

TALLER 1

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

1. INTRODUCCIÓN

OpenStack es una solución de cloud computing del tipo IaaS de código abierto, su misión es proveer una solución flexible tanto para nubes públicas como privadas, sean estas de cualquier tamaño, y para esto se consideran dos requerimientos básicos: las nubes deben ser simples de implementar y masivamente escalables.(tomado de OpenStack)

El software OpenStack controla grandes piscinas de cómputo (procesamiento), almacenamiento, redes y recursos a través de un dashboard, gestionado a través de un panel de control o a través de la API de OpenStack, trabaja con tecnologías empresariales y de código abierto.(tomado de ¿Cómo funciona OpenStack?)

OpenStack y Ubuntu:

- OpenStack es la principal plataforma de nube abierta.
- Ubuntu es el sistema operativo más popular del mundo para OpenStack.
- Ubuntu OpenStack es la nube más estable y manejable del mundo.
- OpenStack-Autopilot (instalador automático de OpenStack en Ubuntu) es la manera más rápida y confiable para construir su propia nube OpenStack.

tomado de Ubuntu OpenStack

2. OBJETIVO

Realizar despliegues de infraestructura sencillos utilizando el lenguaje de orquestación de OpenStack.

3. ACTIVIDADES

1. Crear un archivo denominado “**router.yaml**” con el siguiente contenido. Seguir las indicaciones del instructor para ejecutar esta plantilla.

```
1 heat_template_version: 2013-05-23

3 description: This template deploys a router with a port in the public
  interface

5 parameters:

7 public_network:
  type: string
9 label: Public network name or ID
  description: Public network with floating IP addresses.
11 default: ext-net-doctorado

13 resources:

15 router:
  type: OS::Neutron::Router
```

```
17 properties:  
  external_gateway_info:  
19 network: { get_param: public_network }
```

2. Verificar la correcta creación del router.

Estado Inicial de openStack:

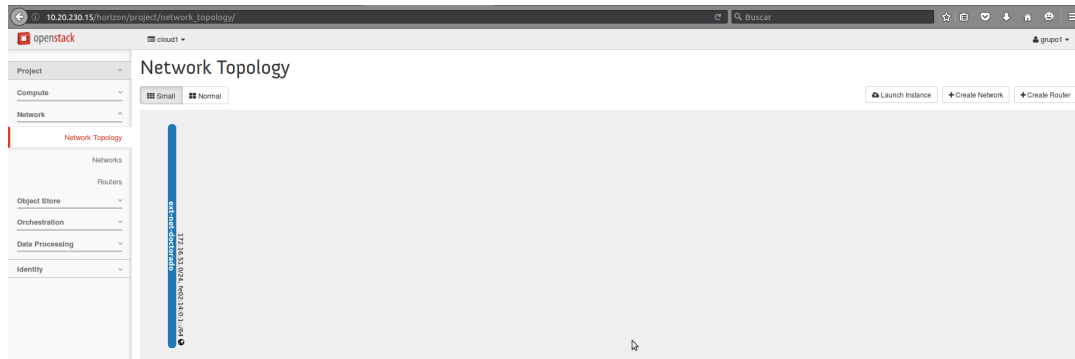


Figura 1: Estado inicial del OpenStack

Se procede a lanzar el Stack:

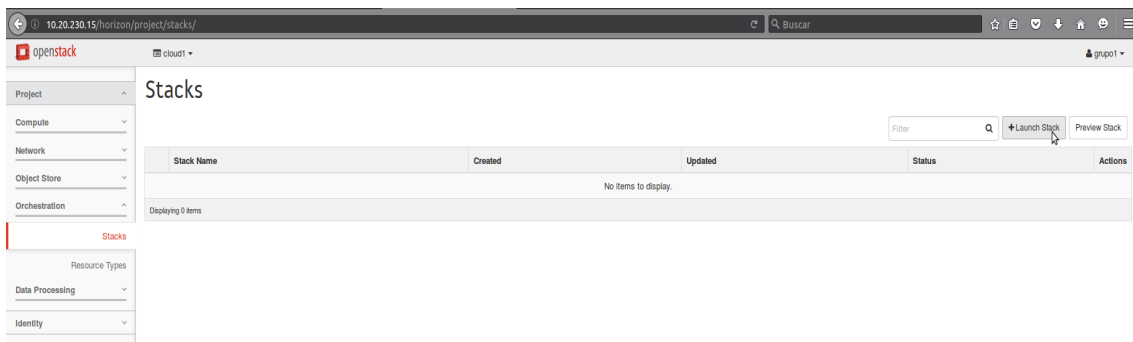


Figura 2: lanzamiento del Stack

Se lanza el fichero `router.yaml`:

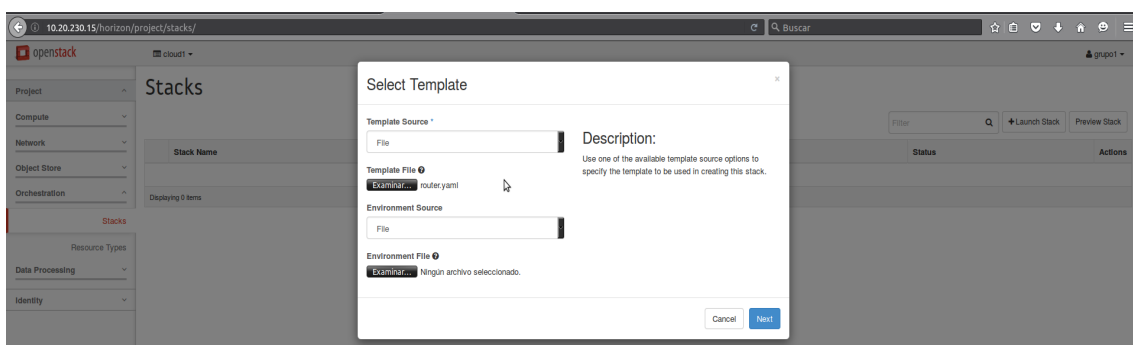
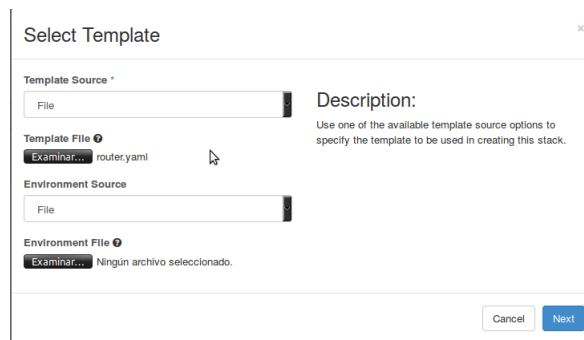


Figura 3: Lanzamiento del fichero



The 'Select Template' dialog box contains the following fields and controls:

- Template Source:** A dropdown menu with 'File' selected.
- Template File:** A text input field containing 'router.yaml'. Below it is a button labeled 'Examinar...'.
- Environment Source:** A dropdown menu with 'File' selected.
- Environment File:** A text input field containing 'Ningún archivo seleccionado.' Below it is a button labeled 'Examinar...'.
- Description:** A text area with the text: 'Use one of the available template source options to specify the template to be used in creating this stack.'
- Buttons:** 'Cancel' and 'Next' buttons at the bottom right.

Figura 4: ventana para lanzar archivos

Se verifica la creación del Stack:

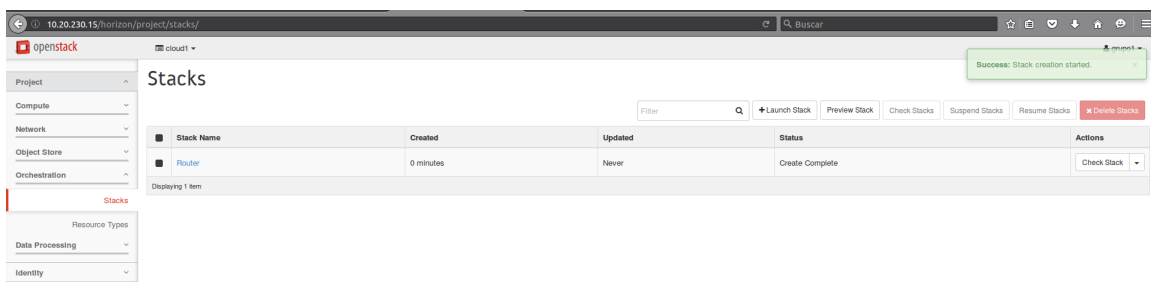


Figura 5: Creación del Stack

Verificación de Router creado:

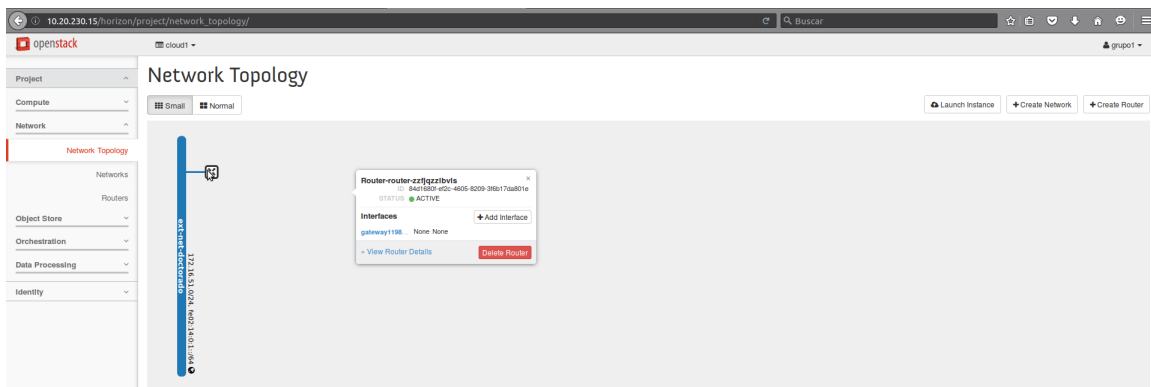


Figura 6: Topología donde se observa el router creado

Detalles del Router:

Router Details

Overview Interfaces	
Name	Router-router-zzfjgzzlbvls
ID	84d1680f-ef2c-4605-8209-3f6b17da801e
Project ID	0609b2eb69a24f5eafccdbd88a20c8a7
Status	ACTIVE
Admin State	UP
External Gateway	
Network Name	ext-net-doctorado
Network ID	1198bbb6-cf96-49b1-83ab-95d69abba3a7
Subnet ID	2fd4a7d-0908-4768-a911-d337b5827793
External Fixed IPs	IP Address 172.16.51.145 Subnet ID 3459b183-ea91-40f2-962a-556616744393 IP Address fe02:14:0:1::55
SNAT	Enabled

Figura 7: Descripción de los detalles del router

3. Crear un archivo denominado “**network.yaml**” con el siguiente contenido y ejecutar la plantilla.

```

1 heat_template_version: 2013-05-23
2
3 description: This template deploys a router with a port in the public
4             interface
5
6 parameters:
7
8   private_network_cidr:
9     type: string
10    label: Private network CIDR
11    description: Private Network CIDR
12    default: 192.168.200.0/24
13
14 resources:
15
16   private_network:
17     type: OS::Neutron::Net
18
19   private_subnet:
20     type: OS::Neutron::Subnet
21     properties:
22       network_id: { get_resource: private_network }
23       cidr: {get_param: private_network_cidr}
24       dns_nameservers:
25         - 8.8.8.8

```

Se comienza lanzando un Stack:

Figura 8: Ventana para lanzar el Stack

Se verifica la creación correcto del Stack:

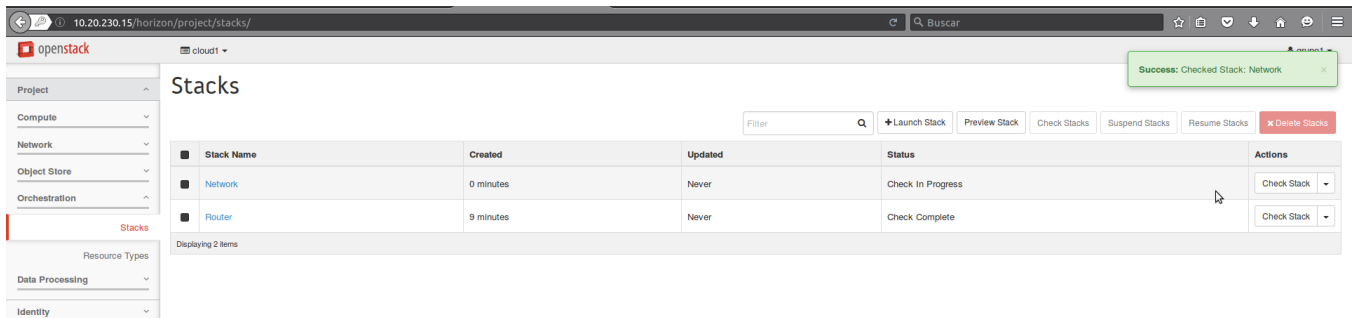


Figura 9: Verificación de la creación del Stack

Nos dirigimos a redes (Networks) para verificar la correcta creación de la red:

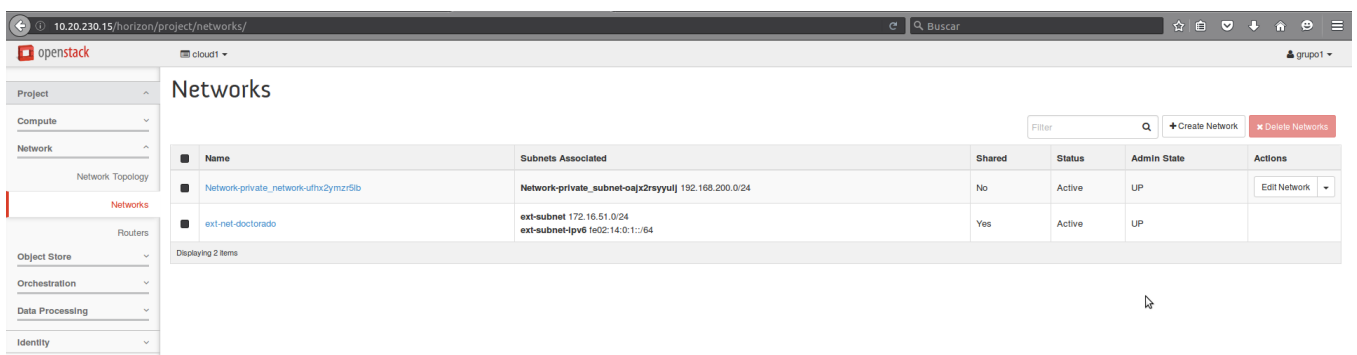


Figura 10: Verificación de la creación de la red

4. Finalmente en topología de red vemos el router y la red creadas anteriormente:

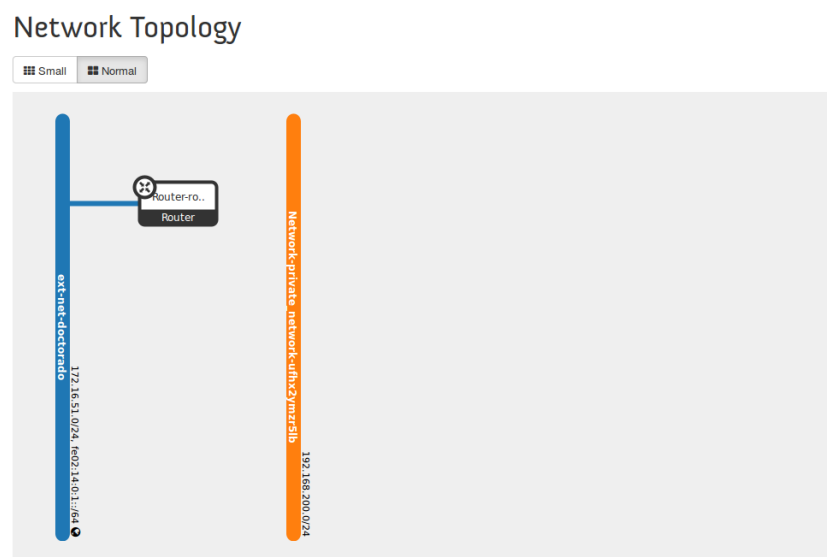


Figura 11: Topología de la red

5. Eliminar las pilas previamente creadas.
6. Crear un archivo denominar “complete-network.yaml” con el siguiente contenido y ejecutar la plantilla. En este paso, se va a crear un router, una red privada, y se le va a asignar un puerto al router dentro de esa red:

```

2      heat_template_version: 2013-05-23

4      description: This template deploys a router with a port in the public
           interface

6      parameters:

8      public_network:
          type: string
10     label: Public network name or ID
          description: Public network with floating IP addresses.
12     default: ext-net-doctorado

14     private_network_cidr:
          type: string
16     label: Private network CIDR
          description: Private Network CIDR
18     default: 192.168.200.0/24

20     resources:

22     router:
          type: OS::Neutron::Router
24     properties:
          external_gateway_info:
26     network: { get_param: public_network }

28     private_network:
          type: OS::Neutron::Net

30     private_subnet:
          type: OS::Neutron::Subnet
32     properties:
          network_id: { get_resource: private_network }
          cidr: {get_param: private_network_cidr}
34     dns_nameservers:
          - 8.8.8.8

38     router-interface:
          type: OS::Neutron::RouterInterface
40     properties:
          router_id: { get_resource: router }
42     subnet: { get_resource: private_subnet }

```

lanzamos el Stack:

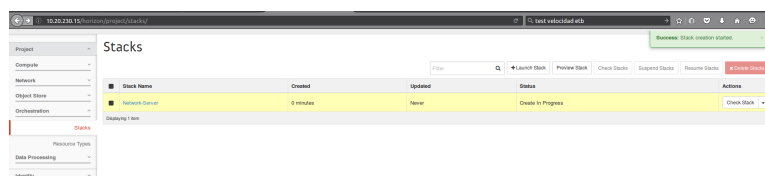


Figura 12: Lanzamiento del Stack

Se verifica la topología de la red:

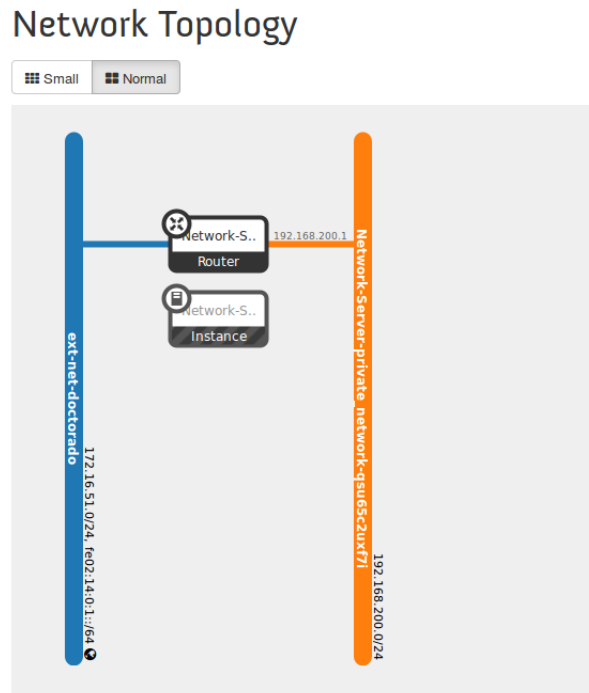


Figura 13: Topología de la red

Una vez la infraestructura de red, el siguiente paso es crear servidores. Inicialmente, se desplegará únicamente el servidor con su respectivo grupo de seguridad. Posteriormente se le configurará en la plantilla el software a instalar y se le asignará un IP flotante.

7. Eliminar las pilas previamente creadas.
8. Crear un archivo denominar “**network-server.yaml**” con el siguiente contenido y ejecutar la plantilla. En este paso, se va a crear un router, una red privada, se le va a asignar un puerto al router dentro de esa red, y se va a lanzar una instancia con un grupo de seguridad denominado “**web_server_security_group**” y una llave “**cloudapps**” .

```

1 heat_template_version: 2013-05-23
2
3 description: This template deploys a router, a private network and a single
4             basic server with a security group.
5
6 parameters:
7
8   public_network:
9     type: string
10    label: Public network name or ID
11    description: Public network with floating IP addresses.
12    default: ext-net-doctorado
13
14   private_network_cidr:
15     type: string
16     label: Private network CIDR
17     description: Private Network CIDR
18     default: 192.168.200.0/24

```

```
image:
20 type: string
   label: Image name or ID
22 description: Image to be used for compute instance
   default: Ubuntu-Server-14.04-CECAD-r20141201
24
flavor:
26 type: string
   label: Flavor
28 description: Type of instance (flavor) to be used
   default: m1.small
30
resources:
32
  router:
34 type: OS::Neutron::Router
   properties:
36   external_gateway_info:
       network: { get_param: public_network }
38
   private_network:
40 type: OS::Neutron::Net
42
   private_subnet:
44 type: OS::Neutron::Subnet
   properties:
46   network_id: { get_resource: private_network }
       cidr: {get_param: private_network_cidr}
       dns_nameservers:
48 - 8.8.8.8
50
  router-interface:
52 type: OS::Neutron::RouterInterface
   properties:
54   router_id: { get_resource: router }
       subnet: { get_resource: private_subnet }
56
  web_server_security_group:
58 type: OS::Neutron::SecurityGroup
   properties:
60   name: web_server_security_group
       rules:
62 - protocol: tcp
       port_range_min: 80
       port_range_max: 80
64 - protocol: tcp
       port_range_min: 443
       port_range_max: 443
66 - protocol: icmp
68 - protocol: tcp
       port_range_min: 22
70 port_range_max: 22
72
  my_keypair:
74 type: OS::Nova::KeyPair
   properties:
76   name: cloudapps
       save_private_key: True
78
  my_instance:
```



```

type: OS::Nova::Server
80 properties:
  image: { get_param: image }
82 flavor: { get_param: flavor }
  key_name: { get_resource: my_keypair }
84 networks:
  - network: { get_resource: private_network }
86 security_groups:
  - { get_resource: web_server_security_group }
88 user_data: |
  #!/bin/sh
90 sudo apt-get -y update sudo apt-get -y install apache2 sudo service apache2
  restart
  user_data_format: RAW
92
94 outputs:
  my_instance_name:
    description: Name of the instance
    value: { get_attr: [my_instance, name] }
96 my_instance_ip:
    description: IP address of the instance
    value: { get_attr: [my_instance, first_address] }
98

```

Lanzamiento del Stack:

Figura 14: Lanzamiento del Stack

Stack Name	Created	Updated	Status	Actions
Network-Server	0 minutes	Never	Create In Progress	Check Stack

Figura 15: Lanzamiento del Stack

Se verifica la creación del Stack en la topología de la red

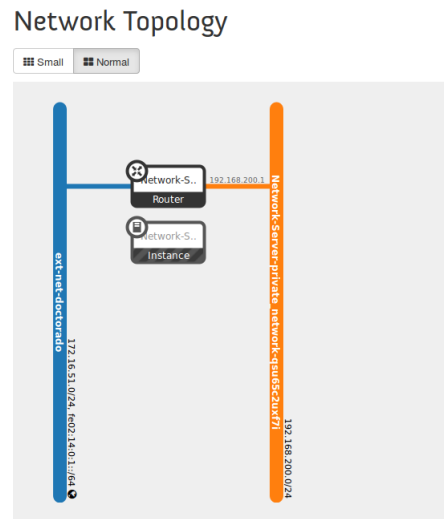


Figura 16: Verificación de la creación del Stack

Topología de la red:

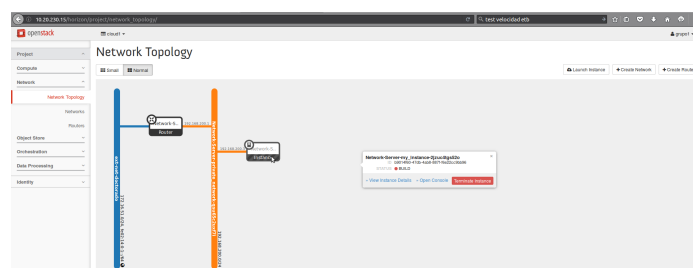


Figura 17: Topología de la red

Después de creada la instancia se abre la consola:

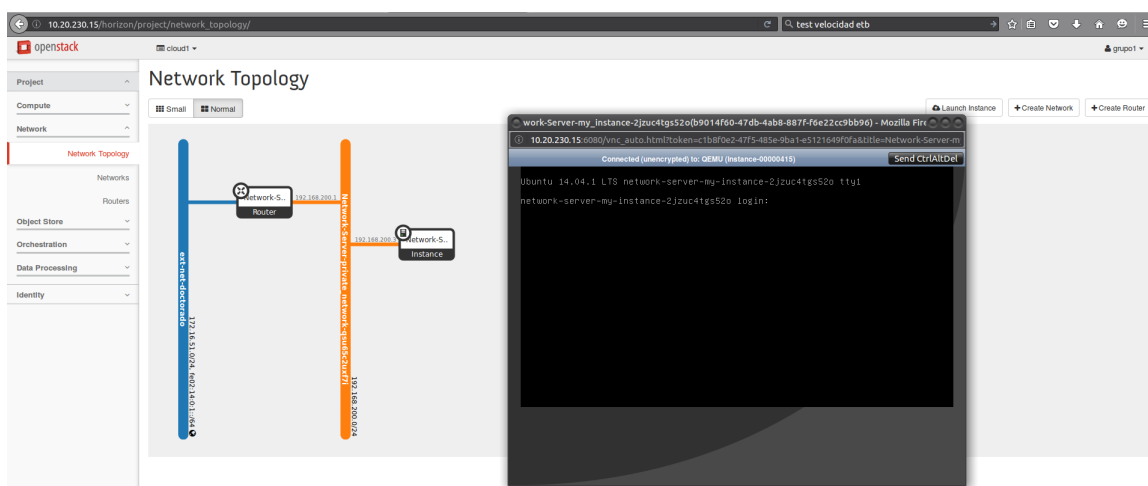


Figura 18: Apertura de la consola en la instancia

9. Se le Asigna una IP Flotante a la instancia: 172.18.51.148:

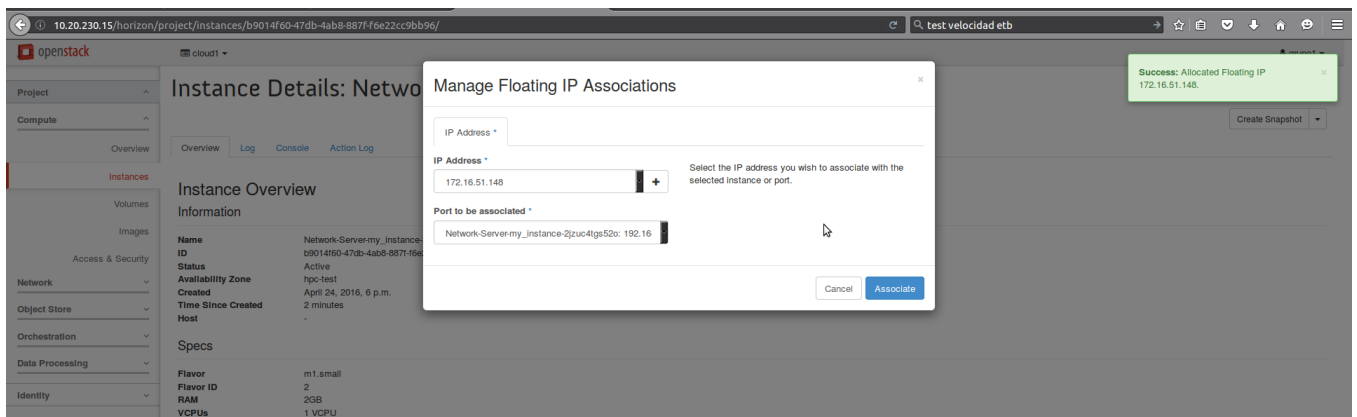


Figura 19: Asignación de una IP flotante

A continuación se muestran los resultados de la Instancia

Instance Details: Network-Server-my_instance-2jzuc4tgs520

Overview Log Console Action Log

Instance Overview

Information

Name	Network-Server-my_instance-2jzuc4tgs520
ID	b9014f60-47db-4ab8-887f-f6e22cc9bb96
Status	Active
Availability Zone	hpc-test
Created	April 24, 2016, 6 p.m.
Time Since Created	5 minutes
Host	-

Specs

Flavor	m1.small
Flavor ID	2
RAM	2GB
VCPUs	1 VCPU
Disk	20GB

IP Addresses

Network-Server-Private...	192.168.200.3, 172.16.51.148
---------------------------	------------------------------

Security Groups

web_server_security_gr...	ALLOW IPv4 443/tcp from 0.0.0.0/0 ALLOW IPv4 22/tcp from 0.0.0.0/0 ALLOW IPv4 80/tcp from 0.0.0.0/0 ALLOW IPv4 to 0.0.0.0/0 ALLOW IPv6 to ::0 ALLOW IPv4 icmp from 0.0.0.0/0
---------------------------	---

Metadata

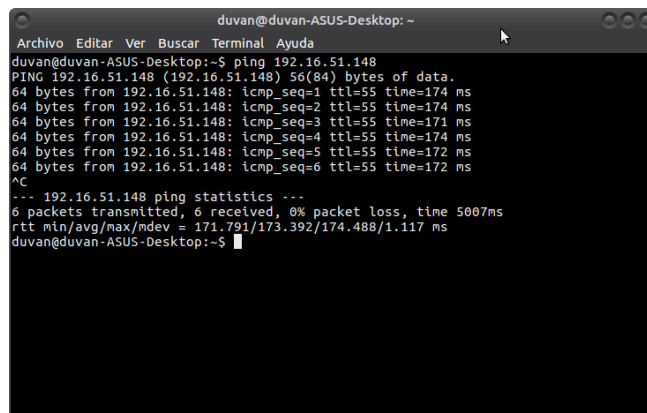
Key Name	cloudapps
Image Name	Ubuntu-Server-14.04-CECAD-r20141201

Volumes Attached

Volume	No volumes attached.
--------	----------------------

Figura 20: Resultados de la Instancia

10. Se realiza un Ping hacia la IP Flotante de la Instancia creada:



```
duvan@duvan-ASUS-Desktop: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
duvan@duvan-ASUS-Desktop:~$ ping 192.16.51.148  
PING 192.16.51.148 (192.16.51.148) 56(84) bytes of data:  
64 bytes from 192.16.51.148: icmp_seq=1 ttl=55 time=174 ms  
64 bytes from 192.16.51.148: icmp_seq=2 ttl=55 time=174 ms  
64 bytes from 192.16.51.148: icmp_seq=3 ttl=55 time=171 ms  
64 bytes from 192.16.51.148: icmp_seq=4 ttl=55 time=174 ms  
64 bytes from 192.16.51.148: icmp_seq=5 ttl=55 time=172 ms  
64 bytes from 192.16.51.148: icmp_seq=6 ttl=55 time=172 ms  
^C  
--- 192.16.51.148 ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, time 5007ms  
rtt min/avg/max/mdev = 171.791/173.392/174.488/1.117 ms  
duvan@duvan-ASUS-Desktop:~$
```

Figura 21: Verificación del funcionamiento mediante un ping

4. Bibliografía

- <http://vmartinezdelacruz.com/en-pocas-palabras-como-funciona-openstack/>
- <https://www.openstack.org/>
- <http://www.ubuntu.com/cloud/openstack>