

ECOLE NATIONALE D'ECONOMIE APPLIQUÉE ET DE MANAGEMENT



Mise en place d'une solution complète d'administration de messagerie

Par :

Picasso T. I. HOUESSOUDOSSOU

Sous la supervision de :

M. Victor OYETOLA

Encadreur :

M. Bruno Bellarmin LAWSON

Année : 2019-2020

République du Bénin

Dédicaces

Je dédie le présent document :

- A ma mère, qui m'a toujours apporté son amour, son soutien inconditionnel, sa patience, sa générosité et pour tous les efforts consentis en ma faveur.
- A mon père, qui m'a donné le sens du travail.

Remerciements

Je tiens à remercier les bonnes volontés qui m'ont aidé dans la réalisation de ce travail, notamment :

- Madame Rosalie WOROU, Directrice de l'ENEAM ;
- Monsieur Théophile DAGBA, Directeur adjoint de l'ENEAM ;
- Le Directeur de JScom pour avoir donné un avis favorable à ma demande de stage ;
- Monsieur Victor OYETOLA, pour le suivi de la rédaction du présent mémoire et pour les conseils prodigués afin de tirer au maximum profit de notre stage ;
- Tout le personnel de JScom pour leur sens des valeurs ;
- Tous les professeurs de l'ENEAM, spécialement ceux de la spécialité Informatique de Gestion pour nous avoir inculqué le savoir, pour leur nombreux conseils et pour leur contribution à la formation de l'informatique au Bénin.

Que Dieu vous Bénisse.

Picasso Houessou

Sigles et Abréviations

BASH : Bourne Again Shell.

CSS : Cascading Style Sheets.

ENEAM : Ecole Nationale d'Economie Appliquée et de Management.

HTML : HyperText Markup Language.

IMAP : Internet Message Access Protocol.

LTS : Long Term Support.

LVM : Logical Volume Management.

LTS : Transport Layer Security.

OSI : Open Systems Interconnection.

RAID : Redundant Array of Independent Disks.

SMTP : Simple Mail Transfer Protocol.

Table des matières

Dédicaces	i
Remerciements	ii
Sigles et Abréviations	iii
Table des matières	v
Introduction	1
1 Cadre de l'étude	2
1.1 Présentation du bureau d'étude JScom	2
1.1.1 Présentation générale	2
1.1.2 Situation géographique	2
1.2 Démarche méthodologique	3
2 Cadre théorique et méthodologique	4
2.1 Contexte et énoncé du problème	4
2.2 Objectif	5
2.3 Hypothèse	6
2.4 Etude théorique	6
2.4.1 Définition d'un serveur	6
2.4.2 Disque dur RAID LVM	7
2.4.3 Serveur web APACHE	7
2.4.4 Base de donnée	7
2.4.5 Modélisation avec GNS3	8
2.4.6 Programmation	8
2.4.7 FTP	9
2.4.8 Fonctionnement du mail	9
2.4.9 Sécurité	12
3 Implémentation du projet	14
3.1 Installation du serveur	14
3.2 Gestion du stockage de fichier	15

3.2.1	Système de fichier	15
3.2.2	Ajout de disque dur au serveur	15
3.3	Configurer le LAMP	17
3.3.1	Gestion de Apache	18
3.3.2	Gestion de Nginx	20
3.4	Modélisation de l'architecture réseau avec GNS3	23
3.4.1	Description du schéma	24
3.4.2	Configuration des équipements	27
3.5	Installation Postfix	27
3.6	Installation de Dovecot	31
3.7	Installation du client mail	34
3.8	Le site d'administration	36
3.9	Configuration du FTP avec vsftpd	40
3.10	Spamassassin	42
3.11	La base de donnée MariaDB	43
3.12	Sécurité	48
3.13	Cas concret	49
3.13.1	Enoncé	49
3.13.2	Pratique	49
	Conclusion	57
	Bibliographie	58

Table des figures

1.1	Situation géographique de JScom, source Google maps.	3
2.1	Principe du SMTP	10
2.2	Réception d'un mail en IMAP	11
2.3	Résumé de l'envoi et de la réception d'un mail	12
3.1	Résumé du fonctionnement web du réseau projetmail	23
3.2	Topologie de notre projet dans GNS3	24
3.3	Connexion à pfsense par un navigateur	25
3.4	Configuration du port forwarding	26
3.5	Ajout des serveurs mails dans rainloop	35
3.6	Connexion de l'administrateur au site d'administration	50
3.7	Création du compte bake@eneam.da	51
3.8	Connexion de Baké au client webmail	52
3.9	Envoi d'un mail de Baké à Toto	52
3.10	Réponse de Toto au mail de Baké	53
3.11	Lecture du mail reçu de Baké par Toto	54
3.12	Suppression du compte de Béréké	55
3.13	Vérification de l'état des services	56
3.14	Arrêt des services Postfix et Dovecot	56

Introduction

Durant mon parcours à ENEAM, mes camarades et moi avons passé de bons moments mais nous avons été également confrontés à plusieurs problèmes. Les problèmes les plus récurrents sont liés à la programmation des cours et à la disponibilité des professeurs. En effet , les cours ne sont possibles que si les professeurs sont disponibles. Très souvent les professeurs réadaptent le programme des cours selon leur disponibilité. J'ai constaté que certains étudiants sont mal informés de l'annulation d'un cours et parfois de l'effectivité d'un cours, cela même parfois lors des devoirs. Je fus surpris lorsqu'en troisième année d'étude un de mes camarades était venu très en retard à une composition parce qu'il n'était pas informé.

Il est donc impératif que nous avons besoin d'un moyen de communication puissant, sûr, efficace, centralisé, sécurisé pour discuter et pour s'échanger les informations qui circulent au sein de l'école. Cet moyen devra servir non seulement aux étudiants mais va aussi constituer un moyen très efficace dont disposera l'administration et le corps enseignant pour les échanges en milieu scolaire.

L'objectif est donc de proposer un essai de solution pour aider et faciliter la communication au sein de l'ENEAM. Par communication, nous entendons toutes informations utiles que l'école peut mettre à disposition des étudiants, mais aussi que les professeurs peuvent échanger avec leurs apprenants, de même que les apprenants peuvent s'échanger entre eux. C'est ainsi qu'en tant qu'étudiant en informatique de gestion à l'ENEAM, je me propose de répondre à cette problématique par la mise en place d'une solution complète de messagerie électronique au sein de mon école.

L'étude dudit thème va articuler autour de trois chapitres à savoir :

- La présentation du cadre d'étude ;
- La présentation du cadre théorique et méthodologique ;
- L'implémentation de la solution ;

Chapitre 1

Cadre de l'étude

1.1 Présentation du bureau d'étude JScom

1.1.1 Présentation générale

JSCOM BENIN Sarl est une société béninoise économiquement autonome. Elle fait partie d'un réseau d'ingénieurs présents en France et aux États-Unis spécialisés en système d'information et en nouvelles technologies de l'information et de la communication. Aujourd'hui JSCOM BENIN avec ses partenaires en France et dans la sous-région, met son expérience au service de l'informatisation d'entreprise et de la mise en place d'un intranet/Internet dans les administrations et les entreprises.

JSCOM-Bénin est spécialisée, entre autre activités de mise en place des systèmes d'information structurée en entreprise en en administration, dans l'implémentation des solutions électroniques de déclaration fiscale, du contrôle fiscal et de la facturation électronique normalisée et certifiée.

1.1.2 Situation géographique

Voici la figure illustrant la situation géographique de JSCom.

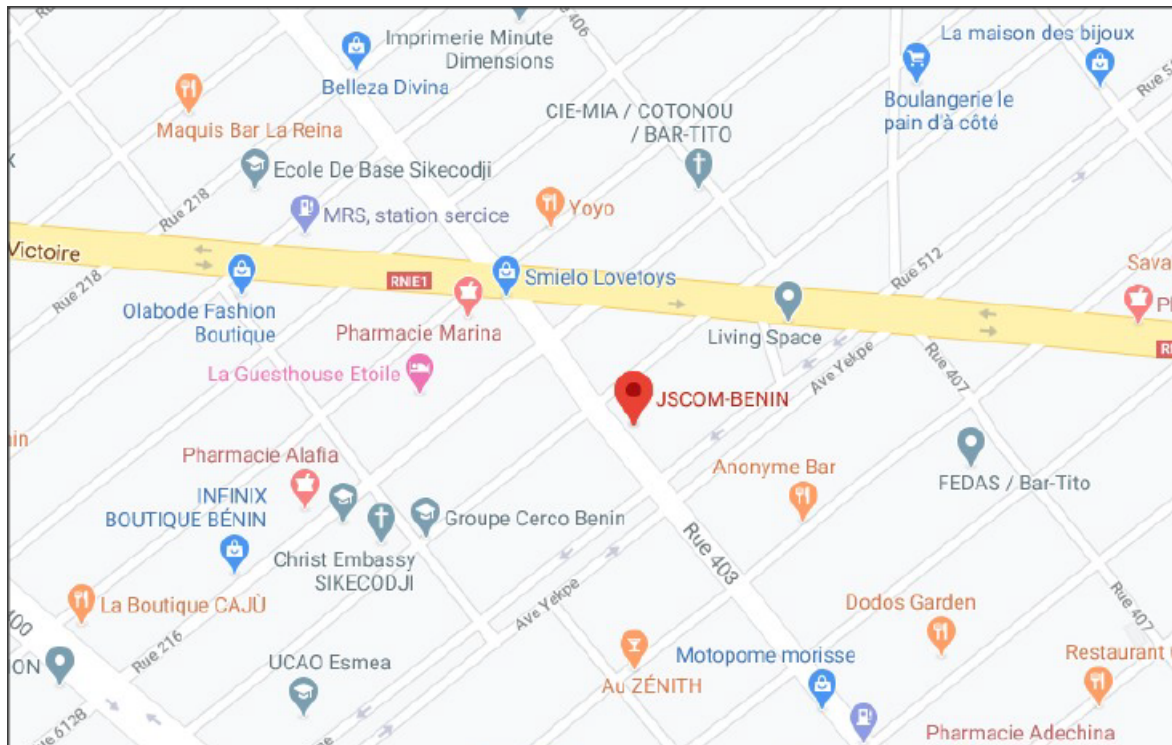


FIGURE 1.1: Situation géographique de JScom, source Google maps.

1.2 Démarche méthodologique

Au cours du stage , il a fallu :

- Faire une revue littéraire sur les notions du réseau informatique ;
- Prendre connaissance de la tâche à effectuer ;
- Me référer à mes connaissances reçues en classe ;
- Effectuer les opérations sur le terrain ;
- Faire le traitement au bureau ;
- Faire des recherches sur le net ;
- Faire des recherches documentaires ;
- Lire beaucoup de cours ;
- Surtout Pratiquer.

Chapitre 2

Cadre théorique et méthodologique

Dans ce chapitre, je vais présenter le thème et toutes les notions fondamentales qui ont rapport au thème.

2.1 Contexte et énoncé du problème

J'ai constaté qu'il a beaucoup d'incompréhensions dans la programmation des cours à ENEAM. Les étudiants s'embrouillent sur l'effectivité de la tenue des cours. De même quand on annule un cours les élèves sont parfois mal informés. Souvent ils sont à l'école et s'amusent et perdent leur temps à ne rien faire. L'administration diffuse certaines informations qui sont sur support papier et sont collées. Et tout le monde n'est pas au courant ; je me rappelle en 3^e année, un de mes camarades est venu très en retard à une composition parce qu'il n'était pas informé de la date des évaluations. Que puis-je faire ? Les étudiants ont besoin d'un moyen de communication puissant pour discuter et ou s'échanger à propos des cours. Comme moyen, il a les réseaux sociaux. Les réseaux sociaux ne répondent pas à ce problème. En effet, on peut communiquer de façon illimitée par les réseaux sociaux, ce qui poussent les étudiants à dire des inepties par ces canaux. Nous avons donc besoin d'un système qui va remplir plusieurs conditions :

- Les professeurs pourront communiquer la programmation des cours aux apprenants de façon efficace ;
- Les élèves pourront échanger de façon simple ;
- Les administrateurs pourront surveiller le flux d'informations qui transite par le réseau ;
- L'administration pourra diffuser des informations et des communiqués ;
- L'administration pourra renforcer la qualité de service : on pourrait créer un canal spécial et périodique pour certaines procédures telles que les procédures de réclamation et autres services disponibles temporairement ;
- Un système de communication sûre, multi-utilisateurs, sécurisée.

Étudions les options dont on dispose. En informatique, on pourrait contribuer en réalisant une application, un réseau social. Présentons ces avantages. Une bonne application peut répondre à tous les problèmes mentionnés ci-dessus mais la conception est une tâche bien pénible

surtout lorsqu'on parle de sécurité, de système multi-utilisateurs, multitâches et d'administration. Il suffit pour s'en convaincre d'observer les médias sociaux qui existent : c'est beaucoup de technologies réunies. Nous voyons bien que cette solution bien que réalisable, pour qu'elle réponde à tous nos besoins, il faut beaucoup de ressources. Quand on dit ressources ici il y'a les ressources matérielles, humaines , le temps, etc. Au vu de tous ces disconvenues essayons de trouver une autre solution. Comme je suis un étudiant en administration réseaux, je vais essayer de parcourir parmi les solutions que proposent l'incroyable monde du réseau.

Prenons le mail. Le e-mail ou courrier électronique est un système de transmission de messages électroniques entre différents utilisateurs. Ce système achemine les messages entre deux nœuds reliés entre eux par le réseau internet. Par message électronique, on comprend des messages textes mais aussi des messages enrichis (c'est à dire des messages formatés dans un langage de description des données : HTML), des fichiers joints (documents , musique , vidéo) et tout autre type de fichiers. Le mail a plusieurs avantages :

- C'est un réseau centralisé : les administrateurs peuvent donc surveiller le flux réseau, limiter le nombre de mails envoyés par les étudiants ; ce qui empêche les apprenants d'envoyer des messages jugés inutiles ou des blagues sur le réseau. Plusieurs filtres peuvent être définis pour mieux analyser les données qui transitent sur le réseau ;
- Les membres de l'administration peuvent envoyer le programme des cours, des devoirs aux étudiants, donc tout type d'information ;
- les élèves peuvent s'échanger entre eux sur les notions reçues ;
- La sécurité est au point : le mail fait intervenir des protocoles réseaux standardisés et sécurisés qui répondent à tous les principes fondamentaux de la cryptographie. C'est un moyen de communication sûr lorsqu'il est bien configuré ;
- il a aussi des avantages personnels : l'apprentissage et l'exploration d'un riche et vaste domaine ainsi que la maîtrise de ces champs d'application.

Au vu de ces avantages, je me propose de mettre en place un serveur de messagerie électronique. Pour cela, je vais procéder étape par étape. Je vais donc mettre en place un serveur simple Linux, monter les disques durs, installer les services de bases DHCP, DNS, Apache, MySQL, faire de la virtualisation, faire de la modélisation réseau grâce à GNS3, configurer le mail proprement avec les protocoles SMTP et IMAP, penser à la sécurité en définissant une politique de sécurité, en configurant les firewalls , en utilisant de la cryptographie (chiffrement symétrique , asymétrique , certificat, fonction de chiffrement de mot de passe), mettre en place la supervision pour prévenir, détecter et corriger les problèmes, programmer également un petit site web d'administrations, écrire des scripts Bash.

2.2 Objectif

L'objectif est de mettre en place un serveur mail opérationnel pour l'ENEAM. De façon spécifique, il s'agit de déployer un serveur de messagerie et d'y proposer un accès par la technologie web.

2.3 Hypothèse

Les problèmes de communication liée à la programmation des cours sont dus à l'absence d'un moyen de communication innovant, combinant les technologies de l'information au sein de l'ENEAM.

2.4 Etude théorique

2.4.1 Définition d'un serveur

Un serveur est un ordinateur qui fournit un ou plusieurs services aux clients . Les clients et le serveur communiquent grâce à des protocoles réseaux. En réseau, un protocole est un langage qui est bien défini par des règles et qui permet aux ordinateurs (en réalité tout équipement électronique qui possède une carte réseau) de communiquer. Du point de vu logiciel, un serveur peut être vu comme un logiciel qui fournit un service à d'autres logiciels.

Les caractéristiques d'un serveur

Un serveur à généralement les caractéristiques suivantes :

- Un serveur est allumé 24h/24h ;
- Très souvent, il ne dispose pas d'un écran , ni d'un clavier , ni d'équipements multimédias ;
- Un serveur Linux n'a généralement pas d'interface graphique ;
- Un serveur utilise très souvent un système d'exploitation spécialisé.

Quelques services

Les serveurs assurent différents services. Citons quelques uns :

- Transfert de fichier : NFS, Samba, bittorrent, FTP ;
- Communication : Messagerie instantanée, téléphonie par IP ;
- Authentification : annuaire LDAP ;
- Web.

Spécification d'un OS seveur

Un système d'exploitation de serveur (OS Operating System) n'est qu'un système d'exploitation optimisé pour l'installation de logiciels serveurs. Les OS serveurs ont les caractéristiques suivantes :

- Les OS serveurs ne sont pas configurés avec les fonctions de veilles. En effet les serveurs restent allumés tout le temps ;
- Les OS serveurs n'ont pas d'interface graphique pour les systèmes Unix et Linux ;

- Ils peuvent gérer des ressources physiques énormes par rapport à un ordinateur personnel. Ils peuvent donc gérer par exemple une machine dotée de plus 1 tera (1 To) de mémoires RAM. Il existe des services de support payants pour les entreprises.

2.4.2 Disque dur RAID LVM

RAID est un ensemble de techniques qui visent à répartir le stockage de données sur plusieurs disques physiques afin d'anticiper la défaillance des disques et de limiter les risques de pertes données. Son principe est simple : on regroupe plusieurs disques pour constituer un seul disque dur visible par l'utilisateur. Ainsi lorsque l'un des disques se détériore, il suffit de le changer. On distingue plusieurs architectures RAID :

- Le RAID 0 : dans cette architecture, si on prend deux disques, les deux travaillent en parallèle. Si un disque est détérioré toutes les données sont perdues ;
- Le RAID 1 : Pour deux disques durs physiques A et B , les données sont écrites simultanément sur les deux disques. Ainsi si A se gâte on peut continuer à travailler sans perte de données. Il suffit après de changer le disque A détérioré par un nouvel disque C pour ne pas risquer de perdre les données définitivement au cas où le seul disque restant s'endommage ;
- Il existe d'autres architectures RAID.

2.4.3 Serveur web APACHE

Un serveur web est le service qui permet d'accéder à des pages web. Tout le web repose sur ce service. On aura à accéder à notre serveur de mail par une interface web. Il faudra donc mettre en place un serveur web. Apache ou plus précisément HTTPD est le serveur web le plus populaire. Il est développé et maintenu par la fondation *Apache*. Il sera installé dans la suite. On va aussi utiliser le serveur web Nginx comme reverse proxy ¹.

2.4.4 Base de donnée

Les données des utilisateurs doivent être stockées dans une base de donnée. Une base de donnée permet de stocker, de structurer, de gérer et d'accéder aux données de façon sûre, rapide et sécurisée. Un SGBD (Système de Gestion de Base de Donnée) est un logiciel qui manipule une base de donnée. Une base de donnée est un fichier ou un ensemble de fichier qui contient des données bien organisées qui peuvent être lues et manipulées par un SGBD à travers un langage informatique (langage de description ou langage de programmation). On distingue deux principaux types de bases de données.

- Les bases de données relationnelles : les données sont stockées dans des tables et sont liées entre elles par des relations. On utilise le langage SQL pour interagir avec les données.

1. Un reverse proxy est un proxy qui filtre les requêtes de l'extérieur à destination d'un serveur interne. Il est utile pour la sécurité et les performances car il permet de gérer aussi un système de cache

Comme SGBD de ce type, on distingue MySQL, PostgreSQL, Oracle, et plein d'autres. En SQL, l'instruction :

```
— SQL —  
SELECT `nom`, `prenom` FROM `utilisateur` WHERE `utilisateur`.`age`  
↪ >=18 ; --Permet de sélectionner les nom, prénoms de tous les  
↪ utilisateurs majeurs.
```

- Les bases de données NoSQL (Not Only SQL) : ceux sont des bases de données qui n'utilisent pas le modèle relationnel. On peut citer à titre d'exemple Poids, MongoDB, Cassandra, ElasticSearch.

2.4.5 Modélisation avec GNS3

Pour mettre en place notre projet il va falloir la modéliser, c'est à dire représenter toute notre architecture réseau de façon visuelle pour avoir un modèle, un plan représentatif de la solution à déployer. GNS3 est un logiciel libre qui permet de modéliser des architectures réseaux, de simuler des architectures réseaux, de visualiser et de tester le résultat grâce à la virtualisation.

2.4.6 Programmation

Un ordinateur ne comprend que le langage binaire , c'est à dire 1 et 0. Pour pouvoir donc donner une instruction à un ordinateur, il va falloir lui parler son langage qu'il comprend (le binaire). Comme le binaire n'est pas un langage accessible aux humains, on a créé les langages de programmation qui sont des instructions écrites dans un langage accessible à l'homme et qui seront ensuite soient lues , soient compilées, soient interprétées par des programmes spécifiques pour permettre à l'ordinateur de comprendre et donc de réaliser une tâche.

HTML, CSS, PHP, JAVASCRIPT, JQUERY, BASH

Les langages de programmation web sont des langages de programmation qui interviennent dans le web. C'est donc un ensemble de technologie qui permettent de créer des pages web dynamiques.

HTML est un langage de description. C'est un langage dérivé du XML et qui permet d'écrire une page web statique.

CSS est un langage de description qui va permettre de rendre le contenu HTML joli(de faire la mise en forme du contenu).

PHP est un langage coté serveur, il va permettre de manipuler les données reçues par notre interface web (site web).

JAVACRIPT ET JQUERY : Javascript est le langage qui va nous permettre de rendre responsive le site web coté client. JQuery est un framework du javascript, c'est donc une brique de code déjà implémenté dont on va se servir pour vite développer notre application.

BASH est un langage de script qui permet d'exécuter des instructions sur un système d'exploitation Linux. Par exemple, on peut écrire un script bash pour dire d'éteindre un ordinateur dans 10 heures de temps. Il est intégré dans tous les systèmes Linux et ne nécessite aucune installation. Il est essentiel car c'est lui qui va nous permettre de créer ou de supprimer les répertoires des utilisateurs, de vérifier l'état d'un service, d'arrêter ou de redémarrer un service.

2.4.7 FTP

Pour envoyer le site d'administration que nous allons développer sur le serveur, nous n'allons pas utilisé le terminal mais plutôt utilisé le protocole d'envoi de fichier FTP et se servir d'un client FTP graphique (FilleZilla). Le protocole FTP permet l'envoi et la réception de fichiers. Il est très utilisé pour le téléchargement de fichiers. La majorité des dépôts Linux utilisent le protocole FTP pour le téléchargement des packages (apt le gestionnaire de paquet Debian télécharge les paquets sur des serveurs FTP).

2.4.8 Fonctionnement du mail

La messagerie informatique fait intervenir plusieurs protocoles réseaux. Il a en effet le protocole SMTP qui permet d'envoyer le message et les protocoles IMAP, POP pour accéder aux données (les mails).

SMTP

Le serveur SMTP permet d'envoyer un mail. Enonçons son fonctionnement par un exemple. Si Toto veut envoyer un mail à Baké. Toto a pour adresse mail **toto@toto.com**. Baké a pour adresse **bake@gmail.com**. Il va se reproduire les étapes suivantes :

- Toto va envoyer le mail depuis son poste à son serveur SMTP.
- Le serveur SMTP de Toto va réceptionner le mail et vérifier s'il lui est destiné (il va vérifier si le destinataire du mail est sur ce serveur, c'est à dire si Baké appartient à ce serveur. **toto@toto.com** a pour domaine **toto.com**. et **bake@gmail.com** a pour domaine **gmail.com**. **toto.com** est différent de **gmail.com** donc les deux comptes mail ne sont pas sur le même serveur). Si oui, il va enregistrer le mail dans le répertoire de réception des mails de Baké. Si non, il va relayer le mail vers le serveur SMTP de Baké.
- Le serveur SMTP de Baké va recevoir le mail de Toto et le stocker dans le répertoire personnel de Toto.

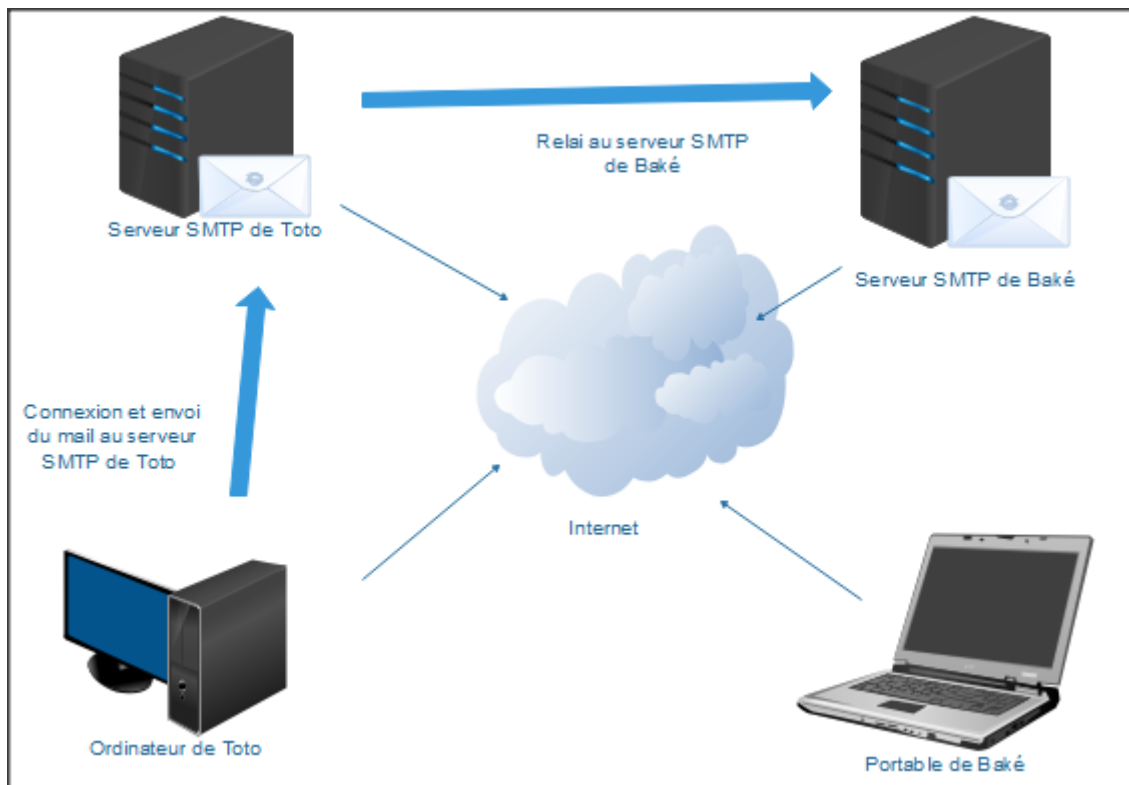


FIGURE 2.1: Principe du SMTP

IMAP

Pour lire mes mails, je dois pouvoir accéder à mon serveur SMTP et lire les messages à distance. Pour cela on utilise le protocole IMAP qui va se connecter au serveur et récupérer les données et l'afficher dans le logiciel qui permet de lire les mails (un exemple Mozilla Thunderbird, Yahoo mail, Zimbra, Rainloop).

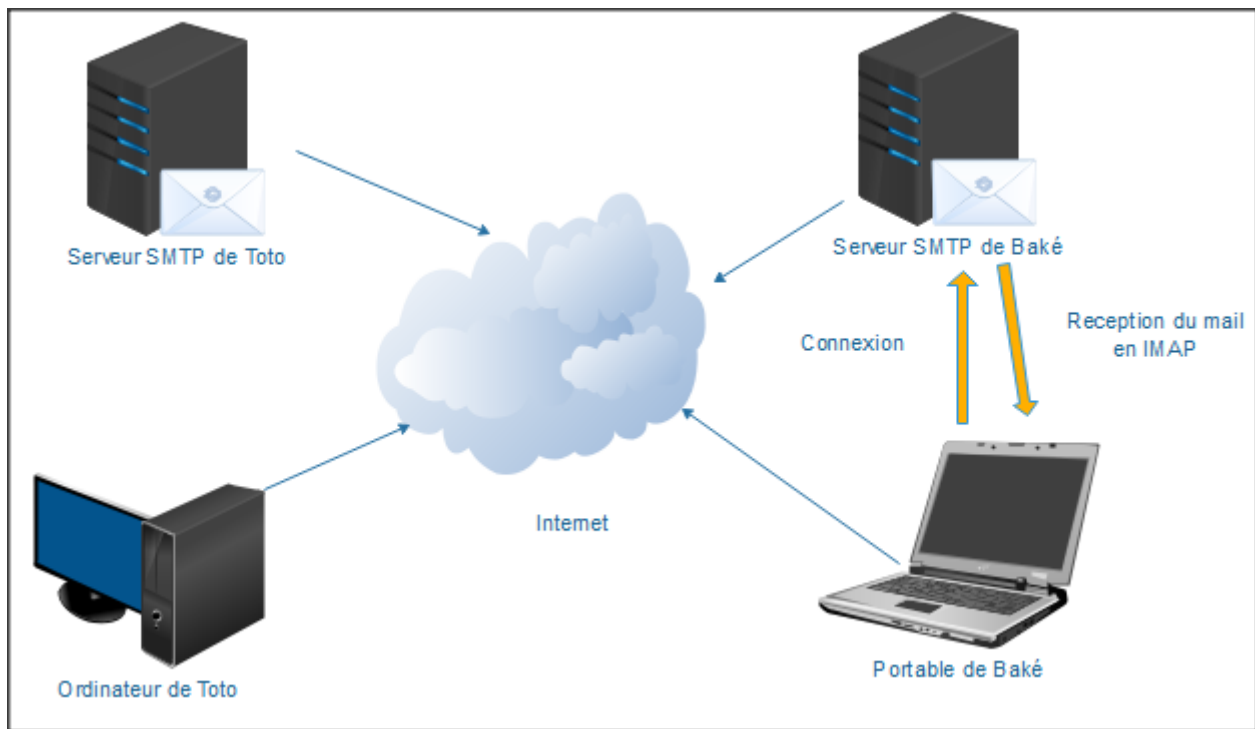


FIGURE 2.2: Réception d'un mail en IMAP

Le protocole POP permet aussi de récupérer les mails. La différence majeure entre IMAP et POP est que IMAP est une connexion directe, unidirectionnelle entre le serveur le client mail tandis que POP est une connexion bidirectionnelle.

- Cela signifie qu'avec IMAP les données sont sur le serveur et on y accède directement et toute modification faite sur les mails depuis le client mail impacte le serveur (est réalisée sur le serveur). En IMAP, si je supprime des mails depuis mon client mail, les mails sont aussitôt supprimés sur le serveur. De même avec IMAP, il est impossible d'avoir une copie local des mails . On ne peut donc accéder aux mails qu'on a déjà lu sans connexion internet.
- Avec POP, on se connecte et on réalise une copie locale de tous les mails qui sont sur le serveur. Ensuite, en interne on peut modifier ses mails sans enregistrer les modifications sur le serveur. On peut lire tous les mails téléchargés hors connexion.

Il est conseillé de préférer le protocole IMAP au protocole POP car si on crée plusieurs copies locales d'un mail qui est sur le serveur, on peut facilement se tromper et écraser après des données sur le serveur sans le vouloir. Ou même supposons que j'ai récupéré en POP le mail identifié par *MAIL1* avec mon ordinateur portable. J'ai naturellement commencé par répondre à se mail et j'ai enregistré le brouillon de la réponse *REPONSEMAIL1* sans avoir à envoyer la réponse au serveur. Je sors après sans mon pc et je réalise que je veux envoyer la réponse rédigée précédemment. Je me connecte à mon serveur mail avec mon portable et je ne retrouve par ce brouillon car le brouillon *REPONSEMAIL1* est enregistré sur mon pc et non sur mon serveur mail. Il m'est donc impossible d'envoyer cette réponse sans passer par mon pc ; ce qui n'allait pas poser problème en IMAP puisque avec ce dernier le brouillon *REPONSEMAIL1* aurait été enregistré sur le serveur.

Le schéma complet de l'envoi et de la réception d'un mail.

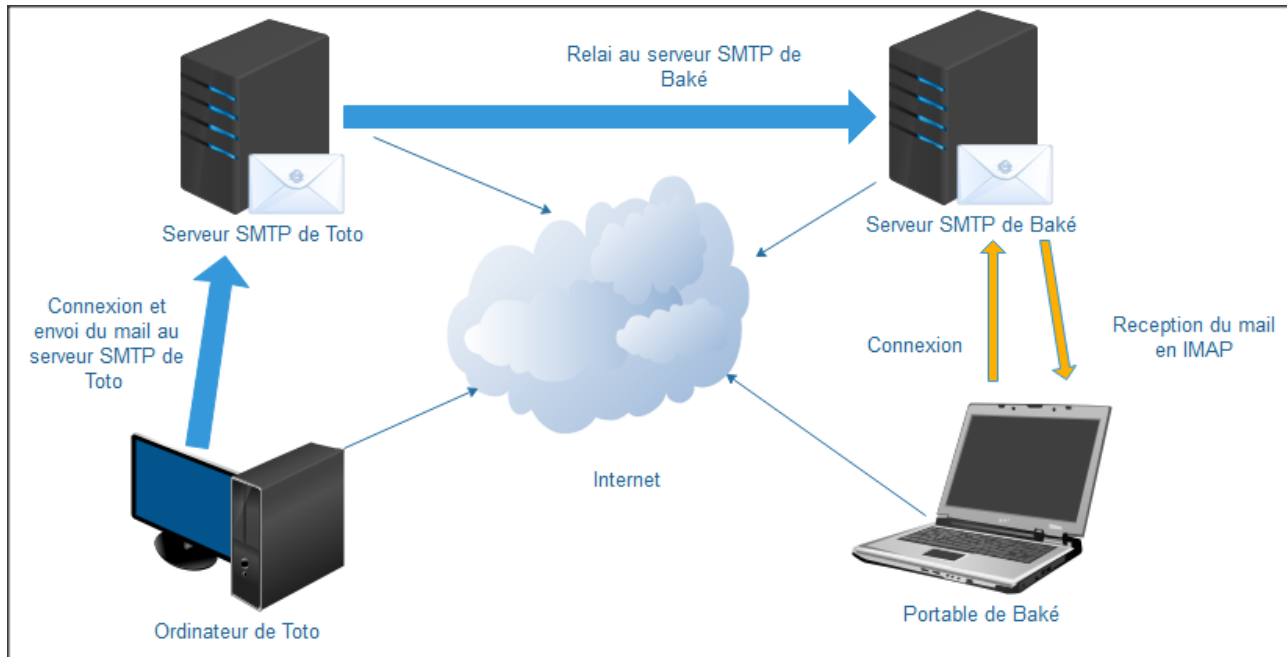


FIGURE 2.3: Résumé de l'envoi et de la réception d'un mail

2.4.9 Sécurité

En matière de sécurité on ne peut pas faire confiance aux utilisateurs et donc aux données reçues par les utilisateurs. Il faut alors établir des règles de sécurité strictes pour limiter les risques de mauvaises utilisations et de piratage informatique.

Firewall

Le firewall (pare-feu en français) est l'instrument qui va permettre de protéger notre serveur contre l'extérieur. Il va installer une barrière entre notre serveur et l'extérieur. On pourra dire par exemple au serveur d'aller sur internet (communication serveur vers extérieur) ou empêcher les utilisateurs depuis l'extérieur à communiquer avec le serveur² (communication extérieur vers serveur). On distingue deux types de firewalls : les firewalls matériels et les firewalls logiciels (ou proxy).

- Le firewall matériel est une protection pour la couche 3 et 4 du modèle OSI. Il filtre le trafic réseau en lisant les entêtes IP et TCP ;
- Le proxy est un firewall de niveau 6. Il filtre la couche applicatif. C'est lui qui va par exemple bloquer les étudiants à se connecter aux serveurs à minuit (00 heures). Il sert aussi de cache.

Conclusion

Nous allons installer un serveur Linux ubuntu qui va contenir un certain nombre de services :

2. Il a un troisième cas , on traverse le serveur. Dans ce cas le serveur joue le rôle d'un routeur

- Un serveur SMTP et IMAP pour l’envoi et la réception des mails ;
- Un serveur web pour envoyer les mails depuis une application web (un webmail) ;
- Un site web d’administration pour créer des comptes mails et réaliser quelques tâches d’administration ;
- Une base de donnée pour sauvegarder les informations d’authentification ;
- Un serveur FTP pour envoyer les données du site d’administration sur le serveur.

Nous allons :

- Coder le site web en PHP et en javascript. Les mails seront stockés sur un disque dur RAID, LVM ;
- Ecrire des règles firewalls pour protéger le serveur ;
- Configurer un antispam pour empêcher les spammeurs (Spamassassin) ;
- Sécuriser toute la communication par le chiffage et des certificats ;
- Utiliser comme nom de domaine eneam.da³. Le webmail aura pour adresse www.eneam.da et le site d’administration www.admin.eneam.da.

3. On pourra changer après ce nom en production

Chapitre 3

Implémentation du projet

Je vais dans un premier temps me mettre dans la peau d'un administrateur système et configurer tous les services réseaux. Dans un second temps, dans la section 3.13 à la page 49, je vais me mettre à la place de l'utilisateur du système et je vais créer des comptes mails et envoyer des mails de façon simple et pratique.

3.1 Installation du serveur

Nous allons utiliser pour notre travail comme OS **Ubuntu Server** et VMware comme hyperviseur¹. Ubuntu est une distribution Linux gratuite et très populaire. Nous allons utiliser la version LTS d'Ubuntu Server qui est disponible sur le site d'Ubuntu . La version LTS signifie Long Term Support qui correspond à une version d' Ubuntu qui sort tous les 2 ans et qui bénéficie d'un support étendu sur 5 ans. L'avantage est qu'on ne va pas s'embêter avec des mises régulières comme avec un OS standard. Le choix de l'hyperviseur VMware est axé sur le fait qu'il supporte une meilleure intégration avec l'émulateur d'architectures réseaux GNS3.

On installe d'abord l'hyperviseur VMware workstation. Pour cela rien de plus simple on se rend sur le site de VMware en cliquant **ici**. On télécharge et installe le logiciel. On ouvre le logiciel. J'utilise la version en anglais de VMware. On ouvre l'onglet File ensuite "New virtual Machine". Une fenêtre s'ouvre et nous suivons le guide durant la configuration de la nouvelle machine virtuelle. Les étapes d'ajout d'une machine virtuelle sont disponibles sur le site de VMware. Nous avons nommé la machine virtuelle serveur et elle a les configurations suivantes 1 Go de mémoire RAM , 40Go de stockage de masse (disque dur).

On démarre la machine virtuelle **"serveur"** avec le disque virtuel Ubuntu Server qu'on a téléchargé **ici** pour démarrer l'installation. On suit les instructions (on choisit la langue, le clavier, on configure le réseau en DHCP). A la fin de l'installation le système nous dit de redémarrer. Après redémarrage, on entre le nom d'utilisateur et le mot de passe. Il faut configurer le réseau. Pour cela il faut déjà avoir d'interfaces réseaux dans la machine virtuelle. On éteint la machine avec la commande

1. Un hyperviseur est un logiciel qui fournit un environnement virtuel pour installer un OS sans avoir un matériel réel. C'est grâce au hyperviseur qu'on pourra installer Linux et tester notre projet depuis un poste Windows

Console

```
sudo shutdown now
```

Dans VMWare on édite les paramètres de la machine virtuelle et on lui ajoute un adaptateur réseau (on choisit NAT). La NAT va permettre à notre machine d'avoir accès à internet ². On démarre la machine. On tape la commande

Console

```
netplan apply
```

3.2 Gestion du stockage de fichier

Nous allons configurer un disque dur spécial pour les données liées aux mails que le serveur va contenir.

3.2.1 Système de fichier

Toutes les données contenues dans un ordinateur sont du binaires(suite de 0 et 1). Il n'a aucun moyen pour un homme de distinguer alors les données. Pour cela on a créé les fichiers. Un fichier est alors un contenu (ou un conteneur de données) binaire qui porte un nom et a une extension qui permet de le distinguer. Ainsi, il est facile pour un homme de reconnaître une image par son extension jpeg ou un document word par docx. Les fichiers sont stockées sur un stockage de masse (disque dur, carte mémoire). Un système de fichier est un index qui définit comment les fichiers sont stockés et organisés sur le stockage afin de permettre et faciliter l'accès aux différents fichiers. On distingue différents systèmes de fichiers fat32 ,NTFS ext3, ext4. EXT4 est l'actuel système de fichier qui est utilisé sur Linux. Il n'est pas compatible Windows, c'est à dire que Windows ne supporte pas le système de fichier EXT4. Donc pour pouvoir stockés des informations sur un disque dur, il faut d'abord le préparer. Cela se fait créant le système de fichier puis en formatant le disque dur.

3.2.2 Ajout de disque dur au serveur

On éteint le serveur. Puis on va dans les paramètres de la machine virtuelle ensuite on ajoute trois disques de 2 giga configurés en SCSI ³.

On redémarre la machine ensuite on exécute les instructions suivantes :

On affiche la liste des disques durs.

Console

```
ls -la /dev/sd*
```

2. cela sera modifié au moment des configurations avec GNS3

3. est une spécification matérielle des disques

Ensuite on verifie la taille de chaque disque avec la commande

```
Console
gdisk dev/sdb
```

Dans gdisk on fait p pour afficher la structure du disque pour être sur que c'est deux giga. q pour quitter. On reprend avec les deux autres disques dur montés.

On crée le raid 1 avec la commande

```
Console
mdadm --create /dev/md0 --level=raid1 --raid-devices=2 /dev/sdb /dev/sdc
↪ spare-devices=1 /dev/sdd
```

On affiche les détails sur le nouveau disque raid créé et on copie l'identifiant UUID

```
Console
mdadm --query --detail /dev/md0
/dev/md0:
    Version : 1.2
    Creation Time : Wed Dec 25 14:33:47 2019
    Raid Level : raid1
    Array Size : 2094080 (2045.00 MiB 2144.34 MB)
    Used Dev Size : 2094080 (2045.00 MiB 2144.34 MB)
    Raid Devices : 2
    Total Devices : 3
    Persistence : Superblock is persistent

    Update Time : Tue Mar 10 11:18:38 2020
    State : clean
    Active Devices : 2
    Working Devices : 3
    Failed Devices : 0
    Spare Devices : 1

Consistency Policy : resync

    Name : serveur:0 (local to host serveur)
    UUID : 1bf998cb:0b420815:25812e34:403d63c5
    Events : 26

    Number  Major  Minor  RaidDevice State
    0        8      16      0      active sync  /dev/sdb
    1        8      32      1      active sync  /dev/sdc
```

```
2      8      48      -      spare      /dev/sdd
```

On ajoute la ligne suivante dans le fichier `/etc/mdadm/mdadm.conf` pour que le disque dur ne change pas de nom lors du prochain démarrage

Console

```
ARRAY /dev/md0 level=raid1 num-devices=2 spares=1
↪ UUID=1bf998cb:0b420815:25812e34:403d63c5
↪ devices=/dev/sdb,/dev/sdc,/dev/sdd
```

Pour que les modifications soient effectives au prochain démarrage du système on fait

Console

```
sudo update-initramfs -u
```

On passe au partitionnement du nouveau disque avec LVM

Console

```
sudo pvcreate /dev/md0
sudo vgcreate raid-volume /dev/md0
sudo lvcreate --name data --size 2000M raid-volume
```

Il faut maintenant monter la partition. Pour cela on crée le dossier de montage de la partition puis on monte notre partition **data** dans ce dossier

Console

```
mkdir -p /externe
mount -t ext4 /dev/raid-volume/data /externe
```

Il faut rendre automatique le montage de la partition permanent à chaque démarrage du système. On édite le fichier `/etc/fstab` et on ajoute la ligne

Console

```
/dev/raid-volume/data /externe ext4 defaults 0 0
```

3.3 Configurer le LAMP

On va installer PHP, MariaDB qui est un fork⁴ de MySQL.

Console

```
sudo su
apt-get update #Pour mettre à jour le cache local
```

4. C'est quand des développeurs d'un logiciel en général libre n'étant plus d'accord pour continuer le projet se séparent et créent leur propre version du logiciel à partir du code source de départ.


```
apt-get install apache2 libapache2-mod-php php-fpm mysql-server
↪ libapache2-mod-rpaf
#Pour activer les modules apache nécessaires
sudo a2enmod proxy_fcgi setenvif
sudo a2enconf php7.2-fpm
sudo a2dismod php7.2 mpm_prefork
sudo a2enmod mpm_event
sudo systemctl restart apache2
```

3.3.1 Gestion de Apache

Nous changeons les ports sur lesquels écoute Apache httpd et il écoutera sur 7080 pour le http et 7443 pour le https en local. L'avantage est que notre serveur apache n'est pas exposé à l'extérieur. On verra seulement notre reverse-proxy Nginx de l'extérieur. On modifie le fichier `/etc/apache2/ports.conf`

Bash

```
Listen 127.0.0.1:7080

<IfModule ssl_module>
    Listen 127.0.0.1:7443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 127.0.0.1:7443
</IfModule>
```

Nous allons configurer les virtualhosts (c'est à dire les deux sites web qui seront hébergés sur le serveur). Ces sites seront sécurisés avec des certificats autosignés⁵. Pour le client webmail qui va permettre l'envoi des mails, on crée le fichier `/etc/apache2/site-available/www.eneam.da.conf`

Console

```
<Virtualhost *:7080>
    ServerName www.eneam.da
    ServerAlias eneam.da
    ServerAdmin houessoupicasso@eneam.da
    <IfModule mod_rewrite.c>
        RewriteEngine On
        RewriteCond %{HTTPS} !=on
        RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI}
↪ [END,QSA,R=permanent]
```

5. Un certificat autosigné est un certificat qui n'est pas créé par une autorité de certification.

```

        </IfModule>

</Virtualhost>

<IfModule mod_ssl.c>
<VirtualHost *:7443>
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apache-autosigne.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-autosigne.key
    ServerName www.eneam.da
    ServerAlias eneam.da
    ServerAdmin houessoupicasso@eneam.da
    DocumentRoot /externe/www/rainloop/public_html
    ProxyPassMatch ^/(.*\.php(/.*)?)$ unix:/run/php/php7.2-fpm.sock|fcg
↪ i://localhost/externe/www/rainloop/public_html/
    ErrorLog /externe/www/rainloop/logs/error.log
    CustomLog /externe/www/rainloop/logs/access.log combined
    <Directory /externe/www/rainloop/public_html>
        Options All
        AllowOverride None
    </Directory>
</VirtualHost>
</IfModule>

```

Pour le site d'administration /etc/apache2/site-available/www.admin.eneam.da.conf

Console

```

<Virtualhost *:7080>
    ServerName www.admin.eneam.da
    ServerAlias admin.da admin.eneam.da
    ServerAdmin houessoupicasso@yahoo.fr
    <IfModule mod_rewrite.c>
        RewriteEngine On
        RewriteCond %{HTTPS} !=on
        RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI}
↪ [END,QSA,R=permanent]
    </IfModule>
</Virtualhost>

<IfModule mod_ssl.c>
<VirtualHost *:7443>

```

```

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apache-autosigne.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-autosigne.key
    ServerName www.admin.eneam.da
    ServerAlias admin.da admin.eneam.da
    ServerAdmin houessoupicasso@yahoo.fr
    DocumentRoot /externe/www/html/www.admin.eneam.da/public_html
    ProxyPassMatch ^/(.*\.php(/.*)?)$ unix:/run/php/php7.2-fpm.sock|fcg
↪ i://localhost/externe/www/html/www.admin.eneam.da/public_html/
    ErrorLog /externe/www/html/www.admin.eneam.da/logs/error.log
    CustomLog /externe/www/html/www.admin.eneam.da/logs/access.log
↪ combined
    ErrorDocument 404 /index.php?page=error
    <Directory /externe/www/html/www.admin.eneam.da/public_html>
        Options All
    </Directory>

</VirtualHost>

</IfModule>

```

On crée aussi les dossiers `/externe/www/html/www.admin.eneam.da/logs/` récursivement.

```

----- Console -----
mkdir -P /externe/www/html/www.admin.eneam.da/logs/
↪ /externe/www/html/www.admin.eneam.da/public_html/

```

Nous n'avons pas activé ces virtualhosts pour le moment sinon Apache va sortir une erreur.

3.3.2 Gestion de Nginx

```

----- Console -----
sudo apt-get install nginx libapache2-mod-rpaf

```

Il faut créer les “**server blocs**”⁶ associés à nos virtualhosts. Pour le premier `/etc/nginx/site-available/www.admin.eneam.da`

```

----- Bash -----
upstream backend_admin.eneam {
    server 127.0.0.1:7443;
}

```

6. Dans Nginx on appelle les virtualhosts server blocs.

```
server {  
    listen 80;  
  
    server_name www.admin.eneam.da admin.eneam.da ;  
    return 301 https://$host$request_uri;  
}  
  
server {  
    listen 443 ssl;  
    server_name www.admin.eneam.da admin.eneam.da ;  
    ssl_certificate /etc/ssl/certs/nginx-autosigne.crt;  
    ssl_certificate_key /etc/ssl/private/nginx-autosigne.key;  
  
    location /{  
        include proxy_params;  
        proxy_pass https://backend_admin.eneam ;  
    }  
}
```

Le fichier /etc/nginx/site-available/www.eneam.da

```
_____ Bash _____  
  
upstream backend_jenkins {  
    server 127.0.0.1:8080;  
}  
  
upstream backend_apache {  
    server 127.0.0.1:7443;  
}  
  
server {  
    listen 80;  
    server_name www.eneam.da eneam.da;  
    return 301 https://$host$request_uri;  
}  
  
server {  
    listen 443 ssl;  
    server_name www.eneam.da eneam.da ;  
    ssl_certificate /etc/ssl/certs/nginx-autosigne.crt;  
    ssl_certificate_key /etc/ssl/private/nginx-autosigne.key;
```

```
location /jenkins {
    include proxy_params;
    proxy_pass http://backend_jenkins;
}
location /{
    include proxy_params;
    proxy_pass https://backend_apache ;
}
#Empêcher l'accès au dossier de configuration par les utilisateurs
↳ du web
location ^~ /data {
    deny all ;
}
}
```

Résumé

En somme, nous avons configuré un serveur web Nginx accessible depuis l'extérieur. Lorsqu'on va se connecter aux différents sites depuis un navigateur, les requêtes (HTTP et HTTPS pour être précis) sont envoyées à Nginx. Nginx filtre ces requêtes et va ensuite rediriger les requêtes en interne vers Apache. Apache en traitant ses requêtes, va déléguer le traitement des scripts PHP au processus FastCGI **PHP-FPM**. Tout ceci se fait de façon transparente pour l'utilisateur.

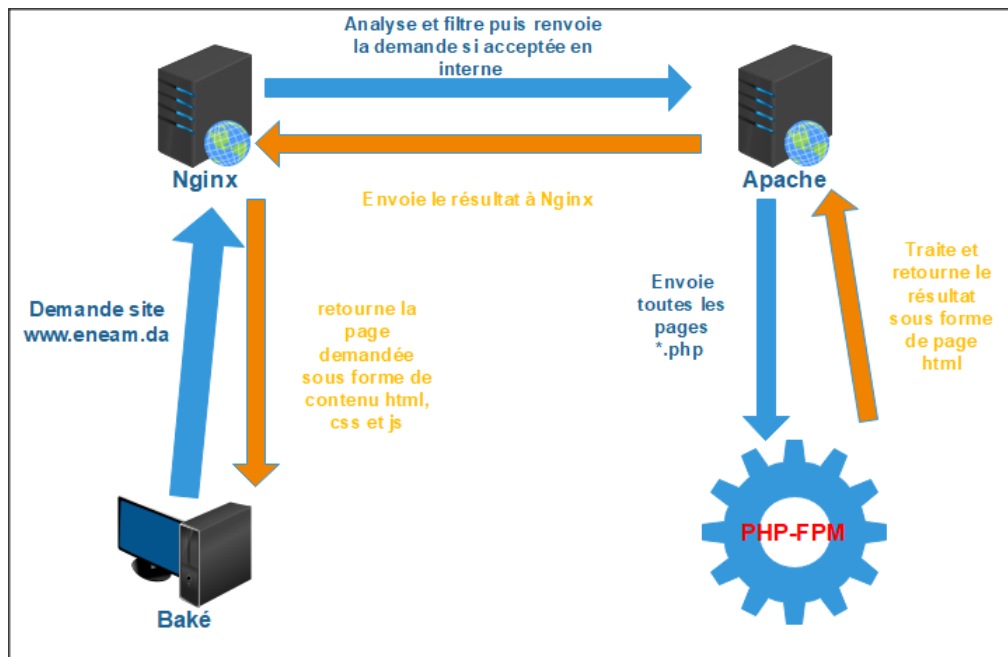


FIGURE 3.1: Résumé du fonctionnement web du réseau projetmail

3.4 Modélisation de l'architecture réseau avec GNS3

On télécharge les logiciels GNS3 et GNS3 VM sur le site officiel de GNS3. On installe GNS3. On lance le logiciel VMWare et dans VMWare on fait :

- Fichier puis ouvrir fichier ;
- On choisit le fichier GNS VM au format ova téléchargé ;

Cela va installer une nouvelle machine virtuelle. Il faut ensuite importer cette machine dans le logiciel GNS3.

- On lance GNS3. Menu Edit -> Préférences -> GNS3 VM ;
- On coche la case "Enable the GNS3 VM" et on sélectionne "VMware Workstation/Player" dans le champ Virtualize engine ;
- Cliquez sur Refresh et GNS3 VM apparaît dans VM name ;
- Cliquez sur Apply.

On va ajouter notre machine Linux "**serveur**" de VMware pour l'utiliser lors de la modélisation

- Dans Preferences -> VMware -> VMware VMs -> new ;
- Une fenêtre apparaît on coche run this VMware VM on my local computer puis next ;
- Dans VM list on choisit la machine serveur un clic sur le bouton finish.

3.4.1 Description du schéma

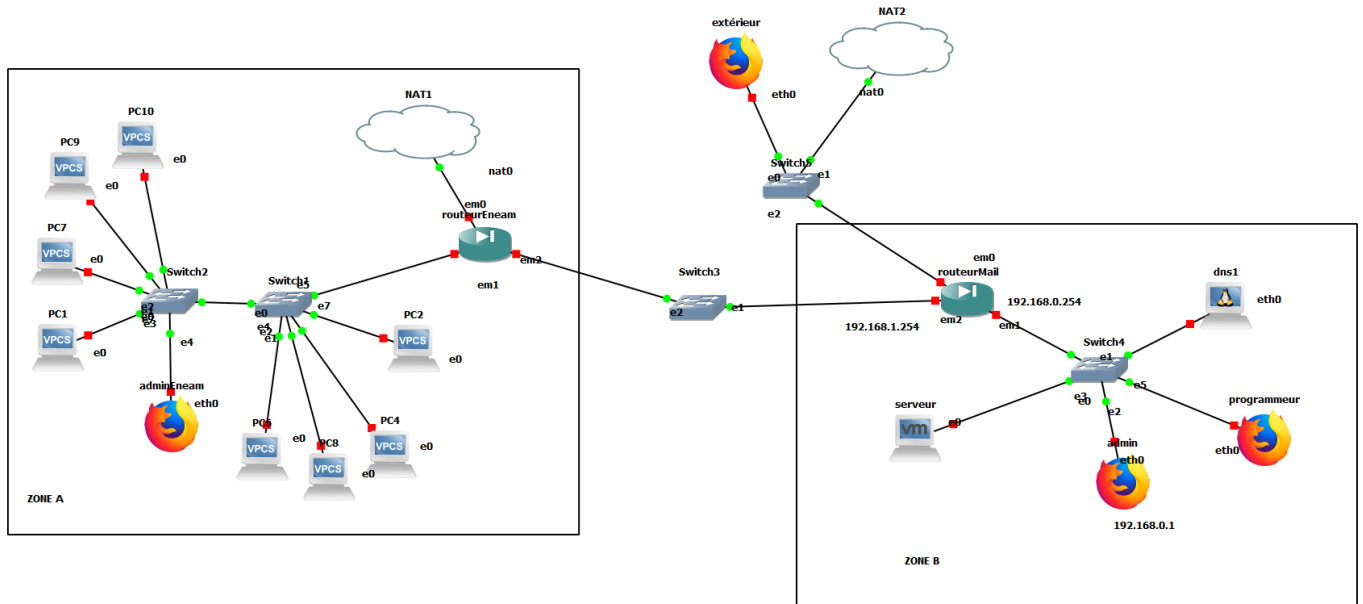


FIGURE 3.2: Topologie de notre projet dans GNS3

Nous avons opté pour un modèle générique. Le réseau est séparé en deux grandes parties : la zone A et la zone B

- La zone A : représente le réseau existant de l'ENEAM. Il est géré par **routeurEneam** et est relié au routeur routeurMail. La configuration de cette zone ne nous intéresse pas. Elle permet de mieux comprendre l'architecture de notre projet et de comprendre que notre projet est générique car il peut s'adapter à tout type d'organisation existante sans toucher au système d'information⁷ existant.
- La zone B : le nouveau réseau relié par le routeur routeurMail. L'avantage d'une telle architecture est que l'administrateur du réseau ENEAM peut communiquer avec le réseau mail à travers un tunnel (VPN) s'il est configuré.
- **routeurEneam** procède trois interfaces réseaux : em1 dans son réseau local, em2 qui le relie à routeurMail et em0 pour le WAN ;
- **routeurMail** de même à trois interfaces ;
- **dns1** est un docker⁸ qui contient un petit serveur DNS qui va servir à la résolution dynamique des noms de domaine au sein du réseau local ;
- **serveur** : c'est notre serveur ubuntu installé depuis VMware ;
- **routeurMail** et **routeurEneam** sont des routeurs Pfsense⁹ ;

7. Un système d'information est un ensemble de ressources matérielles, humaines qui permet de collecter, de stocker, de traiter et de diffuser l'information.

8. Est un conteneur qui contient un service et toutes les dépendances de ce dernier. Il permet d'isoler le service qu'on veut déployer(ici DNS) sans avoir à s'encombrer d'autres services dont on a pas besoin. Sans un docker, on aurait à installer un autre serveur Linux juste pour le DNS mais qui contient déjà plein de programmes dont nous n'avons pas besoin. En somme un docker contient le strict minimum

9. Est un routeur firewall open source.

- **NAT2** permet de connecter routeurMail à internet et de pouvoir simuler l'accès de notre architecture à internet. Notre routeur aura un adressage privé (pour em0) ce qui n'est pas le cas dans la réalité puisque em0 devait avoir une adresse publique directement accessible sur internet. En effet, la modélisation avec GNS3 nous impose de faire comme cela ;

Les étapes de la créations du projet :

- On crée un nouveau projet "projetmail" ;
- On ajoute tous les équipements à notre topologie ;
- On démarre et configure routeurEneam ;
- on configure le serveur DNS sur le poste **dns1** ;
- On accède à routeurEneam depuis le poste admin et on configure la NAT et le port forwarding (redirection de port en français). Pour cela :
 - Depuis le poste **admin**, on ouvre le navigateur Firefox ;
 - On tape l'adresse 192.168.0.254 et on accède à routeurEneam. Les identifiants par défaut sont admin et pour mot de passe pfsense. On change le mot de passe par défaut.

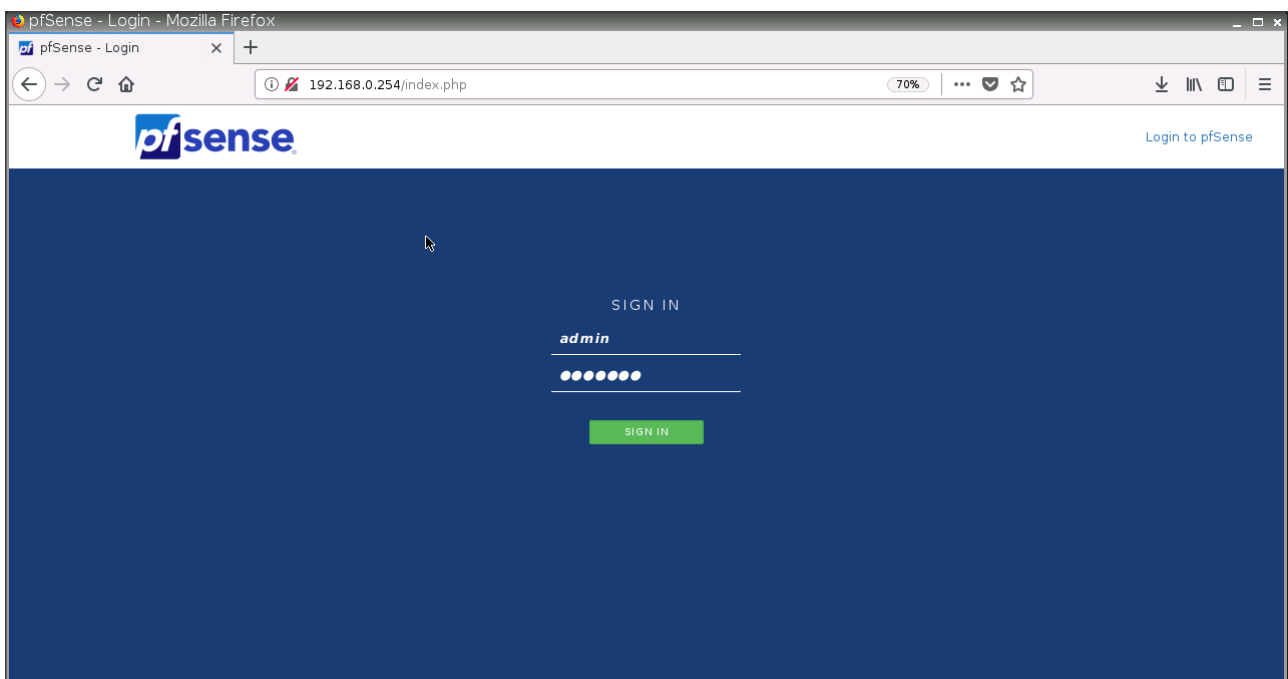


FIGURE 3.3: Connexion à pfsense par un navigateur

Ce qu'on a fait va permettre d'accéder au mail par le webmail mais aussi d'accéder par d'autres clients mail tels que Mozilla Thunderbird puisqu'on a rendu directement les ports SMTP et IMAP ouverts vers l'extérieur. Si on fermait ces ports, il serait impossible de s'envoyer des mails sans passer par un client webmail installé sur le serveur

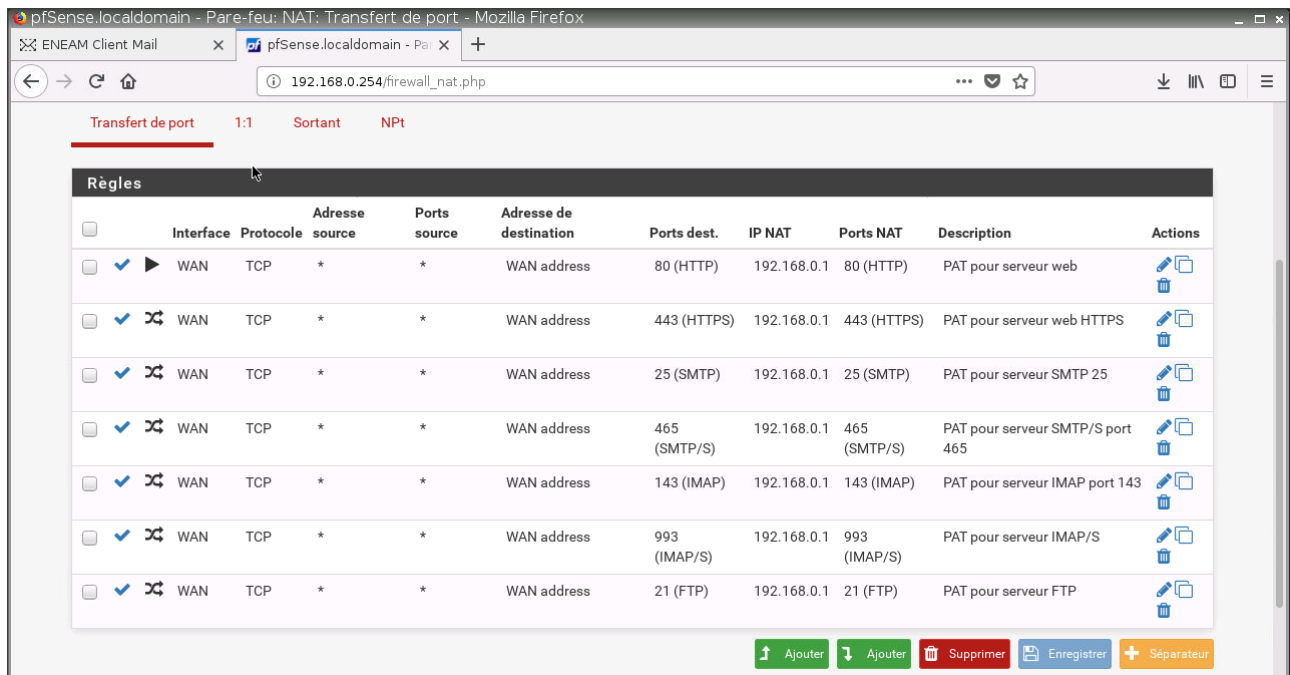


FIGURE 3.4: Configuration du port forwarding

- Il faut modifier les interfaces réseaux de notre serveur nommé *serveur*. Pour cela dans VMware, on le configure pour qu'il ait deux interfaces réseaux ; l'une représente son interface dans lequel il est connecté dans GNS3 (e0 selon notre schéma) et la seconde interface est configuré en Host-Only¹⁰. On crée aussi le fichier `/etc/netplan/60-lan_statique.yaml` pour configurer les interfaces du serveur.

YAML

```
network:
  renderer: networkd
  ethernets:
    ens33:
      dhcp4: no
      dhcp6: no
      gateway4: 192.168.0.254
      addresses: [192.168.0.1/24]
      nameservers:
        search: [eneam.da]
        addresses: [192.168.0.5, 8.8.8.8]
    ens34:
      dhcp4: true
  version: 2
```

¹⁰. En effet, le site d'administration est codé sur ma machine physique Windows, il faut un moyen pour pouvoir envoyer le site sur le serveur virtuel, alors on a créé un réseau privé entre le serveur de VMware et mon ordinateur Windows

3.4.2 Configuration des équipements

3.5 Installation Postfix

L'installation du serveur SMTP Postfix va se faire toujours avec notre gestionnaire de paquet apt

Console

```
apt-get install postfix postfix-mysql #et on reponds aux boites de  
↪ dialogues qui s'affiche
```

Les fichiers de configuration de postfix sont dans le dossier `/etc/postfix/`. On fait une copie des fichiers de configuration avant toute modification. On modifie le fichier `/etc/postfix/main.cf`

Console

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version

#myorigin = /etc/mailname

#smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
smtpd_banner = $myhostname ESMTP $mail_name ( Tout droit réservé à Picasso
↪ Houessou)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no
# Uncomment the next line to generate "delayed mail" warnings
delay_warning_time = 1h

readme_directory = no
compatibility_level = 2

virtual_transport = lmtp:unix:private/dovecot-lmtp
#SASL AUTHENTICATION
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_local_domain =
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_sasl_auth_enable = yes
# TLS parameters
smtp_tls_security_level = may
```

```

smtpd_tls_security_level = may
smtp_tls_note_starttls_offer = yes
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_cert_file = /etc/ssl/certs/dovecot-autosigne.pem
smtpd_tls_key_file = /etc/ssl/private/dovecot-private-autosigne.pem
smtpd_use_tls = yes
#smtpd_tls_CApath = /etc/ssl/certs
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

smtpd_recipient_restrictions = permit_sasl_authenticated permit_mynetworks
↪ reject_unauth_destination
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
↪ defer_unauth_destination
myhostname = vm-serveur
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, serveur, localhost.localdomain, localhost
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 51200000
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all

#Utilisation de boite aux mails virtuels
#local_transport = virtual
local_recipient_maps = $virtual_mailbox_maps
virtual_mailbox_domains =
↪ mysql:/etc/postfix/sql/mysql-virtual-mailbox-domains.cf
virtual_mailbox_base = /externe/mail/vhosts/
virtual_mailbox_maps = mysql:/etc/postfix/sql/mysql-virtual-mailbox-maps.cf
#virtual_alias_maps = mysql:/etc/postfix/mysql-virtual-alias-maps.cf
virtual_minimum_uid = 100
virtual_uid_maps = static:5000
virtual_gid_maps = static:5000
relayhost =

```

Dans ce fichier :

- le serveur postfix écoute sur IPV4 et IPV6 ;
- la partie SASL AUTHENTICATION renforce la sécurité du serveur. En effet SASL est un protocole qui permet de sécuriser une connexion non sécurisée ;
- la partie TLS renforce la sécurité du serveur. SSL est un protocole qui va aussi permettre de sécuriser le serveur ;
- Toujours pour la sécurité on utilise des certificats autosignés ;
- la propriété `virtual_mailbox_domains = mysql:/etc/postfix/sql/mysql-virtual-mailbox-domains.cf` définit le fichier qui va contenir la directive de connexion à notre base de données mysql plus précisément MariaDB ;
- la partie utilisation de boîte virtuelle va permettre d'utiliser les virtualhosts c'est à dire des comptes mails virtuels qui ne représentent pas les mails des utilisateurs réels de la machine **serveur**. L'utilisateur vhosts est créé par la commande

Console

```
groupadd -g 5000 vhosts
useradd -g vhosts -u 5000 vhosts -d /externe/mail/vhosts -s /bin/false
↵ -m
```

Le fichier `mysql-virtual-mailbox-domains.cf` contient

Console

```
user = messagerieUser
password = isidore
hosts = 127.0.0.1
dbname = messagerie
query = SELECT 1 FROM virtual_domains WHERE name='%s'
```

Le fichier `mysql-mailbox-map` contient

Console

```
user = messagerieUser
password = xxxxxxxx
hosts = 127.0.0.1
dbname = messagerie
query = SELECT maildir FROM virtual_users WHERE email='%s'
```

Le certificat est créé grâce à la célèbre bibliothèque cryptographique openssl.

Console

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
↵ /etc/ssl/private/dovecot-autosigne.key -out
↵ /etc/ssl/certs/dovecot-autosigne.crt
```

Nous modifions le fichier `master.cf`. Il va permettre toujours de renforcer la sécurité et d'obliger le serveur à accepter que des connexions sécurisées

Console

```

# =====
# service type private unpriv chroot wakeup maxproc command + args
#           (yes)    (yes)    (no)    (never) (100)
# =====
smtp      inet  n       -       y       -       -       smtpd
submission inet n       -       y       -       -       smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o content_filter=spamassassin
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
smtps     inet  n       -       y       -       -       smtpd
  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
#628      inet  n       -       y       -       -       qmqpd
pickup    unix  n       -       y       60      1       pickup
cleanup   unix  n       -       y       -       0       cleanup
qmgr       unix  n       -       n       300     1       qmgr
#qmgr      unix  n       -       n       300     1       oqmgr
tlsmgr     unix  -       -       y       1000?   1       tlsmgr
rewrite    unix  -       -       y       -       -       trivial-rewrite
bounce     unix  -       -       y       -       0       bounce
defer      unix  -       -       y       -       0       bounce
trace      unix  -       -       y       -       0       bounce
verify     unix  -       -       y       -       1       verify
flush      unix  n       -       y       1000?   0       flush
proxymap   unix  -       -       n       -       -       proxymap
proxywrite unix  -       -       n       -       1       proxymap
smtp       unix  -       -       y       -       -       smtp
relay      unix  -       -       y       -       -       smtp
  -o syslog_name=postfix/$service_name
#       -o smtp_helo_timeout=5 -o smtp_connect_timeout=5
showq      unix  n       -       y       -       -       showq
error      unix  -       -       y       -       -       error
retry      unix  -       -       y       -       -       error
discard    unix  -       -       y       -       -       discard

```

```

local      unix  -      n      n      -      -      local
virtual    unix  -      n      n      -      -      virtual
lmtp        unix  -      -      y      -      -      lmtp
anvil       unix  -      -      y      -      1      anvil
scache      unix  -      -      y      -      1      scache

spamassassin unix - n n - - pipe
           user=spamd argv=/usr/bin/spamc -f -e
           /usr/sbin/sendmail -oi -f ${sender} ${recipient}

maildrop    unix  -      n      n      -      -      pipe
           flags=DRhu user=vmail argv=/usr/bin/maildrop -d ${recipient}

uucp        unix  -      n      n      -      -      pipe
           flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail
↳ ($recipient)
#
# Other external delivery methods.
#
ifmail      unix  -      n      n      -      -      pipe
           flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)
bsmtp       unix  -      n      n      -      -      pipe
           flags=Fq. user=bsmtp argv=/usr/lib/bsmtp/bsmtp -t$nexthop -f$sender
↳ $recipient
scalemail-backend
↳ unix      -      n      n      -      2      pipe
           flags=R user=scalemail argv=/usr/lib/scalemail/bin/scalemail-store
↳ ${nexthop} ${user} ${extension}
mailman     unix  -      n      n      -      -      pipe
           flags=FR user=list argv=/usr/lib/mailman/bin/postfix-to-mailman.py
           ${nexthop} ${user}

```

Les directives spamassassin permettent d'indiquer à Postfix d'utiliser spamassassin comme filtre antisпам.

3.6 Installation de Dovecot

Dovecot est un serveur IMAP et POP

Console

```
apt-get install dovecot-core dovecot-imapd dovecot-mysql dovecot-lmtpd
↪ dovecot-pop3d
```

Dans allons dans le dossier de configuration de dovecot `/etc/dovecot/` . Nous modifions le fichier `/etc/dovecot/conf.d/10-mail.conf` dans lequel nous allons éditer deux lignes :

Console

```
mail_location = maildir:/externe/mail/vhosts/%d/%n
mail_uid = vhosts
mail_gid = vhosts
mail_privileged_group = mail
first_valid_uid = 5000
last_valid_uid = 5000
```

On modifie le fichier `/etc/dovecot/conf.d/10-auth.conf` et on ajoute

Console

```
disable_plaintext_auth = yes
auth_mechanisms = plain login
!include auth-sql.conf.ext
```

Le fichier `/etc/dovecot/conf.d/auth-sql.conf.ext` :

Console

```
passdb {
    driver = sql
    args = /etc/dovecot/dovecot-sql.conf.ext
}
userdb {
    driver = static
    args = uid=vhosts gid=vhosts home=/externe/mail/vhosts/%d/%n
}
```

Le fichier `/etc/dovecot/dovecot-sql.conf.ext` :

Console

```
driver = mysql
default_pass_scheme = ARGON2I
connect = host=127.0.0.1 dbname=messagerie user=messagerieUser
↪ password=AMETTRE
password_query = SELECT email as user, password FROM virtual_users WHERE
↪ email='%u'
```

Le fichier `10-master.conf`

Console

```
service imap-login {
    inet_listener imap {
        #port = 143
    }
    inet_listener imaps {
        #port = 993
        #ssl = yes
    }
}

service pop3-login {
    inet_listener pop3 {
        #port = 110
    }
    inet_listener pop3s {
        #port = 995
        #ssl = yes
    }
}

service submission-login {
    inet_listener submission {
        #port = 587
    }
}

service lmtp {
    unix_listener /var/spool/postfix/private/dovecot-lmtp {
        mode = 0600
        user = postfix
        group = postfix
    }
}

service imap {
}

service pop3 {
}

service submission {
    # Max. number of SMTP Submission processes (connections)
    #process_limit = 1024
}

service auth {
```



```

    unix_listener auth-userdb {
        mode = 0600
        user = vhosts
        group = vhosts
    }
    # Postfix smtp-auth
    unix_listen er /var/spool/postfix/private/auth {
        mode = 0666
        user = postfix
        group = postfix
    }
}
service auth-worker {
}
service dict {
    unix_listener dict {
        mode = 0600
        user = vhosts
        group = vhosts
    }
}
\end{exempleConsole}
Le fichier /etc/dovecot/10-ssl.conf va contenir les informations pour
↔ sécuriser le serveur avec les certificats :
\begin{exempleconsole}
ssl = yes
ssl_cert = </etc/ssl/certs/dovecot-autosigne.pem
ssl_key = </etc/ssl/private/dovecot-private-autosigne.pem
ssl_dh = </etc/ssl/certs/dovecot-dh-autosigne.pem
ssl_prefer_server_ciphers = yes

```

3.7 Installation du client mail

Console

```

mkdir -p /externe/www/rainloop/public_html /externe/www/rainloop/logs
cd /var/www/rainloop/public_html
curl -sL https://repository.rainloop.net/installer.php | php

```

Le virtualhost associé à rainloop dans Apache a déjà été créé dans la section 3.3.1 à la page 18. Il faut activer ce virtualhost

Console

```
ln -s /etc/nginx/sites-available/www.eneam.da.conf
↪ /etc/nginx/sites-enabled/www.eneam.da.conf
#On active tous les autres virtualhosts
ln -s /etc/nginx/sites-available/www.admin.eneam.da.conf
↪ /etc/nginx/sites-enabled/www.admin.eneam.da.conf
a2ensite www.eneam.da.conf www.eneam.da.conf
systemctl restart nginx apache2 php7.2-fpm
```

On change le propriétaire et les droits d'accès aux répertoires pour permettre aux utilisateurs d'accéder aux sites.

Console

```
chown -R www-data:www-data /erterne/www/html/www.admin.eneam.da/
chown -R www-data:www-data /erterne/www/rainloop/public_html/
chmod -R 660 /erterne/www/html/www.admin.eneam.da/
chmod -R 660 /erterne/www/rainloop/public_html/
```

Nous allons profiter pour configurer notre domaine dans rainloop , pour cela :

- on tape `www.eneam.da/?admin` dans le navigateur Firefox depuis le poste admin ;
- On entre les identifiants par défaut admin et mot de passe 12345 ;
- On change les identifiants ;
- On configure la langue en français ;
- On ajoute les serveurs SMTP et IMAP de notre domaine en cliquant sur le menu domaine.

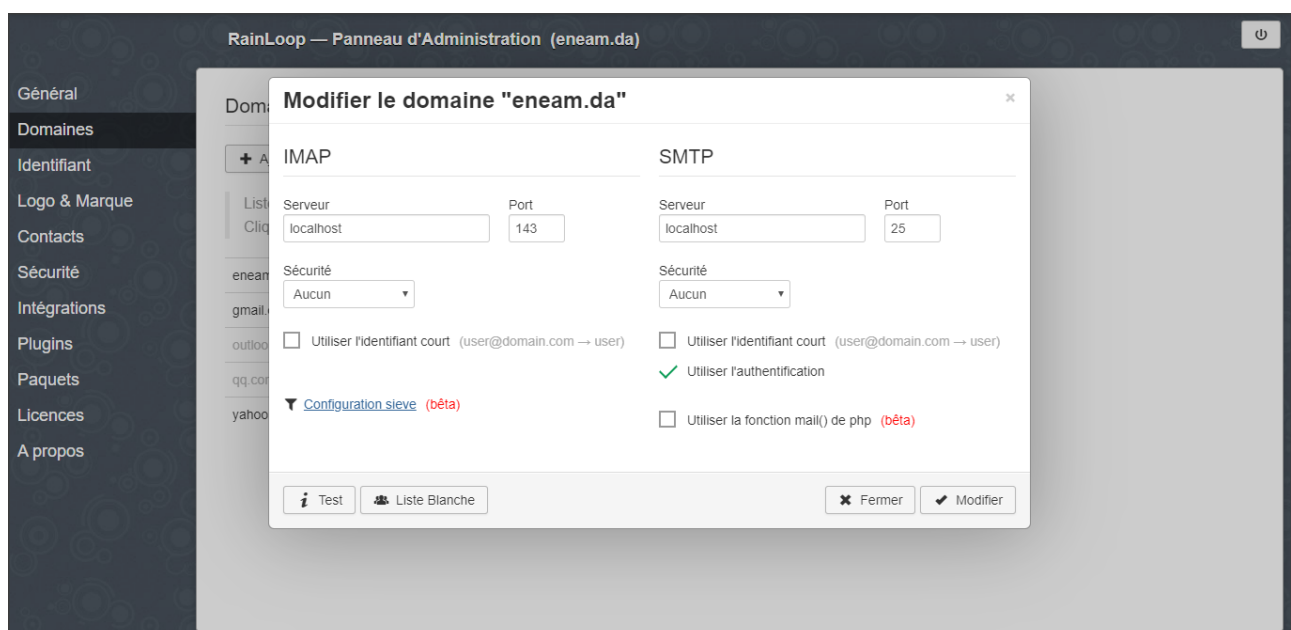


FIGURE 3.5: Ajout des serveurs mails dans rainloop

3.8 Le site d'administration

Le site d'administration va permettre de créer et de supprimer les comptes emails, de voir la liste de quelques services (SMTP, IMAP , APACHE NGINX), leur états, de les arrêter ou de les éteindre. Il existe des solutions gratuites pour l'administration du mail comme *postfixadmin*

Pourquoi n'ai-je pas utilisé une solution existante ?

- Les solutions existantes contiennent beaucoup de fonctions et parfois d'autres que j'ai jugé inutile dans notre contexte. Prenons les alias. On conçoit difficilement qu'il puisse avoir d'alias de comptes (c'est à dire un alias d'un email ou un étudiant possédant plusieurs comptes).
- Le bonheur de coder : on va comprendre les principes de base pour coder un site d'administration de mails et écrire des scripts bash ; ce qui est très intéressant et montre qu'on arrive à combiner toutes les technologies qu'on nous à enseigner lors des cours et dont on dispose pour produire un résultat ;
- Postfixadmin est un outils puissant mais son interface est un peu vieillissant ;
- La flexibilité : étant donné que nous codons nous allons adapter l'application à notre besoin ;
- Il existe des outils puissant, complet mais pas gratuit et qui contiennent des fonctionnalités dont on a pas besoin pour le moment ou pour un début.¹¹

L'administrateur du système va exécuter des instructions depuis l'interface web et pour communiquer avec la machine **serveur** via l'interface web, il va appeler des scripts bash. Nous avons un script bash pour créer le répertoire d'un utilisateur, un autre pour supprimer un répertoire lors de la suppression d'un compte, un autre pour connaître l'état d'un service, un pour arrêter ou redémarrer un service. Voici le contenu du script qui redémarre un service

```

                                Bash
#!/bin/bash
#SYNOPSIS restartOrStopService [restart|stop] [service|all]
#Pour redemarrer les services recoit start ou stop ou restart ou plus les
↪ parametres services
#Ici nous considerons que start est égal à restart
#DETAILS
#      all tous les services
#      start demarrer le service
#      stop arreter le service
#      restart redemarrer le service
#Dans le cas ou on n'a pas envoyé de paramètre on redemarre tous les
↪ service
```

11. Dans le monde professionnel, on préférera des solutions complètes, faciles d'utilisation et complète comme cPanel. Mais ces solutions sont à des prix très onéreux.

```
declare -A service
service[apache2]="apache2"
service[nginx]="nginx"
service[postfix]="postfix"
service[dovecot]="dovecot"
service[phpfpm]="php7.2-fpm"
#service[spamassassin]="spamassassin"
#service[vsftpd]="vsftpd"
serviceValeur="none" # utile pour la fonction
#On initialise retour à 1 c'est à dire echec
retour=1
function restartOrStopService ()
{
    systemctl status $serviceValeur | grep 'active (running)' >
    ↪ /dev/null 2>&1
    if [ $? = 0 ]
    then
        systemctl restart $serviceValeur > /dev/null 2>&1
        if [ $? = 0 ]
        then
            retour=0
        else
            retour=1
        fi
    else
        #On fait des verification plus poussées pour être sur que
        ↪ la commande n'est pas active afin de pouvoir redémarrer
        systemctl is-active $serviceValeur > /dev/null 2>&1
        #Si on est sur que le service est actif alors on le
        ↪ redémarre (restart) sinon on le démarre (start)
        if [ $? = 0 ]
        then
            systemctl restart $serviceValeur > /dev/null 2>&1
            if [ $? = 0 ]
            then
                retour=0
            else
                retour=1
            fi
        fi
    fi
}
```

```

        else
            systemctl start $serviceValeur > /dev/null 2>&1
            if [ $? = 0 ]
            then
                retour=0
            else
                retour=1
            fi
        fi
    fi

fi

}

#Si on a envoyé aucun parametre ou qu'on a envoyé $0 restart all
if [ $# = 0 ] || ( [ $1 = 'restart' ] && [ $2 = 'all' ] )
then
    #Cas de apache2
    # On verifie l'etat du service apache2 ensuite on redemarre
    serviceValeur="apache2"
    restartOrStopService

    #Cas de nginx
    serviceValeur="nginx"
    restartOrStopService

    #Cas de php7.2-fpm
    # On verifie l'etat du service ensuite on redemarre
    serviceValeur="php7.2-fpm"
    restartOrStopService

    #Cas de dovecot
    serviceValeur="postfix"
    restartOrStopService

    #Cas de dovecot
    serviceValeur="dovecot"
    restartOrStopService

    #A la fin on retourne le code qui caracterise l'etat du programme
    ↪ reussite 0 ou echec 1

```

```

        exit $retour

#Si on a fait $0 restart avec $0 le nom du service en question
elif [ $1 = "restart" ]
then
    for key in "${!service[@]}"; do
        if [[ $key = $2 ]]; then
            serviceValeur=$2
            restartOrStopService
            #On quitte la fonction et arrete le programme
            exit $retour
        fi
    done
elif [ $1 = "stop" ]
then
    for key in "${!service[@]}"; do
        # Nous ne pouvons pas arreter le service web ni le service
        ↪ php sinon on ne peut plus y acceder via l'interface web
        if [ $key = $2 ] && [ $2 != "nginx" ] && [ $2 != "apache2"
        ↪ ] && [ $2 != "phpfpm" ]; then
            systemctl status $2 | grep 'active (running)' >
            ↪ /dev/null 2>&1
            if [ $? = 0 ]
            then
                systemctl stop $2 > /dev/null 2>&1
                if [ $? = 0 ]
                then
                    exit 0
                else
                    exit 1
                fi
            else
                #On fait des verification plus poussées
                ↪ pour etre sur que la commande est
                ↪ active afin de pouvoir arreter le
                ↪ service en question
                systemctl is-active $2 > /dev/null 2>&1
                #Si on est sur que le service est actif
                ↪ alors on l'arrete
                if [ $? = 0 ]

```

```

                                then
                                    systemctl stop $2 > /dev/null 2>&1
                                    if [ $? = 0 ]
                                    then
                                        exit 0
                                    else
                                        exit 1
                                    fi
                                else
                                    exit 0
                                fi
                            fi
                        fi
                    fi
                done
            fi

```

Pour permettre l'exécution de script bash depuis le web, il faut autoriser l'utilisateur web **www-data** à lancer des scripts en ayant les droits administrateurs. Pour cela on installe le programme sudo et on ajoute quelques lignes dans le fichier `/etc/sudoers`

```

Console
sudo apt-get install sudo
#Les lignes dans /etc/sudoers
www-data ALL = NOPASSWD:
↳ /externe/www/html/www.admin.eneam.da/public_html/scripts/*
# On admet que tous nos scripts qui ont besoins des droits root sont dans ce
↳ répertoire

```

3.9 Configuration du FTP avec vsftpd

```

Console
sudo apt-get update
sudo apt-get install vsftpd

```

Le fichier de configuration de vsftpd `/etc/vsftpd.conf`

```

Console
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
#listen=NO

```

```
listen=YES

#listen_ipv6=YES

# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
local_enable=YES

# Uncomment this to enable any form of FTP write command.
write_enable=NO
# Pour les utilisateurs anonymes interdiction totales
anon_upload_enable=NO
anon_mkdir_write_enable=NO
anon_other_write_enable=NO

#Activation des utilisateurs virtuels
guest_enable=YES
guest_username=www-data

#On definit les droits par defauts de fichiers uploadés
anon_umask=022
use_localtime=YES

#Maximum session
max_clients=100
max_per_ip=5

#Activation du log
xferlog_enable=YES
log_ftp_protocol=YES

connect_from_port_20=YES

# It is recommended that you define on your system a unique user which the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsecure
ftpd_banner=Par Picasso Houessou

chroot_local_user=YES
allow_writeable_chroot=YES
```



```
#ls_recurse_enable=YES
secure_chroot_dir=/var/run/vsftpd
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/certs/vsftpd-autosigne.crt
#inutile de mettre 1 clé privée les deux sont dans le certificat
rsa_private_key_file=/etc/ssl/private/vsftpd-autosigne.key
ssl_enable=YES
#Permet d'utiliser des configurations individuelles pour chaque utilisateur
user_config_dir=/etc/vsftpd

#Definir la plages de ports utilisée par le mode passif
pasv_min_port=20000
pasv_max_port=20050

#Autoriser les utilisateurs virtuels à changer les permissions de leur
↪ fichiers
chmod_enable=YES
virtual_use_local_privs=YES

# Uncomment this to indicate that vsftpd use a utf8 filesystem.
utf8_filesystem=YES
```

On crée le dossier `/etc/vsftpd/` puis le fichier `/etc/vsftpd/programmeur` qui va contenir les directives pour connecter l'utilisateur virtuel programmeur au serveur FTP :

Console

```
anon_world_readable_only=NO
local_root=/externe/www/html/www.admin.eneam.da/public_html
write_enable=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES
anon_other_write_enable=YES
hide_file=(none)
force_dot_files=YES
```

3.10 Spamassassin

Spamassassin est un antispam. Il va lire dans les logs et vérifier le nombre de tentatives de connexion échoué ou autres paramètres. Si ça atteint ou dépasse un seuil, il bloque les

connexions du client au serveur. Ce qui empêche les spammeurs d'utiliser le serveur pour envoyer du spam¹². Plus les règles sont strictes, plus il va rejeter de mails.

Console

```
apt-get install spamassassin
```

On crée un utilisateur propre à spamassassin

Console

```
sudo adduser spamd --disabled-login
```

Le fichier `/etc/default/spamassassin` sera modifié

Console

```
ENABLED =1
OPTIONS="--create-prefs --max-children 5 --username spamd --helper-home-dir
↪ /home/spamd/ -s /home/spamd/spamd.log"
CRON =1
```

Nous ajoutons les règles dans le fichier `/etc/spamassassin/local.cf`

Console

```
rewrite_header Subject [***** SPAM _SCORE_ *****]
required_score 5.0
use_bayes 1
bayes_auto_learn 1
```

3.11 La base de donnée MariaDB

Voici le script sql complet qui gère notre domaine

SQL

```
-- MySQL dump 10.16  Distrib 10.1.44-MariaDB, for debian-linux-gnu (x86_64)
--
-- Host: localhost    Database: messagerie
-- -----
-- Server version      10.1.43-MariaDB-0ubuntu0.18.04.1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
```

12. Contenu, mail indésirable.

```

/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
↳ FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `admin`
--

DROP TABLE IF EXISTS `admin`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `admin` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `email` varchar(200) COLLATE utf8_unicode_ci NOT NULL,
  `password` varchar(200) COLLATE utf8_unicode_ci NOT NULL,
  `nom` varchar(200) COLLATE utf8_unicode_ci NOT NULL,
  `prenom` varchar(200) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `emailUnique` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8
↳ COLLATE=utf8_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `admin`
--

LOCK TABLES `admin` WRITE;
/*!40000 ALTER TABLE `admin` DISABLE KEYS */;
INSERT INTO `admin` VALUES
↳ (1,'admin@eneam.da','$argon2i$v=19$m=1024,t=2,p=2$RFNLaHNyS1ROUmlqTOYurQ',
↳ Q$YyWyEK06b6SPLN0nxo4xuBpZgT2pnLGhtN4Cm+vp4f0','Houessou','Picasso'),(2
↳ ,'master@eneam.da','$argon2i$v=19$m=1024,t=2,p=2$RFNLaHNyS1ROUmlqTOYurQ',
↳ $YyWyEK06b6SPLN0nxo4xuBpZgT2pnLGhtN4Cm+vp4f0','master','master');
/*!40000 ALTER TABLE `admin` ENABLE KEYS */;
UNLOCK TABLES;

```

```
--
-- Table structure for table `date`
--

DROP TABLE IF EXISTS `date`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `date` (
  `date` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `date`
--

LOCK TABLES `date` WRITE;
/*!40000 ALTER TABLE `date` DISABLE KEYS */;
INSERT INTO `date` VALUES ('2018-07-22');
/*!40000 ALTER TABLE `date` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `virtual_domains`
--

DROP TABLE IF EXISTS `virtual_domains`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `virtual_domains` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(200) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8
  ↳ COLLATE=utf8_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `virtual_domains`
--
```

```

LOCK TABLES `virtual_domains` WRITE;
/*!40000 ALTER TABLE `virtual_domains` DISABLE KEYS */;
INSERT INTO `virtual_domains` VALUES (1,'eneam.da');
/*!40000 ALTER TABLE `virtual_domains` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `virtual_users`
--

DROP TABLE IF EXISTS `virtual_users`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `virtual_users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `domain_id` int(11) NOT NULL,
  `password` varchar(200) COLLATE utf8_unicode_ci NOT NULL,
  `email` varchar(200) COLLATE utf8_unicode_ci NOT NULL,
  `maildir` varchar(200) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `email` (`email`),
  KEY `domain_id` (`domain_id`),
  CONSTRAINT `virtual_users_ibfk_1` FOREIGN KEY (`domain_id`) REFERENCES
    ↪ `virtual_domains` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8
  ↪ COLLATE=utf8_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `virtual_users`
--

LOCK TABLES `virtual_users` WRITE;
/*!40000 ALTER TABLE `virtual_users` DISABLE KEYS */;
INSERT INTO `virtual_users` VALUES (10,1,'$argon2i$v=19$m=65536,t=4,p=1$cVkJ
  ↪ 1TFVRY2JnTjlaLmNRWQ$tWZ+MIaPTRPGiJlct3dFoEuypXLzSUEo4MBYS2nebYM','faceb
  ↪ ook@eneam.da','eneam.da/facebook/'),(11,1,'$argon2i$v=19$m=65536,t=4,p=
  ↪ 1$d01VMm44NjFWZlo2NUZLZQ$g5CjjLk51liBwJz9TG6v1ZTy6v/lZgnlKvrzIacaQ58','
  ↪ houessoupicasso1@eneam.da','eneam.da/houessoupicasso1/'),(12,1,'$argon2
  ↪ i$v=19$m=65536,t=4,p=1$bWF6T1l4V1hQZDJheC9wdw$hX2Ndb+/nu9jTawGrECLH0zLG
  ↪ FpH4/1fGrPqGw0unYk','fcxerwrexcr@eneam.da','eneam.da/fcxerwrexcr/'),(13
  ↪ ,1,'$argon2i$v=19$m=65536,t=4,p=1$by9wZ25rUWkyeGhrVERuSw$du7tQ5d8JQHhZSq

```

```

/*!40000 ALTER TABLE `virtual_users` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `virtual_users_infos`
--

DROP TABLE IF EXISTS `virtual_users_infos`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `virtual_users_infos` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `virtual_user_id` int(11) NOT NULL,
  `nom` varchar(200) COLLATE utf8_unicode_ci DEFAULT NULL,
  `prenom` varchar(200) COLLATE utf8_unicode_ci DEFAULT NULL,
  `matricule` int(11) DEFAULT NULL,
  `telephone` int(11) DEFAULT NULL,
  `pays` varchar(200) COLLATE utf8_unicode_ci DEFAULT NULL,
  `date_fin` date DEFAULT NULL,
  `delete_token` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `cle_etrangere` (`virtual_user_id`)
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=utf8
  ↳ COLLATE=utf8_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `virtual_users_infos`
--

LOCK TABLES `virtual_users_infos` WRITE;
/*!40000 ALTER TABLE `virtual_users_infos` DISABLE KEYS */;
INSERT INTO `virtual_users_infos` VALUES (10,10,NULL,NULL,NULL,NULL,'Bénin'
↳ ,NULL,NULL),(11,11,NULL,NULL,NULL,NULL,'Bénin',NULL,NULL),(12,12,NULL,N
↳ ULL,NULL,NULL,'Bénin',NULL,NULL),(13,13,NULL,NULL,NULL,NULL,'Bénin',NUL
↳ L,NULL),(14,14,'uytrtxt','yvutrt',NULL,NULL,'Bénin',NULL,NULL),(15,15,N
↳ ULL,NULL,NULL,NULL,'Bénin',NULL,NULL),(16,16,NULL,NULL,NULL,NULL,'Bénin
↳ ','NULL,NULL),(18,18,'Bake','Bake',NULL,NULL,'Bénin',NULL,NULL),(19,19,'
↳ Toto','Toto',NULL,NULL,'Bénin',NULL,NULL);
/*!40000 ALTER TABLE `virtual_users_infos` ENABLE KEYS */;

```

```

UNLOCK TABLES;
/#!/40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/#!/40101 SET SQL_MODE=@OLD_SQL_MODE */;
/#!/40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/#!/40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/#!/40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/#!/40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/#!/40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/#!/40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2020-03-20 11:23:32

```

3.12 Sécurité

Nous allons écrire des règles iptables

Console

```

iptables --policy FORWARD DROP
iptables --policy INPUT DROP
iptables --policy OUTPUT DROP
#FTP
iptables --append INPUT --protocol tcp --dport 21 -m conntrack --ctstate
↪ NEW,ESTABLISHED -j ACCEPT
#SSH
iptables --append INPUT --protocol tcp --dport 22 -m conntrack --ctstate
↪ NEW,ESTABLISHED -j ACCEPT
#SMTP et SMTPS SMTP sur STARTLS
iptables --append INPUT --protocol tcp --dport 25 -m conntrack --ctstate
↪ NEW,ESTABLISHED -j ACCEPT
iptables --append INPUT --protocol tcp --dport 465 -m conntrack --ctstate
↪ NEW,ESTABLISHED -j ACCEPT
iptables --append INPUT --protocol tcp --dport 587 -m conntrack --ctstate
↪ NEW,ESTABLISHED -j ACCEPT
#HTTP et HTTPS
iptables --append INPUT --protocol tcp --dport 80 -m conntrack --ctstate
↪ NEW,ESTABLISHED -j ACCEPT
iptables --append INPUT --protocol tcp --dport 443 -m conntrack --ctstate
↪ NEW,ESTABLISHED -j ACCEPT

```

```
iptables --append INPUT --protocol tcp --dport 7080 -m conntrack --ctstate
↪ NEW,ESTABLISHED -j ACCEPT
iptables --append INPUT --protocol tcp --dport 7443 -m conntrack --ctstate
↪ NEW,ESTABLISHED -j ACCEPT
#IMAP
iptables --append INPUT --protocol tcp --dport 143 -m conntrack --ctstate
↪ NEW,ESTABLISHED -j ACCEPT
iptables --append INPUT --protocol tcp --dport 993 -m conntrack --ctstate
↪ NEW,ESTABLISHED -j ACCEPT
#MYSQL
iptables --append INPUT --protocol tcp --dport 3306 -m conntrack --ctstate
↪ NEW,ESTABLISHED -j ACCEPT
```

3.13 Cas concret

Nous allons illustrer l'utilisation du système par un exemple pratique. Voici l'énoncé.

3.13.1 Enoncé

Baké et Toto sont deux nouveaux étudiants de ENEAM. L'administrateur va créer leur comptes emails respectifs. Pour cela, il se connecte à la plateforme `www.admin.eneam.da`. Une fois les comptes créés, Baké va se connecter par le webmail et va ensuite envoyer un message à Toto. Toto va lui aussi se connecter et répondre au mail reçu. L'administrateur va envoyer aussi un mail de convocation à Toto qui est le responsable de la IG1. Ensuite il va supprimer le compte de Béréké. De même, il va consulter la liste des services et arrêter temporairement le service mail pour raison de maintenance.

3.13.2 Pratique

- Création du compte mail de Baké : On allume le poste admin. On ouvre le navigateur et on saisit l'adresse `www.admin.eneam.da` ou `admin.eneam.da`.
- L'administrateur renseigne ses informations de connexion : adresse mail ***admin.eneam.da*** mot de passe ***amettre***.

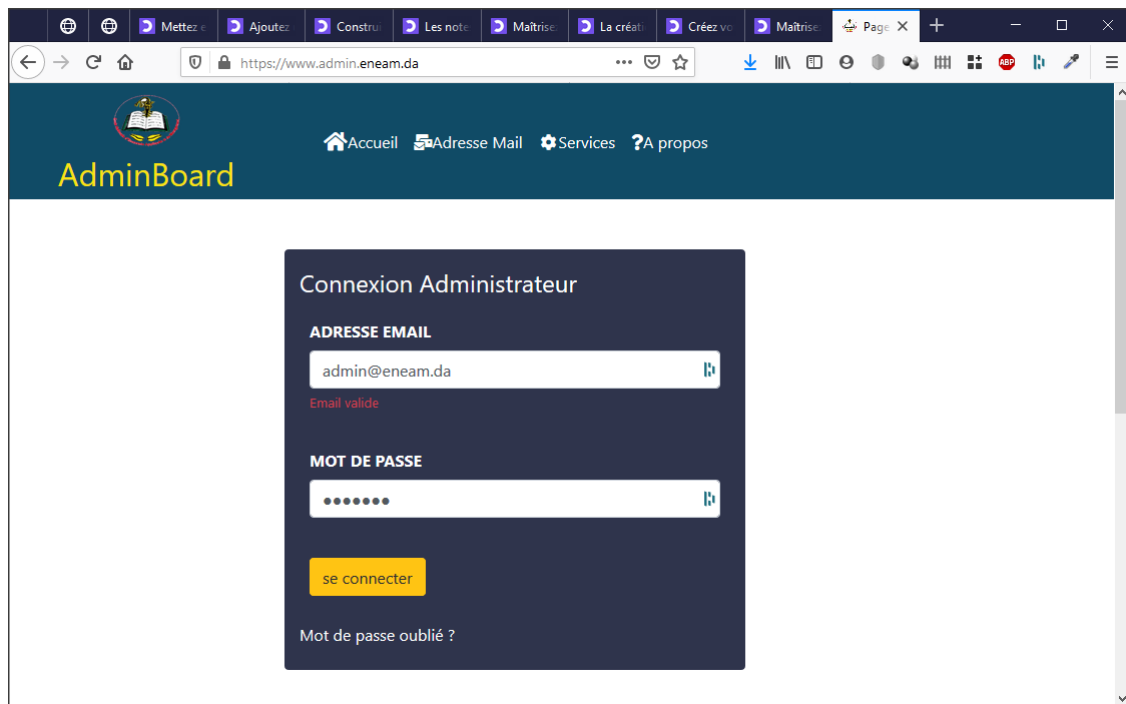


FIGURE 3.6: Connexion de l'administrateur au site d'administration

- Sur la page d'accueil il renseigne le mail qu'il veut créer. Ici nous allons mettre simplement `bake@eneam.da`. Puis on renseigne le mot de passe qui doit avoir une forte entropie¹³. On suppose ici `Ba21@kesccT`. On confirme le mot de passe.
- Facultatif : On coche la case Afficher les informations facultatives. Ce qui permet de renseigner le nom et prénom de l'étudiant, son numéro matricule, numéro de téléphone, la date d'expiration du compte.¹⁴

13. il est obligatoire d'avoir au moins 8 caractères, un caractère spécial, une minuscule et une majuscule

14. Les comptes étant essentiellement pour des étudiants on considère que le compte est valide durant la période d'étude. On utilisera un cron pour désactiver automatiquement les comptes expirés.

Bienvenue Monsieur/Madame **Picasso Houessou** sur la plateforme d'administration. On est aujourd'hui le **16-03-2020**
Si vous rencontrez des problèmes ou constatez des bugs, veuillez bien me contacter par mail [Picasso Houessou](#)

Création rapide de compte email

bake@eneam.da

.....

Doit contenir au moins 8 caractères, une lettre majuscule, un chiffre et un caractère spécial !@&#%\$%^&*.

.....

Veuillez confirmer le mot de passe

☒ Afficher les options facultatives

Nom: Bake Prénoms: Bake

Matricule: Matricule ex: 112222 Numéro de téléphone: Numero de telephone Date d'expiration: jj / mm / aaaa Pays: Bénin

Ne peut dépasser 2025-03-15

Créer le compte

Attention Information importante
Le nouveau compte a été bien créé

AdminBoard Accueil Adresse Mail Services ? A propos

Bienvenue Monsieur/Madame **Picasso Houessou** sur la plateforme d'administration. On est aujourd'hui le **16-03-2020**
Si vous rencontrez des problèmes ou constatez des bugs, veuillez bien me contacter par mail [Picasso Houessou](#)

Création rapide de compte email

exemple@eneam.da

.....

Doit contenir au moins 8 caractères, une lettre majuscule, un chiffre et un caractère spécial !@&#%\$%^&*.

.....

Veuillez confirmer le mot de passe

☐ Afficher les options facultatives

Créer le compte

FIGURE 3.7: Création du compte bake@eneam.da

- On appuie sur le bouton *Créer le compte*
- Le système renvoie une information pour notifier que le compte a été créé ou s'il a eu une erreur (par exemple si le compte existe déjà).
- Il reprend la même opération pour Toto avec pour adresse mail toto@eneam.da et mot de passe to21@kesccT .
- Baké tape www.eneam.da pour accéder au client webmail. Il saisit ses informations de connexion et se connecte.

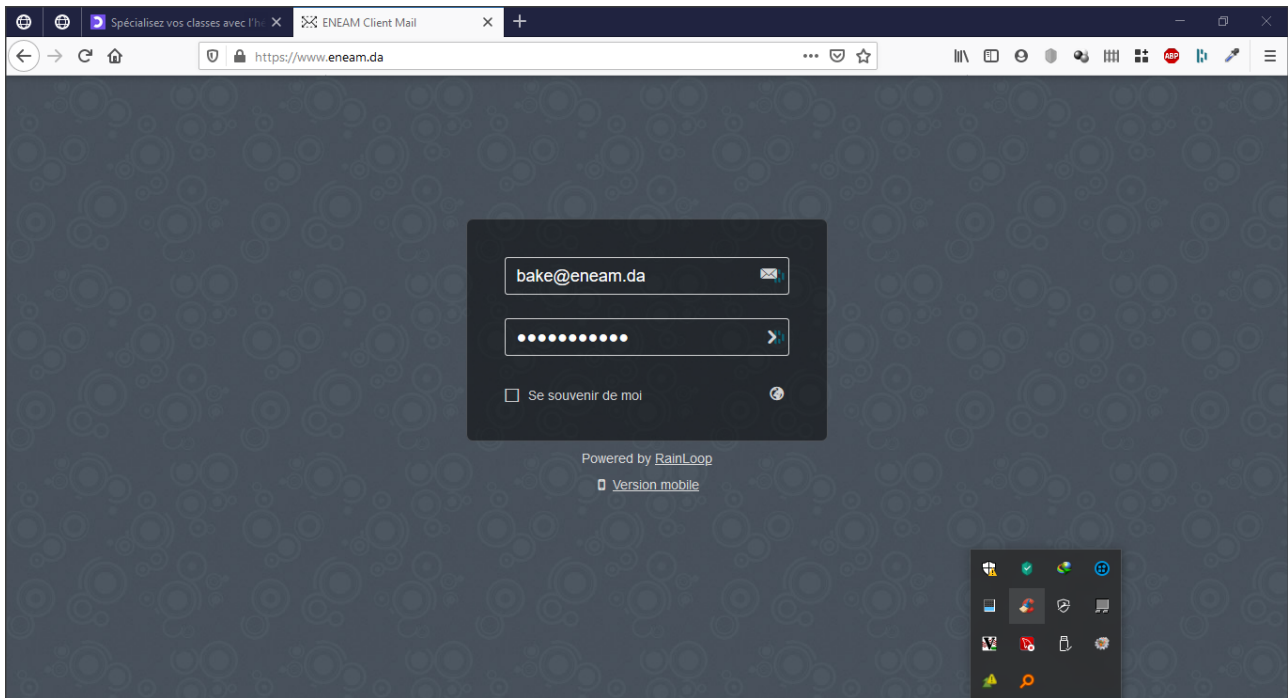


FIGURE 3.8: Connexion de Baké au client webmail

— Il crée un nouveau message à destination de Toto et l'envoie

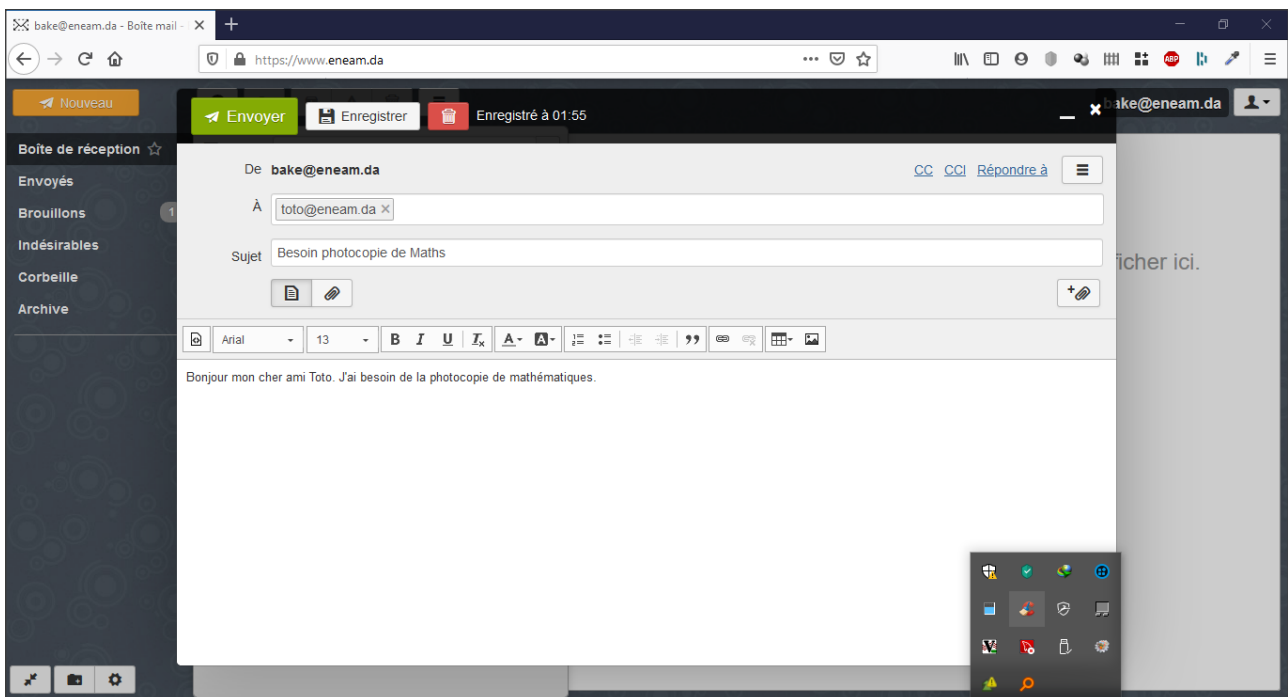


FIGURE 3.9: Envoi d'un mail de Baké à Toto

— Toto se connecte voit le message et répond.

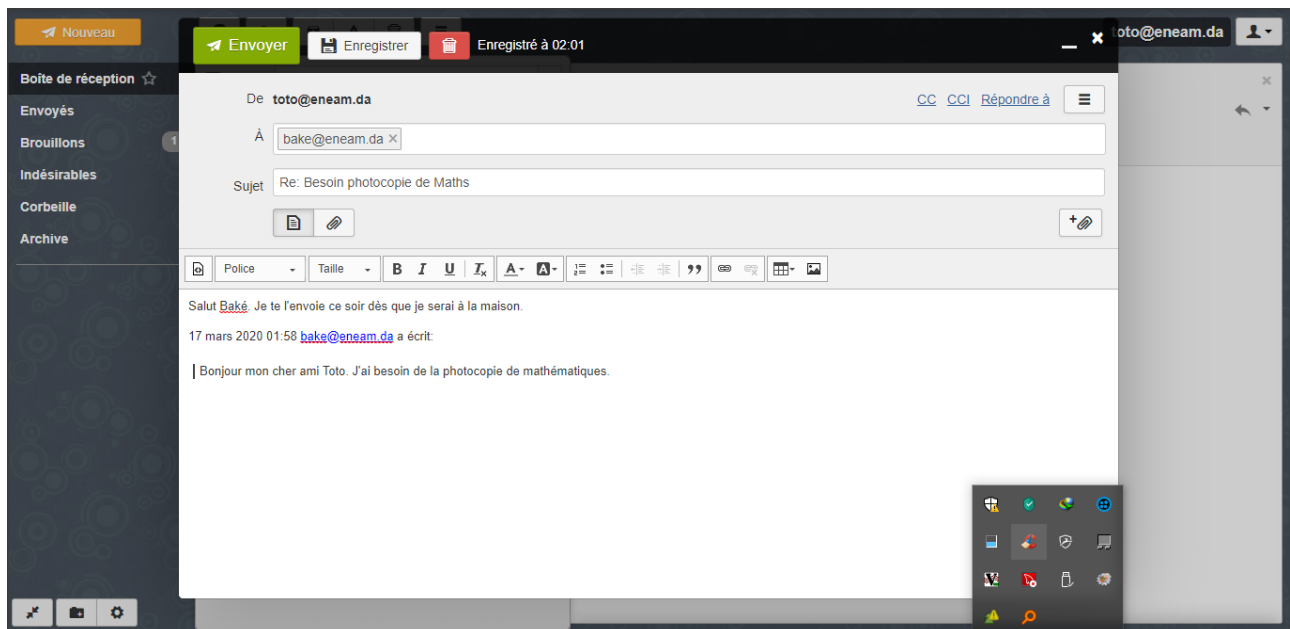
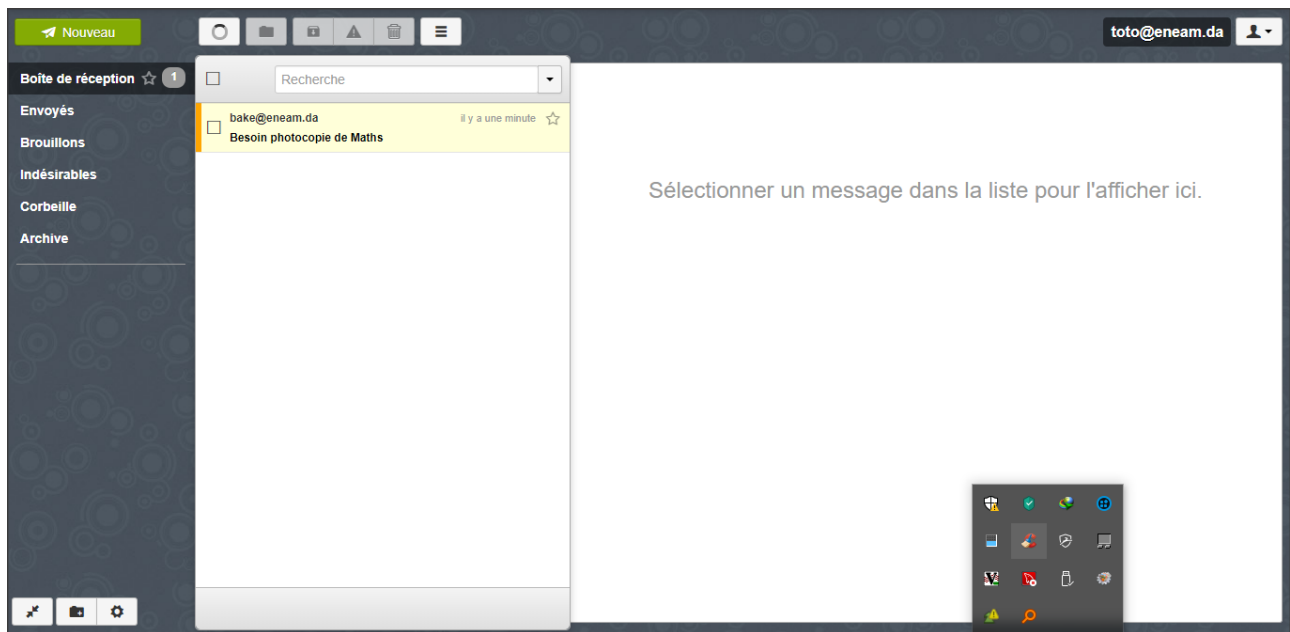


FIGURE 3.10: Réponse de Toto au mail de Baké

- L'administrateur se connecte aussi et envoie un message de convocation à Toto
- Toto se connecte voit le message de Baké et répond.

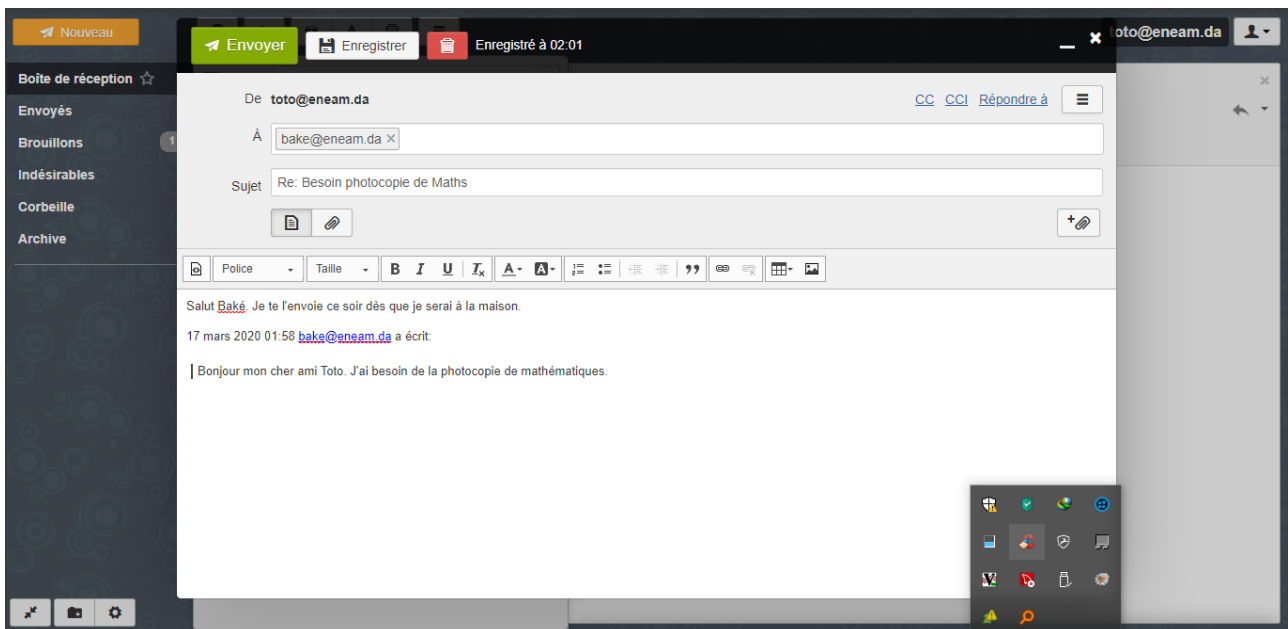
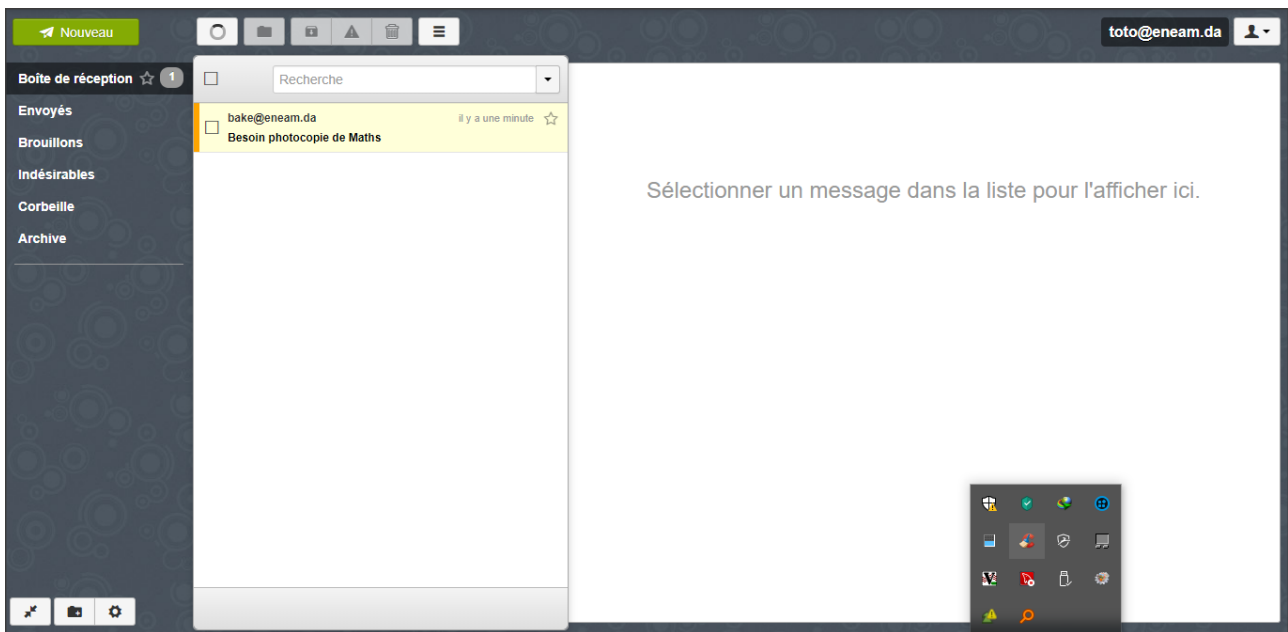


FIGURE 3.11: Lecture du mail reçu de Baké par Toto

- L'administrateur supprime le compte de Béréké : Pour cela il clique sur le menu Adresse mail. Ensuite il recherche le compte de Toto et clique sur le bouton représenté par un bonhomme avec une croix. Une boîte de dialogue apparaît et demande de confirmer la suppression. Il clique sur oui supprimer. Des toasts apparaissent pour notifier si le compte est supprimé. Il a la possibilité de les effacer ou de les enregistrer en un format texte au cas où il voudrait écrire des notes plus tard.

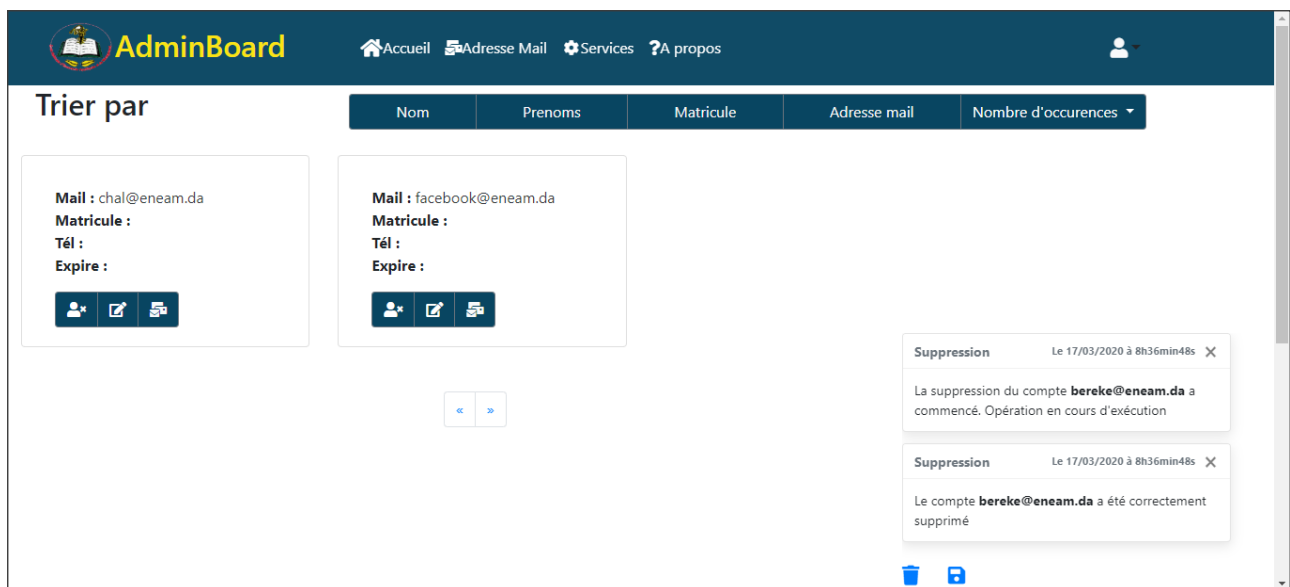
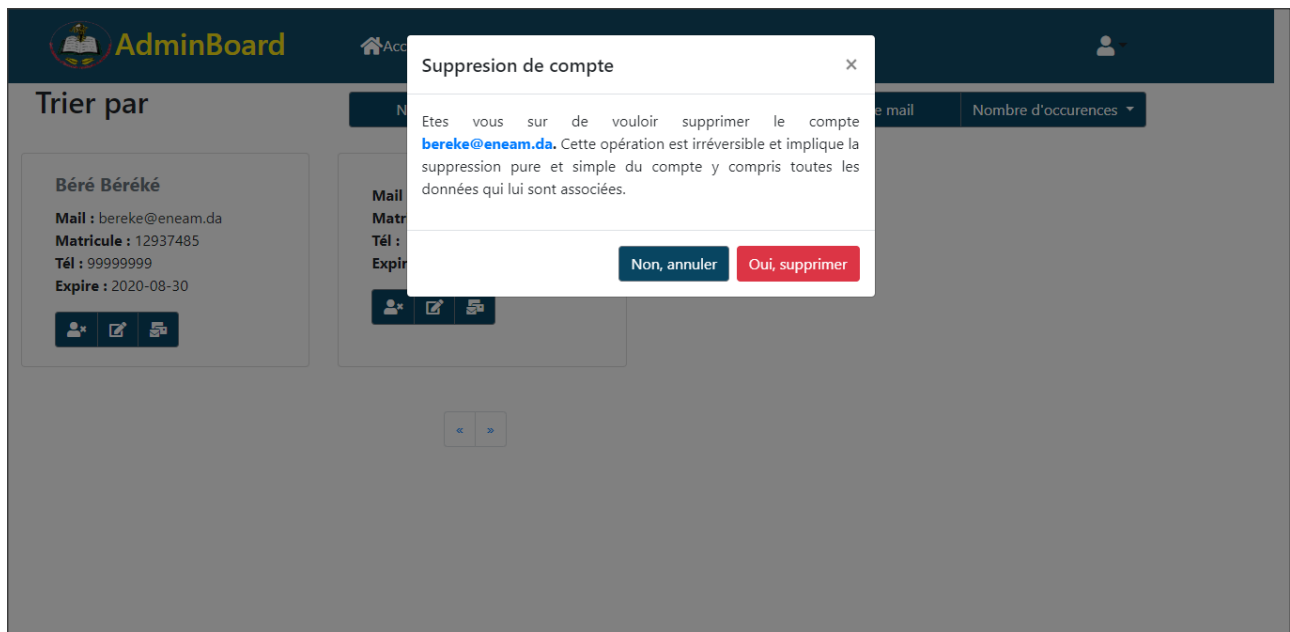
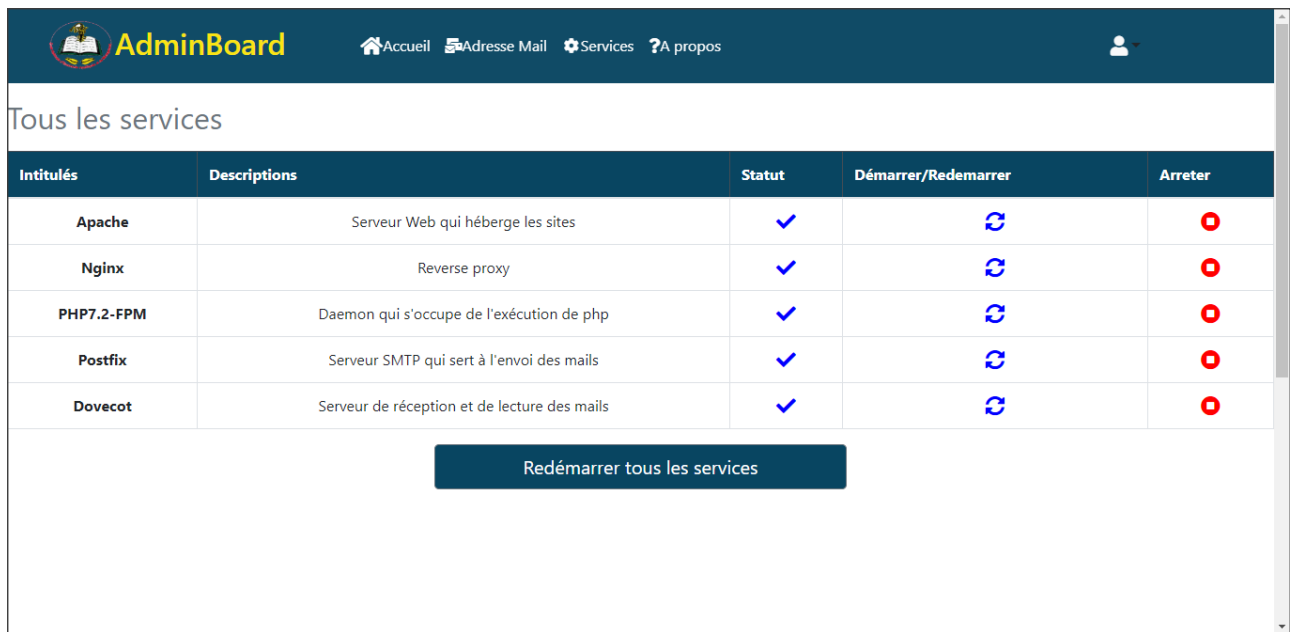


FIGURE 3.12: Suppression du compte de Béréké

- L'administrateur clique sur le menu Services. Il observe sur cette page 5 services. Le service Apache, Nginx, PHP7.2-FPM, Postfix, Dovecot. Il peut choisir de redémarrer un service en cliquant sur l'icône redémarrer dans le champ correspondant ou redémarrer tous les services à la fois en cliquant sur le bouton redémarrer tous les services. Il peut de même arrêter un service au besoin. Il est impossible d'arrêter les services web (Apache, Nginx, PHP7.2-FPM). En effet, il contrôle le serveur par l'interface web. S'il arrête donc les services web, il serait impossible de manipuler le serveur depuis le navigateur et il sera bloqué. C'est d'ailleurs la raison pour laquelle l'icône arrêter est désactivé pour ces trois services.

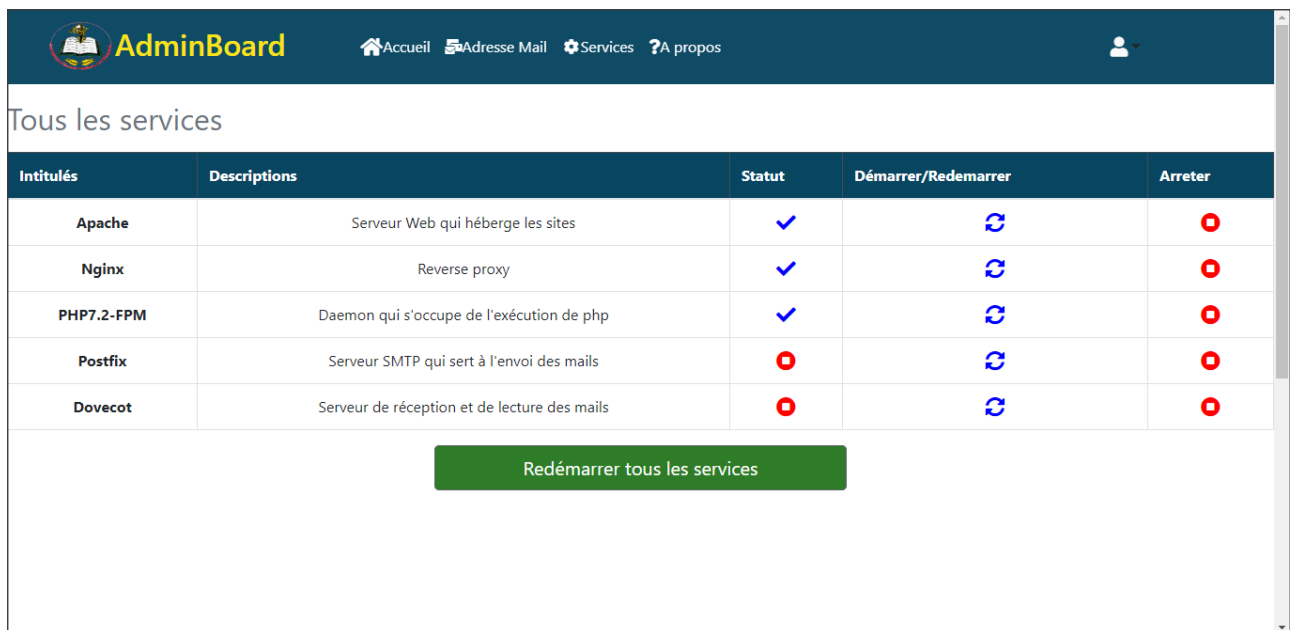


Intitulés	Descriptions	Statut	Démarrer/Redémarrer	Arrêter
Apache	Serveur Web qui héberge les sites	✓		
Nginx	Reverse proxy	✓		
PHP7.2-FPM	Daemon qui s'occupe de l'exécution de php	✓		
Postfix	Serveur SMTP qui sert à l'envoi des mails	✓		
Dovecot	Serveur de réception et de lecture des mails	✓		

Redémarrer tous les services

FIGURE 3.13: Vérification de l'état des services

— L'administrateur arrête les services mails(Postfix et dovecot)



Intitulés	Descriptions	Statut	Démarrer/Redémarrer	Arrêter
Apache	Serveur Web qui héberge les sites	✓		
Nginx	Reverse proxy	✓		
PHP7.2-FPM	Daemon qui s'occupe de l'exécution de php	✓		
Postfix	Serveur SMTP qui sert à l'envoi des mails			
Dovecot	Serveur de réception et de lecture des mails			

Redémarrer tous les services

FIGURE 3.14: Arrêt des services Postfix et Dovecot

— Il se déconnecte en cliquant sur l'icone située à l'extrême droite de l'écran et en appuyant sur se déconnecter. Pour des raisons de sécurité , il est aussi déconnecté automatiquement après une durée d'inactivité de 15 minutes.

Conclusion

Mon stage académique effectué au sein de JScom s'est révélé être une expérience marquante et m'a montré un aperçu des réalités quotidiennes en milieu professionnel. J'ai mis en place un système d'envoi de mails pour faciliter les échanges au sein de l'ENEAM. Ce qui m'a permis d'explorer le vaste monde de l'administration système sous Linux. J'ai donc pu manipuler et découvrir plusieurs services réseaux.

Bibliographie

- [1] Install and configure postfix and dovecot, Janvier 2019. www.linuxize.com.
- [2] Maurice Chavelli. Prenez en main bootstrap, Septembre 2019.
- [3] Dovecot. Dovecot manual. <http://doc.dovecot.org>.
- [4] Enguerran Gillier. Sécurisez vos données avec la cryptographie, Août 2019.
- [5] Chantal Gribaumont. Administrez vos bases de données avec mysql, Septembre 2019.
- [6] Karnaj and TorxicScorpui. Introduction à latex, Janvier 2020.
- [7] Eric Latitte. Apprenez le fonctionnement des réseaux tcp/ip, Avril 2019.
- [8] Eric Latitte. Maîtrisez vos applications et réseaux tcp/ip, Juin 2019.
- [9] Etienne Lavanant. Gérez votre serveur linux et ses services, Novembre 2018.
- [10] Etienne Lavanant. Montez un serveur de fichiers sous linux, Novembre 2019.
- [11] Noël-Arnaud Maguis. Rédigez des documents de qualité avec latex, Juin 2019.
- [12] Michel Martin. Simplifiez vos développements javascript avec jquery, Novembre 2017.
- [13] Lélío Motta. Simulez des architectures réseaux avec gns3, Juin 2019.
- [14] Mathieu Nebra. Concevez votre site web avec php et mysql, Mai 2019.
- [15] Mathieu Nebra. Reprenez le contrôle à l'aide de linux!, Juin 2019.
- [16] Thedownloader. Un serveur d'hébergement multiutilisateur sous linux, Octobre 2017.