Guide d'utilisation du Cloud HPC "Interface en lignes de commandes (CLI)"

UNIVERSITÉ DE LILLE

Cyrille TOULET
cyrille.toulet@univ-lille1.fr

18 octobre 2016

Table des matières

Chapitre 1

Avant-propos

Cette documentation détaille l'utilisation de l'infrastructure Cloud HPC de l'Université de Lille au travers des clients OpenStack en lignes de commandes (CLI).

Elle est principalement inspirée de la documentation du Cloud OpenStack de Strasbourg Grand-Est rédigée par Jérôme PANSANEL et Vincent LEGOLL.

Si vous remarquez une erreur, un manque ou qu'une explication est perfectible, n'hésitez pas à me faire parvenir vos remarques à l'adresse suivante : cyrille.toulet@univ-lille1.fr.

Chapitre 2

Installation des clients OpenStack

2.1 Installation

Les clients OpenStack sont disponibles pour différents systèmes d'exploitation, soit à travers une installation avec PIP ou par les gestionnaires de paquets comme APT ou YUM.

Dans tous les cas, il est nécessaire que le logiciel Python (version 2.6 ou ultérieure) soit installé. Une seule installation est nécessaire, soit avec le gestionnaire de paquets de votre distribution soit avec PIP.

Nous recommandons d'utiliser l'installation à l'aide des gestionnaires de paquets quand c'est possible.

2.1.1 Python PIP

Pour suivre cette partie, il est nécessaire que PIP soit installé. Cette installation est détaillée sur le site internet de PIP.

Les clients OpenStack sont installés avec les commandes suivantes :

```
pip install python-keystoneclient
pip install python-novaclient
pip install python-glanceclient
pip install python-cinderclient
pip install python-neutronclient
```

Notez que si l'exécution d'une des commandes précédentes échoue avec le message d'erreur **error : invalid command 'egg_info'**, vous devez au préalable exécuter la commande suivante :

```
pip install --upgrade setuptools
```

2.1.2 Systèmes de type Debian

Si votre système est de type Debian (Debian, Ubuntu, Mint, Kali Linux, Tails, etc), l'installation est relativement aisée.

Il suffit d'exécuter en tant que super-utilisateur (c'est-à-dire avec l'utilisateur root ou avec la commande sudo si votre utilisateur a les droits requis) :

```
sudo apt-get install python-keystoneclient
sudo apt-get install python-novaclient
sudo apt-get install python-glanceclient
sudo apt-get install python-cinderclient
sudo apt-get install python-neutronclient
```

Note : Pour les versions les plus anciennes d'Ubuntu, il peut être nécessaire de réaliser au préalable :

```
sudo apt-get install python-software-properties
sudo add-apt-repository cloud-archive:juno
sudo apt-get update
sudo apt-get dist-upgrade
```

Attention, cette procédure mettera à jour votre système d'exploitation vers la dernière version stable !

2.1.3 Systèmes de type RHEL

Si votre système est de type RHEL (RedHat, Fedora, CentOS, Scientific Linux, etc), l'installation est réalisée en quelques étapes avec la commande yum :

```
sudo yum install epel-release
sudo yum install python-keystoneclient
sudo yum install python-novaclient
sudo yum install python-glanceclient
sudo yum install python-cinderclient
sudo yum install python-neutronclient
```

2.2 Configuration

Après avoir réalisé l'installation, vous êtes prêt(e) à utiliser le service.

La première étape est de paramétrer l'environement du client. Pour cela, créez le fichier *$HOME/cloud-hpc-lille.rc* avec le contenu suivant (les valeurs **username**, **password** et **tenant** sont à remplacer par celles qui vous ont été transmises lors de la création de votre compte sur l'infrastructure) :

```
export OS_USERNAME=username
export OS_PASSWORD=password
export OS_TENANT_NAME=tenant
export OS_AUTH_URL=http://ouranos.univ-lille1.fr:5000/v2.0
```

Une fois que ce fichier est créé, il doit être sourcé pour charger les différentes variables d'environnement dans votre shell actuel :

```
source ${HOME}/cloud-hpc-lille.rc
```

Il est nécessaire de réaliser cette étape à chaque fois que vous lancez un nouveau shell, à moins que vous n'ajoutiez la commande source précédente au fichier d'initialisation de votre shell (par exemple, le fichier **$HOME/.bashrc** pour le shell bash).

Vous pouvez maintenant tester que votre client fonctionne et qu'il arrive à se connecter correctement au Cloud :

```
nova list
```

2.3 Découverte

Quelques commandes permettent de voir les éléments disponibles pour construire une machine virtuelle.

Tout d'abord, la liste des images de systèmes d'exploitation pour les machines virtuelles est obtenue avec la commande **nova image-list** :

```
+--------------------------------------+--------------------------+--------+--------+
| ID                                   | Name                     | Status | Server |
+--------------------------------------+--------------------------+--------+--------+
| 9f0070de-8604-40f6-9970-e16f54d961d3 | CentOS 6.6 (64 bit)      | ACTIVE |        |
| 7d6890f7-9c1e-46ef-982d-36e1f8fed7f7 | CentOS 7.0 (64 bit)      | ACTIVE |        |
| 9cf55b54-1fae-4cda-a272-b62ea5f7d504 | Debian 8.0 (64 bit)      | ACTIVE |        |
| 57360bf6-66ef-4056-9e6e-366da7b58f6e | QIIME 1.9.1 (64 bit)     | ACTIVE |        |
| 9f5fc883-0a7b-4c01-b2e4-fb92db8e8e69 | Ubuntu 14.04.2 (64 bit)  | ACTIVE |        |
+--------------------------------------+--------------------------+--------+--------+
```

Ensuite, la liste des types de machines virtuelles disponibles est affichée avec **nova flavor-list** :

```
+--------------------------------------+----------------+-----------+------+-------+
| ID                                   | Name           | Memory_MB | Disk | VCPUs |
+--------------------------------------+----------------+-----------+------+-------+
| aaf36e69-aec6-450d-a486-5c0a9f2e1a87 | 8-CPU-12GB-RAM | 12288     | 25   | 8     |
+--------------------------------------+----------------+-----------+------+-------+
```

Enfin, pour connaître les réseaux utilisables par la machine virtuelle, utilisez **nova net-list** :

```
+--------------------------------------+--------------+------+
| ID                                   | Label        | CIDR |
+--------------------------------------+--------------+------+
| 16ce5bd6-ded2-4592-9317-80f53083ae7d | fg-net       | None |
| e7320b8e-8be0-4f0f-ba85-89b839fdb9ec | ext-net      | None |
+--------------------------------------+--------------+------+
```

Chapitre 3

Réseaux virtuels

3.1 Réseau, sous-réseau et routeur virtuel

Si la commande **nova net-list** tapée précédemment ne liste que le réseau *ext-net*, il vous faut avant toute chose créer un réseau virtuel.

Les commandes qui suivent décrivent la procédure de création d'un réseau virtuel accompagné de son sous-réseau et d'un routeur virtuel.

Tout d'abord, créons un réseau *my-net* :

```
neutron net-create my-net
```

Maintenant, créons un sous-réseau *my-subnet* attaché au réseau *my-net* :

```
neutron subnet-create my-net --name my-subnet --gateway 192.168.0.254 \
  --dns-nameserver 208.67.222.222 192.168.0.0/24
```

Pour ce sous-réseau, l'adresse réseau est *192.168.0.0/24*, l'adresse du routeur est *192.168.0.254* et l'adresse du DNS est *208.67.222.222* (OpenDNS).

Il nous reste donc à créer un routeur virtuel pour router notre nouveau sous-réseau (*my-subnet*) sur le réseau public (*ext-net*). Ce routeur sera attaché à *my-net* sur l'adresse *192.168.0.254* (gateway choisie à la création du sous-réseau) et au réseau *ext-net* sur une adresse obtenue automatiquement.

Créons donc un routeur *my-router* et attachons le aux réseaux décrits précédemment :

```
neutron router-create my-router
neutron router-gateway-set my-router ext-net
neutron router-interface-add my-router my-subnet
```

Vous noterez que la configuration des adresses s'est faite automatiquement.

Votre réseau est maintenant créé, vous pouvez vous en assurer avec la commande **neutron net-list**.

3.2 Adresse IP publique

Avant d'attacher une adresse IP publique à une machine virtuelle, il faut vérifier si une adresse est disponible (champ *Fixed IP* vide) avec la commande **nova floating-ip-list** :

```
+---------------+-----------+----------+---------+
| Ip            | Server Id | Fixed Ip | Pool    |
+---------------+-----------+----------+---------+
| 193.54.101.38 | -         | -        | ext-net |
| 193.54.101.39 | -         | -        | ext-net |
+---------------+-----------+----------+---------+
```

Si et seulement si il n'y a aucune adresse disponible, demandez en une nouvelle :

```
nova floating-ip-create
```

Cette commande vous renvoie une nouvelle adresse (par exemple 193.54.101.40) qui sera maintenant listée par la commande **nova floating-ip-list** :

```
+---------------+-----------+----------+---------+
| Ip            | Server Id | Fixed Ip | Pool    |
+---------------+-----------+----------+---------+
| 193.54.101.38 | -         | -        | ext-net |
| 193.54.101.39 | -         | -        | ext-net |
| 193.54.101.40 | -         | -        | ext-net |
+---------------+-----------+----------+---------+
```

Pour cet exemple, considérons que vous ayez choisi d'attribuer l'adresse IP publique *193.54.101.38* à votre machine virtuelle nommée *MY_VM_NAME*.

Pour attribuer cette adresse à votre instance, utilisez la commande :

```
nova add-floating-ip MY_VM_NAME 193.54.101.38
```

Il est également possible de vérifier que l'adresse IP a bien été attachée :

```
nova list --name MY_VM_NAME
```

Note : Maintenant que la machine virtuelle a une adresse IP publique, vous pouvez vous y connecter via SSH :

```
ssh -i ${HOME}/.ssh/cloud-hpc-lille.key root@193.54.101.38
```

3.3 Pare-feu virtuel

Dans OpenStack, les pare-feu se gèrent à travers des groupes de sécurité (security-group).

Les groupes de sécurité sont des ensembles de règles de filtrage appliqués à des machines virtuelles.

Par défaut, il n'existe qu'un groupe de sécurité nommé *default*. Les règles de ce groupe n'autorisent que les paquets allant de la VM vers l'extérieur (les paquets sortants).

Nous allons donc créer un nouveau groupe de sécurité pour vous donner accès à vos machines virtuelles.

Tout d'abbord, créez un groupe de sécurité nommé *private* :

```
nova secgroup-create private "The private rules (mgmt, admin, etc.)"
```

Si vous listez vos groupes de sécurité, vous devriez voir le groupe *default* et le groupe *private* :

```
nova secgroup-list
```

L'intéret de ce groupe est de n'autoriser que vos stations de travail pour administrer vos machines virtuelles.

Pour cela, vous allez avoir besoin de votre adresse IP publique. Si vous ne la conaissez pas, vous pouvez la trouver sur le site monip.org ou demander à votre service réseau.

Supposons ici que votre adresse IP est *134.1.1.1*.

Autorisons la à établir une connexion SSH (port 22 en TCP) :

```
nova secgroup-add-rule private tcp 22 22 134.1.1.1/32
```

Note : Nous venons d'autoriser les connexions TCP dans l'intervale de ports allant de 22 à 22 (donc uniquement le port 22) pour l'adresse IP 134.1.1.1.

De la même manière, si votre machine est vouée à être publiquement accessible (par exemple une page web), nous créerons un groupe de sécurité nommé *public* dans lequel nous autoriserons toutes les adresses à accéder aux ports web :

```
nova secgroup-create public "The public rules (web, API, etc.)"
nova secgroup-add-rule public tcp 80 80 0.0.0.0/0
nova secgroup-add-rule public tcp 443 443 0.0.0.0/0
```

Maintenant que vos groupes de sécurité sont créés, il reste à les appliquer à vos VMs.

Par exemple :

```
nova add-secgroup MY_VM_NAME private
nova add-secgroup MY_WEB_VM_NAME private
nova add-secgroup MY_WEB_VM_NAME public
```

En cas de doute, n'hésitez pas à contacter votre administrateur Cloud.

Chapitre 4

Machines virtuelles

4.1 Gestion des clés

Afin de pouvoir se connecter à vos futures machines virtuelles, il est nécessaire d'utiliser une clé SSH qui sera enregistrée auprès du serveur de clefs.

Si vous n'avez pas de clé, commencez par en générer une :

```
ssh-keygen -t rsa -f ${HOME}/.ssh/cloud-hpc-lille
```

Ensuite, enregistrez votre clé sur le cloud :

```
nova keypair-add --pub-key=${HOME}/.ssh/cloud-hpc-lille.pub cloudkey
```

Note : Une fois votre clé enregistrée, il n'est plus nécessaire d'en générer d'autres, ni de les enregistrer.

Vous pouvez également lister vos clés enregistrées avec la commande suivante :

```
nova keypair-list
```

4.2 Création d'une instance

Dans la section **Découverte (2.3)**, nous avons récupéré la liste de tous les éléments utilisables pour composer la machine virtuelle.

Une fois que vous avez choisi les différents éléments de votre machine virtuelle, elle peut être instanciée à l'aide de la commande **nova boot**.

Par exemple, si nous souhaitons lancer une image Debian avec 8 CPUs, 12 Go de RAM et 25 Go de disque dur sur le réseau fg-net et dont le nom sera MY_VM_NAME, nous utiliserons la commande :

```
nova boot \
  --key-name=cloudkey \
  --image=9cf55b54-1fae-4cda-a272-b62ea5f7d504 \
  --flavor=aaf36e69-aec6-450d-a486-5c0a9f2e1a87 \
  --nic net-id=16ce5bd6-ded2-4592-9317-80f53083ae7d \
  MY_VM_NAME
```

Note : Lors du lancement de machines virtuelles, vous pouvez vous retrouver confronté à des problèmes de dépassement de quota.

Vous pouvez soit attendre que d'autres utilisateurs en libèrent, ou alors demander à votre administrateur de cloud de vous attribuer un quota supplémentaire. Vous pouvez consulter les limites de quota grâce à la commande **nova quota-show**.

Note : Lors de l'utilisation des commandes nova, il est possible d'utiliser aussi bien les noms (par exemple MY_VM_NAME) que les identifiants (par exemple 070da4c0-5ec4-475c-9177-e5bfaba63339).

Il est recommandé d'utiliser les identifiants, car ils sont uniques (il est possible de lancer deux machines virtuelles avec le même nom).

4.3 Surveillance d'une instance

Pour suivre l'état de votre machine virtuelle, utilisez la commande **nova show MY_VM_NAME**.

Le status **ACTIVE** renvoyé par cette commande nous indique que la VM est prête à être utilisée.

Si vous venez toutefois de l'instancier, elle ne possède pas encore d'interface vers le réseau externe puisque son adresse IP est dans un réseau interne. Avant de pouvoir s'y connecter par SSH, il est donc nécessaire de lui attacher une adresse IP publique (voir le chapitre **Adresse IP publique (3.2)**).

Le status **SPAWNING** signifie que l'image système que vous avez choisi est en train d'être transférée vers l'hyperviseur qui hébergera votre instance.
Votre instance ne passera pas nécessairement par cet état si l'hyperviseur possède déjà l'image système choisie en local.

Le status **CREATING** signifie que l'image système est en cours d'instanciation sur l'hyperviseur.

Enfin, le status **SHUTOFF** signifie que votre instance est éteinte.

4.4 Suppression d'une instance

Une fois les travaux terminés sur la VM, vous pouvez l'arrêter pour la redémarrer plus tard :

```
nova stop MY_VM_NAME
...
nova start MY_VM_NAME
```

Cependant, nous vous recommandons de la supprimer si vous ne comptez plus l'utiliser avant un certain temps.

Dans ce cas, toutes les modifications que vous avez apporté à l'image (installation de paquets, etc.) seront supprimées, hormis celles qui sont sur un disque persistant (aussi appelé *volume virtuel*).

Avant de supprimer l'instance, il faut se connecter à la VM et démonter le disque persistant pour éviter de corrompre les données (voir le chapitre **Utilisation d'un volume (5.2)**).

Enfin, vous pourez supprimer votre instance :

```
nova delete MY_VM_NAME
```

Attention : Cette action est iréversible !

Chapitre 5

Stockage virtuel

5.1 Création d'un volume

Par défaut, lorsqu'une machine virtuelle est détruite, tous les changements que vous avez pu y apporter disparaissent. Pour pouvoir stocker des données réutilisables entre plusieurs sessions, il est nécessaire de créer des disques permanents.

La gestion des disques permanents se fait avec le client cinder.

Pour afficher la liste de vos disques, utilisez :

```
cinder list
```

Pour créer un nouvel espace de stockage persistant (nommé MY_VOLUME_NAME d'une taille de 8 Go), exécutez :

```
cinder create --display_name MY_VOLUME_NAME 8
```

Votre volume virtuel est maintenant prêt à être utilisé.

5.2 Utilisation d'un volume

Pour attacher ce nouveau volume à la machine virtuelle à l'aide de son identifiant, utilisez la commande :

```
nova volume-attach MY_VM_NAME MY_VOLUME_NAME /dev/vdb
```

Le stockage sera vu par l'OS sous le nom de */dev/vdb*.

Pour vérifier que le disque est bien associé, vérifiez que la colonne *Status* à pour valeur *in-use* et que la colonne *Attached* to contient bien l'identifiant de la VM :

```
cinder list
```

Pour détacher ce volume, utilisez la commande :

```
nova volume-detach MY_VM_NAME MY_VOLUME_NAME
```

Avant de détacher un volume virtuel pour le déplacer d'une machine virtuelle à une autre, démontez le au sein de la VM pour garantir l'intégrité des données stockées sur ce disque !

Note : Un volume virtuel ne peut-être attaché qu'à une VM a la fois.

A sa création, un volume virtuel ne contient aucune donnée et n'est pas formaté, à l'instar d'un disque dur neuf.

Si ce disque est destiné à contenir des données confidentielles, nous vous recommandons de le chiffrer (voir le chapritre **Chiffrement d'un volume (5.3)**).

Dans le cas contraire, connectez-vous à votre VM pour formater ce disque et le monter :

```
mkfs.ext4 /dev/vdb
mkdir /media/storage1
mount /dev/vdb /media/storage1
df -h /media/storage1
```

Note : La dernière commande permet de vérifier que nous avons bien l'espace disponible de la taille choisie (ici 8 Go) monté sur /media/storage1.

Notez également que si vous redémarrez la machine virtuelle, le disque ne sera pas remonté automatiquement. Pour cela, référez-vous à la documentation de fstab pour le système d'exploitation choisi.

Le disque virtuel peut également être partitionné avant le formatage. Pour ce faire, référez-vous à la documentation du système d'exploitation choisi.

5.3 Chiffrement d'un volume

Cette section détaille l'utilisation de l'outil dm-crypt/LUKS pour le chiffrement des disques permanents. Cet outil est fourni en standard par les distributions Linux et peut facilement être installé avec le gestionnaire de paquets dans votre machine virtuelle.

Pour chiffrer un disque permanent, il faut tout d'abord l'initialiser correctement. Dans l'exemple ci-dessous, le disque dénommé */dev/vdb* est dans un premier temps rempli de données aléatoires, puis il est initialisé à l'aide de la commande **cryptsetup** ci-dessous :

```
dd if=/dev/urandom of=/dev/vdb bs=4k
cryptsetup -v \
```

```
  --cipher aes-xts-plain64 \
  --key-size 512 \
  --hash sha512 \
  --iter-time 5000 \
  --use-random luksFormat \
  /dev/vdb
```

Note : Cette première étape peut être assez longue ...

Ensuite, vérifiez que le disque est maintenant du type LUKS :

```
cryptsetup luksDump /dev/vdb
```

Cette commande produit un affichage similaire à ce qui suit :

```
LUKS header information for /dev/vdb

Version:        1
Cipher name:    aes
Cipher mode:    xts-plain64
Hash spec:      sha512
Payload offset: 4096
MK bits:        512
MK digest:      c4 f7 4b 02 2a 3f 12 c1 2c ba e5 c9 d2 45 9a cd 89 20 6c 73
MK salt:        98 58 3e f3 f6 88 99 ea 2a f3 cf 71 a0 0d e5 8b
                d5 76 64 cb d2 5c 9b d1 8a d3 1d 18 0e 04 7a eb
MK iterations:  81250
UUID:           c216d954-199e-4eab-a167-a3587bd41cb3

Key Slot 0: ENABLED
Iterations:          323227
Salt:                a0 45 3e 98 fa cf 60 74 c6 09 3d 54 97 89 be 65
                     5b 96 7c 1c 39 26 47 b4 8b 0e c1 3a c9 94 83 c2
Key material offset: 8
AF stripes:          4000
Key Slot 1: DISABLED
Key Slot 2: DISABLED
Key Slot 3: DISABLED
Key Slot 4: DISABLED
Key Slot 5: DISABLED
Key Slot 6: DISABLED
Key Slot 7: DISABLED
```

Le disque est maintenant prêt à être utilisé !

La première fois que vous l'utilisez, il faut effectuer les étapes suivantes :

1. Ouvrez le disque chiffré avec la commande **cryptsetup luksOpen** (le nom storage1 n'est qu'indicatif, vous pouvez choisir ce que vous voulez) :

   ```
   cryptsetup luksOpen /dev/vdb storage1
   ```

2. Créez un système de fichier sur le disque :

   ```
   mkfs.ext4 /dev/mapper/storage1
   ```

3. Créez le point de montage du disque :

   ```
   mkdir /mnt/storage1
   ```

4. Montez le disque :

   ```
   mount -t ext4 /dev/mapper/storage1 /mnt/storage1
   ```

5. Vérifiez l'espace disponible (cela peut être légèrement différent de ce qui a été choisi à la création du volume) :

   ```
   df -h /mnt/storage1
   ```

Note : Une fois que le disque est opérationnel, les étapes 2 et 3 ne sont plus nécessaires.

Enfin, lorsque vous avez terminé votre travail sur le disque, vous pouvez le démonter proprement avec les commandes suivantes :

```
umount /mnt/storage1
cryptsetup close storage1
```

5.4 Suppression d'un volume

Si vous n'avez plus besoin des données sur le disque persistant, il faut le réinitialiser avec des données aléatoires pour des raisons de confidentialité :

```
dd if=/dev/urandom of=/dev/vdb bs=4k
```

Cette opération peut être assez longue.

Note : Si vous avez ajouté votre volume persistant dans le fichier /etc/fstab de votre machine virtuelle, pensez à le supprimer avant de démonter le disque virtuel sous peine de ne plus pouvoir démarrer votre VM.

Puis, une fois la suppression des données effective, détachez et supprimez le volume avec :

```
nova volume-detach MY_VM_NAME MY_VOLUME_NAME
cinder delete MY_VOLUME_NAME
```

Attention : Cette action est iréversible !