

Serveur de messagerie autonome avec Postfix et Fetchmail

Charles Vejnár
chv@fr.st

Ce document concerne la mise en oeuvre de l'ensemble d'un serveur de messagerie. J'utilise une Mandrake® dont Postfix est le gestionnaire de messagerie par défaut et Fetchmail le récupérateur mais les explications qui suivent sont compatibles avec une RedHat®.

Ce document est mis à votre disposition grâce à mon travail et à mon hébergeur. Il est soumis aux termes de la GNU FDL et je vous prie de m'envoyer un courriel pour toutes références ou commentaires à <chv@fr.st>. Les informations contenues dans ces pages ne sont aucunement garanties.

Ce document est disponible aux formats SGML¹, DVI² et PDF³. Sa version la plus récente est disponible sur Linux-France⁴ ou mon site personnel⁵.

Table des matières

1. Introduction & Installation	1
1.1. Principes de l'installation.....	2
1.2. Installation des packages.....	2
1.3. Quelques vérifications.....	2
2. Configuration	2
2.1. Configuration de Postfix.....	2
2.1.1. Postfix de base.....	2
2.1.2. Postfix vers l'extérieur.....	3
2.1.3. Réécriture des adresses.....	3
2.1.4. Test de la configuration.....	3
2.2. Configuration de Fetchmail.....	3
2.3. Configuration de la connexion.....	4
2.4. Assemblage du tout.....	4
3. Utilisation du serveur	5
3.1. Administration à distance.....	5
3.1.1. Administration par SSH (Secure SHell).....	5
3.1.2. Administration par Webmin.....	5
3.2. Envoi régulier d'une page web.....	5
3.3. Etapes d'ajout d'un compte de messagerie.....	5
4. Statistiques des connexions et des courriers	6
4.1. Sur le serveur : envoi des logs.....	6
4.1.1. Scripts et programmes.....	6
4.1.2. Configuration de Logrotate.....	6
4.2. Sur un poste Linux : analyse des logs.....	7
4.2.1. Analyse des connexions.....	7
4.2.2. Analyse des échanges de courrier.....	7
5. Remarques générales et techniques	7

1. Introduction & Installation

1.1. Principes de l'installation

Le but de cet article est d'accompagner le lecteur de manière concise et rapide à configurer un serveur de messagerie autonome. Situé sur une machine indépendante, il se connecte périodiquement et automatiquement grâce à une tâche cron par une connexion analogique avec ppp pour envoyer par SMTP les messages mis en queue et rapatrier ceux présents sur différents comptes POP3 avec Fetchmail.

Cet article ne prétend pas permettre la compréhension du fonctionnement de toutes les applications utilisées mais davantage d'être un plan de mise en oeuvre accompagné de quelques suggestions. Tous les logiciels employés sont cependant assez simple, seul Postfix reclame un peu plus d'explications, je vous conseille pour cela l'article d'Eric Jacoboni sur Linux-France⁶.

1.2. Installation des packages

Installez les packages RPM suivants :

- Postfix, le gestionnaire de messagerie
- Fetchmail, le récupérateur de mails à partir d'un serveur distant
- Fetchmailconf, l'utilitaire de configuration de Fetchmail
- Procmail, le distributeur de courrier
- Ppp, le daemon de connexion à un provider

Vérifier que Sendmail n'est pas installé par **rpm -q sendmail**.

Afin de lancer Postfix à chaque démarrage de la machine, activez le service Postfix du Système V en le cochant dans le menu proposé par la commande **ntsysv**.

1.3. Quelques vérifications

La machine est nommé serveur et appartient au domaine mozart.com ce que l'on peut vérifier :

```
$ hostname
serveur.mozart.com
```

Toutes les manipulations sont effectuées sous le compte root. Les comptes des utilisateurs ont été créés par les commandes **useradd nom_utilisateur** puis **passwd nom_utilisateur**.

2. Configuration

2.1. Configuration de Postfix

2.1.1. Postfix de base

Allez dans le répertoire `/etc/postfix` puis faites une copie du fichier `main.cf`. Ce fichier constitue l'essentiel de la configuration de Postfix. Ouvrez-le avec votre éditeur favori. La plupart des options sont correctement positionnées mais il faut en préciser certaines.

- Pour préciser les noms de machine et de domaine :

```
myhostname = serveur.mozart.com
mydomain = mozart.com
```
- Pour préciser l'envoi puis la réception du courrier :

```
myorigin = $myhostname
mydestination = $myhostname, localhost.$mydomain
```
- Pour préciser le mode de transport des courriers sortants :

```
default_transport = smtp
```
- Pour préciser la gestion des alias :

```
alias_maps = hash:/etc/postfix/aliases
alias_database = hash:/etc/postfix/aliases
```

Pour prendre en compte les changements de configuration, entrez la commande **postfix reload**.

Pour afficher les différences de votre configuration avec celle par défaut, entrez **postconf -n**.

Pour vérifier la configuration du serveur, lancez **postfix check**.

Vérifiez la présence de Postfix et associés par :

```
$ ps axf
 733 ?        S          0:00 /usr/lib/postfix/master
 738 ?        S          0:00 \_ qmgr -l -t fifo -u
1900 ?        S          0:00 \_ pickup -l -t fifo
```

Dès à présent le serveur *local* de messagerie fonctionne.

2.1.2. Postfix vers l'extérieur

Pour ouvrir le serveur vers l'extérieur, il faut lui indiquer la marche à suivre. Pour cela ajoutez les lignes suivantes au fichier `/etc/postfix/main.cf` en remplaçant `mail.provider.com` par l'adresse du serveur SMTP de votre provider (en conservant les crochets) :

```
relayhost = [mail.provider.com]
defer_transports = smtp
```

2.1.3. Réécriture des adresses

Enfin, il faut définir la correspondance entre les adresses locales de type `utilisateur@serveur.mozart.com` qui n'ont aucune valeur pour vos correspondants extérieurs et celles qu'ils connaissent de type `identifiant@provider.com`. Editez le fichier `/etc/postfix/sender_canonical` pour que Postfix remplace l'adresse locale de la forme `utilisateur@serveur.mozart.com` par `identifiant@provider.com` lorsque `utilisateur` envoie un message :

```
utilisateur identifiant@provider.com
```

De même, un message adressé par un utilisateur local à `identifiant@provider.com` ne doit pas être dirigé vers l'extérieur car bien que l'adresse `identifiant@provider.com` corresponde à une adresse non-locale, l'utilisateur dont l'adresse est `identifiant@provider.com` lui est *local*. Editez le fichier `/etc/postfix/recipient_canonical` :

```
identifiant@provider.com utilisateur
```

Générez alors à partir de ces fichiers les bases DB utilisées par Postfix par les commandes :

```
postmap /etc/postfix/sender_canonical
postmap /etc/postfix/recipient_canonical
```

Ajoutez alors dans le fichier `/etc/postfix/main.cf` :

```
sender_canonical_maps = hash:/etc/postfix/sender_canonical
recipient_canonical_maps = hash:/etc/postfix/recipient_canonical
puis relancez Postfix par postfix reload.
```

2.1.4. Test de la configuration

- Envoyez des messages aux utilisateurs locaux par la commande **mail**
- Envoyez des messages aux utilisateurs non-locaux par la commande **mail**
- Consultez la file d'attente par la commande **mailq**
- Affichez les répertoires des files d'attente par la commande **tree /var/spool/postfix**

2.2. Configuration de Fetchmail

Lancez l'utilitaire **fetchmailconf** en étant sous root. Choisissez tous les paramètres de vos comptes emails distants dans l'interface. Vous devriez obtenir en sortant du programme un fichier `/root/.fetchmailrc` de ce type :

```
poll pop.provider.com with proto POP3
  user 'utilisateur1' there with password 'passe' is utilisateur1 here options fetchall
  user 'utilisateur2' there with password 'passe' is utilisateur2 here options fetchall
```

2.3. Configuration de la connexion

Un script très pratique est livré avec pppd. Il s'agit de **ppp-on**. Pour l'installer, il suffit de le copier (`/usr/share/doc/` peut-être aussi `/usr/doc`) :

```
cp /usr/share/doc/ppp-2.4.0/scripts/ppp-on /etc/ppp/
cp /usr/share/doc/ppp-2.4.0/scripts/ppp-on-dialer /etc/ppp/
cp /usr/share/doc/ppp-2.4.0/scripts/ppp-off /etc/ppp/
```

Editez le fichier `/etc/ppp/ppp-on` pour affecter les valeurs aux paramètres TELEPHONE, ACCOUNT et PASSWORD.

Editez le fichier `/etc/ppp/ppp-on-dialer` pour ajuster le chemin de recherche de la commande **chat** : remplacez **chat** par `/usr/sbin/chat`.

2.4. Assemblage du tout

On établit la connexion régulièrement par une entrée cron. Une fois la connexion établie le fichier `/etc/ppp/ip-up.local` est exécuté automatiquement par le système ; on y place les commandes pour envoyer et recevoir les messages puis pour raccrocher.

Note : Afin d'éviter d'entrer en concurrence avec le script qui suit, supprimez dans le fichier `/etc/ppp/ip-up` toute ligne contenant `/usr/sbin/sendmail`. En effet cette commande est exécutée avec davantage de précaution par le script suivant.

Ajoutez une entrée à cron par la commande **crontab -e** qui ouvre vi dans lequel vous pouvez taper une ligne comme la suivante qui lance deux fois par heure du lundi au vendredi de 9 heures à 19 heures la connexion donc l'envoi et la récupération du courrier :

```
0,30 9-19 * * 1-5 /etc/ppp/ppp-on
```

Ajoutez au script `/etc/ppp/ip-up.local` les lignes suivantes et rendez-le exécutable par **chmod +x /etc/ppp/ip-up.local** :

```
#!/bin/bash
# Récupération des messages
/usr/bin/fetchmail -logfile /root/fetchlog &
fetchid=$!
fetchps=$(ps -no-headers -o pid -p $fetchid)
if [ -z "$fetchps" ]
  then fetchps=0
fi
sleep 10
while [ "$fetchps" -eq "$fetchid" -a -e "/var/lock/LCK..ttyS0" ]
do
  sleep 20
  fetchps=$(ps -no-headers -o pid -p $fetchid)
  if [ -z "$fetchps" ]
    then fetchps=0
  fi
done
if [ "$fetchps" -ne 0 ]
  then kill $fetchid
fi
# Envoi des messages mis en queue
/usr/sbin/sendmail -q &>/dev/null
sleep 10
mailq=$(/usr/bin/mailq)
while [ "$mailq" != "Mail queue is empty" -a -e "/var/lock/LCK..ttyS0" ]
```

```
do
    sleep 20
    mailq=$( /usr/bin/mailq )
done
# On raccroche
/etc/ppp/ppp-off
```

3. Utilisation du serveur

3.1. Administration à distance

3.1.1. Administration par SSH (Secure SHell)

SSH permet d'obtenir un shell sur une machine distante de manière sécurisée. On l'utilise en lieu et place de Telnet. Sur le serveur, installez le package RPM `openssh-server`. Sur le poste client, installez le package RPM `openssh-client` pour un client Linux. Il existe un excellent client SSH pour Windows® PuTTY disponible à l'adresse <http://www.chiark.greenend.org.uk/~sgtatham/> qui est à la fois un client SSH et une émulation de terminal de type bash. Sur le poste client, lancez la commande **ssh adresse-ip-serveur** sous Linux® ou double-cliquez sur l'exécutable `putty.exe` sous Windows®, connectez-vous alors sous un utilisateur puis faites un **su** pour passer sous le compte root (la connexion en root directe n'est pas recommandée pour des raisons de sécurité).

3.1.2. Administration par Webmin

Installez le package RPM `webmin`. Dans un navigateur web, allez à l'adresse <https://adresse-ip-serveur:10000/>⁸ (https pour Mandrake 7.2 et supérieures, http pour les autres).

3.2. Envoi régulier d'une page web

Recevoir tous les matins la page de la bourse ou autre dans ses messages peut être tout à fait agréable et utile. Pour cela, exécutez la commande **date -d '1 days ago' +%d >/root/datepage** pour initialiser le système puis ajoutez les lignes suivantes au fichier `/etc/ppp/ip-up.local` avant la commande qui raccroche en mettant le nom de l'utilisateur qui doit recevoir le message (champ `To:`) et l'adresse de la page à envoyer :

```
datepage=$(cat /root/datepage)
jour=$(date +%d)
heure=$(date +%H)
if [ $jour != $datepage -a $heure = 09 -a -e "/var/lock/LCK..ttyS0" ]
    then wget -O /root/devises.html http://fr.finance.yahoo.com/m3/index.html &>/dev/null
        head="From: MailServeur\nTo: utilisateur\nSubject: Devises\n
            Content-Type: text/html; charset=iso-8859-1;\n"
        echo -e $head >/root/mel
        cat /root/devises.html >/root/mel
        /usr/sbin/sendmail -t < /root/mel
        rm -f /root/mel
        date +%d >/root/datepage
    fi
```

3.3. Etapes d'ajout d'un compte de messagerie

Note : Le cas échéant, n'oubliez-pas de créer un compte POP3 chez votre hébergeur de messagerie.

Sur le serveur

1. Ajoutez un compte utilisateur par :
 - a. `useradd nom-utilisateur`
 - b. `passwd nom-utilisateur`
2. Complétez les fichiers `/etc/postfix/sender_canonical` et `/etc/postfix/recipient_canonical` ;
3. Regénérez les bases par :
 - a. `postmap /etc/postfix/sender_canonical`
 - b. `postmap /etc/postfix/recipient_canonical`
4. Relancez Postfix par `postfix reload` ;
5. Complétez le fichier `/root/.fetchmailrc`.

Sur le poste client

1. Configurez l'adresse IP du client ;
2. Dans le logiciel de messagerie, mettre l'adresse IP du serveur de messagerie pour l'adresse des serveurs POP3 et SMTP.

4. Statistiques des connexions et des courriers

Le but de cette partie est d'exploiter les logs du système en vue d'effectuer des statistiques de l'utilisation du serveur de messagerie. Pour cela, les logs bruts sont envoyés périodiquement cryptés par courrier électronique à un administrateur qui dispose d'outils d'analyse sous Linux® présentés ci-après.

4.1. Sur le serveur : envoi des logs

4.1.1. Scripts et programmes

Le premier script utile permet l'envoi crypté des logs. Log2mail (disponible⁹) est à copier dans le répertoire `/usr/local/bin/` par exemple. Il est nécessaire de modifier les permissions d'accès à ce script car le mot de passe de cryptage y est écrit en clair. Vérifiez que GnuPG est correctement installé.

```
cp log2mail /usr/local/bin/  
chmod 700 log2mail  
rpm -q gnupg
```

Editer le fichier `/usr/local/bin/log2mail` afin de modifier les variables du destinataire et du mot de passe. Il est aussi nécessaire de modifier le paramètre `-f` de Matt par une adresse email réelle si les logs sont destinés à un personne externe au réseau local.

Un programme est indispensable à la bonne marche du script présenté ci-dessus : Matt (disponible¹⁰ ou <ftp://ftp.riverdrums.com/pub/matt/>¹¹) qui envoie en pièce attachée un fichier qu'on lui désigne.

```
cp matt /usr/local/bin/
```

4.1.2. Configuration de Logrotate

Logrotate est le logiciel qui effectue la rotation des logs du système. Le principe est le suivant : périodiquement (chaque jour, semaine, mois) logrotate efface chaque fichier de log et en garde auparavant une copie qu'il numérote. Pour éviter l'accumulation de ces fichiers numérotés seuls les plus récents sont conservés. Modifiez le fichier `/etc/logrotate.d/syslog` comme suit :

```
/var/log/messages {  
    rotate 10  
    monthly  
    prerotate  
    /usr/local/bin/log2mail /var/log/messages log-messages  
    endscrip  
    postrotate  
    /usr/bin/killall -HUP syslogd  
    endscrip  
}
```

```
/var/log/mail/info {
```

```

rotate 10
monthly
prerotate
/usr/local/bin/log2mail /var/log/mail/info postfix-info
endscript
postrotate
/usr/bin/killall -HUP syslogd
endscript
}

```

4.2. Sur un poste Linux : analyse des logs

Note : Cette partie est encore en développement.

Les messages reçus par courriel sont cryptés. Pour les décrypter, exécuter **gpg --output log-messages-*date* --decrypt log-messages-*date*.gpg**.

4.2.1. Analyse des connexions

Ppplag est un script Perl disponible sur <http://www.biscom.net/~cade/away/ppplag/>¹² qui analyse `/var/log/messages`.

4.2.2. Analyse des échanges de courrier

Pflogsumm est un autre script Perl disponible sur http://jimsun.linxnet.com/postfix_contrib.html¹³ qui analyse `/var/log/mail/info`.

5. Remarques générales et techniques

Toutes les remarques sont bienvenues à chv@fr.st.

1. Si le relais ne fonctionne pas, la connexion reste ouverte inutilement.

Inexact pour moitié. En effet Fetchmail est pourvu d'un *timeout* de 300 secondes mais la commande **sendmail** de Postfix ne l'est pas. C'est un problème à résoudre.

2. Si la connexion ne fonctionne pas, le relais reste ouvert inutilement.

Inexact. En effet grâce au paramètre `-e "/var/lock/LCK..ttyS0"` des boucles `while` du script `ip-up.local`, si la connexion n'est plus ouverte le script se termine.

3. Le `ppp-on` peut être déjà lancé quand il est lancé par cron.

Exact mais sans conséquence. En effet, si un `ppp-on` est déjà lancé, la connexion échoue et le nouveau `ppp-on` se ferme.

4. Que faire lorsque l'on dispose d'une connexion permanente (ADSL en particulier) ?

Outre l'inutilité des demandes de connexion régulière par cron, un paramètre de la configuration de Postfix s'avère être inutile. Ce paramètre est le `relayhost` du fichier `/etc/postfix/main.cf`. En effet lors de connexions intermittentes (modem) Postfix transmet au relayhost (serveur SMTP distant) l'intégralité des messages sortants et le relayhost ensuite se charge de transmettre les messages à chaque correspondant. Cependant Postfix peut tout à fait jouer le rôle du relayhost c'est-à-dire contacter le serveur mail de chaque correspondant et assurer ainsi lui-même l'envoi des messages. Cette dernière méthode est utilisable et recommandée pour des connexions permanentes mais peu pratique dans un cas contraire.

Notes

1. serveur-mdk.sgml
2. serveur-mdk.dvi
3. serveur-mdk.pdf
4. <http://www.linux-france.org/article/mail/serveur-mdk/>
5. <http://chv.fr.st>
6. <http://www.linux-france.org/article/mail/postfix-jaco/>
7. <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

8. <https://adresse-ip-serveur:10000/>
9. [script/log2mail](#)
10. [script/matt](#)
11. <ftp://ftp.riverdrums.com/pub/matt/>
12. <http://www.biscom.net/~cade/away/ppplag/>
13. http://jimsun.linxnet.com/postfix_contrib.html