Projet Admin

Rapports

Rapport client

Cahier des charges

L'entreprise souhaite remplacer ses serveurs vieillissants, et fait appel à nous pour la phase de conception et de validation d'une nouvelle infrastructure d'hébergement des services informatiques. En clair, l'entreprise a besoin de :

- Deux sites internet.
- Un site intranet.
- Les terminaux des employés doivent pouvoir aller sur internet.
- Chaque employé doit pouvoir être joignable par mail à l'intérieur et à l'extérieur de l'entreprise.
- Chaque employé doit pouvoir être contacté par téléphone grâce à la téléphonie IP.

Traduction des besoins

En clair, il faut que nous mettions en place ceci :

- 3 serveurs web (2 externes et 1 interne)
- Une base de données
- 3 serveurs dns (2 SOA et 1 résolveur)
- Un serveur mail
- Un serveur de téléphonie IP

Propositions

Pour répondre aux attentes du client, nous proposons de mettre en place ceci :

- 2 serveurs web externes apache : www.wt1-10.ephec-ti.be et b2b.wt1-10.ephec-ti.be
- Une base de données mysql.
- 1 serveur web interne apache : local.woodytoys.be (Ce serveur ne sera accessible que par les machines de l'entreprise)
- 1 Dns interne (SOA) qui regroupe ce qui se trouve dans la zone woodytoys.be (Ce serveur ne sera accessible que par les machines de l'entreprise)
- 1 Dns externe (SOA) qui regroupe ce qui se trouve dans la zone wt1-10.ephec-ti.be
- 1 Résolveur dns pour permettre aux machines de naviguer sur internet (Ce serveur ne sera accessible que par les machines de l'entreprise)
- Un serveur mail
- Un serveur voIp avec la répartition suivante

Département		
Direction	100 150	Directeur Secrétaire
Comptables	200 210 220 310 320	Service comptable Comptable 1 Comptable 2 Commercial 1 Commercial 2
Hangar	400	Ouvriers

Justification

En ce qui concerne les serveurs web, il en existe plusieurs : Apache, Nginx, IIS, LiteSpeed, etc... En ce qui nous concerne, nous avons choisi de travailler avec Apache pour les raisons suivantes :

- Le serveur web le plus utilisé
- Gratuit
- Fiable
- Facile à configurer
- Utilise peu de ressources et est donc rapide

En ce qui concerne les DNS, il existe plusieurs serveurs comme : PowerDNS, CoreDNS, Amazon Route, Bind9, etc... En ce qui nous concerne, nous avons choisi de travailler avec Bind9 pour les raisons suivantes :

- Le serveur dns le plus utilisé
- Open Source
- Stable
- Grosse communeauté

En ce qui concerne la base de données, il en existe plusieurs comme : mysql, oracle, postgresql, mongodb, etc… Pour la réalisation de notre DB, nous avons choisi mysql pour les raisons suivantes :

- Le plus populaire
- Il est multiplateforme
- Il est natif dans la majorité des Framework web
- Il offre des performances élevées en restitution (lecture)

Pour VoIP nous avons choisi Asterisk. Asterisk est un logiciel gratuit pour les ordinateurs de toutes sortes, qui offre les fonctionnalités d'un système téléphonique. Il prend en charge la téléphonie IP (VoIP) avec différents protocoles de réseau.

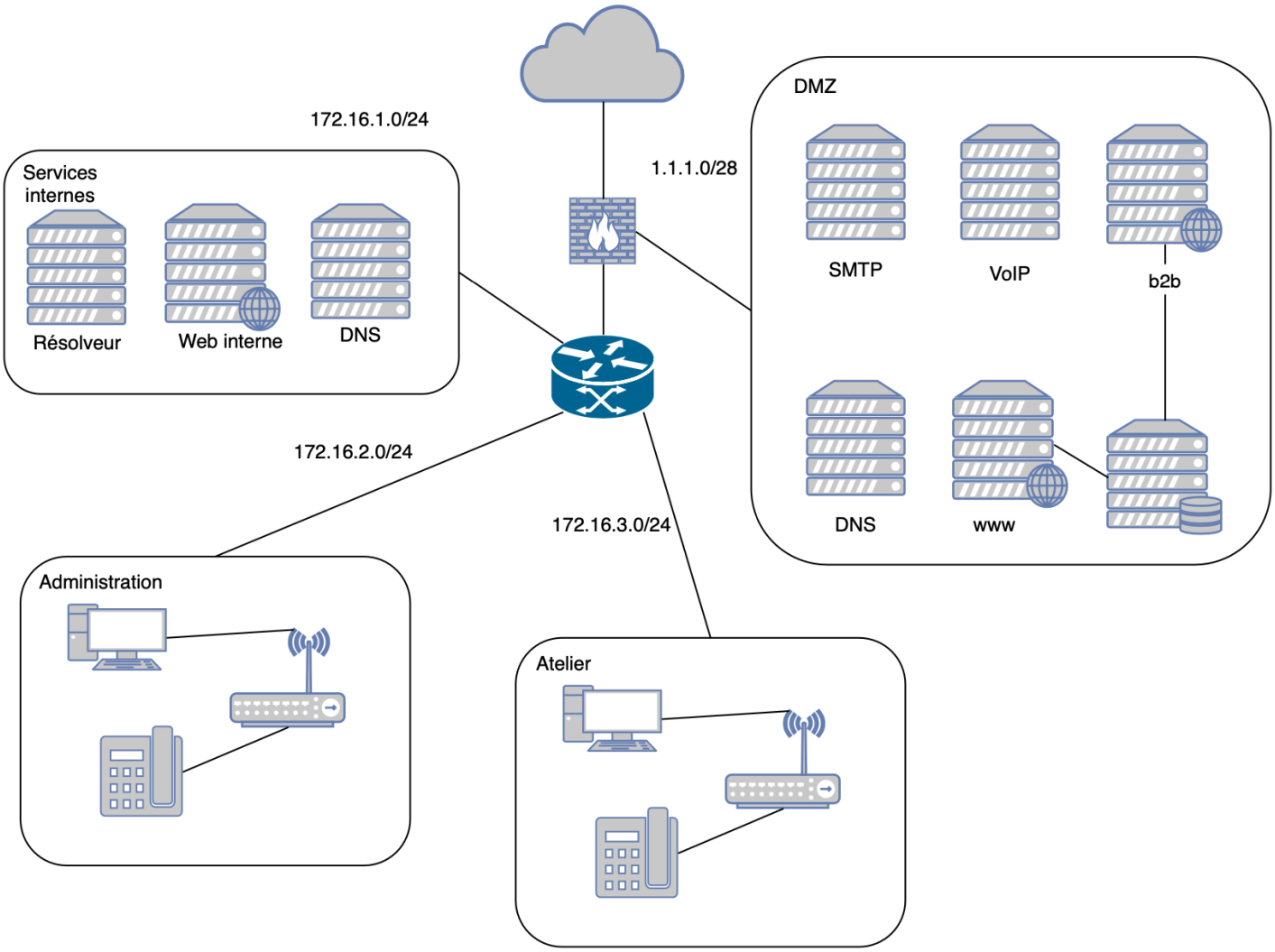
État

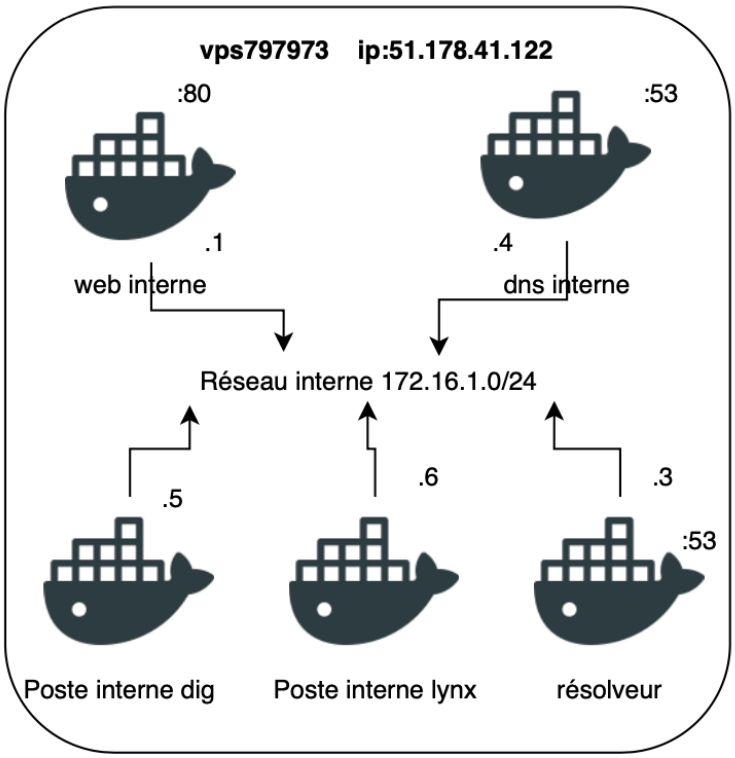
A ce stade, nous avons déployé les trois serveurs web, les trois serveurs dns ainsi que la base de données. Les serveurs mail et voIp sont en cours, ils devraient être déployé d'ici deux semaines.

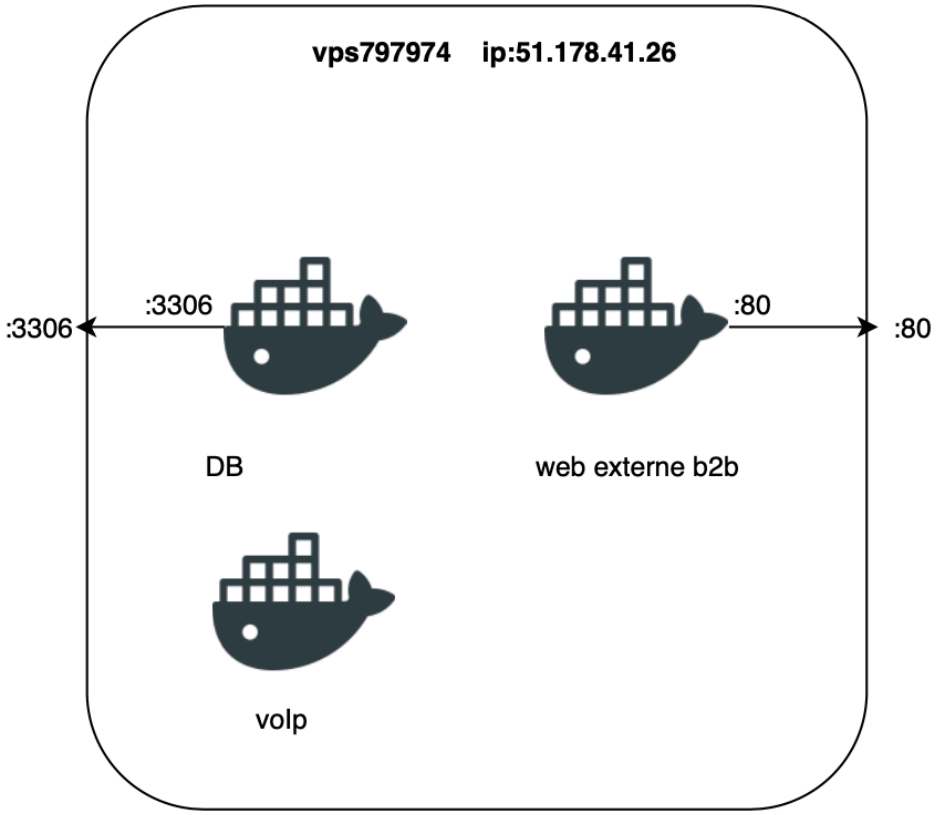
Besoins en maintenance

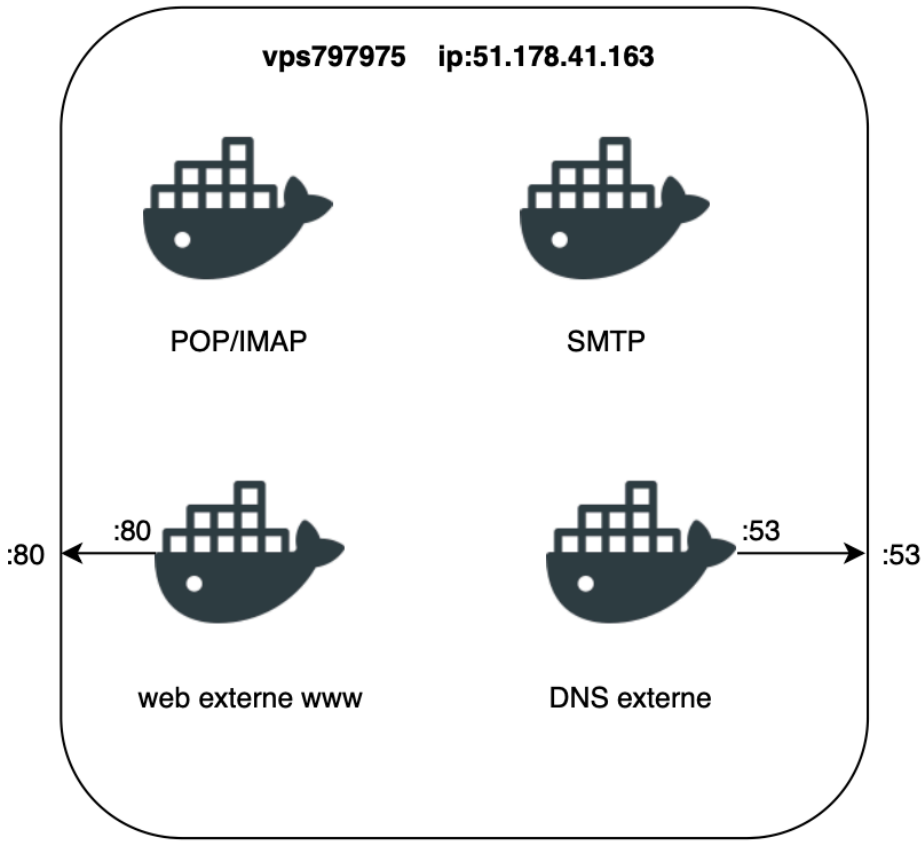
Notre groupe se tient à l'écoute si besoin de maintenance. En effet, il est réalisable pour nous de constater si un problème se passe, et si c'est le cas, nous le résoudrons en relançant le serveur si nécessaire.

Schémas


172.16.1.0/24
Services internes
Résolveur
Web interne
DNS
1.1.1.0/28
DMZ
SMTP
VoIP
b2b
DNS
www
172.16.2.0/24
172.16.3.0/24
Administration
Atelier


vps797973 ip:51.178.41.122
:80
:53
.1
.4
web interne
dns interne
Réseau interne 172.16.1.0/24
.5
.6
.3
:53
Poste interne dig
Poste interne lynx
résolveur


vps797974 ip:51.178.41.26
:3306
:3306
DB
:80
:80
web externe b2b
volp


vps797975 ip:51.178.41.163
POP/IMAP
SMTP
:80
:80
web externe www
:53
:53
DNS externe

Analyse de sécurité

Vps et docker

Tout d'abord, les risques encourus quand nous nous connectons par mot de passe à nos vps sont simples, si quelqu'un arrive à intercepter la communication, il pourra lire notre mot de passe et donc prendre possession de notre vps. De plus, il est important de ne pas mettre des containers docker accessibles depuis l'extérieur sur le même vps que des containeurs qui gèrent l'interne. En effet, les serveurs en liens avec l'extérieur sont plus susceptibles de se faire pirater et donc pourraient donner des informations internes à l'entreprise.

Solutions

Pour la connexion au vps nous allons nous connecter qu'une seule fois par mot de passe et ensuite utiliser la connexion par clé partagée. En effet, nous allons créer 4 utilisateurs sur chaque vps, et partager une clé pour chaque utilisateur. Grâce à cela, seuls les utilisateurs ayant la clé pourront se connecter au vps et le trafic sera chiffré.

Pour la répartition des containeurs, nous avons séparé les serveurs externes et les serveurs internes.

Risques

Web

- Dénis de service
- Lecture des informations qui transitent

Dns

- Dns poisoning
- Déni de service

Solutions

Web

En ce qui concerne le web externe, aucune mesure de sécurité n'a été prise. Pour ce qui est du web interne, seul les adresses IP internes à l'entreprise ont le droit d'y accéder. (commande Require)

Dns interne

Pour ce qui est du web externe, aucune mesure de sécurité n'a été prise. En ce qui concerne le dns interne et le résolveur nous n'avons autorisé que les IP interne à l'entreprise à faire des requêtes. Pour ce faire, nous avons modifié le fichier named.conf.options et ajouté allow-query{#ip} pour n'autoriser que les requêtes venant de ces adresses IP.

Rapport technique

Groupe

Groupe 2TL1-10. Membres : DELESTIENNE Damien, SERVAIS Léon

Damien : Pour cette partie, Damien a mis en place les 3 serveurs web, les 3 serveurs dns ainsi, les 2 postes client et la base de données.

Léon : S'occupe de la configuration de voIp

Justification schémas

Tout d'abord pour le schéma logique, nous avons préféré le séparer en trois parties :

Une partie DMZ qui reprend tous les serveurs qui seront joignables depuis l'extérieur. Cela comprend : les webs externes, la db, le dns externe, le mail, le voip. Nous avons préféré isoler les services joignables depuis l'extérieur pour éviter que quelqu'un puisse accéder à un service interne à l'entreprise.

Une partie service interne qui reprend tous les serveurs joignables par les machines internes à l'entreprise et donc : le web interne, le dns interne, le résolveur. En mettant les services internes entre eux, nous nous assurons qu'ils ne seront pas joignables de l'extérieur mais que par les machines internes.

Une partie terminaux, qui elle, comprend toutes les machines internes à l'entreprise.

Pour ce qui est du schéma physique, nous avons décidé de séparer les services de manière équivalente pour s'assurer une vitesse de travail correcte de la part de nos serveurs. Le seul choix logique a été de séparer les serveurs joignables de l'extérieur et les serveurs interne. Il y a donc deux VPS qui contiennent des serveurs de DMZ et un VPS qui contient toute la partie interne.

Adressage

Nous avons choisi de diviser l'entreprise et ses terminaux en quatre parties : les serveurs internes, l'atelier et l'administration, la dmz. Pour ces trois parties internes, nous leur avons attribués un sous réseaux d'adresses privées de classe b : 172.16.1.0/24 172.16.2.0/24 172.16.3.0/24 . En ce qui concerne la partie DMZ nous leur avons attribué le sous réseau 1.1.1.0/28 car il nous faut des IP publiques.

Méthodologie

Pour travailler sur ce projet, nous travaillons de la sorte :

Tout d'abord prise de connaissance de la tâche à réaliser, des adresses IP, des pages web, etc… Après cela, nous essayons de mettre en place notre serveur sur une vm ubuntu et de rendre ce serveur opérationnel et respectant les conditions. Une fois cette étape franchie, nous réalisons le docker file et construisons l'image docker qui va avec pour après faire tourner le container docker sur base de cette image. Après vérification du fonctionnement du container docker, nous envoyons les dossiers de config sur le vps, nous construisons l'image et ensuite on lance le container.

Validation

Pour valider nos containers il y a plusieurs étapes :

Tout d'abord la validation globale : nous lançons notre container docker et vérifions qu'il tourne bien, grâce à la commande docker ps

Validation du web : nous lançons notre container et ensuite nous essayons une recherche sur notre navigateur avec la bonne adresse IP et le bon port.

Validation web interne : Création d'un container utilisateur lynx que l'on ajoute dans le même sous réseau que le web interne.

Validation dns externe : Recherche dans notre navigateur sur le nom de domaine.

Validation dns internes : Création d'un container utilisateur qui contient dig, on l'ajoute dans le même sous réseau que nos dns et on effectue une requête dig.

Problèmes rencontrés

- Gérer l'accès au web interne (solution : commande require)
- Tester les serveurs dns (solution : requête dig)
- Tester les dns internes (solution : créer un réseau local, y ajouter un utilisateur avec une adresse ip interne, et effectuer une requête dig)
- La connexion avec Zoiper

Changements apportés

- Ajuster le schéma logique : rajouter des liens, rajouter un deuxième web externe, mettre la dmz en lien direct avec le firewall, rajouter l'adresse ip des serveurs internes.
- Ajuster le schéma physique : rajouter deux containers docker, un pour le deuxième web externe et un pour la db
- Ajuster l'orthographe et la ponctuation.
- Rajouter les parties concernant notre avancement

11.05.2020

- Reformulation du cahier des charges
- Reformulation de la traduction des besoins
- Reformulation des justifications + comparaisons
- Mise à jour de schéma et ajout des ports
- Ajout de quelques failles de sécurité
- Ajout de la validation
- Correction de l'orthographe