00
```

-Para cambiar el número de Cpus asignadas a la máquina virtual usamos el comando:

```
VBoxManage modifyvm demo00 --cpus 1
```

-Para cambiar el tamaño de la RAM asignada a la máquina virtual usamos el comando:

```
VBoxManage modifyvm demo00 --memory 720
```

-Para cambiar el porcentaje de Cpu asignado a la máquina virtual usamos el comando

```
VBoxManage modifyvm demo00 --cpuexecutioncap 50
```

-Para activar el port forwarding en el puerto 8080 usamos el comando

```
VBoxManage modifyvm demo00 --natpf1 "tcp-port8080,tcp,127.0.0.1,8080,,80"
```

VIDEO: https://asciinema.org/a/M4NiNjqYhhs1xNDcVBGhhDrnK

3. Los dos comandos que funcionan con el controlvm serían cpuexecutioncap y el natpf1 con una pequeña diferencia que seria quitando los guiones (--) Recordemos que el controlvm es cuando la maquina ya esta corriendo osea hacer cambios en ejecución y el modifyvm es cuando la máquina está apagada

Punto 3

Modifique el archivo Vagrantfile que acaba de construir y haga las modificaciones necesarias para que al momento de crear la máquina virtual esta quede con las modificaciones sugeridas en la actividad anterior

- Nombre de la máquina virtual a demo-00
- Número de CPUs asignadas a la máquina virtual a 1 CPU
- Tamaño en RAM de la máquina virtual a 720 MB
- Modificación del porcentaje de CPU que se le asigna a una máquina virtual

El archivo queda de la siguiente forma

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

Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"
  config.vm.provider :virtualbox do |vb|
        vb.customize [ 'modifyvm', :id, '--name', 'demo-00' ]
        vb.customize [ 'modifyvm', :id, '--cpus', 1 ]
        vb.customize [ 'modifyvm', :id, '--memory', 720 ]
        vb.customize [ 'modifyvm', :id, '--cpuexecutioncap', 60 ]
 end
end
```

se agregan las líneas que comienzan con vb.customize con los comandos mencionados anteriormente en el taller (--name,--cpus,--memory,--cpuexecutioncap)

VIDEO https://asciinema.org/a/dGIGcfxycCHeZrFtvLeR1z4n7

Punto 4

Haga las modificaciones que resulten pertinentes de modo que Vagrant cree una máquina virtual pero con el servidor web de Apache instalado.

Creamos un archivo llamado script.sh con el comando:

```
nano script.sh
```

y en el insertamos lo siguiente

```
#!/usr/bin/env bash
sudo apt-get update
sudo apt-get -y install htop --fix-missing
sudo apt-get -y install apache2
```

guardamos con ctrl+o, después enter, despues ctrl+x para salir

además debemos modificar el Vagrantfile para ellos usamos el comando

```
nano Vagrantfile
```

y en el agregamos la linea

```
config.vm.provision "shell", path: "script.sh"
```

por lo tanto nuestro archivo queda de la siguiente forma

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

Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"
  config.vm.provision "shell", path: "script.sh"
  config.vm.provider :virtualbox do |vb|
        vb.customize [ 'modifyvm', :id, '--name', 'demo-00' ]
        vb.customize [ 'modifyvm', :id, '--cpus', 1 ]
        vb.customize [ 'modifyvm', :id, '--memory', 720 ]
        vb.customize [ 'modifyvm', :id, '--cpuexecutioncap', 60 ]
 end
end
```

VIDEO https://asciinema.org/a/6snKw2XmIn1Zq1PMtzR5HCA0B

Punto 5

Construya un Vagrantfile que contenga todos los comandos que se definieron en:

- Personalizando la máquina virtual
- Aprovisionando una máquina virtual
- Redirección de puertos

Valide que ahora usted es capaz en un solo comando de crear una máquina virtual con un servidor web y que ese servidor web está accesible desde el navegador que corre en su estación de trabajo.

ahora para habilitar el port forwarding agregamos en el Vagrantfile la siguiente línea

```
config.vm.network "forwarded_port", guest: 80, host: 8080
```

Y nos queda así:

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

Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"
  config.vm.provision "shell", path: "script.sh"
  config.vm.network "forwarded_port", guest: 80, host: 8080
  config.vm.provider :virtualbox do |vb|
       vb.customize [ 'modifyvm', :id, '--name', 'demo-00' ]
       vb.customize [ 'modifyvm', :id, '--cpus', 1 ]
       vb.customize [ 'modifyvm', :id, '--memory', 720 ]
       vb.customize [ 'modifyvm', :id, '--cpuexecutioncap', 60 ]
 end
end
```

y el archivo script.sh nos queda asi

```
#!/usr/bin/env bash
sudo apt-get update
sudo apt-get -y install htop --fix-missing
```

```
sudo apt-get -y install apache2
```

Punto 6

Siguiendo con el Vagrantfile con el que ha venido trabajando encarguese ahora de que las páginas HTML del servidor web en la máquina virtual esten disponibles desde el *host*.

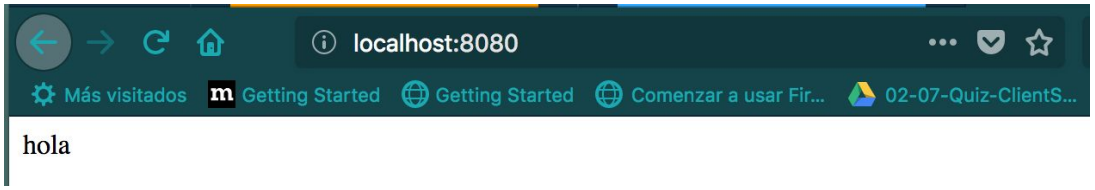
Recomendaciones

- Cree una carpeta en el *host* que se llame *www*. En esta carpeta cree un archivo HTML que contenga lo siguiente:

```
<html>
<body>
hola
</body>
</html>
```

- Haga los ajustes en su Vagrantfile de modo que la carpeta donde están almacenados los archivos *www* de su máquina virtual apunten realmente al directorio *www* del *host*[2].

Valide ahora que al apuntar al URL http://localhost:8080 tenga acceso a una página con esta información



ahora para compartir directorios agregamos en el Vagrantfile la siguiente línea

```
config.vm.synced_folder "./www", "/var/www/html"
```

La parte de este **color** es para mi directorio en el host osea mi computador
y la parte de este otro **color** es para la carpeta dentro de la máquina virtual

ahora en nuestro computador local creamos una carpeta con el nombre www y dentro ponemos un archivo index.html que contiene lo siguiente:

```
<html>
<body>
```

2

```
hola
</body>
</html>
```

y nuestro Vagrantfile queda de la siguiente manera:

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

Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"
  config.vm.provision "shell", path: "script.sh"
  config.vm.synced_folder "./www", "/var/www/html"
  config.vm.network "forwarded_port", guest: 80, host: 8080
  config.vm.provider :virtualbox do |vb|
      vb.customize [ 'modifyvm', :id, '--name', 'demo-00' ]
      vb.customize [ 'modifyvm', :id, '--cpus', 1 ]
      vb.customize [ 'modifyvm', :id, '--memory', 720 ]
      vb.customize [ 'modifyvm', :id, '--cpuexecutioncap', 60 ]
 end
end
```

Punto 7

Genere un archivo Vagrantfile el cual tenga la definición de dos máquinas virtuales, web_1 y web_2. Ambas máquinas tendrán instalado el servidor apache2 pero:

- web_1 escuchará por el puerto 8080
- web_2 escuchará por el puerto 8008
- web_1 tendrá sus archivos web disponibles en el directorio www_1 en el *host* y dentro de este directorio tendrá un archivo HTML como el que vimos anteriormente en este documento
- web_2 tendrá sus archivos web disponibles en el directorio www_2 en el *host* y dentro de este directorio tendrá un archivo HTML que desplegará la palabra “hello”

He creado una carpeta llamada punto 7 entonces entramos a esta carpeta y usamos el siguiente comando para crear un Vagrantfile

```
vagrant init ubuntu/xenial64
```

después entramos a modificar el archivo con el comando nano Vagrantfile

Nuestro archivo Vagrantfile con todas las instrucciones necesarias queda de la siguiente manera

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

Vagrant.configure("2") do |config|
  config.vm.define "web_1" do |web_1|
     web_1.vm.box = "ubuntu/xenial64"
     web_1.vm.provision "shell", path: "script.sh"
     web_1.vm.network "forwarded_port", guest: 80, host: 8080
     web_1.vm.synced_folder "./www_1", "/var/www/html"
  end
  config.vm.define "web_2" do |web_2|
     web_2.vm.box = "ubuntu/xenial64"
     web_2.vm.provision "shell", path: "script.sh"
     web_2.vm.network "forwarded_port", guest: 80, host: 8008
     web_2.vm.synced_folder "./www_2", "/var/www/html"
  end
end
```

En la misma carpeta que está el Vagrantfile creamos un archivo **script.sh** igual al de los puntos 1 al 6 que queda de la siguiente manera

```
!/usr/bin/env bash
sudo apt-get update
sudo apt-get -y install htop --fix-missing
sudo apt-get -y install apache2
```

En la carpeta que estan estos archivos creamos una carpeta llamada www_1 y www_2.

dentro de www_1 ponemos un archivo llamado index.html con lo siguiente

```
<html>
<body>
hola
</body>
</html>
```

y dentro de www_2 ponemos un archivo muy parecido que en vez de decir hola diga hello y queda así

```
<html>
<body>
hello
</body>
</html>
```

ya hecho todo esto, vamos a la consola de comandos y ejecutamos

```
vagrant up
```

después en nuestro navegador vamos a la dirección

http://localhost:8080
http://localhost:8008

el primero dirá hola y el segunda dirá hello
VIDEO : https://asciinema.org/a/VYmLauhbbOcsrm02s9TaBeluw

VIDEOS

PUNTO 1: https://asciinema.org/a/qhwm7yJTtx6y9GkcXI4x8RMwn

PUNTO 2: https://asciinema.org/a/M4NiNjqYhhs1xNDcVBGhhDrnK

PUNTO 3 https://asciinema.org/a/dGIGcfxycCHeZrFtvLeR1z4n7

PUNTO 4 https://asciinema.org/a/6snKw2XmIn1Zq1PMtzR5HCA0B

PUNTO 7 : https://asciinema.org/a/VYmLauhbbOcsrm02s9TaBeluw

A algunos puntos no les hice videos ya que toma bastante tiempo en borrar y volver a iniciar las máquinas, el punto más completo es el 7 que contiene todo corriendo