

openstack / python nova

Introduction

Openstack est un projet qui est né 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 privé ou public. Le but d'Openstack est d'offrir à son utilisateur une multitude de module qui va permettre à son utilisateur de faire de l'infrastructure service c'est à dire déployée des machines virtuelle en optimisant les ressources matérielles. On peut les déployer dynamiquement pour une courte durée, mais il est également possible de déployer un ensemble de machines constituant une infrastructure externe.

Les modules : services d'OpenStack.

Keystone, le service d'identité

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

Glance, la gestion d'images

Stocke et récupère des images disques de machines virtuelles. OpenStack Compute les utilise lors du provisioning d'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 d'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'autres services d'OpenStack, comme Compute. Fournit une API utilisateur pour définir les réseaux et les attachements à ces réseaux. Possède une architecture modulaire qui permet le support de la plupart des fournisseurs et des technologies réseau.

Swift, le stockage d'objet

Stocke et récupère des objets de données non structurées via une API RESTful basée sur HTTP. Le service est hautement tolérant aux pannes avec sa réplication de données et son architecture de type scale-out. Son implémentation diffère des serveurs de fichiers à répertoires montables. Le service écrit les objets et les fichiers vers plusieurs disques, en s'assurant que les données sont répliquées sur un cluster de serveurs.

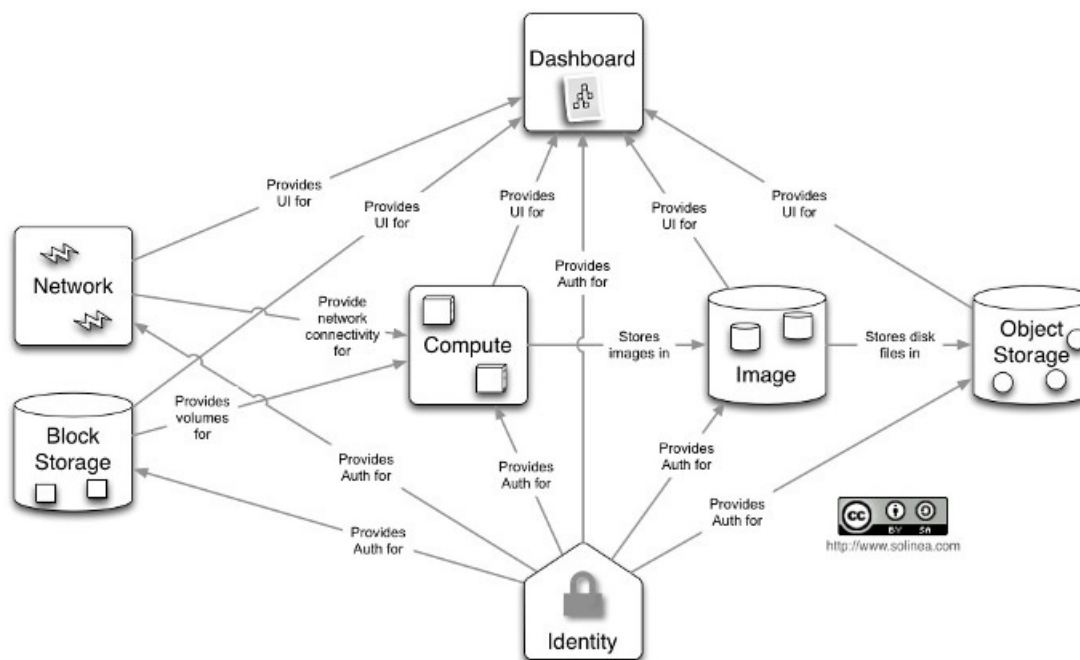
Heat, le service d'orchestration

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

Ceilometer, le service de métrologie

Surveille et mesure un cloud OpenStack dans un but de facturation, de mesure de performances, de scalabilité et de statistiques.

Voici un shema qui permet de montrée 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 accès et sécurité puis accès 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 systèmes 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 quatres éléments :

- le nom du keypair

-l'id du flavor

- l'id de l'image

- le nom de l'instance

```

2 4 5 7 8
usage: nova [--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>]
           [--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.

yoshi@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 in errors in future releases
  if '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
id	t1.cw.ting (16)
image_ref	6514405a-5190-495e-ac41-63b6a5e53720
key_name	SSHKEY
metadata	{}
name	instance1
os-extended-volumes:volumes_attached	[]
progress	0
security_groups	default
status	BUILD
tenant_id	54126a95359b4ad29eb8bac4d4f153c2
updated	2017-02-08T09:48:06Z
user_id	ed34305ee0fd4f4134a5b1d146bea45b77

```

yoshi@yoshi:~$

```