Postfix Email Server Documentation

Release 1.0

Ganapathi Chidambaram

Sep 21, 2018

Contents:

CHAPTER 1

About Postfix Email Server

Postfix is a free email server originally developed as an alternative,simpler and more secure to sendmail. This document will show you how to setup complete email server with postfix on Ubuntu 18.04 server.

1.1 Required Packages

- Postfix - Mail Transfer Agent (MTA)
- Dovecot - Local Delivery Agent(LDA) for incoming emails (IMAP & POP3)
- SASL - Simple Authentication and Secure Layer
- Postfixadmin - Web Interface to manage mailboxes,virtual domains and aliases.
- Nginx - Web Server to run Webmail Client & postfix admin
- MySQL - Database Storage for mail users and domains configurations.
- PHP - Web access for Postfix admin & Webmail Client

CHAPTER 2

Basics of Package Installation for Email Server

2.1 Nginx Installation

2.1.1 Installation

- Install the Nginx Web Server to run the postfix admin and other related web interface to run.

  ```
  apt install nginx
  ```

- Enable the Nginx Service to run on startup on machine.

  ```
  systemctl enable nginx
  ```

- Start the Nginx Web Server.

  ```
  systemctl start nginx
  ```

- Remove Default Nginx Site and add our custom site directory configuration.

  ```
  rm -f /etc/nginx/site-enabled/default
  ```

2.1.2 Setup TLS Certificates

A modern e-mail server can't be operated seriously without TLS certificates. We will use Let's Encrypt certificates for this purpose, as they are free and yet accepted by all browsers, mail clients and operating systems. If you already have valid certificates, you can use them instead.

- Install letsencrypt package on through ubuntu repository.

  ```
  apt install certbot
  ```

- Create directory and allocate necessary permission for letsencrypt domain verification.

```
mkdir -p /var/lib/letsencrypt/.well-known
chgrp www-data /var/lib/letsencrypt
chmod g+s /var/lib/letsencrypt
```

- Add mentioned below snippets for Letsencrypt configuration under **/etc/nginx/snippets/letsencrypt.conf**.

```
location ^~ /.well-known/acme-challenge/ {
allow all;
root /var/lib/letsencrypt/;
default_type "text/plain";
try_files $uri =404;
}
```

- Generate DH Params

```
openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```

- Create mentioned below snippets for SSL configuration under **/etc/nginx/snippets/ssl.conf**.

```
ssl_dhparam /etc/ssl/certs/dhparam.pem;
ssl_session_timeout 1d;
ssl_session_cache shared:SSL:50m;
ssl_session_tickets off;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers 'ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-
↪POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
↪SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-
↪RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-
↪SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-
↪SHA384:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-
↪AES256-SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-
↪SHA:DHE-RSA-AES256-SHA256:DHE-RSA-AES256-SHA:ECDHE-ECDSA-DES-CBC3-
↪SHA:ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:AES128-GCM-
↪SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-
↪SHA:DES-CBC3-SHA:!DSS';
ssl_prefer_server_ciphers on;
ssl_stapling on;
ssl_stapling_verify on;
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 30s;
add_header Strict-Transport-Security "max-age=15768000; includeSubdomains;
↪ preload";
add_header X-Frame-Options SAMEORIGIN;
add_header X-Content-Type-Options nosniff;
```

- Add mentioned below our custom site directory configuration into **/etc/nginx/site-enabled/postfix** as per our need.

```
server {
       listen [::]:80 default_server;
       root /var/www/html;
       index index.php index.html index.htm index.nginx-debian.html;
       server_name mail.hoppr.in;
       location / {
       try_files $uri $uri/ =404;
       }
       location ~ \.php$ {
```

(continues on next page)

(continued from previous page)

```
       include snippets/fastcgi-php.conf;
       fastcgi_pass unix:/run/php/php7.2-fpm.sock;
       fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
       include fastcgi_params;
       }
       location /rspamd/ {
               proxy_pass http://127.0.0.1:11334/;
               proxy_set_header Host $host;
               proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_
↪for;
       }
       # redirect server error pages to the static page /50x.html
       error_page 500 502 503 504 /50x.html;
       location = /50x.html {
       root /var/www/html;
       }
       location ~ /\.ht {
       deny all;
       }
       include snippets/letsencrypt.conf;
}
server {
       listen 443 ssl http2;
       root /var/www/html;
       index index.php index.html index.htm index.nginx-debian.html;
       server_name mail.hoppr.in;
       location / {
       try_files $uri $uri/ =404;
       }
       location ~ \.php$ {
               include snippets/fastcgi-php.conf;
   fastcgi_pass unix:/run/php/php7.2-fpm.sock;
   fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
   include fastcgi_params;
       }
       location /rspamd/ {
       proxy_pass http://127.0.0.1:11334/;
       proxy_set_header Host $host;
       proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
       }
       # redirect server error pages to the static page /50x.html
       error_page 500 502 503 504 /50x.html;
       location = /50x.html {
               root /var/www/html;
       }
       location ~ /\.ht {
       deny all;
       }
       ssl_certificate /etc/letsencrypt/live/mail.hoppr.in/fullchain.pem;
       ssl_certificate_key /etc/letsencrypt/live/mail.hoppr.in/privkey.
↪pem;
       ssl_trusted_certificate /etc/letsencrypt/live/mail.hoppr.in/chain.
↪pem;
       include snippets/ssl.conf;
       include snippets/letsencrypt.conf;
}
```

- Restart nginx for effective configuration

  ```
  systemctl restart nginx
  ```

- Generate certificate using below command.

  certbot certonly –standalone –rsa-key-size 4096 -d mail.hoppr.in -d imap.hoppr.in -d smtp.hoppr.in

- And letsencrypt certificate valid only for 90 days, so add cron jobs to auto renewal.

  ```
  certbot renew --pre-hook "systemctl stop nginx" --post-hook "systemctl␣
  ↪start nginx" --renew-hook "systemctl reload nginx; systemctl reload␣
  ↪dovecot; systemctl reload postfix"
  ```

2.2 PHP Packages Installation

- Below mentioned PHP packages required to run the php related tools which is used for postfix Email Server.

  ```
  apt install php-imap php-mbstring php7.2-imap php7.2-mbstring  php-fpm␣
  ↪php-mysql
  ```

- Set Timezone as per our local TimeZone on php configuration(/etc/php/7.2/fpm/php.ini) under value of **date.timezone**.

  ```
  date.timezone = Asia/Calcutta
  ```

- And Restart php to take effective

  ```
  systemctl restart php7.2-fpm
  ```

2.3 MySQL Installation

The mail server's virtual users and passwords are stored in a MySQL database. Dovecot and Postfix require this data. Follow the steps below to create the database tables for virtual users, domains and aliases.

```
apt install mysql-server
```

- Set Password for root user of MySQL.

  ```
  mysql_secure_installation
  ```

  Answer Y at the following prompts to secure mysql.

  – Change the root password?.

  – Remove anonymous users?.

  – Disallow root login remotely?.

  – Remove test database and access to it?.

  – Reload privilege tables now?.

CHAPTER 3

Postfix Admin - Web Interface for Manage Users

Postfix Admin is a web based interface to configure and manage a Postfix based email server for virtual users. PostfixAdmin requires php packages and one web server to run.

3.1 Download Postfixadmin

Normally postfixadmin is present on the default repositories of Ubuntu 18.04 but it will try to install Apache and PostgreSQL instead of Nginx and MySql. So, to keep our configuration, we will download the actual latest version 3.2 from source on the Github site project and save it in /opt folder.:

```
wget -P /opt https://github.com/postfixadmin/postfixadmin/archive/postfixadmin-3.2.
↪tar.gz
```

Now go to that folder and uncompress it.:

```
cd /opt && tar xvf postfixadmin-3.2.tar.gz
```

Now we should rename it.:

```
mv postfixadmin-postfixadmin-3.2/ postfixadmin
```

Normally we should move postfixadmin to the root folder of our Nginx web server but with this new version, we will only create a symbolic link of the public folder which contains the script then, rename it for some security

```
ln -s /opt/postfixadmin/public/ /var/www/html/pfa
```

3.2 Create the postfix database

Now we should connect to mysql database for some configurations

```
mysql -u root -p
```

Then create the database and the user

```
mysql> CREATE DATABASE postfix;
mysql> CREATE USER 'postfix'@'127.0.0.1' IDENTIFIED BY 'postfix-password';
mysql> GRANT ALL PRIVILEGES ON `postfix` . * TO 'postfix'@'127.0.0.1';
mysql> FLUSH PRIVILEGES;
mysql> exit
```

3.3 Configure postfixadmin

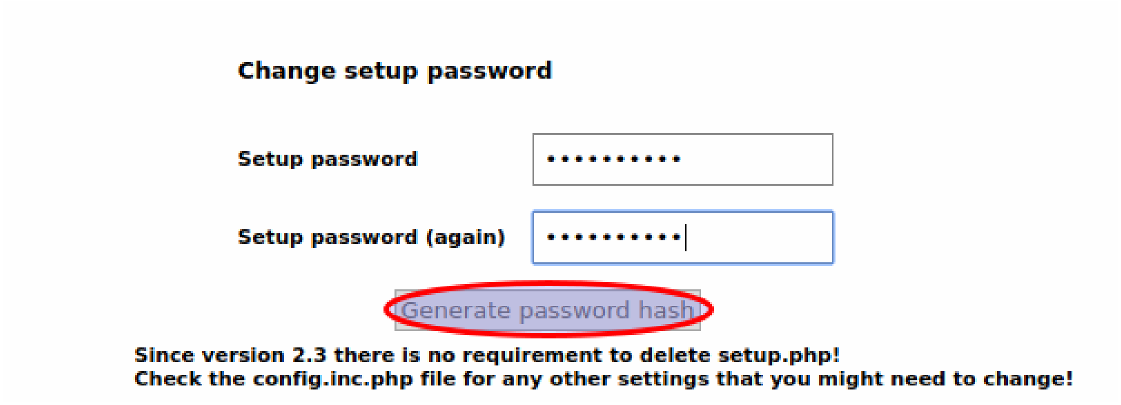
Now that we have configured the database, we should indicate it to postfixadmin so that it could know where to get the information. To do this, create the /opt/postfixadmin/config.local.php file and add the content on **/opt/postfixadmin/config.local.php** file.

```
<?php
$CONF['database_type'] = 'mysqli';
$CONF['database_user'] = 'postfix';
$CONF['database_password'] = 'postfix-db-password';
$CONF['database_name'] = 'postfix';
$CONF['configured'] = true;
?>
```

we should need to create **templates_c** directory manually and give www-data permission.

```
mkdir /opt/postfixadmin/templates_c
chmod 755 -R /opt/postfixadmin/templates_c
chown -R www-data:www-data /opt/postfixadmin/templates_c
```

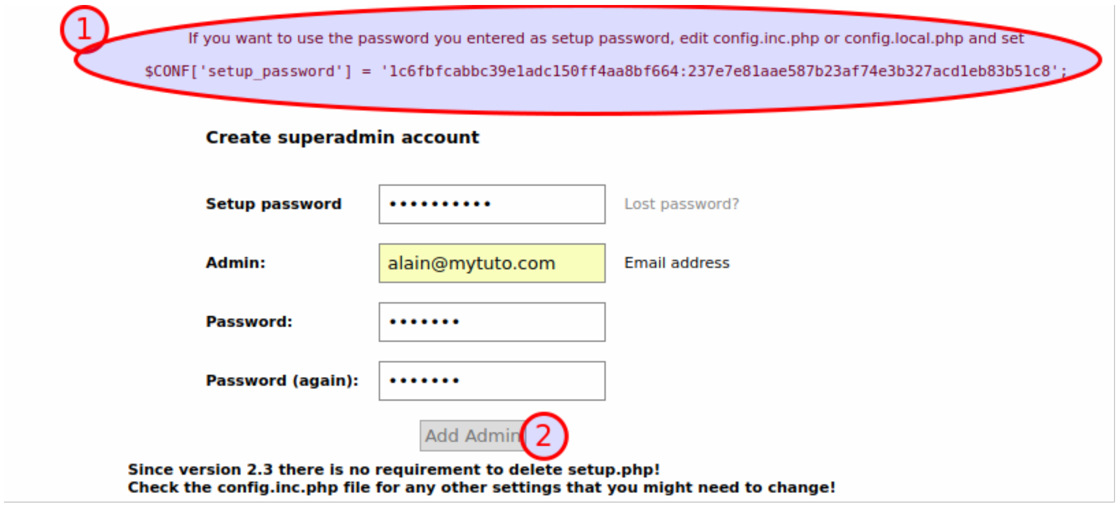
Then access https://yourdomain.com/pfa/setup.php you will see that it works now and it checks all the configuration, shows the version of your php and Ubuntu. Now we should create a password for the setup and generate its hash


Change setup password

Setup password	••••••••••
Setup password (again)	••••••••••

Since version 2.3 there is no requirement to delete setup.php!
Check the config.inc.php file for any other settings that you might need to change!

Now create your superuser account by filling below form.



If you want to use the password you entered as setup password, edit config.inc.php or config.local.php and set

```
$CONF['setup_password'] = '1c6fbfcabbc39e1adc150ff4aa8bf664:237e7e81aae587b23af74e3b327acd1eb83b51c8';
```

Create superadmin account

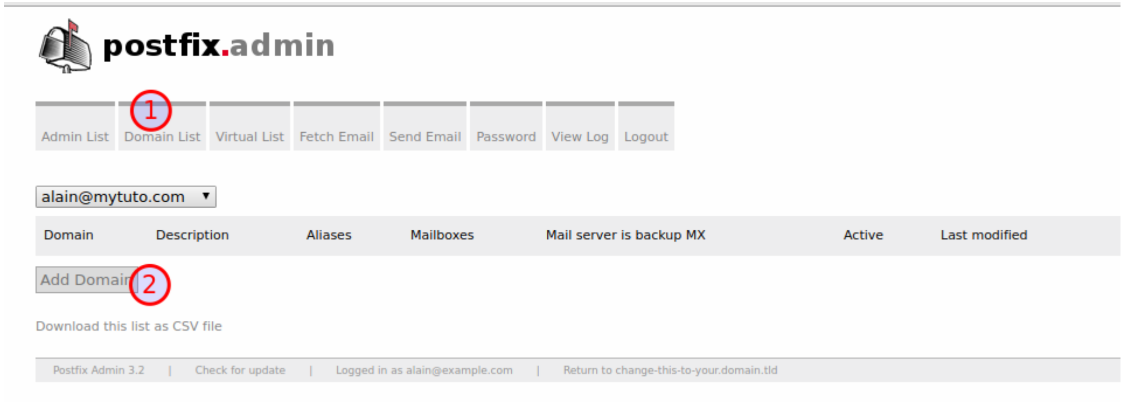
Setup password	••••••••••	Lost password?
Admin:	alain@mytuto.com	Email address
Password:	•••••••	
Password (again):	•••••••	

Since version 2.3 there is no requirement to delete setup.php!
Check the config.inc.php file for any other settings that you might need to change!

3.3.1 Create virtual domains, users, and alias

Now we will create the virtuals domains, then the users. A domain is a domain name such as mytuto.com. You can have emails on your domain using the same server. When you start you don't have any domain.

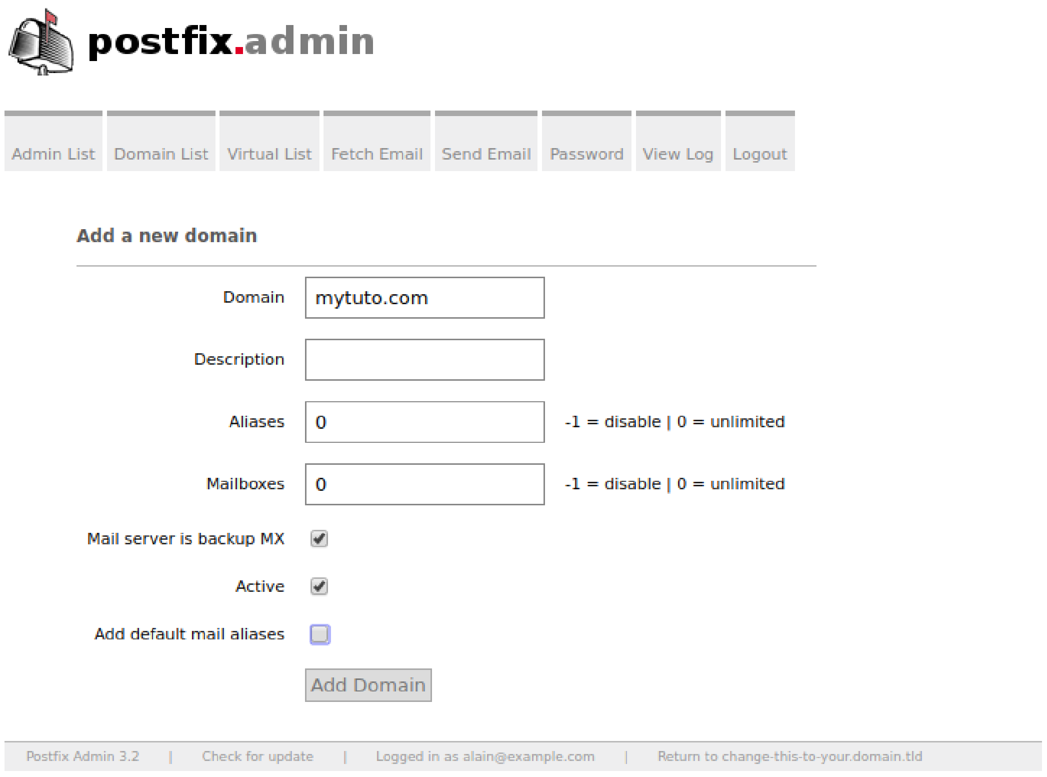
To add a domain, go to Domains List -> New Domain.



Domain	Description	Aliases	Mailboxes	Mail server is backup MX	Active	Last modified

Postfix Admin 3.2 | Check for update | Logged in as alain@example.com | Return to change-this-to-your.domain.tld

Then enter the information about the domain. You can limit the number of aliases and mailboxes. Remember to add the MX record. When finish to choose your values, add the domain.

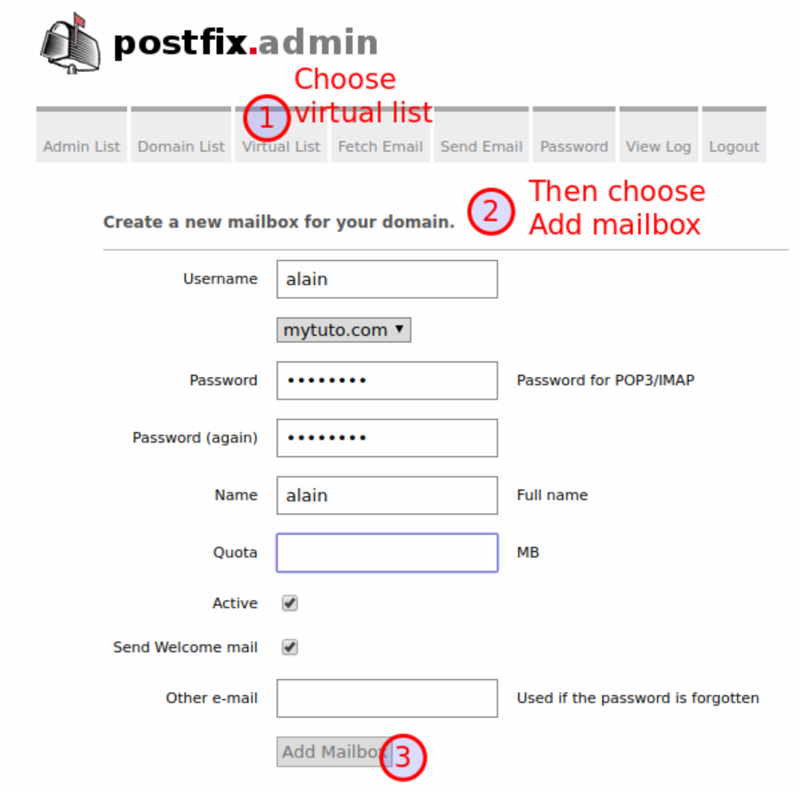


Add a new domain

Domain	mytuto.com	
Description		
Aliases	0	-1 = disable | 0 = unlimited
Mailboxes	0	-1 = disable | 0 = unlimited
Mail server is backup MX	✓	
Active	✓	
Add default mail aliases	☐	

Postfix Admin 3.2 | Check for update | Logged in as alain@example.com | Return to change-this-to-your.domain.tld

We can now create our virtual users. To do this, go to Virtual List -> Add mailbox.



Create a new mailbox for your domain.

Username	alain	
	mytuto.com ▾	
Password	••••••••	Password for POP3/IMAP
Password (again)	••••••••	
Name	alain	Full name
Quota		MB
Active	✓	
Send Welcome mail	✓	
Other e-mail		Used if the password is forgotten

CHAPTER 4

Postfix - Mail Transfer Agent

Postfix is a free and open-source mail transfer agent that routes and delivers electronic mail.

4.1 Install postfix

Now we can install the postfix packages.

```
apt install postfix postfix-mysql sasl2-bin
```

You will have to answer two question about the type of mail and the name of your mail server. Make sure to replace the hostname and domain values with yours

- the type of mail configuration: Internet Site
- the system mail name: hostname.domain.com

Make sure that sasl run at the startup by editing its configuration file(/etc/default/saslauthd)

```
# Should saslauthd run automatically on startup? (default: no)
START=yes
```

Now restart the service

```
# systemctl restart saslauthd
```

As we are configuring a mail server with virtual users, we need an owner of all mailboxes so will create a system user which will be used by all virtual users to access email on the server. First, create the group owner and the folder which will store the mailboxes.

```
# groupadd -g 5000 vmail && mkdir -p /var/mail/vmail
```

Now create the owner

```
# useradd -u 5000 vmail -g vmail -s /usr/sbin/nologin -d /var/mail/vmail
```

Make sure to give the permission of the mail directory to the owner so that it can store the mails into the appropriate directories.

```
# chown -R vmail:vmail /var/mail/vmail
```

If you don't do this, dovecot will not be able to create the required folders to store the emails.

4.2 Create the configuration files for the database

Now create a folder which will contain some database files

```
# mkdir -p /etc/postfix/sql
```

Postfix need 03 database files which will allow it to access the database that we created earlier:

Domains to contain the list of domain names hosted on the server. it will allow postfix to determine if our server is in charge of a domain (mytuto.com) when it receives an email (user@mytuto.com) on it. If it's the case, it will mean that the domain is in our database.

```
# vim /etc/postfix/sql/mysql_virtual_domains_maps.cf
user = postfix
password = postfix-db-password
hosts = 127.0.0.1
dbname = postfix
query = SELECT domain FROM domain WHERE domain='%s' AND active = '1'
```

We will enable the configuration and add it automatically to the /etc/postfix/main.cf file and reload the postfix configuration to avoid having to do it manually. So the file will be updated everytime you use this command with new values.

```
# postconf -e virtual_mailbox_domains=mysql:/etc/postfix/sql/mysql_virtual_
↪domains_maps.cf
```

Now we can check the configuration. We will run a command that will execute the query contained in the file in order to search for a domain in our database. An element (the searched domain) must be returned or nothing if the domain is not present.

```
# postmap -q mytuto.com mysql:/etc/postfix/sql/mysql_virtual_domains_maps.cf
mytuto.com
```

As you can see, postfix is able to retrieve the domains stored in our database

Mailbox to store all the virtual email addresses. It will be used to verify also if the mailboxes exist

```
# vim /etc/postfix/sql/mysql_virtual_mailbox_maps.cf
user = postfix
password = postfix-db-password
hosts = 127.0.0.1
dbname = postfix
query = SELECT maildir FROM mailbox WHERE username='%s' AND active = '1'
```

Now let's update the configuration file

```
# postconf -e virtual_mailbox_maps=mysql:/etc/postfix/sql/mysql_virtual_
↪mailbox_maps.cf
```

Run the command to test the query on the database

```
# postmap -q alain@mytuto.com mysql:/etc/postfix/sql/mysql_virtual_mailbox_
↪maps.cf
mytuto.com/alain/
```

Alias to contain the different email aliases.

```
# vim /etc/postfix/sql/mysql_virtual_alias_maps.cf
user = postfix
password = postfix-db-password
hosts = 127.0.0.1
dbname = postfix
query = SELECT goto FROM alias WHERE address='%s' AND active = '1'
```

Now add the configuration

```
# postconf -e virtual_alias_maps=mysql:/etc/postfix/sql/mysql_virtual_alias_
↪maps.cf
```

Now run the command to test the query. It is the destination user (alain@mytuto.com) that should be displayed and not the abuse address. It shows that postfix can do the matching.

```
# postmap -q abuse@mytuto.com mysql:/etc/postfix/sql/mysql_virtual_alias_
↪maps.cf
alain@mytuto.com
```

Make sure that those files are not readable by the normal users because the passwords are stored in clear. In order for postfix to read those file, we can change the group owner to postfix

```
# chgrp postfix /etc/postfix/sql/mysql_*.cf
```

4.3 Configure postfix

Now we will manually edit the postfix main configuration file. So, make a copy before editing.

```
# cp /etc/postfix/main.cf /etc/postfix/main.cf.bak
```

Now we will activate SASL to force authentication for sending emails and hand off authentication to Dovecot. Be sure to add lines below

```
# vim /etc/postfix/main.cf
# --------------------------------------
myhostname = mail.mytuto.com
mydomain = mytuto.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
virtual_alias_domains = mail.mytuto.com
mydestination = $myhostname, mail.mytuto.com, ip-172-30-1-40, localhost.
↪localdomain, localhost
```

(continues on next page)

(continued from previous page)

```
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
# -------------------------------------
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
# -------------------------------------
virtual_mailbox_domains = mysql:/etc/postfix/sql/mysql_virtual_domains_maps.
↪cf
virtual_mailbox_maps = mysql:/etc/postfix/sql/mysql_virtual_mailbox_maps.cf
virtual_alias_maps = mysql:/etc/postfix/sql/mysql_virtual_alias_maps.cf
# -------------------------------------
## Path to the Postfix auth socket
smtpd_sasl_path = private/auth
smtp_sasl_path = private/auth
# -------------------------------------
## Tells Postfix to let people send email if they've authenticated to the␣
↪server.
## Otherwise they can only send if they're logged in (SSH)
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtp_sasl_security_options = noanonymous
smtpd_sasl_local_domain = $myhostname
# -------------------------------------
# TLS parameters
smtpd_use_tls=yes
smtp_use_tls = yes
smtpd_tls_security_level = may
smtpd_tls_auth_only = yes
smtp_tls_security_level = may
smtpd_tls_cert_file=/etc/letsencrypt/live/mail.mytuto.com/fullchain.pem
smtpd_tls_key_file=/etc/letsencrypt/live/mail.mytuto.com/privkey.pem
smtp_tls_cert_file=/etc/letsencrypt/live/mail.mytuto.com/fullchain.pem
smtp_tls_key_file=/etc/letsencrypt/live/mail.mytuto.com/privkey.pem
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtpd_sender_restrictions = permit_sasl_authenticated
smtpd_recipient_restrictions = check_recipient_access hash:/etc/postfix/
↪custom_replies
```

Now let's edit the /etc/postfix/master.cf configuration file. It's the process configuration file. We will enable secure SMTP ports by adding or uncomment the lines below and make a copy before.

```
# cp /etc/postfix/master.cf /etc/postfix/master.cf.bak
```

```
# vim /etc/postfix/master.cf
submission inet n       -       y       -       -       smtpd
       -o syslog_name=postfix/submission
       -o smtpd_tls_security_level=encrypt
       -o smtpd_tls_ask_ccert=yes
       -o smtpd_sasl_auth_enable=yes
       -o smtpd_reject_unlisted_recipient=no
       -o smtpd_client_restrictions=permit_sasl_authenticated,reject
       -o smtpd_recipient_restrictions=permit_sasl_authenticated,reject
       -o milter_macro_daemon_name=ORIGINATING
smtps     inet  n       -       y       -       -       smtpd
```

(continues on next page)

(continued from previous page)

```
       -o syslog_name=postfix/smtps
       -o smtpd_tls_wrappermode=yes
       -o smtpd_sasl_auth_enable=yes
       -o smtpd_client_restrictions=permit_sasl_authenticated,reject
       -o milter_macro_daemon_name=ORIGINATING
```

Now you can run the postconf -n command to check some errors.

```
# postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
...
...
```

If you have no warning messages, it means that your files do not contain errors. Now you can restart the postfix service.

```
# systemctl restart postfix
# systemctl status postfix
       * postfix.service - Postfix Mail Transport Agent
       Loaded: loaded (/lib/systemd/system/postfix.service; enabled; vendor␣
↪preset: enabled)
       Active: active (exited) since Wed 2018-09-22 10:16:02 UTC; 27s ago
       Process: 12225 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
       Main PID: 12225 (code=exited, status=0/SUCCESS)
```

CHAPTER 5

Dovecot - IMAP & POP3 Handler

Dovecot is an open-source IMAP and POP3 server for Linux/UNIX-like systems, written primarily with security in mind. Now that Postfix is installed and configured, we need to install postfix to manage the pop and imap protocols, which allow us to recover our emails.

5.1 Installation of Dovecot

Dovecot packages are presents in the Ubuntu 18.04 default repositories. We will install it with the mysql support. We will install sieve which is useful because it will automatically put the mails into the corresponding folders. It means that, for each domain, it will create a corresponding folder containing the corresponding folder of a virtual user to store its email files.

```
apt install dovecot-imapd dovecot-mysql dovecot-managesieved
```

5.2 Configuration of Dovecot

Now go to the folder containing the configuration files.

```
# cd /etc/dovecot/conf.d
```

- **10-auth.conf file to modify the connection mechanisms by adding or uncommenting the lines.** Dovecot uses the system users by default but we use Mysql users

  ```
  auth_mechanisms = plain login
  #!include auth-system.conf.ext
  !include auth-sql.conf.ext
  ```

- **auth-sql.conf.ext** file for the sql configuration

```
passdb {
        driver = sql
        args = /etc/dovecot/dovecot-sql.conf.ext
}
userdb {
        driver = static
        args = uid=vmail gid=vmail home=/var/mail/vmail/%d/%n
}
```

- **/etc/dovecot/dovecot-sql.conf.ext** to tell dovecot how to connect to the SQL database

```
driver = mysql
connect = host=127.0.0.1 dbname=postfix user=postfix password=postfix-db-
↪password
password_query = SELECT username,domain,password FROM mailbox WHERE␣
↪username='%u';
default_pass_scheme = MD5-CRYPT
```

- **10-mail.conf** file to configure the mail location directory

```
mail_location = maildir:/var/mail/vmail/%d/%n/Maildir
mail_privileged_group = mail
```

- **10-master.conf** file for the connection to the socket.

```
service auth {
        unix_listener auth-userdb {
                mode = 0600
                user = vmail
        }
        unix_listener /var/spool/postfix/private/auth {
                mode = 0660
                user = postfix
                group = postfix
        }
        user = dovecot
}
```

- **15-lda.conf** file to indicate sieve in order to automatically organize mail into the corresponding folder

```
protocol lda {
# Space separated list of plugins to load (default is global mail_
↪plugins).
mail_plugins = $mail_plugins sieve
}
```

We should give permission if we want that the vmail user can launch dovecot

```
# chgrp vmail /etc/dovecot/dovecot.conf
```

Now you can restart the dovecot service

```
# systemctl restart dovecot
```

5.3 Integrate dovecot to postfix

Now that we have configured dovecot, we should indicate postfix to work with dovecot. Edit the master postfix configuration file(**/etc/postfix/master.cf**) and add the lines below at the end of the file

```
dovecot   unix  -       n       n       -       -       pipe
flags=DRhu user=vmail:vmail argv=/usr/lib/dovecot/deliver -f ${sender} -d $
↪{user}@${nexthop}
```

Now edit the main postfix configuration file (**/etc/postfix/main.cf**)

```
# Allow authenticated users to send email, and use Dovecot to authenticate␣
↪them. Tells Postfix to use Dovecot for authentication
virtual_transport = dovecot
dovecot_destination_recipient_limit = 1
smtpd_sasl_type = dovecot
smtp_sasl_type = dovecot
```

Then restart postfix

```
# systemctl restart postfix
```

CHAPTER 6

Delivery - Better Delivery of Email

To Avoid Spam need to ensure some configuration on mail server for smooth delivery process.

- DKIM Signing
- DMARC
- SPF

6.1 DKIM Signing

Install opendkim package for dkim signing.

```
apt install opendkim opendkim-tools postfix-policyd-spf-python postfix-pcre
```

Create folder for dkim key and allocate necessary permission to access.

```
mkdir -p /etc/opendkim/keys/mytuto.com
chown -R opendkim:opendkim /etc/opendkim
chmod go-rw /etc/opendkim/keys
```

Generate opendkim for signing email for better delivery.

```
opendkim-genkey -b 2048 -D /etc/opendkim/keys/mytuto.com -h rsa-sha256 -r -s␣
↪dkim -d mytuto.com -v
```

And configure **/etc/opendkim.conf** to use the generated key for signing.

```
# selector '2007' (e.g. 2007._domainkey.example.com)
Domain     mytuto.com
KeyFile     /etc/opendkim/keys/mytuto.com/dkim.private
Selector    dkim
SOCKET     inet:8891@127.0.0.1
Canonicalization        relaxed/simple
```

(continues on next page)

(continued from previous page)

```
Mode                        sv
SubDomains                  no
AutoRestart         yes
AutoRestartRate     10/1M
Background          yes
DNSTimeout          5
SignatureAlgorithm  rsa-sha256
```

And add DNS TXT entry with content of dkim.txt which is generated from opendkim-genkey

```
      #cat /etc/opendkim/keys/mytuto.com/dkim.txt
      dkim._domainkey IN      TXT      ( "v=DKIM1; h=rsa-sha256; k=rsa;␣
↪s=email; "

↪"p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtNchMEHZ4U+7sYE69ZapO+hCPgbqx87muMKwwcM/
↪voqrgLhCv/OOnHhcawoCb6buCwVrb+GgU0hHS+UqcTsFS3BTeFuPis5fXdoXzqUgOj1q6k/
↪wqlscYRQJq+M+j+cufR2i7e8O1DQ/
↪KO8tCjkZenOhPYZ8LA6HaagMTQgyGBP8HqgAMsY2PEGchdfB2SezGrZ1ZogvoUeGaH"
"2A9AmUGJQzU3SPAbBs53v6SG5ePrhTRf6spC47THccCJfE7za5smMjVzkO9jD85XyQvAR6q/
↪jVtaM9HbLT6+ipcydmaMT/9+SOG5JvvDHPrnDEAPKf3oTKSEmCa1VRKJNWCi8EpQIDAQAB" )
```

Now integrate opendkim with postfix to use opendkim key by adding below line at end of file(**/etc/postfix/main.cf**)

```
# DKIM
# --------------------------------------
milter_default_action = accept
milter_protocol = 2
smtpd_milters = inet:127.0.0.1:8891
non_smtpd_milters = inet:127.0.0.1:8891
```

6.2 SPF Record

The value in an SPF DNS record will look something like the following examples.

Example 1 Allow mail from all hosts listed in the MX records for the domain:

```
v=spf1 mx -all
```

Example 2 Allow mail from a specific host:

```
v=spf1 a:mail.mytuto.com -all
```

- The **v=spf1** tag is required and has to be the first tag.
- The last tag, **-all**, indicates that mail from your domain should only come from servers identified in the SPF string.Anything coming from any other source is forging your domain. An alternative is **~all**, indicating the same thing but also indicating that mail servers should accept the message and flag it as forged instead of rejecting it outright. **-all** makes it harder for spammers to forge your domain successfully; it is the recommended setting. **~all** reduces the chances of email getting lost because an incorrect mail server was used to send mail. **~all** can be used if you don't want to take chances.

The tags between identify eligible servers from which email to your domain can originate.

- **mx** is a shorthand for all the hosts listed in MX records for your domain. If you've got a solitary mail server, mx is probably the best option. If you've got a backup mail server (a second MX record), using mx won't cause any problems.Your backup mail server will be identified as an authorized source for email although it will probably never send any.

- The **a** tag lets you identify a specific host by name or IP address, letting you specify which hosts are authorized. You'd use a if you wanted to prevent the backup mail server from sending outgoing mail or if you wanted to identify hosts other than your own mail server that could send mail from your domain (e.g., putting your ISP's outgoing mail servers in the list so they'd be recognized when you had to send mail through them).

For now, we're going to stick with the mx version. It's simpler and correct for most basic configurations, including those that handle multiple domains. To add the record, go to your DNS management interface and add a record of type TXT for your domain itself (i.e., a blank hostname) containing this string:

```
mytuto.com TXT v=spf1 mx -all
```

6.3 DMARC

Add below entries on your DNS Server for DMARC record for some mail server's better email delivery.

```
_dmarc TXT  "v=DMARC1; p=none; adkim=r; aspf=r;"
```

The **none** indicates that the remove server should not drop the mails, even if they are not coming from the servers listed in the SPF record. Once you're sure everything is fine, change the **none** to **reject**.