

# ÉCOLE NATIONALE D'ÉCONOMIE APPLIQUÉE ET DE MANAGEMENT



---

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

---

Par :

Picasso T. I. HOUESSOU-DOSSOU

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 leurs 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
<b>1 Cadre de l'étude</b>	<b>2</b>
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
<b>2 Cadre théorique et méthodologique</b>	<b>4</b>
2.1 Enoncé du problème . . . . .	4
2.2 Objectif . . . . .	4
2.3 Hypothèse . . . . .	5
2.4 Etude théorique . . . . .	5
2.4.1 Définition d'un serveur . . . . .	5
2.4.2 Disque dur RAID LVM . . . . .	6
2.4.3 Serveur web APACHE . . . . .	7
2.4.4 Base de données . . . . .	7
2.4.5 Modélisation avec GNS3 . . . . .	7
2.4.6 Programmation . . . . .	8
2.4.7 FTP . . . . .	8
2.4.8 Fonctionnement du mail . . . . .	8
2.4.9 Sécurité . . . . .	12
<b>3 Implémentation du projet</b>	<b>14</b>
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 . . . . .	18
3.3.1	Gestion de Apache . . . . .	18
3.3.2	Gestion de Nginx . . . . .	21
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 . . . . .	32
3.7	Installation du client mail . . . . .	35
3.8	Le site d'administration . . . . .	36
3.9	Configuration du FTP avec vsftpd . . . . .	41
3.10	Spamassassin . . . . .	43
3.11	La base de données MariaDB . . . . .	44
3.12	Sécurité . . . . .	49
3.13	Cas concret . . . . .	50
3.13.1	Enoncé . . . . .	50
3.13.2	Pratique . . . . .	50
<b>Conclusion</b>		<b>58</b>
<b>Bibliographie</b>		<b>59</b>

# 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 . . . . .	10
2.3	Résumé de l'envoi et de la réception d'un mail . . . . .	12
3.1	Résumé du fonctionnement web du réseau <b>projetmail</b> . . . . .	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 . . . . .	36
3.6	Connexion de l'administrateur au site d'administration . . . . .	51
3.7	Création du compte bake@eneam.da . . . . .	52
3.8	Connexion de Baké au client webmail . . . . .	53
3.9	Envoi d'un mail de Baké à Toto . . . . .	53
3.10	Réponse de Toto au mail de Baké . . . . .	54
3.11	Lecture du mail reçu de Baké par Toto . . . . .	55
3.12	Suppression du compte de Béréké . . . . .	56
3.13	Vérification de l'état des services . . . . .	57
3.14	Arrêt des services Postfix et Dovecot . . . . .	57

# Introduction

L'école est un lieu d'enseignement de connaissances générales ou de connaissances particulières nécessaires à l'exercice d'un métier, d'une profession. C'est également l'ensemble des élèves et professeurs qui fréquentent cet établissement. L'apprentissage est donné par la communication. Une bonne communication et une rigueur des apprenants sont primordiales à leur réussite scolaire. La communication désigne l'ensemble des moyens qui permet pour à sujet, de transmettre une information à un autre sujet : communication verbale, écrite, par les affiches, par les signes. Des difficultés de communication sont préjudiciables au bon déroulement des activités de l'école.

En particulier, en tant qu'étudiant à l'ENEAM nous avons rencontré des obstacles dans le processus de communication de la programmation des cours. En effet, une bonne programmation des cours nécessite souvent de la part de l'administration et des professeurs de faire face à des contraintes de dernières minutes qui obligent un réajustement de l'horaire des cours et d'informer les étudiants des nouveaux horaires. Toutefois, certains étudiants par inadvertance, d'autres par manque de communication entre eux, ne sont pas informés de ces modifications. Il est donc impératif que nous avons besoin d'un moyen de communication sûr, efficace, centralisé, sécurisé pour discuter et pour s'échanger les informations qui circulent au sein de l'école. Ce moyen va servir non seulement aux étudiants mais va aussi constituer un outil essentiel que l'administration et les professeurs peuvent utiliser pour les échanges en milieu scolaire. Quel facteur peut améliorer la communication au sein de 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 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.

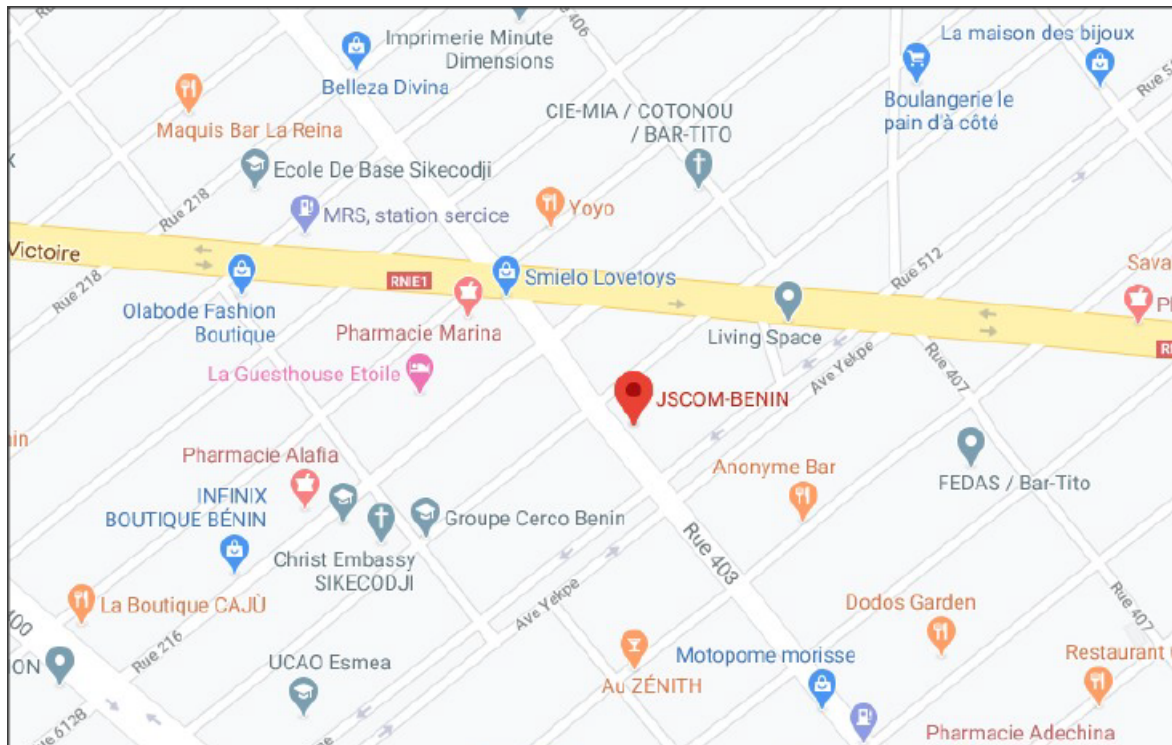


FIGURE 1.1 – Situation géographique de JScom, source Google maps.

## 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, nous allons présenter le thème et toutes les notions fondamentales qui ont rapport au thème.

### 2.1 Enoncé du problème

La communication est importante dans toute société car elle met en relation les hommes. Ainsi, la communication est au cœur de toutes les activités menées à l'école. A l'école, la communication regroupe toutes informations utiles que l'administration peut mettre à disposition des étudiants, que les professeurs peuvent échanger avec leurs apprenants, de même que les apprenants peuvent s'échanger entre eux. Une bonne communication en milieu scolaire est donc essentiels à la réussite des apprenants. Cependant, durant nos années études à ENEAM, nous avons rencontré plusieurs obstacles dans la communication :

- Le manque de communication entre étudiant des changements dans les horaires des cours ;
- Le manque d'échanges entre étudiants sur les notions reçues ;
- La mauvaise communication entre étudiants hors du cadre scolaire ;
- L'accès à certaines informations ;
- La non-conformité ou la divergence des informations relayées par les étudiants ;
- La véracité de certaines informations en général diffusées par les étudiants.

Nous avons donc éprouvé le besoin de rechercher les facteurs d'amélioration de la qualité de la communication à l'école de façon spécifique à l'ENEAM. D'où la question fondamentale de recherche :

Quel facteur peut contribuer à améliorer la communication entre étudiants au sein de l'ENEAM ?

### 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 définit des règles, des conventions qui permettent aux ordinateurs (de façon générale, tout équipement électronique qui possède une carte réseau) de communiquer. Un serveur peut être également considéré comme un logiciel qui fournit un service à d'autres logiciels.

#### Les caractéristiques d'un serveur

Un serveur a 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 peut avoir ;
- Un serveur Linux n'a généralement pas d'interface graphique. Comme un serveur n'est pas souvent connecté à un écran, on préfère ne pas installer un environnement graphique ;
- Un serveur utilise très souvent un système d'exploitation spécialisé.

#### Quelques services

Les serveurs assurent différents services. Parmi lesquels, nous pouvons citer :

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

#### Spécification d'un OS serveur

Un système d'exploitation de serveur (OS Operating System) est 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 des fonctions de veilles. En effet les serveurs restent allumés tout le temps ;
- Les OS serveurs n'ont pas souvent d'interface graphique pour les systèmes Unix et Linux. En effet, ils peuvent avoir un environnement graphique mais on préfère ne pas l'installer puisqu'on y connecte pas de matériel graphique (écran) ;
- Ils peuvent gérer des ressources physiques énormes par rapport à un ordinateur personnel. Une machine serveur peut par exemple être dotée de plus 200 gigas (200 Go) 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 de 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, le disque endommagé peut être remplacé sans risque de perte de données. 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 nouveau disque C pour ne pas risquer de perdre les données définitivement puisque le seul disque restant pourrait s'endommager ;
- Le raid 10 : consiste à assembler au moins deux unités RAID 1 en un ensemble de RAID 0. Il faut donc un minimum de quatre disques et toujours en nombre pair. Il offre la sécurité puisque les données sont répliquées sur plusieurs grappes et améliorent la performance car les données sont écrites sur plusieurs disques.
- Le raid 5 : utilise au moins trois disques et répartit les données sur tous les disques en rajoutant des blocs de parité. Une donnée perdue peut être récupérée grâce aux blocs de parité. Si plus d'un disque s'endommage, on perd toutes les données.

LVM (Logical Volume Management) permet la création et la gestion des volumes logiques sans perte de données. C'est un système qui permet par exemple de diminuer la taille d'un système de fichier pour pouvoir en agrandir un autre, sans se préoccuper de leur emplacement sur le disque. Les opérations de redimensionnement deviennent quasiment sans risque, contrairement au redimensionnement des partitions. Si un des volumes physiques est endommagé, alors l'ensemble des volumes logiques qui utilise ce volume physique est perdu. Pour éviter ce problème, il faut utiliser LVM sur des disques RAID par exemple.

### 2.4.3 Serveur web APACHE

Un serveur web est le service qui permet d'accéder à des pages web. Notre serveur de mail sera accessible par une interface web. Nous allons installer le serveur web Apache. Apache ou plus précisément HTTPD est le serveur web le plus populaire. Il est développé et maintenu par la fondation *Apache*. Le serveur web Nginx sera utilisé comme reverse proxy<sup>1</sup>.

### 2.4.4 Base de données

Les données des utilisateurs doivent être stockées dans une base de données. Une base de données 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 Données) est un logiciel qui manipule une base de données. Une base de données 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 relationnels : les données sont stockées dans des tables et sont liées entre elle par des relations. Le langage SQL est utilisé pour interagir avec les données. 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. Elles permettent la réplication des serveurs et de distribuer les données. On peut citer à titre d'exemple Poids, MongoDB, Cassandra, ElasticSearch.

### 2.4.5 Modélisation avec GNS3

La mise en place du projet nécessite de faire de la modélisation, c'est à dire représenter l'architecture réseau du projet 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.

---

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.

### 2.4.6 Programmation

Les langages de programmation sont des instructions écrites dans un langage accessible à l'homme et qui sont ensuite soit lues, soit compilées, soit interprétées par des programmes spécifiques pour permettre à l'ordinateur 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. Il s'agit d'un ensemble de technologies qui permettent de créer des pages web dynamiques.

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

**CSS** est un langage de description qui permet de rendre le contenu HTML attractif ( de faire la mise en forme du contenu).

**PHP** est un langage orienté serveur, il va permettre de manipuler les données reçues par l'interface web d'administration du projet (site web).

**JAVACRIPT ET JQUERY :** Le Javascript est le langage qui va nous permettre de rendre responsive le site web du côté client. JQuery est un framework du javascript, c'est un ensemble de lignes de code déjà implémenté qui va accélérer le développement de notre application.

**BASH** est un langage de script qui permet d'exécuter des instructions sur un système d'exploitation Linux. Par exemple, un script bash peut être écrit pour éteindre un ordinateur à une heure donnée. Il est intégré dans tous les systèmes Linux et ne nécessite aucune installation. Il est essentiel car il va 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

Le site d'administration développé va être envoyé sur le serveur grâce à un client graphique (comme FilleZila) qui utilise le protocole FTP. Le protocole FTP permet l'envoi et la réception de fichiers. Son usage est courant pour le téléchargement de fichiers et la majorité des dépôts Linux s'en sert pour le téléchargement des packages (apt le gestionnaire de paquets 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 et POP pour accéder

aux données (les mails).

## SMTP

Le serveur SMTP permet d'envoyer un mail. A titre d'exemple, si Toto (qui a pour adresse mail **toto@toto.coms**) veut envoyer un mail à Baké (qui a pour adresse **bake@gmail.com**), il va se produire les étapes suivantes :

- Toto va se connecter depuis son poste (ordinateur, téléphone portable) à son serveur SMTP, rédiger le mail et l'envoyer ;
- Le serveur SMTP de Toto (Toto est l'émetteur ou expéditeur) avant d'envoyer le mail, va vérifier si le destinataire du mail (dans notre cas Baké) est sur le même serveur. En d'autres termes, il vérifie si l'expéditeur et le destinataire sont sur le même serveur. Il se sert de la partie nom de domaine de l'adresse email pour la vérification. Dans notre cas le serveur SMTP de Toto vérifie si Baké appartient à ce même serveur. La partie nom de domaine de toto est **toto.com** et de baké est **gmail.com** . Les deux ne sont pas identiques donc Toto et Baké sont sur deux serveurs SMTP différents ;
- Dans le cas où le destinataire appartient au serveur SMTP de l'émetteur, le serveur SMTP de l'émetteur va enregistrer directement le mail dans le dossier de réception des mails du destinataire qui est situé sur ce même serveur ;
- Dans le cas où l'expéditeur et le destinataire sont sur différents serveurs, le serveur SMTP de l'expéditeur va transmettre le mail à un autre serveur SMTP. Ce serveur va ensuite vérifier si le courriel lui est destiné. Si le courriel ne lui est pas destiné, il va à son tour router le mail, c'est-à-dire envoyer le courriel à un autre serveur SMTP. Le mail sera routé jusqu'à arriver sur le serveur SMTP du destinataire. Le serveur du destinataire va stocker le mail dans le répertoire de réception du destinataire. Très souvent, il n'a pas de routage des mails et le serveur SMTP de l'expéditeur envoie directement le message au serveur SMTP du destinataire. Pour notre illustration, le serveur SMTP du domaine eneam.da va transmettre le mail au serveur SMTP du domaine gmail.com. Le serveur SMTP de gmail.com va réceptionner le mail dans le répertoire de Baké.



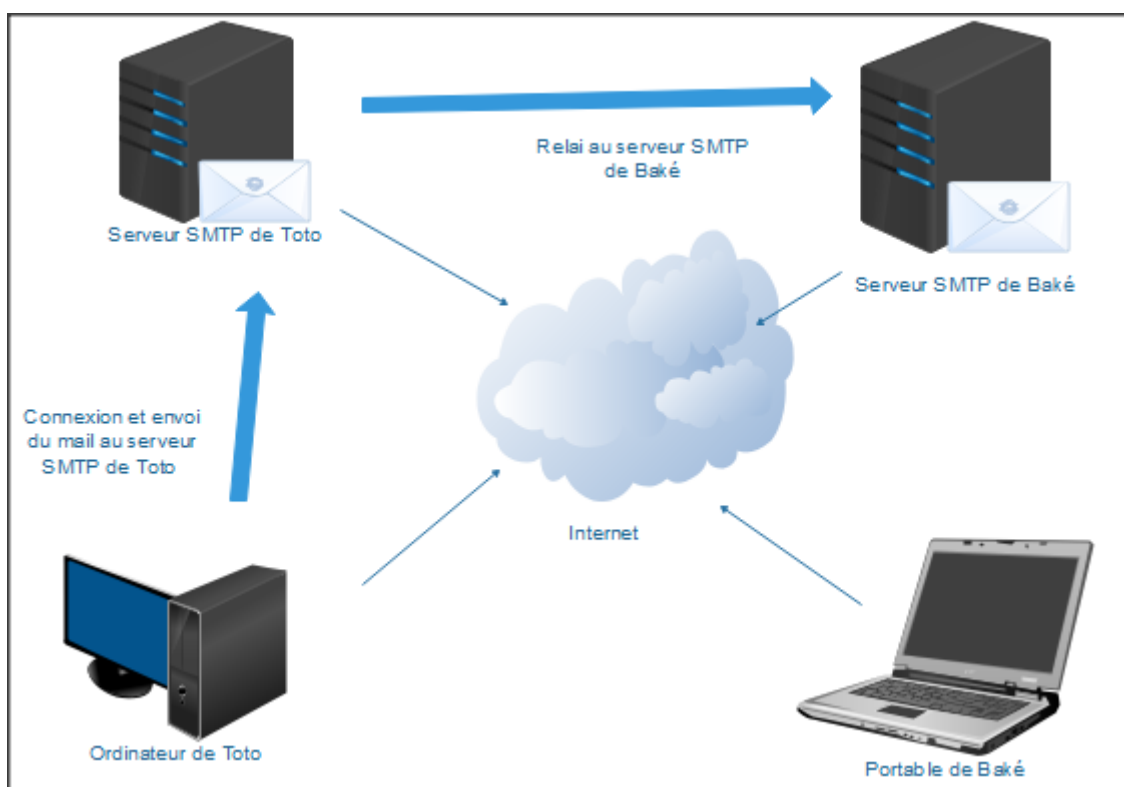


FIGURE 2.1 – Principe du SMTP

## IMAP

La lecture des mails se fait par l'établissement d'une connexion au serveur. On utilise le protocole IMAP qui va se connecter au serveur, récupérer et afficher les mails dans un logiciel approprié (Mozilla Thunderbird, Yahoo mail, Zimbra, Rainloop).

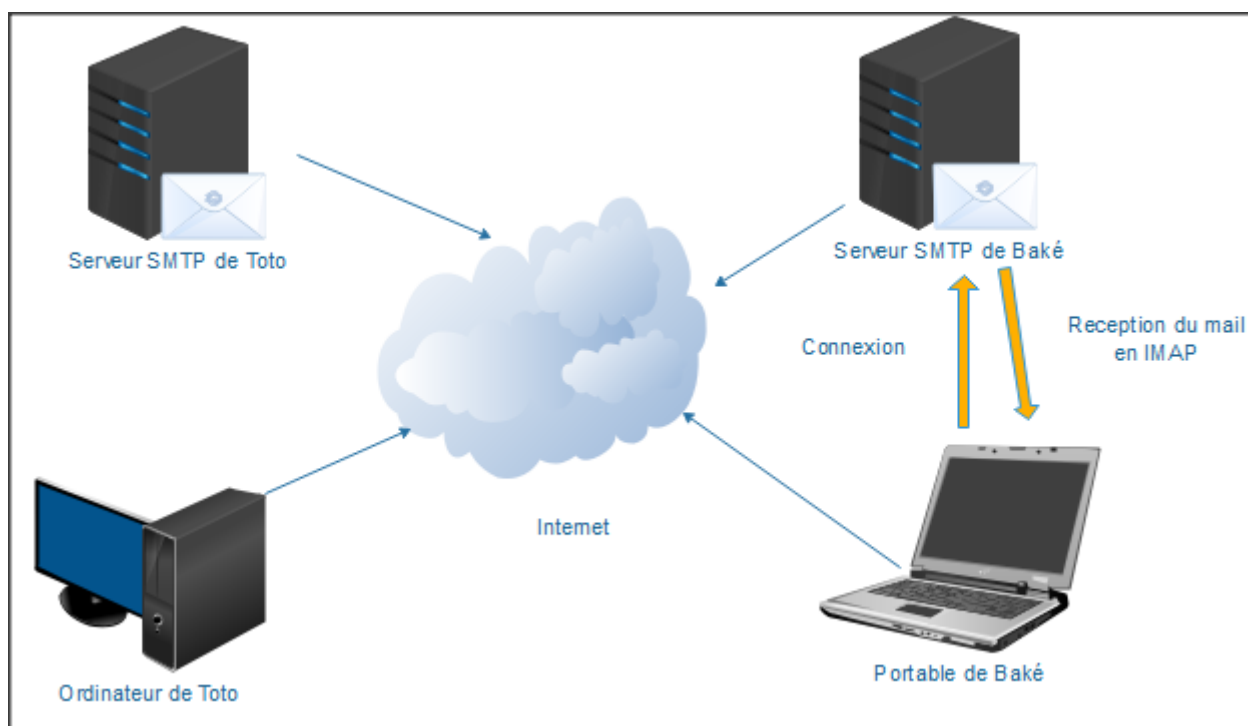


FIGURE 2.2 – Réception d'un mail en IMAP

Lorsqu'un compte mail est configuré avec IMAP, le client email établit une connexion avec le serveur avant chaque consultation. On accède donc aux dossiers et emails dont le contenu sera chargé directement depuis le serveur sur demande. Avec ce protocole tous les messages peuvent être enregistrés sur le serveur et donc être accessibles jusqu'à leur suppression définitive. IMAP permet de ce fait la consultation des messages depuis n'importe quel ordinateur ou client connecté à internet. IMAP utilise le port TCP 143 et permet avec la TLS (Transport Layer Security) de fournir un accès sécurisé au serveur. Il est aussi possible d'avoir un accès sécurisé en SSL via le port 993 mais ceci est déconseillé par la RFC 2595.

## POP

Avec le protocole POP (Post Office Protocol), le client de messagerie se connecte au serveur, copie tous les nouveaux messages présents sur sa boîte mail vers le disque dur de son ordinateur puis les courriels sont supprimés du serveur. Certains logiciels de courriel peuvent être configurés pour permettre de garder une copie des courriels sur le serveur. Grâce à POP on peut donc se connecter et récupérer ses messages vers son poste. En cas d'interruption de la connexion la gestion des mails se fait en hors-ligne. De plus, le protocole POP fournit un accès bloqué à la boîte mail c'est-à-dire qu'aucune autre connexion n'est permise en même temps que la connexion déjà en cours.

## Différences entre IMAP et POP

IMAP	POP
Connexion au port 143 (993)	Connexion au port 110 (995)
Connexion persistante (durable)	Connexion lors de l'extraction d'emails uniquement
Les emails sont conservés sur le serveur	Les emails sont supprimés immédiatement après avoir été téléchargés avec succès
Il est possible de charger les mails sur plusieurs clients	Un courriel ne peut être chargé que sur un seul client
Seuls les messages souhaités sont téléchargés	Tous les messages sont téléchargés

Il est conseillé de préférer le protocole IMAP au protocole POP car en plus de permettre l'accès multiple au serveur, le protocole IMAP rend possible le changement de client de messagerie sans risque de perte de données.

Le schéma complet de l'envoi et de la réception d'un mail avec le protocole IMAP.

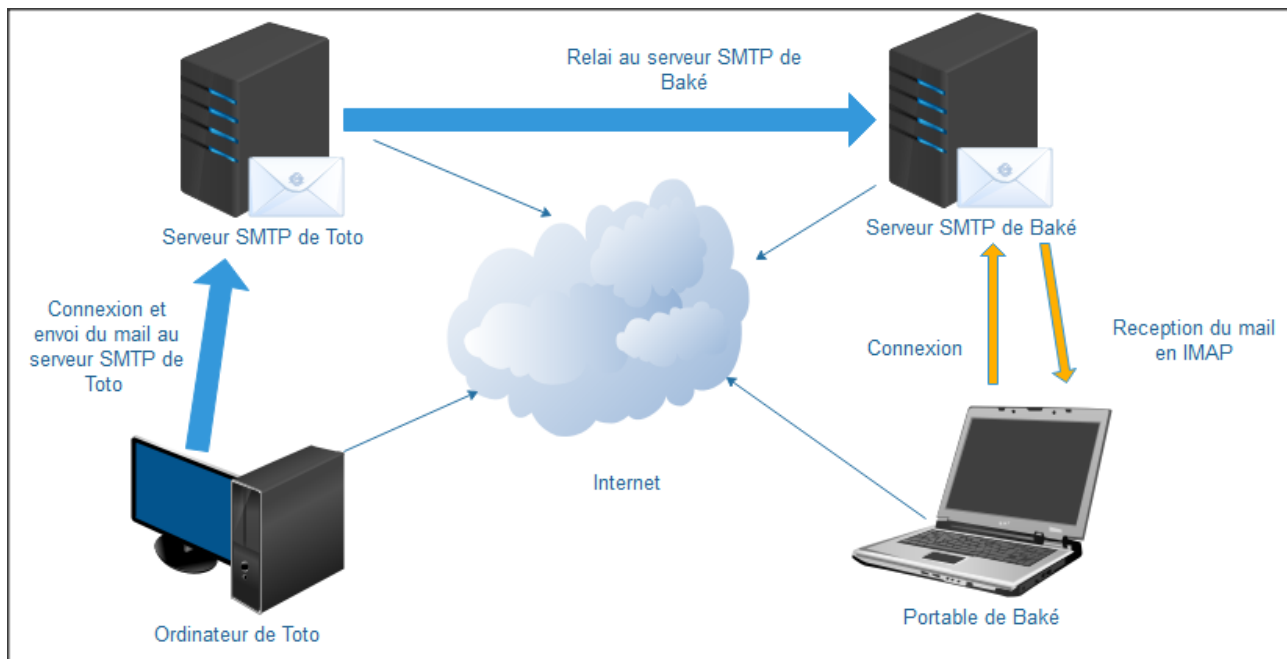


FIGURE 2.3 – Résumé de l'envoi et de la réception d'un mail

### 2.4.9 Sécurité

La sécurité impose de ne pas faire confiance aux utilisateurs et aux données reçues des utilisateurs. Il convient d'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 le serveur contre l'extérieur. Il va installer une barrière entre notre serveur et l'extérieur. Par exemple, il sera possible au serveur d'aller sur internet (communication serveur vers extérieur) ou d'empêcher les utilisateurs depuis l'extérieur à communiquer avec le serveur<sup>2</sup> (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 applicative (couche application). Par exemple, il peut empêcher les étudiants à se connecter aux serveurs à certaines heures). Il peut servir aussi de cache.

### Conclusion

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

- Un serveur SMTP et IMAP pour l'envoi et la réception des mails ;

2. Il a un troisième cas : traverser le serveur. Dans ce cas le serveur joue le rôle d'un routeur.

- 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 données pour sauvegarder les informations d'authentification ;
- Un serveur FTP pour envoyer les données du site d'administration sur le serveur ;
- Un site web d'administration développé en PHP et en javascript ;
- Des règles firewalls pour protéger le serveur ;
- Un anti spam pour empêcher les spammeurs(Spamassassin) ;
- Une communication sécurisée par le chiffrement et les certificats ;
- Un serveur DNS qui gère le domaine eneam.da. Le webmail aura pour adresse **eneam.da** et le site d'administration **admin.eneam.da**.

# Chapitre 3

## Implémentation du projet

Ce chapitre est subdivisé en deux parties. La première partie va mettre en évidence la configuration des différents services et leurs fichiers de configuration. En seconde partie, la section 3.13 à la page 50 va traiter d'un cas pratique de création et d'envoi de mails.

### 3.1 Installation du serveur

Nous utilisons comme OS **Ubuntu Server** et VMware comme hyperviseur<sup>1</sup>. 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 et correspond à une version d'Ubuntu qui sort tous les 2 ans et qui bénéficie d'un support étendu sur 5 ans. Son intérêt est que cette version est stable et ne nécessite pas de mise à jour régulière. Le choix de l'hyperviseur VMware est lié au fait qu'il supporte une meilleure intégration avec l'émulateur d'architectures réseaux GNS3.

La première étape est l'installation de l'hyperviseur VMware workstation. Pour cela, il faut :

- Se rendre sur le site de VMware en utilisant le lien <https://www.vmware.com> ;
- Télécharger, installer et ouvrir le logiciel. La version en anglais est utilisée ;
- Ouvrir l'onglet File ensuite "New virtual Machine". Une fenêtre s'ouvre et il faut suivre le guide durant la configuration de la nouvelle machine virtuelle. Les différentes étapes d'ajout d'une machine virtuelle sont disponibles sur le site de VMware. La machine virtuelle serveur a les configurations suivantes : 1 Go de mémoire RAM, 40Go de stockage de masse (disque dur).

Il faut ensuite :

- Démarrer la machine virtuelle **serveur** avec le disque Ubuntu Server qui est téléchargé en utilisant le lien <https://ubuntu-fr.org/telechargement> pour commencer l'installation ;

---

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 que nous pouvons installer Linux et tester le projet depuis un poste Windows.

- Suivre les instructions (choisir la langue, le clavier, configurer le réseau en DHCP). A la fin de l'installation le système demande de redémarrer ;
- Après redémarrage, entrer le nom d'utilisateur et le mot de passe. Il faut configurer le réseau.

La configuration du réseau nécessite d'avoir d'interfaces réseaux dans la machine virtuelle. Voici les étapes :

- Eteindre la machine virtuelle avec la commande

Console

```
sudo shutdown now
```

- Dans VMWare, éditer les paramètres de la machine virtuelle et lui ajouter un adaptateur réseau configuré sur NAT. La NAT va permettre à la machine d'avoir accès à internet<sup>2</sup> ;
- Démarrer la machine et taper la commande

Console

```
netplan apply
```

## 3.2 Gestion du stockage de fichier

Un disque dur spécial est utilisé pour stocker toutes les données liées aux mails que le serveur va contenir.

### 3.2.1 Système de fichier

Un fichier est un contenu (ou un conteneur de données) binaire qui porte un nom et a souvent une extension qui permet de le distinguer. Ainsi, il est facile de reconnaître une image par son extension jpeg ou un document word par docx. Les fichiers sont stockés sur un support de masse (disque dur, carte mémoire). Un système de fichier est un index qui définit comment les fichiers sont stockés et organisés sur un disque afin de permettre et de 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 (Windows ne supporte pas nativement le système de fichier EXT4). Donc pour stocker des informations sur un disque dur, il faut d'abord le préparer par la création du système de fichier et le formatage du disque.

### 3.2.2 Ajout de disque dur au serveur

- Eteindre le serveur s'il est allumé. Dans les paramètres de la machine virtuelle, nous ajoutons trois disques de 2 giga configurés en SCSI<sup>3</sup> ;

---

2. L'accès à internet sera modifié après lors de la configuration avec GNS3.

3. Est une spécification matérielle des disques.

- Redémarrer la machine et exécuter la commande :

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

- Vérifier la taille de chaque disque avec la commande

```
Console  
gdisk dev/sdb
```

Dans gdisk, afficher la structure du disque pour vérifier que le stockage du disque est effectivement de deux giga. Taper q pour quitter. Répéter la commande pour les deux autres disques durs montés ;

- Créer 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
```

- Afficher les détails sur le nouveau disque RAID créé et copier 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	

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

```

----- 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, faire

```

----- Console -----
sudo update-initramfs -u

```

- Passer 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

```

- Créer le dossier de montage puis monter la partition **data** dans ce dossier

```

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

```

- Rendre automatique le montage de la partition à chaque démarrage du système. Pour cela, éditer le fichier `/etc/fstab` et ajouter la ligne

```

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

```



## 3.3 Configurer le LAMP

Dans cette section, Nous allons installer PHP, MariaDB qui est un fork<sup>4</sup> de MySQL.

### Console

```
sudo su
apt-get update #Pour mettre à jour le cache local
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 en local (l'interface localhost) sur 7080 pour le HTTP et 7443 pour le HTTPS. L'avantage est que notre serveur apache n'est pas exposé à l'extérieur. On verra seulement notre reverse-proxy Nginx de l'extérieur. Nous modifions 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 auto-signés<sup>5</sup>. Nous créons le fichier `/etc/apache2/site-available/www.eneam.da.conf`; il est associé au client webmail qui va lire les courriers.

---

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.

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

## 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]
    </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>

```

Nous créons le fichier `/etc/apache2/site-available/www.admin.eneam.da.conf`; il est associé au site d'administration.

## 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>

```

Nous créons aussi les répertoires `/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'activons pas ces virtualhosts pour éviter une erreur d'Apache puisque leurs répertoires respectifs n'ont pas été créés.

### 3.3.2 Gestion de Nginx

Console

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

Il faut créer les **server blocs**<sup>6</sup> 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;
}

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 {
```

---

6. Dans Nginx, les virtualhosts sont appelés des servers blocs.

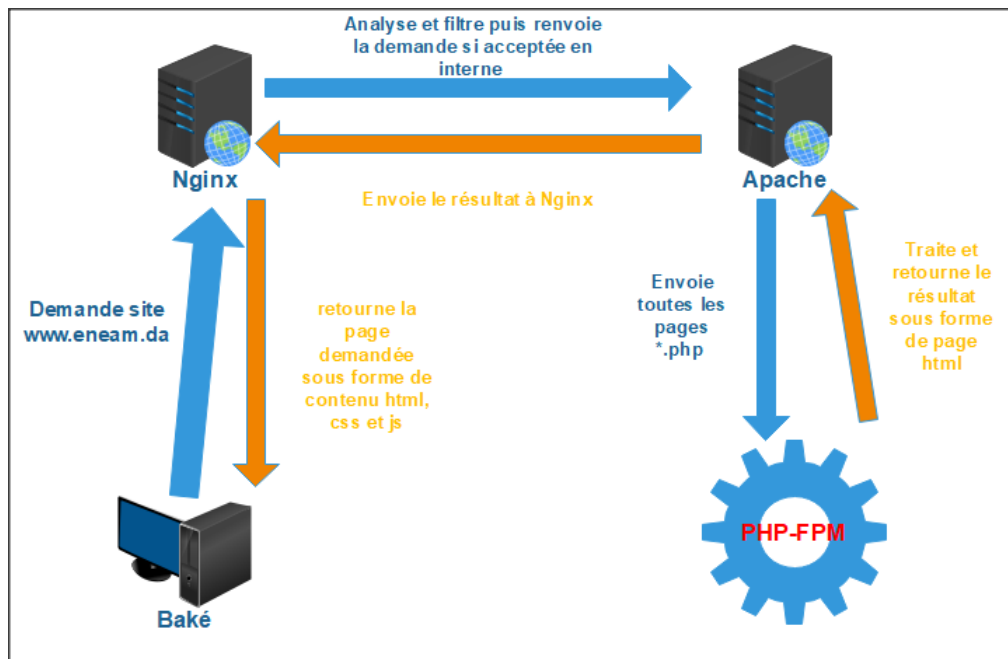
```
listen 80;
server_name www.eneam.da eneam.da;
return 301 https://$host$request_uri;
}

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 la gestion 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

Il faut :

- Télécharger les logiciels GNS3 et GNS3 VM sur le site officiel de GNS3 et installer GNS3 ;
- Lancer le logiciel VMWare et cliquer sur Fichier puis sur ouvrir fichier ;
- Choisir le fichier GNS VM au format ova téléchargé. Une nouvelle machine virtuelle va être ainsi installée. Il faut ensuite importer cette machine dans le logiciel GNS3 ;
- Lancer GNS3. Menu Edit -> Préférences -> GNS3 VM ;
- Cocher la case “Enable the GNS3 VM” et sélectionner “VMware Worstation/Player” dans le champ Virtualize engine ;
- Cliquer sur Refresh et GNS3 VM apparaît dans VM name ;
- Cliquer sur Apply ;
- Ajouter notre machine Linux **serveur** de VMware pour l'utiliser lors de la modélisation. Pour cela, se rendre dans Preferences -> VMware -> VMware VMs -> new. Une fenêtre apparaît. Cocher run this VMware VM on my local computer puis next ;
- Dans VM list, choisir la machine serveur, un clic sur le bouton finish.

### 3.4.1 Description du schéma

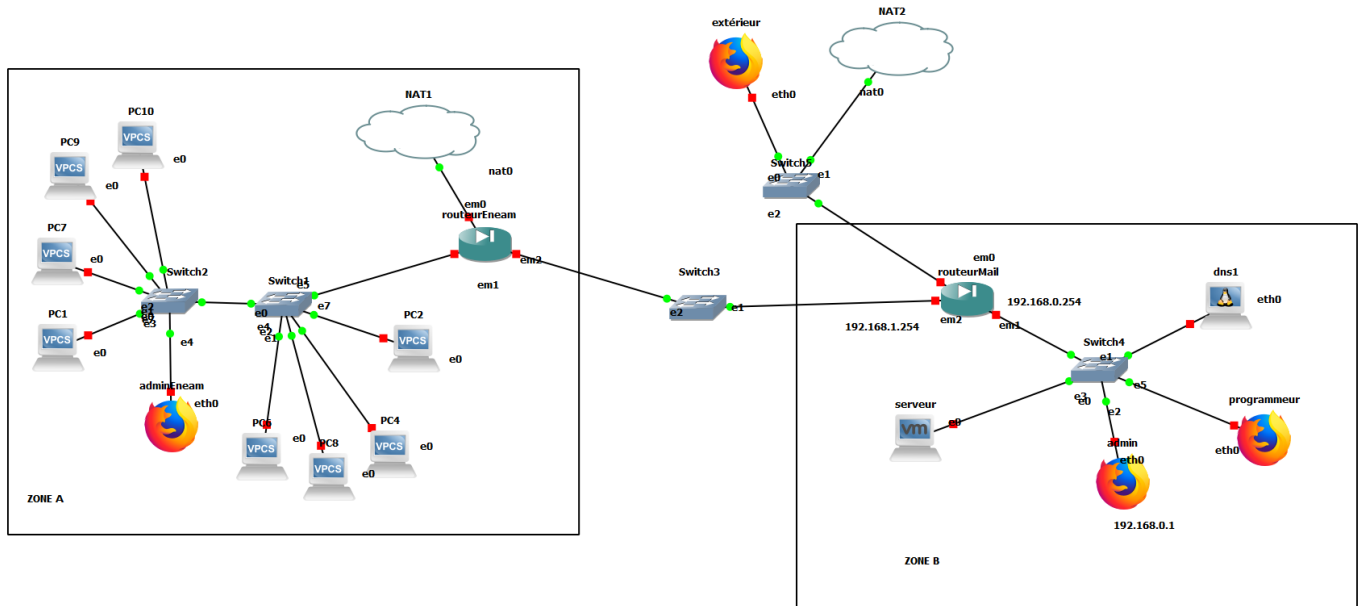


FIGURE 3.2 – Topologie de notre projet dans GNS3

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 du projet et de comprendre que le projet est générique car il peut s'adapter à tout type d'organisation existante sans toucher au système d'information<sup>7</sup> 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** possède 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 docker<sup>8</sup> 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 Pfsense<sup>9</sup> ;

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 n'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 création du projet :

- Créer un nouveau projet "projetmail" ;
- Ajouter tous les équipements à notre topologie ;
- 
- Renommer tous les équipements ajoutés ;
- Démarrer et configurer routeurEneam ;
- Configurer le serveur DNS sur le poste **dns1** ;
- Accéder à routeurEneam depuis le poste admin et configurer la NAT et le port forwarding (redirection de port en français). Pour cela :
  - Depuis le poste **admin**, ouvrir le navigateur Firefox ;
  - Entrer l'adresse 192.168.0.254 et accéder à routeurEneam. Les identifiants par défaut sont admin et le mot de passe pfsense. Changer le mot de passe par défaut.

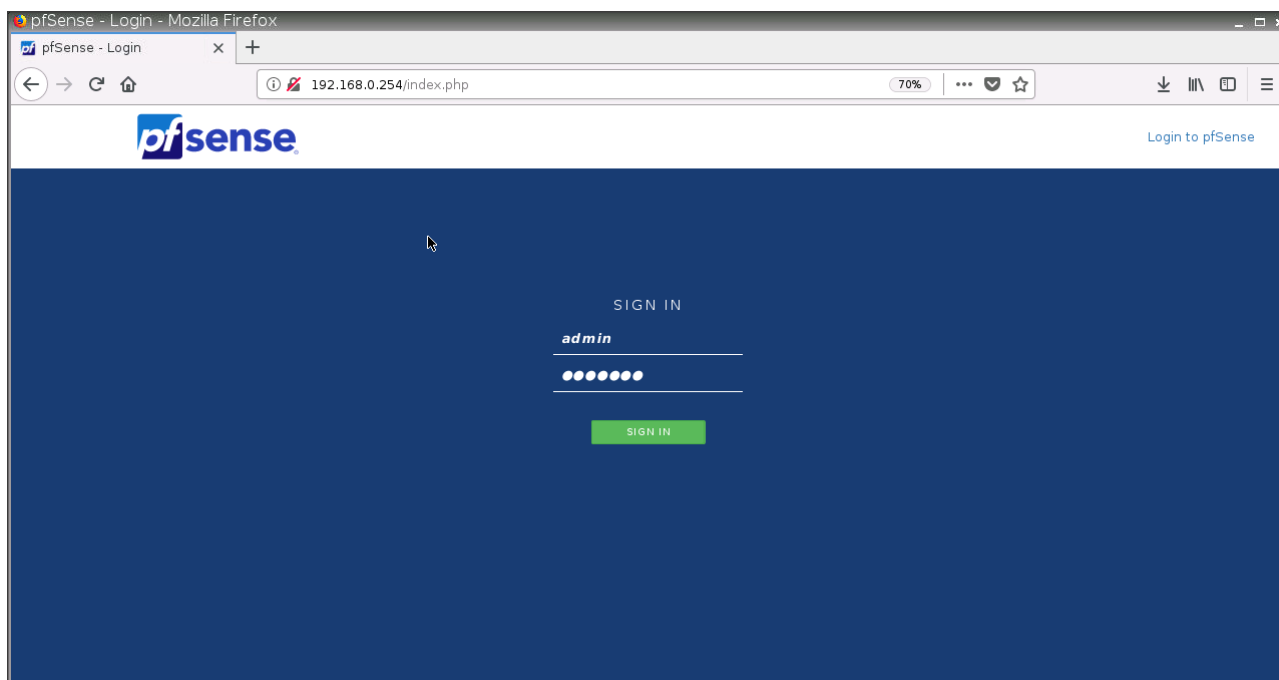


FIGURE 3.3 – Connexion à pfsense par un navigateur

Cette configuration va permettre d'accéder au mail par le webmail mais aussi d'y accéder par d'autres clients mails tels que Mozilla Thunderbird puisque que les ports STMP et IMAP sont ouverts vers l'extérieur. L'intérêt est qu'il serait possible de s'envoyer des mails sans passer par un client webmail installé sur le serveur.



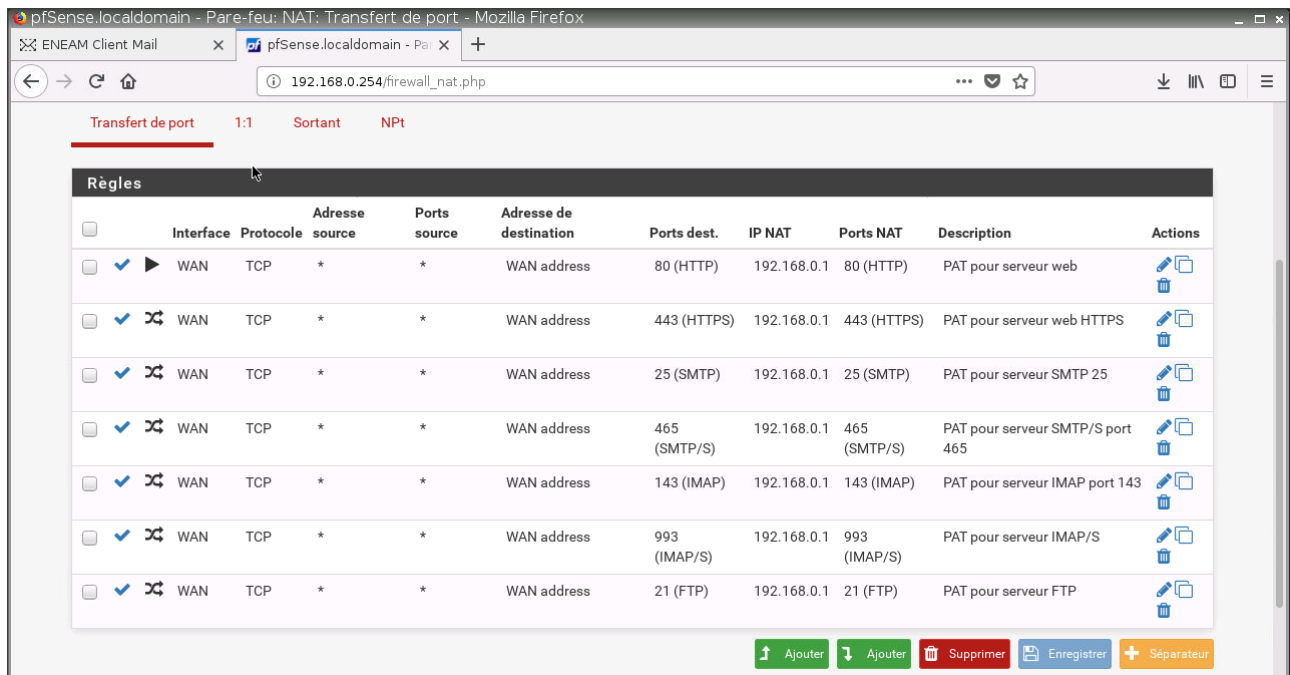


FIGURE 3.4 – Configuration du port forwarding

- 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-Only<sup>10</sup>. Nous créons aussi le fichier `/etc/netplan/60-lan_statique.yaml` pour configurer les interfaces du serveur.

```

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 notre machine physique Windows, il faut un moyen pour pouvoir envoyer le site sur le serveur virtuel, alors nous avons configuré un réseau privé entre le serveur de VMware et notre 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/`. Nous réalisons une copie des fichiers de configuration avant toute modification. Nous modifions 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 nous utilisons des certificats auto-signés ;
- 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 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 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 anti spam.

## 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
```

Nous 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
```

Nous modifions le fichier `/etc/dovecot/conf.d/10-auth.conf`.

Console

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

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

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
    }
}

```

Le fichier `/etc/dovecot/10-ssl.conf` va contenir les informations pour sécuriser le serveur avec les certificats :

#### Console

```

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

```

## Autoriser l'administrateur à accéder à tous les comptes

Nous créons le fichier `/etc/dovecot/conf.d/auth-master.conf.ext`

Console

```
passdb {
    driver=sql
    #driver = passwd-file
    master=yes
    args=/etc/dovecot/dovecot-sql-master.conf.ext
    #important pour que l'admin se connecte aux utilisateurs qui existent
    ↪ uniquement
    result_success = continue
}
```

Nous créons le fichier `/etc/dovecot/dovecot-sql-master.conf.ext`

Console

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

### 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
#Nous activons 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
```

Nous changeons 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 :

- Nous tapons `www.eneam.da/?admin` dans le navigateur Firefox depuis le poste admin ;
- Nous entrons les identifiants par défaut admin et mot de passe 12345 ;
- Nous changeons les identifiants ;
- Nous configurons la langue en français ;
- Nous ajoutons les serveurs SMTP et IMAP de notre domaine en cliquant sur le menu domaine.

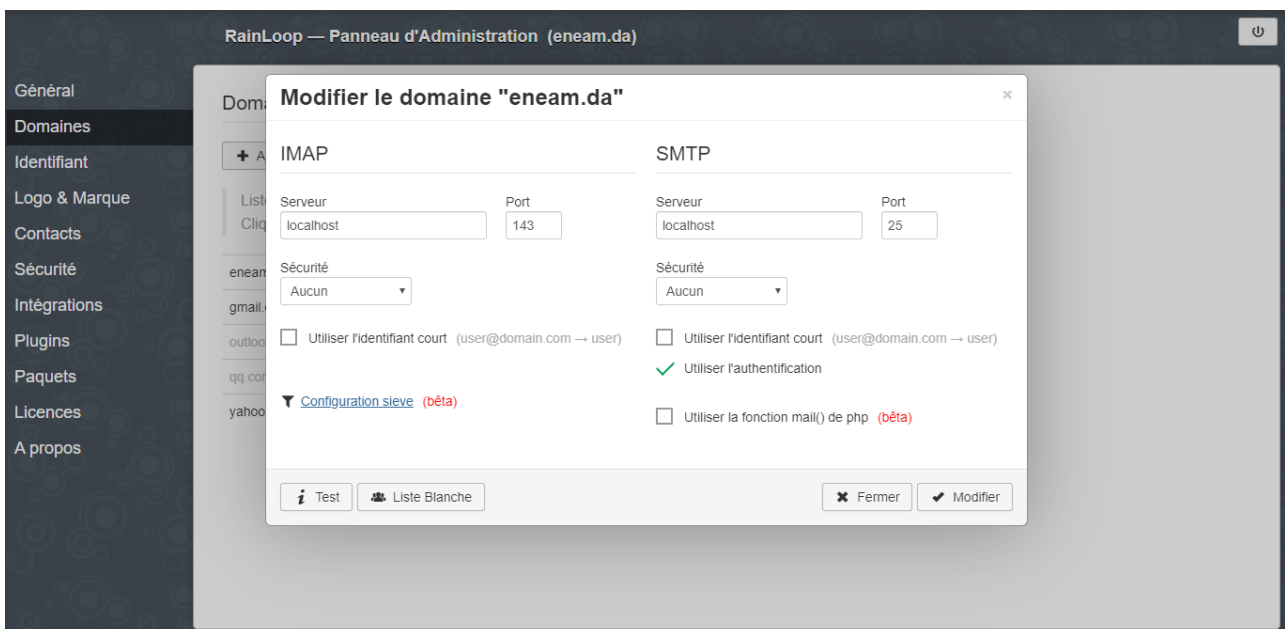


FIGURE 3.5 – Ajout des serveurs mails dans rainloop

## 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 ), leurr états, de les arrêter ou de les éteindre. Il existe des solutions gratuites pour l'administration du mail comme **postfixadmin**

### Les raisons pour lesquelles nous n'utilisons pas une solution existante

- Les solutions existantes contiennent beaucoup de fonctionnalités que nous avons jugé inutile dans notre contexte. Par exemple nous n'avons pas besoin des alias. Nous ne

voulons pas qu'un utilisateur soit en mesure d'accéder à son compte avec différents adresse mails.

- L'apprentissage : nous allons comprendre les principes de base pour développer un site web d'administration de mails et écrire des scripts bash. Nous allons combiner toutes les technologies reçues lors de notre apprentissage pour produire un résultat.
- Postfixadmin est un outil 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 efficaces, complets mais pas gratuits et qui contiennent des fonctionnalités qui ne sont pas nécessaire pour nos besoins.<sup>11</sup>

## Configuration

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

declare -A service
service[apache2]="apache2"
service[nginx]="nginx"
service[postfix]="postfix"
service[dovecot]="dovecot"
service[phpfpm]="php7.2-fpm"
```

11. Beaucoup d'entreprises préfèrent souvent des solutions complètes, faciles d'utilisation comme cPanel. Mais ces outils sont à des prix très onéreux.

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

```

                                retour=1
                                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
                fi
            fi
        done
    done
fi

```

```

                                fi
                        else
                                exit 0
                        fi
                fi
        fi
done
fi

```

Pour permettre l'exécution de scripts bash depuis le web, il faut autoriser l'utilisateur web **www-data** à lancer des scripts avec des droits administrateurs. Nous installons le programme `sudo` et nous ajoutons 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 définit les droits par défauts 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 la 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
```

Nous créons 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 anti spam. Il va lire dans les logs et vérifier le nombre de tentatives de connexion échouée ou autres paramètres. Si on 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 spam<sup>12</sup>. Plus les règles établies sont strictes, plus les mails seront rejetés.

---

12. Contenu, mail indésirable.

## Console

```
apt-get install spamassassin
```

Nous créons un utilisateur propre à spamassassin

## Console

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

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 données 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 */;
/*!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$by9wZ25rUWkyeGhrVERuSw$du7tQ5d8JQHsSq
  ↪ 6FnwTIsyrta2IFULwyNLau4DHEsMg','chal@eneam.da','eneam.da/chal/'),(14,1,
  ↪ '$argon2i$v=19$m=65536,t=4,p=1$ekZ5dG0ubGRFTmdwSmtFNw$ADVj4M3MIvvnK8jzm
  ↪ OEwMHZbbPuo62XfA9010wBPJ3M','gdg@eneam.da','eneam.da/gdg/'),(15,1,'$arg
  ↪ on2i$v=19$m=65536,t=4,p=1$czdIQ2tnLkhjYlIORVY4Ng$fiTJkvfRPvon656BGtRN6J
  ↪

```

```

/*!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 : Nous allumons le poste admin. Nous ouvrons le navigateur et nous entrons l'adresse `www.admin.eneam.da` ou `admin.eneam.da`.
- L'administrateur renseigne ses informations de connexion : son mot de passe est **amettre** et son adresse mail est **admin@eneam.da**.

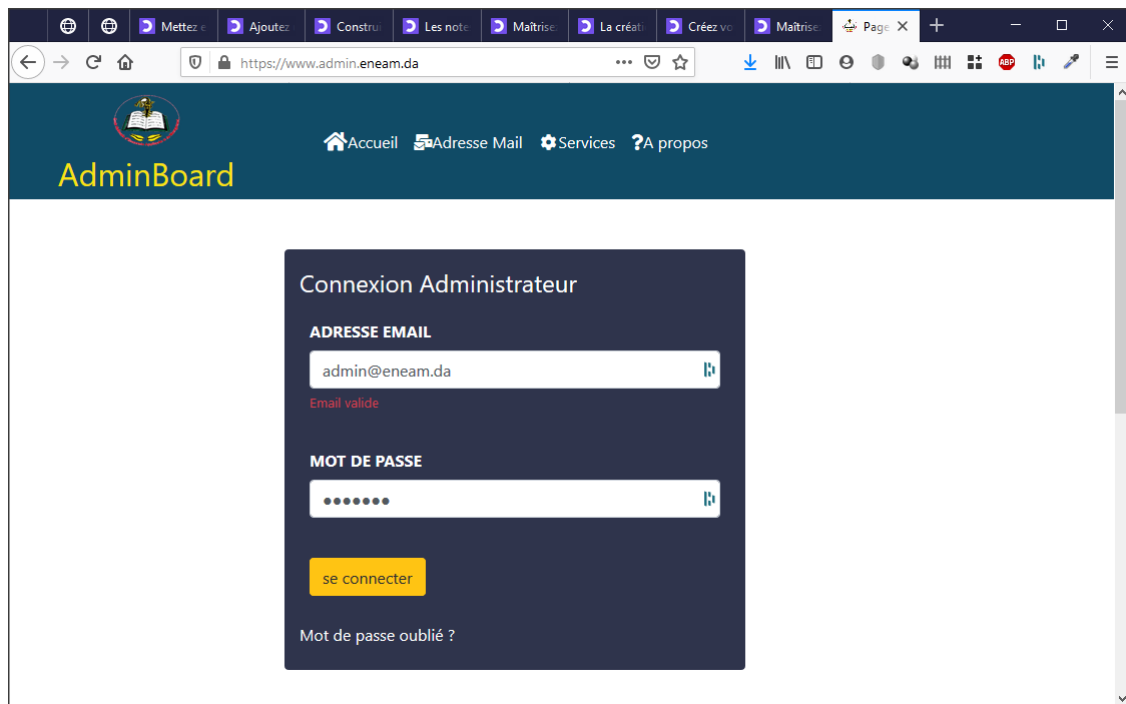


FIGURE 3.6 – Connexion de l’administrateur au site d’administration

- Sur la page d’accueil, il renseigne le compte mail qu’il veut créer. Ici nous allons mettre `bake@eneam.da`. Puis nous renseignons le mot de passe qui doit avoir une forte entropie<sup>13</sup>. On suppose ici `Ba21@kesccT`. Nous confirmons le mot de passe.
- Facultatif : Nous cochons 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.<sup>14</sup>

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, il est considéré qu’un compte est valide durant la période d’étude. On utilisera le programme **cron** pour désactiver automatiquement les comptes expirés.

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

- Nous cliquons 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.

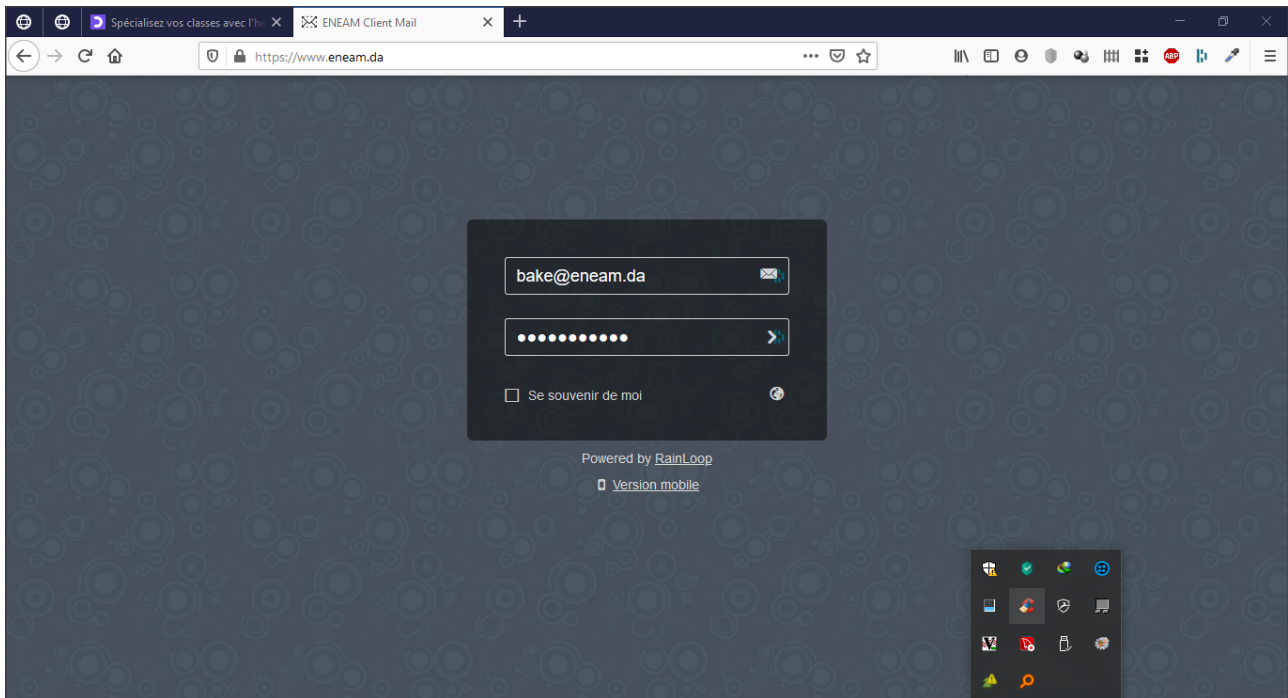


FIGURE 3.8 – Connexion de Baké au client webmail

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

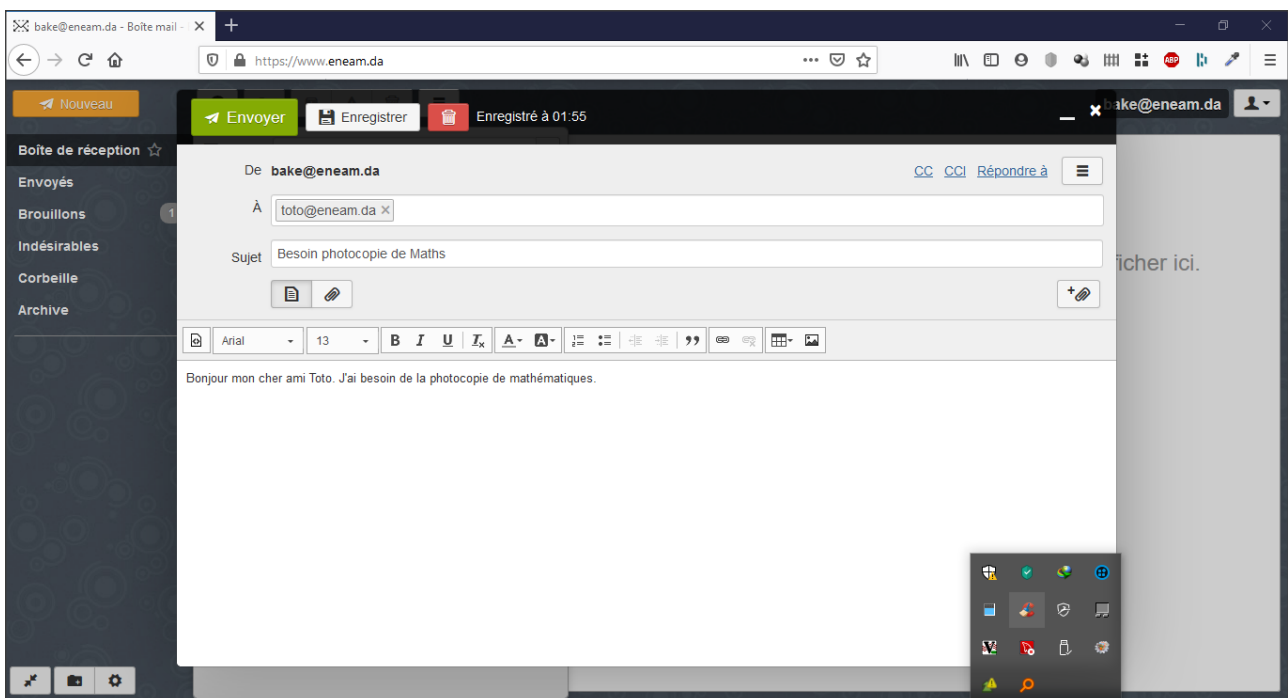


FIGURE 3.9 – Envoi d’un mail de Baké à Toto

— Toto se connecte voit le message et répond.

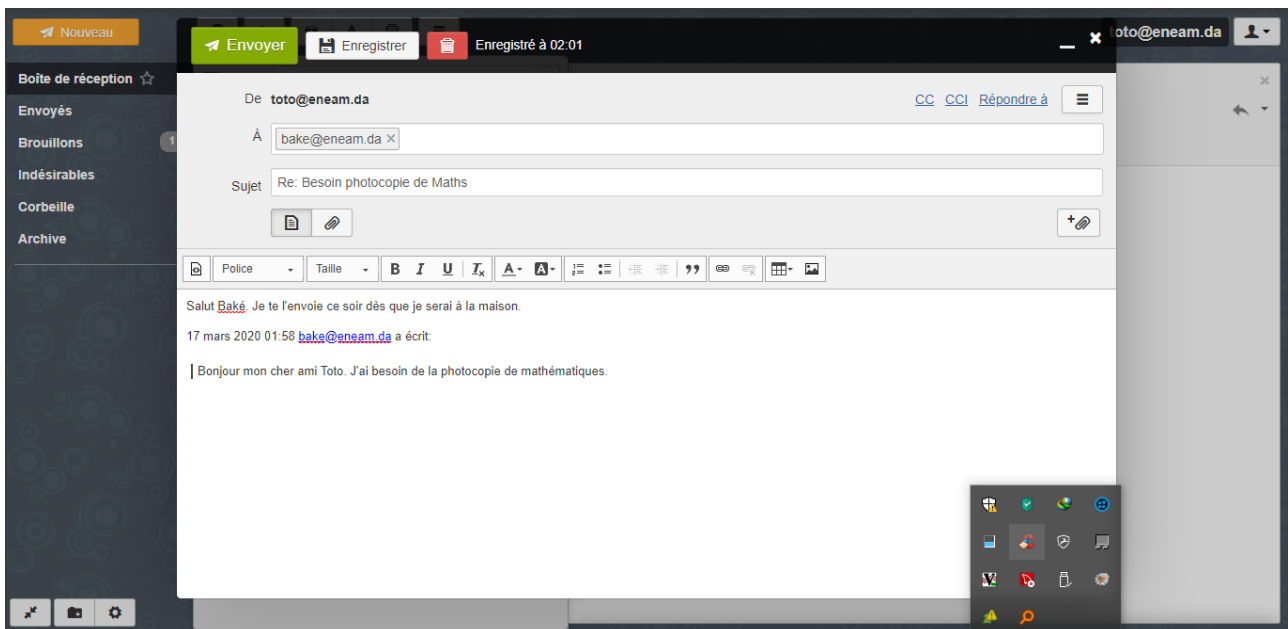
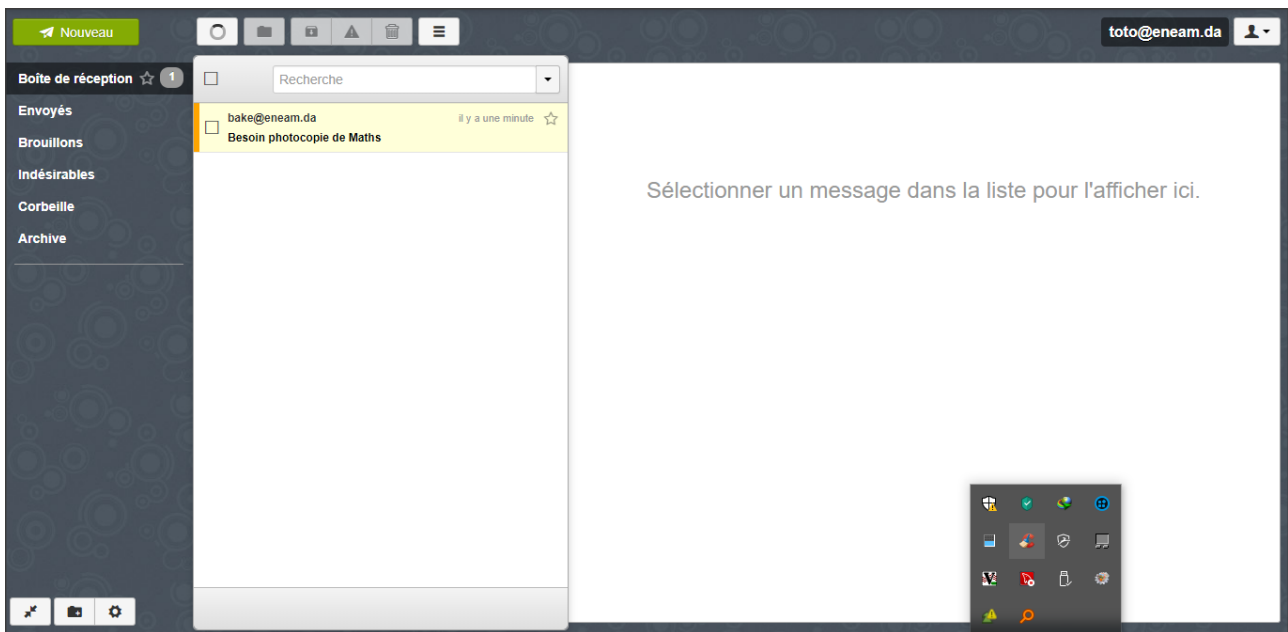


FIGURE 3.10 – Réponse de Toto au mail de Baké

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

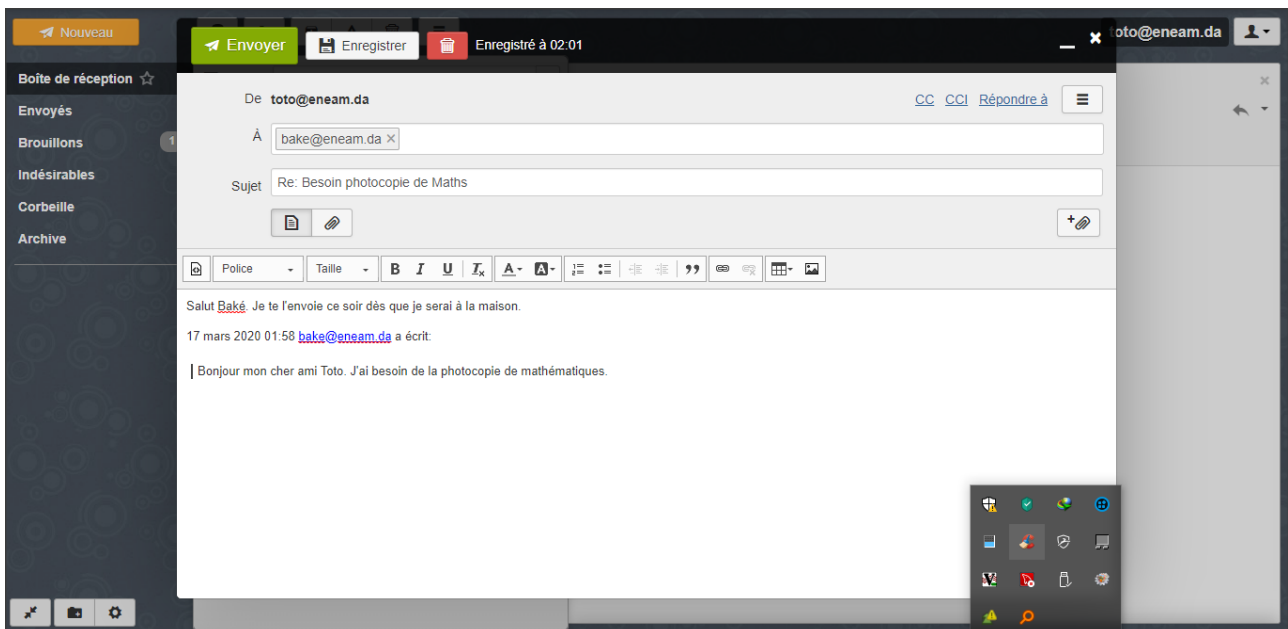
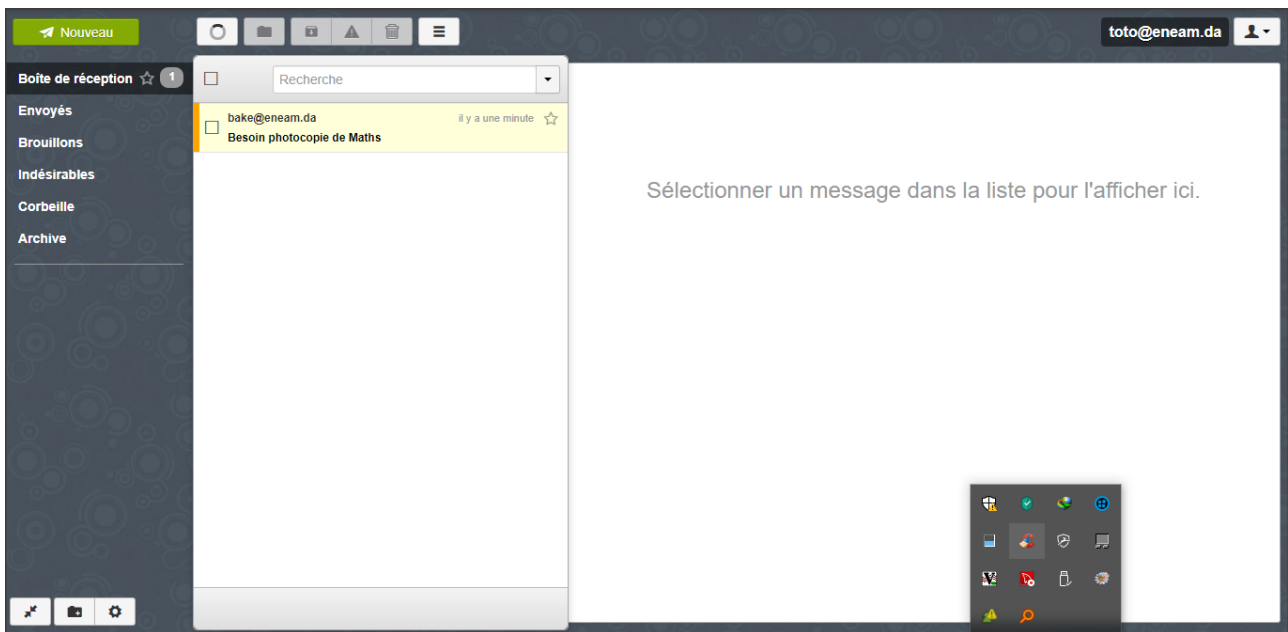


FIGURE 3.11 – Lecture du mail reçu de Baké par Toto

- 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 pop-ups apparaissent pour notifier si le compte a été supprimé. Il a la possibilité de les effacer ou de les enregistrer dans un fichier au format texte.

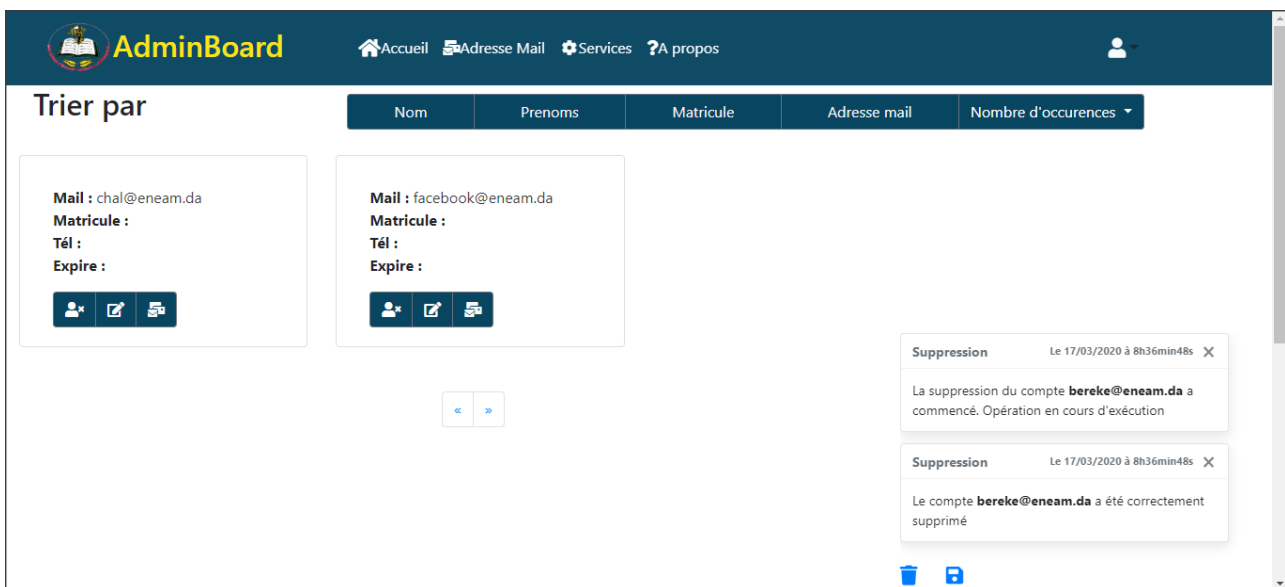
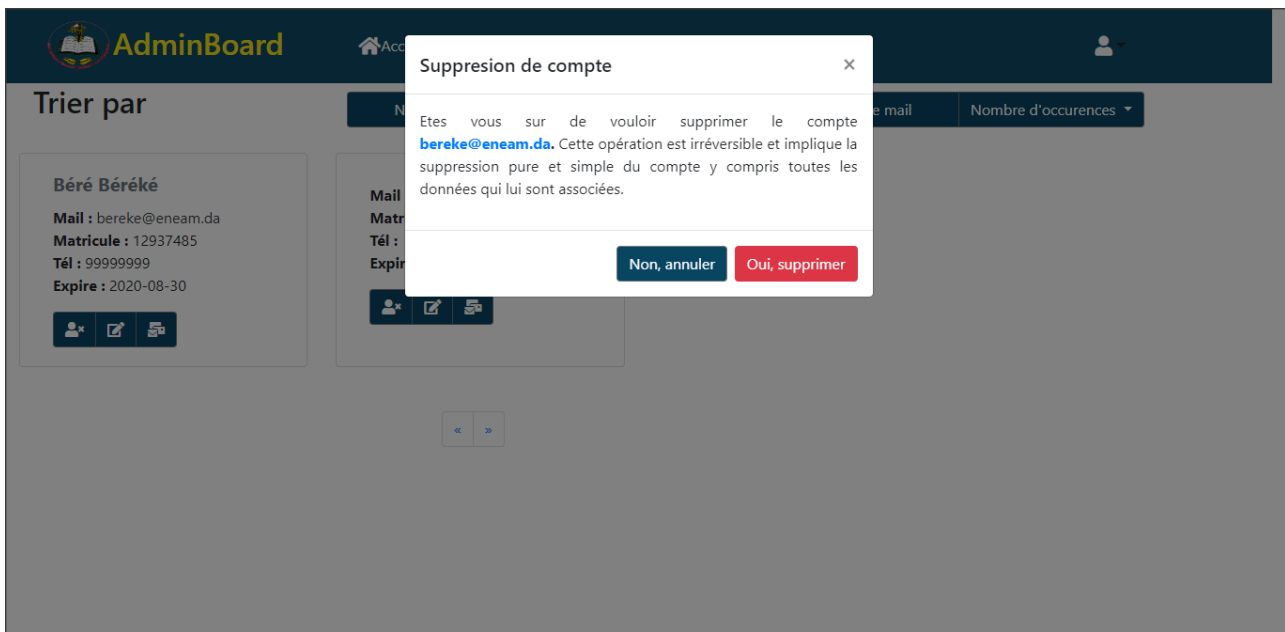
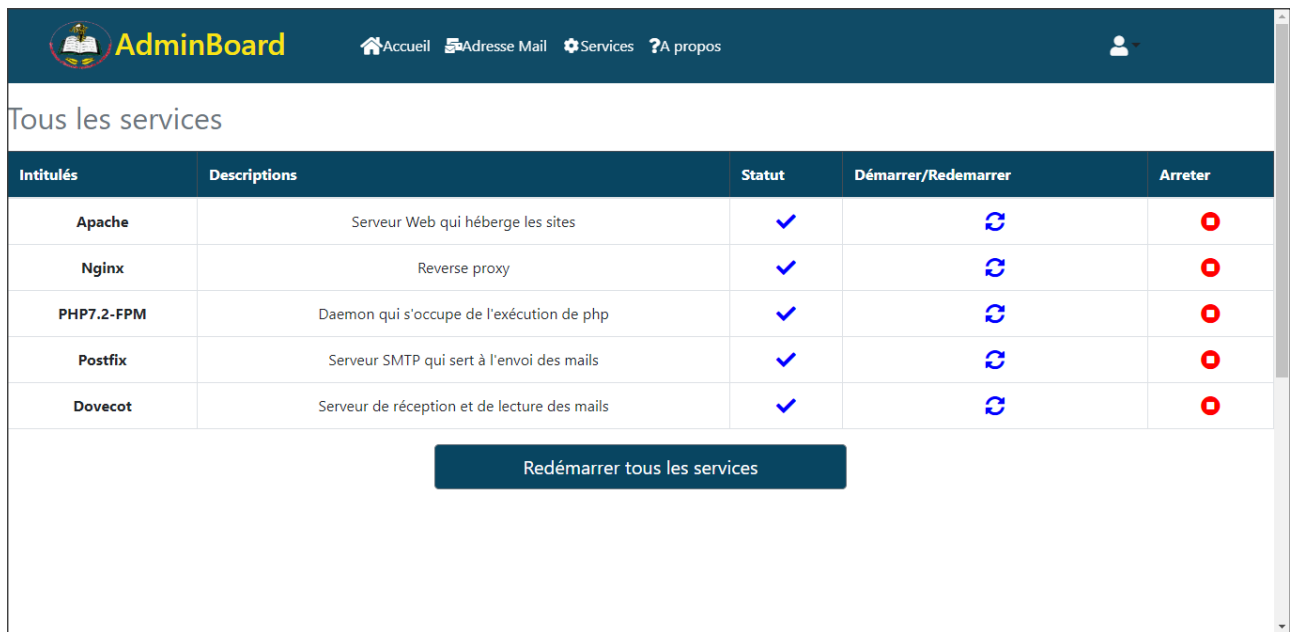


FIGURE 3.12 – Suppression du compte de Béréké

- 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.

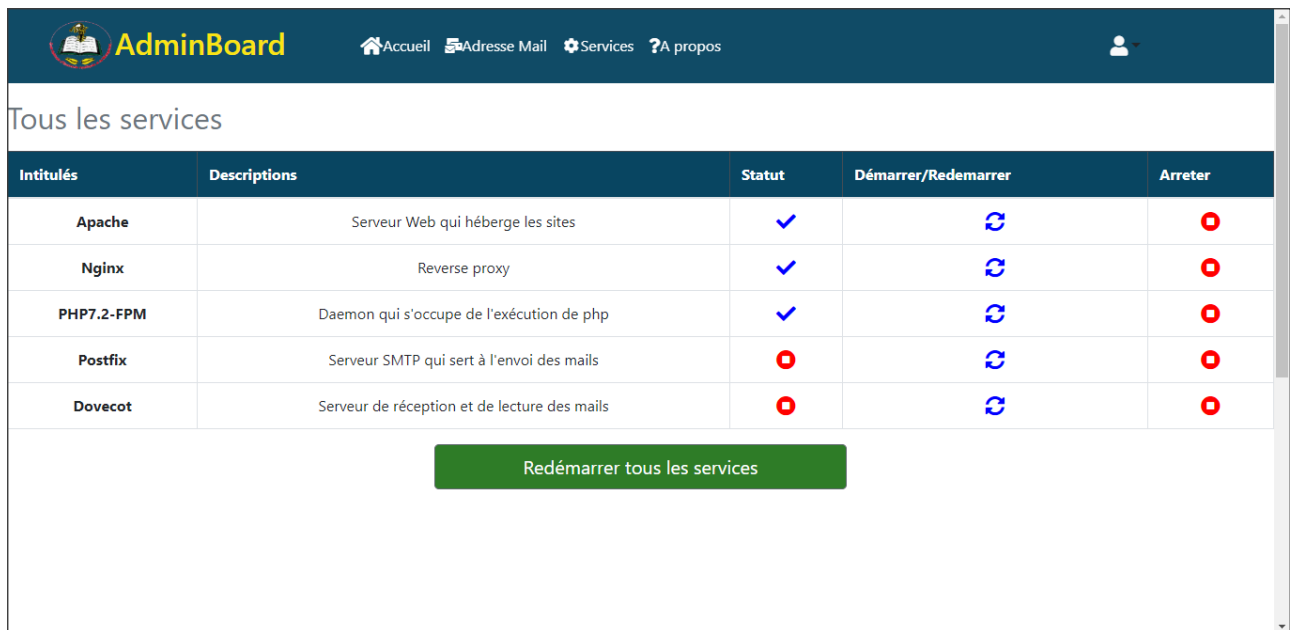


Intitulés	Descriptions	Statut	Démarrer/Redémarrer	Arrêter
Apache	Serveur Web qui héberge les sites	✓		
Nginx	Reverse proxy	✓		
PHP7.2-FPM	Daemon qui s'occupe de l'exécution de php	✓		
Postfix	Serveur SMTP qui sert à l'envoi des mails	✓		
Dovecot	Serveur de réception et de lecture des mails	✓		

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'icône 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

Nous avons déployé un serveur Linux Ubuntu qui permet l'envoi de messages électroniques. Le service de messagerie va contribuer à favoriser les échanges entre étudiants à l'école. En effet, par ce canal l'administration et les professeurs peuvent diffuser certaines informations aux étudiants sans risque d'altération ou de modification de l'information par les étudiants. La mise en place a nécessité d'installer un serveur de fichier, de configurer les services DHCP, DNS, Apache, MySQL, de faire de la virtualisation avec VMware et de la modélisation réseau grâce à GNS3, de configurer les protocoles SMTP et IMAP, de définir une politique de sécurité, d'utiliser la cryptographie pour sécuriser la communication et de développer un site web d'administration.

Notre stage académique effectué au sein de JScom s'est révélé être une expérience marquante et nous a montré un aperçu des réalités quotidiennes en milieu professionnel. Nous avons pu découvrir quelques outils d'administration système sous Linux.

# Bibliographie

- [1] Install and configure postfix and dovecot, Janvier 2019. [www.linuxize.com](http://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.