|  |  |
| --- | --- |
| **FIEDORCZUCK Dominik**  **MIRICA Constantin**  **SHATSKIY Filipp** | **Coordinateur : VAN DEN SCHRIECK Virginie**  **E.P.H.E.C. Louvain-la-Neuve**  **Groupe 20 - 2TL2**  **16.05.2018** |

Rapport technique

# **Liens outils**

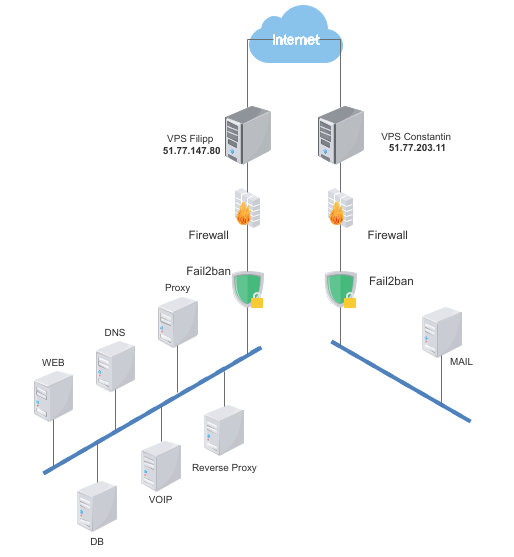
Afin d’atteindre un niveau un fonctionnement optimal et complet, nous avons choisi d’utiliser GitHub et Docker Hub pour partager les fichiers et images Docker nécessaires pour notre projet. Nous envoyons les fichiers sur le git du groupe, et tout est ensuite transmis sur la Docker Hub qui crée automatiquement l’image.

<https://github.com/fshatskiy/ProjetAdminSys-R-seaux>

<https://cloud.docker.com/u/adminprojet/repository/docker/adminprojet/admin_projet_repo>

# **Schémas**

## **Schéma infrastructure**

La zone DMZ est destinée aux serveurs qui vont sur internet. En revanche, le serveur DB ne doit pas y aller car il est le plus protégé ; ce dernier et le reste de l’entreprise font partie du réseau sécurisé interne.

*Les sites (www, b2b) sont tous sur le même container et cela a des avantages et des désavantages :*

*+ La sécurité est plus facile à mettre en place car il faut sécuriser un seul container*

*+ Des sauvegardes simplifiées*

*+ Gestion plus facile*

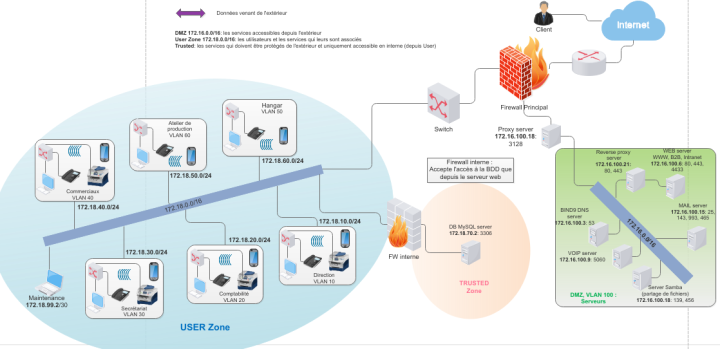
* *Il faut ouvrir plusieurs ports, donc plusieurs failles potentilles*
* *Plusieures adresses IP*

Pour l’adressage, nous avons réfléchi en fonction du nombre de travailleurs se situant dans chaque département de l’entreprise ainsi que de leurs besoins (VOIP, appareils). C’est aussi pour ces raisons que les subnets sont réalisés d’une telle manière.

## 

## **Schéma prototype**

Pour accéder à nos serveurs de test, il faut passer par un pare-feu mais aussi par la sécurité Fail2ban afin d’empêcher les attaques brute force. Nous avons mis les serveurs Web – Dns et Voip sur le VPS de Filipp car nous voulons utiliser le serveur de Dominik en tant que copie identique au cas où le premier viendrait à tomber. Cette fonctionnalité est encore à réaliser.



# **Choix et difficultés**

Durant les différentes phases de ce projet, nous avons rencontré certains problèmes avec les services. Cependant, avec l’aide des différents supports et à grâce à nos choix, nous avons pu en résoudre la quasi-totalité.

## **Web**

* Affichage de la page PHP – le service ne démarrait pas – il a fallu le lancer par Dockerfile
* IPv6 – il n’y avait pas moyen d’implémenter IPv6 *– de manière non persistante. Lorsqu’on redémarre les VPS il ne sera plus d’actualité. Il fallait rajouter l’adresse IPv6 dans eth0. On a rajouté des routes dans eth0 comme ça lorsqu’on ping en IPv6 il passe de manière automatique par le default Gateway.*
* Comprendre NGINX – lire beaucoup de documentation
* *Accès intranet depuis le conteneur Web – mettre l’intranet dans le conteneur Reverse proxy*

## **DNS**

Pour ce service, nous avons choisi Bind9 car il est bien documenté et représente une solution stable dû à son énorme communauté et au fait qu’il est le serveur DNS le plus utilisé au monde. Dans le passé, ses plus grands désavantages étaient ses vulnérabilités qui ont par la suite été corrigées dans la version 9 de BIND.

* Mise en place de glue record
* Image bin9 ne montre pas les logs – *limiter la taille des logs avec « logrotate »*
* La propagation du DNS – *résolution du problème de configuration du fichier named.conf.local*

## **DB**

MySQL est le choix direct, considérant le fait qu’il soit développé avec l’entreprise Oracle, ce qui le rend extrêmement performant, mis à jour, et par conséquent, fiable. La plupart de ses « compétiteurs » ne sont pas aussi bien documentés que MySQL.

* *Erreur « access denied » lors de la tentative de connexion aux sites dynamiques. – Il a fallu commenter la ligne « bindaddress » dans mysql.conf*

## **MAIL**

Le choix du Postfix est justifié par la documentation complète du serveur et par les nombreuses commandes de débogage.

* Création d’une image valide – *le mauvaise compréhension du Docker a posé problème, mais suite à une lecture du cours la solution a été trouvée*
* Le Docker ne se faisait pas suite à une erreur dans un fichier – *à cause de l’erreur dans le fichier « main.cf » le Dockerfile ne fonctionnait pas (EXIT89 erreur à cause d’une accolade pas fermée)*
* Envoyer automatiquement le Git vers Docker - *tutoriel pour résoudre*
* La communication avec le DNS – *la zone MX n’était pas bien établie dans le serveur DNS*
* Trouver les erreurs dans la configuration de base – Pas encore fait

## **VOIP**

Pour le service Asterisk, car il est gratuit et qu’il s’agit d’un des seuls qui fonctionne très bien. *Nous avons la possibilité de contacter et d’être contacté via n’importe quel PC et d’une connexion internet. Il est très facile à utiliser mais aussi open-source ce qui le différentie énormément des autres PABX.*

* Nous avons eu un problème dans la connexion de deux réseaux, ce problème est à moitié résolu, compte tenu que nous n’entendons pas un des deux appelants des réseaux différents

## **Validation**

**DNS** : pour valider le service DNS*, nous avons interrogé le serveur grâce aux commandes « dig » « nslookup » et finalement « nmap » pour vérifier les ports. Nous nous sommes aussi connectés sur les sites externes, et pour celui en interne, nous avons accès via Elinks, installé dans le VPS.*

***Web****: pour la validation de la config NGINX on tape « nginx -t » pour voir si la configuration est bonne, sinon on vérifie au niveau des logs et pour voir si les requêtes du Reverse proxy passent bien, on regarde access.log .*

***Reverse******proxy****: la même chose que pour le web sauf que pour bien voir les requêtes de l’extérieur passent bien par le reverse proxy on regarde access.log .*

***Proxy****: on a bloqué deux sites dans configuration du Squid et nous avons ouvert le port et nous sommes rentrés sur internet, via notre proxy au lieu du Proxy Proximus.*

***Ipv6****: cette fonctionnalité a été testée en regardant les routes ouvertes et en faisant un ping IPv6 sur Google. Ensuite on l’ajoute dans le DNS et nous accédons à nos sites via l’IPv6.*

***Mail :***

*Le mail est vérifié grâce aux différents sites créés dans ce but :*

[*https://mxtoolbox.com/SuperTool.aspx?action=smtp%3awt20.ephec-ti.be&run=toolpage#*](https://mxtoolbox.com/SuperTool.aspx?action=smtp%3awt20.ephec-ti.be&run=toolpage#)[*https://hunter.io/verify/secretariat@wt20.ephec-ti.be*](https://hunter.io/verify/secretariat@wt20.ephec-ti.be)[*https://www.ultratools.com/tools/emailTestResult*](https://www.ultratools.com/tools/emailTestResult)

*Pour utiliser ces sites, il suffit d'entrer l'adresse email et d'attendre que les résultats du test apparaissent.*

*Vérifications faites :*

* *Vérification de la syntaxe*
* *Si le format est incorrect, il sera signalé pourquoi. Par exemple : caractères non pris en charge, @-signe manquant.*
* *Recherche DNS*
* *Tout problème de recherche de domaine (la partie après le signe @) dans DNS (Domain Name System) sera signalé.*
* *Vérification SMTP*
* *Si possible, une connexion SMTP sera établie avec un serveur de messagerie responsable du courrier électronique. Tous les problèmes sont rapportés.*

Le **VoIP** fut le plus simple à tester car nous avons juste eu à faire des appels sur un de nos portables.

Même si nous n’avons pas de vérifications automatiques, tous nos services ont été validés et vérifiés en détails. *Nous nous sommes assurés que tout fonctionne bien grâce aux différentes commandes de débogage et aux manipulations expliquées plus haut.*

## **Monitoring**

Le Fail2Ban est déjà une forme de monitoring pour la sécurité, en tenant compte que nous recevons des mails lors des tentatives de connexions échouées. Pour le reste, des moyens de monitoring doivent encore être mis en place. *Nous vérifions nos servies en les utilisant mais aussi via des sites de test. L’inconvénients étant que si nous n’utilisons pas un service, nous ne savons pas nous rendre compte d’une éventuelle anomalie.*