



UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA

MESTRÍA EN CIENCIAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

ASIGNATURA: Programación para la Web y la Nube

PROFESOR: José Nelson Pérez Castillo

### Taller # 1: Orquestación en OpenStack (I)

#### OBJETIVO

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

#### ACTIVIDADES

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

```
heat_template_version: 2013-05-23
description: This template deploys a router with a port in the public interface
parameters:
  public_network:
    type: string
    label: Public network name or ID
    description: Public network with floating IP addresses.
    default: ext-net-doctorado
resources:
  router:
    type: OS::Neutron::Router
    properties:
      external_gateway_info:
        network: { get_param: public_network }
```

2. Verificar la correcta creación del *router*.
3. Crear un archivo denominado “network.yaml” con el siguiente contenido y ejecutar la plantilla.

```
heat_template_version: 2013-05-23
description: This template deploys a router with a port in the public interface
parameters:
  private_network_cidr:
    type: string
    label: Private network CIDR
    description: Private Network CIDR
    default: 192.168.200.0/24
resources:
  private_network:
    type: OS::Neutron::Net

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

4. Verificar la correcta creación 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.

```
heat_template_version: 2013-05-23
description: This template deploys a router with a port in the public interface
parameters:
  public_network:
    type: string
    label: Public network name or ID
    description: Public network with floating IP addresses.
    default: ext-net-doctorado

  private_network_cidr:
    type: string
    label: Private network CIDR
    description: Private Network CIDR
    default: 192.168.200.0/24
resources:
  router:
    type: OS::Neutron::Router
    properties:
      external_gateway_info:
        network: { get_param: public_network }

  private_network:
    type: OS::Neutron::Net

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

  router-interface:
    type: OS::Neutron::RouterInterface
    properties:
      router_id: { get_resource: router }
      subnet: { get_resource: private_subnet }
```

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” .

```
heat_template_version: 2013-05-23

description: This template deploys a router, a private network and a single basic server with a security group.

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

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

  image:
    type: string
    label: Image name or ID
    description: Image to be used for compute instance
    default: Ubuntu-Server-14.04-CECAD-r20141201

  flavor:
    type: string
    label: Flavor
    description: Type of instance (flavor) to be used
    default: m1.small

resources:
  router:
    type: OS::Neutron::Router
    properties:
      external_gateway_info:
        network: { get_param: public_network }

  private_network:
    type: OS::Neutron::Net

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

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

  web_server_security_group:
    type: OS::Neutron::SecurityGroup
    properties:
      name: web_server_security_group
      rules:
        - protocol: tcp
          port_range_min: 80
          port_range_max: 80
        - protocol: tcp
          port_range_min: 443
          port_range_max: 443
        - protocol: icmp
        - protocol: tcp
          port_range_min: 22
          port_range_max: 22
```

```

web_server_security_group:
  type: OS::Neutron::SecurityGroup
  properties:
    name: web_server_security_group
    rules:
      - protocol: tcp
        port_range_min: 80
        port_range_max: 80
      - protocol: tcp
        port_range_min: 443
        port_range_max: 443
      - protocol: icmp
      - protocol: tcp
        port_range_min: 22
        port_range_max: 22

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

my_instance:
  type: OS::Nova::Server
  properties:
    image: { get_param: image }
    flavor: { get_param: flavor }
    key_name: { get_resource: my_keypair }
    networks:
      - network: { get_resource: private_network }
    security_groups:
      - { get_resource: web_server_security_group }
    user_data: |
      #!/bin/sh
      sudo apt-get -y update && sudo apt-get -y install apache2 && sudo service apache2 restart
    user_data_format: RAW

outputs:
  my_instance_name:
    description: Name of the instance
    value: { get_attr: [my_instance, name] }
  my_instance_ip:
    description: IP address of the instance
    value: { get_attr: [my_instance, first_address] }

```

9. Finalmente, se le va a asignar una IP flotante a la instancia. Modificar el archivo “network-server.yaml” en la sección “resources” para que luzca de la siguiente forma:

```

web_server_security_group:
  type: OS::Neutron::SecurityGroup
  properties:
    name: web_server_security_group
    rules:
      - protocol: tcp
        port_range_min: 80
        port_range_max: 80
      - protocol: tcp
        port_range_min: 443
        port_range_max: 443
      - protocol: icmp
      - protocol: tcp
        port_range_min: 22
        port_range_max: 22

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

floating_ip:
  type: OS::Neutron::FloatingIP
  properties:
    floating_network: { get_param: public_network }

server_port:
  type: OS::Neutron::Port
  properties:
    network: { get_resource: private_network }
    security_groups:
      - { get_resource: web_server_security_group }

my_instance:
  type: OS::Nova::Server
  properties:
    image: { get_param: image }
    flavor: { get_param: flavor }
    key_name: { get_resource: my_keypair }
    availability_zone: hpc-test
    networks:
      - port: { get_resource: server_port }
    user_data: |
      #!/bin/sh
      sudo apt-get -y update && sudo apt-get -y install apache2 && sudo service apache2 restart
    user_data_format: RAW

floating_ip_assoc:
  type: OS::Neutron::FloatingIPAssociation
  properties:
    floatingip_id: { get_resource: floating_ip }
    port_id: { get_resource: server_port }

```

10. Esperar un breve tiempo y acceder a la página pre-establecida de Apache direccionando cualquier explorador a la dirección <http://<ip-flotante-instancia>>.

## HISTORIAL DE REVISIONES

Autor	Fecha	Comentarios
Rodolfo Cáliz Ospino rcaliz@gmail.com	2016/03/03	Versión inicial del documento