

INF4410 – Systèmes répartis et infonuagique

TP3 – Initiation aux services de l'infonuagique

Chargés de laboratoire:

Housseem Daoud

16 novembre 2015

École Polytechnique de 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: lundi le 7 décembre 2015, avant 16h, voir le plan de cours pour les pénalités de retard.
- Format: Une archive au format .tar.gz contenant:
 - Un rapport contenant les réponses aux questions.
 - Un script *install.sh* permettant d'installer Apache/MySQL/PHP sans aucune intervention humaine.
 - Un gabarit Heat pour le déploiement d'un serveur web sur OpenStack.
 - Un gabarit Heat pour le déploiement sur 2 machines virtuelles, avec répartition de charge.

Vous pouvez faire le laboratoire seul, mais le travail en binôme est encouragé.

Barème

Partie 1.1 : 4 points

Partie 1.2 : 4 points

Partie 2 : 6 points

Questions: 6 points

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

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

Spécifications

Partie 1 - Déploiement d'un serveur web dans l'infonuage

La virtualisation a beaucoup facilité le déploiement des services web. Au lieu d'avoir un serveur dédié pour chaque service, on peut lancer plusieurs machines virtuelles sur le même serveur.

La seule tâche qui reste à l'administrateur système est d'installer les services nécessaires dans chaque machine virtuelle et de configurer les interactions entre eux .

Pour faciliter cette tâche, Openstack fournit les gabarits Heat permettant de spécifier des règles pour l'allocation des différentes ressources OpenStack (machines virtuelles, adresses IP flottantes, moniteurs d'activités, alarmes...). Ces gabarits permettent aussi de lancer des commandes sur les machines virtuelles au moment de la création.

Dans la première partie de ce TP, on vous demande de déployer un serveur web (Apache, MySQL et PHP) sur OpenStack avec un gabarit HEAT.

1- Automatisation de l'installation des paquets nécessaires

On doit commencer par préparer un script Shell permettant d'installer et de configurer Apache/MySQL/PHP sur une machine sans aucune intervention de la part de l'administrateur système. Ce script va être ensuite utilisé pour automatiser le processus de création de la machine virtuelle offrant le service web sur Openstack.

Connectez vous à l'interface graphique de OpenStack en utilisant votre compte utilisateur et créer une

machine virtuelle en utilisant l'image **INF4410-Ubuntu-trusty-mini** et la flavor **INF4410-mini** et connectez vous à cette machine en utilisant la commande ssh (il faut d'abord créer une IP flottante).

Question 1 : Les paramètres par défaut de sécurité ne permettent pas aux machines externes de se connecter à la machine virtuelle. Quelles sont les règles que vous avez ajouté au groupe de sécurité *default* pour que la machine virtuelle soit accessible par ssh et pour qu'elle puisse servir les requêtes web.

La mise en place du serveur web nécessite l'installation des paquets suivants :

mysql-server, mysql-client, apache2, php5, libapache2-mod-php5, php5-mysql, phpmyadmin

Lorsqu'un paquet a des dépendances, l'outil *apt-get* nous demande si on veut vraiment les installer. Avec ce comportement, l'utilisateur doit être toujours présent pour intervenir. Il faut donc trouver un moyen pour automatiser le processus d'acceptation.

Pire encore, au cours de l'installation des paquets **mysql-server** et **phpmyadmin**, *apt-get* nous demande d'entrer un mot de passe pour configurer les bases de données. Ceci empêche l'installation d'être totalement automatique. Heureusement, Le paquet **debconf-utils** contient des outils permettant de répondre à ces questions avant le début de l'installation.

Question 2 : En prenant en considérations les difficultés expliquées ci-dessus, écrivez un script permettant d'installer Apache/Php/ MySQL sans aucune intervention de l'utilisateur (voir *install.sh* dans l'archive)

Pour tester si la configuration est terminée avec succès, on peut utiliser le navigateur et accéder aux pages suivantes (en supposant par exemple que l'adresse IP flottante de la machine virtuelle est 111.111.111.111)

- <https://111.111.111.111> permet vérifier si apache est installé
- <https://111.111.111.111/phpmyadmin> permet de vérifier si php et mysql sont installés

2- Automatisation du déploiement du serveur web dans OpenStack

Vous devez créer un gabarit Heat permettant de déployer le serveur web sur une seule machine virtuelle dans l'infonuage Openstack. La machine virtuelle doit utiliser l'image **INF4410-Ubuntu-trusty-mini** et la flavor **INF4410-mini**. Elle doit aussi être connectée sur le réseau **INF4410-01-net** pour pouvoir

accéder à l'internet.

Les gabarits Heats permettent d'exécuter des commandes sur la machine virtuelle au moment de sa création en utilisant *user_data*. Utiliser le contenu du script développé dans la partie précédente pour automatiser l'installation des paquets nécessaires.

Il est recommandé de consulter l'exemple suivant:

https://github.com/openstack/heat-templates/blob/master/hot/hello_world.yaml

Partie 2 - Déploiement d'un service Web avec répartition de charge

Vous devez créer un gabarit Heat qui permet de déployer un service Web, mais cette fois-ci avec un répartiteur de charge. Dans cette partie, on va utiliser un serveur web *server.py* simple développé en python pour simplifier le déploiement. (*server.py* offre le service http sur le port 8000)

Le champs **user_data** sera :

```
#!/bin/bash
wget http://secretaire.dorsal.polymtl.ca/~hdaoud/infonuagique/server.py
python server.py &
```

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 2 machines virtuelles différentes.

Un moniteur d'activité doit vérifier le statut des machines à intervalles réguliers. Le status INACTIVE doit être affecté à une machine si elle ne répond pas aux requêtes TCP envoyées par un moniteur d'activité. Le moniteur d'activité envoie un message à intervalles de 11 secondes aux machines pour vérifier leur activité. S'il ne reçoit pas de réponse dans un délai de 8 secondes, il envoie une seconde requête. S'il ne reçoit pas de réponse à cette seconde requête dans un autre délai de 8 secondes, il déclare la machine inactive.

Nous fournissons donc avec le travail pratique un gabarit dans lequel devez modifier les sections identifiées par un commentaire « # À compléter ». Vous pouvez aussi vous inspirer de l'exemple suivant:

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

Questions

Question 1: OpenStack est en fait un regroupement de plusieurs composants distincts provenant originalement d'autres projets. On vous demande de décrire dans vos mots les composants utilisés dans ce travail pratique: *Heat*, *Neutron* et *Nova*.

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::HealthMonitor
- OS::Neutron::Pool
- OS::Neutron::LoadBalancer
- OS::Nova::Server

Question 3: Nous avons créé 2 machines virtuelles qui demeurent actives en tout temps pour offrir un service Web. Si nous devons 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 OS::Heat::AutoScalingGroup afin de pouvoir modifier dynamiquement le nombre d'instances du serveur.

Quels 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://pingouin.info.polymtl.ca/dashboard>

Chaque équipe dispose d'un compte d'utilisateur. Les noms d'utilisateur et mots de passe seront distribués lors de votre première séance de TP. 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 «demo-net». Ce réseau est virtuel. Il est impossible d'y accéder à l'extérieur du nuage. Cependant, vous pouvez sans problème utiliser des adresses IP de ce réseau pour communiquer entre les machines du nuage.

Pour connecter une machine virtuelle à «demo-net» à partir de la fenêtre de démarrage d'instance de l'interface Web, accédez à l'onglet «Networking» et glissez «demo-net» 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.

Images disponibles

Deux images sont disponibles pour vos machines virtuelles.

INF4410-Ubuntu-trusty-mini

Cette image doit être utilisée pour déployer le service Web.

L'image fonctionne uniquement avec la *flavor* "**INF4410-mini**".

Python est préinstallé sur cette image.

Il n'est pas possible d'utiliser la console Web pour accéder aux instances utilisant cette image. Pour exécuter des commandes sur ces instances, vous devez les spécifier lors de leur démarrage. Si vous

lancez l'image avec un gabarit Heat, spécifiez les commandes dans la propriété «user_data».

Vous pouvez aussi accéder a cette image en utilisant des clé ssh pour faire des essais (partie1 – question1)

cirros-0.3.3-x86_64

L'image est très minimale et peut être démarrée avec la *flavor* "**INF4410-micro**" pour éviter de surcharger le nuage.

Il est possible d'afficher la console de ces images dans un navigateur Web. Cliquez simplement sur «More > Console» dans l'onglet «Compute > Instances» de l'interface Web.

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.