

## TALLER 2

Nombre del Estudiante: **Duván Pardo, Wilson López**

### 1. INTRODUCCIÓN

Cientos de las marcas más grandes del mundo confían en OpenStack para manejar sus negocios todos los días, lo que reduce los costes y ayudar a que se muevan más rápido. OpenStack tiene un fuerte ecosistema, y los usuarios que buscan apoyo comercial puede elegir entre diferentes productos y servicios OpenStack-powered en el mercado. El software está construido por una próspera comunidad de desarrolladores, en colaboración con los usuarios, y está diseñado para solventar nuestras necesidades.

Para cumplir con estos principios OpenStack está dividido en diferentes componentes que trabajan en conjunto. Esta integración es lograda a través de interfaces de programación de aplicaciones – APIs – que cada servicio ofrece y consume, gracias a estas APIs, los servicios pueden comunicarse entre ellos y además se posibilita que un servicio sea reemplazado por otro de similares características siempre que se respete la forma de comunicación. Es decir, OpenStack es extensible y se ajusta a las necesidades de quien desee implementarlo. (tomado de ¿Cómo funciona OpenStack? y openstack)

**OpenStack y RedHat:** “Las personas tienden a optar por las mejores ideas. En 2004, Linux® se parecía mucho al actual OpenStack®. OpenStack está creciendo rápidamente, con el apoyo de una comunidad de individuos y empresas, y creando una plataforma que todos podemos usar para crear nubes abiertas enormemente escalables. Por eso creemos que OpenStack es el mejor exponente de la tecnología de nube.” (tomado de RedHat)

### 2. OBJETIVO

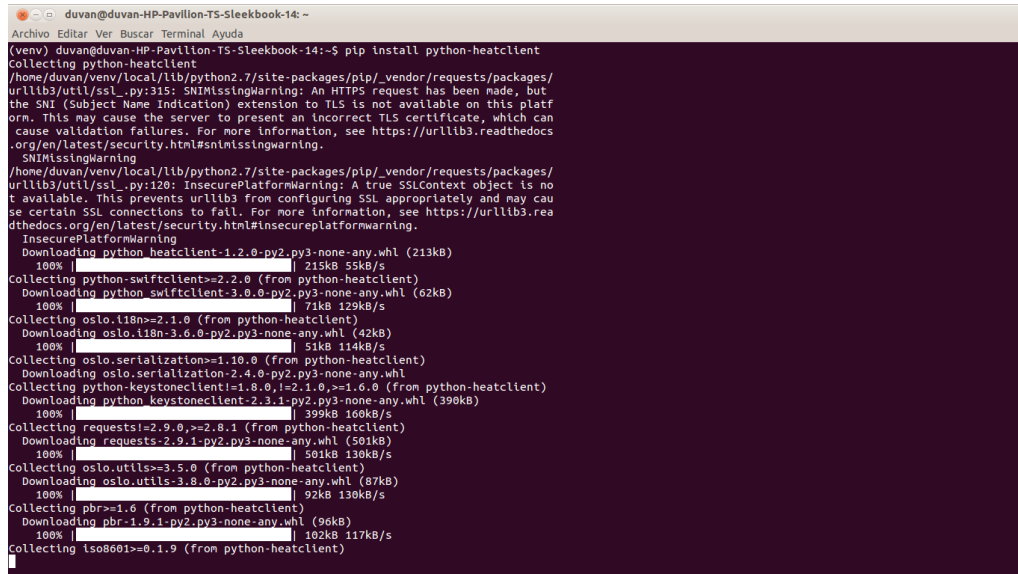
Realizar despliegues de infraestructura utilizando el lenguaje de orquestación de OpenStack. En particular, realizar un despliegue multi-instancia cuyos servicios deben colaborar.

### 3. ACTIVIDADES

Antes de comenzar con el taller es necesaria la instalación de algunos elementos antes de iniciar, se resume en el siguiente código para ubuntu 14.04 lts

```
1      sudo apt-get -y install python-pip python-dev
2      sudo pip2 install virtualenv
3      nano creds
4      export OS_TENANT_ID=0689b2eb69a24f5eafccdbd88a20c8a7
5      export OS_TENANT_NAME=cloud1
6      export OS_PROJECT_NAME=cloud1
7      export OS_USERNAME=grupo1
8      export OS_PASSWORD=1grupo
9      export OS_AUTH_URL="http://10.20.230.15:5000/v2.0"
10     export OS_REGION_NAME=regionOne
11     source creds
12     virtualenv venv
13     source venv/bin/activate
14     (venv) $ pip install python-heatclient
15     (venv) $ openstack stack list
```

A continuación se observa el la ejecución en consola.

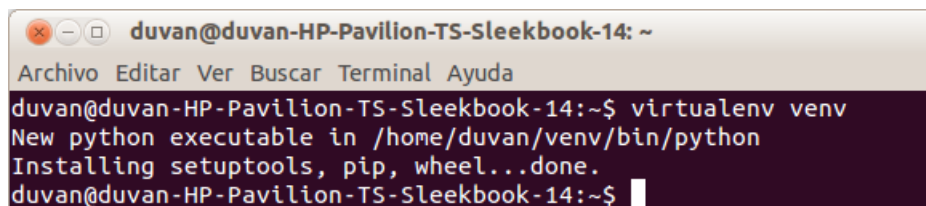


```

duvan@duvan-HP-Pavilion-TS-Sleekbook-14: ~
Archivo Editar Ver Buscar Terminal Ayuda
(venv) duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~$ pip install python-heatclient
Collecting python-heatclient
  Downloading python-heatclient-1.2.0-py2.py3-none-any.whl (213kB)
    100% |#####| 215kB 55kB/s
Collecting python-swiftclient>=2.2.0 (from python-heatclient)
  Downloading python-swiftclient-3.0.0-py2.py3-none-any.whl (62kB)
    100% |#####| 71kB 129kB/s
Collecting oslo.i18n>=2.1.0 (from python-heatclient)
  Downloading oslo.i18n-3.6.0-py2.py3-none-any.whl (42kB)
    100% |#####| 51kB 114kB/s
Collecting oslo.serialization>=1.10.0 (from python-heatclient)
  Downloading oslo.serialization-2.4.0-py2.py3-none-any.whl
Collecting python-keystoneclient>=1.8.0,!=2.1.0,!=1.6.0 (from python-heatclient)
  Downloading python-keystoneclient-2.3.1-py2.py3-none-any.whl (390kB)
    100% |#####| 399kB 160kB/s
Collecting requests>=2.9.0,!=2.8.1 (from python-heatclient)
  Downloading requests-2.9.1-py2.py3-none-any.whl (501kB)
    100% |#####| 501kB 130kB/s
Collecting oslo.utils>=3.5.0 (from python-heatclient)
  Downloading oslo.utils-3.8.0-py2.py3-none-any.whl (87kB)
    100% |#####| 92kB 130kB/s
Collecting pbr>=1.6 (from python-heatclient)
  Downloading pbr-1.9.1-py2.py3-none-any.whl (96kB)
    100% |#####| 102kB 117kB/s
Collecting iso8601>=0.1.9 (from python-heatclient)

```

Figura 1: Preparación del entorno



```

duvan@duvan-HP-Pavilion-TS-Sleekbook-14: ~
Archivo Editar Ver Buscar Terminal Ayuda
duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~$ virtualenv venv
New python executable in /home/duvan/venv/bin/python
Installing setuptools, pip, wheel...done.
duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~$

```

Figura 2: Preparación del entorno

1. Crear un directorio “wordpress-openstack”, y dentro de ese directorio, un sub-directorio “lib”.

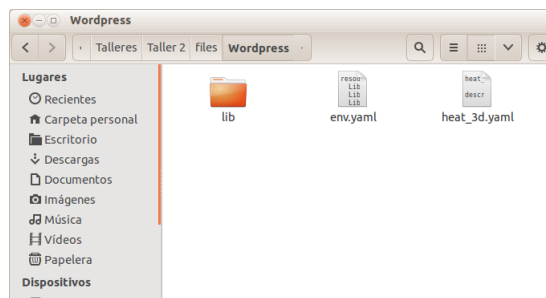
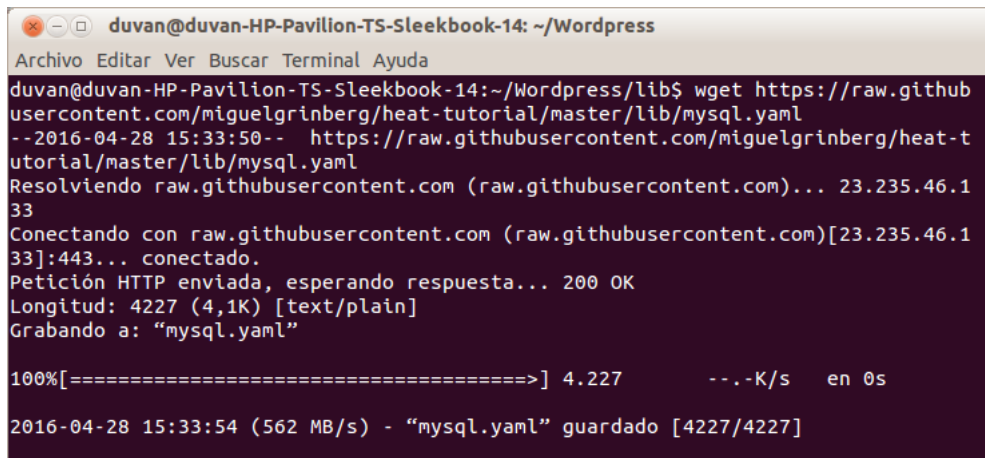


Figura 3: Directorio creado

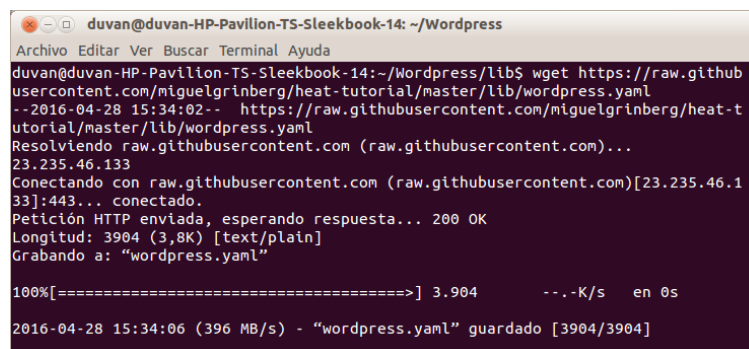
2. Descargar los archivos mysql.yaml , wordpress.yaml, private\_network.yaml y floating\_ip.yaml en el directorio “lib”.



```
duvan@duvan-HP-Pavilion-TS-Sleekbook-14: ~/Wordpress
Archivo Editar Ver Buscar Terminal Ayuda
duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~/Wordpress/lib$ wget https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/mysql.yaml
--2016-04-28 15:33:50-- https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/mysql.yaml
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com)... 23.235.46.133
Conectando con raw.githubusercontent.com (raw.githubusercontent.com)[23.235.46.133]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 4227 (4,1K) [text/plain]
Grabando a: "mysql.yaml"

100%[=====] 4.227 --.-K/s en 0s
2016-04-28 15:33:54 (562 MB/s) - "mysql.yaml" guardado [4227/4227]
```

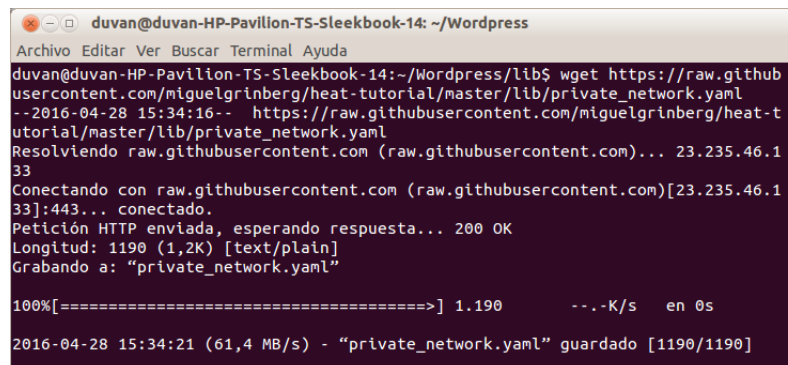
Figura 4: Descarga de archivos



```
duvan@duvan-HP-Pavilion-TS-Sleekbook-14: ~/Wordpress
Archivo Editar Ver Buscar Terminal Ayuda
duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~/Wordpress/lib$ wget https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/wordpress.yaml
--2016-04-28 15:34:02-- https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/wordpress.yaml
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com)... 23.235.46.133
Conectando con raw.githubusercontent.com (raw.githubusercontent.com)[23.235.46.133]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 3904 (3,8K) [text/plain]
Grabando a: "wordpress.yaml"

100%[=====] 3.904 --.-K/s en 0s
2016-04-28 15:34:06 (396 MB/s) - "wordpress.yaml" guardado [3904/3904]
```

Figura 5: Descarga de archivos



```
duvan@duvan-HP-Pavilion-TS-Sleekbook-14: ~/Wordpress
Archivo Editar Ver Buscar Terminal Ayuda
duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~/Wordpress/lib$ wget https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/private_network.yaml
--2016-04-28 15:34:16-- https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/private_network.yaml
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com)... 23.235.46.133
Conectando con raw.githubusercontent.com (raw.githubusercontent.com)[23.235.46.133]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1190 (1,2K) [text/plain]
Grabando a: "private_network.yaml"

100%[=====] 1.190 --.-K/s en 0s
2016-04-28 15:34:21 (61,4 MB/s) - "private_network.yaml" guardado [1190/1190]
```

Figura 6: Descarga de archivos

```

duvan@duvan-HP-Pavilion-TS-Sleekbook-14: ~/Wordpress
Archivo Editar Ver Buscar Terminal Ayuda
duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~/Wordpress/lib$ wget https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/floating_ip.yaml
--2016-04-28 15:34:27-- https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/floating_ip.yaml
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com)... 23.235.46.133
Conectando con raw.githubusercontent.com (raw.githubusercontent.com)[23.235.46.133]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 836 [text/plain]
Grabando a: "floating_ip.yaml"

100%[=====] 836 --.-K/s en 0s

2016-04-28 15:34:32 (61,3 MB/s) - "floating_ip.yaml" guardado [836/836]

```

Figura 7: Descarga de archivos

```

duvan@duvan-HP-Pavilion-TS-Sleekbook-14: ~/Wordpress
GNU nano 2.2.6 Archivo: lib/private_network.yaml

heat_template_version: 2013-05-23

description: Template that creates a private network.

parameters:
  public_network:
    type: string
    label: Public network name or ID
    description: Public network with floating IP addresses.
    default: public
  cidr:
    type: string
    label: CIDR
    description: The CIDR of the private network.
    default: '10.10.10.0/24'
  dns:
    type: comma_delimited_list
    label: DNS nameservers
    description: Comma separated list of DNS nameservers for the private network.
    default: '8.8.8.8'

resources:
  private_network:
    type: OS::Neutron::Net

```

Figura 8: Descarga de archivos

3. Editar el archivo `lib/private_network.yaml` de forma que tenga el siguiente contenido.

```
1 heat_template_version: 2013-05-23

3 description: Template that creates a private network.

5 parameters:
  public_network:
7   type: string
   label: Public network name or ID
9   description: Public network with floating IP addresses.
   default: public
11  cidr:
   type: string
13   label: CIDR
   description: The CIDR of the private network.
15   default: '10.10.10.0/24'
   dns:
17   type: comma_delimited_list
   label: DNS nameservers
19   description: Comma separated list of DNS nameservers for the private network.
   default: '8.8.8.8'

21 resources:
23   private_network:
   type: OS::Neutron::Net

25   private_subnet:
27   type: OS::Neutron::Subnet
   properties:
29   network_id: { get_resource: private_network }
   cidr: { get_param: cidr }
31   dns_nameservers: { get_param: dns }

33   router:
   type: OS::Neutron::Router
35   properties:
   external_gateway_info:
37   network: { get_param: public_network }

39   router-interface:
   type: OS::Neutron::RouterInterface
41   properties:
   router_id: { get_resource: router }
43   subnet: { get_resource: private_subnet }

45 outputs:
   name:
47   description: The private network.
   value: { get_attr: [private_network, name] }
```

Algo interesante del enfoque planteado en este laboratorio es utilizar las plantillas previamente definidas como *cajas negras*, funcionalidades ya probadas de quienes únicamente interesa sus entradas y sus salidas. Este enfoque se denomina “*plantillas anidadas*” y provee una manera más extensible de depurar los diferentes despliegues orquestados.

4. Descargar el archivo `heat_3d.yaml` y se ubica en el directorio “`wordpress-openstack`”.

```

heat_template_version: 2013-05-23

description: Template that installs a wordpress server and supporting MySQL database running on separate servers

parameters:
  image:
    type: string
    label: Image name or ID
    description: Image to be used for server. Please use an Ubuntu based image.
    default: ubuntu-server-14.04-CECAD-r20141201
  flavor:
    type: string
    label: Flavor
    description: Type of instance (flavor) to be used on the compute instance.
    default: m1.small
  public_network:
    type: string
    label: Public network name or ID
    description: Public network to attach server to.
    default: ext-net-doctorado
  private_cidr:
    type: string
    label: private network CIDR
    description: Private network CIDR.
    default: '192.168.200.0/24'

resources:
  my_keypair:
    type: OS::Nova::KeyPair
    properties:
      name: cloudapps
      save_private_key: True

```

Figura 9: Descarga de archivo heat\_3d.yaml

5. Realizar las modificaciones al archivo heat\_3d.yaml, de forma que luzca como sigue a continuación:

```

1 heat_template_version: 2013-05-23
2
3 description: Template that installs a wordpress server and
4   supporting MySQL database running on separate servers
5
6 parameters:
7   image:
8     type: string
9     label: Image name or ID
10    description: Image to be used for server. Please use an Ubuntu based image.
11    default: ubuntu-server-14.04-CECAD-r20141201
12   flavor:
13     type: string
14     label: Flavor
15     description: Type of instance (flavor) to be used on the compute instance.
16     default: m1.small
17   public_network:
18     type: string
19     label: Public network name or ID
20     description: Public network to attach server to.
21     default: ext-net-doctorado
22   private_cidr:
23     type: string
24     label: private network CIDR
25     description: Private network CIDR.
26     default: '192.168.200.0/24'
27
28 resources:
29
30   my_keypair:
31     type: OS::Nova::KeyPair
32     properties:
33       name: cloudapps
34       save_private_key: True
35
36

```

```

network:
38 type: Lib::CECAD::PrivateNetwork
   properties:
40   public_network: { get_param: public_network }
   cidr: {get_param: private_cidr}
42
mysql:
44 type: Lib::CECAD::MySQL
   properties:
46   image: { get_param: image }
   flavor: { get_param: flavor }
48   key: { get_resource: my_keypair }
   private_network: { get_attr: [network, name] }
50   database_name: wordpress
   database_user: wordpress_user
52
wordpress:
54 type: Lib::CECAD::Wordpress
   properties:
56   image: { get_param: image }
   flavor: { get_param: flavor }
58   key: { get_resource: my_keypair }
   private_network: { get_attr: [network, name] }
60   mysql_server: { get_attr: [mysql, ip] }
   database_name: wordpress
62   database_user: wordpress_user
   database_password: { get_attr: [mysql, database_password] }
64
floating_ip:
66 type: Lib::CECAD::FloatingIP
   properties:
68   port: { get_attr: [wordpress, port] }
   public_network: { get_param: public_network }
70
outputs:
72 ip:
   description: The public IP address to access Wordpress.
74   value: { get_attr: [floating_ip, ip] }

```

6. Descargar el archivo env.yaml en el directorio “wordpress-openstack” y editarlo teniendo en cuenta la direcciones absolutas de los archivos \*.yaml.

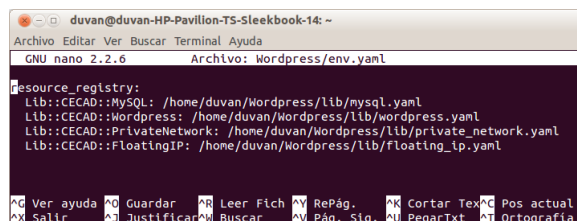


Figura 10: Descarga de archivo env.yaml

7. Lanzar la pila y acceder a la consola de Wordpress en la dirección de IP flotante asignada por la orquestación. Se requiere para trabajo desde heat (en equipo en red) instalar algunos paquetes y emplear un fichero de credenciales para acceder al servidor, el fichero creds contiene lo siguiente.

```

duvan@duvan-HP-Pavilion-TS-Sleekbook-14: ~
GNU nano 2.2.6 Archivo: creds Modificado

export OS_TENANT_ID=0689b2eb69a24f5eafccdbd88a20c8a7
export OS_TENANT_NAME=cloud1
export OS_PROJECT_NAME=cloud1

export OS_USERNAME=grupo1
export OS_PASSWORD=grupo1
export OS_AUTH_URL="http://10.20.230.15:5000/v2.0"

export OS_REGION_NAME=regionOne

^G Ver ayuda ^O Guardar ^R Leer Fich ^Y RePág. ^K Cortar Text ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág. Sig. ^U PegarTxt ^T Ortografía

```

Figura 11: acceso a la consola de Wordpress en la dirección de IP flotante asignada

Por otra parte, es necesario modificar el fichero hosts e incluir algunos elementos, dependiendo de los requerimientos de cada orquestación.

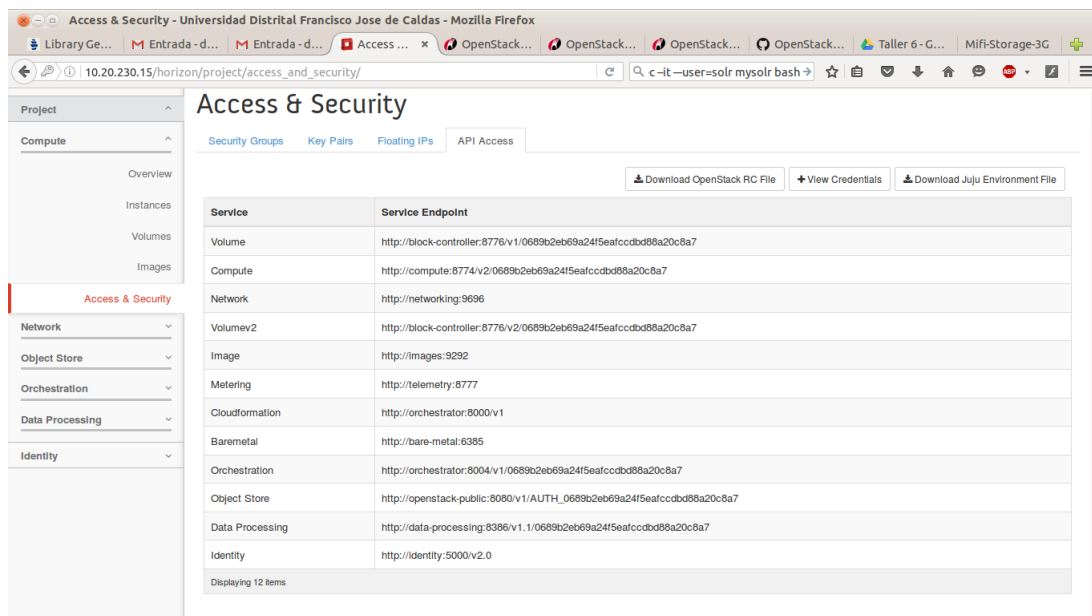


Figura 12: modificación el fichero hosts

```

duvan@duvan-HP-Pavilion-TS-Sleekbook-14: ~
GNU nano 2.2.6 Archivo: hosts

127.0.0.1    localhost
127.0.1.1    duvan-HP-Pavilion-TS-Sleekbook-14

# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters

10.20.230.15 orchestrator

^G Ver ayuda ^O Guardar ^R Leer Fich ^Y RePág. ^K Cortar Te ^C Pos actual
^X Salir ^J Justifica ^W Buscar ^V Pág. Sig. ^U PegarTxt ^T Ortografía

```

Figura 13: fichero hosts

Para ejecutar la prueba (teniendo la carpeta creada en el directorio del usuario), se realiza el montaje de las credenciales con la función “Source”, para el ejemplo a continuación se verifica que no hay stacks en la nube:



```

duvan@duvan-HP-Pavilion-TS-Sleekbook-14: ~
Archivo Editar Ver Buscar Terminal Ayuda
duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~$ virtualenv venv
New python executable in /home/duvan/venv/bin/python
Installing setuptools, pip, wheel...done.
duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~$ source venv/bin/activate
(venv) duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~$ source creds
(venv) duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~$ heat stack-list

```

Figura 14: Espacio de trabajo y variables de entorno

Ahora es posible lanzar la orquestación desde heat, de la siguiente manera:

```

(venv) duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~$ heat stack-create wordpress -f Wordpress/heat_3d.yaml -e Wordpress/env.yaml
+-----+-----+-----+-----+
| id | stack_name | stack_status | creation_time |
+-----+-----+-----+-----+
| 5fcd440-59af-4a67-ba83-e0eb6605425c | wordpress | CREATE_IN_PROGRESS | 2016-04-28T21:33:19Z |
+-----+-----+-----+-----+
(venv) duvan@duvan-HP-Pavilion-TS-Sleekbook-14:~$

```

Figura 15: Lanzamiento de la orquestación

Por medio de la interfaz de horizon verificamos la orquestación:

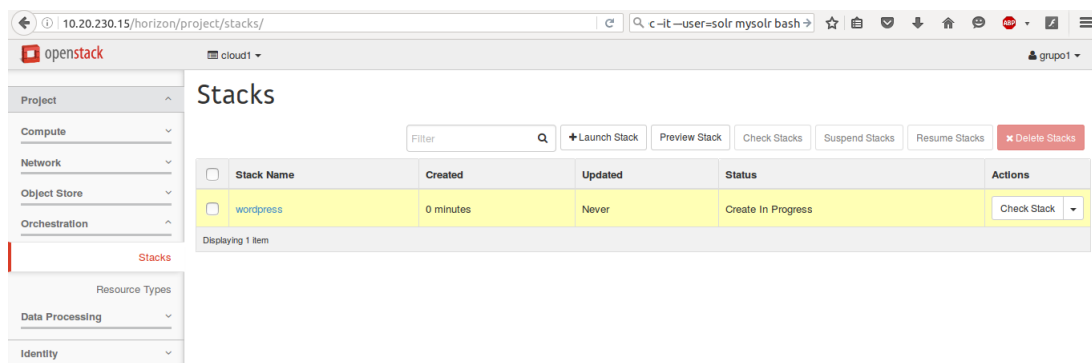


Figura 16: Se verifica la orquetación

Observamos los detalles de la orquestación

## 4. BIBLIOGRAFIA

- <http://vmartinezdelacruz.com/en-pocas-palabras-como-funciona-openstack/>
- <https://www.openstack.org/>
- <https://www.redhat.com/es/insights/openstack>

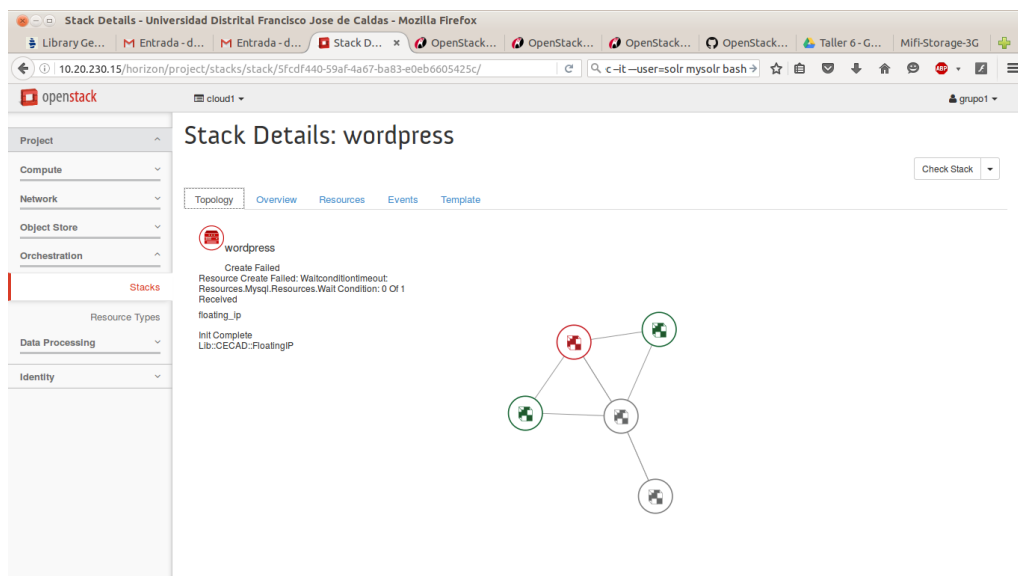


Figura 17: Detalles de la orquestación