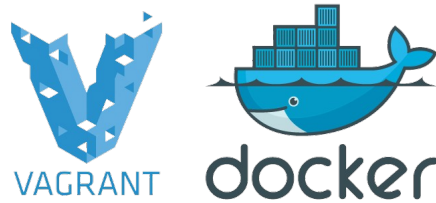


Escuela técnica superior
FACULTAD DE INGENIERÍA INFORMÁTICA



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](#):

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

//Instalamos dependencias
sudo apt-get update

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

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

//Instalamos 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 añadiendo 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) x 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
```

```
1920-Practica-1 git:(master) x docker build -t holamundo-flask .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 79.36kB
Step 1/5 : FROM python:3
--> af69f3af77f
Step 2/5 : COPY requirements.txt ./
--> Using cache
--> ef5581dd8762
Step 3/5 : RUN pip install --no-cache-dir -r requirements.txt
--> Running in ef604dc1cac
Collecting Flask from https://files.pythonhosted.org/packages/36/42/015c2396649b08c809c69388a805a571a3bea44362fe07e33fc3afa01f/flask-3.0.0-py3-none-any.whl.metadata
  Downloading flask-3.0.0-py3-none-any.whl.metadata (3.6 kB)
Collecting Werkzeug from https://files.pythonhosted.org/packages/c3/fc/254c3e9b5fe089f5b9076a23218dafbc99c96ac5941e900b71206e6313b/werkzeug-3.0.1-py3-none-any.whl.metadata
  Downloading werkzeug-3.0.1-py3-none-any.whl.metadata (4.1 kB)
Collecting Jinja2 from https://files.pythonhosted.org/packages/73/01/0000000000000000000000000000000000000000000000000000000000000000/jinja2-3.1.2-py3-none-any.whl.metadata
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
Collecting itsdangerous from https://files.pythonhosted.org/packages/01/01/0101010101010101010101010101010101010101010101010101010101010101/itsdangerous-2.1.2-py3-none-any.whl.metadata
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting click from https://files.pythonhosted.org/packages/00/2e/d53f4dbefbf2cfa713394affc7ca780c4fc1fd6710527771b58311a3229/click-8.1.7-py3-none-any.whl.metadata
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker from https://files.pythonhosted.org/packages/fa/2a/7f3714cb6356a8efec525c7a0613d581072ed6eb5eb7b754f3db807/blinker-1.7.0-py3-none-any.whl.metadata
  Downloading blinker-1.7.0-py3-none-any.whl.metadata (1.9 kB)
Collecting MarkupSafe from https://files.pythonhosted.org/packages/51/94/9a04085114ff2c247424db809a281d735a74ea935dc2e09c66a3bd558/MarkupSafe-2.1.3-cp12-cp12-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
  Downloading MarkupSafe-2.1.3-cp12-cp12-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (2.9 kB)
Installing collected packages: Werkzeug, Jinja2, Flask, itsdangerous, click, blinker, MarkupSafe
Successfully installed Flask-3.0.0 Jinja2-3.1.2 MarkupSafe-2.1.3 Werkzeug-3.0.1 blinker-1.7.0 click-8.1.7 itsdangerous-2.1.2
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: pip install --upgrade pip
Removing intermediate container ef604dc1cac
--> a4dd3a8d443
Step 4/5 : COPY holamundo.py ./
```

```
➔ 1920-Practica-1 git:(master) x docker run -p 8020:80 holamundo-flask
• 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
1
```

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) x 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) x 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
```