Projet Cloud Infra / Cloud Usage

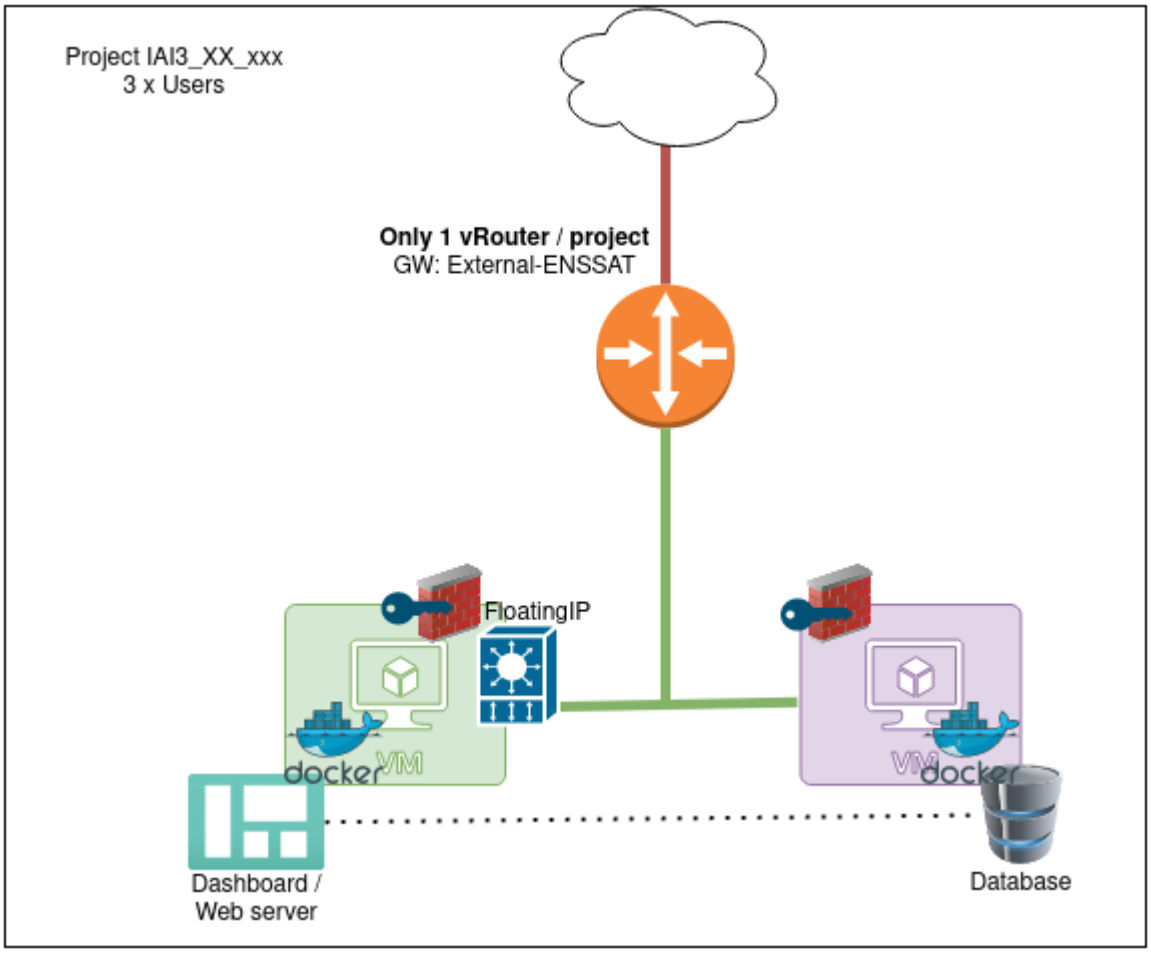
Novembre / Décembre 2023

Contexte

Vous allez mettre en œuvre les différentes notions vues au travers des TP pour déployer une application sous forme de containers docker, dans une infrastructure virtualisée également déployer par vos soins sur la plateforme de TP OpenStack.

Afin de rendre le déploiment de cette application pertinent dans une infrastructure de type cloud, cette application devra se composer d'un frontend et d'un backend.

Voici un schéma résumant le déploiement qui sera mis en œuvre:



1. Deploiement de l'infra (OpenStack HEAT)

Vous partirez d'un tenant OpenStack remis à zero. Vous conserverez les groupes mis en place lors des TPs.

Vous mettrez en place l'infra structure virtualisée **via un template HOT** (Heat Orchestration Template) accompagné de son fichier d'environnement.

Le fichier template devra **à minima** comporter la **création** (au sein de votre tenant) des ressources suivantes:

- Stockage:
  - 1 x volume (2 Gb, + si besoin)
- Réseau:
  - 1 x Réseau
    - 1 x subnet mappé sur le réseau précédemment créé
    - 2 x ports (Frontend/Backend) sur ce réseau, chacun sera utilisé par une VM
  - 1 x Router
    - external GW="External-ENSSAT"
    - Création d'un port sur ce router (=GW du subnet créé en amont)
  - 2 x Security groups
    - Sec.Group Frontend:
      - port TCP 22, TCP 80, TCP 443 en ingress. Rajouter d'autres ports en fonction des besoins de l'appli (et du filtrage ENSSAT). Ces ports seront accessibles via la Floating IP associée à la VM.
    - Sec.Group Backend:
      - Pas de besoin particulier en ingress. Les ports seront accessibles sur le réseau interne, mais ne seront pas accessibles de l'extérieur (car pas de floating IP sur la VM correspondante).
      -
- Instances:
  - 1 x VM Frontend
    - image = debian10-docker (ID=c19f8c61-20da-4c08-b249-2005ba0efd10)
    - flavor m1.mini
    - user/passwd OU keypair dédiée (créée en amont via la webGUI)
    - port réseau Frontend créé en amont
    - security group Frontend créé en amont

debian10-docker_insec-reg

- 1 x VM Backend
  - image = debian10-docker (ID=c19f8c61-20da-4c08-b249-2005ba0efd10)
  - flavor m1.mini
  - user/passwd OU keypair dédiée (créée en amont via la webGUI)
  - port réseau Backend créé en amont
  - security group Backend créé en amont
  - Utilisation du Volume créé en amont.
  - Utilisation de userdata pour provisionner cette VM
    - user/passwd (sans Floating IP, cette VM ne sera accessible que par rebond via la VM Frontend)
    - Management du volume
    - Provisionning de datas pour docker.

- Floating IP:
  - Allocation 1x Floating IP
  - Association de cette floating IP à la VM Frontend
  - Verification de l'association (section 'output' du template)

- Keypair:
  - Son utilisation est optionnelle si vous mettez en place une authentification user/password via cloud-init sur vos VMs
  - Si vous l'utilisée, pas besoin de la créer via le template Heat. Vous utiliserez une keypair déjà existante, et vous pointerez sur cette clef dans le fichier environnement de HEAT

Remarque:
Si besoin, vous pouvez rajouter des volumes, réseaux/ports, instances, utiliser différentes flavors, etc...
L'idée est d'ajuster/optimiser les ressources utilisées par l'application qui sera déployée au moyen de containers dans ces VMs

2. Déploiement de l'application (docker-compose)

L'image debian utilisée coté openstack intégre docker et docker-compose. Cela vous permet de pousser votre déploiement Docker à partir du template HEAT.

Pour ce faire, vous pouvez notamment creuser l'utilisation des ressources HEAT suivantes, qui sont un point d'entrée pour une utilisation avancée des userdata via OS::Nova::Server
- http://148.60.225.51/project/resource_types/OS::Heat::MultipartMime/
- http://148.60.225.51/project/resource_types/OS::Heat::SoftwareConfig/

L'utilisation d'un script bash via user-data est également envisageable pour pousser le déploiement docker.

Dans la mesure du possible, vous essaierez de mettre à profit vos précédentes expériences en terme de développement web vis-à-vis de l'application déployée.

Vous pouvez vous appuyer sur cet exemple de déploiement d'une stack LAMP pour la mise en œuvre de votre déploiement : https://github.com/sprintcube/docker-compose-lamp

Dans la mesure du possible et en vous inspirant de l'exemple en lien, vous mettrez en place au sein du docker-compose les resources docker suivantes:
- build
- volume
- variables d'environnement
- exposition de ports
- links/extra-hosts

Livrables

Vous remetterez une archive comprenant tout le nécessaire (template HOT+env, docker-compose.yml, …) pour le déploiement de votre solution, ainsi qu'un README la décrivant.

Le déploiement que vous proposerez sera joué 'from scratch' pour son évaluation.