Configuring Mail Services

Before configuring an e-mail client or server, you need to understand how e-mail works and the programs to use or make available to your users. Several key components are essential for e-mail to work properly, and as a system administrator it is your responsibility to configure the following items. These items are explained in more detail later in this chapter.

◆ Programs:

■ A Mail User Agent (MUA) for users to be able to read and write e-mail

■ A Mail Transfer Agent (MTA) to deliver the e-mail messages between computers across a network

■ A Local Delivery Agent (LDA) to deliver messages to users' mailbox files

■ An mail notification program to tell users that they have new mail

◆ The TCP/IP protocols for storing e-mail messages and transferring e-mail between MTAs

◆ Other communication and mail storage components:

■ Ports
■ Mail queues
■ Mailbox files

Mail User Agent (MUA)

To be able to send mail, you, or your users, need a program called a Mail User Agent (MUA). The MUA, also called a mail client, enables users to write and read mail messages. Two types of MUAs are available: a graphical user interface (GUI), such as Netscape Messenger, and a command-line interface, such as Pine. Whether your MUA is a GUI or command-line interface, after the message is composed, the MUA sends it to the mail transfer agent (MTA). The MTA is the program that sends the message out across the network and does its work without any intervention by the user. In fact, most users are unaware of the MTA, they just see their mail client.

Mail Transfer Agent (MTA)

Now that the MTA has received the message from the MUA, it can do its job. The MTA installed by default on your Red Hat system is called Sendmail. The MTAChapter 14: Configuring Mail Services
reads the information in the To section of the e-mail message and determines the IP address of the recipient's mail server. Then the MTA tries to open a connection to the recipient's server through a communication port, typically port 25. If the MTA on the sending machine can establish a connection, it sends the message to the MTA on the recipient's server using the Simple Message Transfer Protocol (SMTP). The MTA on the receiving server adds header information to the message. The

header contains information that is used for tracking the message and ensuring that it is delivered. Next the receiving MTA passes the message to another program to inform the receiver that new mail has arrived.

Local Delivery Agent (LDA)

After the LDA receives the message from the MTA, it places the message in the receiver's mailbox file that is identified by the username. On your Red Hat system this is a program called procmail. The location of the user's mailbox file is /usr/spool/mail/<user's name> .

The final step in the process happens when the user who is the intended receiver of the message reads the message. The user does this using the MUA on his or her PC. An optional program is a mail notifier that periodically checks your mailbox file for new mail. If you have such a program installed, it notifies you of the new mail. The Red Hat Linux shell has a built-in mail notifier that looks at your mailbox file once a minute. If new mail has arrived, the shell displays a message just before it displays the next system prompt. It won't interrupt a program you're running. You can adjust how frequently the mail notifier checks and even which mailbox files to watch.

If you are using a GUI, there are mail notifiers available that play sounds or display pictures to let you know that new mail has arrived.

Introducing SMTP

In the section describing the MTA, you learned that messages are sent between MTAs using SMTP. This section explains SMTP and two other protocols used to send mail, Post Office Protocol (POP3) and Internet Message Access Protocol (IMAP4). SMTP is the TCP/IP protocol for transferring e-mail messages between computers on a network. SMTP specifies message movement between MTAs, by the path the message takes. Messages may go directly from the sending to the receiving MTA or through other MTAs on other network computers.

The SMTP protocol can transfer only ASCII text. It can't handle fonts, colors, graphics, or attachments. If you want to be able to send these items, you need to add another protocol to SMTP.

Understanding POP3

POP3 is the Post Office Protocol version 3. This protocol runs on a server that is connected to a network and continuously sends and receives mail. The POP3 server stores any messages it receives. POP3 was developed to solve the problem of what happens to messages when the recipient is not connected to the network.

Without POP3, the message could not be sent to the recipient if the recipient were offline. But with POP3, when you want to check your e-mail, you connect to the POP3 server to retrieve your messages that were stored by the server. After you retrieve your messages, you can use the MUA on your PC to read them. Of course, your MUA has to understand the POP3 to be able to communicate with the POP3 server.

The messages you retrieve to your PC are then typically removed from the server. This means that they are no longer available to you if you want to retrieve them to another PC.

Understanding IMAP4

The Internet Message Access Protocol version 4 (IMAP4) provides sophisticated client/server functionality for handling e-mail. IMAP4 has more features than POP3. IMAP4 enables you to store your e-mail on a networked mail server, just as POP3 does. The difference is that POP3 requires you to download your e-mail before your MUA reads it, whereas IMAP4 enables your e-mail to reside permanently on a remote server, from which you can access your mail. And you can do so from your office, your home, or anywhere else. Your MUA must understand IMAP4 to retrieve messages from an IMAP4 server.

This tutorial explains how to setup mail server on ubuntu 14.04 using postfix,dovecot and squirrelmail.
» **Postfix** (for sending)
» **Dovecot** (for receiving)
» **Squirrelmail** (for webmail access).
Here i have used mail.krizna.com for hostname and krizna.com for Domain . please replace with your domain .

Setup mail server on ubuntu 14.04

» Installing and configuring postfix

» Installing and configuring dovecot

» Installing and configuring squirrelmail

» Installing and configuring postfix

Step 1 » Assign static IP and hostname and add a host entry for the host name .
Assign hostname in **/etc/hostname**
`mail.krizna.com`Add a hostentry in **/etc/hosts**

```
192.168.1.10 mail.krizna.com
```

Step 2 » Update the repositories.

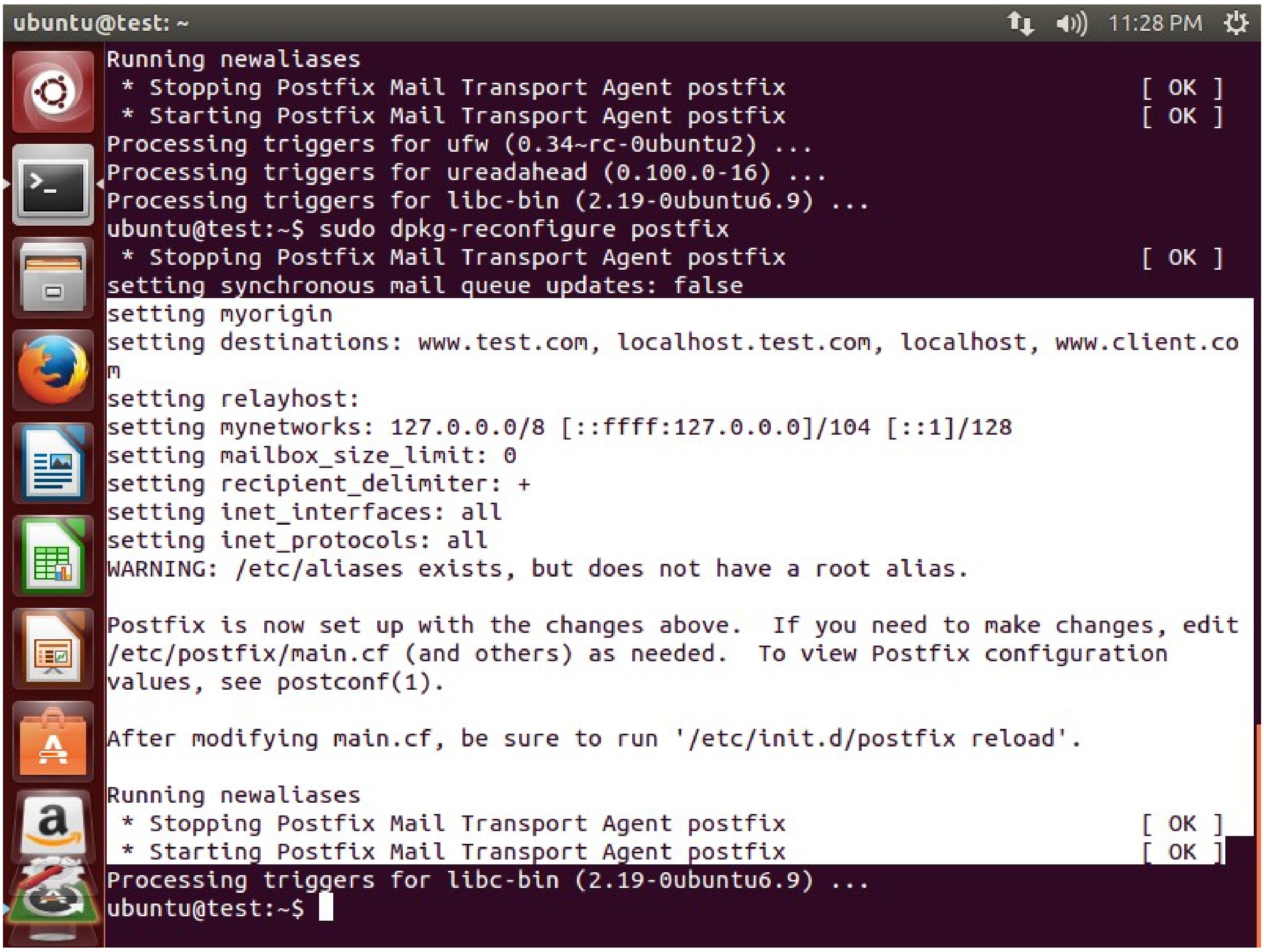
```
krizna@mail:~$ sudo apt-get update
```

Step 3 » Install postfix and dependencies . Press enter for all prompted questions during installation. we will do that in the next step.

```
krizna@mail:~$ sudo apt-get install postfix
```

Step 4 » After installation issue the below command to configure postfix.`krizna@mail:~$ sudo dpkg-reconfigure postfix`Now you will be prompted for set of details . choose the following values and replace krizna.com with your domain name.

1. Internet Site : type of mail server (local,global etc...)
2. krizna.com : FQDN
3. krizna : user insted of root
4. krizna.com, localhost.localdomain, localhost : destination
5. No : sync server for reduce loss of data
6. 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.0.0/24 : local networks
7. 0 : prevent runaway error
8. + : local address extension
9. all : both ipv4 & ipv6 protocol used



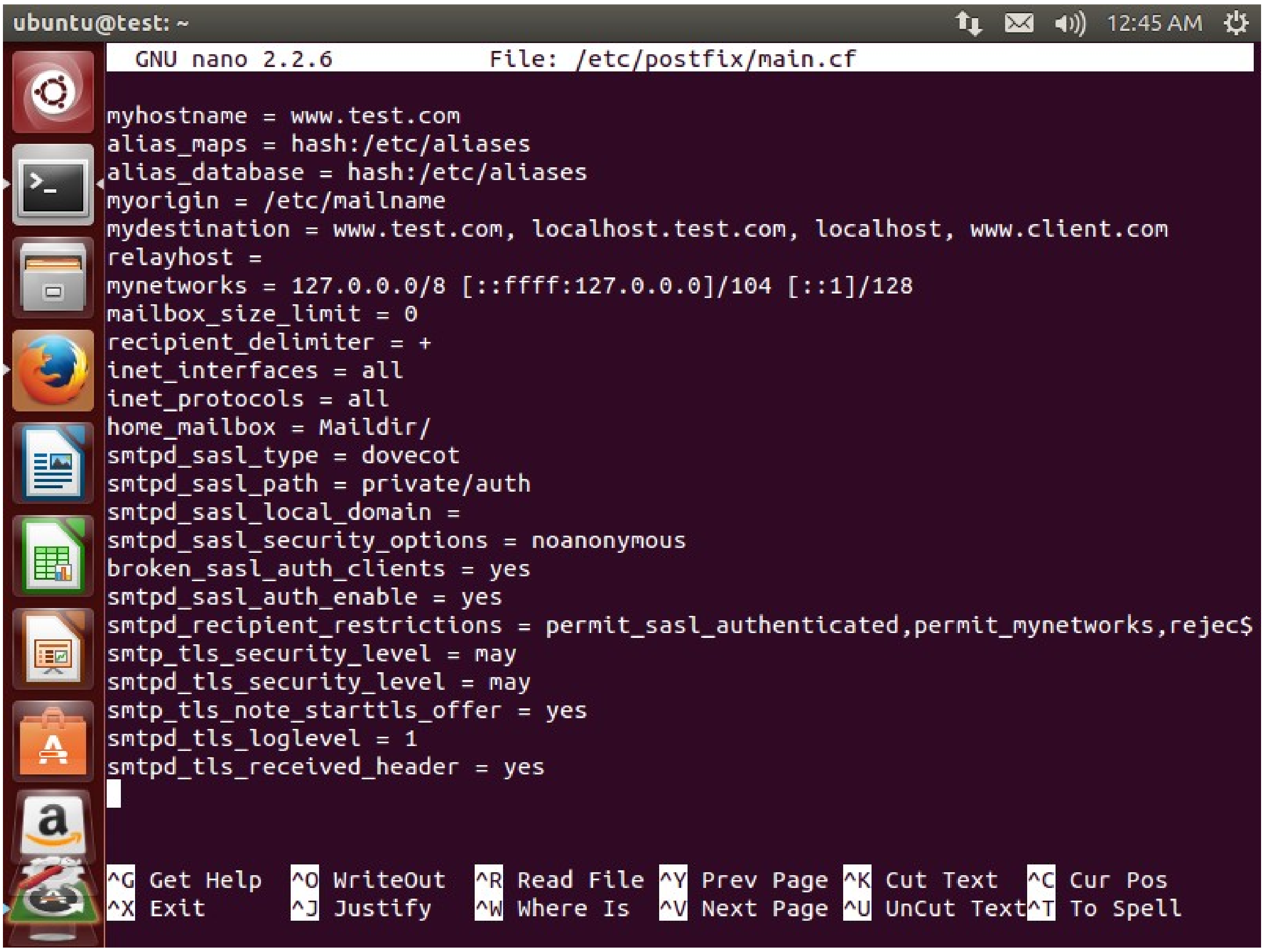
```
Running newaliases
 * Stopping Postfix Mail Transport Agent postfix                        [ OK ]
 * Starting Postfix Mail Transport Agent postfix                        [ OK ]
Processing triggers for ufw (0.34~rc-0ubuntu2) ...
Processing triggers for ureadahead (0.100.0-16) ...
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...
ubuntu@test:~$ sudo dpkg-reconfigure postfix
 * Stopping Postfix Mail Transport Agent postfix                        [ OK ]
setting synchronous mail queue updates: false
setting myorigin
setting destinations: www.test.com, localhost.test.com, localhost, www.client.co
m
setting relayhost:
setting mynetworks: 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
setting mailbox_size_limit: 0
setting recipient_delimiter: +
setting inet_interfaces: all
setting inet_protocols: all
WARNING: /etc/aliases exists, but does not have a root alias.

Postfix is now set up with the changes above.  If you need to make changes, edit
/etc/postfix/main.cf (and others) as needed.  To view Postfix configuration
values, see postconf(1).

After modifying main.cf, be sure to run '/etc/init.d/postfix reload'.

Running newaliases
 * Stopping Postfix Mail Transport Agent postfix                        [ OK ]
 * Starting Postfix Mail Transport Agent postfix                        [ OK ]
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...
ubuntu@test:~$
```

eg: fig. Config. File Make by akhil

Step 5 » Now configure Postfix for SMTP-AUTH using Dovecot SASL by adding the below lines to postfix config file **/etc/postfix/main.cf** .

```
home_mailbox = Maildir/
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_local_domain =
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_sasl_auth_enable = yes
```

```
smtpd_recipient_restrictions =
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destinat
ion
smtp_tls_security_level = may
smtpd_tls_security_level = may
smtp_tls_note_starttls_offer = yes
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
```



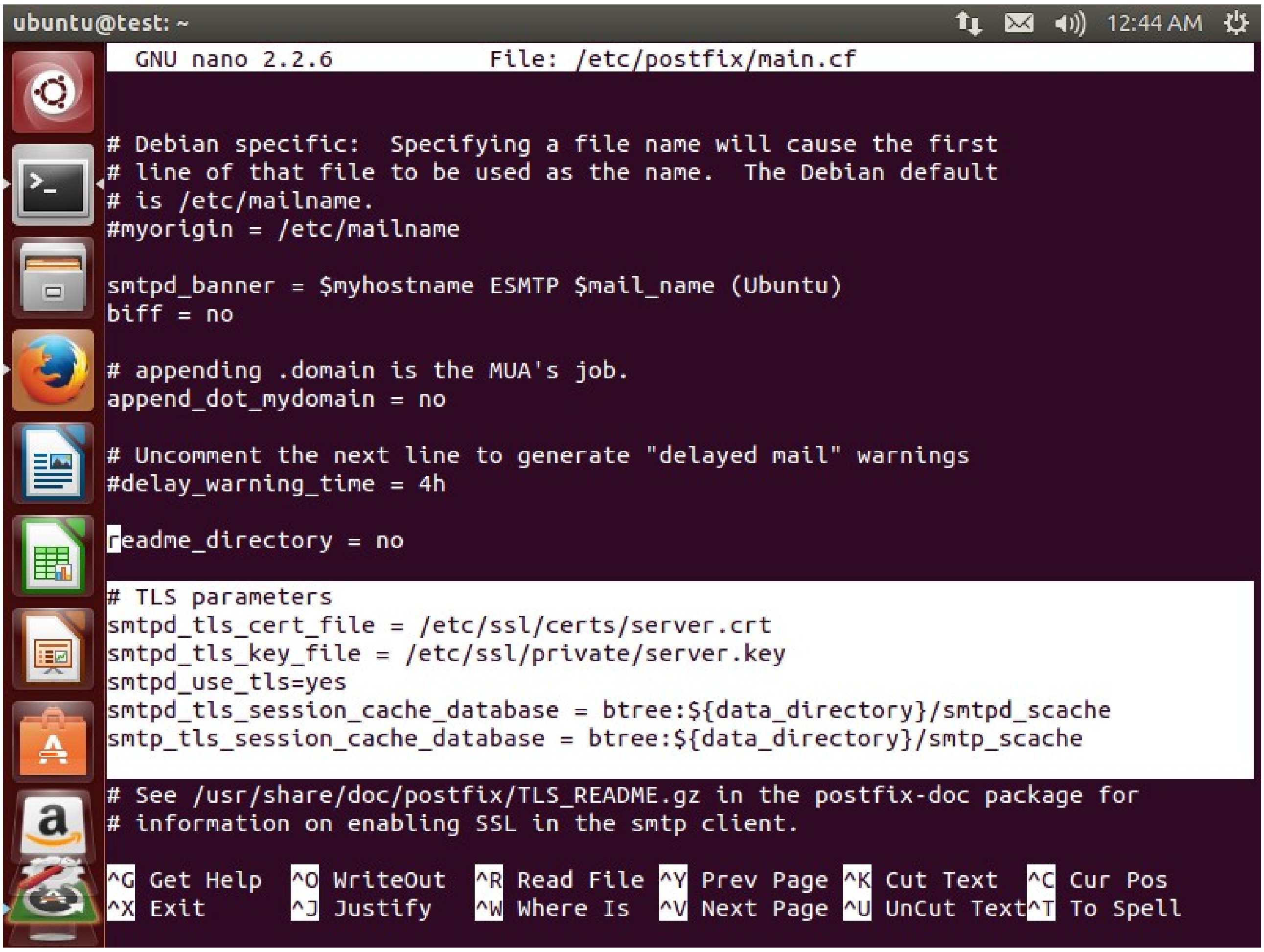
```
myhostname = www.test.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = www.test.com, localhost.test.com, localhost, www.client.com
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
home_mailbox = Maildir/
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_local_domain =
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_sasl_auth_enable = yes
smtpd_recipient_restrictions = permit_sasl_authenticated,permit_mynetworks,rejec$
smtp_tls_security_level = may
smtpd_tls_security_level = may
smtp_tls_note_starttls_offer = yes
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
```

Step 6 » Now generate a digital certificate for tls. Issue the commands one by one and provide details as per your domain.

```
krizna@mail:~$ openssl genrsa -des3 -out server.key 2048
krizna@mail:~$ openssl rsa -in server.key -out server.key.insecure
krizna@mail:~$ mv server.key server.key.secure
krizna@mail:~$ mv server.key.insecure server.key
krizna@mail:~$ openssl req -new -key server.key -out server.csr
krizna@mail:~$ openssl x509 -req -days 365 -in server.csr -signkey
server.key -out server.crt
krizna@mail:~$ sudo cp server.crt /etc/ssl/certs
krizna@mail:~$ sudo cp server.key /etc/ssl/private
```

Step 7 » Now configure certificate path

```
krizna@mail:~$ sudo postconf -e 'smtpd_tls_key_file =
/etc/ssl/private/server.key'
krizna@mail:~$ sudo postconf -e 'smtpd_tls_cert_file =
/etc/ssl/certs/server.crt'
```



```
  GNU nano 2.2.6            File: /etc/postfix/main.cf


# Debian specific:  Specifying a file name will cause the first
# line of that file to be used as the name.  The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

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

readme_directory = no

# TLS parameters
smtpd_tls_cert_file = /etc/ssl/certs/server.crt
smtpd_tls_key_file = /etc/ssl/private/server.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```

Step 8 » Open **/etc/postfix/master.cf** file and uncomment below lines to enable smtps (465) and submission (587) .

submission inet n - - - - smtpd -o syslog_name=postfix/submission -o

smtpd_tls_security_level=encrypt -o smtpd_sasl_auth_enable=yes -o

smtpd_relay_restrictions=permit_sasl_authenticated,reject -o

milter_macro_daemon_name=ORIGINATING smtps inet n - n - - smtpd -o

syslog_name=postfix/smtps -o smtpd_tls_wrappermode=yes -o smtpd_sasl_auth_enable=yes -o

smtpd_relay_restrictions=permit_sasl_authenticated,reject -o

milter_macro_daemon_name=ORIGINATING

```
submission inet n  -   -   -   -  smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
Smtps   inet  n   -   n   -    -    smtpd
  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
```

Step 9 » Now install Dovecot SASL by typing the below command.
`krizna@mail:~$ sudo apt-get install dovecot-common`Issue the following values for the prompts during installation.
1. yes
2. mail.krizna.com

Step 10 » Make changes to the files as follows.

Open **/etc/dovecot/conf.d/10-master.conf** file and find ***# Postfix smtp-auth*** line (line no:95) and add the below lines .

```
# Postfix smtp-auth
unix_listener /var/spool/postfix/private/auth

{
mode = 0660
user = postfix
group = postfix
}
```

Open **/etc/dovecot/conf.d/10-auth.conf** file and find (line no:100)
`auth_mechanisms = plain` and replace `auth_mechanisms = plain login`

Step 11 » Restart postfix and dovecot services

```
krizna@mail:~$ sudo service postfix restart
krizna@mail:~$ sudo service dovecot restart
```

Step 12 » Now test SMTP-AUTH and smtp/pop3 port access .
Type the below command and should get below response.

```
krizna@mail:~$ telnet mail.krizna.com smtp
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mail.kriznaa.com ESMTP Postfix (Ubuntu)
```

now type **ehlo mail.krizna.com** and should get below response , please make sure you get those bolded lines .

```
ehlo mail.krizna.com
250-mail.krizna.com
--------
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
---------
```

`250 DSN` and try the same with port 587 (telnet mail.krizna.com 587).
Postfix configuration is over, continue for dovecot installation.

» Installing and configuring dovecot

Step 12 » Install dovecot using the below command

```
krizna@mail:~$ sudo apt-get install dovecot-imapd dovecot-pop3d
```

Step 13 » Now configure mailbox. Open **/etc/dovecot/conf.d/10-mail.conf** file and find (Line no:30)

`mail_location = mbox:~/mail:INBOX=/var/mail/%u`Replace with

```
mail_location = maildir:~/Maildir
```

Step 14 » Now change pop3_uidl_format . Open **/etc/dovecot/conf.d/20-pop3.conf** file and find and uncomment the below line (Line no : 50)

```
pop3_uidl_format = %08Xu%08Xv
```

Step 15 » Now enable SSL . Open **/etc/dovecot/conf.d/10-ssl.conf** file and find and uncomment the below line (Line no : 6)

```
ssl = yes
```

Step 16 »Restart dovecot service.

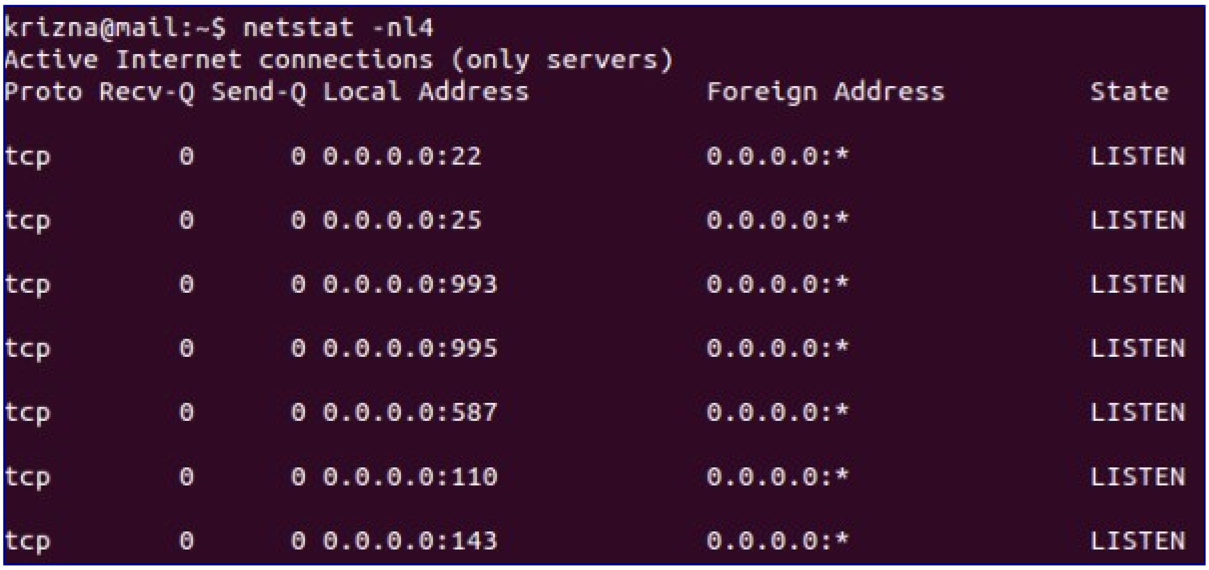
```
krizna@mail:~$ sudo service dovecot restart
```

Step 17 » Now test pop3 and imap port access using the telnet command.

```
krizna@mail:~$ telnet mail.krizna.com 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK Dovecot (Ubuntu) ready.
```

Repeat the same for 995,993,143 ports.
OR check for listening ports using netstat command .

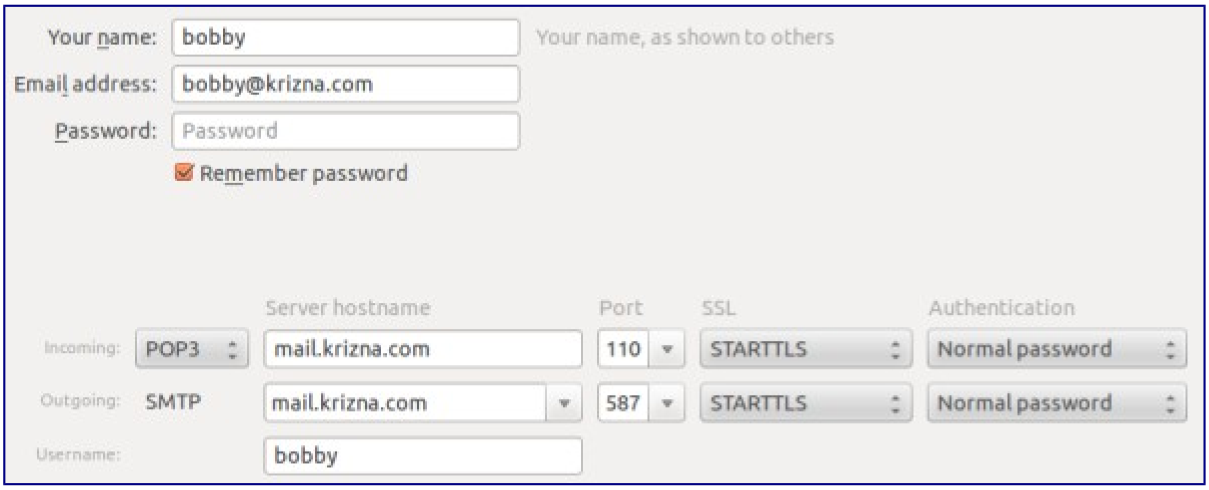
`krizna@mail:~$ netstat -nl4` you should get the result like below.



```
krizna@mail:~$ netstat -nl4
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State

tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN

tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN

tcp        0      0 0.0.0.0:993             0.0.0.0:*               LISTEN

tcp        0      0 0.0.0.0:995             0.0.0.0:*               LISTEN

tcp        0      0 0.0.0.0:587             0.0.0.0:*               LISTEN

tcp        0      0 0.0.0.0:110             0.0.0.0:*               LISTEN

tcp        0      0 0.0.0.0:143             0.0.0.0:*               LISTEN
```

Step 18 » Create some users and check using mail clients like thunderbird or outlook

```
krizna@mail:~$ sudo useradd -m bobby -s /sbin/nologin
krizna@mail:~$ sudo passwd bobby
```



Your name: bobby — Your name, as shown to others

Email address: bobby@krizna.com

Password: Password

☑ Remember password

		Server hostname	Port	SSL	Authentication
Incoming:	POP3	mail.krizna.com	110	STARTTLS	Normal password
Outgoing:	SMTP	mail.krizna.com	587	STARTTLS	Normal password
Username:		bobby			

Now mail server is ready, you can send and receive mail using the server. Continue for squirrelmail ..

» Installing and configuring squirrelmail

Step 19 » Install squirrelmail using the below command. This will install apache and PHP packages.

```
krizna@mail:~$ sudo apt-get install squirrelmail
```

Step 20 » Configure squirrelmail

```
krizna@mail:~$ sudo squirrelmail-configure
```

Everything is pre-configured , we just need to change Organization name .
» Press 1 (Organization Preferences) » again press 1 (Organization Name) » Organization Name » Press S » Press Q to quit

Step 19 » Now configure apache to enable squirrelmail .

```
krizna@mail:~$ sudo cp /etc/squirrelmail/apache.conf
/etc/apache2/sites-available/squirrelmail.conf
krizna@mail:~$ sudo a2ensite squirrelmail
```

Step 20 » Restart Apache service

```
krizna@mail:~$ sudo service apache2 restart
```

Step 21 » Now open **http://serverIP/squirrelmail** in your browser and login using username (bobby) . you can send and receive mail using squirrelmail.
Okay .. its over now ..

to send message in command line use,

```
echo "This is the body of the email" | mail -s "subject" user@example.com
```

IMAP uses port 143 ,
but SSL/TLS encrypted IMAP uses port **993** .

POP uses port **110** ,
but SSL/TLS encrypted POP uses port 995 .

SMTP uses port **25** ,
but SSL/TLS encrypted SMTP uses port **465** .