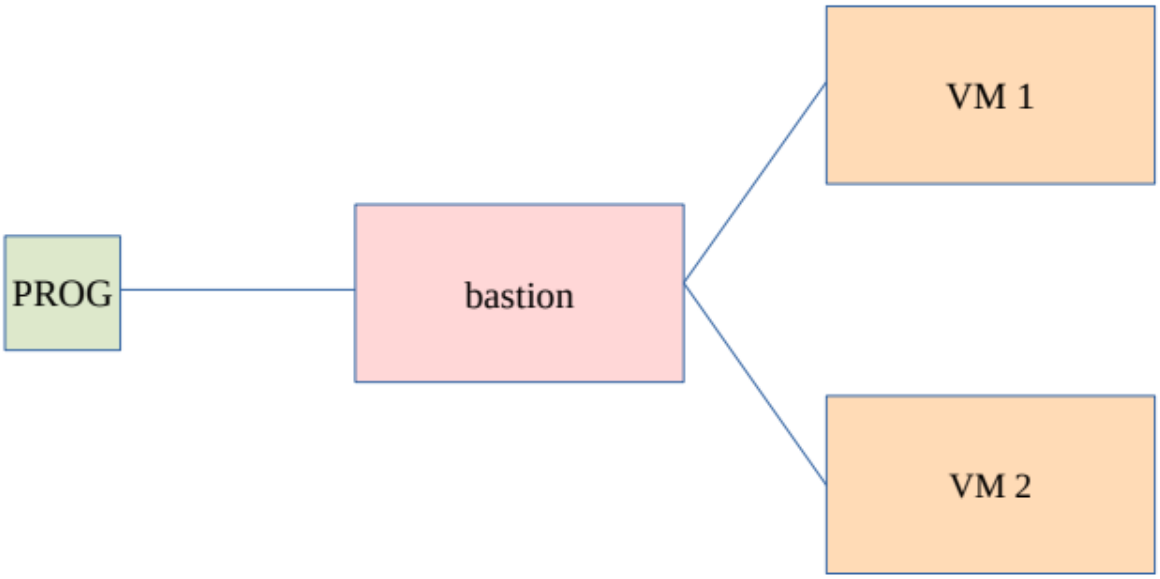
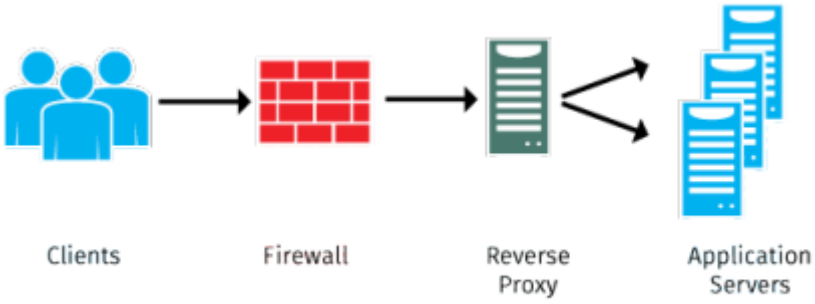


Ici on va discuter de la méthode de communication entre les VM et le monde extérieur, on ne traitera pas la communication entre les programmes et la BDD.
Les VM ont un adressage réseau privé (en 192.168.x.y), elles ne sont donc pas directement accessibles depuis l'Internet. On passe donc par un bastion qui possède une adresse IP publique et qui sert de relais pour accéder à ce réseau privé.
On accède au bastion par ssh et donc on accède au VM (redirection automatique) ;
Le bastion sert de reverse proxy (une machine qui se met entre les serveurs où les programmes sources tournent et le client) pour les serveurs web que vous hébergeriez sur vos VMs.



Pour accéder à un serveur web qui écouterait sur le port 80 d'une VM, il faut accéder à l'url **https://IP_du_bastion/nom_de_la_vm/ (le https et les / sont nécessaires).** Une alerte de sécurité de votre navigateur web pourrait se présenter au premier accès. Sur Firefox vous pouvez passer outre en cliquant sur "Avancés" puis "Acceptez le risque et poursuivre".
Pour les autres navigateurs, référez vous à la documentation concernant l'acceptation des certificats SSL auto-signés. Le but principal du bastion (reverse proxy) est de garantir la sécurité de nos machines.

Choix du logiciel serveur:

Un « serveur web » peut faire référence à des composants logiciels (software) ou à des composants matériels (hardware) ou à des composants logiciels et matériels qui fonctionnent ensemble.

1. Au niveau des composants matériels, un serveur web est un ordinateur qui stocke les fichiers qui composent un site web (par exemple les documents HTML, les images, les feuilles de style CSS, les fichiers JavaScript) et qui les envoie à l'appareil de l'utilisateur qui visite le site. Cet ordinateur est connecté à Internet et est généralement accessible via un nom de domaine tel que google.com.

Dans notre cas :

- Les machines (matériel ordinateur) sont les VM fournies par la fac ;
- Accessibles via https://IP_du_bastion/nom_de_la_vm/.

2. Au niveau des composants logiciels, un serveur web contient différents fragments qui contrôlent la façon dont les utilisateurs peuvent accéder aux fichiers hébergés. On trouvera au minimum un serveur *HTTP*. Un serveur HTTP est un logiciel qui comprend les URL et le protocole HTTP (le protocole utilisé par le navigateur pour afficher les pages web).
Dans notre cas, il faudra décider de quel logiciel qui comprend un protocole HTTP utiliser.

On a fait de la programmation bas niveau et moyen niveau (jusqu'à la couche SESSION), on ne peut pas juste executer un fichier C et écouter dans une adresse ou se mettre à envoyer des paquets partout. On a besoin d'un protocole HTTPS qui est un protocole de communication client-serveur, sécurisé et qui nous propose des méthodes pour réaliser des actions comme la demande de ressource (GET), transmission de données (POST) ...

Détails : *Au niveau le plus simple, à chaque fois qu'un navigateur a besoin d'un fichier hébergé sur un serveur web, le navigateur demande (on dit qu'il envoie une requête) le fichier via HTTP. Quand la requête atteint le bon serveur web (matériel), le serveur HTTP (logiciel) renvoie le document demandé, également grâce à HTTP.*

Nginx et Apache sont des serveurs web populaires utilisés pour fournir des pages au navigateur d'un utilisateur.

	Apache	NGINX
Architecture	Une approche basée sur les processus : Cela consiste à créer un nouveau fil ou *thread* pour chaque requête.	Une approche basée sur les évènements : Cela consiste à traiter plusieurs requêtes dans un seul thread.
Les modules complémentaire	**60 modules** sont téléchargeables et peuvent être activés ou désactivés à tout moment.	Modules disponibles par des tiers (non chargeables dynamiquement)

Performance (contenu statique)	Utilise la méthode basée sur les fichiers pour traiter un contenu statique.	Surpasse Apache dans la gestion du contenu statique
Performance (contenu dynamique)	Traite le contenu dynamique au sein du serveur.	Ne traite pas les contenus dynamiques.
Interprétation des requêtes	Passe l'emplacement du système de fichiers ou FSL (file system linux).	Passe par URI (Uniform Resource Identifier) pour interpréter les requêtes.
Sécurité	Grande sécurité, il offre des astuces de configuration pour la gestion des attaques par Déni de service, tout comme le module *mod_evasive* pour répondre aux attaques http, DoS, DDoS ou autres types.	Une meilleure sécurité avec une base-code plus fine. Le code de base de NGINX est toutefois significativement plus petit, donc celui-ci est plus avantageux en termes de sécurité. NGINX a aussi une liste de balises sécuritaires.

Sources :

Qu'est-ce qu'un serveur web ? - Apprendre le développement web | MDN
Comparatif Apache vs Nginx 2020 Points Forts et Faibles. Lequel choisir
What is Apache? In-Depth Overview of Apache Web Server | Sumo Logic