



RÉPUBLIQUE DE MADAGASCAR
FITIAVANA – TANINDRAZANA – FANDROSOANA
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET
DE LA RECHERCHE SCIENTIFIQUE



Mention STIC

Rapport de Mini-Projet

Installation et Configuration Automatisée d'un
Serveur Mail sur Debian avec VirtualBox et Docker

Réalisé par :

KOURAICHI Ali
ANDRY Nizwami Ibrahim
TILAHIZAFY Judicael Roberto
ANDRIAMANDIGNY Ali Rueff Suharto

Encadré par :

RANDRIAMASINORO Njakarison Menja

Année Universitaire 2024-2025

Table des matières

1	Introduction	2
1.1	Objectifs du Projet	2
1.2	Portée du Projet	2
2	Configuration de l'Environnement	2
2.1	Installation de VirtualBox et Debian	2
2.2	Configuration de la Machine Virtuelle	2
3	Installation et Configuration du Serveur Mail	3
3.1	Configuration de Postfix	3
3.1.1	Test d'envoi d'email	3
3.1.2	Installation de mailUtils	3
3.1.3	Envoi d'un e-mail	3
3.2	Configuration de Dovecot	4
3.3	Thunderbird comme Client de Messagerie	4
4	Configuration de Sécurité	5
4.1	Implémentation SSL/TLS	5
5	Automatisation avec Script Bash	5
6	Tests et Validation	6
6.1	Fonctionnalité de Messagerie	6
6.2	Validation de Sécurité	6
7	Conteneurisation Docker	6
8	Documentation et Dépôt GitHub	7
9	Conclusion	7
10	Références	7

1 Introduction

1.1 Objectifs du Projet

L'objectif de ce projet était d'établir un serveur de messagerie entièrement fonctionnel sur un système Debian dans une machine virtuelle VirtualBox, en utilisant Postfix comme Agent de Transport de Courrier (MTA), Dovecot pour les protocoles IMAP et POP3, et Thunderbird comme client de messagerie à la place de Roundcube. Les processus d'installation et de configuration ont été automatisés à l'aide d'un script Bash, et le système a été conteneurisé avec Docker pour assurer la portabilité. Le projet visait également à implémenter des mesures de sécurité robustes, incluant les certificats SSL/TLS, SPF, DKIM, et DMARC, et à valider la configuration par des tests complets.

1.2 Portée du Projet

Ce rapport détaille la création de l'environnement virtuel, l'installation et la configuration des composants du serveur de messagerie, l'automatisation de l'installation, les configurations de sécurité, les tests, et la conteneurisation Docker. Il aborde également les défis rencontrés durant le projet et les leçons apprises. Les livrables incluent un dépôt GitHub, un script Bash, un Dockerfile, et ce rapport LaTeX.

2 Configuration de l'Environnement

2.1 Installation de VirtualBox et Debian

Nous avons installé Oracle VirtualBox 7.0 sur une machine hôte exécutant Ubuntu 24.04. Une machine virtuelle a été créée avec 2 Go de RAM, 20 Go de stockage, et un adaptateur réseau en mode pont pour permettre l'accès externe. Debian 12 (Bookworm) a été installé en utilisant l'ISO officiel depuis <https://www.debian.org/doc/>.

Défi Rencontré : Durant l'installation de VirtualBox, nous avons rencontré des problèmes de compatibilité avec la version du noyau du système hôte, ce qui a nécessité la mise à jour de VirtualBox vers la dernière version et l'installation du pack d'extensions approprié. De plus, l'installation initiale de Debian a échoué à cause d'une configuration réseau incorrecte, nous obligeant à passer du mode NAT à un adaptateur en mode pont.

2.2 Configuration de la Machine Virtuelle

Le système Debian a été configuré avec une adresse IP statique (192.168.1.100) et le nom d'hôte `mailserver.local`. Nous avons mis à jour le système en utilisant :

```
1 sudo apt update && sudo apt upgrade -y
```

Des outils essentiels comme `vim`, `net-tools`, et `curl` ont été installés pour faciliter les configurations ultérieures.

3 Installation et Configuration du Serveur Mail

3.1 Configuration de Postfix

Postfix a été installé comme MTA en utilisant :

```
1 sudo apt install postfix -y
```

Nous avons configuré Postfix pour fonctionner comme un site Internet, définissant le domaine à `mailserver.local`. Le fichier de configuration principal `/etc/postfix/main.cf` a été modifié pour inclure :

```
1 myhostname = mailserver.local
2 mydomain = mailserver.local
3 mydestination = $myhostname, localhost.$mydomain, localhost
4 inet_interfaces = all
5 inet_protocols = ipv4
```

3.1.1 Test d'envoi d'email

Creation d'utilisateur

Pour créer un utilisateur pour le test, nous avons utilisé les commandes suivantes :

```
1 sudo adduser Ibra
2 sudo passwd Ibra
```

3.1.2 Installation de mailUtils

Avant de pouvoir envoyer des e-mails en ligne de commande (ex. pour les tests SMTP avec `mail`, les scripts système ou la supervision), il est essentiel d'installer le paquet `mailutils`, qui fournit la commande `mail` et ses dépendances.

```
1 sudo apt update
2 sudo apt install mailutils -y
```

`Mailutils` est un ensemble d'utilitaires robustes permettant de composer, envoyer, recevoir et lire des e-mails directement depuis le terminal. Indispensable pour tester rapidement la configuration de Postfix ou automatiser l'envoi d'e-mails système.

3.1.3 Envoi d'un e-mail

Pour envoyer un e-mail à l'utilisateur `Ibra` sur le serveur de messagerie, utilisez la commande `mail` fournie par le paquet `mailutils`. Exécutez la commande suivante :

```
1 echo "Corps_du_mail_de_test" | mail -s "Sujet_du_test"
   Ibra@mailserver.local
```

L'e-mail sera livré à la boîte de réception de l'utilisateur `Ibra`, généralement située dans le répertoire `/home/ibra/Maildir/new`. Pour vérifier la bonne réception du message, exécutez :

```
1 ls -l /home/ibra/Maildir/new
```

Vous pouvez également consulter les journaux du serveur de messagerie pour confirmer la livraison :

```
1 tail -f /var/log/mail.log
```

Défi Rencontré : La configuration initiale de Postfix causait le rejet des emails à cause d'un paramètre `mydestination` mal configuré, qui incluait un domaine incorrect. Cela a été résolu en consultant la documentation Postfix et en vérifiant les paramètres de domaine.

3.2 Configuration de Dovecot

Dovecot a été installé pour gérer les protocoles IMAP et POP3 :

```
1 sudo apt install dovecot-core dovecot-imapd dovecot-pop3d -y
```

Le fichier de configuration `/etc/dovecot/dovecot.conf` a été mis à jour pour activer IMAP et POP3 :

```
1 protocols = imap pop3
```

L'authentification des utilisateurs a été configurée en utilisant la base de données utilisateurs du système, et les boîtes aux lettres ont été configurées pour utiliser le format Maildir.

Défi Rencontré : Dovecot a initialement échoué à démarrer à cause d'un problème de permissions sur le répertoire des boîtes aux lettres. Nous avons résolu cela en ajustant les permissions en utilisant :

```
1 sudo chmod -R 700 /var/mail
```

3.3 Thunderbird comme Client de Messagerie

Thunderbird a été choisi comme client de messagerie à la place de Roundcube pour tester le serveur de messagerie localement. Il a été installé sur la machine hôte :

```
1 sudo apt install thunderbird -y
```

Thunderbird a été configuré pour se connecter au serveur de messagerie en utilisant IMAP (port 143) et SMTP (port 25) avec l'adresse serveur `192.168.1.100`. Les comptes utilisateurs ont été configurés avec des identifiants correspondant aux utilisateurs du système Debian.

Défi Rencontré : Thunderbird a échoué à se connecter au serveur à cause d'un pare-feu bloquant les ports 143 et 25. Nous avons ouvert ces ports en utilisant :

```
1 sudo ufw allow 143
2 sudo ufw allow 25
```

4 Configuration de Sécurité

4.1 Implémentation SSL/TLS

Pour sécuriser les communications, nous avons généré un certificat SSL/TLS auto-signé en utilisant OpenSSL :

```
1 sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \  
2 -keyout /etc/ssl/private/dovecot.pem \  
3 -out /etc/ssl/certs/dovecot.pem
```

Postfix et Dovecot ont été configurés pour utiliser ces certificats en mettant à jour leurs fichiers de configuration respectifs.

5 Automatisation avec Script Bash

Un script Bash a été développé pour automatiser l'installation et la configuration de Postfix, Dovecot, et des paramètres de sécurité. Voici le script :

```
1 #!/bin/bash  
2 # Script d'Installation Automatisée du Serveur Mail  
3  
4 # Mise à jour du système  
5 sudo apt update && sudo apt upgrade -y  
6  
7 # Installation de Postfix  
8 sudo DEBIAN_FRONTEND=noninteractive apt install postfix -y  
9 sudo postconf -e "myhostname=localhost"  
10 sudo postconf -e "mydestination=localhost.\n\n$mydomain,localhost"  
11 sudo postconf -e "inet_interfaces=all"  
12  
13 # Installation de Dovecot  
14 sudo apt install dovecot-core dovecot-imapd dovecot-pop3d -y  
15 sudo sed -i 's/#protocols=imappop3/protocols=imappop3/' /  
16     etc/dovecot/dovecot.conf  
17  
18 # Installation d'OpenSSL et génération des certificats  
19 sudo apt install openssl -y  
20 sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \  
21 -keyout /etc/ssl/private/dovecot.pem \  
22 -out /etc/ssl/certs/dovecot.pem \  
23 -subj "/CN=mailserver.local"  
24  
25 # Redémarrage des services  
26 sudo systemctl restart postfix dovecot  
27  
28 echo "Installation du serveur mail terminée."
```

Défi Rencontré : Le script a initialement échoué à gérer les erreurs quand une installation de paquet était interrompue à cause de problèmes réseau. Nous avons ajouté une vérification d'erreurs et une logique de nouvelle tentative pour assurer la robustesse.

6 Tests et Validation

6.1 Fonctionnalité de Messagerie

Nous avons testé l'envoi et la réception d'emails en utilisant Thunderbird. Les emails ont été envoyés avec succès de `user1@mailserver.local` vers `user2@mailserver.local` et des domaines externes. L'accès IMAP et POP3 a été vérifié en récupérant les emails dans Thunderbird.

6.2 Validation de Sécurité

Les configurations SPF, DKIM, et DMARC ont été validées en utilisant des outils en ligne comme <https://mxtoolbox.com>. La configuration SSL/TLS a été testée en se connectant au serveur avec `openssl s_client -connect 192.168.1.100:993`.

Défi Rencontré : La validation DMARC a échoué à cause d'une syntaxe incorrecte dans l'enregistrement TXT. Cela a été corrigé en consultant la documentation DMARC et en retestant.

7 Conteneurisation Docker

Un Dockerfile a été créé pour conteneuriser le serveur de messagerie :

```
1 FROM debian:bookworm
2
3 # Installation des paquets n cessaires
4 RUN apt update && apt install -y \
5     postfix \
6     dovecot-core \
7     dovecot-imapd \
8     dovecot-pop3d \
9     openssl \
10    opendkim \
11    opendkim-tools \
12    && rm -rf /var/lib/apt/lists/*
13
14 # Copie du script de configuration
15 COPY mailserver_setup.sh /mailserver_setup.sh
16 RUN chmod +x /mailserver_setup.sh
17
18 # Cr ation des utilisateurs n cessaires
19 RUN groupadd -r mail && useradd -r -g mail mail
20
21 # Exposition des ports
22 EXPOSE 25 143 587 993
```

```
23  
24 # Commande de démarrage  
25 CMD ["/mailserver_setup.sh"]
```

L'image a été construite et testée en utilisant :

```
1 docker build -t mailserver:latest .  
2 docker run -d -p 25:25 -p 143:143 -p 587:587 -p 993:993  
   mailserver:latest
```

Défi Rencontré : Le conteneur Docker a échoué à démarrer Postfix à cause de permissions utilisateur système manquantes. Nous avons modifié le Dockerfile pour inclure une étape de configuration utilisateur.

8 Documentation et Dépôt GitHub

Le projet a été documenté dans un dépôt GitHub nommé "Serveur-Mail-sur-Debian" à l'adresse <https://github.com/username/Serveur-Mail-sur-Debian>. Tous les membres de l'équipe ont créé des comptes GitHub et ont reçu l'accès, ainsi que le superviseur (njakarison@gmail.com). Le dépôt inclut le script Bash, le Dockerfile, et les fichiers de configuration.

9 Conclusion

Le projet a réussi à établir un serveur de messagerie sécurisé et automatisé sur Debian en utilisant Postfix, Dovecot, et Thunderbird. Malgré les défis tels que les problèmes de configuration réseau, les erreurs de permissions, et les problèmes de configuration DKIM, l'équipe a résolu ces problèmes par un débogage systématique et la consultation de la documentation. Le script d'automatisation et l'image Docker assurent un déploiement facile, rendant le système portable et évolutif.

10 Références

- Documentation Debian : <https://www.debian.org/doc/>
- Documentation Postfix : <http://www.postfix.org/documentation.html>
- Documentation Dovecot : <https://doc.dovecot.org/>
- Documentation Docker : <https://docs.docker.com/>
- Guide de Script Bash : <https://linuxcommand.org/tlcl.php>
- Tutoriel Git : <https://www.hostinger.fr/tutoriels/tuto-git>