HAUTE ÉCOLE EPHEC

2017 - 2018

Administration Système
WoodyToys - Rapport Technique

Groupe 1 :
Simon Fauconnier
Steve Henriquet
Adrien Nini Pereira

Référent :
V. Van den Schrieck

11 juin 2018

Table des matières

1 Schéma **1**
- 1.1 Schéma Physique . . . 1
- 1.2 Schéma Logique . . . 2
- 1.3 Infrastructure . . . 3
  - 1.3.1 Base de donnés . . . 3
  - 1.3.2 Mail . . . 3
  - 1.3.3 VoIP . . . 3

2 Difficulté **4**
- 2.1 Difficultés rencontrées . . . 4
  - 2.1.1 Serveur Web . . . 4
  - 2.1.2 Serveur base de données . . . 4
  - 2.1.3 Serveur DNS . . . 4
  - 2.1.4 Service Mail . . . 5
  - 2.1.5 VoIP . . . 5

3 Sécurité **6**
- 3.1 Sécurités mises en place sur les VPS . . . 6
- 3.2 Techniques de sécurisation . . . 6
  - 3.2.1 Serveur Web . . . 6
  - 3.2.2 Serveur Base de Donnée . . . 6
  - 3.2.3 Serveur DNS . . . 6
  - 3.2.4 Service Mail . . . 6
  - 3.2.5 VoIP . . . 7
- 3.3 Procédure de validation du déploiement de la solution . . . 7
- 3.4 Monitoring . . . 7

1. Schéma

1.1 Schéma Physique

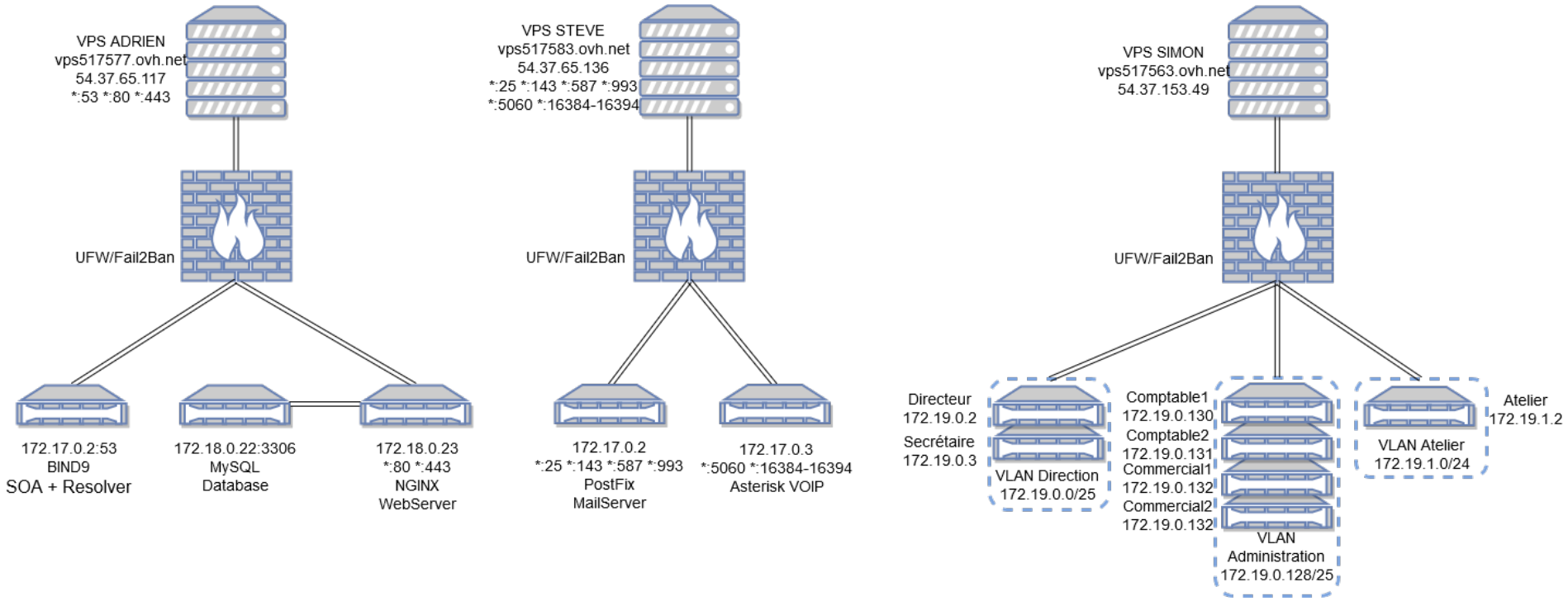

FIGURE 1.1 – Schéma physique

Chaque VPS représente un pôle d'activité différent. Le vps517577 est utilisé pour la partie web et DNS, il contient donc le résolveur DNS ainsi que le SOA, le serveur Nginx avec les trois sites (intranet, b2b et vitrine) et la base de données contenant le catalogue, les commandes, des informations clients et le répertoire employé. Le vps517583 est utilisé pour tout ce qui a attrait à la communication et comprend par conséquent le mail et la VOIP. Le vps517563, quant à lui, a été utilisé pour effectuer des tests dans un premier temps et maintenant, il simule des employés de la société.

1.2 Schéma Logique

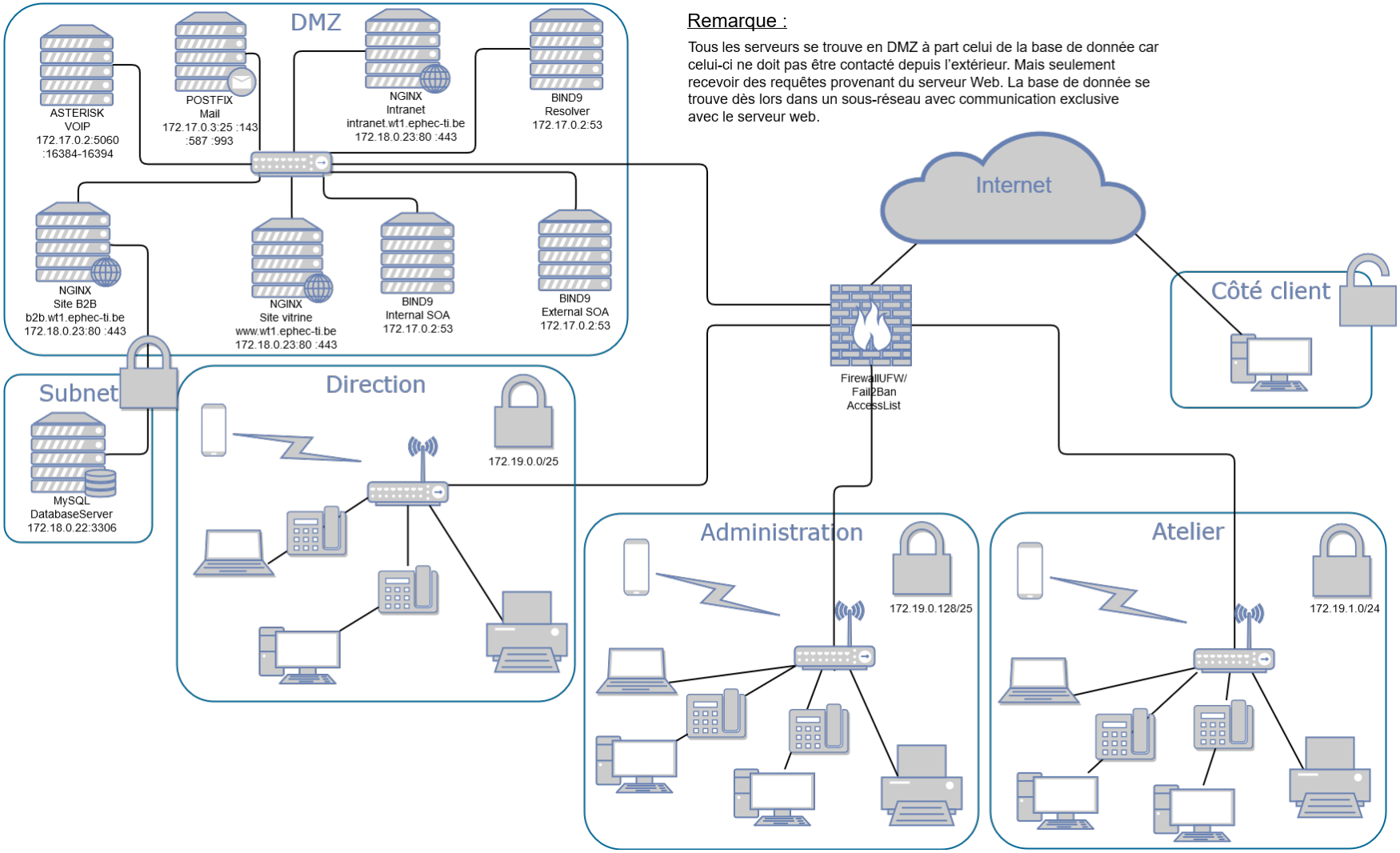

FIGURE 1.2 – Schéma Logique

1.3 Infrastructure

1.3.1 Base de donnés

La base de données se trouve sur un sous-réseau exclusif où communique avec le serveur web pour prévenir contre des attaques directes sur la DB depuis l'extérieur.

1.3.2 Mail

Pour le moment, nous avons configuré un seul serveur mail. Cela nous permet de déployer facilement et rapidement le service mail. Dans l'idéal nous devrions mettre en place un deuxième serveur relai smtp et de l'ajouter au fichier de zone du DNS avec un poids plus élevé. Cela nous permettrait d'assurer la continuité de fonctionnement du service mail en cas de problèmes avec le premier relai smtp.

1.3.3 VoIP

Nous avons mis en place un simple serveur VoIP avec Asterisk sans avoir ajouté de plugins ou autres systèmes supplémentaires. Dans le cadre des besoins de notre client, ce que nous avons implémenté est largement suffisant. Mais nous pouvons imaginer mettre en place un système plus sophistiqué qui permettra la gestion plus simple et plus poussée du service.

2. Difficulté

2.1 Difficultés rencontrées

2.1.1 Serveur Web

— Nous avons d'abord testé toutes les configurations dans un container Docker et, lorsqu'elles étaient correctes, nous nous sommes attelés à la création du Dockerfile.

— Nginx est assez facile à mettre en place, les difficultés ont été rencontrée lors de l'écriture du Dockerfile. Il fallait en effet apprendre la syntaxe de ce dernier.

— L'installation de PHP sur le serveur n'a pas posé de réels problèmes, hormis l'implémentation dans le Dockerfile.

— La mise en place du protocole https a été plus compliqué à mettre en place que prévu. Nous avons heureusement trouvé un site *https ://www.sslforfree.com/* qui permet d'obtenir rapidement les certificats nécessaires. Les sites wt1.ephec-ti.be et b2b.wt1.ephec-ti.be utilisent le protocole contrairement à l'intranet. En effet, l'intranet étant inaccessible aux personnes extérieures à l'entreprise, le site délivrant les certificats ne peut pas le vérifier. Une solution serait de désactiver le blocage de l'intranet le temps de demander le certificat.

2.1.2 Serveur base de données

— L'installation de mysql ne nous a posé aucun souci lors de l'installation dans un container ubuntu. Néanmoins, pour l'installer via un Dockerfile, il a fallu introduire un mot de passe en utilisant une redirection de l'output en input avec l'utilisation de trois chevrons "<".
L'inconvénient est que les containers Ubuntu utilisent par défaut le dash et non le bash, il a donc fallu utiliser RUN ["/bin/bash", "-c", "Les commandes contenant les chevrons"] pour passer ce problème.

— Grâce à un tutoriel assez fourni, la base de données créée via le Dockerfile inclut déjà l'intégralité des tables et des informations pour les tests.

— Un problème restant est la difficulté de mettre en place une sécurité au niveau des mots de passe car la compréhension des variables d'environnement est requise et le temps imparti ne nous a pas permis de creuser aussi loin. Dès lors, les mots de passe apparaissent en clair dans les fichiers de configuration.

— Lors de la mission 1, on n'arrivait pas à se connecter à la base de données.
Il fallait effectivement ajouter des utilisateurs avec certains droits et surtout les ajouter dans la base de données. Ce faisant, on crée de nouvelles failles de sécurité car les logins d'accès se trouvent dans un .sql en libre sur Github.
Cependant, la seule manière d'y accéder reste de se trouver à l'intérieur du subnet où ne se trouvent actuellement que la base de données et le serveur web.

— Le deuxième problème était la lenteur de réponse de la page web faisant appel à la base de données. Il a été corrigée grâce à deux commandes ajoutées au fichier de config mysqld.cnf.
En effet, *skip-name-resolve* désactive la recherche DNS ce qui permet l'accélération de la connection car cette dernière ne travaille qu'avec l'IP et *skip-host-cache* permet de remettre à zéro la cache du serveur de base de données.

2.1.3 Serveur DNS

— Partir de zéro pour la création du Dockerfile pour bind9 fût un challenge. Étant donné que la configuration de DNS était quelque chose de nouveau pour nous, il nous a fallu un temps d'adaptation pour transcrire nos connaissances théoriques à la pratique.

— On a commencé par configurer un serveur DNS directement depuis l'intérieur d'un container docker. Ensuite, lorsque nous avons compris comment se passait la configuration, nous nous sommes lancés dans la création d'un Dockerfile personnalisé.

— A un point donné, nous avions un SOA qui fonctionnait correctement mais nous n'arrivions pas à faire fonctionner le résolveur. Après quelques recherches sur la question, le fait d'inverser la view internal et la view external dans le fichier named.conf.local a résolu notre problème.

2.1.4 Service Mail

— Le seul réel problème rencontré fût avec les boîtes mail de Google.
En effet, Google est extrêmement strict en matière de spam, et malgré notre score au test du mail (voir Fig. 2.1), nous arrivons toujours dans les spams des boîtes Google.
Cela est principalement dû à quelques configurations DNS (conseillées par Google) que nous n'avons pas implémenté (ex : deux serveurs SOA ns1 et ns2).
Une autre hypothèse est que Google attend que notre serveur mail soit effectif depuis minimum 3 mois pour nous considérer comme non-spam.

FIGURE 2.1 – Le service mail est opérationnel

2.1.5 VoIP

— Après l'étape de recherche et de compréhension des fichiers de configuration, qui était l'étape la plus longue, l'étape de la configuration fût simple et rapide grâce à toute la documentation et aux ressources mises à notre disposition.

— Au début, lorsque l'on appelait le numéro de la compta, un premier numéro sonnait, ensuite le suivant si le premier n'avait pas répondu. Maintenant, tous les numéros sonnent en même temps.

3. Sécurité

3.1 Sécurités mises en place sur les VPS

Le tableau (Fig 3.1) présente les différentes mesures de sécurité installées sur les VPS

Sécurités	Raisons
Mises à jour effectuées régulièrement	Permet de profiter des corrections des failles de sécurité
Création d'un utilisateur avec des droits restreints	Permet de ne pas être directement connecté avec l'utilisateur Root lors de la connexion au VPS
Désactivation de l'accès au VPS en ssh pour l'utilisateur Root	Permet de limiter l'accès au compte root
Désactivation des connexions par mots de passe	Connexion uniquement possible en ssh avec clé privée
Installation de Fail to Ban	Permet de se prémunir contre les tentatives d'intrusion répétées

FIGURE 3.1 – Tableau récapitulatif des sécurités mises en place sur les VPS

3.2 Techniques de sécurisation

Techniques de sécurisation utilisées pour garantir l'intégrité, la confidentialité et la disponibilité des services implémentés :

3.2.1 Serveur Web

Utilisation du protocole HTTPS sur le site principal et le b2b. Cela permet à l'internaute de vérifier l'identité du site web grâce à un certificat unique. Ce protocole chiffre également les données. Une redirection par défaut de l'HTTP vers l'HTTPS a été mise en place (cette redirection nécessite l'ouverture du port 80).

3.2.2 Serveur Base de Donnée

Utilisation de variables d'environnements afin que le mot de passe ne se trouve pas en clair dans le DockerFile et dans les fichiers de configuration de la base de données.
Création d'utilisateur pour que l'administration ne s'effectue pas en root, ce qui permet de se prémunir contre d'éventuelle perte d'informations dues à de mauvaises manipulations.

3.2.3 Serveur DNS

Nous avons activé dnssec-validation qui permet de signer les paquets afin d'éviter l'usurpation d'identité.

3.2.4 Service Mail

- Actuellement il n'y a pas de réelle sécurité sur le mail. Pour le moment tout ce qui transite n'est pas crypté. Une solution est de changer de protocole utilisé pour passer sur une version sécurisée en TLS ou SSL.
- Au niveaux des mots de passe des utilisateurs mail, ils sont cryptés dans un fichier sur le serveur.
- Il n'y a pour le moment pas de système redondant pour garantir la disponibilité du service. Pour le réaliser, il nous suffit de créer un serveur mail identique sur un autre serveur et de gérer le poids de cette serveur au niveau du DNS.

3.2.5 VoIP

— Il n'y a pas de sécurité à proprement dit pour le moment, mais c'est une piste d'amélioration à explorer.

3.3 Procédure de validation du déploiement de la solution

Nous nous sommes réparti les 3 services (Web, BDD, DNS) entre nous. Nous avons tous testé en local l'installation et la configuration de nos services respectifs. Nous avons ensuite écrit des Dockerfiles permettant de déployer rapidement et facilement nos différents services.

Ci-dessous, notre check-list de validation :
— Le container reste actif, tourne en mode démon ?
— Les services sont-ils accessibles ?
— Les services sont-ils pleinement opérationnels ?

3.4 Monitoring

Possibilité d'installer Webmin qui permettra l'administration et le monitoring du DNS ainsi que d'autres services.

Le site mxtoolbox.com propose un service de monitoring mail qui permet de vérifier l'état de son service mail ainsi que des records DNS lié au mail. Mais il existe aussi d'autres solutions comme MailMonitor et bien d'autres.

Une interface web telle que Digium permettrait de monitorer le service VoIP mais l'idéal serait de réunir le monitoring de tous les services dans une seule application/interface web.

L'outil Uptime robot permet de vérifier rapidement que tous les services sont opérationnels (*Voir Fig. 3.2*).

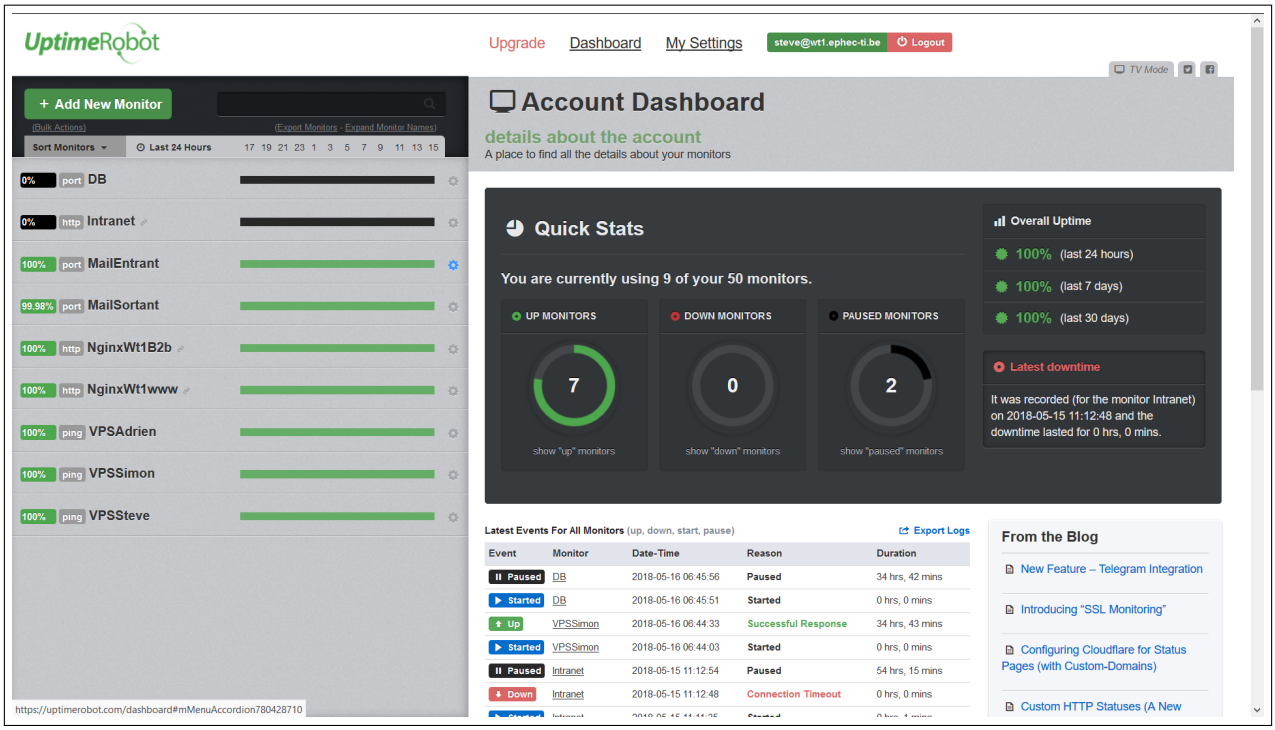

FIGURE 3.2 – Uptime robot

3. Bibliographie

[1] Sécurisation, Mitchell Anicas, *https ://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-16-04.*

[2] Sécurisation, A2 Hosting, *https ://www.a2hosting.com/kb/getting-started-guide/accessing-your-account/disabling-ssh-logins- for-root.*

[3] Docker Hub, Frédéric Aoustin, *http ://hrb85-1-88-121-176-85.fbx.proxad.net/blog/20161126115209/*

[4] Dockerfile, WebSetNet, *https ://websetnet.com/fr/create-docker-images-dockerfile//*

[5] Dockerfile NGINX, Justin Ellingwood, *https ://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-in-ubuntu-16-04/*

[6] Dockerfile NGINX, Justin Ellingwood, *https ://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-16-04/*

[7] PHP-fpm, tranquilhosting, *https ://www.tqhosting.com/kb/464/How-to-install-PHP-70-PHP-FPM-on-Ubuntu-1604-LTS-Xenial-Xerus.html/*

[8] PHP-fpm, stackoverflow, *https ://stackoverflow.com/questions/39391522/how-to-start-php7-0-fpm-in-dockerfile/*

[9] Bind9, Justin Ellingwood, *https ://www.digitalocean.com/community/tutorials/how-to-configure-bind-as-a-caching-or-forwarding-dns-server-on-ubuntu-14-04/*

[10] Bind9, zytrax, *http ://www.zytrax.com/books/dns/ch7/acl.html/*

[11] Bind9, StackExchange, *https ://serverfault.com/questions/637668/bind-failing-to-resolve-with-warning-recursion-requested-but-not-available/*

[12] Bind9, resystit, *https ://hub.docker.com/r/resystit/bind9//*

[13] Bind9, Mitchell Anicas, *https ://www.digitalocean.com/community/tutorials/how-to-configure-bind-as-a-private-network-dns-server-on-ubuntu-14-04/*

[14] Bind9, fprochazka, *https ://gist.github.com/fprochazka/1853976/*

[15] Bind9, gweatherby, *https ://help.ubuntu.com/community/BIND9ServerHowto/*

[16] mySQL, Stackoverflow, *https ://stackoverflow.com/questions/29145370/how-can-i-initialize-a-mysql-database-with-schema-in-a-docker-container/*

[17] mySQL, Stackoverflow, *https ://stackoverflow.com/questions/7739645/install-mysql-on-ubuntu-without-a-password-prompt/*

[18] mySQL, mySQL Oracle, *https ://dev.mysql.com/doc/refman/5.7/en/host-cache.html/*

[19] mySQL, mySQL Oracle, *https ://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html/*

[20] mail, tomav, *https ://github.com/tomav/docker-mailserver/*

[21] mail, Thomas VIAL, *https ://hub.docker.com/r/tvial/docker-mailserver/*

[22] mail, Grafikart.fr, *https ://www.youtube.com/watch ?v_ sCqM8MivCw&t2̄55s/*

[23] mail, Julien Castiaux (Inspiration des réalisations des groupes des années précédentes), *https ://github.com/Julien00859/Projet_ Admin_ Sys/wiki/Postfix-&-Dovecot/*

[24] HTTPS, Let's Encrypt, *https ://www.sslforfree.com*

[25] HTTPS, Cecile Muller, *https ://gist.github.com/cecilemuller/a26737699a7e70a7093d4dc115915de8*

[26] VoIP, Valentin Weber, *https ://www.networklab.fr/configuration-basique-dasterisk/*

[27] VoIP, titi les bons tutos, *https ://www.youtube.com/watch ?v=lk5bnoy7YcE*

[28] Monitoring, UpTime Robot, *https ://uptimerobot.com/*