PostFix : serveur de messagerie sous Linux

Cet article est destiné aux débutants qui veulent configurer un **serveur de messagerie électronique** de base. Il est préférable d'avoir une connaissance élémentaire en administration système, ainsi que la capacité d'installer des logiciels et de modifier des fichiers de configuration. L'article a été rédigé en se basant sur Debian 11, mais les instructions devraient également convenir aux autres versions. Veuillez noter que des différences peuvent exister dans les autres versions.

Postfix est un logiciel de serveur de messagerie open source largement adopté. En tant que “MTA” (Agent de Transfert de Message), il joue un rôle central dans le traitement, la transmission et la distribution des courriels. Doté de fonctionnalités avancées en matière de sécurité, de filtrage et de personnalisation, Postfix est un choix prisé pour la gestion des systèmes de messagerie.

- Réécriture des adresses e-mail sortantes dans Postfix
- Configurer un Script de Hook Let's Encrypt pour Postfix
- Configuration de Postfix avec un relais SMTP externe utilisant l'authentification LOGIN ou PLAIN

1. Introduction

L'objectif fondamental de cette procédure est de permettre à n'importe quelle machine ou serveur d'envoyer des courriels vers une adresse spécifique. Pour y parvenir, il est nécessaire de préparer le courrier électronique à l'aide d'un programme externe, puis de le transmettre efficacement au serveur de messagerie de destination en utilisant le protocole SMTP (Simple Mail Transfer Protocol).

Le processus de l'envoi de courriel via SMTP s'articule comme suit, prenons un exemple concret avec un courriel destiné à l'adresse alice@example.com :

1. L'utilisateur ou un programme externe crée le courrier électronique, en spécifiant les informations du destinataire (alice@example.com), en rédigeant le contenu du message et en incluant d'autres détails nécessaires.

2. Le courriel est ensuite remis au serveur SMTP local, qui se trouve sur la machine ou le serveur à partir duquel l'envoi est effectué.

3. Le serveur SMTP analyse le domaine du destinataire (dans ce cas, “example.com”) pour déterminer comment atteindre le serveur de messagerie de destination.

4. Le serveur SMTP établit un contact avec le serveur de messagerie de destination (le serveur SMTP de “example.com” dans cet exemple) en utilisant le protocole SMTP.

5. Le serveur de messagerie de destination accepte le courriel, le stocke temporairement, puis le transfère éventuellement dans la boîte aux lettres de l'utilisateur Alice, située sur son propre serveur

de messagerie.

6. Si tout se déroule sans problème, le courriel est ainsi livré avec succès à Alice, qui peut alors le consulter dans sa boîte de réception.

Ce processus est la façon dont le protocole SMTP assure la transmission de courriels, encheminant ces derniers de l'expéditeur au destinataire, en utilisant les serveurs de messagerie appropriés à travers Internet.

1.1 Axe de travail

Il existe de nombreuses configurations et combinaisons différentes possibles lors de la mise en place d'un serveur de messagerie électronique, bien trop nombreuses pour être toutes couvertes ici. Par conséquent, cet article effectue certaines choix fondamentaux pour vous, tels que les logiciels que nous allons utiliser (Postfix et Dovecot). D'autres options nécessiteront des modifications de la part de l'utilisateur, comme les adresses réseau et les noms de domaine. Les paramètres plus avancés, comme la gestion de domaines virtuels et des utilisateurs, ne sont pas abordés dans cet article et ne seront pas traités ici.

Dans ce contexte, nous utilisons **Postfix** comme agent de transfert de messagerie (MTA). **Dovecot** est utilisé pour permettre aux utilisateurs d'accéder à leur courrier électronique via les protocoles **IMAP** ou **POP**. Nous partons du principe que le nom de domaine utilisé est **example.com**, mais cela devrait être adapté par le lecteur. Vous pouvez utiliser un véritable nom de domaine pour un serveur de messagerie pleinement qualifié ou un faux nom de domaine si vous souhaitez uniquement créer un serveur de messagerie interne.

Notre exemple suppose que le serveur de messagerie physique (hôte) porte le nom **mail.example.com** et est situé à l'adresse IP privée **192.168.0.1** (veuillez personnaliser ces informations en fonction de vos besoins). Le serveur de messagerie fournira des comptes de messagerie basés sur les comptes système d'utilisateurs standards, et les utilisateurs accéderont à leur courrier en utilisant leur nom d'utilisateur et leur mot de passe de compte système. Nous illustrons cela avec un utilisateur nommé **John Smith**, qui dispose d'un compte système avec le nom d'utilisateur **john**.

1.2 Serveurs SMTP

Sous Linux Debian, il existe plusieurs programmes d'envoi de courriels, chacun avec ses propres fonctionnalités et avantages. Voici quelques-uns des programmes les plus couramment utilisés pour envoyer des courriels sous Debian :

1. **ssmtp**: Simple SMTP est un programme léger qui permet d'envoyer des courriels via SMTP. Il est particulièrement adapté aux tâches d'envoi de courriels automatisées et ne prend pas en charge la réception de courriels.

2. **msmtp**: MSMTP est un autre client SMTP léger qui facilite l'envoi de courriels depuis la ligne de commande ou depuis des scripts. Il peut être configuré pour transmettre des courriels à travers un serveur SMTP externe.

3. **Postfix**: Bien que Postfix soit principalement un serveur de messagerie, il peut également être

utilisé pour envoyer des courriels depuis une machine Debian. Il offre une grande flexibilité en matière de configuration, mais sa configuration peut être plus complexe que celle des clients SMTP plus simples.

4. **sendmail**: Sendmail est un programme de messagerie historique sous Unix/Linux, bien qu'il soit maintenant souvent remplacé par des alternatives plus modernes. Cependant, il est toujours disponible sur Debian et peut être utilisé pour envoyer des courriels.

5. **Exim**: Exim est un autre serveur de messagerie qui peut être configuré pour envoyer des courriels. Il est également capable de gérer la réception de courriels, ce qui en fait une option plus complète.

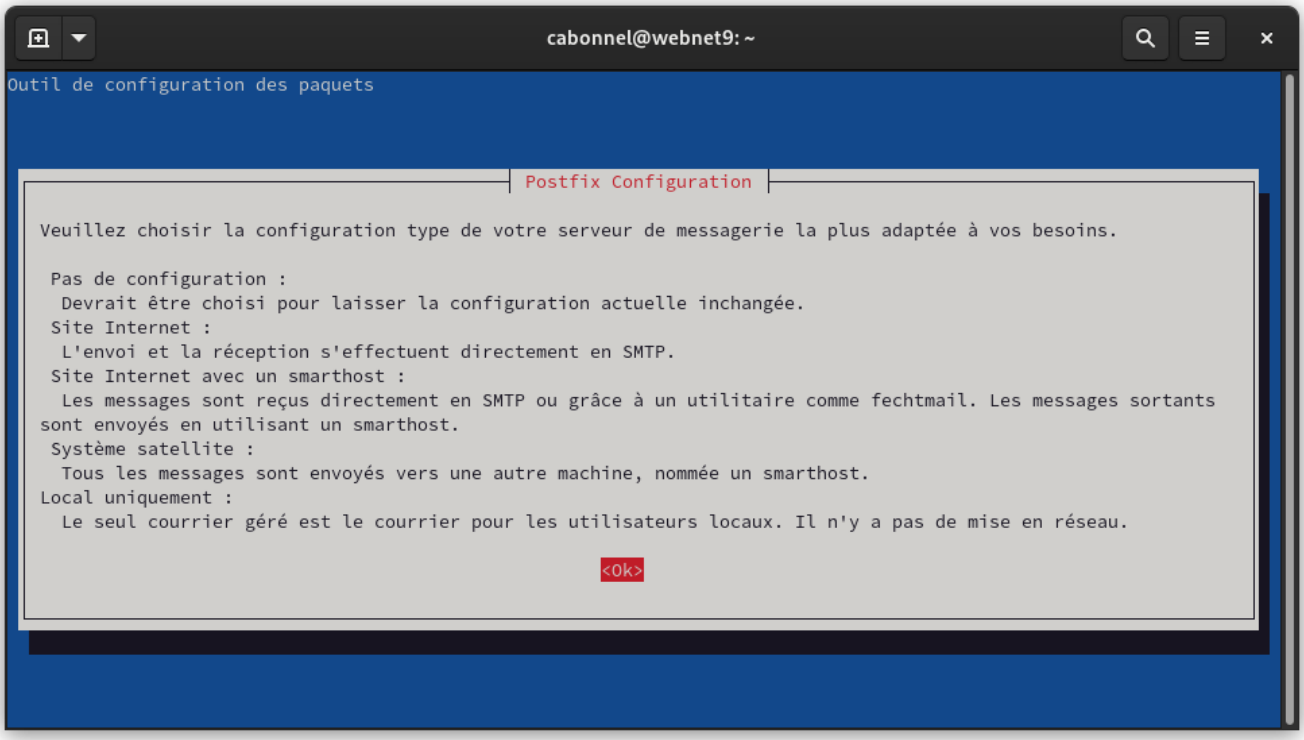
Le choix du programme d'envoi de courriels dépendra de vos besoins spécifiques, de votre niveau de confort avec la configuration et de la complexité de votre infrastructure de messagerie. Pour des tâches simples d'envoi de courriels depuis la ligne de commande ou depuis des scripts, **ssmtp** ou **msmtp** sont souvent des choix pratiques. Pour des besoins plus avancés, **Postfix** ou **Exim** peuvent être mieux adaptés.

2. Installer Postfix

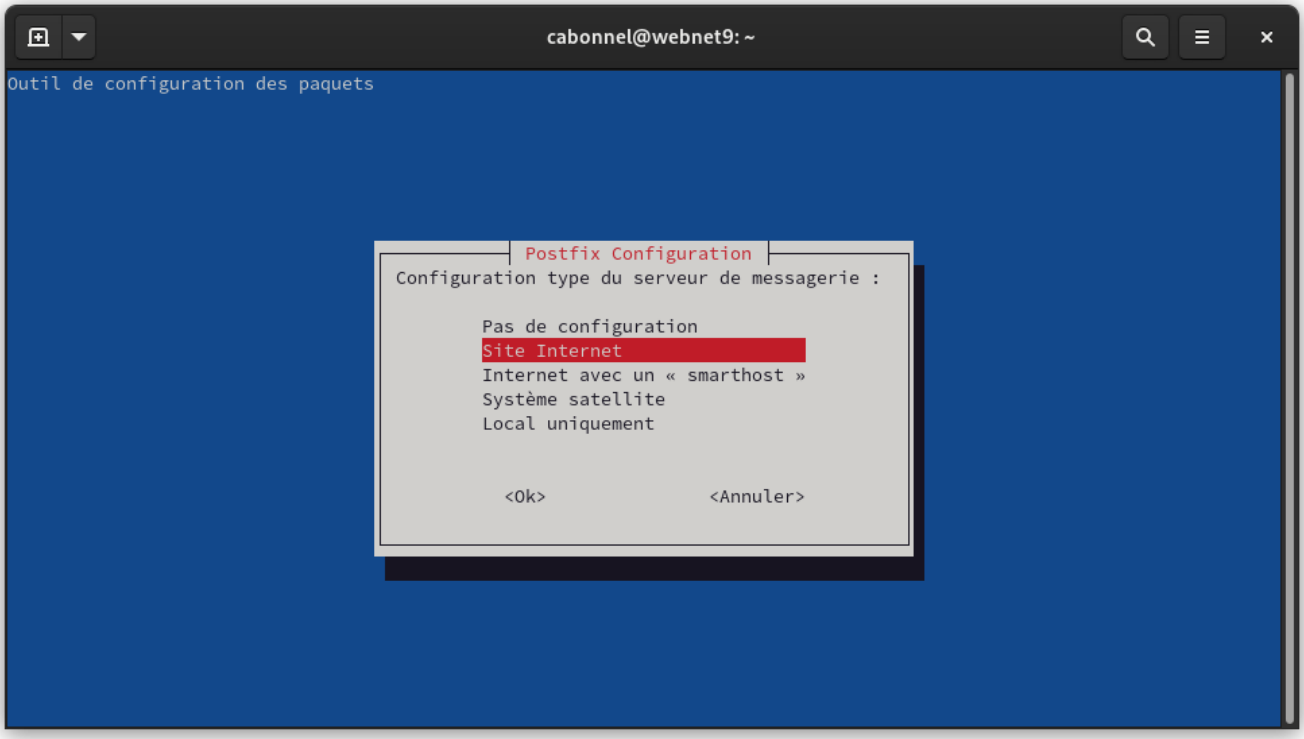
Pour installer Postfix sur Debian, vous devez utiliser le gestionnaire de paquets APT (Advanced Package Tool). Voici comment vous pouvez procéder :

```
sudo apt update
sudo apt install postfix
```

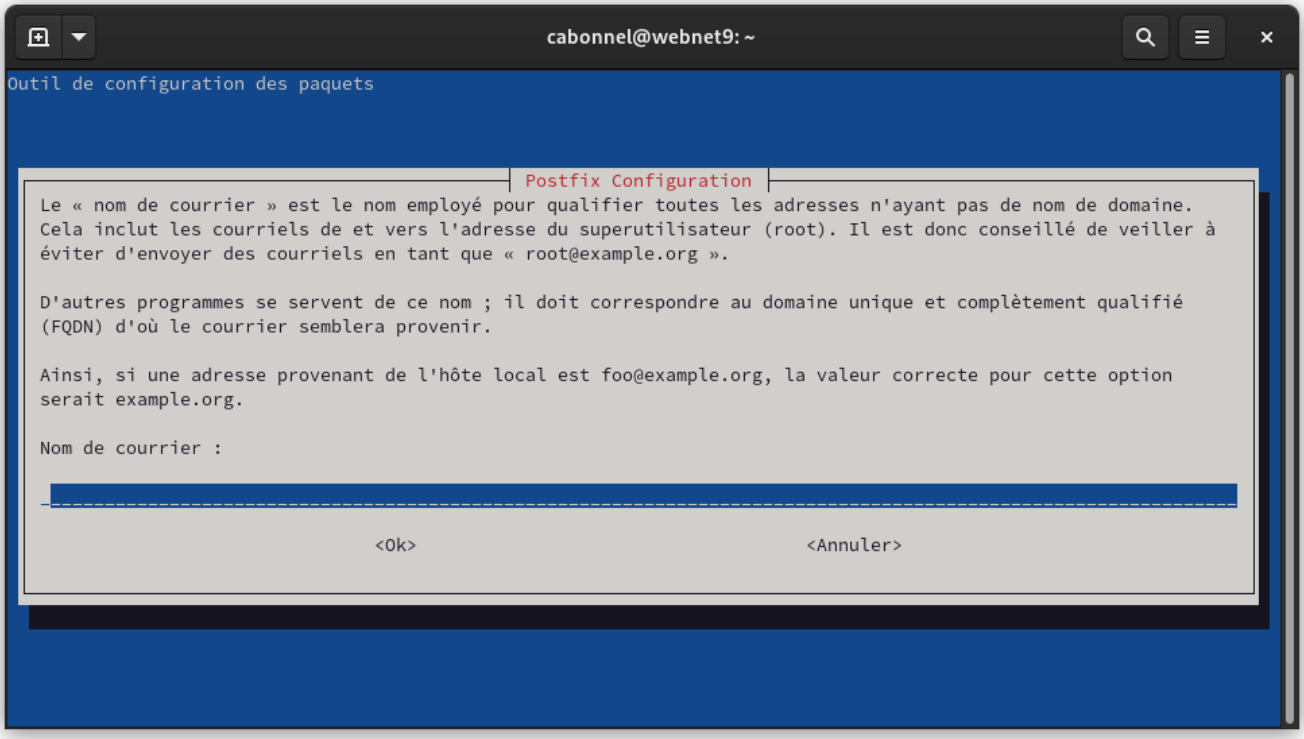
La première commande `apt update` met à jour la liste des paquets disponibles dans les dépôts Debian, et la deuxième commande “apt install” installe Postfix ainsi que ces dépendances.



Choisir `Site Internet`



Entrer la valeur FQDN de votre adresse de serveur



si vous devez relancer la configuration de Postfix

```
sudo dpkg-reconfigure postfix
```

Pour supprimer **Sendmail**, vous pouvez utiliser la commande suivante :

```
sudo apt-get remove sendmail
```

Cette commande supprime le programme **Sendmail** de votre système Debian.

Après avoir installé **Postfix**, vous devrez configurer ces logiciels pour les adapter à vos besoins spécifiques.

3. Configurer Postfix

Les fichiers de configuration de postfix sont stockés dans /etc/postfix. Les deux principaux fichiers de configuration de postfix sont master.cf et main.cf, bien que nous ne traiterons que de main.cf ici. Tout d'abord, nous allons ajouter ou modifier certaines lignes dans le fichier de configuration main.cf. Les lignes suivantes doivent être ajoutées, modifiées ou décommentées :

```
myhostname = mail.example.com
mydomain = example.com
myorigin = $mydomain
inet_interfaces = all
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
mynetworks = 192.168.0.0/24, 127.0.0.0/8
relay_domains =
home_mailbox = Maildir/
```

4. Tests

Faire un essai d'envoi de mail

```
echo "Le contenu du mail" | mail -s "ceci est le sujet" mail@domaine.tld
```

Le programme **mail** est une composante du package **mailutils**. Donc, si le programme n'est pas installer sur la machine, utilisez `sudo apt install mailutils`

–

Pour modifier un paramètre dans **Postfix**, il faut éditer le fichier de configuration

```
sudo nano /etc/postfix/main.cf
```

Redémarrer le service

```
sudo systemctl restart postfix
```

Gestion des Alias

Ajouter dans le fichier de configuration de **Postfix**, `/etc/postfix/main.cf`

```
virtual_alias_maps = hash:/etc/postfix/virtual
```

Puis ajouter dans le fichier `/etc/postfix/virtual` les **alias désirés** tel que le modèle suivant :

```
@monserveur.tld mail@domain.tld
```

Enfin, exécuter le bloc suivant. Il sera nécessaire de l'exécuter à chaque modifications effectuées du fichier `/etc/postfix/virtual`.

```
sudo postmap /etc/postfix/virtual
sudo systemctl restart postfix
```

Mails en attente

Pour connaître les mails en attente

```
sudo postqueue -p
```

–

Pour traiter tous les mails en attente

```
sudo postqueue -f
```

–

Pour supprimer tous les mails en attente

```
sudo postsuper -d ALL
```

Reprise de la configuration de Postfix

Le fichier de configuration de **Postfix** est `/etc/postfix/main.cf`. Il est éditable par **nano** ou **vim**. On va le reprendre pour configurer **Postfix**.

—

```
myhostname = myserver.example.com
```

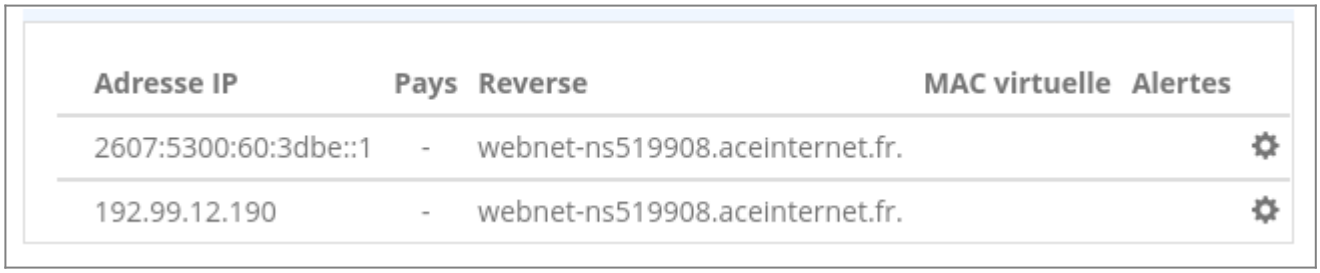
Il est important que l'option `myhostname` corresponde au FQDN (fully qualified domain name) du serveur. La valeur à renseigner et celle qui renvoyée par la commande :

```
nslookup 91.134.243.56
```

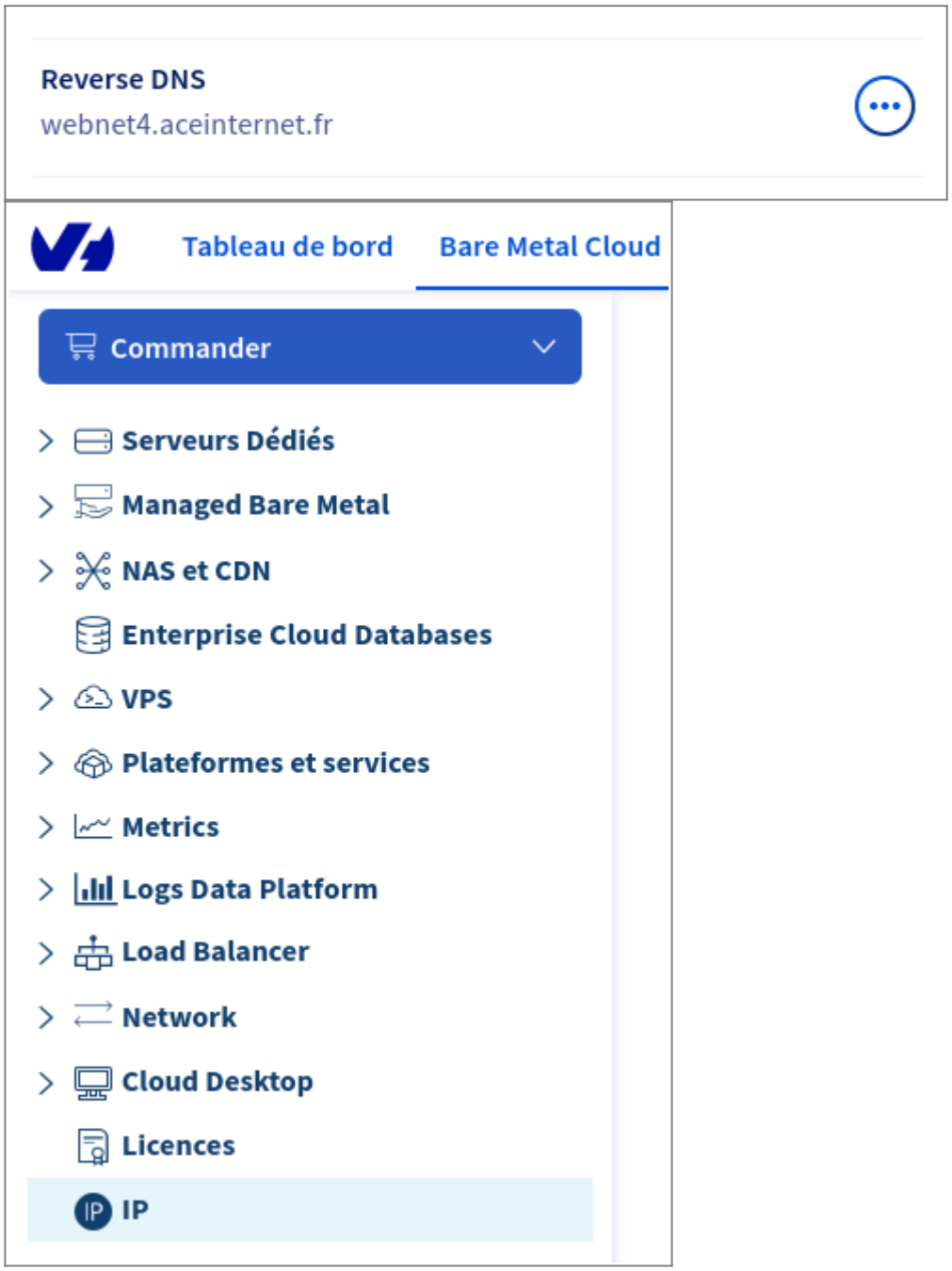
```
(87) cedric24c@dskcdc001:~ $ nslookup 91.134.243.56
56.243.134.91.in-addr.arpa      name = webnet-711770.aceinternet.fr.

Authoritative answers can be found from:
```

Cette information est gérée par le serveur DNS

Cette option se trouve les paramètres `Gérer les paramètres IPs`, chez **kimsufi.com**



Cette option, **reverse DNS**, se trouve dans les options du **serveur VPS** de vos **serveurs dédiés**, chez **ovh.com**



—

Configurer le **nom du serveur SMTP**, domaine à afficher dans le courrier sortant

```
myorigin = example.com
```

Configuer le **nom du serveur SMTP**

```
mydomain = example.com
```

Configure to which SMTP domains to relay messages to, for example:

```
relay_domains = example.com
```

—

Configuration minimaliste du SMTP Greeting Banner:

```
smtpd_banner = $myhostname
```

—

Limiter les attaques par déni de services :

```
default_process_limit = 100
smtpd_client_connection_count_limit = 10
smtpd_client_connection_rate_limit = 30
queue_minfree = 20971520
header_size_limit = 51200
message_size_limit = 10485760
smtpd_recipient_limit = 100
```

Consulter le fichier log

Le fichier log standard de **postfix** est `/var/log/mail.info`

Vous pouvez garder un oeuil sur les logs

```
sudo tail -f /var/log/mail.info&
```

Envoyer un mail

Il y a deux possibilités :

- envoie depuis un client : **mail**
- connexion en **Telnet** sur le serveur SMTP

L'utilitaire **mail** fait parti de la suite **mailutils**

```
sudo apt install mailutils
```

—

Utilisation de l'utilitaire **mail** depuis un poste client.

Pour envoyer un mail à `mail@example.com` de la part de `sender@example.com`

```
echo "This is the message body" | mail -s "This is the subject"
mail@example.com -aFrom:sender@example.com
```

Pour envoyer un mail à `username`

```
 echo "This is the message body" | mail -s "Hello World" username
```

—

Utilisation de **telnet** pour se connecter sur le serveur SMTP `mail.mymailserver.com`

```
telnet mail.mymailserver.com 25
```

Puis saisir les commandes SMTP

```
EHLO checkeremail.com
```

```
MAIL FROM:<sender@mailserver.fr>
```

```
RCPT TO:<dest@mymailserver.com>
```

```
DATA
Subject: Sending an email using telnet
Hello,
Here is my body? Do you like it?
Cédric
.
```

```
QUIT
```

Vider tous les mails

Vider tous les mails présents dans la boite d'un utilisateur. On considère que la boite mail (mbox) de l'utilisateur `www-data` se trouve dans le fichier `/var/mail/www-data`

```
sudo sh -c "> /var/mail/www-data"
```

Gestion des certificats

Pour configurer **Postfix** et **Certbot** pour utiliser les certificats SSL/TLS de “smtp.monserveur.fr” avec **Let's Encrypt**, suivez ces étapes générales. Assurez-vous d'avoir les droits nécessaires sur le serveur et que vous êtes à l'aise avec l'édition de fichiers de configuration en ligne de commande.

Configurer Postfix pour utiliser SSL/TLS

1. **Accédez à la configuration de Postfix**:

- Connectez-vous à votre serveur en tant que sudouser.
- Ouvrez le fichier de configuration principal de Postfix avec un éditeur de texte, tel que `nano` ou `vim`. Le fichier est généralement situé à `/etc/postfix/main.cf`.

2. **Définissez les chemins des certificats**:

- Localisez ou ajoutez les lignes suivantes dans `main.cf` pour spécifier l'emplacement des fichiers de certificat et de clé privée (remplacez les chemins par les vôtres si nécessaire) :

```
smtpd_tls_cert_file = /etc/letsencrypt/live/smtp.monserveur.fr/fullchain.pem
smtpd_tls_key_file = /etc/letsencrypt/live/smtp.monserveur.fr/privkey.pem
```

- Activez l'utilisation de TLS en ajoutant ou en s'assurant que la ligne suivante est présente :

```
smtpd_use_tls = yes
```

3. **Redémarrez Postfix**:

- Sauvegardez vos modifications et fermez le fichier.
- Exécutez la commande `sudo systemctl restart postfix` pour appliquer les modifications.

Configurer Dovecot pour SSL/TLS

Si vous utilisez Dovecot comme serveur IMAP/POP3 :

1. Les fichiers de configuration de **Dovecot** se trouvent généralement dans `/etc/dovecot/`. Le fichier principal de configuration est souvent nommé `dovecot.conf`, et il peut inclure d'autres fichiers de configuration situés dans `/etc/dovecot/conf.d/`.

2. Dans les fichiers de configuration de **Dovecot**, vous devrez trouver et modifier les lignes qui définissent le chemin du certificat SSL et de la clé privée. Recherchez quelque chose comme ceci :

```
ssl_cert = </etc/letsencrypt/live/smtp.serveur.fr/fullchain.pem
ssl_key = </etc/letsencrypt/live/smtp.serveur.fr/privkey.pem
```

Pensez à désactiver la configuration présente dans `/etc/dovecot/conf.d/10-ssl.conf`.

3. **Redémarrez Dovecot** avec `sudo systemctl restart dovecot`.

4. Après le redémarrage, assurez-vous que tout fonctionne comme prévu. Vous pouvez vérifier que **Dovecot** écoute avec le nouveau certificat en vous connectant avec un client de messagerie ou en utilisant OpenSSL :

```
openssl s_client -connect smtp.serveur.fr:993
```

Configurer Let's Encrypt pour le renouvellement automatique

1. **Certbot** gère généralement les renouvellements automatiquement. Cependant, vous pouvez personnaliser ou ajouter des scripts de renouvellement dans le dossier de hooks de renouvellement.

2. **Scripts de renewal-hooks**:

- Placez les scripts personnalisés dans `/etc/letsencrypt/renewal-hooks/`. Vous pouvez avoir des scripts `pre`, `deploy`, et `post` pour s'exécuter avant, pendant, et après le renouvellement.
- Un script typique dans `post` pourrait redémarrer Postfix et Dovecot pour appliquer les nouveaux certificats.

Voir les pages :

- Configurer un Script de Hook Let's Encrypt pour Postfix
- Créer un script de Hook Let's Encrypt pour Dovecot

Si vous avez deux scripts distincts, `reload-dovecot.sh` et `reload-postfix.sh`, et vous souhaitez exécuter les deux après le renouvellement de certificat **Let's Encrypt** par **Certbot**, vous pouvez configurer les hooks dans le fichier de configuration de renouvellement de **Certbot** ou les placer dans les répertoires de hook appropriés.

Vous devriez ajouter des lignes pour post_hook dans la section `[renewalparams]`. Votre fichier pourrait ressembler à ceci :

```
[renewalparams]
post_hook = /etc/letsencrypt/renewal-hooks/post/reload-dovecot.sh &&
/etc/letsencrypt/renewal-hooks/post/reload-postfix.sh
```

3. **Tester le renouvellement**:

- Exécutez `sudo certbot renew –dry-run` pour tester le processus de renouvellement et s'assurer que tout fonctionne comme prévu.

Vérification et maintenance

- **Vérifiez les logs** de **Postfix** et **Dovecot** pour les erreurs liées aux certificats SSL/TLS.
- Assurez-vous que les certificats se renouvellent correctement en vérifiant les dates d'expiration et en observant le comportement du système lors des renouvellements planifiés.

Remarques

- Faites toujours une copie de sauvegarde des fichiers de configuration avant de les modifier.
- Les chemins exacts et les commandes peuvent varier légèrement en fonction de votre distribution Linux et de la version de vos logiciels.
- Assurez-vous que les ports nécessaires sont ouverts sur votre pare-feu pour permettre les connexions TLS/SSL.

En suivant ces étapes, vous devriez être capable de configurer **Postfix** et **Dovecot** pour utiliser les certificats SSL/TLS avec **Let's Encrypt**, améliorant ainsi la sécurité de votre serveur de messagerie. Assurez-vous de tester votre configuration pour vérifier que tout fonctionne correctement avant de la mettre en production.

Biblio

- https://www.tecmint.com/install-postfix-mail-server-with-webmail-in-debian/
- https://wiki.centos.org/HowTos(2f)postfix.html

From:
https://www.abonnel.fr/ - **notes informatique & technologie**

Permanent link:
https://www.abonnel.fr/informatique/serveur/postfix

Last update: **2023/12/29 18:29**