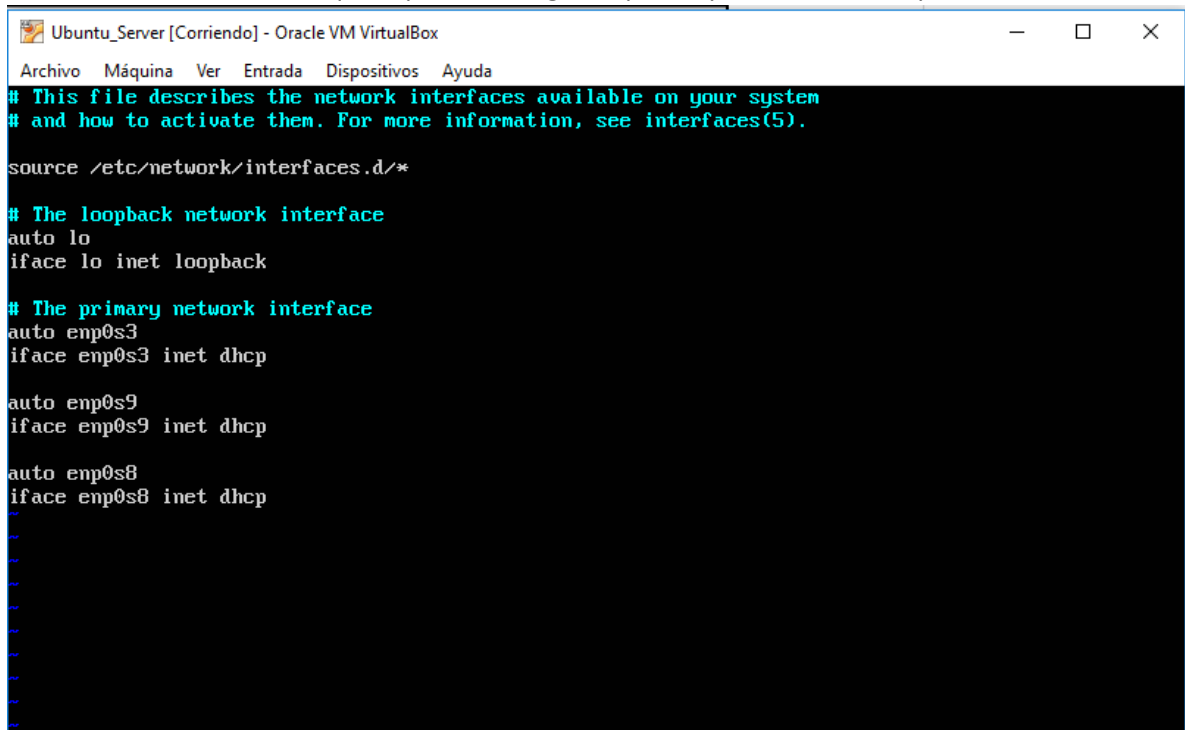


MINIPROYECTO SISTEMAS OPERATIVOS

- A00320176 - Daniel Gutierrez
- A00309828 - Emmanuel Bolaños
- A00170892 - Carlos Llano

URL repositorio:

- 1) Para instalar el sistema operativo de Ubuntu Server 16.04 se debe descargar el ISO de la página: <https://www.ubuntu.com/download/server>. Se selecciona el archivo ISO en almacenamiento y se configuran las interfaces de red igual que en CENTOS. Luego se debe seleccionar las características generales como: lenguaje de la máquina, configuración de usuarios y contraseñas, partición de discos, selección de programas.
- 2) En Ubuntu se deben configurar las interfaces de la siguiente manera:
 - a. Se agregan al archivo interfaces que se encuentra en la ruta `/etc/network/interfaces`. Aquí se puede configurar que su ip se tome via dhcp



```
Ubuntu_Server [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet dhcp

auto enp0s9
iface enp0s9 inet dhcp

auto enp0s8
iface enp0s8 inet dhcp

~
~
~
~
~
```

- b. Se reinicia el servicio de networking con el comando `service networking restart`



```
Ubuntu_Server [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
ubuntu@ubuntu:~$ sudo service networking restart
ubuntu@ubuntu:~#
```

- 3) Para abrir los puertos en Ubuntu utilizamos el comando **sudo ufw allow 5000**.
Después de esto debemos utilizar el comando **sudo ufw enable**.
- 4) Para el funcionamiento del proyecto se usaron los siguientes comandos para bajar las respectivas dependencias:
 - a. **Descargar dependencias necesarias para el funcionamiento de python**
 - i. `sudo apt-get install build-essential checkinstall`
 - ii. `sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev`
 - b. **Instalar Python en caso de que no se quiera usar la versión por defecto, que es python3**
 - i. `wget https://www.python.org/ftp/python/2.7.13/Python-2.7.13.tgz`
 - ii. `tar xzf Python-2.7.13.tgz`
 - iii. `cd Python-2.7.13 & sudo ./configure`
 - iv. `sudo make install`
- 5) Para crear el ambiente virtual debemos instalar pip, virtualenv y Flask.
 - a. Instalar virtualenv
 - i. `pip install virtualenv`
 - ii. `cd ~/`
 - iii. `mkdir envs`
 - iv. `cd envs`
 - v. `virtualenv flask_env`
 - b. Activar el ambiente
 - i. `. flask_env/bin/activate`
 - c. Desactivar el ambiente
 - i. `deactivate`

```

commands.py modelo.py modelo.pyc script.py web.py
ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples$ pip install vir
tualenv
Collecting virtualenv
  Downloading virtualenv-15.1.0-py2.py3-none-any.whl (1.8MB)
    100% |#####| 1.8MB 763kB/s
Installing collected packages: virtualenv
Successfully installed virtualenv-15.1.0
ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples$ mkdir env
ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples$ cd env/
ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples/env$ virtualenv
flask_env
New python executable in /home/ubuntu/proyecto/so-microservices-python/A00170892/07_flask_exam
pl
Installing setuptools, pip, wheel...done.
ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples/env$ ls
bin  include  lib  local  pip-selfcheck.json
ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples/env$ fl
flask flock
ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples/env$ cd flask_en
v/
ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples/env/flask_env$ l
s
bin  include  lib  local  pip-selfcheck.json
ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples/env/flask_env$ .
bin/
activate activate_this.py pip python python-config
activate.csh easy_install pip2 python2 wheel
activate.fish easy_install-2.7 pip2.7 python2.7
ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples/env/flask_env$ .
bin/activate
(flask_env) ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples/env/
flask_env$

```

- 6) Dentro del repositorio de github: https://github.com/CarlosLlano/so-microservices-python/tree/master/A00170892/07_flask_examples se puede encontrar la aplicación en python.
- 7) Validación de la ejecución del servicio

```

ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples$ sudo netstat -plnt
[sudo] password for ubuntu:
Conexiones activas de Internet (solo servidores)
Proto Recib Enviad Dirección local Dirección remota Estado PID/Program name
tcp 0 0 0.0.0.0:5000 0.0.0.0:* ESCUCHAR 7481/python
tcp 0 0 0.0.0.0:22 0.0.0.0:* ESCUCHAR 2991/sshd
tcp6 0 0 :::22 :::* ESCUCHAR 2991/sshd
ubuntu@ubuntu:~/proyecto/so-microservices-python/A00170892/07_flask_examples$

```

8)

API para monitoreo

Segundo parcial de Sistemas Operativos, Universidad Icesi

: Consulta de registros de monitoreo del Sistema operativo

Show/Hide | List Operations | Expand Operations

DELETE	/checks	
GET	/checks	retorna todos los checks
POST	/checks	
PUT	/checks	
DELETE	/checks/cpu/history	
GET	/checks/cpu/history	
POST	/checks/cpu/history	
PUT	/checks/cpu/history	

[BASE URL: / , API VERSION: FINAL]

GET

/checks

retorna todos los checks

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Lista de todos los checks registrados		

Try it out!

Hide Response

Curl

curl -X GET --header 'Accept: application/json' 'http://172.30.192.3:5000/checks'

Request URL

http://172.30.192.3:5000/checks

Response Body

{\"checks\": \"\": [[\"cpu:.8%\", \"memory:6.41%\", \"disk:3,2G\", \"httpd:inactive\"], [\"cpu:.7%\", \"memory:6.41%\", \"

Response Code

200

Response Headers

{
 \"date\": \"Sat, 06 May 2017 00:51:11 GMT\",
 \"server\": \"Werkzeug/0.12.1 Python/2.7.5\",
 \"content-length\": \"6758\",
 \"content-type\": \"application/json\"
}

[illegible]

