

openstack / python nova

Introduction

Openstack est un projet qui est n   en 2010 (licence Apache 2.0) par l'entreprise Rackspace. OpenStack est un logiciel libre qui va nous permettre de faire du cloud computing qui permet de faire du IaaS pour du cloud priv   ou public. Le but d'Openstack est d'offrir    son utilisateur une multitude de module qui va permettre    son utilisateur de faire de l'infrastructure service c'est    dire deploy   des machines virtuelle en optimisant les ressources matrielles. On peut les deployer dynamiquement pour une courte dur  e, mais il est   galement possible de deploy   un ensemble de machines constituant une infrastructure externe.

Les modules : services d'OpenStack : Le développement d'OpenStack est un coup en différents modules (projets). Les modules sont des agrégats de multiples technologies communiquant ensemble par le biais d'une interface de programmation (API) et du protocole HTTP. Ils sont par ailleurs tous indispensables pour arriver au web à l'usage.

Keystone, le service d'identité

Fournit un service d'authentification et d'autorisation pour les autres services OpenStack. Fournit un catalogue de endpoints pour tous les services OpenStack.

Glance, la gestion d'images

Stocke et récupère des images disques de machines virtuelles. OpenStack Compute les utilise lors du provisioning d'une instance.

Nova, le Compute

Nova est le coeur du projet Openstack il gère le cycle de vie des instances dans un environnement OpenStack. Les tâches incluent la planification, la création et la mise hors service de machines virtuelles à la demande.

Horizon, l'interface web

Fournit un portail libre-service de type web permettant d'interagir avec les services sous-jacents OpenStack, comme le lancement d'une instance, l'attribution d'adresses IP ou la configuration des contrôles d'accès.

Cinder, le service de disques persistants

Fournit un stockage bloc persistant aux instances en cours d'exécution. Son architecture basée sur des drivers de type plugin facilite la création et la gestion des devices de stockage bloc.

Neutron, la gestion de réseaux

Permet le Network-Connectivity-as-a-Service pour d  t   autres services d  t   OpenStack, comme Compute. Fournit une API utilisateur pour d  finir les r  seaux et les attachements    ces r  seaux. Poss  de une architecture modulaire qui permet le support de la plupart des fournisseurs et des technologies r  seaux.

Swift, le stockage d'objet

Stocke et r  cup  re des objets de donn  es non structur  es via une API RESTful bas  e sur HTTP. Le service est hautement tol  rant aux pannes avec sa r  plication de donn  es et son architecture de type scale-out. Son impl  mentation diff  re des serveurs de fichiers    r  pertoires montables. Le service   crit les objets et les fichiers vers plusieurs disques, en s  curisant que les donn  es sont r  pliqu  es sur un cluster de serveurs.

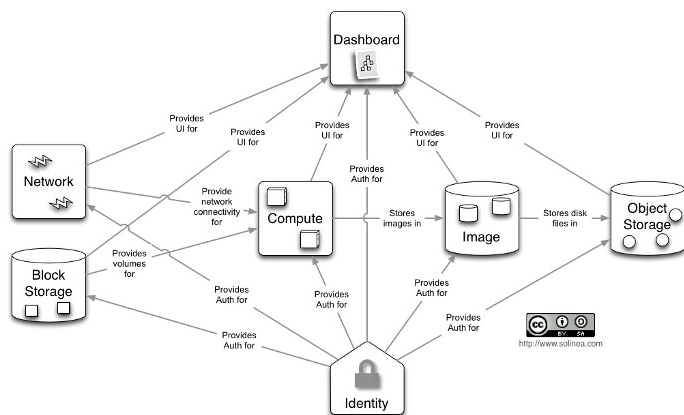
Heat, le service d'orchestration

Orchestre de nombreuses applications de cloud composites en utilisant soit le format de template natif HOT ou le format CloudFormation dâ€™AWS, soit au travers dâ€™une API REST native OpenStack, soit au travers dâ€™une API compatible avec CloudFormation.

Ceilometer, le service de mÃ©trologie

Surveillance et mesure un cloud OpenStack dans un but de facturation, de mesure de performances, de scalabilit  et de statistiques.

Voici un shema qui permet de montrÃ©e le lien entre tout les modules:



Le Client Python nova

Le client Python-nova est un client en ligne de commande pour le module Nova OpenStack , il va nous permettre de mettre en oeuvre 100% de l'API Nova , il va permettre la gestion des instances , des images, etc...

Installation de python-nova

Pour installer le plugin python-nova, il faut avoir préalablement installé python et son système d'installation pip.

Pour lancer l'installation il suffit de taper `pip install -U python-novaclient`.

Configuration des variables d'environnement pour Openstack

Pour configurer toutes les variables, Openstack génère un fichier RC contenant toutes les variables d'environnement à configurer.

Depuis cloudwatt il faut aller dans les paramètres et sélectionner puis accéder à l'API et enfin télécharger le fichier.

L'exécution du fichier se fait grâce à la commande `source 0750182707_projet_tutore_2017-openrc.sh` et permet la configuration automatique des variables.

Liste des instances

La liste des instances créées sont visibles à l'aide de la commande `nova list`.

Création de l'instance

Génération de la clef ssh

`ssh-keygen`

Intégration clef ssh au keypair Openstack

`nova keypair-add --pub-key .ssh/id_rsa.pub SSHKEY`

Choix du flavor

`nova flavor-list` affiche la liste des flavors disponibles. Une fois choisi, il faut récupérer son ID qui sera renseigné lors de la création de l'instance.

Choix de l'image (système installé)

`nova image-list` affiche la liste des images système disponibles. Une fois choisi, il faut récupérer son ID qui sera demandé lors de la génération de l'instance.

Création de l'instance

`nova boot --key-name SSHKEY --flavor 16 --image 185e1975-c9c5-4358-909e-5e329808902e instance1`

Pour la création de l'instance on retrouve quatre éléments :

- le nom du keypair
- l'id du flavor
- l'id de l'image
- le nom de l'instance

```
1 2 3 4 5 7 8
cpu 07% 222.0 GiB lan: 192.168.1.80 100 Mbit/s

Usage: nova [--version] [--debug] [--os-cache] [--timings]
        [--os-region-name <region-name>] [--service-type <service-type>]
        [--service-name <service-name>]
        [--os-endpoint-type <endpoint-type>]
        [--os-compute-api-version <compute-api-ver>]
        [--endpoint-override <bypass-url>] [--profile HMAC_KEY]
        [--insecure] [--os-cacert <ca-certificate>]
        [--os-cert <certificate>] [--os-key <key>] [--timeout <seconds>]
        [--os-auth-type <name>] [--os-auth-url OS_AUTH_URL]
        [--os-domain-id OS_DOMAIN_ID] [--os-domain-name OS_DOMAIN_NAME]
        [--os-project-id OS_PROJECT_ID]
        [--os-project-name OS_PROJECT_NAME]
        [--os-project-domain-id OS_PROJECT_DOMAIN_ID]
        [--os-project-domain-name OS_PROJECT_DOMAIN_NAME]
        [--os-trust-id OS_TRUST_ID]
        [--os-default-domain-id OS_DEFAULT_DOMAIN_ID]
        [--os-default-domain-name OS_DEFAULT_DOMAIN_NAME]
        [--os-user-id OS_USER_ID] [--os-username OS_USERNAME]
        [--os-user-domain-id OS_USER_DOMAIN_ID]
        [--os-user-domain-name OS_USER_DOMAIN_NAME]
        [--os-password OS_PASSWORD]
        <subcommand> ...

error: unrecognized arguments: --key-name instance1
Try 'nova help' for more information.
upsi@yoshi:~$ nova boot --key-name SSHKEY --flavor 16 --image 185e1975-c9c5-4358-909e-5e329808902e instance1
/usr/local/lib/python2.7/dist-packages/novaclient/client.py:278: UserWarning: The 'tenant_id' argument is deprecated in Ocata and its use may result
. As 'project_id' is provided, the 'tenant_id' argument will be ignored.
  warnings.warn(msg)

+-----+-----+
| Property | Value |
+-----+-----+
| OS-DCF:diskConfig | MANUAL |
| OS-EXT-AZ:availability_zone | nova |
| OS-EXT-STS:power_state | 0 |
| OS-EXT-STS:task_state | scheduling |
| OS-EXT-STS:vm_state | building |
| OS-SRV-USG:launched_at | - |
| OS-SRV-USG:terminated_at | - |
| accessIPv4 | |
| accessIPv6 | |
| config_drive | |
| created | 2017-02-08T09:48:05Z |
| flavor | t1.cw.tiny (16) |
| hostId | terminal: mod+Enter |
| id | focused: mod+Shift+q | 6514405a-5190-495e-ac41-63b6a5e53720 |
| image | multi-selection: mod+q | Debian Jessie (185e1975-c9c5-4358-909e-5e329808902e) |
| key_name | mod+d | SSHKEY |
| metadata | mod+z | {} |
| name | mod+f2 | instance1 |
| os-extended-volumes:volumes_attached | [] |
| progress | mod+Ctrl+arrowkeys | 0 |
| security_groups | Shift+Space | default |
| status | | BUILD |
| tenant_id | Shift+h | 54126a95359b4ad29eb8bac4d4f153c2 |
| updated | Shift+h | 2017-02-08T09:48:06Z |
| user_id | Super> | ed4305ee0fd4f134a5b1d146bea45b77 |
+-----+-----+

upsi@yoshi:~$
```