Documentation technique

Projet « Train Commander » Supinfo 2015 – 2016

Sommaire

[Choix des technologies 2](#_Toc453165295)

[Présentation de l’infrastructure 3](#_Toc453165296)

[Présentation 3](#_Toc453165297)

[Fonctionnement des services 4](#_Toc453165298)

[Fonctionnement du Content Delivery Network 4](#_Toc453165299)

[Fonctionnement de l’architecture interne 4](#_Toc453165300)

[Fonctionnement du Domain Name Server 5](#_Toc453165301)

[Fonctionnement du serveur mail 5](#_Toc453165302)

[Fonctionnalités bonus 6](#_Toc453165303)

[SSL 6](#_Toc453165304)

[Réception de mail avec interface graphique 6](#_Toc453165305)

[Load-balancing 6](#_Toc453165306)

# Choix des technologies

Nous avons opté dès le début du développement pour une solution performante et avant tout gratuite. Ces raisons nous ont ainsi dirigés vers le monde de l’open-source.

En ce qui concerne le système d’exploitation des machines virtuelles, nous avons choisi la distribution Linux Debian. L’expérience de notre équipe avec cette distribution nous a orientés vers celle-ci. De plus, la popularité de cette dernière nous a apportés une grande flexibilité.

Pour les mêmes raisons, la sélection du serveur Apache 2 pour le serveur web et MySQL pour la base de données s’est montrée très spontanée.

Nous avons choisi Bind9 pour la création du DNS car, en plus de l’utilisation de machines Debian, nous avons également, au cours de l’année, eu l’occasion de l’utiliser et avons décidé de mettre à profit cette expérience.

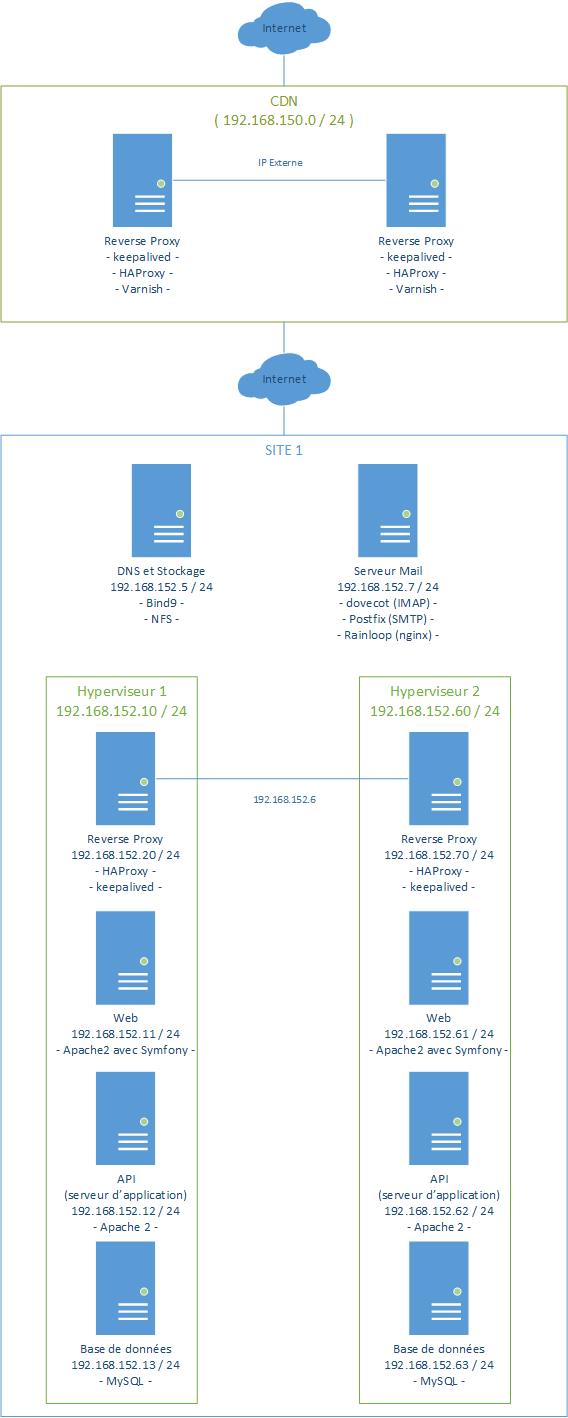
Pour ce qui est de la virtualisation, nous avions besoin d’une solution gratuite et efficace. Notre choix s’est donc naturellement porté sur XenServer, et ce pour de nombreuses raisons. En effet, cette plateforme de virtualisation est open-source, très stable, simple d’utilisation et a de meilleures performances, comparée à ses concurrentes, dans la gestion de machines Linux. De plus, son interface graphique et ses nombreuses fonctionnalités (déplacement et/ou copie de machine(s) virtuelle(s),…) permettent une administration d’autant plus aisée qu’elle est gérable à distance. La résolution des problèmes possibles se fait grâce à son support actif et sa très grande communauté.

Quant à la mise en cluster, nous nous sommes tournés vers Keepalived et HAProxy pour la répartition de la charge. Ces deux solutions répondent parfaitement à nos besoins, autant pour les hyperviseurs que pour le Content Delivery Network. En effet, de par leur grande stabilité et leur excellente réputation, elles sont très souvent utilisées ensemble en production.

Pour les serveurs proxy qui composent le CDN, nous utiliserons Varnish pour sa simplicité de mise en place ainsi que pour la gestion supérieure de son cache, par rapport aux autres possibilités.

# Présentation de l’infrastructure

## Présentation



# Fonctionnement des services

## Fonctionnement du Content Delivery Network

Le Content Delivery Network (ou CDN) est composé d’un cluster de proxys inverses, combiné à une répartition de charge.

Le client accède, depuis l’extérieur, à l’infrastructure par une adresse IP virtuelle. Cette VIP est mise en place par Keepalived. Elle est ensuite « écoutée » par HAProxy. Celui-ci va répartir la charge entre les serveurs proxy.

Varnish va ensuite gérer son cache et envoyer les requêtes nécessaires au serveur web.

## Fonctionnement de l’architecture interne

Le serveur proxy, quant à lui, accède au cluster d’hyperviseur, XenServer. La VIP est également mise en place par Keepalived.

Chaque hyperviseur contient un serveur web (Apache2), une Web API et une base de données (MySQL).

Chaque serveur a sa charge répartie par HAProxy.

Cet équilibrage offre une forme de haute disponibilité et de distribution de charge au sein même du cluster d’hyperviseurs. Ainsi si, par exemple, la base de données du premier hyperviseur tombe, le flux de données sera envoyé sur la deuxième base de données au sein du second hyperviseur.

Nous utilisons différents ports pour permettre à HAProxy d’écouter et de rediriger vers les bons serveurs.

Le serveur web est disponible grâce au service Apache 2.

L’application métier est, quant à elle, proposée par la web API. Elle permet d’effectuer les calculs pour les trajets, confirmer l’authentification de l’utilisateur, etc.

Les deux serveurs cités ci-dessus accèdent à la base de données MySQL afin de stocker toutes les informations nécessaires (les différentes stations, l’historique des trajets, …).

## Fonctionnement du Domain Name Server

La configuration du Domain Name Server (ou DNS) est segmentée en deux zones : une extérieure et une interne.

La première est ici appelée « train-commander.fr ». Elle permet de retrouver le site et l’API pour des clients extérieurs à l’infrastructure.

La seconde, nommée « train-commander.lan », gère les noms des serveurs internes à la solution.

Cette séparation offre une sécurité supplémentaire ainsi qu’une meilleure organisation de la configuration de ce service.

## Fonctionnement du serveur mail

Le serveur mail a été mis en place pour répondre au besoin de l’équipe de développement.

Pour cela, le protocole SMTP a été implémenté avec Postfix afin d’envoyer les mails.

# Fonctionnalités bonus

## SSL

La demande du client d’inclure le paiement possible par Paypal a soulevé la nécessité de sécuriser les connexions entre le client et les serveurs.

Nous avons donc eu recours à l’HTTPS pour les connexions aux serveurs web.

Nous avons opté de même pour l’IMAPS et au SMTPS pour le serveur mail.

## Réception de mail avec interface graphique

Nous nous sommes également permis de rajouter le protocole IMAP (pour la réception de mail) ainsi que l’interface utilisateur Rainloop, afin de mettre à disposition un système de boite mail. Cela pourrait permettre au client la mise en place d’un service de support, par exemple.

## Load-balancing

Comme expliqué dans la partie concernant l’architecture interne, nous avons mis en place une répartition de charge pour les différents serveurs contenus dans le cluster d’hyperviseurs.

Bien que la haute disponibilité fût clairement spécifiée, nous avons jugé utile d’aller plus loin que le simple mode « actif-passif ». Cela permet une meilleure utilisation des ressources disponibles et de meilleures performances pour l’utilisateur.

Vous trouverez aux côtés de cette documentation technique un manuel de déploiement « pas-à-pas ».