Projet Administration Réseau : Web et DNS

Fantuzzi Sébastien
Verschraegen Gauthier
Sanglier Joachim

2TL1
Groupe 9

Fantuzzi Sébastien
Verschraegen Gauthier
Sanglier Joachim

Rapport client (Web, DNS et Mail)

Cahier des charges :

L'entreprise a à disposition une connexion à Internet et un réseau WiFi, ainsi qu'une infrastructure téléphonique IP. Elle souhaite que nous mettions de nouveaux services à disposition de l'usine, ainsi que d'assurer la maintenance et le remplacement des serveurs pré-existants.
WoodyToys contient deux sites web accessibles par leurs clients ainsi que par leurs fournisseurs, en plus d'un outil ERP Web uniquement accessible sur leur réseau interne.
Chaque employé se voit mettre à disposition une boîte mail accessible n'importe ou. Certaines addresses plus globales sont également à définir.
Le réseau téléphonique doit rendre l'entreprise accessible via Internet depuis l'extérieur, ainsi que d'assurer la communication des différents départements selon les règles suivantes ;

- *Ouvriers : Un poste pour pour joindre les autres départements, et joignable par tout le monde*
- *Secrétaire : Un poste pouvant joindre et joignable par n'importe qui*
- *Comptables : Un seul numéro permettant de joindre plusieurs postes ainsi que d'un numéro par bureau pour joindre celui-ci uniquement. Ils peuvent joindre n'importe qui sauf le directeur*
- *Commerciaux : Joignable par n'importe qui et pouvant joindre tout le monde sauf le directeur*
- *Direction : Pouvant joindre tout le monde. Joignable directement uniquement via la secrétaire.*

Enfin, une boîte vocale doit être à disposition de chaque poste

Traduction des besoins :

Nous devons mettre en place différent serveurs. Concernant les services web, il faut deux serveurs distincts pour traiter d'une part les pages accessibles par l'extérieur, et d'autre part l'outil ERP seulement accessible par le réseau interne. il faudra également un service de consultation et de transfert de mail. Le service VoIP est également à configurer pour chaque poste, ainsi qu'un serveur dédié au service.
Afin de joindre tous les différents services, il faudra mettre en place un serveur de nom interne et externe ainsi qu'un résolveur DNS pour permettre de joindre le serveur de nom interne.
Un pare-feu devra être ajouté afin de gérer les accès aux différents services ainsi que d'assurer la sécurité de ceux-ci et des utilisateurs.

Propositions de solutions techniques :

Les différents serveurs pouvant être utilisé par l'extérieur seront dans un sous-réseau à part. L'autre partie du réseau englobera donc tout le réseau interne accessible uniquement par les employés de l'usine. Nous installerons les différents services via un outil de conteneurisation.
Les services extérieur comprennent le serveur web contenant les 2 sites accessible au public et aux revendeurs, le serveur mail de transfert ainsi que le serveur de nom permettant de joindre les deux services précités. Une base de données sera également présente dans ce sous-réseau.

En interne, nous retrouverons le serveur web interne permettant de joindre l'outil ERP, le serveur de consultation de mail ainsi que le serveur de noms permettant de joindre ces services.

Liste des services utilisés :

- Service de *conteneur* : Docker
- Service Web : Apache
- Service DNS : Bind9
- *Transfert de mail : Dovecot*
- *Consultation de mail : Postfix*
- *Base de données : MySQL*

Choix et justification de la solution :

Une telle structure du réseau permet de n'exposer à l'extérieur (et donc à toute potentielle attaque) que les services nécessaires aux personnes extérieures. Nous limiterons les attaques vers ces services plus vulnérables via le pare-feu.

En ce qui concerne Apache, nous l'avons choisi plutôt que Nginx par exemple car *il reste leader dans son domaine en plus d'être libre d'accès. Ses fonctionnalités sont très connues et appréciés, choses pour lesquelles nous avons choisi ce service.*
Docker a été choisi parce qu'il est également très répandu, pour sa fonctionnalité de lien avec GitHub et qu'il est libre d'accès contrairement à Azure par exemple.
Bind9 n'ayant pas de réel concurrent, il paraît évident de le choisir lui.
Dovecot, Postfix et MySQL ont été choisi ensemble car ils forment un trio très répandu, pouvant directement fonctionner l'un avec l'autre et proposant les meilleurs services d'authentifications, de sécurité et de rapidité tout en étant libre d'accès

Etat des lieux

Le serveur web est préparé, il ne manque plus que de le gérer via les serveurs de noms. Cela ne devrait prendre que quelques heures pour le configurer parfaitement.
La résolution de noms est fonctionnelle. A partir de là, les différents serveurs de nom devraient facilement être prêt en 5 heures tout au plus.
La configuration du mail n'étant qu'à peine entamée, celle-ci nécessitera surement encore une bonne dizaine d'heure avant d'être opérationnelle.

Ajustements et maintenance

Il est possible pour le serveur web de rajouter d'autres pages, et donc de rendre ,via les serveurs de nom,s plus de sites disponibles en ligne. Ces services ne nécessitent pas de réel besoin de maintenance.

Fantuzzi Sébastien
Verschraegen Gauthier
Sanglier Joachim

Schémas réseaux

Schéma WoodyToys :

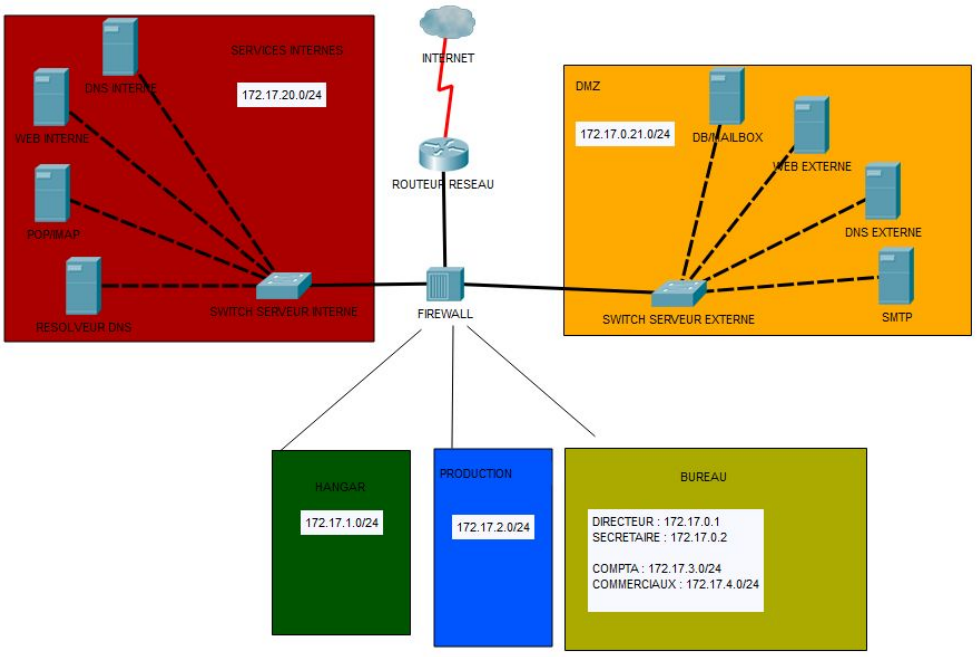
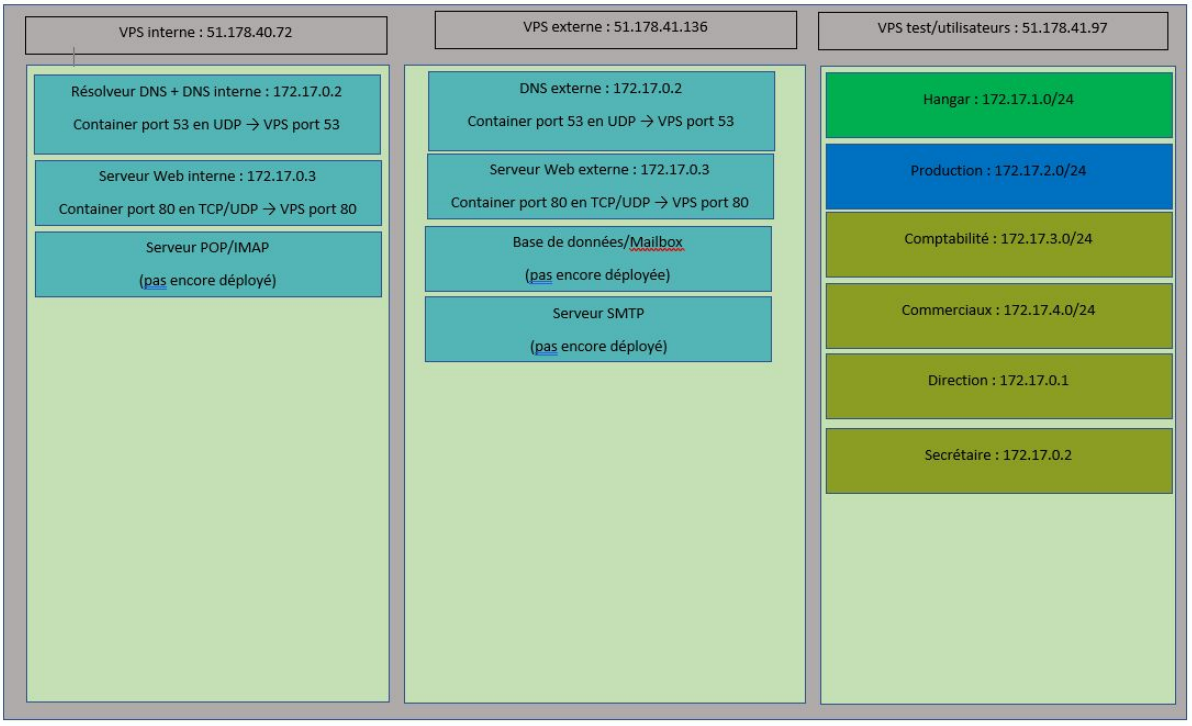


Schéma prototype :



Fantuzzi Sébastien
Verschraegen Gauthier
Sanglier Joachim

Rapport technique

Présentation du bilan du projet :

Le projet continue d'avancer, avec le résolveur DNs venant s'ajouter aux services fonctionnels. Cerner toute la méthodologie Docker explique entre autre le temps que nous prenons pour ajouter les différents services. Quelques problèmes de connexion aux VPS ainsi que des conflits de port ont également retardé notre progression.

Etat d'avancement :

Notre serveur web est toujours fonctionnel bien que sa configuration ait changé. Le résolveur DNS est configuré, et donc le container Bind9 est prêt. A partir de là, la configuration des serveurs DNS interne et externes devraient aisément suivre. Le peaufinement de nos services déjà existants ainsi que le mail sont nos prochains gros objectifs.

Commentaire explicatif des schémas réseaux :

Schéma WoodyToys :

Nous avons répartis le réseau en 3 parties : le réseau externe (ou DMZ), le réseau interne et le réseau utilisateur. Le réseau externe contient toutes les machines ayant à subir des requêtes provenant de l'extérieur. Nous y avons donc placé le serveur DNS externe ainsi que *le serveur de transfert de mail SMTP, le serveur web externe et la base de données.*
Vient ensuite le réseau interne, qui contient donc tous les serveurs n'ayant pas à subir des requêtes venant d'internet. Il est constitué du serveur Web interne,du résolveur DNS, *du serveur DNS interne et du serveur de consultation POP/IMAP.*
Enfin, le réseau utilisateur représente l'ensemble des périphériques utilisés pour garantir une connectivité vers internet et les différents serveurs mis à disposition.

Schéma prototype :

Afin de reproduire le mieux possible les circonstances de l'entreprise, nous avons décidé de reproduire le schéma WoodyToys au travers de nos différents VPS, où une zone est simulée dans un VPS dédié. *Toutefois, nous avons regroupé le serveur de nom interne et le résolveur en un seul conteneur, étant donné que ceux-ci ont tout les deux à utiliser le port 53 et qu'ils ne sont accessibles que depuis le réseau interne, ils avaient suffisamment de point communs pour les grouper en un seul container et ainsi éviter de devoir les répartir sur plusieurs VPS.*

Difficultés rencontrées :

Nous avons rencontré bon nombre de difficultés lors de l'installations des serveurs web et D?S. La plupart de ces difficultés sont survenues pendant l'installation de Bind9. En effet, il nous a fallu de longs moments de réflexions pour démarrer l'image docker. Ces erreurs sont survenues lors de l'exécution de l'image docker, plus particulièrement celle de Bind, la commande étant longue et complexe. *Les soucis de connexion au VPS ainsi que de ports ont également rendu notre tâche plus ardue.*

Méthodologie

N'ayant découvert que tardivement la réelle utilité du lien entre GitHub et DockerHub, nous n'avons dans un premier temps pas usé de cette fonctionnalité et sommes simplement passé par des docker push et pull afin de publier et récupérer nos images "pré-construites" sur nos machines personnelles. Une fois amenée sur nos VPS, nous avons alors docker run nos différents images dans différents conteneurs.
Le serveur Apache a été validé comme fonctionnel une fois qu'une commande curl sur lui-même rendait bien la page pré-conçue d'Apache, tandis que la résolution DNS fut validée lorsqu'une commande dig via le conteneur en question affichait des résultats cohérents au site passé en paramètre.
Il est possible de surveiller le fonctionnement des deux services nommés ci-dessus via les logs disponibles pour chacun de ces services.

Analyse de la sécurité

La sécurité est un point important d'un réseau tel que celui demandé dans ce projet. En effet, il est nécessaire de mettre en place une zone démilitarisée (DMZ) accessible via un firewall. Cette zone nous permettra d'y placer nos serveurs public et ainsi éviter toute entrée intrusive dans notre réseau privé.
Les VPS ont été sécurisé via une authentification exclusivement par clé, dont nous sommes les seuls propriétaires. De plus, l'authentification par l'utilisateur "root" a été désactivée. Pour l'instant, nos services sont tous déployé sur un seul VPS, mais ceux-ci sont tous des services internes et sont donc fidèles à notre schéma.

Ajustement par rapport au feedback

Après la correction croisée, nous nous sommes rendus compte que nous avions été peu attentif aux consignes concernant les points abordés dans le rapport. De plus, le temps nous a permis de mieux cerner les différents services et donc des différentes erreurs de compréhension que nous avions commises. Il y a eu donc énormément d'ajouts, de suppression et de remaniement du contenu du rapport. Cette correction fut donc très bénéfique pour nous.