Ubuntu 18.04 » Ubuntu Server Guide » Email Services »

Postfix

Previous Next

Postfix is the default Mail Transfer Agent (MTA) in Ubuntu. It attempts to be fast and easy to administer and secure. It is compatible with the MTA *sendmail*. This section explains how to install and configure *postfix*. It also explains how to set it up as an SMTP server using a secure connection (for sending emails securely).

This guide does not cover setting up Postfix *Virtual Domains*, for information on Virtual Domains and other advanced configurations see References.

Installation

To install *postfix* run the following command:

```
sudo apt install postfix
```

Simply press return when the installation process asks questions, the configuration will be done in greater detail in the next stage.

Basic Configuration

To configure *postfix*, run the following command:

```
sudo dpkg-reconfigure postfix
```

The user interface will be displayed. On each screen, select the following values:

1. Internet Site
2. mail.example.com
3. steve
4. mail.example.com, localhost.localdomain, localhost
5. No
6. 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.0.0/24
7. 0
8. +
9. all

Replace mail.example.com with the domain for which you'll accept email, 192.168.0.0/24 with the actual network and class range of your mail server, and steve with the appropriate username.

Now is a good time to decide which mailbox format you want to use. By default Postfix will use **mbox** for the mailbox format. Rather than editing the configuration file directly, you can use the `postconf` command to configure all *postfix* parameters. The configuration parameters will be stored in /etc/postfix/main.cf file. Later if you wish to re-configure a particular parameter, you can either run the command or change it manually in the file.

To configure the mailbox format for **Maildir:**

```
sudo postconf -e 'home_mailbox = Maildir/'
```

This will place new mail in /home/*username*/Maildir so you will need to configure your Mail Delivery Agent (MDA) to use the same path.

SMTP Authentication

SMTP-AUTH allows a client to identify itself through an authentication mechanism (SASL). Transport Layer Security (TLS) should be used to encrypt the authentication process. Once authenticated the SMTP server will allow the client to relay mail.

1. Configure Postfix for SMTP-AUTH using SASL (Dovecot SASL):

   ```
   sudo postconf -e 'smtpd_sasl_type = dovecot'
   sudo postconf -e 'smtpd_sasl_path = private/auth'
   sudo postconf -e 'smtpd_sasl_local_domain ='
   sudo postconf -e 'smtpd_sasl_security_options = noanonymous'
   sudo postconf -e 'broken_sasl_auth_clients = yes'
   sudo postconf -e 'smtpd_sasl_auth_enable = yes'
   sudo postconf -e 'smtpd_recipient_restrictions = \
   permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination'
   ```

   The *smtpd_sasl_path* configuration is a path relative to the Postfix queue directory.

2. Next, generate or obtain a digital certificate for TLS. See Certificates for details. This example also uses a Certificate Authority (CA). For information on generating a CA certificate see Certification Authority.

   MUAs connecting to your mail server via TLS will need to recognize the certificate used for TLS. This can either be done using a certificate from a commercial CA or with a self-signed certificate that users manually install/accept. For MTA to MTA TLS certficates are never validated without advance agreement from the affected organizations. For MTA to MTA TLS, unless local policy requires it, there is no reason not to use a self-signed certificate. Refer to Creating a Self-Signed Certificate for more details.

3. Once you have a certificate, configure Postfix to provide TLS encryption for both incoming and outgoing mail:

   ```
   sudo postconf -e 'smtp_tls_security_level = may'
   sudo postconf -e 'smtpd_tls_security_level = may'
   sudo postconf -e 'smtp_tls_note_starttls_offer = yes'
   sudo postconf -e 'smtpd_tls_key_file = /etc/ssl/private/server.key'
   sudo postconf -e 'smtpd_tls_cert_file = /etc/ssl/certs/server.crt'
   sudo postconf -e 'smtpd_tls_loglevel = 1'
   sudo postconf -e 'smtpd_tls_received_header = yes'
   sudo postconf -e 'myhostname = mail.example.com'
   ```

4. If you are using your own *Certificate Authority* to sign the certificate enter:

   ```
   sudo postconf -e 'smtpd_tls_CAfile = /etc/ssl/certs/cacert.pem'
   ```

   Again, for more details about certificates see Certificates.

After running all the commands, *Postfix* is configured for SMTP-AUTH and a self-signed certificate has been created for TLS encryption.

Now, the file /etc/postfix/main.cf should look like this:

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete
# version

smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no

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

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

myhostname = server1.example.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = server1.example.com, localhost.example.com, localhost
relayhost =
mynetworks = 127.0.0.0/8
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
smtpd_sasl_local_domain =
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions =
permit_sasl_authenticated,permit_mynetworks,reject _unauth_destination
smtpd_tls_auth_only = no
smtp_tls_security_level = may
smtpd_tls_security_level = may
smtp_tls_note_starttls_offer = yes
smtpd_tls_key_file = /etc/ssl/private/smtpd.key
smtpd_tls_cert_file = /etc/ssl/certs/smtpd.crt
```

```
smtpd_tls_CAfile = /etc/ssl/certs/cacert.pem
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
```

The postfix initial configuration is complete. Run the following command to restart the postfix daemon:

```
sudo systemctl restart postfix.service
```

Postfix supports SMTP-AUTH as defined in RFC2554. It is based on SASL. However it is still necessary to set up SASL authentication before you can use SMTP-AUTH.

Configuring SASL

Postfix supports two SASL implementations Cyrus SASL and Dovecot SASL. To enable Dovecot SASL the *dovecot-core* package will need to be installed. From a terminal prompt enter the following:

```
sudo apt install dovecot-core
```

Next you will need to edit /etc/dovecot/conf.d/10-master.conf. Change the following:

```
service auth {
  # auth_socket_path points to this userdb socket by default. It's typically
  # used by dovecot-lda, doveadm, possibly imap process, etc. Its default
  # permissions make it readable only by root, but you may need to relax these
  # permissions. Users that have access to this socket are able to get a list
  # of all usernames and get results of everyone's userdb lookups.
  unix_listener auth-userdb {
    #mode = 0600
    #user =
    #group =
  }

  # Postfix smtp-auth
  unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    user = postfix
    group = postfix
  }
```

In order to let *Outlook* clients use SMTP-AUTH, in the *authentication mechanisms* section of /etc/dovecot/conf.d/10-auth.conf change this line:

```
auth_mechanisms = plain
```

To this:

```
auth_mechanisms = plain login
```

Once you have *Dovecot* configured restart it with:

```
sudo systemctl restart dovecot.service
```

Mail-Stack Delivery

Another option for configuring *Postfix* for SMTP-AUTH is using the *mail-stack-delivery* package (previously packaged as dovecot-postfix). This package will install *Dovecot* and configure *Postfix* to use it for both SASL authentication and as a Mail Delivery Agent (MDA).

> You may or may not want to run IMAP, IMAPS, POP3, or POP3S on your mail server. For example, if you are configuring your server to be a mail gateway, spam/virus filter, etc. If this is the case it may be easier to use the above commands to configure Postfix for SMTP-AUTH than using *mail-stack-delivery*.

To install the package, from a terminal prompt enter:

```
sudo apt install mail-stack-delivery
```

You should now have a working mail server, but there are a few options that you may wish to further customize. For example, the package uses the certificate and key from the *ssl-cert* (self signed) package, and in a production environment you should use a certificate and key generated for the host. See Certificates for more details.

Once you have a customized certificate and key for the host, change the following options for postfix in /etc/postfix/main.cf to match your new keys:

```
smtpd_tls_cert_file = #yourcertfile#
smtpd_tls_key_file = #yourkeyfile#
```

And for *Dovecot* in /etc/dovecot/conf.d/10-ssl.conf:

```
ssl_cert = <#yourcertfile#
ssl_key = <#yourkeyfile#
```

Then restart Postfix:

```
sudo systemctl restart postfix.service
```

Testing

SMTP-AUTH configuration is complete. Now it is time to test the setup.

To see if SMTP-AUTH and TLS work properly, run the following command:

```
telnet mail.example.com 25
```

After you have established the connection to the postfix mail server, type:

```
ehlo mail.example.com
```

If you see the following lines among others, then everything is working perfectly. Type `quit` to exit.

```
250-STARTTLS
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250 8BITMIME
```

Troubleshooting

This section introduces some common ways to determine the cause if problems arise.

Escaping chroot

The Ubuntu *postfix* package will by default install into a *chroot* environment for security reasons. This can add greater complexity when troubleshooting problems.

To turn off the chroot operation locate for the following line in the /etc/postfix/master.cf configuration file:

```
smtp      inet  n       -       -       -       -       smtpd
```

and modify it as follows:

```
smtp      inet  n       -       n       -       -       smtpd
```

You will then need to restart Postfix to use the new configuration. From a terminal prompt enter:

```
sudo systemctl restart postfix.service
```

Smtps

If you need smtps, edit /etc/postfix/master.cf and uncomment the following line:

```
smtps     inet  n       -       -       -       -       smtpd
  -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
```

Log Files

Postfix sends all log messages to /var/log/mail.log. However error and warning messages can sometimes get lost in the normal log output so they are also logged to /var/log/mail.err and /var/log/mail.warn respectively.

To see messages entered into the logs in real time you can use the *tail -f* command:

```
tail -f /var/log/mail.err
```

The amount of detail that is recorded in the logs can be increased. Below are some configuration options for increasing the log level for some of the areas covered above.

1. To increase *TLS* activity logging set the *smtpd_tls_loglevel* option to a value from 1 to 4.

   ```
   sudo postconf -e 'smtpd_tls_loglevel = 4'
   ```

2. If you are having trouble sending or receiving mail from a specific domain you can add the domain to the *debug_peer_list* parameter.

   ```
   sudo postconf -e 'debug_peer_list = problem.domain'
   ```

3. You can increase the verbosity of any *Postfix* daemon process by editing the /etc/postfix/master.cf and adding a *-v* after the entry. For example edit the *smtp* entry:

   ```
   smtp      unix  -       -       -       -       -       smtp -v
   ```

It is important to note that after making one of the logging changes above the *Postfix* process will need to be reloaded in order to recognize the new configuration: `sudo systemctl reload postfix.service`

1. To increase the amount of information logged when troubleshooting *SASL* issues you can set the following options in `/etc/dovecot/conf.d/10-logging.conf`

```
auth_debug=yes
auth_debug_passwords=yes
```

Just like *Postfix* if you change a *Dovecot* configuration the process will need to be reloaded: `sudo systemctl reload dovecot.service`.

Some of the options above can drastically increase the amount of information sent to the log files. Remember to return the log level back to normal after you have corrected the problem. Then reload the appropriate daemon for the new configuration to take affect.

References

Administering a *Postfix* server can be a very complicated task. At some point you may need to turn to the Ubuntu community for more experienced help.

A great place to ask for *Postfix* assistance, and get involved with the Ubuntu Server community, is the *#ubuntu-server* IRC channel on freenode. You can also post a message to one of the Web Forums.

For in depth *Postfix* information Ubuntu developers highly recommend: The Book of Postfix.

Finally, the Postfix website also has great documentation on all the different configuration options available.

Also, the Ubuntu Wiki Postfix page has more information.

Previous Next