Virtualize users and domains with Postfix, Courier, MySQL and SquirrelMail

Postfix is ??a MTA (Mail Transport Agent), written by Wietse Venema when he was working at IBM's TJ Watson research center. Postfix features: easy to manage, fast, safe ...

TipsMake.com - Postfix is ??a MTA (Mail Transport Agent), written by Wietse Venema when he was working at IBM's TJ Watson research center. Postfix features: easy to manage, fast and safe. Just need a server with normal hardware, Postfix can transfer millions of emails a day. In the following article, we will show you how to install a basic Postfix mail server on a virtual user and domain (ie user and domain are stored in the MySQL database).

>>>**Run SquirrelMail on Nginx (LEMP) in Debian Squeeze / Ubuntu 11.04**

In addition, the article will show the process of installing and configuring Courier (Courier-POP3, Courier-IMAP), from which Courier can authenticate the MySQL database that Postfix uses. As a result, Postfix server can authenticate **SMTP-AUTH** , **TLS** and **quota** (quota is not built in Postfix by default, we will patch Postfix appropriately). The passwords will be stored in the **encrypted** form of the database. At the same time, we will show you how to install Amavisd, SpamAssassin and ClamAV so that these emails can be scanned and checked for spam and viruses. Finally, we will install the SquirrelMail webmail interface so that users can read and send mail, change the password.

overview

The advantage of a ' *virtual* ' setting (*virtual* user and domain in the MySQL database) is that it greatly increases performance compared to a basic ' *real* ' user-based system setup. With virtual setup, your mail server can handle thousands of domains and users. Besides, the administration also becomes easier because you only need to deal with the MySQL database when adding new users / domains or editing existing accounts. No *postmap* commands are needed to create the db file, no need to reload Postfix . the management of the MySQL database can be done via web tools like **phpMyAdmin** (will be installed in this tutorial). The third advantage is that users will have an email address that is the user name instead of a username and an annoying email.

1. Some notes

Here we use the Fedora 15 x86_64 system, have a static IP address *192.168.0.100* and hostname *server1.example.com* . You need to make sure that the firewall and **SELinux** are disabled.

2. Some software needed

First we need to update the packages available on the system:

yum update

Then install some necessary software:

yum groupinstall 'Development Tools'

yum groupinstall 'Development Libraries'

3. Install Apache, MySQL, phpMyAdmin

This software suite can be installed at the same time via the following command line (including the packages to be built for Courier-IMAP):

yum install ntp httpd mysql-server php php-mysql php-mbstring rpm gcc build mysql-devel openssl-devel cyrus-sasl-devel pkgconfig zlib-devel phpMyAdmin pcre-devel openldap-devel postgresql-devel expect libtool-ltdl-devel openldap -servers libtool gdbm-devel pam-devel gamin-devel libidn-devel

4. Install Courier-IMAP, Courier-Authlib, And Maildrop

Unfortunately, the rpm package is not available for Courier-IMAP, Courier-Authlib, and Maildrop. Therefore we need to build it for use. Rpm packages should not be built as **root** , courier-imap will refuse to compile if it detects that the package runs at root. So we will create a normal account (eg, *quantrimang*) and assign it a password:

useradd -m -s / bin / bash quantrimang
passwd quantrimang

Then we use the sudo command so that *quantrimang* can compile and install rpm packages. But first, let *quantrimang* run all commands using sudo:

visudo

In the file opened, in the line root ALL = (ALL) ALL add a similar line for *quantrimang* below:

```
[.] ## Allow root to run any commands anywhere root ALL = (ALL) ALL quantriman
```

Now it's time to build the rpm package. First become *quantrimang* :

su quantrimang

Next create the built environment:

mkdir $ HOME / rpm
mkdir $ HOME / rpm / SOURCES
mkdir $ HOME / rpm / SPECS
mkdir $ HOME / rpm / BUILD

mkdir $ HOME / rpm / BUILDROOT
mkdir $ HOME / rpm / SRPMS
mkdir $ HOME / rpm / RPMS
mkdir $ HOME / rpm / RPMS / i386
mkdir $ HOME / rpm / RPMS / x86_64

echo "% _topdir $ HOME / rpm" >> $ HOME / .rpmmacros

Create the *downloads* folder and download the source file **here** :

mkdir $ HOME / downloads
cd $ HOME / downloads

wget https://sourceforge.net/projects/courier/files/authlib/0.63.0/courier-authlib-0.63.0.tar.bz2/download
wget https://sourceforge.net/projects/courier/files/imap/4.9.3/courier-imap-4.9.3.tar.bz2/download
wget https://sourceforge.net/projects/courier/files/maildrop/2.5.4/maildrop-2.5.4.tar.bz2/download

Still in *$ HOME / downloads* , we build courier-authlib:

sudo rpmbuild -ta courier-authlib-0.63.0.tar.bz2

Once completed, the rpm package can be found in */ root / rpmbuild / RPMS / x86_64* (*/ root / rpmbuild / RPMS / i386* if you are running the i386 system):

sudo ls -l / root / rpmbuild / RPMS / x86_64

Available packages of rpm are displayed as follows:

```
[quantrimang @ server1 downloads] $ sudo ls -l / root / rpmbuild / RPMS / x86_
```

Select a package you want, then install it similarly below:

sudo rpm -ivh /root/rpmbuild/RPMS/x86_64/courier-authlib-0.63.0-1.fc15.x86_64.rpm
/root/rpmbuild/RPMS/x86_64/courier-authlib-mysql-0.63.0-1.fc15 .x86_64.rpm
/root/rpmbuild/RPMS/x86_64/courier-authlib-devel-0.63.0-1.fc15.x86_64.rpm

Go back to *downloads* folder:

cd $ HOME / downloads

Run the following commands to create directories that receive / allow changes (otherwise Courier-Imap will fail):

sudo mkdir / var / cache / ccache / tmp
sudo chmod o + rwx / var / cache / ccache /
sudo chmod 777 / var / cache / ccache / tmp

Now run rpmbuild again, note that not through sudo because the compiler doesn't allow it to run as root.

rpmbuild -ta courier-imap-4.9.3.tar.bz2

Then the rpm package will be found in *$ HOME / rpm / RPMS / x86_64* (*$ HOME / rpm / RPMS / i386* if it is an i386 system):

cd $ HOME / rpm / RPMS / x86_64

Run the following command:

ls -l

You will receive the available rpm packages:

[quantrimang @ server1 x86_64] $ ls -l
total 1040
-rw-rw-r-- 1 quantrimang quantrimang 315872 May 25 18:33 courier-imap-4.9.3-1.15.x86_64.rpm
-rw-rw-r-- 1 quantrimang quantrimang 743200 May 25 18:33 courier-imap-debuginfo-4.9.3-1.15.x86_64.rpm
[quantrimang @ server1 x86_64] $

Install courier-imap with the following command:

sudo rpm -ivh courier-imap-4.9.3-1.15.x86_64.rpm

Go back to *downloads* folder:

cd $ HOME / downloads

Run *rpmbuild* to build maildrop package:

sudo rpmbuild -ta maildrop-2.5.4.tar.bz2

Then the rpm package can be found at */ root / rpmbuild / RPMS / x86_64* (*/ root / rpmbuild / RPMS / i386* with i386 system).

sudo ls -l / root / rpmbuild / RPMS / x86_64

Available rpm packages are listed:

```
[quantrimang @ server1 downloads] $ sudo ls -l / root / rpmbuild / RPMS / x86_
```

Install maildrop with the following command:

sudo rpm -ivh /root/rpmbuild/RPMS/x86_64/maildrop-2.5.4-1.15.x86_64.rpm

After compiling and successfully installing the necessary packages, you can return to root with the command:

exit

5. Apply the Patch Quota patch to Postfix

We have to get the Postfix rpm source, patch it with the quota patch, build a new Postfix rpm package and install it:

cd / usr / src
wget http://ftp-stud.fht-esslingen.de/pub/Mirrors/fedora/linux/releases/15/Everything/source/SRPMS/postfix-2.8.2-2.fc15.src.rpm
rpm -ivh postfix-2.8.2-2.fc15.src.rpm

The following warnings appear, you can ignore it:

warning: mockbuild user does not exist - dùng root
c?nh báo: mockbuild group không t?n t?i - dùng root

cd / root / rpmbuild / SOURCES
wget http://vda.sourceforge.net/VDA/postfix-vda-v10-2.8.2.patch
cd / root / rpmbuild / SPECS /

We need to fix the postfix.spec file:

en postfix.spec

And add Patch0: postfix-vda-v10-2.8.2.patch to the # Patches section, adding % patch0 -p1 -b .vda-v10 to % setup -q as follows:

```
[.] # Patches Patch0: postfix-vda-v10-2.8.2.patch Patch1: postfix-2.7.0-config
```

Then build new Postfix rpm package with quota and MySQL support:

rpmbuild -ba postfix.spec

Go to the postfix rpm directory in */ root / rpmbuild / RPMS / x86_64* (*/ root / rpmbuild / RPMS / i386* if it is an i386 system):

cd / root / rpmbuild / RPMS / x86_64

Run the command:

ls -l

You will receive the available packages:

```
[root @ server1 x86_64] # ls -l total 8308 -rw-r - r-- 1 root root 123528 Jun 2
```

Select the package you want and install it as follows:

rpm -ivh postfix-2.8.2-2.fc15.x86_64.rpm

---

6. Install password for MySQL and configure phpMyAdmin

First we start MySQL:

chkconfig --levels 235 mysqld on
/etc/init.d/mysqld start

Then set up MySQL passwords for root accounts:

mysql_secure_installation

```
  [root @ server1 ~] # mysql_secure_installation NOTE: RUNNING ALL PARTS OF THIS
?nh ?? ??ng nh?p vào MySQL ?? th?c hi?n nó, s? c?n ph?i m?t kh?u hi?n th?
i cho ng??i ch? root.  N?u ?ã ?ã ???c cài ??t MySQL, và b?n không ??t tên m
?t kh?u nh?ng nào, m?t kh?u s? ???c tr?ng, vì b?n nên ch? ??ng nh?p vào ?
ây.  Enter current password cho root (nh?p cho không): OK, ?
ã successfully dùng m?t kh?u, chuy?n ??i khi .
  Thi?t l?p m?t kh?u g?c c?n thi?t mà không th? ??ng nh?p vào MySQL
  root user without the proper authoring.
  Set root password?  [Y / n] New password: Re-enter new password: Password upda
  Reloading privilege tables .
  . Success!
  Vì m?c ??nh, m?t cài ??t MySQL có m?t ng??i dùng không rõ, Allow anyone
  ?? ??ng nh?p vào MySQL không có th? có m?t ng??i dùng ??ng nh?p cho cho
  more.  ?ây ???c s? d?ng ch? ?? ki?m tra, và ?? t?o cài ??t
  go a bit smoother.  B?n nên g? b? chúng tr??c khi chuy?n vào
  production environment.
  Remove anonymous users?  [Y / n] . Success!
  Th??ng, root nên ch? ???c phép phép k?t n?i t? 'localhost'.  This
  ensures that someone cannot guess at the root password from the network.
  Disallow root login command?  [Y / n] . Success!
  By default, MySQL comes with a database ???c xác ??nh 'th?' mà có th?
  access.  ?ây không ph?i ???c xác ??nh ch? cho th? th?, và nên ???c g? b?
 tr??c khi chuy?n vào m?t c? s? d? li?u.
  Remove test database and access to it?  [Y / n] - Dropping test database .
  . Success!
  - Removing privileges on test database .
  . Success!
  Reloading the privilege tables will ensure that all changes do so far
  s? x? lý ngay ngay.
  Reload privilege tables now?  [Y / n] . Success!
  Cleaning up .
  All done!  If you've completed all steps theo ?ây, b?n MySQL
  cài ??t nên ???c b?o v?.
  Thanks for dùng MySQL!
  [root @ server1 ~] #
```

Next we configure phpMyAdmin. Change Apache so that phpMyAdmin allows other connections (not just from *localhost*) by creating comments in the paragraph.

```
  # phpMyAdmin - Web based MySQL browser written in php # # Only allowed localhos
?ã xác ??nh c?a SSL không ???c xác ??nh # b? Danger khi ???c xác ??nh b?
i SSL Alias ??
/ phpMyAdmin usr / share / phpMyAdmin # # Order Deny, Allow # Deny from All # Al
```

Create a boot path for Apache and activate it:

chkconfig --levels 235 httpd on
/etc/init.d/httpd start

Now you can access the link *http://server1.example.com/phpMyAdmin/* or *http://192.168.0.100/phpMyAdmin/* on your browser, then log in with the *root* username and password root MySQL.

7. Create MySQL database for Postfix / Courier

First we create a db named mail:

mysqladmin -u root -p create mail

Next to MySQL shell:

mysql -u root -p

On the MySQL shell, we create a *mail_admin* user with password *mail_admin_password* (replace with your own password) with basic rights such as *SELECT, INSERT, UPDATE, and DELETE* on the mail database. This user will be used by Postfix and Courier to connect to the mail database:

GRANT SELECT, INSERT, UPDATE, DELETE ON mail. * TO 'mail_admin' @ 'localhost' IDENTIFIED BY 'mail_admin_password';
GRANT SELECT, INSERT, UPDATE, DELETE ON mail. * TO 'mail_admin'@'localhost.localdomain' IDENTIFIED BY 'mail_admin_password';
FLUSH PRIVILEGES;

Still in the MySQL shell, we create the tables that Postfix and Courier need:

USE mail;
CREATE TABLE domains (
domain varchar (50) NOT NULL,
PRIMARY KEY (domain))
ENGINE = MyISAM;
CREATE TABLE forwardings (
source varchar (80) NOT NULL,
destination TEXT NOT NULL,
PRIMARY KEY (source))
ENGINE = MyISAM;
CREATE TABLE users (
email varchar (80) NOT NULL,
password varchar (20) NOT NULL,
bigint quota (20) DEFAULT '10485760',

PRIMARY KEY (email)
) ENGINE = MyISAM;
CREATE TABLE transport (
domain varchar (128) NOT NULL default '',
transport varchar (128) NOT NULL default '',
UNIQUE KEY domain (domain)
) ENGINE = MyISAM;
quit;

By using the **quit** command **;** we will leave the MySQL shell and return to the Linux shell.

Domain tables will store each virtual domain for Postfix to receive emails (eg *example.com*).

domain example.com

The *forwardings* table is for emails pointing to another email, for example pointing from *info@example.com* to *sales@example.com* .

source destination info @ example.comsales @ example.com

The *users* table stores all virtual account information and password with the mail box quota value (in this example, the default value is *10485760* bytes equivalent to *10MB*).

email password quota sales@example.comNo9.E4skNvGa. ("secret" in encrypted form) 10485760

The *transport* table is an optional option, for advanced users, that allows mail forwarding for each single user, or the entire domain, as well as all mail to another server. For example:

domain transport example.comsmtp: [1.2.3.4]

Here the entire email will forward to *example.com* via the smtp protocol to the server with the IP address *1.2.3.4* , *[]* brackets mean ' *do not perform a search of the DNS MX record* ' (for IP addresses). If you use a fully qualified domain name (FQDN) then this *[]* mark is not required.

8. Configure Postfix

Now we need to tell Postfix where it can find all the information in the database. First need to create 6 text files, then notify Postfix to connect to MySQL on IP address *127.0.0.1* instead of *localhost* . We have to do this because Postfix is ??running on a chroot jail and does not have access to the MySQL socket, so if it is not adjusted it will try to connect via *localhost* . When using *127.0.0.1* , Postfix will pass TCP to connect to MySQL without any problems in a chroot jail.

Create 6 text files:

vi /etc/postfix/mysql-virtual_domains.cf

```
user = mail_admin password = mail_admin_password dbname = mail query = SELECT
```

vi /etc/postfix/mysql-virtual_forwardings.cf

```
user = mail_admin password = mail_admin_password dbname = mail query = SELECT
```

vi /etc/postfix/mysql-virtual_mailboxes.cf

```
user = mail_admin password = mail_admin_password dbname = mail query = SELECT
```

vi /etc/postfix/mysql-virtual_email2email.cf

```
user = mail_admin password = mail_admin_password dbname = mail query = SELECT
```

en /etc/postfix/mysql-virtual_transports.cf

```
user = mail_admin password = mail_admin_password dbname = mail query = SELECT
```

vi /etc/postfix/mysql-virtual_mailbox_limit_maps.cf

```
user = mail_admin password = mail_admin_password dbname = mail query = SELECT
```

chmod o = /etc/postfix/mysql-virtual_*.cf
chgrp postfix /etc/postfix/mysql-virtual_*.cf

Now we create a user and group named *vmail* with the home directory of */ home / vmail* . This will store the entire inbox.

groupadd -g 5000 vmail
useradd -g vmail -u 5000 vmail -d / home / vmail -m

Next is the postfix configuration process. Make sure that you have replaced *server1.example.com* with a valid FQDN, otherwise Postfix may not work properly.

```
postconf -e 'myhostname = server1.example.com' postconf -e 'mydestination = se
```

Then create the SSL certificate needed for TLS:

cd / etc / postfix
openssl req -new -outform PEM -out smtpd.cert -newkey rsa: 2048 -nodes -keyout smtpd.key -keyform PEM -days 365 -x509

Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]: Organization Name (eg, the name of your company).
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []: Qualified Domain Name of the system (eg "server1.example.com").
Email Address []:

Change permissions for smtpd.key:

chmod o = /etc/postfix/smtpd.key

---

9. Saslauthd configuration

Editing the */etc/sasl2/smtpd.conf* file is as follows:

vi /etc/sasl2/smtpd.conf

```
pwcheck_method: authdaemond log_level: 3 mech_list: PLAIN LOGIN authdaemond_pa
```

Then turn off Sendmail and start Postfix, saslauthd, courier-authlib:

chmod 755 / var / spool / authdaemon
chkconfig --levels 235 courier-authlib on
/etc/init.d/courier-authlib start
chkconfig --levels 235 sendmail off
chkconfig --levels 235 postfix on
chkconfig --levels 235 saslauthd on
/etc/init.d/sendmail stop
/etc/init.d/postfix start
/etc/init.d/saslauthd start

10. Configure Courier

Now we will tell Courier to verify the MySQL database again. First, edit */ etc / authlib / authdaemonrc* and change the value of *authmodulelist* :

en / etc / authlib / authdaemonrc

```
[.] authmodulelist = "authmysql" # authmodulelist = "authuserdb authpam authpgs
```

Then edit */ etc / authlib / authmysqlrc* . Ensure accuracy for what entered.

cp / etc / authlib / authmysqlrc / etc / authlib / authmysqlrc_orig
cat / dev / null> / etc / authlib / authmysqlrc
en / etc / authlib / authmysqlrc

```
MYSQL_SERVER localhost MYSQL_USERNAME mail_admin MYSQL_PASSWORD mail_admin_pass
```

Courier restart:

chkconfig --levels 235 courier-imap on
/etc/init.d/courier-authlib restart
/etc/init.d/courier-imap restart

The first time you launch courier-imap, it will automatically generate authentication files */usr/lib/courier-imap/share/imapd.pem* and */usr/lib/courier-imap/share/pop3d.pem* from the file. */usr/lib/courier-imap/etc/imapd.cnf* and */usr/lib/courier-imap/etc/pop3d.cnf* . Because the contents of the *.cnf* files contain the **CN = localhost** line, our server is named *server1.example.com* so the certificates may have trouble using the TLS connection. To solve this problem, let's remove the authentication .

cd / usr / lib / courier-imap / share
rm -f imapd.pem
rm -f pop3d.pem

. and replace the line **CN = localhost** in */usr/lib/courier-imap/etc/imapd.cnf* and */usr/lib/courier-imap/etc/pop3d.cnf* by **CN = server1.example.com** :

vi /usr/lib/courier-imap/etc/imapd.cnf

```
[.] CN = server1.example.com [.]
```

en /usr/lib/courier-imap/etc/pop3d.cnf

```
[.] CN = server1.example.com [.]
```

Then reconstruct these two certificates .

./mkimapdcert
./mkpop3dcert

. and restart courier-authlib, courier-imap:

/etc/init.d/courier-authlib restart
/etc/init.d/courier-imap restart

Run the command:

telnet localhost pop3

to see if your POP3 server works correctly by displaying the message + *OK Hello there* . (type **quit** to return to the Linux shell):

[root @ server1 share] # telnet localhost pop3
Trying :: 1 .
Connected to localhost.
Escape character is '^]'.
+ OK Hello there.

+ OK Better luck next time.
Connection closed by foreign host.
[root @ server1 share] #

11. Edit / etc / aliases

The next step is to open */ etc / aliases* . Make sure the postmaster points to *root* and *root* to your username or email, for example:

vi / etc / aliases

```
[.] postmaster: root root: postmaster@yourdomain.tld [.]
```

Or like this (if the administrator belongs to you):

```
[.] postmaster: root root: administrator [.]
```

Whenever you modify / etc / aliases, you must run the command:

newaliases

Then restart Postfix:

/etc/init.d/postfix restart

---

12. Install Amavisd-new, SpamAssassin and ClamAV

To install Amavisd-new, SpamAssassin and ClamAV at the same time, run the following command:

yum install amavisd-new spamassassin clamav clamav-data clamav-server clamav-update unzip bzip2

Now you need to do this, edit the */etc/amavisd/amavisd.conf* file.

/etc/amavisd/amavisd.conf

In this file we need to change 5 locations:

12.1: Change

```
$ mydomain = 'example.com';  # m?t m?c ??nh m?c ??nh cho các ph?n khác
```

City

```
$ mydomain = 'localhost';  # $ mydomain = 'example.com';  # m?t m?c ??
nh m?c ??nh cho các ??t khác
```

12.2: Change

```
  $ sa_tag_level_deflt = 2.0;  # add spam headers if at, or above that level $ sa
? sau m?t s? DSN không ???c g?i
```

City

```
  $ sa_tag_level_deflt = 2.0;  # add spam headers if at, or above that level $ sa
? sau mà m?t DSN không ???c g?
i # $ sa_tag_level_deflt = 2.0;  # add spam headers if at, or above that level #
? sau m?t s? DSN không ???c g?i
```

Here you can adjust the spam score as you like.

12.3: Change

```
  # @lookup_sql_dsn = # (['DBI: mysql: database = mail; host = 127.0.0.1; port =
```

City

```
  # @lookup_sql_dsn = # (['DBI: mysql: database = mail; host = 127.0.0.1; port =
```

12.4: Change

```
  # $ recipient_delimiter = '+'; # undef disables address extensions altogether #
```

City

```
 $recipient_delimiter = undef; # undef disables address extensions altogether # $
```

12.5: ??i

```
 $final_virus_destiny = D_DISCARD; $final_banned_destiny = D_BOUNCE; $final_spam_
```

City

```
 $final_virus_destiny = D_REJECT; $final_banned_destiny = D_REJECT; $final_spam_d
```

Sau khi thay ??i xong các ph?n này, file */etc/amavisd/amavisd.conf* s? trông nh? sau:

```
 use strict; # a minimalistic configuration file for amavisd-new with all necessa
 ]*>clean/m, # qr/(?i)
 ]*>infected/m, # qr/(?i)(.+)/m ], # ### http://www.sald.com/, http://www.dials.n
```