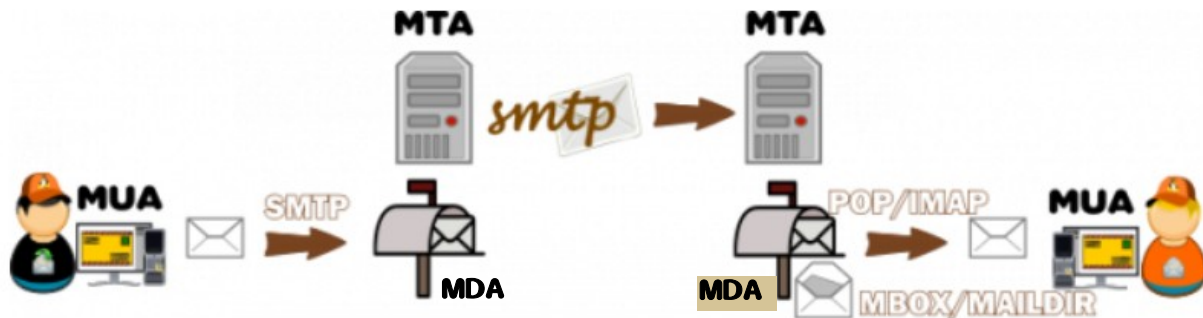


A. PRINCIPE DE FONCTIONNEMENT DE L'ACHEMINEMENT DES EMAILS



1. L'utilisateur rédige son courriel sur son client de messagerie informatique le MUA (Mail User Agent).
2. Le client MUA envoie, en tant que client, par le protocole SMTP, le courriel dans la boîte aux lettres des départs de courriels du serveur de distribution, le MTA (Mail Transfer Agent).
3. Le serveur MTA de départs des courriels expédie le courriel, par le protocole SMTP et en tant que client, vers le serveur destinataire MTA d'arrivée des courriels pour distribution locale.
4. Alors le serveur d'arrivées des courriels trie, relaie, et distribue dans la boîte aux lettres locale de l'utilisateur le courriel.
5. Puis avec le client de messagerie, le MUA, l'utilisateur destinataire, utilise ce courriel dans sa boîte aux lettres personnelle du serveur. Soit directement dans la boîte aux lettres du serveur MTA en manipulant les fichiers du courriel au format Mbox ou Maildir de celui-ci. Soit, s'il est distant de la boîte aux lettres du serveur MTA, le récupère localement sur son poste de travail avec le protocole POP ou IMAP, c'est un serveur MDA (Mail Delivery Agent).
6. Le client de messagerie MUA trie ce courriel ou l'ouvre pour afficher le contenu du courriel sur l'écran de l'ordinateur de l'utilisateur destinataire.

Les courrielleurs Web permettent aux utilisateurs de consulter leurs courriels depuis leur navigateur web, depuis n'importe où dans le monde. Vous aurez besoin d'avoir configuré un MTA, un MDA et un serveur web (p. ex. [Apache2](#)) pour que le courrielleur Web fonctionne.

- SquirrelMail
- OpenWebMail
- [Roundcube](#)

Un serveur de messagerie doit aussi contenir d'autres fonctions essentielles comme un antispam ou un antivirus.

Récapitulatif :

1. **Envoi de l'Email :** Ordinateur de l'expéditeur → Serveur SMTP (MTA)
2. **Acheminement du L'Email:** Serveur SMTP(MTA) expéditeur → Serveurs SMTP(MTA) destinataire → Serveur POP/IMAP (MDA)
3. **Récupération d'Email par le destinataire (à l'aide de son courrielleur) :** Serveur POP/IMAP (MDA) → Ordinateur du destinataire

source : https://doc.ubuntu-fr.org/comment_fonctionne_le_courriel_sous_linux
https://doc.ubuntu-fr.org/serveurs_messagerie_01.12.23

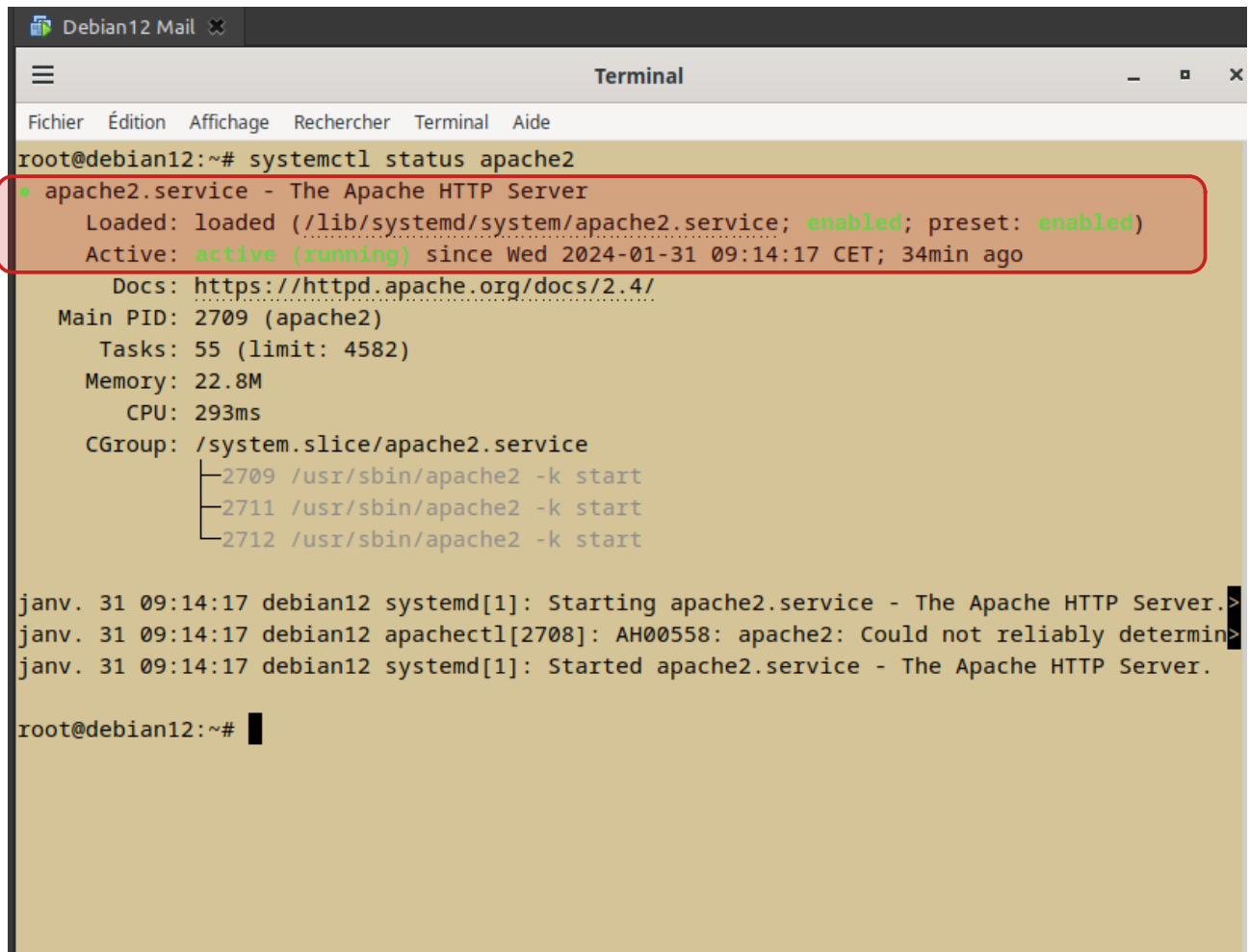
B. INSTALLATION DU SERVEUR WEB ET DES DÉPENDANCES

1 Installation Apache2

apt install apache2

2 Vérifier le fonctionnement de apache2

systemctl status apache2



```
Debian12 Mail x
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
root@debian12:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
  Active: active (running) since Wed 2024-01-31 09:14:17 CET; 34min ago
    Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 2709 (apache2)
   Tasks: 55 (limit: 4582)
  Memory: 22.8M
     CPU: 293ms
   CGroup: /system.slice/apache2.service
           └─2709 /usr/sbin/apache2 -k start
             └─2711 /usr/sbin/apache2 -k start
               └─2712 /usr/sbin/apache2 -k start

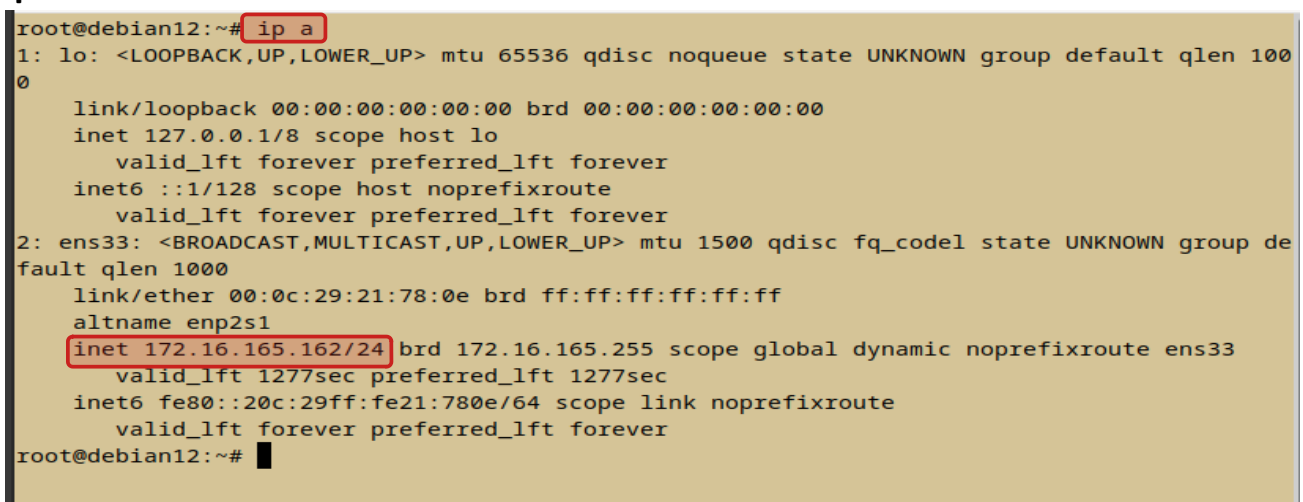
janv. 31 09:14:17 debian12 systemd[1]: Starting apache2.service - The Apache HTTP Server.
janv. 31 09:14:17 debian12 apachectl[2708]: AH00558: apache2: Could not reliably determine
janv. 31 09:14:17 debian12 systemd[1]: Started apache2.service - The Apache HTTP Server.

root@debian12:~#
```

Le serveur est bien démarré et en cours de fonctionnement. Nous allons maintenant vérifier que notre serveur apache2 est correctement installé en tapant son IP dans un navigateur. La page par défaut d'apache2 devrait s'afficher.

Récupération IP de la machine :

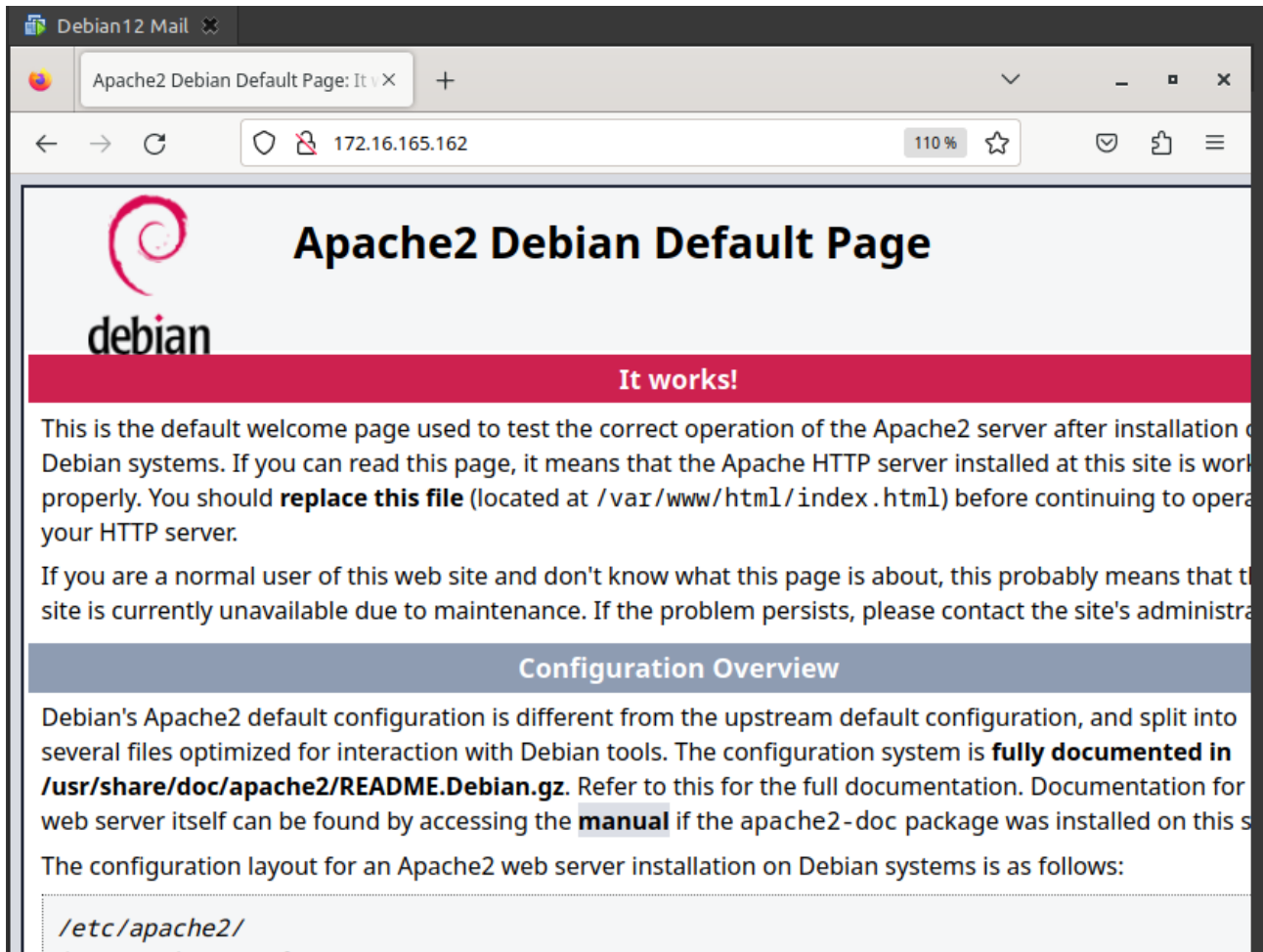
ip a



```
root@debian12:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
   link/ether 00:0c:29:21:78:0e brd ff:ff:ff:ff:ff:ff
   altname enp2s1
   inet 172.16.165.162/24 brd 172.16.165.255 scope global dynamic noprefixroute ens33
       valid_lft 1277sec preferred_lft 1277sec
   inet6 fe80::20c:29ff:fe21:780e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

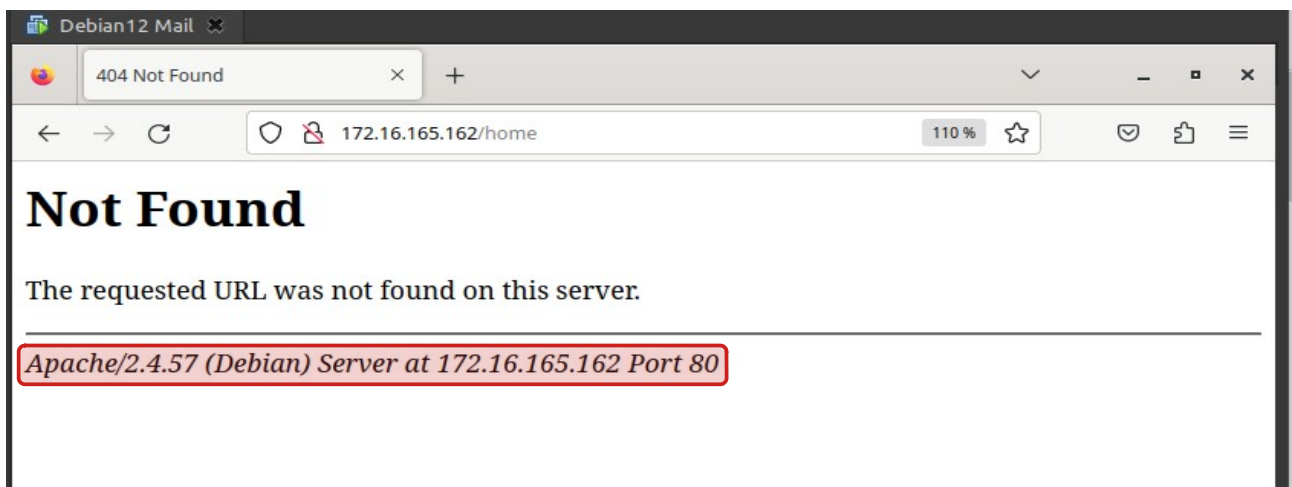
root@debian12:~#
```

Ci-dessous la page par défaut d'Apache2 s'affiche correctement :



3 Sécurisation 1

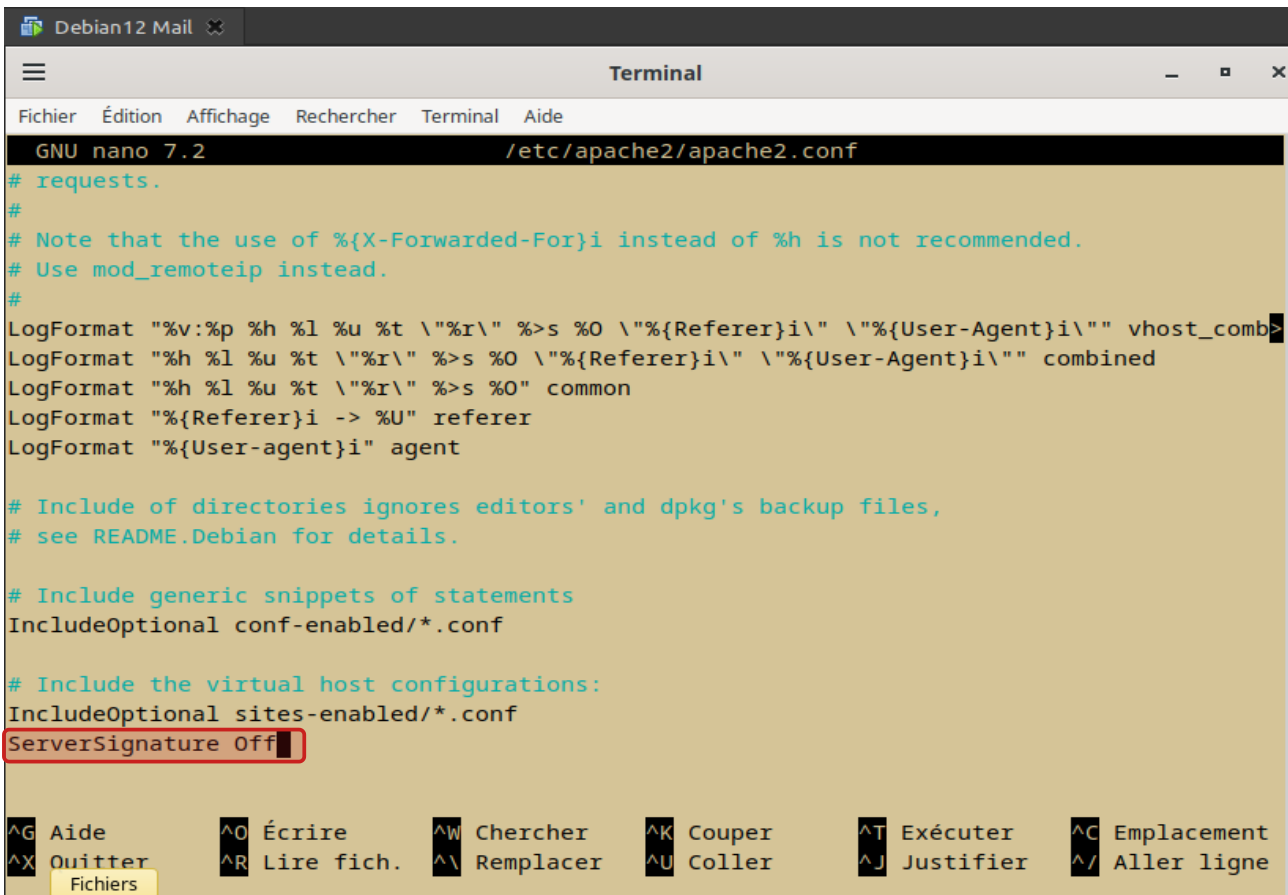
Par défaut lorsque l'on demande d'afficher une page et qu'elle n'existe pas le message d'erreur affiche la version d'Apache2.



Le numéro de version pourrait être utilisé pour éventuellement rechercher des vulnérabilités connues de celle-ci. On va donc faire en sorte que cette information ne soit plus affichée.

Ouvrir le fichier de configuration d'Apache2 :
nano /etc/apache2/apache2.conf

Rajouter cette ligne à la fin du fichier apache2.conf :
ServerSignature Off



```
Debian12 Mail x
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
GNU nano 7.2 /etc/apache2/apache2.conf
# requests.
#
# Note that the use of %{X-Forwarded-For}i instead of %h is not recommended.
# Use mod_remoteip instead.
#
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_comb>
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

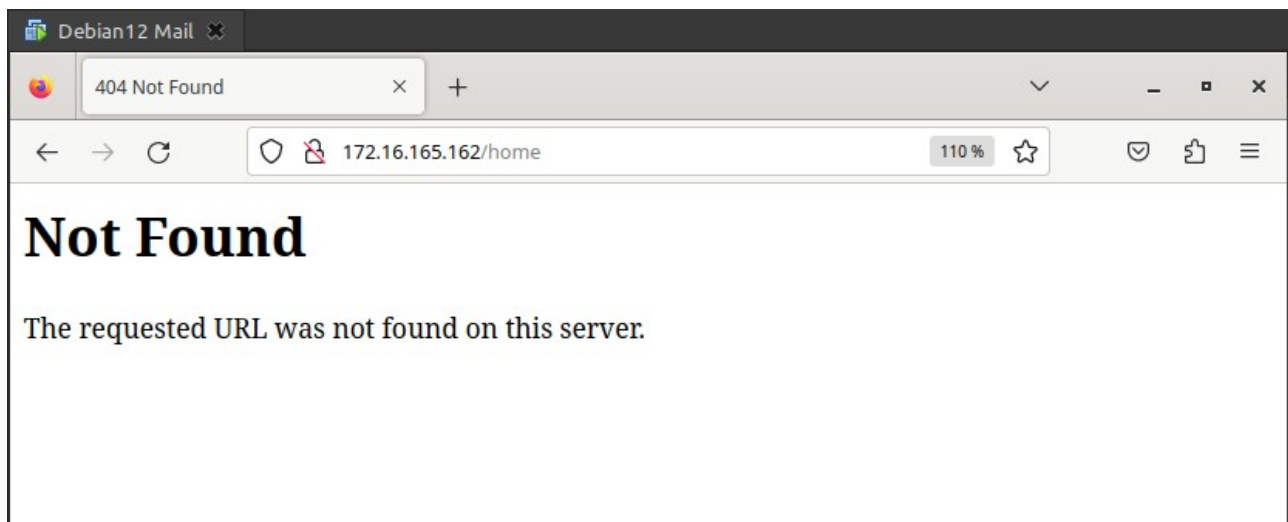
# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf
ServerSignature Off

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^V Remplacer  ^U Coller   ^J Justifier ^/ Aller ligne
Fichiers
```

Redémarrer Apache2 :
service apache2 restart

Maintenant en actualisant la page de notre navigateur, les informations de version et du service utilisé n'apparaissent plus en cas de page introuvable, comme on peut le voir sur l'image ci-dessous :

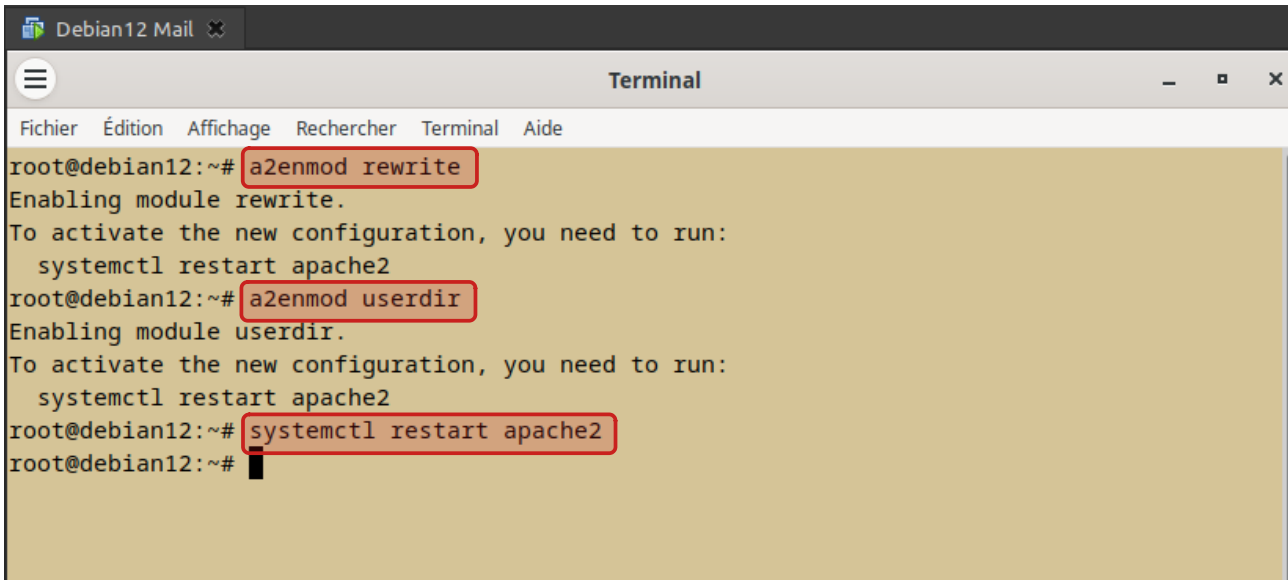


4 Installation de modules complémentaires pour Apache2

Activation de la réécriture d'URL(URL rewriting) pour avoir des adresses plus lisibles :
a2enmod rewrite

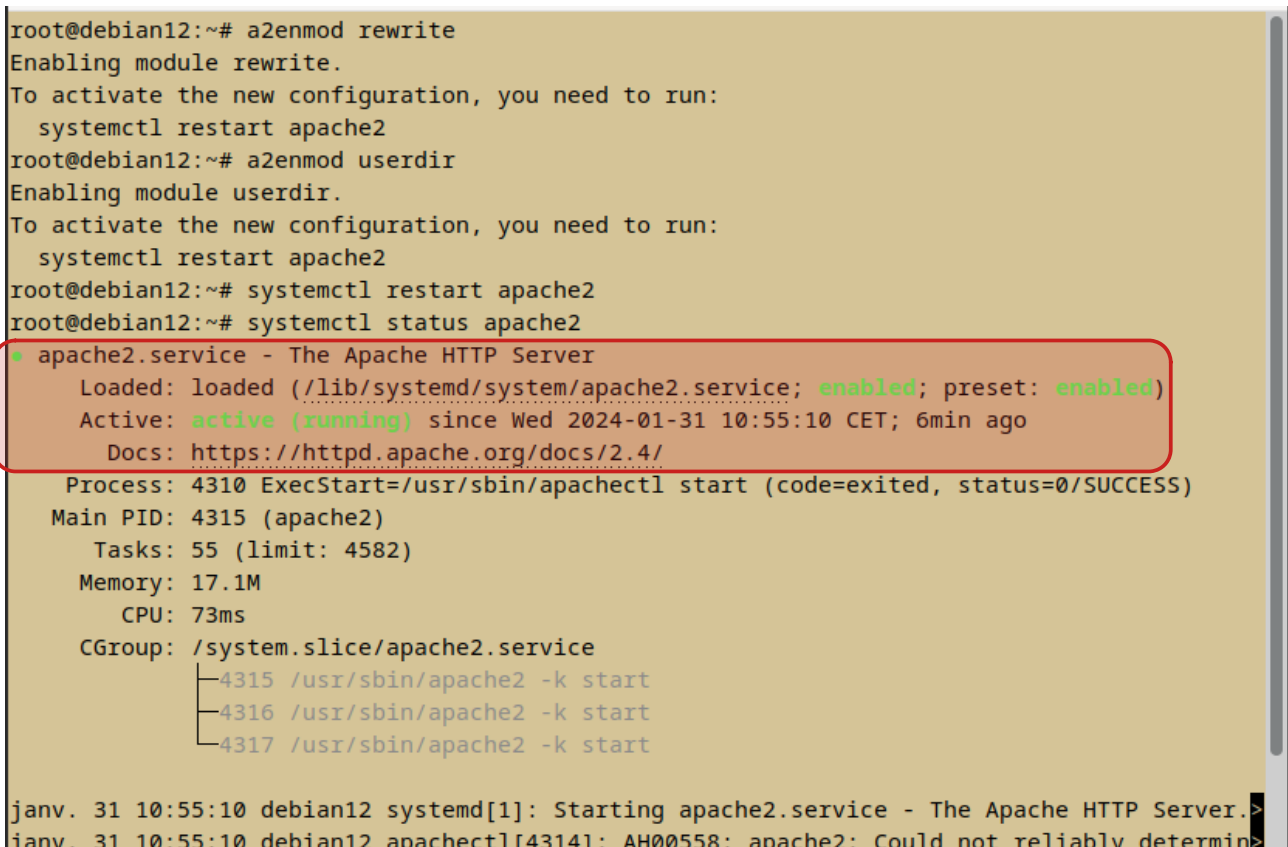
Activation du module Userdir pour assigner 1 utilisateur par site web :
a2enmod userdir

Redémarrer Apache2 :
systemctl restart apache2



```
Debian12 Mail x Terminal
Fichier Édition Affichage Rechercher Terminal Aide
root@debian12:~# a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian12:~# a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian12:~# systemctl restart apache2
root@debian12:~#
```

Vérifier que le service apache2 fonctionne correctement :
systemctl status apache2



```
root@debian12:~# a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian12:~# a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian12:~# systemctl restart apache2
root@debian12:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-01-31 10:55:10 CET; 6min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 4310 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 4315 (apache2)
    Tasks: 55 (limit: 4582)
   Memory: 17.1M
      CPU: 73ms
   CGroup: /system.slice/apache2.service
           └─4315 /usr/sbin/apache2 -k start
             └─4316 /usr/sbin/apache2 -k start
               └─4317 /usr/sbin/apache2 -k start

janv. 31 10:55:10 debian12 systemd[1]: Starting apache2.service - The Apache HTTP Server.>
ianv. 31 10:55:10 debian12 apachectl[43141]: AH00558: apache2: Could not reliably determin
```

Installation de curl (permet de récupérer du contenu de pages web via la ligne de commande) :
apt install curl

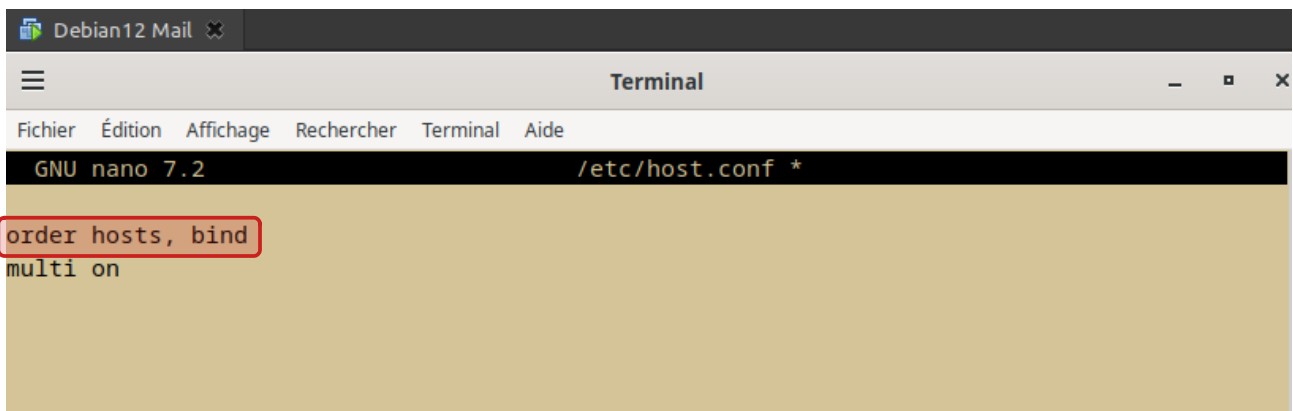
5 Résolution de noms

Nous allons configurer le fichier **host.conf** pour que la résolution des noms se fasse **d'abord** via ce fichier, **sinon** via le **server DNS**, si la résolution n'a pas pu être effectuée.

Ainsi les requêtes seront traitées plus rapidement en local, du moment que le nombre de machines est relativement limité et que les IP varient peu.

Ouvrir le fichier host.conf :
nano /etc/host.conf

Donner la priorité au fichier host.conf pour la résolution des noms :
order hosts, bind



```
Debian12 Mail x Terminal
Fichier Édition Affichage Rechercher Terminal Aide
GNU nano 7.2 /etc/host.conf *
order hosts, bind
multi on
```

5.1 Donner la priorité au fichier « host.conf »

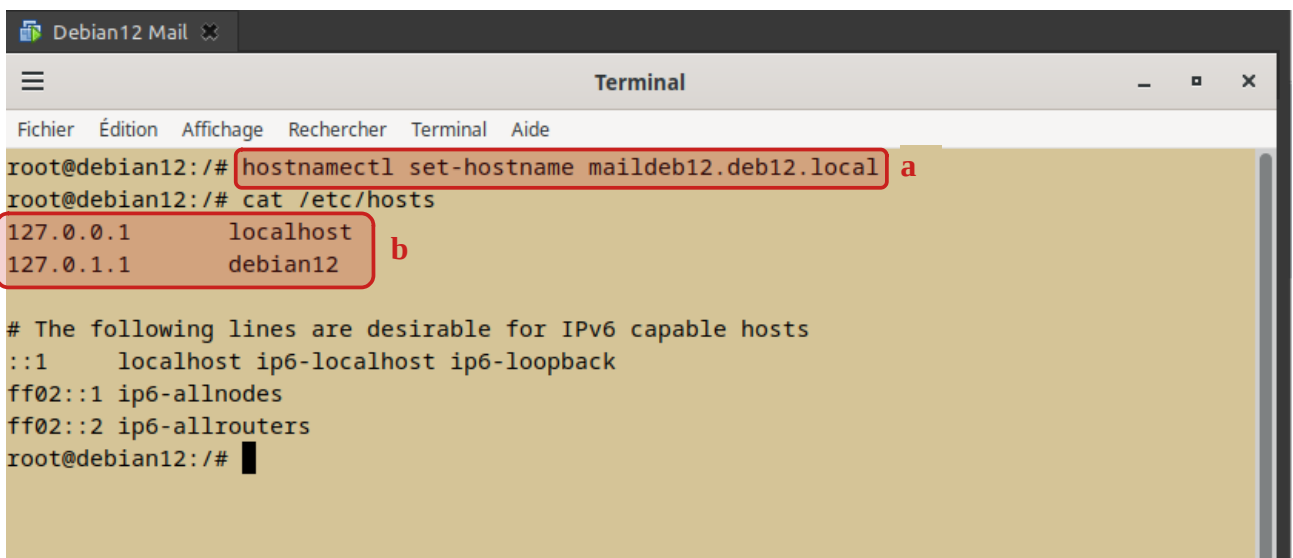
Modifier le FQDN :

Le FQDN est l'association du nom d'hôte avec le nom de domaine. Dans notre exemple :

nom d'hôte  **maildeb12 . deb12.local**  **nom de domaine**

a) Il faut donc nommer notre machine et indiquer le nom de domaine auquel elle est rattachée :
hostnamectl set-hostname maildeb12.deb12.local

b) En affichant le contenu de notre fichier hosts on voit que notre machine n'est pas reconnue avec son IP et son FQDN.



```
Debian12 Mail x Terminal
Fichier Édition Affichage Rechercher Terminal Aide
root@debian12:/# hostnamectl set-hostname maildeb12.deb12.local a
root@debian12:/# cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 debian12 b

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
root@debian12:/#
```

Editer le fichier hosts avec l'IP et FQDN de notre serveur :

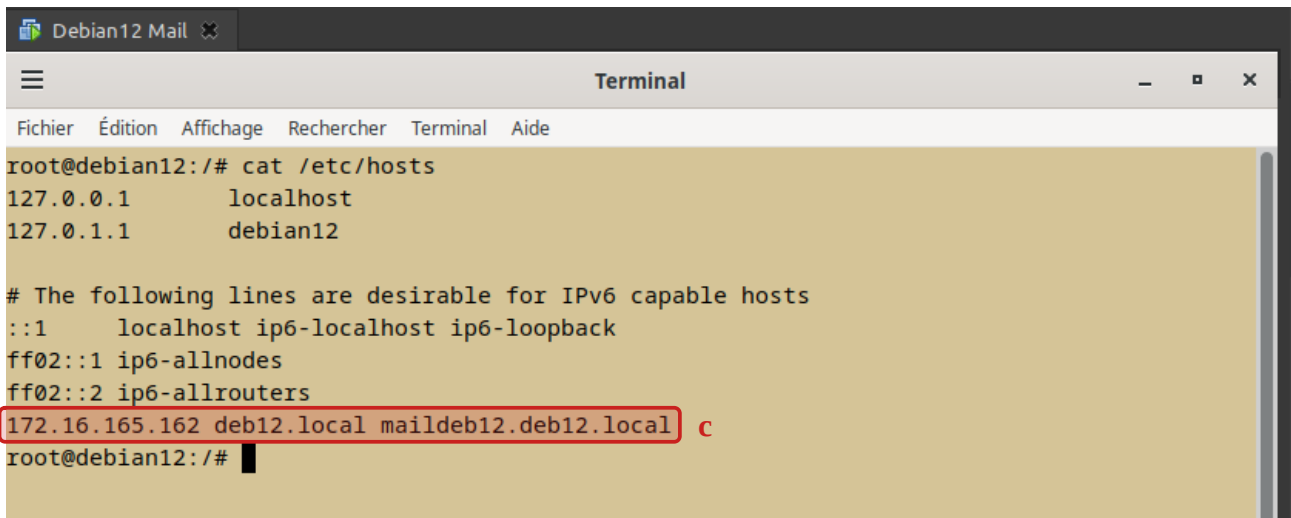
ip a → pour récupérer IP

c) Taper cette commande pour afficher les éléments entre cotes(chaine de caractères) et les rediriger à la fin du fichier hosts :

echo '172.16.165.162 deb12.local maildeb12.deb12.local' >> /etc/hosts

On vérifie avec **cat ro/etc/hosts** que le fichier a été modifié correctement.

Les informations de la commande précédente sont bien écrites à la fin du fichier hosts.



```
Debian12 Mail x
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
root@debian12:/# cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      debian12

# The following lines are desirable for IPv6 capable hosts
::1           localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
172.16.165.162 deb12.local maildeb12.deb12.local c
root@debian12:/#
```

A noter que l'opérateur de redirection « >> » rajoute la sortie de la commande le précédent, à la fin du fichier de destination, sans écraser celui-ci.

Redémarrer le serveur :

systemctl reboot

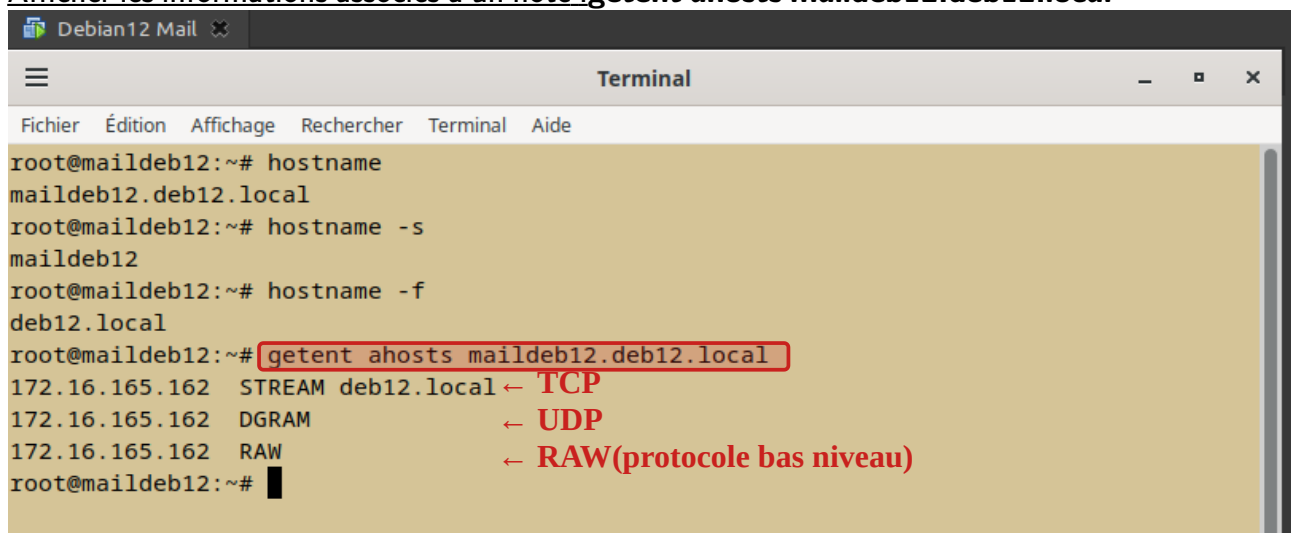
Vérifier la configuration :

hostname → 1. affiche le FQDN

hostname -s → 2. affiche le nom d'hôte

hostname -f → 3. affiche le nom de domaine

Afficher les informations associés à un hôte :getent ahosts maildeb12.deb12.local



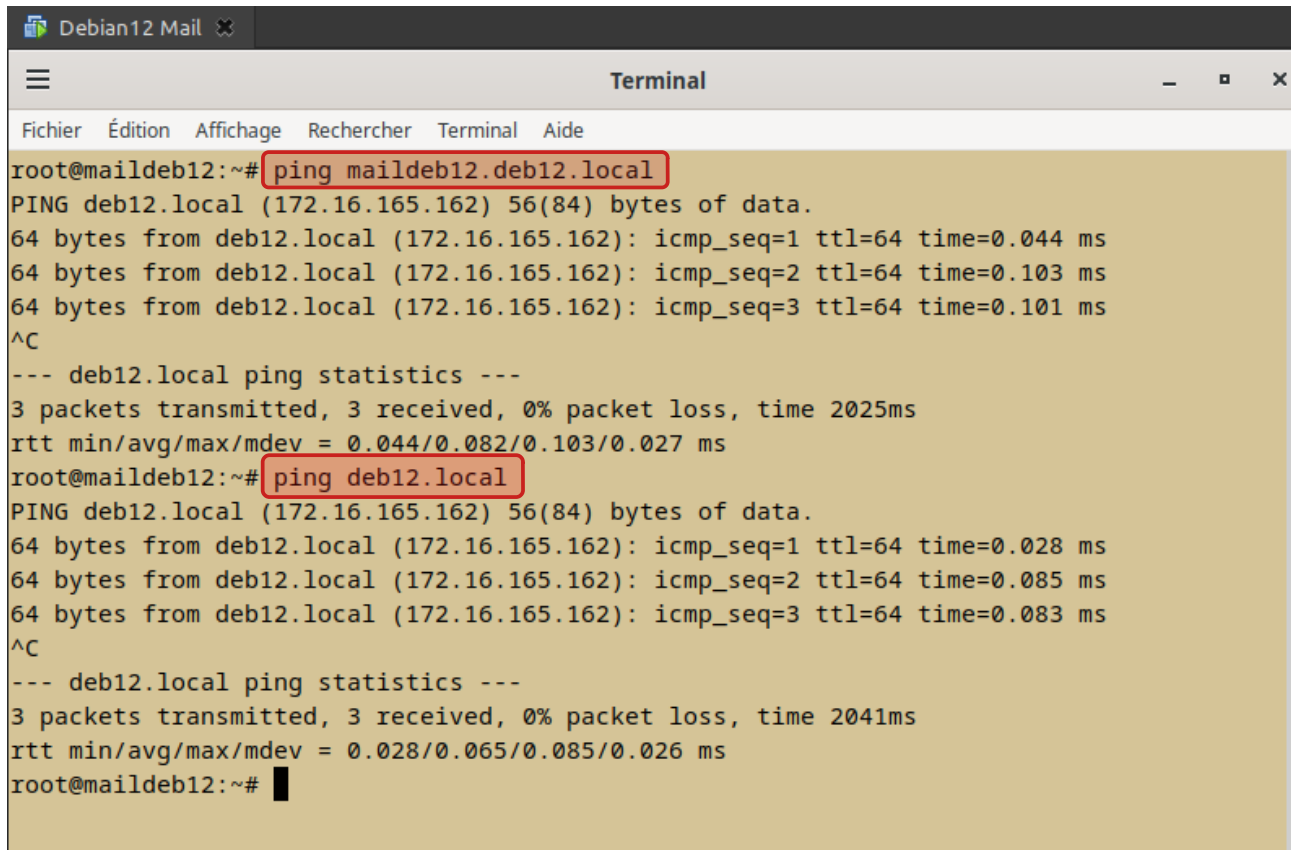
```
Debian12 Mail x
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
root@maildeb12:~# hostname
1 maildeb12.deb12.local
root@maildeb12:~# hostname -s
2 maildeb12
root@maildeb12:~# hostname -f
3 deb12.local
root@maildeb12:~# getent ahosts maildeb12.deb12.local
172.16.165.162  STREAM deb12.local ← TCP
172.16.165.162  DGRAM      ← UDP
172.16.165.162  RAW        ← RAW(protocol bas niveau)
root@maildeb12:~#
```

L'adresse IP sur chaque ligne confirme que notre nom d'hôte est résolu avec succès.

Faire le test avec un Ping sur le FQDN et le nom de domaine :

ping maildeb12.deb12.local

ping deb12.local



```
Debian12 Mail x
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
root@maildeb12:~# ping maildeb12.deb12.local
PING deb12.local (172.16.165.162) 56(84) bytes of data.
64 bytes from deb12.local (172.16.165.162): icmp_seq=1 ttl=64 time=0.044 ms
64 bytes from deb12.local (172.16.165.162): icmp_seq=2 ttl=64 time=0.103 ms
64 bytes from deb12.local (172.16.165.162): icmp_seq=3 ttl=64 time=0.101 ms
^C
--- deb12.local ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.044/0.082/0.103/0.027 ms
root@maildeb12:~# ping deb12.local
PING deb12.local (172.16.165.162) 56(84) bytes of data.
64 bytes from deb12.local (172.16.165.162): icmp_seq=1 ttl=64 time=0.028 ms
64 bytes from deb12.local (172.16.165.162): icmp_seq=2 ttl=64 time=0.085 ms
64 bytes from deb12.local (172.16.165.162): icmp_seq=3 ttl=64 time=0.083 ms
^C
--- deb12.local ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2041ms
rtt min/avg/max/mdev = 0.028/0.065/0.085/0.026 ms
root@maildeb12:~#
```

La machine est joignable via son nom FQDN ou son nom de domaine, la partie résolution DNS est terminée.

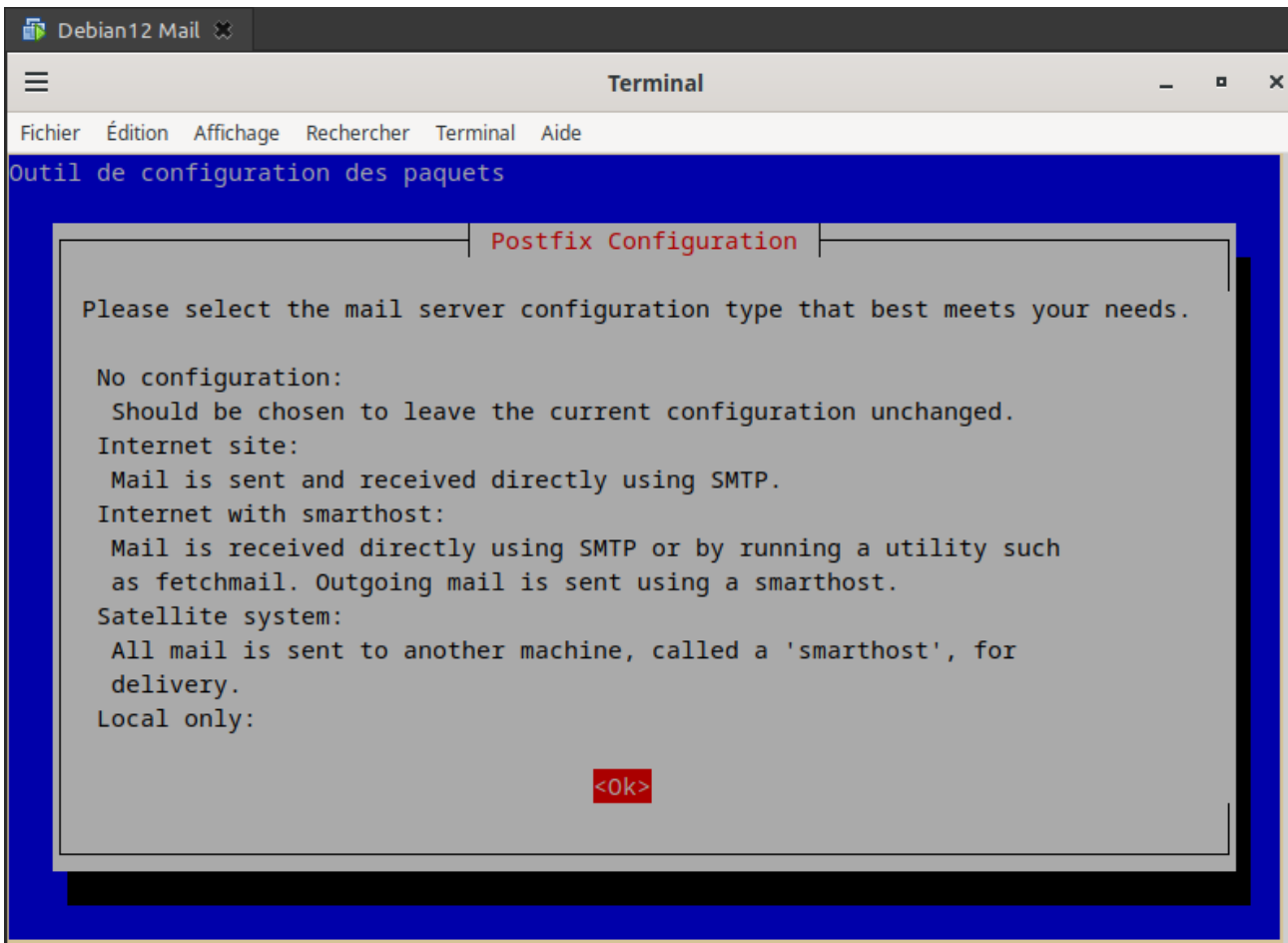
C. INSTALLATION DU SERVEUR DE MESSAGERIE

6 Installation de Postfix

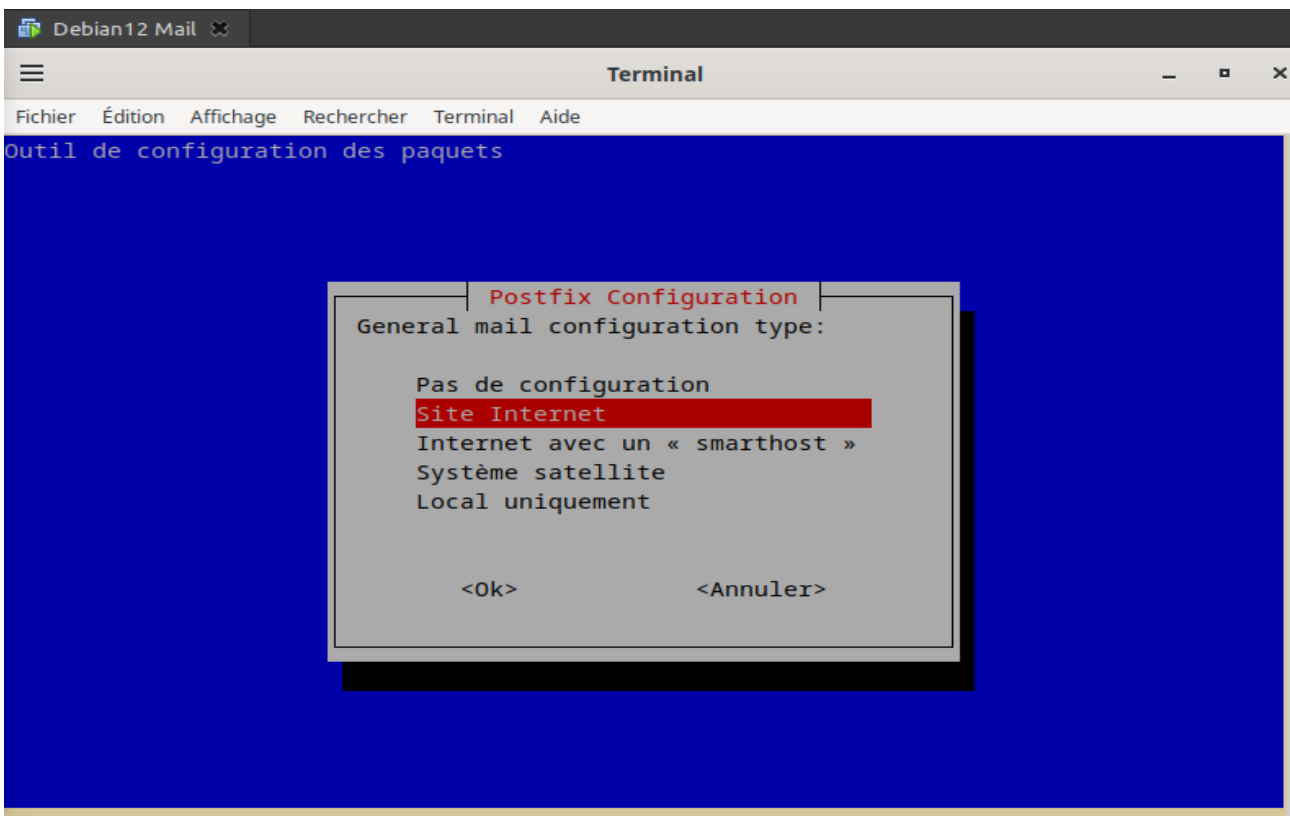
Postfix est l'agent qui va nous servir à envoyer les mails(MTA).

Installer Postfix :

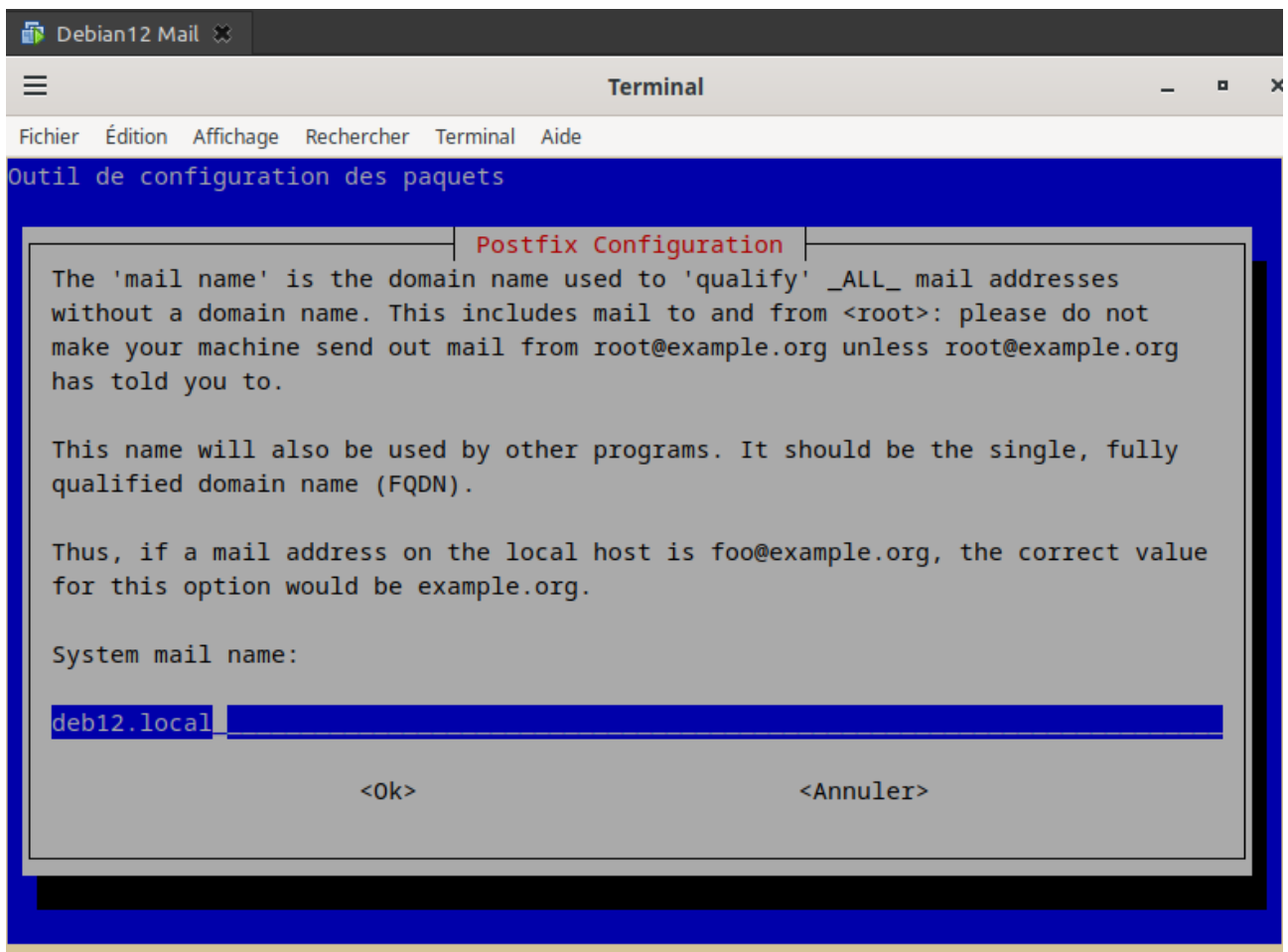
apt install postfix



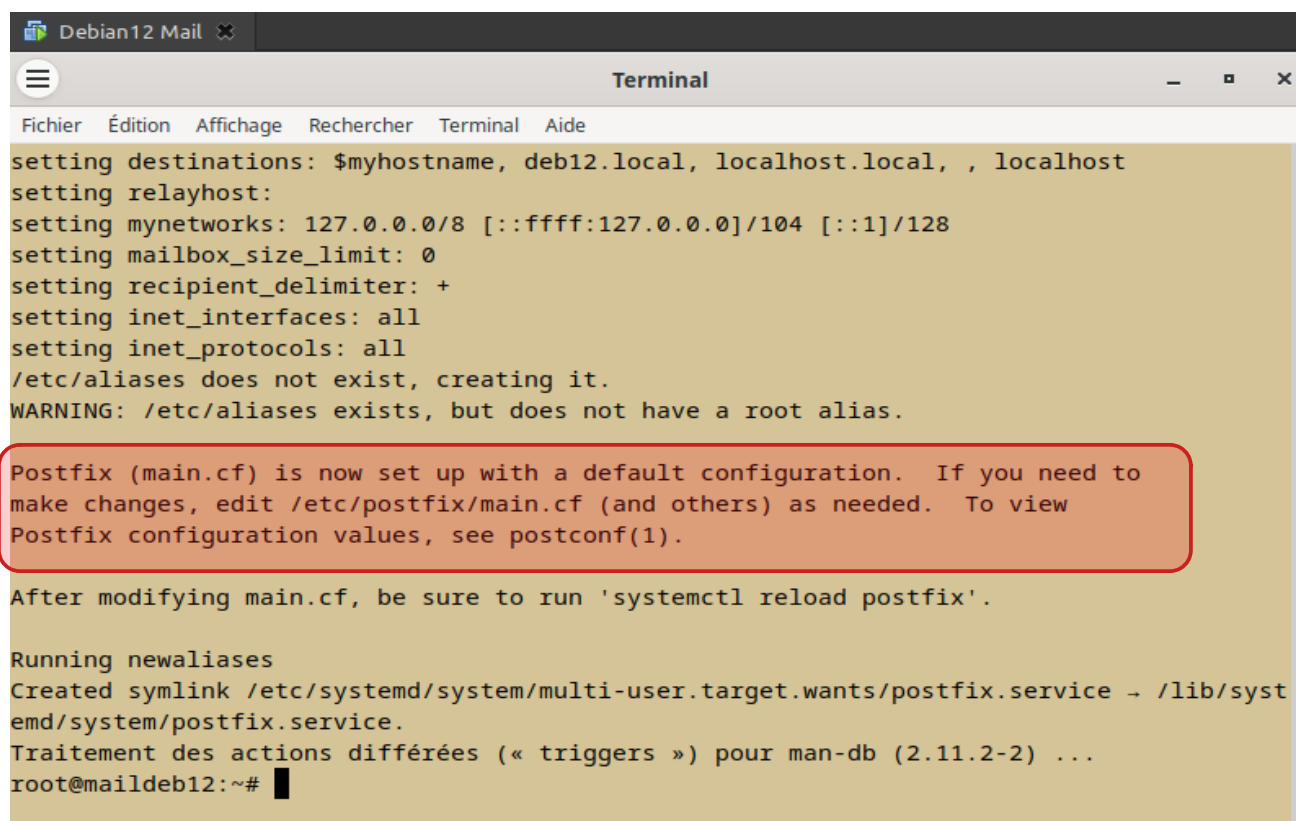
1. Appuyer sur la touche tab, puis sur Entrée



2. Laisser « Site Internet » sélectionné, appuyer sur Entrée



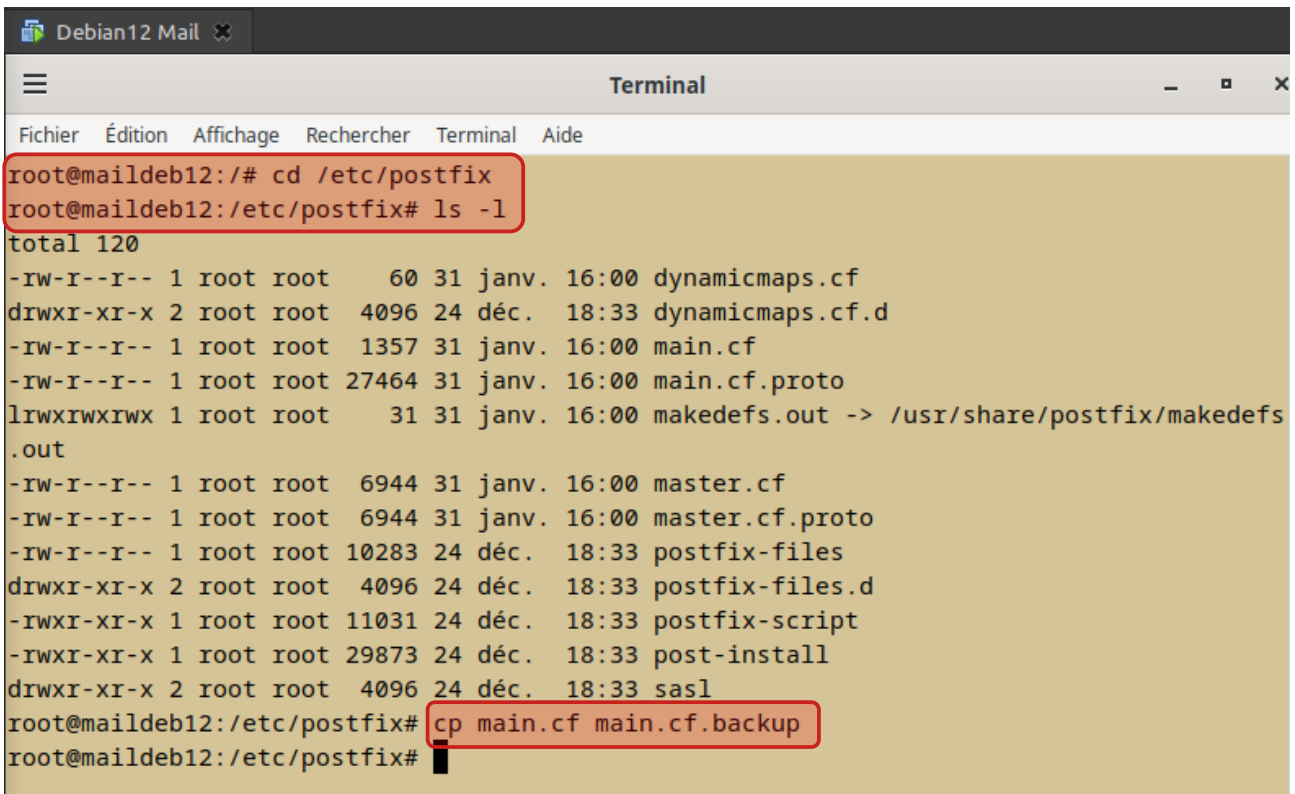
3. Entrer le nom de domaine et appuyer sur Entrée



4. Postfix est maintenant installé

7 Configuration de postfix

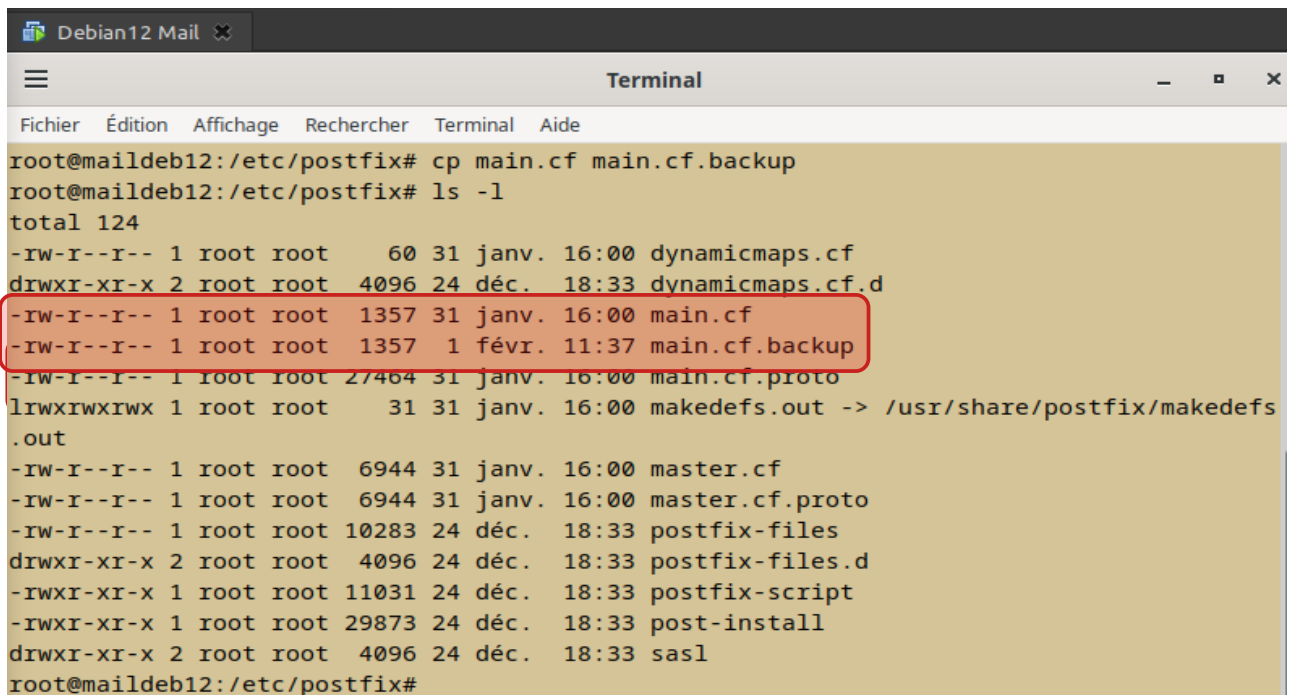
Se déplacer dans le dossier d'installation de Postfix → `cd /etc/postfix`
Lister les dossier et fichiers → `ls -l`
Faire une copie du **fichier de configuration** → `cp main.cf main.cf.backup`



```
root@maildeb12:/# cd /etc/postfix
root@maildeb12:/etc/postfix# ls -l
total 120
-rw-r--r-- 1 root root 60 31 janv. 16:00 dynamicmaps.cf
drwxr-xr-x 2 root root 4096 24 déc. 18:33 dynamicmaps.cf.d
-rw-r--r-- 1 root root 1357 31 janv. 16:00 main.cf
-rw-r--r-- 1 root root 27464 31 janv. 16:00 main.cf.proto
lrwxrwxrwx 1 root root 31 31 janv. 16:00 makedefs.out -> /usr/share/postfix/makedefs.out
-rw-r--r-- 1 root root 6944 31 janv. 16:00 master.cf
-rw-r--r-- 1 root root 6944 31 janv. 16:00 master.cf.proto
-rw-r--r-- 1 root root 10283 24 déc. 18:33 postfix-files
drwxr-xr-x 2 root root 4096 24 déc. 18:33 postfix-files.d
-rwxr-xr-x 1 root root 11031 24 déc. 18:33 postfix-script
-rwxr-xr-x 1 root root 29873 24 déc. 18:33 post-install
drwxr-xr-x 2 root root 4096 24 déc. 18:33 sasl
root@maildeb12:/etc/postfix# cp main.cf main.cf.backup
root@maildeb12:/etc/postfix#
```

5. Sauvegarder le fichier "main.cf" en faisant une copie

La copie du fichier « main.cf » en « main.cf.backup », nous permet en cas problème lors de la configuration de ce fichier, de revenir à la configuration par défaut.



```
root@maildeb12:/etc/postfix# cp main.cf main.cf.backup
root@maildeb12:/etc/postfix# ls -l
total 124
-rw-r--r-- 1 root root 60 31 janv. 16:00 dynamicmaps.cf
drwxr-xr-x 2 root root 4096 24 déc. 18:33 dynamicmaps.cf.d
-rw-r--r-- 1 root root 1357 31 janv. 16:00 main.cf
-rw-r--r-- 1 root root 1357 1 févr. 11:37 main.cf.backup
-rw-r--r-- 1 root root 27464 31 janv. 16:00 main.cf.proto
lrwxrwxrwx 1 root root 31 31 janv. 16:00 makedefs.out -> /usr/share/postfix/makedefs.out
-rw-r--r-- 1 root root 6944 31 janv. 16:00 master.cf
-rw-r--r-- 1 root root 6944 31 janv. 16:00 master.cf.proto
-rw-r--r-- 1 root root 10283 24 déc. 18:33 postfix-files
drwxr-xr-x 2 root root 4096 24 déc. 18:33 postfix-files.d
-rwxr-xr-x 1 root root 11031 24 déc. 18:33 postfix-script
-rwxr-xr-x 1 root root 29873 24 déc. 18:33 post-install
drwxr-xr-x 2 root root 4096 24 déc. 18:33 sasl
root@maildeb12:/etc/postfix#
```

6. Vérifier que la copie a bien été effectuée avec « ls -l »

Maintenant on ouvre le fichier main.cf :

nano main.cf

Modifier ces lignes avec les bonne informations :

- a) **myhostname = maildeb12.deb12.local** → nom d'hôte de la machine
- b) **mynetworks = 172.16.165.0/24** → réseau sur lequel se trouve ma machine

Rajouter ces lignes si elles n'existent pas :

- c) **mydomain = deb12.local** → nom de domaine dont ma machine fait partie
- d) **home_mailbox = Maildir/** → les mails seront stockés dans **/home/Maildir** pour chaque utilisateur

Rajouter ces lignes à la fin du fichier :

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_security_options = noanonymous
```

Ces dernières directives configurées dans le fichier de configuration de postfix permettent d'activer l'authentification SASL.

De définir Dovecot comme serveur SASL.

De définir le chemin du socket pour la communication avec Dovecot.

Et enfin de spécifier que l'authentification anonyme n'est pas autorisée.

Cela renforce la sécurité du serveur de messagerie en exigeant une authentification valide pour l'envoi de courriels.

```
GNU nano 7.2 main.cf *
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

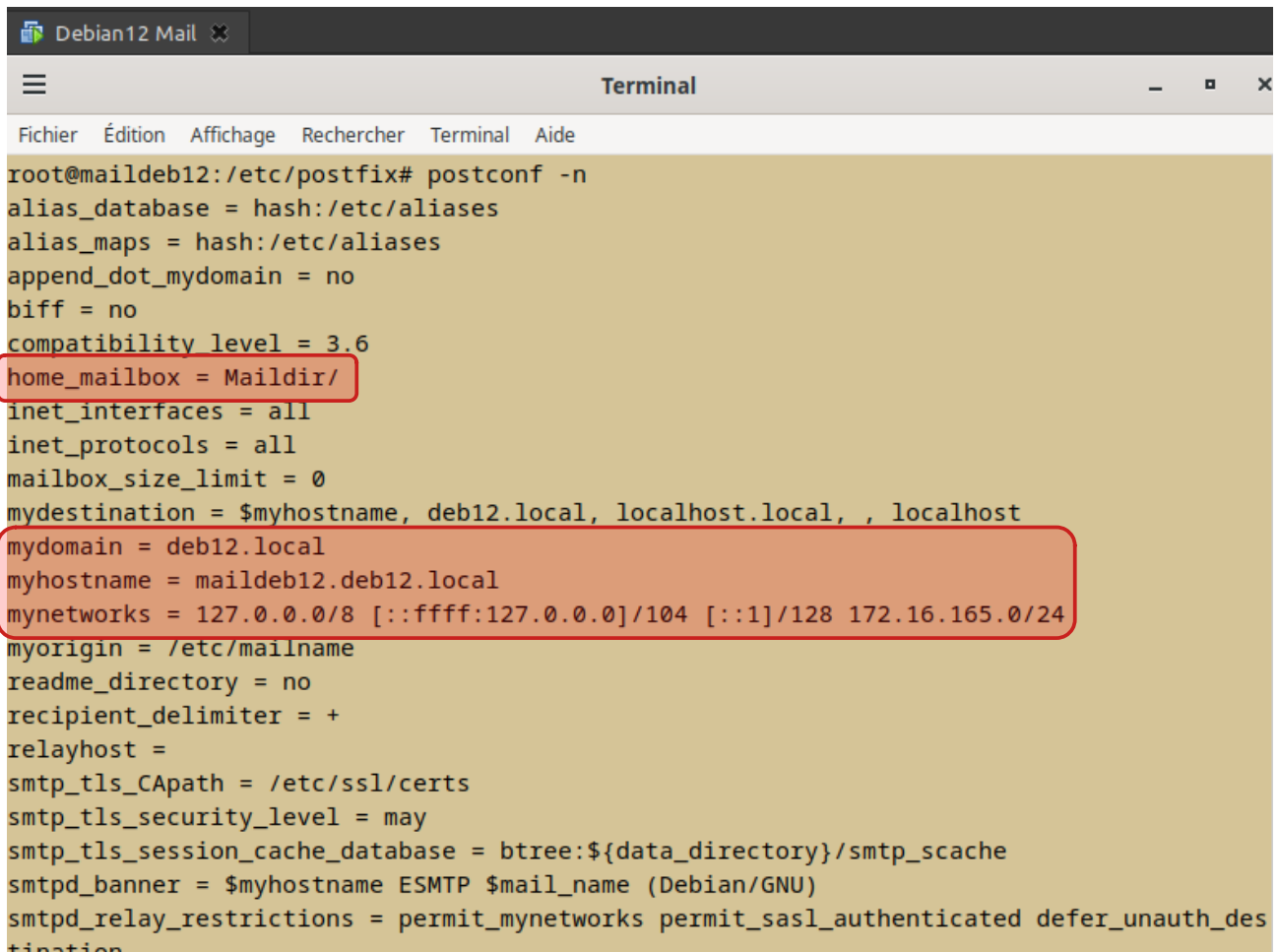
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_de>
myhostname = maildeb12.deb12.local a
mydomain = deb12.local c
home_mailbox = Maildir/ d
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, deb12.local, localhost.local, , localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 172.16.165.0/24 b
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^\ Remplacer  ^U Coller    ^J Justifier  ^/ Aller ligne
```

7.1. Modification du fichier « main.cf »

Comme d'habitude avec **nano** on sauvegarde les modifications avec **ctrl +x** puis **Entrée**.

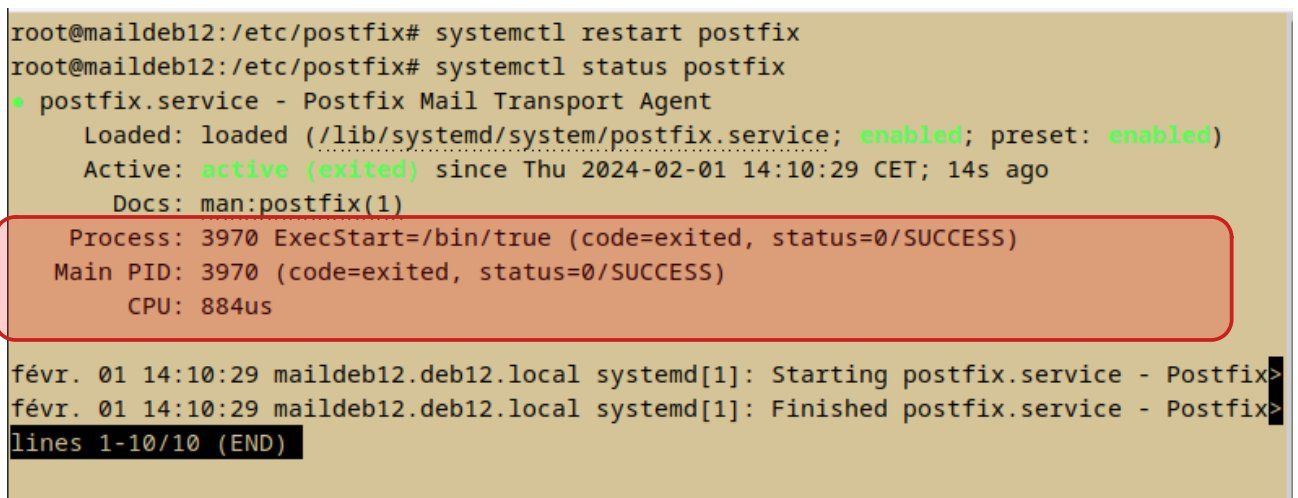
On vérifie la configuration du fichier main.cf :
postconf -n



```
root@maildeb12:/etc/postfix# postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
append_dot_mydomain = no
biff = no
compatibility_level = 3.6
home_mailbox = Maildir/
inet_interfaces = all
inet_protocols = all
mailbox_size_limit = 0
mydestination = $myhostname, deb12.local, localhost.local, , localhost
mydomain = deb12.local
myhostname = maildeb12.deb12.local
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 172.16.165.0/24
myorigin = /etc/mailname
readme_directory = no
recipient_delimiter = +
relayhost =
smtp_tls_CApath = /etc/ssl/certs
smtp_tls_security_level = may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_des
tination
```

7.2. Vérification avec la commande "postconf -n"

8 Redémarrer le service postfix :
systemctl restart postfix
systemctl status postfix



```
root@maildeb12:/etc/postfix# systemctl restart postfix
root@maildeb12:/etc/postfix# systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
   Loaded: loaded (/lib/systemd/system/postfix.service; enabled; preset: enabled)
   Active: active (exited) since Thu 2024-02-01 14:10:29 CET; 14s ago
     Docs: man:postfix(1)
   Process: 3970 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 3970 (code=exited, status=0/SUCCESS)
      CPU: 884us

févr. 01 14:10:29 maildeb12.deb12.local systemd[1]: Starting postfix.service - Postfix>
févr. 01 14:10:29 maildeb12.deb12.local systemd[1]: Finished postfix.service - Postfix>
lines 1-10/10 (END)
```

8. Redémarrer Postfix et vérifier son statut

9 Vérification du port 25(port d'envoi de mail par défaut)

1. On va faire une vérification et voir si Postfix « écoute » sur le port 25, ce qui devrait être le cas par défaut.

ss -tlnp

t → Affiche les connexions TCP.

l → Affiche les connexions qui écoutent (en mode écoute).

p → Affiche le numéro de processus et le nom du programme associés à chaque connexion.

n → Affiche les adresses IP et les numéros de port sous forme numérique plutôt que de résoudre les noms d'hôtes et de services.

```
root@maildeb12:/etc/postfix# ss -tlnp
State      Recv-Q    Send-Q    Local Address:Port    Peer Address:Port    Process
LISTEN     0         100      0.0.0.0:25            0.0.0.0:*            users
:(("master",pid=3967,fd=13))
LISTEN     0         128      127.0.0.1:631        0.0.0.0:*            users
:(("cupsd",pid=987,fd=7))
LISTEN     0         128      [::1]:631            [::]:*               users
:(("cupsd",pid=987,fd=6))
LISTEN     0         511      *:80                 *:.*                  users
:(("apache2",pid=1051,fd=4),("apache2",pid=1050,fd=4),("apache2",pid=1049,fd=4))
LISTEN     0         100      [::]:25              [::]:*               users
:(("master",pid=3967,fd=14))
root@maildeb12:/etc/postfix#
```

9.1. Vérifier que Postfix écoute sur le port 25

Sur la capture ci-dessus on peut constater que le port 25 est en mode « écoute »(LISTEN)

10 Envoi d'un mail de test

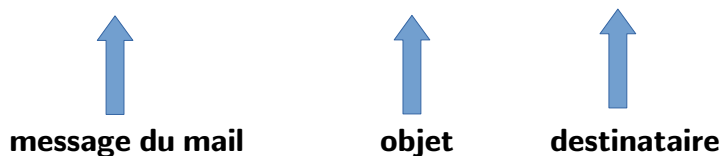
Cette étape peut être considérée comme facultative, mais il peut être intéressant de confirmer par l'envoi d'un mail de test, que notre configuration est fonctionnelle **jusqu'à ce stade**.

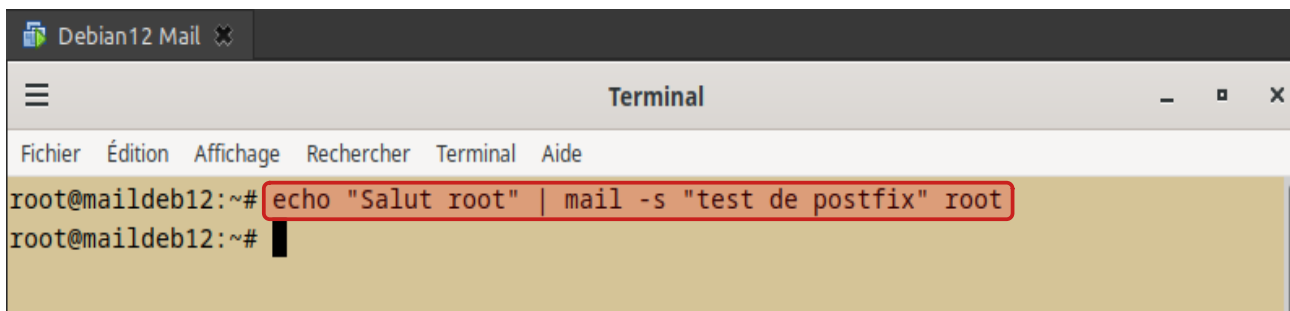
On va pour ce faire installer **mailutils**, un utilitaire qui permet entre autre d'envoyer et de recevoir des mails en ligne de commande :

apt install mailutils

Commande pour envoyer le mail à l'utilisateur root :

echo "Salut root" | mail -s "test de postfix" root






```
Debian12 Mail x
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
root@maildeb12:~# echo "Salut root" | mail -s "test de postfix" root
root@maildeb12:~#
```

9.2. Envoi du mail de test à "root"

Maintenant on va s'assurer qu'il n'y a pas de mail en attente d'envoi ou de nouveaux messages:

mailq
mail



```
Debian12 Mail x
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
root@maildeb12:~# echo "Salut root" | mail -s "test de postfix" root
root@maildeb12:~# mailq
Mail queue is empty
root@maildeb12:~# mail
Pas de courrier pour root
root@maildeb12:~#
```

9.3. Pas de nouveaux mails ou en attente d'envoi

La sortie de la commande « mailq » indique qu'il n'y a pas de mail en attente d'envoi
La sortie de la commande « mail » indique qu'il n'y a pas de nouveaux mail

Pour la consultation du mail il faudra se rendre dans le dossier **Maildir/new** de l'utilisateur :

cd /root/Maildir/new

ls → pour lister le contenu du dossier (on constate qu'il y a un message)

cat 1706862785.V801Iff16fM424124.maildeb12.deb12.local → afficher le message


```

root@maildeb12:~# cd /root/Maildir/new
root@maildeb12:~/Maildir/new# ls
1706862785.V801Iff16fM424124.maildeb12.deb12.local
root@maildeb12:~/Maildir/new# cat 1706862785.V801Iff16fM424124.maildeb12.deb12.local
Return-Path: <root@maildeb12.deb12.local>
X-Original-To: root
Delivered-To: root@deb12.local
Received: by maildeb12.deb12.local (Postfix, from userid 0)
        id 62C5ADF641; Fri, 2 Feb 2024 09:33:05 +0100 (CET)
Subject: test de postfix Objet du message
To: root@deb12.local Adresse destinataire
User-Agent: mail (GNU Mailutils 3.15)
Date: Fri, 2 Feb 2024 09:33:05 +0100
Message-Id: <20240202083305.62C5ADF641@maildeb12.deb12.local>
From: root <root@maildeb12.deb12.local> Adresse expéditeur

Salut root Message
root@maildeb12:~/Maildir/new# █

```

9.4. Consultation du mail

Notre envoi de mail par postfix a été parfaitement effectué, on va donc passer à l'installation de Dovecot(MDA).

10 Installation de Dovecot

Commande pour installer Dovecot et le service imap :
apt install dovecot-core dovecot-imapd

Modifier le fichier dovecot.conf :
nano /etc/dovecot/dovecot.conf

```

GNU nano 7.2 dovecot.conf *
# Default values are shown for each setting, it's not required to uncomment
# those. These are exceptions to this though: No sections (e.g. namespace {})
# or plugin settings are added by default, they're listed only as examples.
# Paths are also just examples with the real defaults being based on configure
# options. The paths listed here are for configure --prefix=/usr
# --sysconfdir=/etc --localstatedir=/var

# Enable installed protocols
!include_try /usr/share/dovecot/protocols.d/*.protocol

# A comma separated list of IPs or hosts where to listen in for connections.
# "*" listens in all IPv4 interfaces, "::" listens in all IPv6 interfaces.
# If you want to specify non-default ports or anything more complex,
# edit conf.d/master.conf.
listen = *, :: Décommenter cette ligne

# Base directory where to store runtime data.
#base_dir = /var/run/dovecot/

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier  ^/ Aller ligne

```

10.1. Supprimer le commentaire devant "Listen"

En **supprimant** le commentaire devant « Listen », Dovecot **écouter**a sur **toutes** les cartes réseau disponibles sur la machine que ce soit en IPv4 ou IPv6.

Se déplacer dans le dossier conf.d :
`cd /etc/dovecot/conf.d`

Ce dossier contient plusieurs fichiers de configuration de Dovecot et on va en modifier certains :
nano 10-auth.conf

```
GNU nano 7.2 10-auth.conf *
##
## Authentication processes
##
# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
# matches the local IP (ie. you're connecting from the same computer), the
# connection is considered secure and plaintext authentication is allowed.
# See also ssl=required setting.
disable_plaintext_auth = no
# Authentication cache size (e.g. 10M). 0 means it's disabled. Note that
# bsdauth and PAM require cache_key to be set for caching to be used.
#auth_cache_size = 0
# Time to live for cached data. After TTL expires the cached record is no
```

10.2. Décommenter et remplacer "yes" par "no"

```
GNU nano 7.2 10-auth.conf *
#auth_ssl_require_client_cert = no
# Take the username from client's SSL certificate, using
# X509_NAME_get_text_by_NID() which returns the subject's DN's
# CommonName.
#auth_ssl_username_from_cert = no
# Space separated list of wanted authentication mechanisms:
# plain login digest-md5 cram-md5 ntlm rpa apop anonymous gssapi otp
# gss-spnego
# NOTE: See also disable_plaintext_auth setting.
auth_mechanisms = plain login
##
## Password and user databases
##
```

10.3. Ajouter "login" en fin de ligne

La configuration **auth_mechanism = plain login** signifie que **Dovecot** autorise à la fois les méthodes d'authentification **plain** et **login** lors de la connexion des clients. Ces méthodes sont considérées comme des méthodes d'authentification **en texte brut** d'où le choix → **disable_plaintext_auth = no**.

Il faudra s'assurer que les clients **peuvent** utiliser des protocoles plus sécurisés **avant** de désactiver l'authentification **en texte brut**.

nano 10-mail.conf

```
GNU nano 7.2 10-mail.conf *
#
# There are a few special variables you can use, eg.:
#
# %u - username
# %n - user part in user@domain, same as %u if there's no domain
# %d - domain part in user@domain, empty if there's no domain
# %h - home directory
#
# See doc/wiki/Variables.txt for full list. Some examples:
#
mail_location = maildir:~/Maildir Décommenter cette ligne
# mail_location = mbox:~/mail:INBOX=/var/mail/%u
# mail_location = mbox:/var/mail/%d/%1n/%n:INDEX=/var/indexes/%d/%1n/%n
#
# <doc/wiki/MailLocation.txt>
#
#mail_location = mbox:~/mail:INBOX=/var/mail/%u Commenter cette ligne
#
# If you need to set multiple mailbox locations or want to change default
# namespace settings, you can do it by defining namespace sections.

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich. ^\ Remplacer ^U Coller    ^J Justifier ^/ Aller ligne
```

10.4. Choisir « Maildir » pour stocker les mails

nano 10-master.conf

```
GNU nano 7.2 10-master.conf *
# permissions (e.g. 0777 allows everyone full permissions).
unix_listener auth-userdb {
    #mode = 0666
    #user =
    #group =
}

# Postfix smtp-auth
unix_listener /var/spool/postfix/private/auth {
    mode = 0666
    user = postfix
    group = postfix
}

# Auth process is run as this user.
#user = $default_internal_user
}
```

10.5. Décommenter et ajouter les lignes "user" et "group"

Cette dernière modification permet à **Postfix** d'échanger des informations d'authentification avec **Dovecot** de manière **sécurisée**.

Ainsi le socket utilisé pour la communication entre **Dovecot** et **Postfix** est spécifié par le chemin **/var/spool/postfix/private/auth**. De plus cette communication est **restreinte** pour ce socket **uniquement** à **Postfix** ou à des membres du **groupe Postfix**.

Redémarrer Dovecot :

```
root@maildeb12:/etc/dovecot/conf.d# systemctl restart dovecot.service
root@maildeb12:/etc/dovecot/conf.d# systemctl status dovecot.service
• dovecot.service - Dovecot IMAP/POP3 email server
Loaded: loaded (/lib/systemd/system/dovecot.service; enabled; preset: enabled)
Active: active (running) since Fri 2024-02-02 14:25:57 CET; 2min 33s ago
  Docs: man:dovecot(1)
        https://doc.dovecot.org/
Main PID: 7015 (dovecot)
Status: "v2.3.19.1 (9b53102964) running"
Tasks: 4 (limit: 4582)
Memory: 3.2M
CPU: 24ms
CGroup: /system.slice/dovecot.service
├─7015 /usr/sbin/dovecot -F
├─7017 dovecot/anvil
├─7018 dovecot/log
└─7019 dovecot/config

févr. 02 14:25:57 maildeb12.deb12.local systemd[1]: Starting dovecot.service - Dovecot >
févr. 02 14:25:57 maildeb12.deb12.local dovecot[7015]: master: Dovecot v2.3.19.1 (9b53>
févr. 02 14:25:57 maildeb12.deb12.local systemd[1]: Started dovecot.service - Dovecot >
lines 1-19/19 (END)
```

10.6. Redémarrer le service Dovecot et contrôler son statut

11 Test du fonctionnement de Dovecot

Vérifier que Dovecot « écoute » sur le port 143 :
ss - tlpn

Ci-dessous on constate que Dovecot écoute bien sur le port 143 en IPv4 et IPv6

```

root@maildeb12:/etc/dovecot/conf.d# ss -tlnp
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port
Process
LISTEN     0            128         127.0.0.1:631           0.0.0.0:*
  users: (("cupsd",pid=975,fd=7))
LISTEN     0            100         0.0.0.0:993            0.0.0.0:*
  users: (("dovecot",pid=7015,fd=36))
LISTEN     0            100         0.0.0.0:25             0.0.0.0:*
  users: (("master",pid=1493,fd=13))
LISTEN     0            100         0.0.0.0:143            0.0.0.0:*
  users: (("dovecot",pid=7015,fd=34))
LISTEN     0            128         [::1]:631              [::]:*
  users: (("cupsd",pid=975,fd=6))
LISTEN     0            100         [::]:993                [::]:*
  users: (("dovecot",pid=7015,fd=37))
LISTEN     0            100         [::]:25                 [::]:*
  users: (("master",pid=1493,fd=14))
LISTEN     0            511         *:80                    *: *
  users: (("apache2",pid=1007,fd=4),("apache2",pid=1006,fd=4),("apache2",pid=1002,fd=4))
LISTEN     0            100         [::]:143                [::]:*
  users: (("dovecot",pid=7015,fd=35))
root@maildeb12:/etc/dovecot/conf.d#

```

11.1. Dovecot écoute bien sur le port 143

Maintenant, on va envoyer un mail de test et vérifier si Dovecot le réceptionne bien.

Se connecter à Postfix sur le port 25 :

nc localhost 25

netcat(nc) est utilisé ici pour ouvrir une connexion **TCP** avec un **hôte** et un **port spécifiques**.

Commencer la discussion avec le protocole SMTP :

ehlo localhost

```

root@maildeb12:/etc/dovecot/conf.d# nc localhost 25
220 maildeb12.deb12.local ESMTP Postfix (Debian/GNU)
ehlo localhost
250-maildeb12.deb12.local
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING

```

11.2. Se connecter en SMTP à Postfix

```

250-maildeb12.deb12.local
250-PIPELINING
250-SIZE 10240000
250-VERFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
mail from: root a) Expéditeur
250 2.1.0 Ok
rcpt to: haucine b) Destinataire
250 2.1.5 Ok
data Commande pour indiquer que l'on va rentrer des données
354 End data with <CR><LF>.<CR><LF>
subject: test dovecot c) Objet
Salut Dovecot ! d) Message du mail et on finit par un point
.
250 2.0.0 Ok: queued as 1F53EDF41C
quit
221 2.0.0 Bye
root@maildeb12:~# █

```

11.3. Envoi du mail à Dovecot

Ensuite on va se déplacer dans `/home/haucine/Maildir/new` pour consulter notre mail :
`cd/home/haucine/Maildir/new`
`ls` → pour lister les mails
`cat + nom du fichier` → pour afficher le message

```

root@maildeb12:/home/haucine/Maildir# cd /home/haucine/Maildir/new
root@maildeb12:/home/haucine/Maildir/new# ls
1706882749.V801Ia1a74M573508.maildeb12.deb12.local
root@maildeb12:/home/haucine/Maildir/new# cat 1706882749.V801Ia1a74M573508.maildeb12.de
b12.local
Return-Path: <root@deb12.local>
X-Original-To: haucine
Delivered-To: haucine@deb12.local b) Destinataire
Received: from localhost (localhost [127.0.0.1])
        by maildeb12.deb12.local (Postfix) with ESMTP id 1F53EDF41C
        for <haucine>; Fri, 2 Feb 2024 15:03:04 +0100 (CET)
subject: test dovecot c) Objet
Message-Id: <20240202140323.1F53EDF41C@maildeb12.deb12.local>
Date: Fri, 2 Feb 2024 15:03:04 +0100 (CET)
From: root@deb12.local a) Expéditeur

Salut Dovecot ! d) Message
root@maildeb12:/home/haucine/Maildir/new# /home/haucine/Maildir/new █

```

11.4. Dovecot fonctionne, le contenu du mail est identique

Le contenu du mail est conforme à celui envoyé, Dovecot fonctionne correctement. On peut maintenant passer à la mise en place d'un Webmail pour les utilisateurs.

D. Mise en place du Webmail Zimbra Desktop 8.7.11_GA

12 Installation de Zimbra Desktop

Se rendre sur le site officiel et télécharger la version pour Ubuntu 16.04 LTS :

<https://www.zimbra.com/product/download/zimbra-collaboration-archives>

Ou lancer cette commande :

wget https://files.zimbra.com/downloads/8.7.11_GA/zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956.tgz

```
root@maildeb12:~# wget https://files.zimbra.com/downloads/8.7.11_GA/zcs-NETWORK-8.7.11_GA_1854.
UBUNTU16_64.20170531151956.tgz
--2024-02-15 12:49:03-- https://files.zimbra.com/downloads/8.7.11_GA/zcs-NETWORK-8.7.11_GA_185
4.UBUNTU16_64.20170531151956.tgz
Résolution de files.zimbra.com (files.zimbra.com)... 13.32.220.233
Connexion à files.zimbra.com (files.zimbra.com)|13.32.220.233|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 353839967 (337M) [binary/octet-stream]
Sauvegarde en : « zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956.tgz »

zcs-NETWORK-8.7.11_GA 100%[=====] 337,45M 11,1MB/s ds 36s

2024-02-15 12:49:39 (9,40 MB/s) - « zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956.tgz »
sauvegardé [353839967/353839967]

root@maildeb12:~# █
```

12.1 Téléchargement du client Zimbra Desktop

Extraire les fichiers :

cd ~/Téléchargements

tar -xzvf zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956.tgz

-x: Indique l'extraction du contenu.

-z: Spécifie que l'archive est compressée en utilisant gzip.

-v: Mode verbeux pour afficher les fichiers à mesure qu'ils sont extraits.

-f: Permet de spécifier le nom du fichier avec lequel travailler.kk

```
root@maildeb12:~/Téléchargements# ls
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956.tgz
root@maildeb12:~/Téléchargements# tar -xzvf zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.2017
0531151956.tgz
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956/
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956/README.txt
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956/util/
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956/util/addUser.sh
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956/util/globals.sh
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956/util/modules/
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956/util/modules/packages.sh
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956/util/modules/getconfig.sh
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956/util/modules/postinstall.sh
```

12.2 Extraire les fichiers de Zimbra Desktop

Se déplacer dans le répertoire extrait :

```
cd zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956
```

```
root@maildeb12:~/Téléchargements# ls
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956.tgz
root@maildeb12:~/Téléchargements# cd zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956
root@maildeb12:~/Téléchargements/zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956# ls
bin data docs install.sh lib packages readme_binary_en_US.txt README.txt util
root@maildeb12:~/Téléchargements/zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956# █
```

Et lancer le script d'installation :

```
./install.sh --platform-override
```

On utilisera « --platform-override » pour passer outre l'étape de vérification de notre OS car notre version de Zimbra Desktop opensource a été conçue pour une ancienne version de Ubuntu.

```
root@maildeb12:~/Téléchargements# ls
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956
zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956.tgz
root@maildeb12:~/Téléchargements# cd zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956
root@maildeb12:~/Téléchargements/zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956# ls
bin data docs install.sh lib packages readme_binary_en_US.txt README.txt util
root@maildeb12:~/Téléchargements/zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956# ./install.sh
--platform-override █
```

12.3 Lancer le script d'installation de Zimbra Desktop

Répondre aux questions :

```
"Synacor" and "Zimbra" are registered trademarks of Synacor, Inc. and, along with other Synacor trad
emarks, services marks and product names, may not be used without the prior permission of Synacor, I
nc.
```

```
Do you agree with the terms of the software license agreement? [N] y █
```

```
Agreement. The parties acknowledge and agree that a material breach of this
Agreement adversely affecting Autonomy's proprietary rights would cause
irreparable harm to Autonomy for which a remedy at law would be inadequate and
that Autonomy shall be entitled to injunctive relief in addition to any
remedies it may have hereunder or at law.
```

```
Do you agree with the terms of the software license agreement? [N] y █
```

```
Checking for installable packages
```

```
Found zimbra-core (local)
Found zimbra-ldap (local)
Found zimbra-logger (local)
Found zimbra-mta (local)
Found zimbra-dnscache (local)
Found zimbra-snmp (local)
Found zimbra-store (local)
Found zimbra-apache (local)
Found zimbra-spell (local)
Found zimbra-convertd (local)
Found zimbra-proxy (local)
Found zimbra-archiving (local)
```

```
Select the packages to install
```

```
Install zimbra-ldap [Y] y
```

```
Install zimbra-logger [Y] y
```

```
Install zimbra-mta [Y] n ← Répondre non car Postfix est le MTA pour notre configuration
```

```
Install zimbra-dnscache [N] y
```

```
Install zimbra-snmp [Y] y
```

```
Install zimbra-store [Y] y
```

```
Install zimbra-apache [Y] y
```

```
Install zimbra-spell [Y] y
```

```
Install zimbra-convertd [Y] y
```

```
Install zimbra-proxy [Y] y
```

```
Install zimbra-archiving [N] y
```

12.4 Installer les fonctionnalités de Zimbra Desktop

Comme nous avons installé Postfix auparavant il ne faut pas installer le MTA de Zimbra !

Confirmer l'installation :

```
This platform is DEBIANUNKNOWN
Packages found: UBUNTU16_64
This may or may not work.
```

```
Using packages for a platform in which they were not designed for
may result in an installation that is NOT usable. Your support
options may be limited if you choose to continue.
You will also be responsible for configuring the system to point
at an appropriate package repository for third party.
```

```
Install anyway? [N] y
```

```
The system will be modified. Continue? [N] y
```

Enfin il est impossible d'installer Zimbra Desktop sur notre serveur Debian 12 avec notre configuration actuelle.

```
Installing local packages (11):
```

```
zimbra-core  
zimbra-ldap  
zimbra-logger  
zimbra-dnscache  
zimbra-snmp  
zimbra-store  
zimbra-apache  
zimbra-spell  
zimbra-convertd  
zimbra-proxy  
zimbra-archiving  
...
```

```
ERROR: Unable to install required packages
```

```
Fix the issues with remote package installation and rerun the installer
```

```
root@maildeb12:~/Téléchargements/zcs-NETWORK-8.7.11_GA_1854.UBUNTU16_64.20170531151956# █
```

12.5 Erreur lors de l'installation de Zimbra Desktop

On va donc se tourner vers un autre webmail qui pourrait convenir, en l'occurrence Rainloop.

E. Mise en place du Webmail Rainloop

13 Installation de Rainloop

D'abord on va installer les dépendances :

```
apt install libapache2-mod-php8.2 php8.2-curl php8.2-xml
```

On supprime l'index par défaut de Apache2 :

```
cd /var/www/html  
rm index.html
```

Puis on lance la commande d'installation de Rainloop dans /var/www/html :

```
curl -sL https://repository.rainloop.net/installer.php | php
```

curl est un outil en ligne de commande utilisé pour transférer des données à partir ou vers un serveur, dans ce cas pour télécharger un script PHP à partir de l'URL spécifiée.

- **-s** est une option pour **curl** qui supprime l'affichage de la progression ou d'informations supplémentaires.
- **-L** est une option pour **curl** qui suit les redirections **HTTP** lors du téléchargement.
- **https://repository.rainloop.net/installer.php** est l'URL du script d'installation de **RainLoop**.
- **|** est l'opérateur de tube (pipe) qui redirige la sortie de la commande **curl** vers l'entrée de la commande **php**.
- **php** est l'interpréteur de commandes PHP qui exécute le script d'installation de **RainLoop** téléchargé précédemment.

```
root@maildeb12:/var/www/html# rm index.html  
root@maildeb12:/var/www/html# ls  
root@maildeb12:/var/www/html# curl -sL https://repository.rainloop.net/installer.php | php  
  
[RainLoop Webmail Installer]  
  
* Connecting to repository ...  
* Downloading package ...  
* Complete downloading!  
* Installing package ...  
* Complete installing!  
  
* [Success] Installation is finished!  
root@maildeb12:/var/www/html# █
```

13.1 Installation de Rainloop

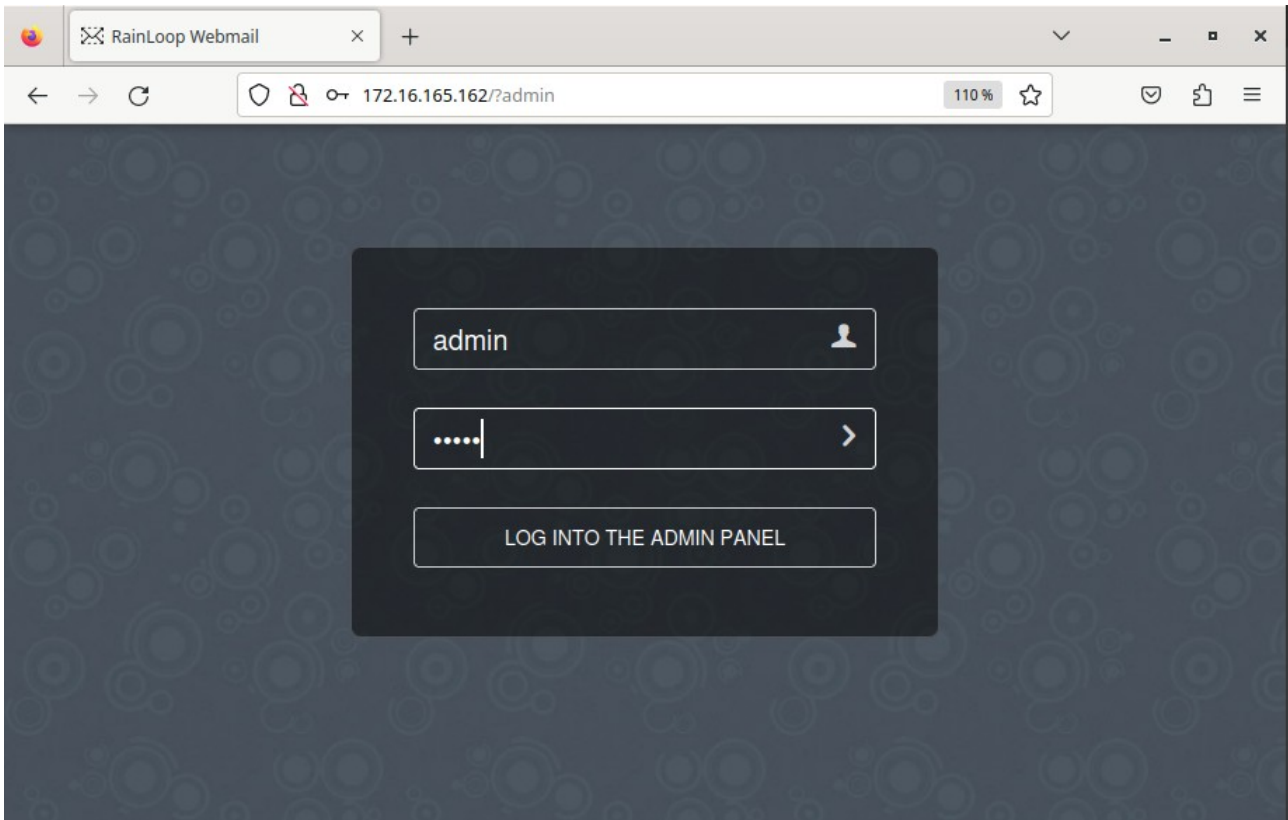
On se connecte à l'interface web d'administration de **Rainloop** :

<http://172.16.165.162/?admin>

login : admin

pass : 12345

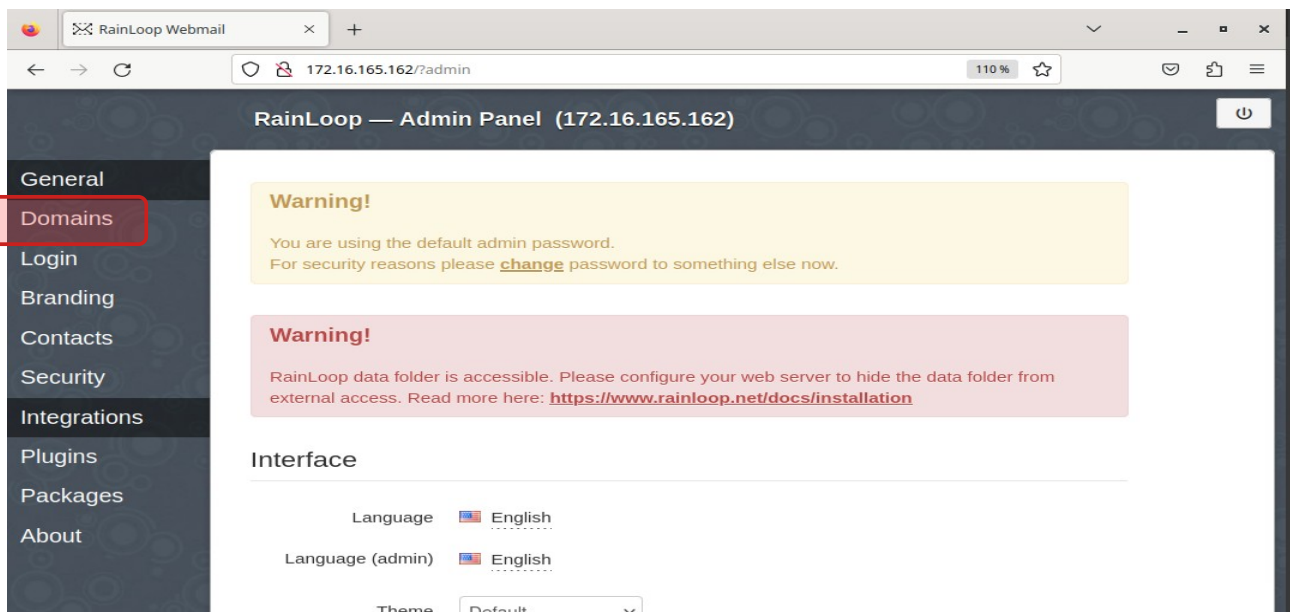
- ← rajouter « **/?admin** » à notre adresse IP
- ← login par défaut
- ← mot de passe par défaut



13.2 Connexion à la page d'administration de **Rainloop**

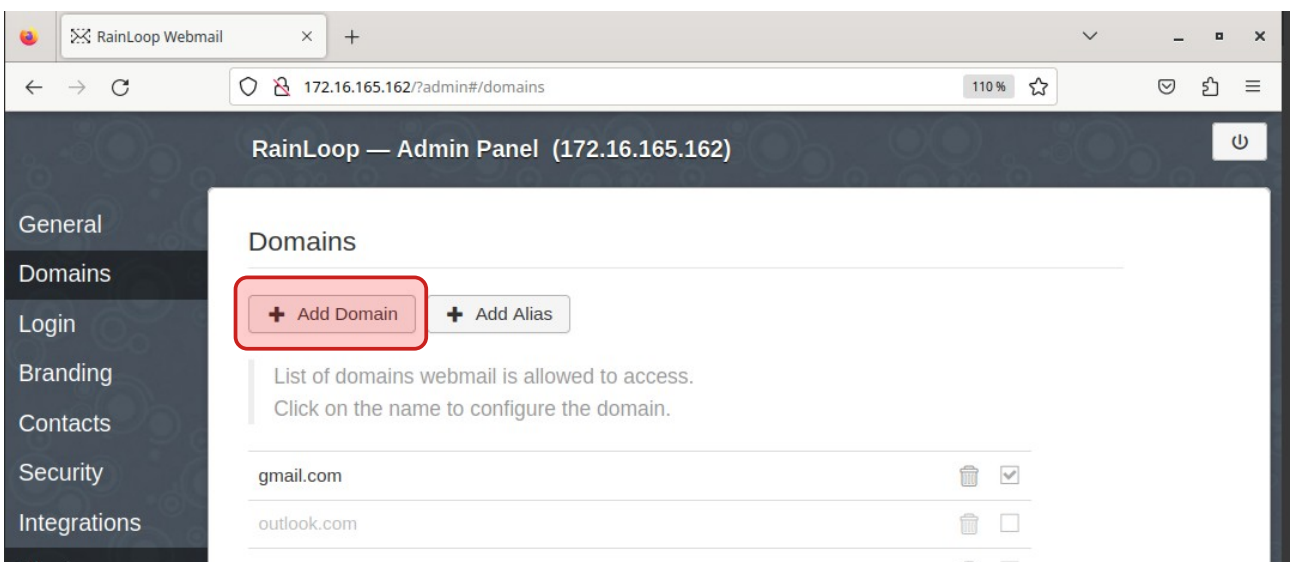
14 Configuration de Rainloop

On clique sur l'onglet **Domains** dans le menu à gauche pour rajouter notre nom de domaine à **Rainloop**.



14.1 Configuration de Rainloop

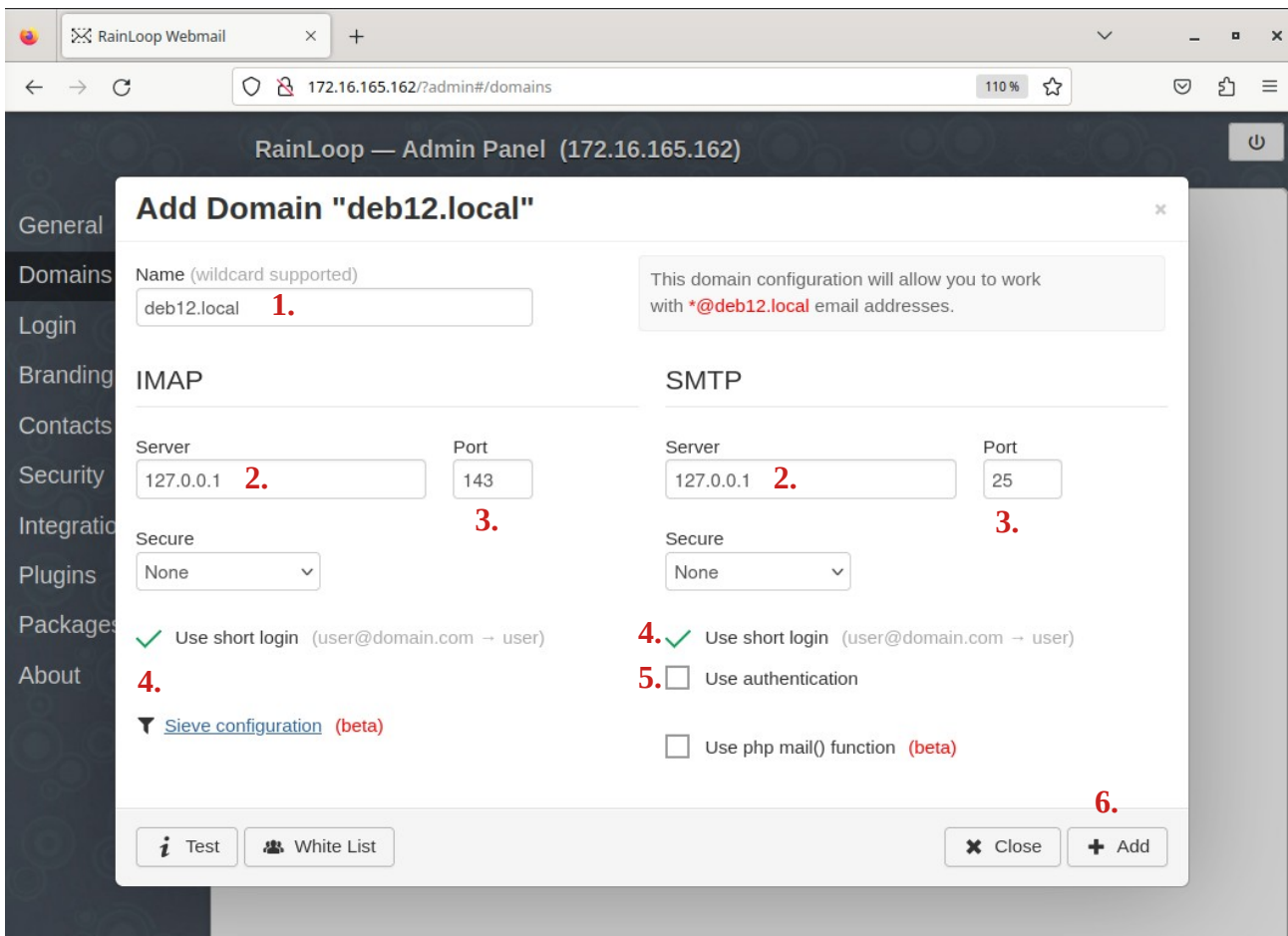
Puis sur on clique sur le bouton « **Add Domain** »



14.2 Configuration de Rainloop

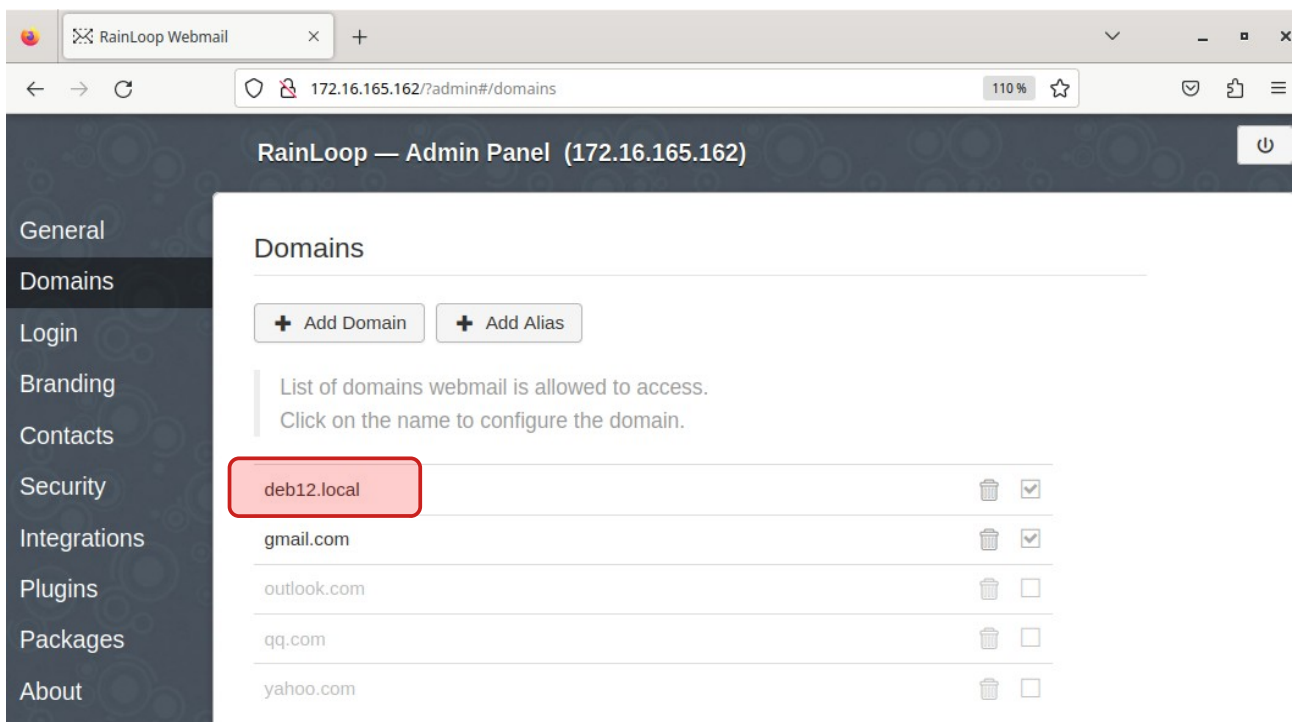
Ensuite :

- 1. on renseigne le nom de notre domaine** → deb12.local
- 2. on définit nos serveurs IMAP et SMTP** → on pourra mettre notre nom de domaine ou bien l'adresse IP locale 127.0.0.1
- 3. on définit les ports** → ici respectivement 143 pour IMAP et 25 pour SMTP
- 4. on clique pour cocher les cases « use short login »**
- 5. on clique pour décocher l'option « use authentication »**
- 6. on clique sur le bouton « Add » pour valider les changements**



14.3 Configuration de Rainloop

Sur l'écran suivant, on constate que notre nom de domaine a bien été rajouté à Rainloop :



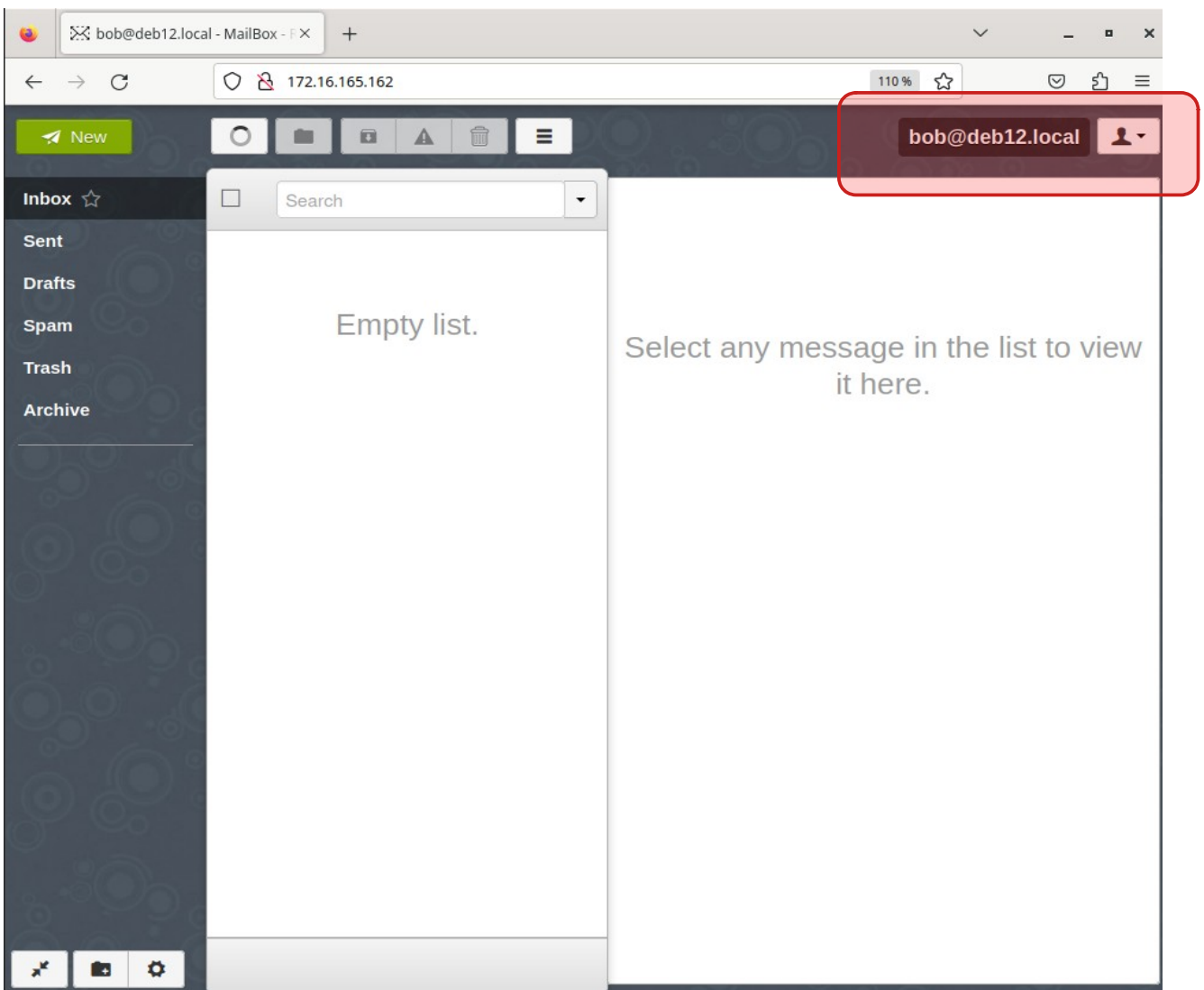
14.4 Configuration de Rainloop

On va maintenant créer un nouvel utilisateur sur notre serveur:
adduser bob

```
root@maildeb12:/var/www/html# adduser bob
Ajout de l'utilisateur « bob » ...
Ajout du nouveau groupe « bob » (1001) ...
Ajout du nouvel utilisateur « bob » (1001) avec le groupe « bob » (1001) ...
Création du répertoire personnel « /home/bob » ...
Copie des fichiers depuis « /etc/skel » ...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
Modifier les informations associées à un utilisateur pour bob
Entrer la nouvelle valeur, ou appuyer sur ENTER pour la valeur par défaut
  NOM []:
  Numéro de chambre []:
  Téléphone professionnel []:
  Téléphone personnel []:
  Autre []:
Cette information est-elle correcte ? [0/n]o
Ajout du nouvel utilisateur « bob » aux groupes supplémentaires « users » ...
Ajout de l'utilisateur « bob » au groupe « users » ...
root@maildeb12:/var/www/html#
```

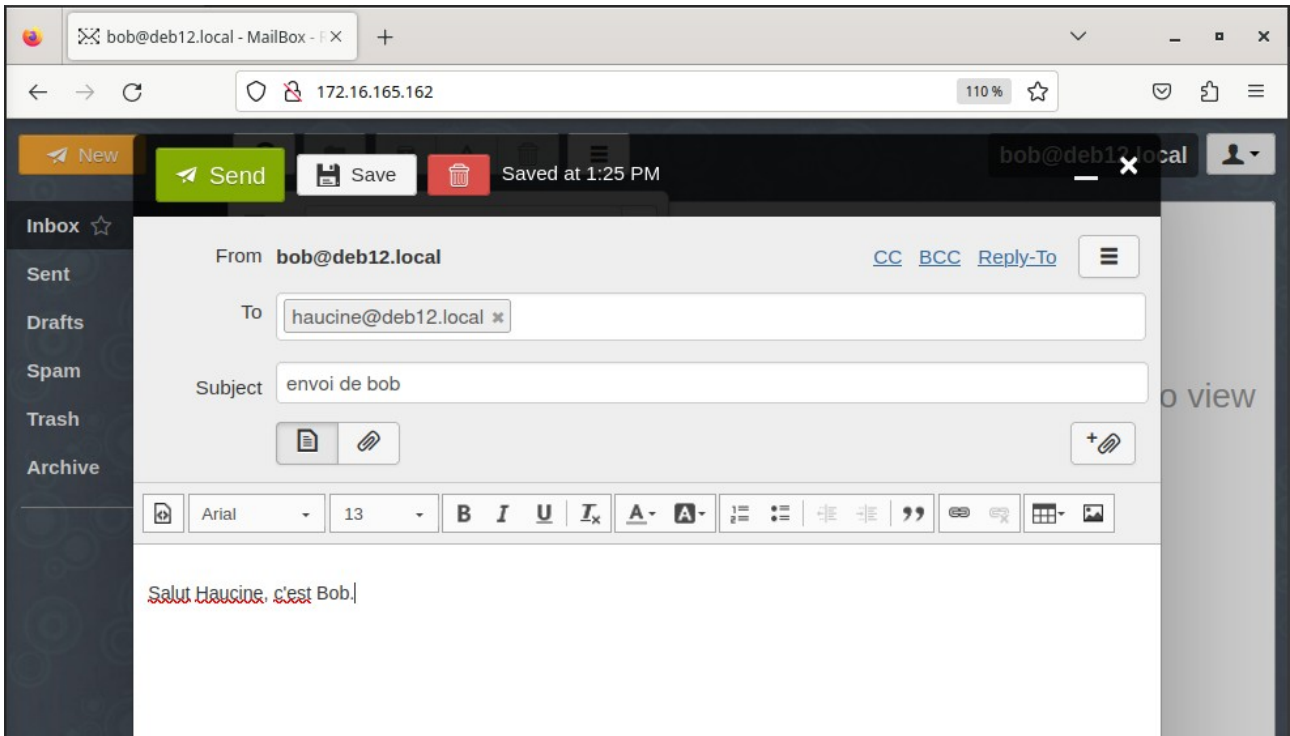
14.5 Création d'un nouvel utilisateur "bob"

Ensuite on va se connecter en tant qu'utilisateur « bob » à Rainloop pour envoyer un mail :

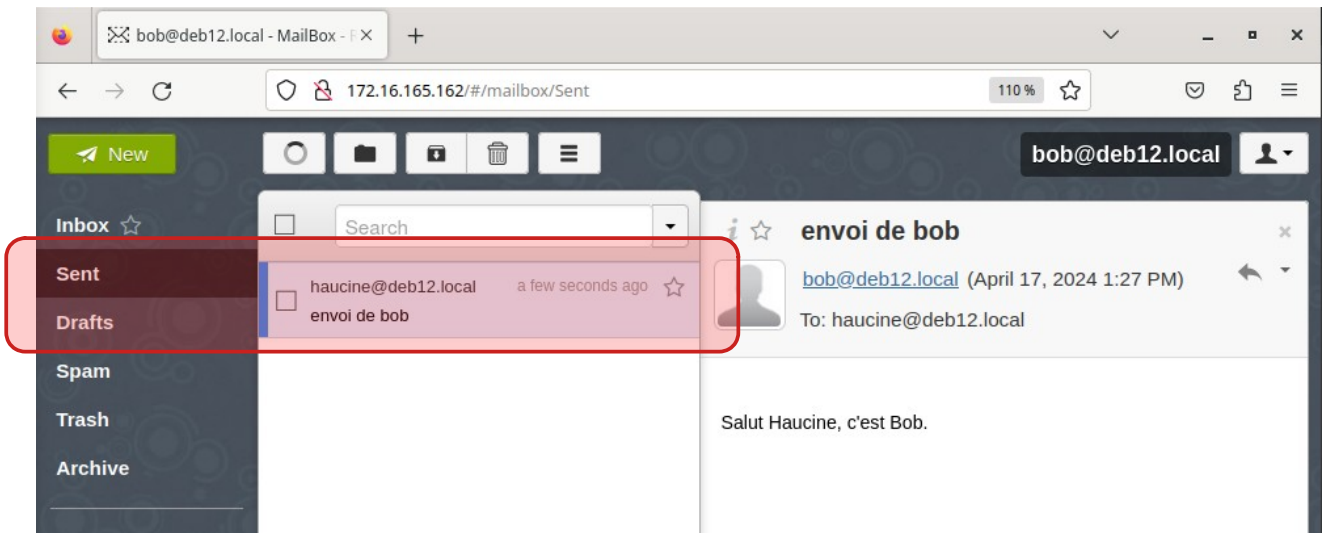


Ci-dessus, on est bien connecté au webmail Rainloop avec le login bob.

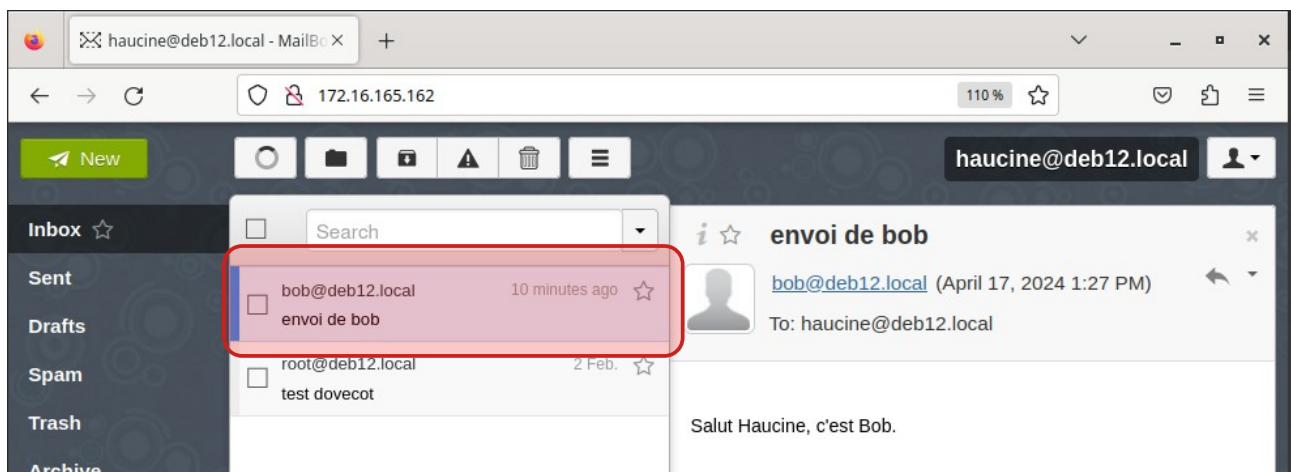
On va faire l'envoi d'un mail de bob à haucine(j'ai crée cet utilisateur lors de l'installation du server) :



On constate que le mail figure bien dans la liste des messages envoyés par bob :



Il ne reste plus qu'à vérifier que l'utilisateur haucine a bien reçu le mail de bob. On va donc se déconnecter de la session de **bob** et se connecter en tant que **haucine** :



Le mail de bob est bien arrivé dans la boîte de réception des mails de haucine, notre webmail fonctionne correctement.

F. Mise en place d'un script pour la sauvegarde des mails

On va créer une tâche **cron** qui va lancer un script de commande **scp** sur un serveur distant. Le dossier des mails de chaque utilisateur sera ainsi sauvegardé à intervalles réguliers. Les utilisateurs ont été créés sur le serveur distant, et il faut à présent créer une paire de clés SSH.

Création de clés SSH :

```
ssh-keygen -t rsa -b 4096
```

Copie des clés publiques sur le serveur distant :

```
ssh-copy-id -i ~/.ssh/cle_root.pub root@172.16.165.170
```

Créer le dossier `/root/.ssh` et le fichier `authorized_keys` manuellement sur le serveur distant, et y copier les clés publiques si besoin.

Ensuite on crée le fichier avec le script pour effectuer une sauvegarde du dossier **Maildir** de chaque utilisateur :

```
nano script_backup_mail.sh
```

Le script ci-dessous automatise la sauvegarde des répertoires des utilisateurs "**haucine**", "**bob**" et "**marie**" vers un serveur distant. Il crée des répertoires distincts pour chaque utilisateur sur le serveur distant, en ajoutant la date et l'heure de la sauvegarde pour une meilleure gestion des données.

!/ Attention, on veillera également à disposer des droits adéquats sur les dossiers de destination (root devra être membre du groupe de chaque utilisateur, avec les droits en écriture)

```

GNU nano 7.2 /usr/local/bin/script_backup_mail.sh
#!/bin/bash
echo "début du script"
date
echo "debut du script" >> /dev/stderr
echo $(date) >> /dev/stderr
# Répertoires locaux à sauvegarder
local_dirs=("/home/haucine" "/home/bob" "/home/marie")

# Adresse IP du serveur distant
remote_server="172.16.165.170"

# Boucle à travers les répertoires locaux
for dir in "${local_dirs[@]}; do
    # Nom de l'utilisateur à partir du répertoire local
    user=$(basename "$dir")

    # Date et heure actuelles
    current_datetime=$(date +"%Y-%m-%d_%H-%M-%S")

    # Nom du répertoire distant pour la sauvegarde
    backup_remote_dir="root@$remote_server:/home/$user/backup_${current_datetime}.tar.gz"

    # Créer l'archive
    backup_file="${user}_backup_${current_datetime}.tar.gz"
    tar -czf "/tmp/$backup_file" "$dir"

    # Transférer l'archive vers le répertoire distant / la directive -i force la clé à utiliser pour la connexion SSH
    scp -i /root/.ssh/cle_root -v "/tmp/$backup_file" "$backup_remote_dir"

    # Supprimer l'archive locale
    rm "/tmp/$backup_file"
done

```

Puis on donne les droits d'exécution à ce fichier :

chmod +x script_backup_mail.sh

```

root@maildeb12:/usr/local/bin# nano script_backup_mail.sh
root@maildeb12:/usr/local/bin# chmod +x script_backup_mail.sh
root@maildeb12:/usr/local/bin# ls -la
total 12
drwxr-xr-x  2 root root 4096 26 mai  12:39 .
drwxr-xr-x 10 root root 4096 31 janv. 08:36 ..
-rwxr-xr-x  1 root root  841 26 mai  12:39 script_backup_mail.sh
root@maildeb12:/usr/local/bin#

```

Le fichier dispose bien des droits d'exécution à présent.

On utilisera la commande tar avec l'option x pour extraire les fichiers sauvegardés:

tar -xvf fichier_archive.tar.gz

Expliquons les options utilisées :

- -x: Cette option indique à tar d'extraire les fichiers de l'archive.
- -v: Cela active le mode verbeux, ce qui affiche les fichiers pendant qu'ils sont extraits.
- -f: Cette option spécifie le nom du fichier archive.

Maintenant, il faut créer la tâche **cron** afin que le script se lance automatiquement à intervalles réguliers.

On vérifie tout d'abord que le service est bien démarré :

systemctl status cron

```

root@maildeb12:/usr/local/bin# sudo systemctl status cron
● cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-05-26 08:40:32 CEST; 4h 15min ago
     Docs: man:cron(8)
  Main PID: 929 (cron)
    Tasks: 1 (limit: 4581)
   Memory: 440.0K
      CPU: 237ms
   CGroup: /system.slice/cron.service
           └─929 /usr/sbin/cron -f

mai 26 12:09:01 maildeb12.deb12.local CRON[11205]: pam_unix(cron:session): session closed for user root
mai 26 12:17:01 maildeb12.deb12.local CRON[11373]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
mai 26 12:17:01 maildeb12.deb12.local CRON[11374]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)

```

Créer une nouvelle tâche cron :

crontab -e

on choisit son éditeur : **1** pour **nano** et **2** pour **vim**

```

root@maildeb12:/usr/local/bin# crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.tiny

Choose 1-2 [1]: 1

```

On renseigne la tâche dans le fichier :

0 0,12 * * * bash /usr/local/bin/script_backup_mail.sh

```

GNU nano 7.2 /tmp/crontab.WEr3Nw/crontab *
0 0,12 * * * bash /usr/local/bin/script_backup_mail.sh
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task

```

Cette directive cron spécifie une tâche planifiée qui exécute un script de sauvegarde des utilisateurs deux fois par jour.

- **0 0,12 * * *** : Cette partie spécifie quand la tâche doit être exécutée :
 - **0** : Signifie que la tâche sera exécutée au début de l'heure.
 - **0,12** : Indique que la tâche sera exécutée deux fois par jour, à minuit (0 heure) et à midi (12 heures).
 - *** * *** : Les trois astérisques signifient que la tâche sera exécutée tous les jours du mois, tous les mois de l'année et tous les jours de la semaine.
- **bash /usr/local/bin/script_backup_mail.sh** : C'est la commande qui sera exécutée à chaque fois que la tâche planifiée sera déclenchée. Dans ce cas, elle exécute le script **script_backup_mail.sh** situé dans le chemin spécifié en utilisant le shell bash.

On vérifie nos tâches cron :
crontab -l

```
root@maildeb12:/usr/local/bin# crontab -l
0,12 * * * bash /usr/local/bin/script_backup_mail.sh

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
```

On constate ci-dessus que notre tâche cron est bien mise en place.

G. Ecoute de POSTFIX sur ports 465 et 587

Maintenant que notre serveur mail est fonctionnel et que nous avons mis en place une tâche **cron** pour la sauvegarde des dossiers contenant les emails, on va configurer **POSTFIX** pour qu'il puisse envoyer les mails de manière sécurisée si besoin via les ports **465** ou **587**.

Ouvrir, décommenter et/ou rajouter les lignes ci-dessous du fichier **/etc/postfix/master.cf** :

```
GNU nano 7.2 /etc/postfix/master.cf *
#tlsproxy unix - - y - 0 tlsproxy
# Choose one: enable submission for loopback clients only, or for any client.
#127.0.0.1:submission inet n - y - - smtpd

##### Config port 587 #####
submission inet n - y - - smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_tls_wrappermode=no
  -o smtpd_sasl_auth_enable=yes
# -o smtpd_tls_auth_only=yes
# -o smtpd_reject_unlisted_recipient=no
# Instead of specifying complex smtpd_<xxx>_restrictions here,
# specify "smtpd_<xxx>_restrictions=$mua_<xxx>_restrictions"
# here, and specify mua_<xxx>_restrictions in main.cf (where
# "<xxx>" is "client", "helo", "sender", "relay", or "recipient").
# -o smtpd_client_restrictions=permit_sasl_authenticated,reject
# -o smtpd_helo_restrictions=
# -o smtpd_sender_restrictions=
# -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
# -o smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_authenticated,reject
# -o smtpd_sasl_type=dovecot
# -o smtpd_sasl_path=private/auth
# -o milter_macro_daemon_name=ORIGINATING
# Choose one: enable submissions for loopback clients only, or for any client.
#127.0.0.1:submissions inet n - y - - smtpd

##### Config port 465 #####
smtps inet n - y - - smtpd
  -o syslog_name=postfix/smtps
```



```
GNU nano 7.2 /etc/postfix/master.cf *
##### Config port 465 #####
smtps      inet  n       -       y       -       -       smtpd
  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
#  -o smtpd_reject_unlisted_recipient=no
#  Instead of specifying complex smtpd_<xxx>_restrictions here,
#  specify "smtpd_<xxx>_restrictions=$mua_<xxx>_restrictions"
#  here, and specify mua_<xxx>_restrictions in main.cf (where
#  "<xxx>" is "client", "helo", "sender", "relay", or "recipient").
#  -o smtpd_client_restrictions=
#  -o smtpd_helo_restrictions=
#  -o smtpd_sender_restrictions=
#  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
#  -o smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_authenticated,reject
#  -o smtpd_sasl_type=dovecot
#  -o smtpd_sasl_path=private/auth
#  -o milter_macro_daemon_name=ORIGINATING

#628      inet  n       -       y       -       -       qmqpd
```

La directive **-o milter_macro_daemon_name=ORIGINATING** améliore la gestion des courriels en fournissant un contexte supplémentaire aux Milters, permettant ainsi un filtrage plus précis, une surveillance facilitée, et une meilleure application des politiques de sécurité et de performance.

On redémarre **POSTFIX** :
systemctl restart postfix

Taper cette commande pour vérifier les ports en écoute de **POSTFIX** :
ss tlnp | grep "master" ← **master** correspond à **POSTFIX** dans notre système

```
root@maildeb12:~# ss -tlnp | grep "master"
LISTEN 0      100          0.0.0.0:587   0.0.0.0:*    users:(("master",pid=4984,fd=18))
LISTEN 0      100          0.0.0.0:465   0.0.0.0:*    users:(("master",pid=4984,fd=22))
LISTEN 0      100          0.0.0.0:25    0.0.0.0:*    users:(("master",pid=4984,fd=13))
LISTEN 0      100          [::]:587     [::]:*      users:(("master",pid=4984,fd=19))
LISTEN 0      100          [::]:465     [::]:*      users:(("master",pid=4984,fd=23))
LISTEN 0      100          [::]:25      [::]:*      users:(("master",pid=4984,fd=14))
root@maildeb12:~# █
```

On constate qu'à présent **POSTFIX** écoute aussi sur les ports sécurisés 465 et 587 en plus du port 25.

Quelques précisions supplémentaires :

Port 465 (SMTPS)

- **Sécurité** : La connexion commence directement avec **SSL/TLS**, assurant que toute la communication est chiffrée dès le départ.
- **Compatibilité** : **Moins flexible**, car il ne permet **que** les **connexions sécurisées**. Certains serveurs de messagerie modernes peuvent ne pas le supporter, car son utilisation est obsolète.
- **Normes** : Le port **465** n'est plus un standard recommandé par **l'IETF**. L'utilisation de ce port a été déconseillée au profit du port **587**.

Port 587 (Submission with STARTTLS)

- **Sécurité** : La connexion commence en clair, mais peut être immédiatement surclassée en une connexion sécurisée avec la commande **STARTTLS**. Cela permet une communication sécurisée similaire à celle du port 465 une fois la connexion sécurisée établie.
- **Flexibilité** : **STARTTLS** offre **plus de flexibilité** car il permet à un serveur d'accepter **à la fois** des **connexions sécurisées et non sécurisées** sur le même port. Cela facilite la transition vers des communications sécurisées sans forcer tous les clients à utiliser **TLS** dès le départ.
- **Normes** : Le port **587** est le standard actuel recommandé par **l'IETF** pour la soumission des courriels avec authentification, tel que défini dans la **RFC 6409**.

Le port **587** est recommandé pour les nouvelles implémentations car il est plus flexible et conforme aux standards modernes.

Bien que le port **465** établisse une connexion sécurisée dès le départ, la sécurité offerte par le port **587** avec **STARTTLS** est équivalente lorsqu'elle est correctement mise en œuvre.

De plus, le port **587** assure une meilleure compatibilité avec divers clients et serveurs de messagerie, ce qui facilite les communications sécurisées dans des environnements **hétérogènes**.