











Pregunta 3

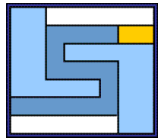
Nombre	
Apellidos	
Itinerario	No colaborativo

1. Describa dos ventajas y dos inconvenientes del uso de máquinas virtuales frente al uso de contenedores.
2. Dada la siguiente búsqueda en <https://vagrantcloud.com/search>. Indique que comando(s) de Vagrant debe lanzar para:

Discover Vagrant Boxes

<input type="text" value="ubuntu"/>		<input type="button" value="q"/>	
Provider: any virtualbox vmware libvirt more		Sort by: Downloads Recently Created Recently Updated	
 ubuntu/trusty64 20190514.0.0 Official Ubuntu Server 14.04 LTS (Trusty Tahr) builds (End of standard support)	virtualbox	Downloads 30,592,541	Released about 2 years ago
 hashicorp/precise64 1.1.0 A standard Ubuntu 12.04 LTS 64-bit box.	hyperv virtualbox vmware_fusion	Downloads 6,793,523	Released almost 8 years ago
 ubuntu/xenial64 20211001.0.0 Official Ubuntu 16.04 LTS (Xenial Xerus) builds (End of standard support)	virtualbox	Downloads 3,495,129	Released 3 months ago
 puphpet/ubuntu1404-x64 20161102 Ubuntu Trusty 14.04 LTS x64	parallels virtualbox vmware_desktop	Downloads 2,510,988	Released about 5 years ago
 hashicorp/precise32 1.0.0 A standard Ubuntu 12.04 LTS 32-bit box.	virtualbox	Downloads 2,292,508	Released almost 8 years ago
 bento/ubuntu-16.04 202112.19.0 Vanilla Ubuntu 16.04 Vagrant box created with Bento by Chef	hyperv parallels virtualbox vmware_desktop	Downloads 1,845,118	Released 6 days ago
 ubuntu/trusty32 20190514.0.0 Official Ubuntu Server 14.04 LTS (Trusty Tahr) builds (End of standard support)	virtualbox	Downloads 1,808,774	Released about 2 years ago
 bento/ubuntu-14.04 201808.24.0 Ubuntu 14.04 Vagrant box created with Bento by Chef	hyperv parallels virtualbox vmware_desktop	Downloads 985,345	Released over 3 years ago
 generic/ubuntu1604 3.6.2 A generic Ubuntu 16.04 (aka Xenial Xerus) image, ready for use as an appliance or development environment.	hyperv libvirt parallels virtualbox and 1 more providers	Downloads 835,339	Released 8 days ago
 generic/ubuntu1804 3.6.2 A generic Ubuntu 18.04 (aka Bionic Beaver) image, ready for use as an appliance or development environment.	hyperv libvirt parallels virtualbox and 1 more providers	Downloads 812,052	Released 8 days ago

- (a) descargar la imagen con la versión más reciente de Ubuntu compatible con "vmware", crear la máquina virtual con la configuración por defecto, y ejecutar la máquina con dicha imagen;
- (b) conectarse desde el sistema host y comprobar la versión exacta de Python 3 que está ejecutando la máquina virtual Ubuntu;



- (c) parar y destruir tanto la máquina virtual como la imagen del sistema.
3. Se desea desplegar una aplicación en una máquina virtual usando Vagrant. Se trata de una aplicación web desarrollada íntegramente en “Python 3” usando el framework de desarrollo web “Flask” y las librerías “numpy”, “pandas”, y “graphviz”, esta última librería usará la versión 0.19.1 y requiere además tener instalado el paquete “graphviz”. La aplicación está disponible en el repositorio “git” <https://github.com/EGC/Final2022.git>, en la rama “final22”, y se lanza con el fichero “main2022.py”. Se usará la misma imagen basada en Ubuntu del Apartado 2 y se usará el hipervisor “vmware”. La imagen seleccionada levanta un servidor web “nginx” en el puerto 8282, mientras que al sistema anfitrión solo le queda libre el puerto 8090. La aplicación es bastante exigente por lo que requiere al menos de 2 GB de memoria y 2 unidades de procesamiento. Dado el siguiente fichero Procfile creado por defecto:

```
1 Vagrant.configure("2") do |config|
2   # config.vm.box = "XXXXX"
3   # config.vm.box_check_update = false
4   # config.vm.network "forwarded_port", guest: 80, host: 8080
5   # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"
6   # config.vm.network "private_network", ip: "192.168.33.10"
7   # config.vm.network "public_network"
8   # config.vm.synced_folder "../data", "/vagrant_data"
9   # config.vm.provider "XXXXXX" do |v|
10    #   v.gui = true
11    #   v.memory = "1024"
12    #   v.cpus = 1
13    # end
14   # config.vm.provision "shell", path: 'provision.sh'
15 end
```

- (a) Indique las líneas que debe descomentar y modifique aquellas que necesite para desplegar nuestra aplicación en una máquina virtual configurada según los requisitos dados.

Linea: X, Y, Z,...
Linea: X -> configuracion
Linea: Y -> configuracion
...

- (b) Escribe el fichero de aprovisionamiento “provision.sh” y el fichero “requirements.txt” para poder ejecutar correctamente la aplicación en la máquina virtual.
4. En lugar de aprovisionar la máquina virtual del Apartado 3 con un script shell .sh, se desea proporcionar el aprovisionamiento con Ansible. Describa el playbook de Ansible definiendo las tareas necesarias y especificando los módulos que debe ejecutar cada tarea.