

Postfix + MySQL + Courier POP + Courier IMAP + Pop-before-smtp + SquirrelMail

ou comment avoir une usine à gaz à traiter le courrier chez vous...

Auteur : Guillaume Libersat, <glibersat AT linux62.org>

1.Introduction

Dans ce howto, nous allons voir comment installer une solution complète de mail sur votre serveur. Elle comportera un smtp (Postfix) utilisant une base de données (MySQL), un pop3 (Courier-pop), un imap (Courier-imap), un système d'authentification smtp (pop-before-smtp), et un webmail (Squirrel Mail). Cette solution sera multi-domaines, c'est à dire que vous pourrez héberger plus d'un nom de domaine sur votre serveur (ex : robert.com et alphonse.org).

2.Prérequis

Je ne détaillerais pas l'installation des logiciels, cela étant propre à chaque distribution. Vous devez avoir d'installé (je mets les versions que j'ai utilisé entre crochets) :

- Postfix avec le support MySQL [1.1.11]
- MySQL 3.x (4.x ?) [3.23.49]
- Courier base, avec le support MySQL [0.37.3]
- Courier pop (inclut dans imap pour certaines distributions) [0.37.3]
- Courier imap [0.37.3]
- Pop-before-smtp [1.28]
- Squirrel Mail [1.2.6]

3.Préliminaires

La base de données doit être lancée, et vous devez pouvoir vous connecter dessus en root. Tous les autres services comme Postfix doivent étre coupés.

4.Préparation de la base de données pour Postfix

4.1.Création de la base

Voici la base de données que j'utilise. Rien ne vous empêche de la modifier pour l'adapter à vos besoins. Celle-ci permet de gérer différents domaines, et est, je pense, logiquement structurée. Vous pouvez soit la créer manuellement, soit copier ceci dans un fichier texte, puis faire lire le dump à mysql. Je vous conseille de créer un utilisateur "postfix" afin de réduire les risques de sécurité. Cependant, vous pouvez effectuer toutes les opérations suivantes sous le compte "root" dans mysql, tant que vous attribuez les droits de relecture à "postfix" ensuite. Revenons donc à la création, copiez ceci dans un fichier :

```
#
# Structure de la table `transport`
#

CREATE TABLE transport (
  domain varchar(128) NOT NULL default '',
  transport varchar(128) NOT NULL default '',
  UNIQUE KEY domain (domain)
) TYPE=MyISAM;
# -----

#
# Structure de la table `users`
#

CREATE TABLE users (
  id varchar(128) NOT NULL default '',
  address varchar(128) NOT NULL default '',
  crypt varchar(128) NOT NULL default '',
  clear varchar(128) NOT NULL default '',
  name varchar(128) NOT NULL default '',
  uid smallint(5) unsigned NOT NULL default '1000',
  gid smallint(5) unsigned NOT NULL default '1000',
  home varchar(128) NOT NULL default '/',
  domain varchar(128) NOT NULL default '',
  maildir varchar(255) NOT NULL default '',
  imapok tinyint(3) unsigned NOT NULL default '1',
  bool1 tinyint(3) unsigned NOT NULL default '1',
  bool2 tinyint(3) unsigned NOT NULL default '1',
  PRIMARY KEY (id),
  UNIQUE KEY address (address),
  UNIQUE KEY id (id),
  KEY id_2 (id),
  KEY address_2 (address)
) TYPE=MyISAM;
# -----

#
# Structure de la table `virtual`
#

CREATE TABLE virtual (
  address varchar(255) NOT NULL default '',
  goto varchar(255) NOT NULL default '',
  UNIQUE KEY address (address)
) TYPE=MyISAM;
```

Puis, connectez vous à MySQL ("mysql -uroot -p", puis tapez votre mot de passe) et créez une database avec la commande :

```
create database maildb;
```

(Dans cet exemple, la database créée sera nommée "maildb").

Pour faire lire le fichier que vous venez de faire (dump.sql par exemple) à MySQL, tapez dans un shell ceci :

```
# mysql -uroot -p maildb < dump.sql
```

Nous allons maintenant créer l'utilisateur "postfix", pour cela, vous devez effectuer ceci dans mysql en "root" :

```
INSERT INTO user (host, user, password) VALUES('localhost','postfix','');
Query OK, 1 row affected (0.00 sec)

mysql> UPDATE user set password=PASSWORD('monpassword') WHERE user='postfix';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT select, insert, update ON maildb.* TO postfix;
Query OK, 0 rows affected (0.00 sec)
```

Si cela ne vous a pas retourné d'erreur, votre base de données est maintenant prête à être utilisée par postfix.

4.2. Quelques explications

Voici à quoi correspondent les tables :

- Transport

La table "transport" permet de définir les domaines que vous allez héberger, ainsi que les méthodes qui vont être utilisées pour acheminer le courrier. Par exemple, si le domaine gnurus.org doit être traité en tant que domaine virtuel, vous devrez alors mettre ceci :

domain	transport
gnurus.org	virtual:

- Users

La table "users" va contenir les utilisateurs, tous domaines confondus. Elle se compose de 13 colonnes :

id : Entrez l'email de la personne, cette colonne servira d'identifiant à l'utilisateur (avec le domaine) ;
address : Entrez l'adresse que l'utilisateur se verra donné (avec le domaine) ;
crypt : Le mot de passe crypté de l'utilisateur. Vous pouvez le générer avec MySQL, grâce à la fonction ENCRYPT().
clear : Le mot de passe clair de l'utilisateur. Ce n'est pas recommandé, et si vous optez pour la version cryptée, veuillez laisser ceci vide.
name : Le nom réel de la personne.
uid : Numéro unique identifiant la personne, cela correspondra aux permissions attribuées sur ses fichiers sur le disque.
gid : Numéro d'appartenance au groupe, bien que ce ne soit pas obligatoire, pour raison de maintenance, je vous conseille d'attribuer un numéro par domaine.
home : Entrez le chemin sur le disque jusqu'au dossier contenant vos domaines.
domain : Le domaine auquel le user appartient.
maildir : Le chemin sur le disque jusqu'au répertoire Maildir de l'utilisateur. N'oubliez pas le "/" à la fin du chemin, cela permet de faire utiliser à Postfix ce répertoire comme Maildir et non en Mailbox. Ce critère est déterminant pour l'utilisation couplée avec Courier.
imapok : Lire "Imap ok ?", cela permet d'activer ou de désactiver le compte temporairement en mettant 0 sur cette valeur.
bool1 et bool2 : Deux autres conditions que vous pouvez utiliser pour désactiver le pop, etc. Mettez "1" par défaut.

Voici un exemple d'utilisateur :

id	address	crypt	clear	name	uid	gid	home
glibersat@gnurus.org	glibersat@gnurus.org	s1kd2ls3jdjK6		Guillaume Libersat	1003	1002	/var/spool/pos

- Virtual

Cette table permet de définir les correspondances virtuelles. C'est à dire que vous pouvez entrer des alias, qui vont renvoyer le courrier à leur destination vers un utilisateur réel. Voici un exemple :

address	goto
webmaster@gnurus.org	glibersat@gnurus.org

Tous les e-mails en direction de webmater@gnurus.org seront donc redirigés vers glibersat@gnurus.org.

Vous avez maintenant votre table de prête, nous allons l'interfacer avec Postfix.

5. Postfix

5.1. Mise en place de Postfix

Par défaut, Postfix utilise des fichiers pour distribuer son courrier. Nous allons donc devoir modifier sa configuration afin d'utiliser les informations de la base de données avant toute distribution. Tous les fichiers cités se trouvent logiquement dans /etc/postfix/.

- Le fichier *main.cf* est le fichier principal, nous allons donc lui indiquer d'utiliser le "driver" mysql, ainsi que les fichiers qui serviront à formuler les requêtes. Le driver "fichier" est désigné par "hash:" et le driver "mysql" est désigné par "mysql:".

```
# Do not change these directory settings - they are critical to Postfix
# operation.
command_directory = /usr/sbin
daemon_directory = /usr/lib/postfix
program_directory = /usr/lib/postfix
```

```

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU) #Le message affiché à la connexion
setgid_group = postdrop
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

#Le nom resolvable de la machine
myhostname = cyclotron.gnurus.org
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

#Voici nos fichiers pour indiquer d'utiliser MySQL
transport_maps=mysql:/etc/postfix/transport.cf
virtual_mailbox_maps=mysql:/etc/postfix/mysql_virt.cf
virtual_uid_maps=mysql:/etc/postfix/uids.cf
virtual_gid_maps=mysql:/etc/postfix/gids.cf
virtual_mailbox_base=/
virtual_maps=mysql:/etc/postfix/virtual.cf

#Laissez cette variable, vous pouvez en ajouter des statiques en séparant
par des virgules
relay_domains = $transport_maps
myorigin = /etc/mailname
mydestination = $transport_maps
relayhost =
#Les réseaux qui pourront envoyer des mails avec ce smtp
mynetworks = 127.0.0.0/8, 192.168.1.0/24
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +

```

- Transport.cf

```

user = postfix
password = votre_password
dbname = maildb
table = transport
select_field = transport
where_field = domain
hosts = localhost

```

- mysql_virt.cf

```

user = postfix
password = votre_password
dbname = maildb
table = users
select_field = maildir
where_field = address
hosts = localhost

```

- uids.cf

```

user = postfix
password = votre_password
dbname = maildb
table = users
select_field = uid
where_field = address
hosts = localhost

```

- gids.cf

```

user = postfix
password = votre_password
dbname = maildb
table = users
select_field = gid
where_field = address
hosts = localhost

```

- virtual.cf

```

user = postfix
password = votre_password
dbname = maildb
table = virtual
select_field = goto
where_field = address
hosts = localhost

```

5.2.Création de comptes

Votre Postfix est donc maintenant interfacé avec MySQL. Vous pouvez le lancer, vérifier les logs s'il n'y a pas de message d'erreur. Si Postfix à l'air de tourner, essayer d'envoyer un mail vers l'extérieur, c'est à dire une adresse hors de votre domaine (ex : @nerim.fr). Si cela se passe bien, vous pouvez continuer, sinon, lisez les logs, et essayez de résoudre le problème. Pour envoyer un mail avec votre serveur, configurez votre MUA (Client Mail) pour utiliser votre serveur postfix en tant que serveur smtp. Nous allons donc maintenant créer des comptes, et essayer d'envoyer un mail vers notre domaine. Vous pouvez soit vous débrouiller avec MySQL et la création locale, soit utiliser des scripts que j'ai fait afin de me faciliter la vie (et libre à vous de les adapter). L'un des deux scripts permet d'ajouter un domaine, et l'autre d'ajouter un utilisateur dans ce domaine. (Créez le répertoire "/var/spool/postfix/virtual" si il n'existe pas avant d'utiliser ces scripts).

- newdomain.sh

```
#!/bin/bash

#CONFIG
MYSQL_USER=postfix
MYSQL_PASS=votre_password
MYSQL_DB=maildb

POSTFIX_BASE_DIR=/var/spool/postfix/virtual/

#####
### DO NOT EDIT AFTER THIS      ###
#####

echo -n "Enter the new domain name : "
read domain

echo -n "Enter the transport type ( ex : "virtual:" ) : "
read transport

echo -n "Enter the GID of this domain : "
read gid

if [ ! `mysql -u$MYSQL_USER -p$MYSQL_PASS --database=$MYSQL_DB -e"INSERT INTO transport (domain, transport) VALUES ('$dom
then
    echo "MySQL INSERT() was successfull"

    if [ ! `mkdir $POSTFIX_BASE_DIR/$domain` ]
    then
        chgrp $gid $POSTFIX_BASE_DIR/$domain
        chmod 770 $POSTFIX_BASE_DIR/$domain
        echo "Content Directory created"
    fi
fi
```

- newuser.sh

```
#!/bin/bash

#CONFIG
MYSQL_USER=postfix
MYSQL_PASS=votre_password
MYSQL_DB=maildb

#WITH a trailing slash
POSTFIX_BASE_DIR=/var/spool/postfix/virtual/

#####
### DO NOT EDIT AFTER THIS      ###
#####

echo -n "-> Enter the new username ( ex : glibersat ) : "
read username

echo -e "\n-> Enter the domain name ( ex : gnurus.org )... "
echo -e "\tDetected domains : "

for domains in `ls $POSTFIX_BASE_DIR | sort`
do
    echo -e "\t(*) $domains"
done
echo -n "domain : "
read domain

echo -n -e "\nEnter the password ( clear text ) : "
read password

echo -n -e "\nEnter the real name ( ex : Guillaume Libersat ) : "
read realname

#Get domain GroupID
vgid=`ls -l /var/spool/postfix/virtual/ | grep $domain | awk '{ print $4 }'`

#Get a free UserID
vuid=`mysql -u$MYSQL_USER -p$MYSQL_PASS -D$MYSQL_DB -e"select MAX(uid) from users" | tail -n1 | awk '{ print $1 }'`
vuid=$((vuid+1))

echo -n -e "\nEnter the GID of this domain (detected $vgid, hit Return to use it) : "
read r_gid
if [ $r_gid ]
then
    vgid=$r_gid
fi

echo -n -e "\nEnter the UID of this user (next free uid is $vuid, hit Return to use it) : "
read r_uid
if [ $r_uid ]
then
    vuid=$r_uid
fi

if [ ! `mysql -u$MYSQL_USER -p$MYSQL_PASS -D$MYSQL_DB -e"INSERT INTO users (id, address, crypt, clear, name, uid, gid, ho
then
    echo "MySQL INSERT() was successfull"

    if [ ! `mkdir $POSTFIX_BASE_DIR/$domain/$username` ]
```

```

        then
            maildirmake $POSTFIX_BASE_DIR/$domain/$username/Maildir
            chown $vuid:$vgid -R $POSTFIX_BASE_DIR/$domain/$username
            chmod 770 $POSTFIX_BASE_DIR/$domain
            echo "User Directory created"
        fi
    fi
fi

```

Ces scripts sont à lancer en `root` (ou tout autre utilisateur capable d'écrire dans le dossier "virtual" de postfix. N'oubliez pas de les "chmod +x xxxx.sh" avant de les lancer.

Essayez donc maintenant d'envoyer un mail vers l'utilisateur que vous avez créé. Vous devriez voir apparaître dans les logs une ligne de ce style :

```

May 20 19:17:00 cyclotron postfix/qmgr[425]: 19C5A33F06: from=<root@gnurus.org>, size=303, nrcpt=1 (queue active)
May 20 19:17:00 cyclotron postfix/virtual[30291]: 19C5A33F06: to=<glibersat@gnurus.org>, relay=virtual, delay=1, status=sent (maildir)

```

Il est probable que vous obteniez des erreurs. Par chance, elles sont souvent explicites, essayez donc de les corriger (problème de permission par exemple). Si cela fonctionne, passez à la suite !

6. Courier POP et IMAP

Maintenant que nous avons notre smtp, il serait utile d'avoir un serveur pop3 et un IMAP afin de pouvoir récupérer nos mails. Nous allons donc mettre en place un serveur pop3 et IMAP avec courier.

6.1. Fichiers communs

La première chose à faire est de dire à courier d'utiliser mysql. Nous allons donc changer le type d'authentification. Cherchez les lignes correspondantes à ces suivantes, et remplacez les par cela (les fichiers sont en général dans `/etc/courier`) :

- `authdaemonrc`

```

authmodulelist="authmysql authpam"
authmodulelistorig="authcustom authcram authuserdb authldap authmysql authpam"
daemons=5 #Libre à vous de le modifier
version="authdaemond.mysql"
authdaemonvar="/var/run/courier/authdaemon" #Selon votre distribution, cela peut changer

```

- `authmysqlrc`

```

MYSQL_SERVER localhost
MYSQL_USERNAME postfix
MYSQL_PASSWORD votre_password

MYSQL_SOCKET /var/run/mysqld/mysqld.sock #Connexion locale
MYSQL_PORT 0
MYSQL_OPT 0

MYSQL_DATABASE maildb
MYSQL_USER_TABLE users

MYSQL_CRYPT_PWFIELD crypt

DEFAULT_DOMAIN gnurus.org #Permet d'autoajouter ce domaine à la fin des noms d'utilisateur si le domaine n'est pas spécifié.

MYSQL_UID_FIELD uid
MYSQL_GID_FIELD gid
MYSQL_LOGIN_FIELD ld
MYSQL_HOME_FIELD home
MYSQL_NAME_FIELD name
MYSQL_MAILDIR_FIELD maildir
MYSQL_WHERE_CLAUSE imapok=1 AND bool1=1 AND bool2=1 #Vous rappelez vous de ces conditions ? :-)

```

6.2. Courier POP3

Configurer maintenant le fichier `pop3d` comme suit :

```

#Ceci peut changer selon les distributions
prefix=/usr
exec_prefix=/usr
sbindir="/usr/sbin"
PIDFILE=/var/run/courier/pop3d.pid
#---

MAXDAEMONS=40
MAXPERIP=4

AUTHMODULES="authdaemon"
AUTHMODULES_ORIG="authdaemon"

POP3AUTH="LOGIN CRAM-MD5 CRAM-SHA1"
POP3AUTH_ORIG="LOGIN CRAM-MD5 CRAM-SHA1"

POP3AUTH_TLS=""
POP3AUTH_TLS_ORIG="LOGIN PLAIN"

PORT=110
ADDRESS=0 #0 signifie simplement d'écouter sur toutes les interfaces
TCPDOPTS="--nodnslookup -noidentlookup"

POP3DSTART=YES

```

6.3. Courier IMAP

C'est maintenant au tour du serveur IMAP. Le fichier s'appelle `imapd`, modifiez le suivant cet exemple :

```

ADDRESS=0
PORT=143

MAXDAEMONS=40
MAXPERIP=4

PIDFILE=/var/run/courier/imapd.pid

TCPDOPTS="--nodnslookup -noidentlookup"

AUTHMODULES="authdaemon"
AUTHMODULES_ORIG="authdaemon"

IMAP_CAPABILITY="IMAP4rev1 CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT"
IMAP_CAPABILITY_ORIG="IMAP4rev1 CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT AUTH=CRAM-MD5 AUTH=CRAM-SHA1 IDLE"

IMAP_IDLE_TIMEOUT=60

IMAP_CAPABILITY_TLS="$IMAP_CAPABILITY AUTH=PLAIN"

```

```

IMAP_CAPABILITY_TLS_ORIG="$IMAP_CAPABILITY_ORIG AUTH=PLAIN"

IMAP_DISABLETHREADSORT=0
IMAP_CHECK_ALL_FOLDERS=0
IMAP_OBSOLETE_CLIENT=0

IMAP_ULIMITD=65536

IMAP_USELOCKS=0

IMAP_EMPTYTRASH=Trash:7

IMAP_MOVE_EXPUNGE_TO_TRASH=0

IMAPDSTART=YES

```

6.4.Essais

Vous devriez donc maintenant avoir un serveur pop et un serveur imap fonctionnel. Vous pouvez l'essayer en configurant votre MUA en indiquant comme serveur pop3 votre serveur courrier. Le login est l'e-mail complet, en effet, vu que notre solution est multi-domaines, il faut pouvoir avoir plusieurs utilisateurs avec le même nom sur plusieurs domaines (ex : glibersat@gnurus.org et glibersat@linux62.com). Le login peut être complètement différent de l'e-mail, il suffit de l'indiquer dans la colonne "ID" de la table "USERS" de MySQL. Si vous n'arrivez pas à remonter vos mails, lisez les logs.

7.Pop-before-smtp

Nous avons maintenant un système pratiquement complet. Cependant, si vous souhaitez pouvoir utiliser votre smtp de l'extérieur depuis n'importe où, sans pour autant être floodé de mails de spam jusqu'à saturer votre connexion (et croyez moi, cela arrive :-), nous allons devoir mettre un système de sécurité en place. Si vous avez des ips fixes, dans ce cas, rajoutez les dans la configuration de postfix, mais si vous n'en disposez pas, nous allons devoir utiliser pop-before-smtp. Il est aussi possible d'utiliser une authentification pour le smtp, mais c'est cette solution qui sera retenue dans ce howto.

Ce logiciel va en fait intercepter dans vos logs les accès au pop, puis débloquer l'envoi de mails depuis l'ip qui a "poppée". En conséquence, une fois que vous vous aurez récupéré vos mails, vous pourrez en envoyer depuis la même adresse (bien entendu, pour un temps limité).

Pop-before-smtp est dit supporter Postfix, cependant, je ne suis pas parvenu à la faire fonctionner par défaut. J'ai donc du modifier `/etc/pop-before-smtp/pop-before-smtp.conf` ainsi :

```

pat2 = '^XXXXX$'
pat = '^((... ..)...) \S+ courierpop3login: LOGIN, user=\S+, ip=\{.....(\d+\.\d+\.\d+\.\d+)\}$'

```

8.Squirrel Mail

SquirrelMail est une interface en php qui permet de récupérer et envoyer des mails à travers un serveur imap et smtp. Cela peut être utile si vous avez de nombreuses personnes qui vont se connecter depuis des endroits où ils ne peuvent accéder à un MUA ou si simplement vous ne souhaitez pas leur laisser cette liberté. Nous allons donc voir comment configurer SquirrelMail et Apache.

8.1.SquirrelMail

Les fichiers se trouvent soit dans le répertoire où vous avez décompressé SquirrelMail ou soit dans `/etc/squirrelmail/` (Debian) :

- `config.php`
Je ne détaillerai pas tout le fichier, simplement les lignes impératives.

```

$org_name = 'Gnurus.org';
$org_title = "Webmail Gnurus.org";
$squirrelmail_default_language = 'fr_FR';
$imapServerAddress = 'localhost';
$imapPort = 143;
$domain = 'gnurus.org';
$smtpServerAddress = 'localhost';
$smtpPort = 25;
[...]
$imap_server_type = 'courier';
[...]

```

8.2.Apache

Nous allons maintenant ajouter un chemin virtuel à Apache. Pour cela, créez un fichier `apache.conf` dans le répertoire de SquirrelMail (ex : `/etc/squirrelmail/`) contenant ceci :

```

#Changer le chemin vers le répertoire de squirrelmail
Alias /squirrelmail /usr/share/squirrelmail

<Directory /usr/share/squirrelmail>
php_flag register_globals on
Options Indexes FollowSymLinks
</Directory>

NameVirtualHost *

# users will prefer a simple URL like http://webmail.example.com
<VirtualHost *>
DocumentRoot /usr/share/squirrelmail
ServerName mail.gnurus.org
</VirtualHost>

```

Ajoutez ensuite ceci dans votre fichier de configuration d'Apache (`/etc/httpd/conf/httpd.conf` en général) :

```
Include /etc/squirrelmail/apache.conf
```

Puis, relancez Apache, mais avec un "stop" et "start", et non pas un "restart", pour éviter le "Gracefull Restart".

9.Conclusion

Vous devriez maintenant avoir un système de mail complet digne des FAI :-). Si vous avez rencontré des problèmes, et/ou si vous en avez encore, n'hésitez pas à me contacter afin d'ajouter vos épreuves à la section problèmes :-). Il ne vous reste plus qu'à faire bon usage de votre système de mails!

10.Problèmes

Q : Quand j'essaye d'envoyer un mail, j'ai cette erreur dans les logs :
May 20 20:37:14 caporal postfix/smtpd[21442]: fatal: open database /etc/aliases.db: No such file or directory
R : créez le fichier `/etc/aliases`, puis lancez le programme "newaliases".