

INF8480 – Systèmes répartis et infonuagique

TP3 – Initiation aux services de l'infonuagique

Chargé de laboratoire:

Adel Belkhiri

Daniel Capelo Borges

18 mars 2019

Polytechnique Montréal

Introduction

Le dernier Travail Pratique a pour objectif la familiarisation avec les services fournis par les clients de l'infonuagique. L'infonuagique permet d'abstraire plusieurs particularités physiques d'une grappe de calcul. Cette abstraction permet entre autres de plus facilement partager les ressources disponibles entre plusieurs utilisateurs. **OpenStack** est une infrastructure qui permet de mettre en place un nuage. Cette technologie sera utilisée pour instancier des machines virtuelles offrant un service Web.

Remise

- **Méthode:** Par Moodle, un seul membre de l'équipe doit remettre le travail, mais assurez-vous d'inclure les deux matricules dans le rapport et le nom du fichier remis.
- **Échéance:** le 15 avril 2019, avant 16h, voir le plan de cours pour les pénalités de retard.
- **Format:** Une archive au format .tar.gz contenant:
 - Un rapport avec les résultats des tests de performance et les réponses aux questions.
 - Un gabarit Heat pour le déploiement d'un serveur web sur **OpenStack**.
 - Un gabarit Heat pour le déploiement sur **3** machines virtuelles avec répartition de charge.

Barème

Respect des exigences et bon fonctionnement des gabarits Heat: 7 points

Résultats des tests de performance et discussion: 5 points

Réponses aux questions: 8 points

Total: 20 points, valant pour 10% de la note finale du cours.

Jusqu'à 2 points peuvent être enlevés pour la **qualité du rapport**.

Spécifications

Généralités

On vous demande de déployer un service Web implémenté en Python sur **OpenStack**. Le déploiement se fera à l'aide de Heat, qui permet de créer des gabarits spécifiant des règles pour l'allocation de différentes ressources **OpenStack** (machines virtuelles, adresses IP flottantes, moniteurs d'activités, alarmes...). Dans un premier temps, vous créerez un gabarit simple permettant de déployer le service Web sur une seule machine virtuelle. Vous devrez ensuite créer un second gabarit capable de gérer un groupe de plusieurs machines virtuelles offrant le même service Web et capable de se mettre à l'échelle automatiquement.

La version d'**OpenStack** installée sur la plateforme Canari est la version *Queens* et la documentation correspondante est disponible sur ce lien :

https://docs.openstack.org/heat/queens/template_guide/openstack.html

Déploiement local du service Web

Déployez le service Web Python fourni avec ce Travail Pratique sur votre machine locale. Notez les commandes que vous exécutez, car vous devez les insérer dans les gabarits créés aux étapes suivantes. Il n'y a rien à remettre pour cette partie.

Notez que les machines virtuelles OpenStack n'ont pas de navigateur Web. Vous devrez sans doute exécuter une commande pour télécharger le script Python à cette URL:

<https://raw.githubusercontent.com/houssemmh/INF8480-TP3/master/server.py>

Déploiement d'un service Web sur une seule machine virtuelle

Vous devez créer un gabarit Heat permettant de déployer le serveur web Python sur une seule machine virtuelle dans l'infonuagique **OpenStack**. La machine virtuelle doit utiliser l'image **xenial-server-cloudimg-amd64-disk1** et le gabarit **2048-10-1-1**. Elle doit aussi être connectée sur le réseau **switch1-nat** pour pouvoir accéder à l'internet.

Nous vous recommandons de consulter le dépo d'**OpenStack** sur GitHub où vous y trouverez plusieurs exemples/gabarits Heat:

<https://github.com/openstack/heat-templates>

Dans le gabarit, il faut bien définir les paramètres, les contraintes et les valeurs par défaut.

Déploiement d'un service Web avec répartition de charge

Vous devez créer un gabarit Heat qui permet de déployer le même service Web, mais cette fois-ci avec un répartiteur de charge.

Les utilisateurs du service Web devront pouvoir envoyer leurs requêtes à une adresse IP fixe, mais les différentes requêtes devront pouvoir être traitées de manière transparente par 3 machines virtuelles différentes.

Un moniteur d'activité doit vérifier le statut des machines à intervalles réguliers. Le moniteur d'activité envoie un message à intervalles de 15 secondes aux machines pour vérifier leur activité. S'il ne reçoit pas de réponse dans un délai de 5 secondes, il envoie une seconde requête. S'il ne reçoit pas de réponse à cette seconde requête dans un autre délai de 5 secondes, il déclare la machine inactive.

Nous fournissons donc avec le travail pratique un gabarit "loadBalancing_etudiants.yaml" que vous devez compléter. Vous pouvez aussi vous inspirer de l'exemple suivant disponible dans le dépo d'**OpenStack** sur GitHub :

<https://github.com/openstack/heat-templates/blob/master/hot/autoscaling.yaml>

Tests de performance

Performance avec et sans répartiteur de charge

Créez un script capable d'envoyer simultanément **50 requêtes Web** et de mesurer le temps de réponse moyen du serveur. La commande **wget** ou **curl** peut être utilisée pour récupérer le contenu de la page web.

Utilisez ce script pour comparer le temps de traitement des requêtes avec le service Web simple et le service Web avec répartition de charge.

Rapportez et discutez les résultats obtenus dans votre rapport.

Si vous obtenez des résultats aberrants, répétez l'expérience. Il est possible que le résultat ait été influencée par les machines virtuelles d'autres équipes ou d'une surcharge au niveau de l'infonuagique.

Questions

Question 1: **OpenStack** est en fait un regroupement de plusieurs composants distincts provenant d'autres projets. On vous demande de donner une description détaillée des composants utilisés dans ce Travail Pratique: Heat, Neutron et Nova.

Donnez aussi une description de trois autres composants de votre choix. Donnez des cas d'utilisation.

Question 2: Dans le gabarit *Heat* du service Web avec répartition de charge, vous avez défini des propriétés permettant d'instancier plusieurs ressources. Expliquez en une phrase le rôle de chacune de ces ressources dans le travail pratique.

- OS::Heat::ResourceGroup
- OS::Neutron::LBaaS::HealthMonitor
- OS::Neutron::LBaaS::Pool
- OS::Neutron::LBaaS::LoadBalancer
- OS::Nova::Server

Question 3: Nous avons créé 3 machines virtuelles qui demeurent actives en tout temps pour offrir un service Web. Si nous devions payer les ressources allouées à l'utilisation, cette solution ne serait guère optimale.

Pour mieux gérer nos ressources, on a décidé de remplacer OS::Heat::ResourceGroup par une autre ressource de **OpenStack** permettant de modifier dynamiquement le nombre d'instances du serveur.

1) Quel est le nom de cette ressource ?

2) Quelles sont 2 ressources OpenStack qui permettent de:

- lancer une alerte lorsque le taux d'utilisation du CPU de vos machines atteint des seuils prédéfinis
 - ajuster automatiquement le nombre de machines virtuelles en fonction de ces alertes
- Expliquez les paramètres nécessaires pour chaque ressource.

Annexe: Détails technique du nuage utilisé

Accès au nuage

Toutes les interactions avec le nuage se font via une interface Web.

URL: <http://canari.info.polymtl.ca>

Chaque équipe dispose déjà d'un compte d'utilisateur. Les noms d'utilisateur et de mots de passe sont les mêmes utilisés lors de votre séance de TP1. Vous pouvez aussi envoyer un courriel au chargé de laboratoire en spécifiant les matricules de tous les membres de votre équipe.

Réseau

Toutes les instances créées pour ce travail pratique doivent être connectées au réseau « **switch1-nat** » Pour connecter une machine virtuelle à « **switch1-nat** » à partir de la fenêtre de démarrage d'instance de l'interface Web, accédez à l'onglet « Networking » et glissez « **switch1-nat** » dans la zone « Selected Networks ». L'adresse IP associée à la machine virtuelle sera visible dans l'écran de détails de la machine virtuelle une fois celle-ci démarrée.

Pour accéder à une instance à partir de l'extérieur du nuage il faut créer une adresse IP flottante et l'associer à la machine virtuelle, comme expliqué dans le prochain paragraphe (ou comme fait dan.

Associer une ip flottante à une machine virtuelle:

Pour associer une ip flottante à une machine virtuelle, il faut suivre les étapes suivantes:

- Récupérer le fichier rc dans le dashboard, dans Projet, « Accès et sécurité », « Accès API » et l'exécuter:

```
source ./[LOGIN]-projet-openrc.sh
```

- S'allouer une ip externe (Il faut que l'instance soit connectée dans un subnet interconnecté avec le réseau externe externe1, comme « switch1-nat »):

```
openstack floating ip create externe1
```

- Ou vérifier celle qu'on a si on se l'est déjà allouée: **openstack floating ip list**
- Associer notre ip externe à une instance (Il faut que l'instance soit connectée dans un subnet interconnecté avec ext-net, comme «réseau-pour-tous»):

```
openstack server add floating ip [Nom ou ID de l'instance] [Mon_IP_flottante]
```

Accès distant à une machine virtuelle

Vous pouvez aussi accéder à cette image à l'aide de SSH. Il faut d'abord créer une paire clés d'authentification: *Accès et sécurité > Paires de clés > créer une paire de clés*

Pour se connecter à une machine virtuelle

ssh -i (clé privée) ubuntu@ip-flottante

Pour envoyer un fichier vers une machine virtuelle :

scp -i (clé privée) nom_du_fichier ubuntu@ip-flottante:(répertoire destination)

Pour envoyer un répertoire vers une machine virtuelle :

scp -i (clé privée) -r nom_du_repertoire ubuntu@ip-flottante:(répertoire destination)

Exécution de gabarits Heat

Pour exécuter un gabarit Heat à partir de l'interface Web:

1. Accédez à l'onglet « Project > Orchestration > Stacks ».
2. Cliquez sur le bouton « Launch Stack ».
3. Sélectionnez « File » comme « Template Source ».
4. Sélectionnez l'emplacement du fichier de gabarit sur votre ordinateur.
5. Cliquez sur le bouton « Next ».
6. Donnez des valeurs aux paramètres que vous avez défini dans votre gabarit.
7. Cliquez sur «Launch».

Une fois le démarrage du gabarit terminé, vous pouvez consulter ses paramètres de sortie en le sélectionnant dans l'onglet « Project > Orchestration > Stacks », puis en accédant à l'onglet « Overview ». Cela est utile pour connaître l'adresse IP à laquelle un service Web a été déployé.

Bonne chance et bon travail ! :-)