

# ECOLE NATIONALE D'ECONOMIE APPLIQUÉE ET DE MANAGEMENT



---

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

---

Par :  
Picasso HOUESSOU-DOSSOU

Sous la supervision de :  
Dr. MM

Encadreur :  
M Bruno Bellarmin LAWSON

March 12, 2020

# 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 John Aoga, pour le suivi de la rédaction du présent mémoire et pour les conseils prodigués afin de tirer au maximum profit de notre stage.

Que Dieu vous Bénisse.

Picasso Houessou

# Sigle et Abréviation

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

**HTML** : HyperText Markup Language.

**CSS** (Cascading Style Sheets

Bash : Bourne Again Shell

**RAID** : Ecole Nationale d'Economie Appliquée et de Management.

**LVM** : HyperText Markup Language.

**IMAP** (Cascading Style Sheets

**SMTP** : Bourne Again Shell

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>1 Cadre de l'étude</b>	<b>8</b>
1.1 Présentation du bureau d'étude JScom . . . . .	8
1.1.1 Présentation générale . . . . .	8
1.1.2 Situation géographique . . . . .	8
1.1.3 Domaine d'intervention . . . . .	8
1.2 Déroulement et observations . . . . .	8
<b>2 Cadre théorique et méthodologique</b>	<b>9</b>
2.1 Contexte et énoncé du problème . . . . .	9
2.2 Objectif . . . . .	10
2.3 Hypothèse . . . . .	11
2.4 Etude théorique . . . . .	11
2.4.1 Définition d'un serveur . . . . .	11
2.4.2 Disque dur RAID LVM . . . . .	12
2.4.3 Serveur web APACHE . . . . .	12
2.4.4 Base de donnée . . . . .	12
2.4.5 Modélisation avec GNS3 . . . . .	13
2.4.6 Programmation . . . . .	13
2.4.7 Fonctionnement du mail . . . . .	14
2.4.8 Sécurité . . . . .	15
<b>3 Implémentation du projet</b>	<b>17</b>
3.1 Monter le serveur Linux . . . . .	17
3.1.1 Installation du serveur . . . . .	17
3.2 Gestion du stockage de fichier . . . . .	18
3.2.1 Systeme de fichier . . . . .	18
3.2.2 Ajout de disque dur au serveur . . . . .	18
3.3 Configurer le LAMP . . . . .	20
3.4 Modélisation de l'architecture reseau avec GNS3 . . . . .	21
3.4.1 Description du schéma . . . . .	21

3.4.2	Configuration des équipements . . . . .	22
3.5	Installation Postfix . . . . .	22
3.5.1	Création de la base de donnée des tables . . . . .	26
3.6	Installation de Dovecot . . . . .	26
3.7	Installation du client mail . . . . .	29
3.8	Le site d'administration . . . . .	29
3.9	Configuration du FTP avec vsftpd . . . . .	34
3.10	Spamassassin . . . . .	36
3.11	Sécurité . . . . .	37

# List of Figures

2.1 Principe du SMTP . . . . . 14

# 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. Ce qui pose plusieurs problèmes. J'ai constaté que les étudiants sont informés tardivement de l'annulation d'un cours et parfois de l'effectivité d'un cours , cela même parfois lors des devoirs. Je fus surpris lors 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, sur, efficace , centralisé, sécurisé pour discuter et ou 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 l'ENEAM.

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 ;



# Chapter 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

Cotonou, Sikècodji, carré 1234 maison

#### 1.1.3 Domaine d'intervention

### 1.2 Déroulement et observations

# Chapter 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 de problèmes et d'incompréhension dans la programmation des cours à ENEAM. Les étudiants sont souvent informé tardivement qu'il y a cours. De même quand on annule un cours les élèves sont informés tardivement. Souvent ils sont à l'école et attendent le professeur qui annonce qu'il ne viendra pas . Ils s'amusent et perdent leur temps à ne rein 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 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, ce qui poussent les étudiants à dirent 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 les procédures de réclamation et autres services disponibles temporairement
- Un système de communication sûre , multiutilisateur sécurisée.

Voyons 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 d'ingénieurs et de technologies réunies. Nous voyons bien que cette solution bien que faisable, 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 acheminent 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 structuration 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 élèves 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.
- L'administration de l'école peut envoyer la programme des cours, les 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 sur s'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 partir de zero. 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 GNS , 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'administration , 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 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 d'autres logiciels.

### Les caractéristiques d'un serveur

Un serveur à généralement les caractéristiques suivantes

- Un serveur est allumé 24h/24h
- tres souvent , il ne dispose pas d'un écran , ni d'un clavier , ni d'équipement multimédias.
- Un serveur Linux n'a généralement pas d'interfaces graphiques
- 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: annuire LDAP
- Web

## Spécification d'un OS serveur

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

- Les OS serveur ne sont pas configurés avec les fonctions de veilles. En effet les serveurs restent allumés tout le temps
- Les OS serveur n'ont pas d'interfaces graphiques 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 To de mémoire RAM. Il existe des services de support payants pour les entreprises.

### 2.4.2 Disque dur RAID LVM

RAID : C'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 d'un des disque se gatte, 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 gate on peut continuer à travailler sans perte de données. Il suffit après de changer de changer le disque A détérioré par un nouvel disque C pour ne pas risquer de perdre les données définitivement au cas ou le seul disque restant s'endommage.

### 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<sup>1</sup>

### 2.4.4 Base de donnée

Les données des utilisateurs doivent êtres stockés dans une base de donnée. Une base de donnée permet de stocker, de structurer, de gérer et d'accéder aux données de façon sure, rapide et

---

<sup>1</sup>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

sécurisée. Un SGBD (Système de gestion de base de données) 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 (description ou 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 elles par des relations. On utilise le langage SQL pour interagir avec les données. Comme SGBD de ce type, on distingue MySQL, PostgreSQL, Oracle, et 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 NO SQL : ceux sont des bases de données qui n'utilisent pas le modèle relation. On peut citer à titre d'exemple Mongo

### 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 qui seront ensuite soit lues , ou compilées, ou 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** HTML est un langage de description. C'est un langage dérivé du XML et qui permet d'écrire une page web statique.

**PHP** PHP est un langage coté serveur, il va permettre de manipuler les données reçues par notre interface web (site web) 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 mon ordinateur dans 10 heures de temps. Il est essentiels 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 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 et lire les données (les mails)

### SMTP

Le serveur SMTP permet d'envoyer un mail. Énonçons son fonctionnement par un exemple. Si Toto veut envoyer un mail à Baké. Toto a pour adresse mail toto@toto.com et Baké 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 compte 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 de Toto

### 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 (exemple Mozilla Thunderbird, Yahoo mail ). Le protocole POP permet aussi de récupérer les mails mails. La différence majeure entre IMAP et POP est que IMAP est une connexion directe unidirectionnel entre le serveur le client mail tandis que POP est une connexion bidirectionnelle.

- Cela signifie que avec IMAP les données sont sur le serveur et on y accède directement et toute modification faite sur les mails depuis le client mail est réalisée sur le serveur. En IMAP si je supprime des mails depuis mon client mail, les mails sont aussi supprimés sur

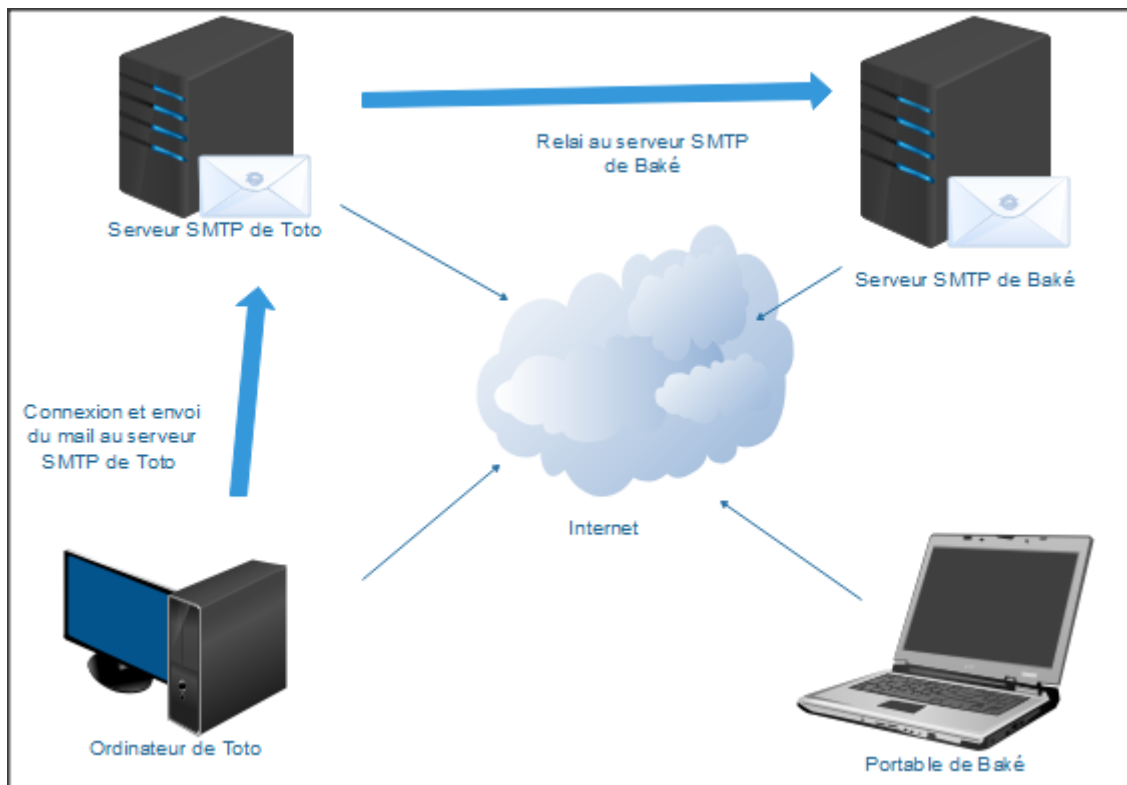


Figure 2.1: Principe du SMTP

le serveur. De même avec IMAP il est impossible d'avoir une copie locale 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 un protocole POP car si on crée plusieurs copies locales d'un mail qui est sur le serveur on peut facilement se tromper et écraser apres 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 enregistrer le brouillon de la réponse REPONSEMAIL1 sans avoir à envoyer la réponse. Je sors après sans mon pc et je réalise que je veux envoyer la réponse. Je me connecte à mon serveur mail avec mon portable et je ne retrouve par le brouillon car ce 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 est enregistre sur le serveur.

## 2.4.8 Sécurité

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



## **firewall**

Le firewall 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 ou d'empêcher les utilisateurs qui répondent. On distingue deux firewalls : les firewalls matériels et les firewalls logiciels. Le firewall est une protection pour la couche 3 et 4. 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. C'est lui qui va par exemple bloquer les étudiants de se connecter aux serveurs à 00h.

## **VPN**

## **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. Un serveur web pour envoyer les mails depuis une application web (un webmail), un site web d'administration pour créer des comptes email et réaliser quelques opérations d'administration. une base de données pour sauvegarder les informations d'authentification. Nous allons coder le site web en PHP. Les mails seront stockés sur un disque dur RAID, LVM. Ecrire des règles firewalls pour protéger le serveur. Configurer un anti spam pour empêcher les spammeurs. Sécuriser par le chiffrement la communication.

# Chapter 3

## Implémentation du projet

### 3.1 Monter le serveur Linux

Nous allons utiliser pour notre travail 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 qui correspond à une version d'Ubuntu qui sort tous les 2ans 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. Nous allons expliquer ce qu'est un hyperviseur plus loin dans ce document. Le choix de l'hyperviseur VMware est axé sur le fait qu'il supporte une meilleure intégration avec l'émulateur d'architectures réseaux GNS3<sup>2</sup>.

#### 3.1.1 Installation du serveur

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 la configuration suivante 1 Go de mémoire RAM, 40Go de stockage de masse (disque dur).

On démarre la machine virtuelle serveur avec le CD 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 des interfaces réseaux dans la machine virtuelle. On éteint la machine avec la commande

---

<sup>1</sup>Un hyperviseur est un logiciel qui fournit un environnement virtuel à pour installer un OS sans avoir un matériel réel. Nous y reviendrons plus tard

<sup>2</sup>Sera détaillé dans la suite

```
sudo shutdown now
```

Dans VMWare on édite les paramètres de la machine virtuelle et on lui ajoute deux adaptateur réseaux en veillant à bien cocher la case Custom et choisir VMnet2 pour le premier adaptateur et pour le second on choisit NAT. La NAT va permettre à notre machine d'avoir accès à internet. On démarre la machine. On tape la commande

```
ip a
```

## 3.2 Gestion du stockage de fichier

Dans ce chapitre nous allons configurer un disque dur spécial pour les données que le serveur va contenir

### 3.2.1 Systeme de fichier

Toutes les données contenues dans un ordinateur sont du binaire( 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 dire reconnaître une image par son extension jpeg ou un document texte par docx. Les fichiers sont stockés 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 le 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 information 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 paramètre de la machine virtuelle ensuite on ajoute trois disques de 2 giga configurés en SCSI<sup>3</sup>

On redemarre la machine ensuite on exécute les instructions suivante : On our affiche la liste des disques durs.

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

---

<sup>3</sup>est une spécification matérielle des disques

Ensuite on verifie la taille de chaque disque avec la commande

```
Console
gdisk dev/sdb
```

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 cree 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 details sur le nous 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 modification soit effective au prochain démarrage 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 partiton 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 à chaque démarrage du système. On édite le fichier `/etc/fstab`

Console

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

## 3.3 Configurer le LAMP

On va installer PHP MariaDB qui est un fork<sup>4</sup> de MySQL

---

<sup>4</sup>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

```
sudo su
apt-get update #Pour mettre à jour le cache local
apt-get install apache2 libapache2-mod-php php-fpm mysql-server
```

## 3.4 Modélisation de l'architecture reseau avec GNS3

On télécharger les logiciels GNS3 et GNS3 VM sur son site officiel. 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 sélectionnez "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 fenetre apparait 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

Nous avons opté pour un modèle générique le réseau est séparé en deux grandes parties le réseau de ENEAM existant relié par le routeur routeurEneam et 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 em1 qui le relie à routeurMail et em2 pour la NAT routeurMail de même trois interfaces dns1 est une docker qui contient un petit serveur DNS qui va servir à la résolution des noms de domaines au sein du réseau local On crée un nouveau projet "projetmail" routeurMail et routeurEneam sont des routeur Pfsense<sup>5</sup>

---

<sup>5</sup>Est un routeur parefeu open source

Postfix est un serveur SMTP. C'est donc lui qui va servir à l'envoi des mails Prenons le poste PC1 du réseau em1 s'il veut envoyer un mail à jean@eneam.da depuis son poste il passe par internet. Il n'a pas communication directe entre les deux routeurs pour des utilisateurs locaux. Il se connecte au serveur mail

### 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 #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 modification on modifie le fichier 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 en local 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<sup>6</sup>
- la propriété virtual\_mailbox\_domains = mysql:/etc/postfix/sql/mysql-virtual-mailbox-domains.cf définit le fichier qui contient la directive de connexion à notre base de données mysql plus précisément MariaDB. virtual\_mailbox\_maps = mysql:/etc/postfix/sql/mysql-virtual-mailbox-maps.cf.
- 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 de la machine. 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 = isidore
hosts = 127.0.0.1
```

---

<sup>6</sup>Un certificat autosigné est un certificat qui n'est pas créé par une autorité de certification

```
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
lmtpl     unix  -      -      y      -      -      lmtpl
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.

### 3.5.1 Création de la base de donnée des tables

## 3.6 Installation de Dovecot

Dovecot est un serveur IMAP et POP

```
Console
apt-get install dovecot-core dovecot-imapd dovecot-mysql
```

Dans le dossier de configuration `/etc/dovecot/` de dovecot on a : 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 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_listener /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

## 3.8 Le site d'administration

Le site d'administration va permettre de créer et de supprimer les comptes email, de voir la liste de quelques services (SMTP, IMAP , APACHE NGINX ), leur état de les arrêter ou éteindre. Il a aussi la fonctionnalité qui va permettre à l'administrateur de voir les mails des étudiants qui n'est en cours de développement. Il existe des solutions gratuites pour l'administration 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 prouve qu'on apprend pas pour rien et qu'on arrive à combiner toutes les technologies 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.

L'administrateur du système va exécuter des instructions depuis l'interface web et pour communiquer avec la machine l'interface web va appeler des script bash. Nous avons un script bash pour créer le répertoire d'un utilisateur , un autre pour supprimer lors de la suppression d'un contre 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

```
#!/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"
#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
    else
        exit 0
    fi
fi
done
fi

```

### 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 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 tentative de connexion échoué ou autre paramètre. Si ça dépasse un seuil il bloque les connexions du client au serveur

Console

```
apt-get install 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 regles 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 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
```

# Conclusion