Practica02

1. Vagrant.

Instalación de Vagrant.

Actualización de paquetes: "sudo apt-get update"

Instalación de Vagrant

```
sudo apt-get install Vagrant
```

En caso de no tener instalado VirtualBox, debemos instalarlo:

```
sudo apt-get install VirtualBox
```

Comandos Básicos de Vagrant.

A continuación se muestran los comandos que indtroduciremos por consola para poder trabajar con Vagrant.

Inciar la máquina.

```
vagrant up
```

Reiniciar la máquina. Esto se utiliza cuando hacemos algun cambio en "Vagrantfile" y queremos que se apliquen los cambios.

```
vagrant reload
```

Ejecutar los scripts "provisioners". Siempre que realicemos algun cambio en estos scripts, debemos hacerlo.

```
vagrant provision
```

Inicializar el vagrantfile.

```
vagrant init
```

Apagar la máquina virtual

```
vagrant halt
```

Elimiar la máquina virtual

```
vagrant destroy
```

Conectar a una máquina por ssh

```
vagrant ssh
```

Comprobar estado de la maquina

```
vagrant status
```

2. Realización de la práctica

Creción del directorio y la máquina

Lo primero que debemos hacer es crear una carpeta, a la cual llamaremos "iaw" y dentro de la misma, volveremos a crear otro directorio que se llame "maquina01".En mi caso utilizaré la cmd de windows10

```
c:\users\Alex>mkdir iaw
c:\users\Alex>cd iaw
c:\users\Alex>mkdir maquina01
c:\users\Alex>cd maquina01
```

El siguiente paso será ejecutar un "vagrant init" para que se cree un "vagrant file" dentro de este directorio llamado "maquina01", y a continuación lanzar un "vagrant up" para encender la máquina.

Modificación del "vagrantfile" y creación del "script.sh"

Vagrantfile

Una vez hayamos ejecutado el vagrant init veremos que se nos ha creado un "vagrantfile" y debemos modificarlo para que quede de la siguiente manera:

```

```

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

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "ubuntu/bionic64"

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

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
  # NOTE: This will enable public access to the opened port
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine and only allow access
  # via 127.0.0.1 to disable public access
  # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip:
"127.0.0.1"

  # Create a private network, which allows host-only access to the machine
  # using a specific IP.
  config.vm.network "private_network", ip: "192.168.33.10"

  # Create a public network, which generally matched to bridged network.
  # Bridged networks make the machine appear as another physical device on
  # your network.
  # config.vm.network "public_network"

  # Share an additional folder to the guest VM. The first argument is
  # the path on the host to the actual folder. The second argument is
  # the path on the guest to mount the folder. And the optional third
  # argument is a set of non-required options.
  # config.vm.synced_folder "../data", "/vagrant_data"

  # Provider-specific configuration so you can fine-tune various
  # backing providers for Vagrant. These expose provider-specific options.
  # Example for VirtualBox:
  #
  # config.vm.provider "virtualbox" do |vb|
```

```
  #   # Display the VirtualBox GUI when booting the machine
  #   vb.gui = true
  #
  #   # Customize the amount of memory on the VM:
  #   vb.memory = "1024"
  # end
  #
  # View the documentation for the provider you are using for more
  # information on available options.

  # Enable provisioning with a shell script. Additional provisioners such as
  # Puppet, Chef, Ansible, Salt, and Docker are also available. Please see the
  # documentation for more information about their specific syntax and use.
  config.vm.provision "shell", path: "script.sh"
end
```

Las líneas que hemos modificado son las siguientes:

```
config.vm.box = "ubuntu/bionic64"
config.vmnetwork "private_network", ip: "192.168.33.10"
config.vm.provision "shell", path: "script.sh"
```

config.vm.box = "ubuntu/bionic64"

En esta línea le decimos a virtualbox que el sistema operativo que vamos a utilizar va a ser ubuntu server de 64bits.

config.vmnetwork "private_network", ip: "192.168.33.10"

Configuramos la ip que queremos utilizar para nuestro server.

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

Indicamos que vamos a utilizar un sript para la configuación de esta máquina. En este ejemplo vamos a configurar un script para que instale, a la vez que se crea nuestra máquina de ubuntuserver, apache2 y mysql.

Script.sh

Vamos a crear un scipt como hemos dicho antes, donde configuraremos la instalación de apache y mysql, para ello debemos crear un archivo ".sh" al cual yo llamaré "script.sh".

La estructura del mismo quedará tal que así:

```
apt-get update
apt-get install -y apache2
apt-get install -y php libapache2-mod-php php-mysql

apt-get update
```

```
apt-get -y install debconf-utils

DB_ROOT_PASSWD=root
debconf-set-selections <<< "mysql-server mysql-server/root_password password
$DB_ROOT_PASSWD"
debconf-set-selections <<< "mysql-server mysql-server/root_password_again password
$DB_ROOT_PASSWD"

apt-get install -y mysql-server
```