

Introducción a heat

Heat es el componente de Openstack encargado de la orquestación. Mediante un template podemos automatizar el despliegue de un escenario formado por distintos recursos (instancias, volúmenes, redes, routers, ...) y establecer las relaciones entre ellos.

La sintaxis estándar de los templates llamada HOT usa el lenguaje YAML, aunque también podríamos utilizar la sintaxis en JSON.

Estructura de un template

Pongamos un ejemplo sencillo:

```
heat_template_version: 2013-05-23

description: Simple template to deploy a single compute instance
parameters:
  network:
    type: string
description: Nombre de la red a utilizar
default: red_por_defecto

resources:
  my_instance:
    type: OS::Nova::Server
properties:
  key_name: clave_ssh
  image: Debian Jessie (stable)
  flavor: m1.small
  networks: [{get_param: network}]

outputs:
  instance_ip:
    description: IP address of the instance
value: { get_attr: [my_instance, first_address] }
```

- Versión del template: Aquí se indica la versión del template utilizado. ** 2013-05-23: OpenStack Icehouse o superior ** 2014-10-16: OpenStack Juno o superior ** 2015-04-30: OpenStack Kilo
- Se indica la descripción del template.
- Parámetros (parameters): En esta sección declaramos los parámetros de entrada para el template, que serán proporcionados al lanzar el mismo. Estos parámetros pueden ser por ejemplo el sabor de una instancia, el nombre de la misma o cualquier parámetro que vayamos a usar en la construcción del escenario.

- Recursos (resources): En esta sección declaramos los recursos que vamos a crear, como por ejemplo una instancia, una ip o un volumen. Puedes ver la lista de recursos en http://docs.openstack.org/developer/heat/template_guide/openstack.html.
- Salida (outputs): Una vez que el template ha sido ejecutado, en esta sección indicamos la información que queremos mostrar al usuario. Esta sección es opcional.

En el ejemplo anterior el usuario tiene que indicar como parámetro la red a la que se va a conectar la instancia. Vamos a crear un recurso que es una instancia con unas características indicadas en el template y por último vamos a mostrar al usuario la dirección IP de la instancia.

Sección de parámetros (parameters)

Como hemos dicho antes en esta sección definimos los parámetros de entrada, que serán proporcionados al inicializar el template. Estos parámetros se usan para personalizar cada despliegue, o para utilizar los parámetros en el entorno del despliegue.

Podemos restringir los valores que el usuario puede introducir en los parámetros de entrada utilizando la sección “constraints”, por ejemplo:

```
parameters:
  image:
    type: string
description: Image ID or image name to use for the server4
default: Debian Jessie (stable)
constraints:
  - allowed_values:
    - Debian Jessie (stable)
    - Fedora 20
```

Para más información sobre la sección de parámetros puedes mirar la página http://docs.openstack.org/developer/heat/template_guide/hot_spec.html#parameters-section.

Sección de recursos (resources)

La sección “Resources” define los recursos que componen un stack o “pila” desplegada desde el template. Tenemos distintos tipos de recursos. Los recursos se definen indicando los siguientes datos:

- resource ID: El nombre del recurso, debe de ser único en la sección “Resources”.
- type: El tipo del recurso, por ejemplo: OS::Nova::Server o OS::Neutron::Port. Este atributo es obligatorio.

- **properties:** Una lista de las propiedades del recurso. Si se desea el valor puede ser proporcionado por una función. Esta sección es opcional.

Aquí tenemos un ejemplo en el que vemos las propiedades más usadas como el nombre, la clave ssh, la imagen a utilizar, el sabor, si queremos crear un usuario, redes.

```
resources:
  Instancia_de_prueba:
    type: OS::Nova::Server
    properties:
      name: Debian
      key_name: {get_param: key_name}
      image: {get_param: image}
      flavor: {get_param: flavor}
      admin_user: {get_param: user_name}
      networks: [{"network": "red1"}]
```

- **depends_on:** Dependencia del recurso en uno o más recursos del mismo template. Para más información ver http://docs.openstack.org/user-guide/hot-guide/hot_spec.html#hot-spec-resources-dependencies. Este atributo es opcional.

Aquí podemos ver un ejemplo de dependencia:

```
resources:
  server1:
    type: OS::Nova::Server
depends_on: [ server2, server3 ]

server2:
  type: OS::Nova::Server

server3:
  type: OS::Nova::Server
```

Intrinsic functions

Existen **diversas funciones** que pueden ser utilizadas para obtener valores de atributos de los recursos del template. Aquí tenemos una lista de algunas de las funciones disponibles:

get_attr

La función `get_attr` hace referencia a un atributo de un recurso.

La sintaxis de la función es la siguiente:

- resource ID: El nombre del recurso del que se necesita conocer el valor del atributo. Este recurso debe existir en la sección resources.
- attribute name: El nombre del atributo del que se quiere obtener el valor.

Un ejemplo del uso de la función get_attr:

```
resources:
  my_instance:
    type: OS::Nova::Server
  # ...

outputs:
  instance_ip:
    description: IP address of the deployed compute instance
    value: { get_attr: [my_instance, first_address] }
  instance_private_ip:
    description: Private IP address of the deployed compute instance
  value: { get_attr: [my_instance, networks, private, 0] }
```

get_file

La función get_file obtiene el contenido de un fichero. Se suele usar como mecanismo de inclusión de ficheros de configuración, scripts, etc. El argumento puede ser representado como ruta estática como absoluta.

Aquí tenemos el ejemplo:

```
resources:
  my_instance:
    type: OS::Nova::Server
properties:
  # general properties ...
  user_data:
    get_file: my_instance_user_data.sh
```

get_param

La función get_param nos permite obtener los datos que hemos indicado en la sección parámetros del template. Obtiene el valor y lo asigna en la ejecución del template. Utilizar esta función y la sección “parameters” adecuadamente nos puede facilitar las cosas en diferentes despliegues. Ejemplo:

```
resources:
  Instancia_de_prueba:
    type: OS::Nova::Server
    properties:
      name: Debian
```

```
key_name: {get_param: key_name}
image: {get_param: image}
flavor: {get_param: flavor}
admin_user: {get_param: user_name}
networks: [{"network": "red1"}]
```

Para más información de heat tienes a tu disposición la documentación oficial:http://docs.openstack.org/developer/heat/template_guide/.