**WoodyToys – Groupe 7 – Rapport technique**

**Bilan**

Malgré une compréhension au départ difficile de la logique de Docker, le problème fut vite réglé après des recherches approfondies sur le sujet.   
Les services mis en places sont (presque) totalement opérationnels et aurait pu l’être totalement si une meilleure gestion du temps avait été appliquée.

**Méthodologie**

Les différents services ont été implémentés un par un. Un nouveau service n’était pas mit en place tant que le précédant n’était pas fonctionnel. L’utilisation des logs permettait une gestion efficace des bugs et donc une avancée constante. Une fois qu’un service était opérationnel, celui-ci était push sur le Dockerhub permettant ainsi une récupération aisée en cas de perte ou de disfonctionnement. Un unique upload final des configuration a été fait sur github, des sauvegardes des fichiers étant faire régulièrement sur ma machine personnelle. Différents autres outils tels que nmap, dig,… ont permit de mettre le doigt sur les éventuels problèmes rencontrés.

**Infrastructure et services**

Les infrastructures et les différents services sont consultable directement dans le rapport client.

**Schémas réseaux et plans d’adressage**

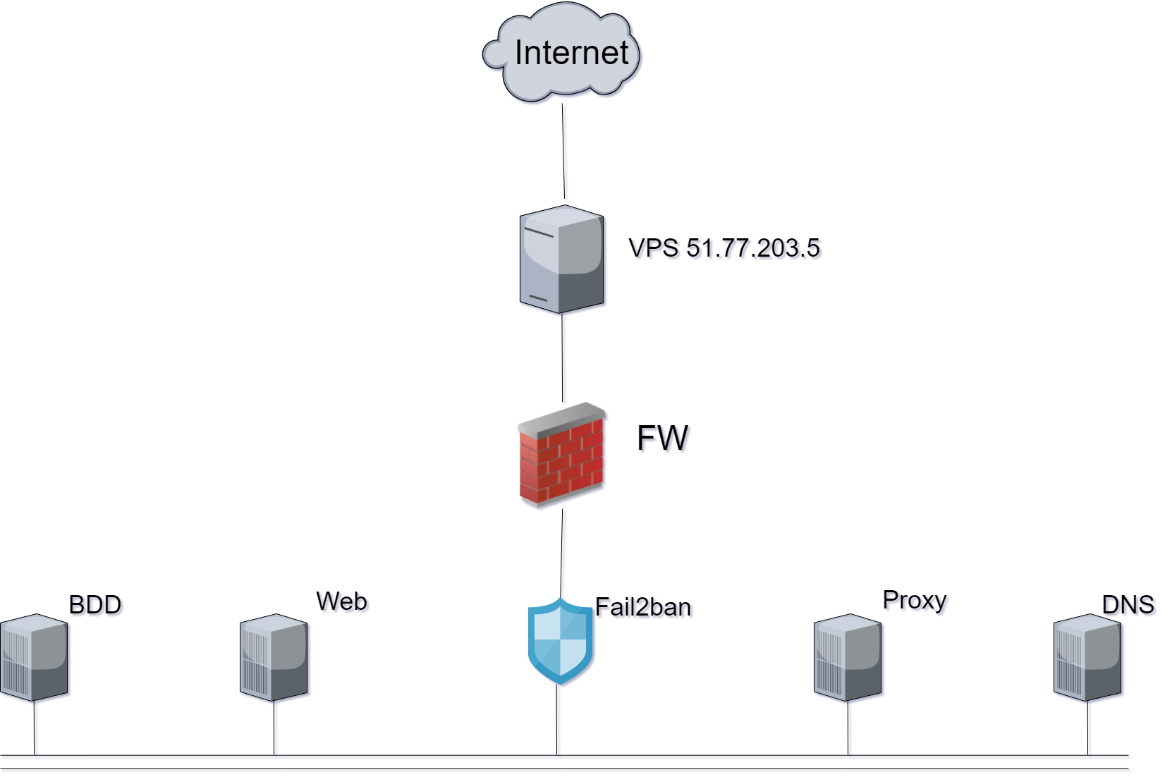


Figure - Schéma physique

**Difficultés rencontrées**

* **Web** :
  + L’intranet n’est pas inaccessible depuis l’extérieur. Ceci est peut-être du à une mauvaise gestion des IPTables ou du fichier intranet.wt7.ephec-ti.be de nginx. Malgré des tentatives diverses de configurations différentes, je ne suis pas parvenu à régler ce problème.
  + Une mauvaise configuration des certificats m’empêchait d’avoir accès aux sites en https. Ce problème a été résolu en recommençant à 0 et en suivant simplement le tutoriel disponible sur le site de Certbot (<https://certbot.eff.org/>).
  + Ayant des lacunes en PHP, il m’était très difficile de configurer la page b2b. Celle-ci ne répondait pas à mes demandes. J’ai pu bénéficier de l’aide d’un collègue de classe qui a pu m’expliquer comment lui s’y était prit pour y arriver. Le b2b est donc maintenant fonctionnel.
  + Lors de la copie via le Dockerfile des dossiers contenant le site web, les sous-dossiers n’étaient pas copiés et tous les fichiers qu’ils contenaient étaient copier dans le fichier source. Cela avait pour conséquence de désactivé le CSS, JS, … qui n’étaient donc plus renseignés correctement. Je n’ai pas trouvé de solutions à ce problème et ai donc opté pour une résolution manuelle du soucis.
* **DNS** :
  + Lors d’un restart du container bind depuis le VPS, si il y avait une erreur de configuration, le container ne redémarrait pas et affichait comme statut « Exited(1) ». Il n’était dés lors plus possible d’y accéder et il fallait donc le supprimer et le relancer à chaque fois que cela se produisait. La solution a été de reload le container depuis celui-ci grâce à la commande « service bind -s reload ». Je n’ai plus eu de soucis depuis car celui-ci ne voulait pas redémarrer si il trouvait une erreur et où elle se situait. Mais étant déjà dans le container, il me suffisait d’aller corriger cette erreur et de recommencer l’opération jusqu’au succès du reload.
* **BDD** :
  + Je n’avais dans un premier temps pas accès à la BDD que j’avais créée. Il se trouve que je n’avais simplement pas les droits requis pour les requêtes que je lui soumettais. La solution a été de créer un utilisateur et de lui attribuer tous les droits.
  + La première base de donnée que j’avais réalisée grâce à mySql workbench ne cessait de me renvoyer des erreurs une fois importée sur le container mySql. J’ai donc décidé de la recommencer à la main, directement dans le container.
* **Proxy** :
  + Un proxy a été mis en place via Squid. Celui-ci ne semble par contre pas empêcher les requêtes interne vers des sites tels que facebook.com ou instagram.com. Le problème peut venir du fait que elinks (le programme installé pour la navigation internet depuis le VPS) est installé à même le VPS et non dans un container simulant un employé de WoddyToys. Le soucis n’est toujours pas résolu.

**Procédure de validation**

* **Web** :
  + La validation du web, et donc de nginx était assez simple. La commande « nginx -t » permet de vérifier automatiquement si la configuration est correcte ou non. Cette commande peut même être directement écrite dans le Dockerfile pour que la vérification se fasse lors de la création du container.
* **DNS** :
  + La validation du DNS s’est faite essentiellement via les commandes nmap (ports) et dig (noms de domaine). Cette dernière m’a été particulièrement utile pour lorsque je ne parvenais pas à accéder aux sites web. Je pouvais consulter directement depuis le vps si la résolution se faisait. Un contrôle sur un moteur de recherche (et via elinks) était également fait régulièrement. L’accès via l’ip(v4 et v6) a également été faire sur un moteur de recherche standard (en faisant [2001:41d0:404:200::12e6] pour la seconde).
* **MySql** :
  + Pour vérifier si la BDD fonctionnait bien, rendez-vous-sur b2b.wt7.ephec-ti.be et rentrer un nom et des coordonnées dans le formulaire. Une fois envoyées, ces informations se retrouvent bien dans le tableau récapitulatif situé juste en dessous.
* **Proxy** :
  + Squid permet notamment d’empêcher l’accès à un site web quelconque. La tentative de validation se fait pour le moment grâce à elinks installé sur le VPS.  
    (Essais non concluants pour le moment)

**Monitoring des services déployés**

Dans un premier temps, il est possible de consulter les logs généraux ainsi que ceux de fail2ban pour avoir un résumé du comportement de notre configuration. Ensuite, des outils de monitoring de DNS sont disponibles en ligne tels que  « <https://mxtoolbox.com/DNSLookup.aspx> » ou « <https://toolbox.googleapps.com/apps/dig/> ». Mes recherches m’ont permises de découvrir qu’il existait quantité de différents services à installer sur le vps mais je n’ai ici pas eu le temps de les utiliser.