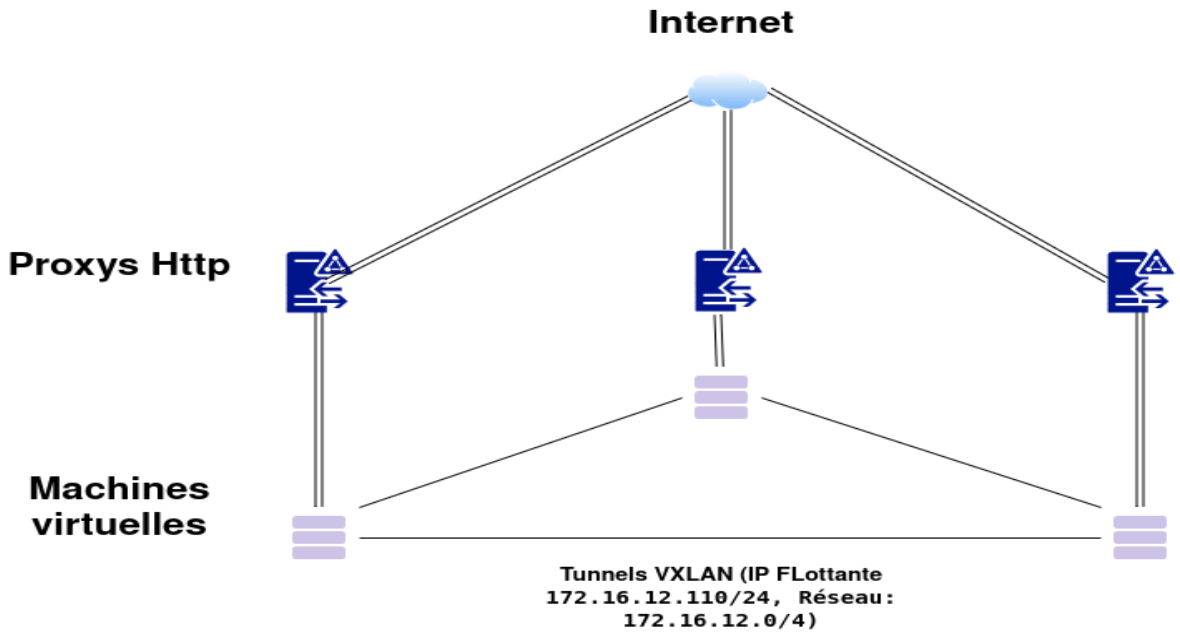
Cloud & Virtualisation

Projet Cloud

Cavalier Killian - Brelot Julien

Infrastructure :



Le projet vise à déployer une application permettant aux utilisateurs d'héberger et de redimensionner des images de manière automatisée et résiliente. L'application se compose d'un backend pour sauvegarder et servir les images, d'un worker pour redimensionner les images, et d'un frontend sous forme d'interface web. Les images sont stockées dans un espace de stockage compatible S3.

L'infrastructure cloud fournie comprend un stockage S3, une file de message RabbitMQ, un serveur Consul et un serveur Nomad, ainsi que trois machines virtuelles avec des tunnels HTTP vers une IP flottante. Les machines virtuelles sont accessibles via un hôte bastion et sont connectées via un tunnel VXLAN offrant un réseau privé.

Afin d'apporter de la souplesse et de la robustesse, nous voudrions intégrer un load balancer pour équilibrer et partager la charge entre les différentes VM. De même, il est plus intéressant de déployer un serveur Consul-Nomad sur chacune des machines virtuelles. Par des algorithmes d'élection, elles éliront entre elles un leader. Ce faisant, cela permettra une robustesse accrue à l'infrastructure. En effet, en cas de panne d'une ou de deux machines, le service web sera toujours accessible. Effectivement, avec la configuration initiale, si seulement une seule machine dispose du serveur Consul-Nomad, alors si elle tombe en panne, le service sera interrompu.

Dans cette même approche de résilience, il faut donc que chaque machine virtuelle puisse lancer un job worker, back-end ou frontend, avec au moins une instance de chaque application exécutée. Ainsi, en cas de panne, n'importe quelle machine pourra prendre le rôle et lancer un job

Pour automatiser le déploiement, nous avons choisi Ansible qui permet d'installer et de configurer plus souplement et plus aisément les services. De plus, c'est facilement adaptable à une autre architecture, il suffit de modifier quelques variables et de définir les hôtes.

De plus, Ansible est très répandu et il est facile d'obtenir de la documentation et des exemples de développement pour des projets similaires au nôtre. Il est assez aisé de le prendre en main.

Enfin, le fournisseur utilisant également Ansible, dans un souci de cohérence est de simplicité, il est de bon ton de l'adopter, d'autant que la plupart des fichiers de configuration pouvaient être repris.

Ainsi, si une mise est faite, on peut relancer une commande pour mettre à jour les services. On pourrait procéder un service après l'autre, de sorte à ce que les autres machines puissent maintenir le service et pour que la mise à jour soit transparente pour l'utilisateur.

Le service web est, quant à lui, une image exécutée dans un Docker. Il y a 2 images qui garantissent le fonctionnement du service web pour l'utilisateur : le front-end et le back-end.
Ces 2 images sont le fruit de la création d'une image Docker à partir des fichiers des branches front-end et back-end, de notre repository git.
Ainsi, pour une mise à jour du service, un développeur pourrait push une nouvelle version stable de son code, puis créer une image docker, la tagger dans le repo distant docker.io/jbunistra/cloud-frontend:latest ou docker.io/jbunistra/cloud-backend:latest. De plus, il est même possible d'automatiser la mise à jour des images docker via des jobs sur gitlab.

Enfin, l'image étant récupérer à la création des jobs, une fois que l'image à été modifié, on peut détruire un job et le relancer sur un nœud pour que le job offre le service web réactualisé.

Il reste néanmoins à installer Ansible et Python sur la machine qui lance le déploiement.
Ansible ne dépend de rien d'autre, il ne nécessite qu'une connexion ssh.
Ainsi, pour lancer le déploiement d'Ansible, nous avons dû installé Ansible sous ubuntu via le script très simple install-ansible.sh. Ensuite, il a fallu créer un fichier ~/.ssh/id_vm, contenant la clef privé octroyé par le fournisseur cloud pour permettre la connexion ssh entre racaillou et gravalanch/grolem.C'est un souci qui peut se poser pour le déploiement si on n'a pas d'accès ssh.

Notons que par manque de temps,nous n'avons pas configuré de rôle keepalived, l'ip virtuelle est ajoutée en ligne de commande ce qui diminue la souplesse du développement. De plus, la configuration ansible n'est pas totalement opérationnelle. Néanmoins nous avions pu déployer 1 serveur consul et des nœuds Nomads qui permettaient la réalisation d'un job intégrant un répartiteur de charge, un proxy, le service web ainsi qu'un healthcheck.