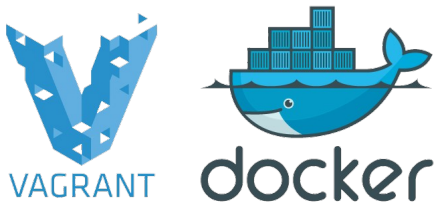
Escuela técnica superior
Facultad de Ingeniería Informática



Práctica Extra

Docker y Vagrant

Ignacio Fernández Contreras

1 Enunciado

Se desea desplegar una aplicación web en un servidor. Para ello se usará la siguiente aplicación ya disponible que puede descargarse (clonarse) del siguiente Repositorio:

Parte 1. Desplegar la aplicación web en un contenedor Docker. Para ello:

1. Instale Docker siguiendo las instrucciones la siguiente página:

```
// Desistalamos versiones antiguas:
sudo apt-get remove docker docker-engine docker.io containerd runc

//Intalamos dependencias
sudo apt-get update

sudo apt-get install \
        ca-certificates \
        curl \
        gnupg \
        lsb-release

//Instalamos llave de crifrado
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /
    etc/apt/keyrings/docker.gpg

//Intalamos docker
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io

//Instalamos docker compose
sudo apt-get install docker-compose-plugin

//Agregamos nuestro usuario al grupo docker
sudo usermod -aG docker $USER
newgrp docker
```

2. Dockerize la aplicación (crea un contenedor y ejecútalo) siguiendo las instrucciones de la siguiente página:

   Comenzamos creando nuestra app en el archivo holamundo.py

```
→  1920-Practica-1 git:(master) ✗ cat holamundo.py
# Importing flask module in the project is mandatory
# An object of Flask class is our WSGI application.
from flask import Flask
# Flask constructor takes the name of
# current module (__name__) as argument.
app = Flask(__name__)
# The route() function of the Flask class is a decorator,
# which tells the application which URL should call
# the associated function.
@app.route('/')
# '/' URL is bound with hello_world() function.
def hello_world():
    return 'Hello World'

@app.route('/hello/<name>')
def hello_name(name):
    return 'Hello %s!' % name

# main driver function
if __name__ == '__main__':
# run() method of Flask class runs the application
# on the local development server.
    app.run(host='0.0.0.0')

#credits to https://www.geeksforgeeks.org/flask-creating-first-simple-application/
```

3. Definimos las dependencias en el archivo requirements.txt

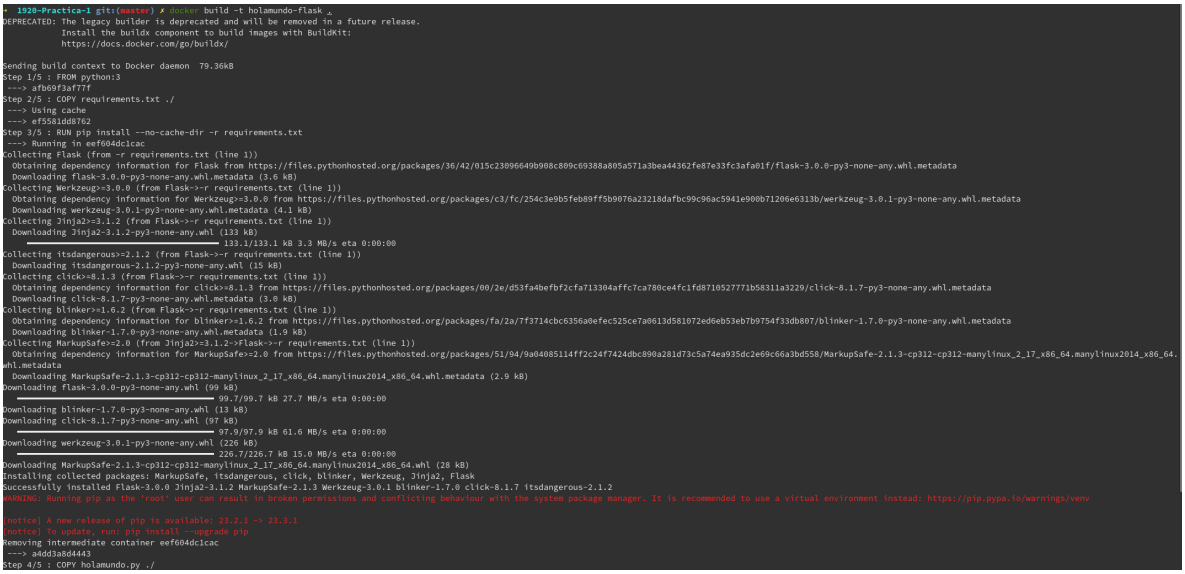
```
Flask
```

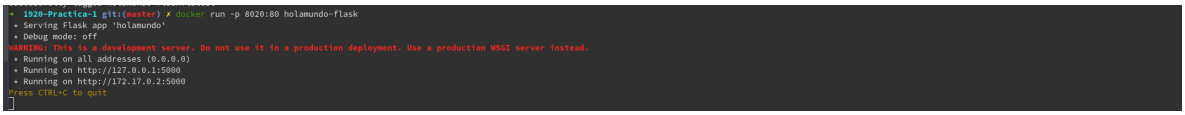
4. Podemos usar la imagen de python3 existente en los repositorios de docker, y modificarla anadiendo la aplicacion que hemos desarrollado. Para ello podemos crear un fichero Dockerfile que nos automatiza la creacion de una imagen modificada:

```
→  1920-Practica-1 git:(master) ✗ cat Dockerfile
# Base image
FROM python:3
#copy the requirements txt file with all our dependencies
COPY requirements.txt ./
#install the dependencies
RUN pip install --no-cache-dir -r requirements.txt
#copy the app in the image
COPY holamundo.py ./
#define a default command to execute
CMD [ "python", "./holamundo.py" ]
```

5. Este fichero indica que docker tiene que descargar la imagen python en su tercera versión y tiene que añadir el fichero requirements a la carpeta de usuario de la imagen. Docker nos ofrece un lenguaje de scripting para automatizar la creacion de imagenes.

```
docker build -t holamundo-flask .
docker run -p 8020:80 holamundo-flask
```





Parte 2. Desplegar la aplicación web en una máquina virtual usando Vagrant. Para ello:

1. Instale Vagrant siguiendo las instrucciones de la siguiente página:

   Intalación:

```
sudo apt update
sudo apt install vagrant ansible virtualbox

//Para probar que funciona
vagrant init ubuntu/trusty32
vagrant up
#Suele tardar varios minutos
vagrant ssh -c 'hostnamectl'
#Comparar que devuelve el comando con respecto a ejecutar hostnamectl en
    local
vagrant destroy #Para eliminar la imagen y liberar espacio
```

```
→ 1920-Practica-1 git:(master) ✗ vagrant ssh -c 'hostnamectl'
  Static hostname: ubuntu-bionic
        Icon name: computer-vm
          Chassis: vm
       Machine ID: 4d91ed0819084139a79b5ec4f2d9af13
          Boot ID: e7063057d39a4dda97d7969f838c4ee0
   Virtualization: oracle
 Operating System: Ubuntu 18.04.6 LTS
           Kernel: Linux 4.15.0-212-generic
     Architecture: x86-64
```

2. Virtualize la aplicación (crea una máquina virtual y ejecútela) siguiendo las instrucciones de la siguiente página:

   Encapsulado y aprovisionamiento de una app

   Creamos un Vagrantfile

```
Vagrant.configure("2") do |config|
        config.vm.box = "ubuntu/bionic64"
        config.vm.provision "shell", inline: "sudo apt-get update && sudo apt
            -get install -y docker.io"
        config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "
            127.0.0.1"
end
```

   Creamos nuestra forma de aprovisionar (de momento con un script bash)

```
sudo apt update
sudo apt upgrade -y
sudo apt install -y git python3 python3-pip screen
git clone https://github.com/EGCETSII/1920-Practica-1.git
cd 1920-Practica-1
pip3 install -r requirements.txt
screen -m -d python3 holamundo.py
```

   vagrant up

```
→  1920-Practica-1 git:(master) ✗ vagrant ssh
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-212-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Dec  2 13:27:27 UTC 2023

  System load:  0.54              Processes:             100
  Usage of /:   3.0% of 38.70GB   Users logged in:       0
  Memory usage: 13%               IP address for enp0s3: 10.0.2.15
  Swap usage:   0%


Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

Enable ESM Infra to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Sat Dec  2 13:20:24 2023 from 10.0.2.2
vagrant@ubuntu-bionic:~$
```

```
vagrant@ubuntu-bionic:/vagrant$ sudo docker build -t holamundo-app .
Sending build context to Docker daemon  144.4kB
Step 1/5 : FROM python:3
3: Pulling from library/python
90e5e7d8b87a: Pull complete
27e1a8ca91d3: Pull complete
d3a767d1d12e: Pull complete
711be5dc5044: Pull complete
48b2d58a56e9: Pull complete
b61fb8c5b702: Download complete
67ddeb5b15df: Download complete
7da1b82bcb72: Download complete
```

```
vagrant@ubuntu-bionic:/vagrant$ sudo docker run -p 8080:80 holamundo-app
 * Serving Flask app 'holamundo'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.2:5000
Press CTRL+C to quit
```