



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, Ph.D

Taller # 2: Orquestación en OpenStack (II)

OBJETIVO

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

ACTIVIDADES

NOTA: En este taller se van a realizar y validar las actividades planteadas en <https://developer.rackspace.com/blog/openstack-orchestration-in-depth-part-3-multi-instance-deployments/> . Todos los derechos pertenecen a sus autores originales.

En caso de no desear leer el artículo original, realizar las siguientes actividades:

1. Crear un directorio “wordpress-openstack”, y dentro de ese directorio, un sub-directorio “lib”.
2. Descargar los archivos <https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/mysql.yaml> , <https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/wordpress.yaml> , https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/private_network.yaml , y

https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/floating_ip.yaml en el directorio “lib”.

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

```
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

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

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

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

outputs:
  name:
    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 https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/heat_3d.yaml y ubicarlo en el directorio “wordpress-openstack”.
5. Realizar las modificaciones al archivo, de forma que luzca como sigue a continuación:

```

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

  network:
    type: Lib::CECAD::PrivateNetwork
    properties:
      public_network: { get_param: public_network }
      cidr: {get_param: private_cidr}

  mysql:
    type: Lib::CECAD::MySQL
    properties:
      image: { get_param: image }
      flavor: { get_param: flavor }
      key: { get_resource: my_keypair }
      private_network: { get_attr: [network, name] }
      database_name: wordpress
      database_user: wordpress_user

  wordpress:
    type: Lib::CECAD::wordpress
    properties:
      image: { get_param: image }
      flavor: { get_param: flavor }
      key: { get_resource: my_keypair }
      private_network: { get_attr: [network, name] }
      mysql_server: { get_attr: [mysql, ip] }
      database_name: wordpress
      database_user: wordpress_user
      database_password: { get_attr: [mysql, database_password] }

  floating_ip:
    type: Lib::CECAD::FloatingIP
    properties:
      port: { get_attr: [wordpress, port] }
      public_network: { get_param: public_network }

outputs:
  ip:
    description: The public IP address to access wordpress.
    value: { get_attr: [floating_ip, ip] }

```

6. Descargar el archivo <https://raw.githubusercontent.com/miguelgrinberg/heat-tutorial/master/lib/env.yaml> en el directorio "wordpress-openstack" y editarlo teniendo en cuenta la direcciones absolutas de los archivos *.yaml.
7. Lanzar la pila y acceder a la consola de Wordpress en la dirección de IP flotante asignada por la orquestación.

HISTORIAL DE REVISIONES

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