École Polytechnique de Montréal

INF8480 - Systèmes répartis et infonuagique

TP3 - Initiation aux services de l'infonuagique

Alexandre Hua (1795724)
Salim Eid (1786419)

Groupe 01

Remise le 4 Décembre 2018

Observation

Nous avons réalisé des tests de performance en envoyant 20 requêtes simultanément. Le temps moyen que nous avons obtenu pour le serveur Web simple est de 0.509509s et celui avec un répartiteur de charge est de 0. 251239s.

On remarque donc clairement que le serveur avec répartiteur est plus rapide vu que le temps moyen est à peu près 2 fois plus petit. Cela peut s'expliquer par le fait deux serveurs exécutés en parallèle qui répondent aux requêtes du client permet de traiter une charge de travail 2 fois plus élevé qu'un serveur seul dans le même intervalle de temps.

Initialement, nous avons dû effectué nos tests avec 30 requêtes, mais à cause de la surcharge au niveau du nuage et du fait que tous les autres équipes effectuent leurs tests en même temps, le taux d'erreur devient trop élevé.

Question 1

Neutron :
Ça permet de gérer le réseautage de Openstack. Sous forme de « network connectivity as a service » permettant . Il permet également de fournir des services de détection d'intrusion, pare-feu, VPN et équilibrage de charge, ainsi que contrôler des accès.

Heat :
Il s'agit d'un service de formatage et de recette pour configurer des paramètres d'orchestration et de déploiement d'un nuage.

Nova :
Il s'agit d'un ordonnanceur qui gérer les instances de serveur de calcul.

Cinder :
Il s'agit d'un système de gestion de stockage dans Openstack. Il permet de créer et allouer des volumes et de les associer à des instances. Un cas d'utilisation possible est de créer un type de volume spécifique et d'instancier un volume de ce type. Par exemple, il est possible de créer un volume de type encrypté.

Horizon :
Il s'agit d'une interface utilisateur pour gérer et interagir avec les différents services offert par Openstack. Le TP1 est un excellent exemple de cas d'utilisation. En effet, nous nous sommes servis de l'interface de Openstack pour créer des instances machines virtuelles.

Question 2

OS::Heat::ResourceGroup :
Cela permet de créer plusieurs instances d'une ressource (le serveur) configuré de la même façon.

OS::Neutron::HealthMonitor :
Il permet de surveiller l'état des serveurs en marche.

OS::Neutron::Pool :
Ça permet de séparer les nœuds de serveur du répartiteur. Ca permet également de spécifier la façon dont la charge est répartie.

OS::Neutron::LoadBalancer :
Il s'agit du composant qui permet de diriger le trafic entre les serveurs.

OS::Nova::Server :
Ça permet de gérer l'instance de la machine virtuelle dans Openstack.

Question 3

1)
Une alternative au OS::Heat::ResourceGroup est d'utiliser OS::Heat::AutoScalingGroup/Policy.

Les paramètres obligatoires pour le AutoScalingGroup sont le nombre maximal de ressource (max_size), le nombre minimal de ressource (min_size) et une définition de la ressource (resource).

Les paramètres obligatoires pour le ScalingPolicy sont le type d'ajustement (ajustment_type), le groupe à mettre à échelle (auto_scaling_group_id), la taille d'ajustement (scaling_ajustment)

2)
- Il existe un composant OS:Aodh:GnocchiAggregationByResourcesAlarm qui permet de lancer une alerte sous condition. On peut donc configurer pour que l'alerte soit lancé lorsque l'utilisation de CPU dépasse un certain seuil.

Les paramètres obligatoires sont les métriques à surveiller (metric), une requête pour filtrer les métriques (query), le type de ressource (resource_type), le seuil sur les métriques (threshold)

- Pour changer le nombre d'instance de serveur selon une alarme, nous pouvons utiliser une alarme donc on peut la mapper avec une action dont il s'agirait d'une mise à l'échelle de ScalingPolicy.

Les paramètres optionnels supplémentaires du composant alarme à utiliser est alarm_action.