EMAIL SERVER USING POSTFIX AND DOVECOT PROJECT WRITE-UP

JARED QUAH FOO ZHUN

Table of Contents

*Commands are **bolded**
Parameters & directories are in *italic*
Outputs are <u>underlined</u>*

1.0 Installing Postfix

Before installation, the Rocky Linux needs to be updated to make sure the system is up to date.

```
[jaredsna@jaredserver ~]$ sudo dnf update -y
[sudo] password for jaredsna:
Rocky Linux 9 - BaseOS                          1.3 kB/s | 4.1 kB     00:03
Rocky Linux 9 - AppStream                       2.9 kB/s | 4.5 kB     00:01
Rocky Linux 9 - Extras                          4.5 kB/s | 2.9 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[jaredsna@jaredserver ~]$ S
```

Then postfix can be installed via the command; **sudo dnf install postfix -y**

```
[jaredsna@jaredserver ~]$ sudo dnf install postfix -y
Last metadata expiration check: 0:01:59 ago on Tue 29 Apr 2025 02:01:26 PM.
Dependencies resolved.
================================================================================
 Package         Architecture   Version                 Repository         Size
================================================================================
Installing:
 postfix         x86_64         2:3.5.25-1.el9          appstream         1.5 M

Transaction Summary
================================================================================
Install  1 Package

Total download size: 1.5 M
Installed size: 4.4 M
Downloading Packages:
postfix-3.5.25-1.el9.x86_64.rpm                 3.0 MB/s | 1.5 MB     00:00
--------------------------------------------------------------------------------
Total                                           748 kB/s | 1.5 MB     00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Running scriptlet: postfix-2:3.5.25-1.el9.x86_64                          1/1
  Installing       : postfix-2:3.5.25-1.el9.x86_64                          1/1
  Running scriptlet: postfix-2:3.5.25-1.el9.x86_64                          1/1
  Verifying        : postfix-2:3.5.25-1.el9.x86_64                          1/1

Installed:
  postfix-2:3.5.25-1.el9.x86_64

Complete!
[jaredsna@jaredserver ~]$
```

2.0 Configuring Postfix

We access Postfix's main configuration file by typing; **sudo nano /etc/postfix/main.cf** and then enter our password to access the file.

```
  GNU nano 5.6.1                                                    /etc/post
# Global Postfix configuration file. This file lists only a subset
# of all parameters. For the syntax, and for a complete parameter
# list, see the postconf(5) manual page (command: "man 5 postconf").
#
# For common configuration examples, see BASIC_CONFIGURATION_README
# and STANDARD_CONFIGURATION_README. To find these documents, use
# the command "postconf html_directory readme_directory", or go to
# http://www.postfix.org/BASIC_CONFIGURATION_README.html etc.
#
# For best results, change no more than 2-3 parameters at a time,
# and test if Postfix still works after every change.

# COMPATIBILITY
#
# The compatibility_level determines what default settings Postfix
# will use for main.cf and master.cf settings. These defaults will
# change over time.
#
# To avoid breaking things, Postfix will use backwards-compatible
# default settings and log where it uses those old backwards-compatible
# default settings, until the system administrator has determined
# if any backwards-compatible default settings need to be made
# permanent in main.cf or master.cf.
#
# When this review is complete, update the compatibility_level setting
# below as recommended in the RELEASE_NOTES file.
#
# The level below is what should be used with new (not upgrade) installs.
#
compatibility_level = 2

# SOFT BOUNCE
#
# The soft_bounce parameter provides a limited safety net for
# testing.  When soft_bounce is enabled, mail will remain queued that
# would otherwise bounce. This parameter disables locally-generated
# bounces, and prevents the SMTP server from rejecting mail permanently
# (by changing 5xx replies into 4xx replies). However, soft_bounce
# is no cure for address rewriting mistakes or mail routing mistakes.
#
#soft_bounce = no
```

We then scroll down to verify, uncomment and modify some key perimeters.

1. *Inet_interfaces* to be set to *"all"*

```
inet_interfaces = all
```

2. *Mynetworks* must be set to *192.168.200.0/24* to allow devices on the subnet to relay mail.

```
mynetworks = 192.168.200.0/24, 127.0.0.0/8
#mynetworks = $config_directory/mynetworks
#mynetworks = hash:/etc/postfix/network_table
```

3. We uncomment and change *myhostname* to our host name which is *jaredserver.bungkus.org*

```
myhostname = jaredserver.bungkus.org
#myhostname = virtual.domain.tld
```

4. We do the same for *mydomain* but we enter the domain name instead. In this case it was *bungkus.org*

```
mydomain = bungkus.org
```

5. We uncomment the line *myorigin = $myhostname*

```
myorigin = $myhostname
```

6. We do the same for home_mailbox as well.

```
home_mailbox = Maildir/
```

7. The directories for the certificate and key files must be uncommented as well.

```
smtpd_tls_cert_file = /etc/pki/tls/certs/postfix.pem

# The full pathname of a file with the Postfix SMTP ser
# in PEM format. The private key must be accessible wit
# i.e. it must not be encrypted.
#
smtpd_tls_key_file = /etc/pki/tls/private/postfix.key
```

- *smtpd_tls_cert_file = /etc/pki/tls/certs/postfix.pem*
- *smptpd_tls_key_file = /etc/pki/tls/private/postfix.key*

8. *Smtp_tls_security_level = may*, should be uncommented, this is to ensure postfix uses TLS.

```
smtp_tls_security_level = may
```

9. Restart Postfix.

```
[jaredsna@jaredserver ~]$ sudo systemctl restart postfix
[sudo] password for jaredsna:
[jaredsna@jaredserver ~]$
```

10. We then view the status of Postfix to ensure it is active and enabled.

```
● postfix.service - Postfix Mail Transport Agent
     Loaded: loaded (/usr/lib/systemd/system/postfix.service; enabled; preset: >
     Active: active (running) since Thu 2025-05-01 14:24:23 +08; 18min ago
    Process: 2902 ExecStartPre=/usr/sbin/restorecon -R /var/spool/postfix/pid (>
    Process: 2905 ExecStartPre=/usr/libexec/postfix/aliasesdb (code=exited, sta>
    Process: 2907 ExecStartPre=/usr/libexec/postfix/chroot-update (code=exited,>
    Process: 2908 ExecStart=/usr/sbin/postfix start (code=exited, status=0/SUCC>
   Main PID: 2976 (master)
      Tasks: 4 (limit: 34970)
     Memory: 6.4M
        CPU: 413ms
     CGroup: /system.slice/postfix.service
             ├─2976 /usr/libexec/postfix/master -w
             ├─2977 pickup -l -t unix -u
             ├─2978 qmgr -l -t unix -u
             └─3001 tlsmgr -l -t unix -u
```

3.0 Installing Dovecot

We install Dovecot using

- **sudo dnf install dovecot -y**

```
[jaredsna@jaredserver ~]$ sudo dnf install dovecot -y
Last metadata expiration check: 0:10:58 ago on Tue 29 Apr 2025 02:42:15 PM.
Package dovecot-1:2.3.16-14.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[jaredsna@jaredserver ~]$
```

4.0 Configuring Dovecot

1. We access Dovecot's configuration file with

- **sudo nano /etc/dovecot/dovecot.conf**

```
  GNU nano 5.6.1                                        /etc/dovecot/dovecot.conf
## Dovecot configuration file

# If you're in a hurry, see http://wiki2.dovecot.org/QuickConfiguration

# "doveconf -n" command gives a clean output of the changed settings. Use it
# instead of copy&pasting files when posting to the Dovecot mailing list.

# '#' character and everything after it is treated as comments. Extra spaces
# and tabs are ignored. If you want to use either of these explicitly, put the
# value inside quotes, eg.: key = "# char and trailing whitespace  "

# Most (but not all) settings can be overridden by different protocols and/or
# source/destination IPs by placing the settings inside sections, for example:
# protocol imap { }, local 127.0.0.1 { }, remote 10.0.0.0/8 { }
```

2. Uncommenting protocols so Dovecot can use the imap/pop3 protocol.

```
# Protocols we want to be serving.
protocols = imap pop3 lmtp submission
```

3. We uncomment and verify that *listen* = * is present, this will allow postfix to listen in all ipv4 interfaces, we can delete the comma (,) and the colons (: :) as we will not be using ipv6. Save and exit the file after

```
listen = *
```

4. We now enter the mail configuration file**. sudo nano /etc/dovecot/conf.d/10-mail.conf**

```
  GNU nano 5.6.1                                        /etc/dovecot/conf.d/10-mail.conf
##
## Mailbox locations and namespaces
##

# Location for users' mailboxes. The default is empty, which means that Dovecot
# tries to find the mailboxes automatically. This won't work if the user
# doesn't yet have any mail, so you should explicitly tell Dovecot the full
# location.
#
# If you're using mbox, giving a path to the INBOX file (eg. /var/mail/%u)
# isn't enough. You'll also need to tell Dovecot where the other mailboxes are
# kept. This is called the "root mail directory", and it must be the first
# path given in the mail_location setting.
```

5. Uncommenting the *mail_location* will allow the mail to be redirected accordingly.

```
  mail_location = maildir:~/Maildir
#   mail_location = mbox:~/mail:INBOX=/var/mail/%u
#   mail_location = mbox:/var/mail/%d/%1n/%n:INDEX=/var/indexes/%d/%1n/%n
```

6. We save and exit the mail configuration file, then open Dovecot's authority file and add login to *auth_mechanisms.*

```
# Space separated list of wanted authentication mechanisms:
#   plain login digest-md5 cram-md5 ntlm rpa apop anonymous gssapi otp
#   gss-spnego
# NOTE: See also disable_plaintext_auth setting.
auth_mechanisms = plain login
```

7. We enter the master configuration file and scroll to the service authentication section.

```
service auth {
  # auth_socket_path points to this userdb socket by default. It's typically
  # used by dovecot-lda, doveadm, possibly imap process, etc. Users that have
  # full permissions to this socket are able to get a list of all usernames and
  # get the results of everyone's userdb lookups.
  #
  # The default 0666 mode allows anyone to connect to the socket, but the
  # userdb lookups will succeed only if the userdb returns an "uid" field that
  # matches the caller process's UID. Also if caller's uid or gid matches the
  # socket's uid or gid the lookup succeeds. Anything else causes a failure.
  #
  # To give the caller full permissions to lookup all users, set the mode to
  # something else than 0666 and Dovecot lets the kernel enforce the
  # permissions (e.g. 0777 allows everyone full permissions).
  unix_listener auth-userdb {
    #mode = 0666
    #user =
    #group =
  }

  # Postfix smtp-auth
  #unix_listener /var/spool/postfix/private/auth {
  #  mode = 0666
  #}

  # Auth process is run as this user.
  #user = $default_internal_user
}
```

8. Then change some configurations, particularly changing the mode to 0660 and adding group = postfix and uncommenting the bracket to close it to avoid errors.

```
service auth {
  # auth_socket_path points to this userdb socket by default. It's typically
  # used by dovecot-lda, doveadm, possibly imap process, etc. Users that have
  # full permissions to this socket are able to get a list of all usernames and
  # get the results of everyone's userdb lookups.
  #
  # The default 0666 mode allows anyone to connect to the socket, but the
  # userdb lookups will succeed only if the userdb returns an "uid" field that
  # matches the caller process's UID. Also if caller's uid or gid matches the
  # socket's uid or gid the lookup succeeds. Anything else causes a failure.
  #
  # To give the caller full permissions to lookup all users, set the mode to
  # something else than 0666 and Dovecot lets the kernel enforce the
  # permissions (e.g. 0777 allows everyone full permissions).
  unix_listener auth-userdb {
    #mode = 0666
    #user =
    #group =
  }

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

  # Auth process is run as this user.
  #user = $default_internal_user
}
```

9. Change ssl to yes and ensure ssl_cert and ssl_key directories are correct.

```
jaredsna@jaredserver:~ — sudo nano /etc/dovecot/conf.d/10-s...

  GNU nano 5.6.1          /etc/dovecot/conf.d/10-ssl.conf
# disable plain pop3 and imap, allowed are only pop3+TLS, pop3s, imap+TLS and i>
# plain imap and pop3 are still allowed for local connections
ssl = yes

# PEM encoded X.509 SSL/TLS certificate and private key. They're opened before
# dropping root privileges, so keep the key file unreadable by anyone but
# root. Included doc/mkcert.sh can be used to easily generate self-signed
# certificate, just make sure to update the domains in dovecot-openssl.cnf
ssl_cert = </etc/pki/tls/certs/postfix.pem
ssl_key = </etc/pki/tls/private/postfix.key
```

10. We then request a digital certificate that will be placed into folders for the certificate and key.

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:JP
State or Province Name (full name) []:Hiroshima
Locality Name (eg, city) [Default City]:Hiroshima
Organization Name (eg, company) [Default Company Ltd]:Bungkus.org
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:jaredserver.bungkus.org
Email Address []:jared@bungkus.org
```

11. Now we view the status of Dovecot in the same way as postfix. Again, we want to ensure that Dovecot is running and enabled.

```
jaredsna@jaredserver:~ — sudo systemctl status dovecot

● dovecot.service - Dovecot IMAP/POP3 email server
     Loaded: loaded (/usr/lib/systemd/system/dovecot.service; enabled; preset: >
     Active: active (running) since Thu 2025-05-01 14:13:03 +08; 33min ago
       Docs: man:dovecot(1)
             https://doc.dovecot.org/
    Process: 1148 ExecStartPre=/usr/libexec/dovecot/prestartscript (code=exited>
   Main PID: 1189 (dovecot)
     Status: "v2.3.16 (7e2e900c1a) running"
      Tasks: 11 (limit: 34970)
     Memory: 20.3M
        CPU: 597ms
     CGroup: /system.slice/dovecot.service
```

5.0 TLS/SSL Implementation

Restart Postfix and Dovecot

- sudo systemctl restart postfix
- sudo systemctl restart dovecot

Then we entered the following commands to configure the firewall accordingly, specifically to allow traffic to pass through port 993 and 465 for our email communication to be successful.

- sudo firewall-cmd --add-service=smtp --permanent

This command adds port 25, which enables us to receive mail from other mail services.

- sudo firewall-cmd --add-service=smtps --permanent

This opens port 465 for secure client mail submission using TLS

- sudo firewall-cmd --add-service=imaps –permanent

This adds port 993, which allows for secure client mail retrieval using IMAP over TLS.

```
[jaredsna@jaredserver ~]$ sudo firewall-cmd --add-service=imaps --permanent
Warning: ALREADY_ENABLED: imaps
success
[jaredsna@jaredserver ~]$ sudo firewall-cmd --add-service=smtp --permanent
success
[jaredsna@jaredserver ~]$ sudo firewall-cmd --add-service=smtps --permanent
Warning: ALREADY_ENABLED: smtps
success
```

6.0 Restarting Services

We restart Postfix, Dovecot and the firewall to apply the changes made. All of them restart successfully without any errors which indicate no fatal mistakes in the configuration.

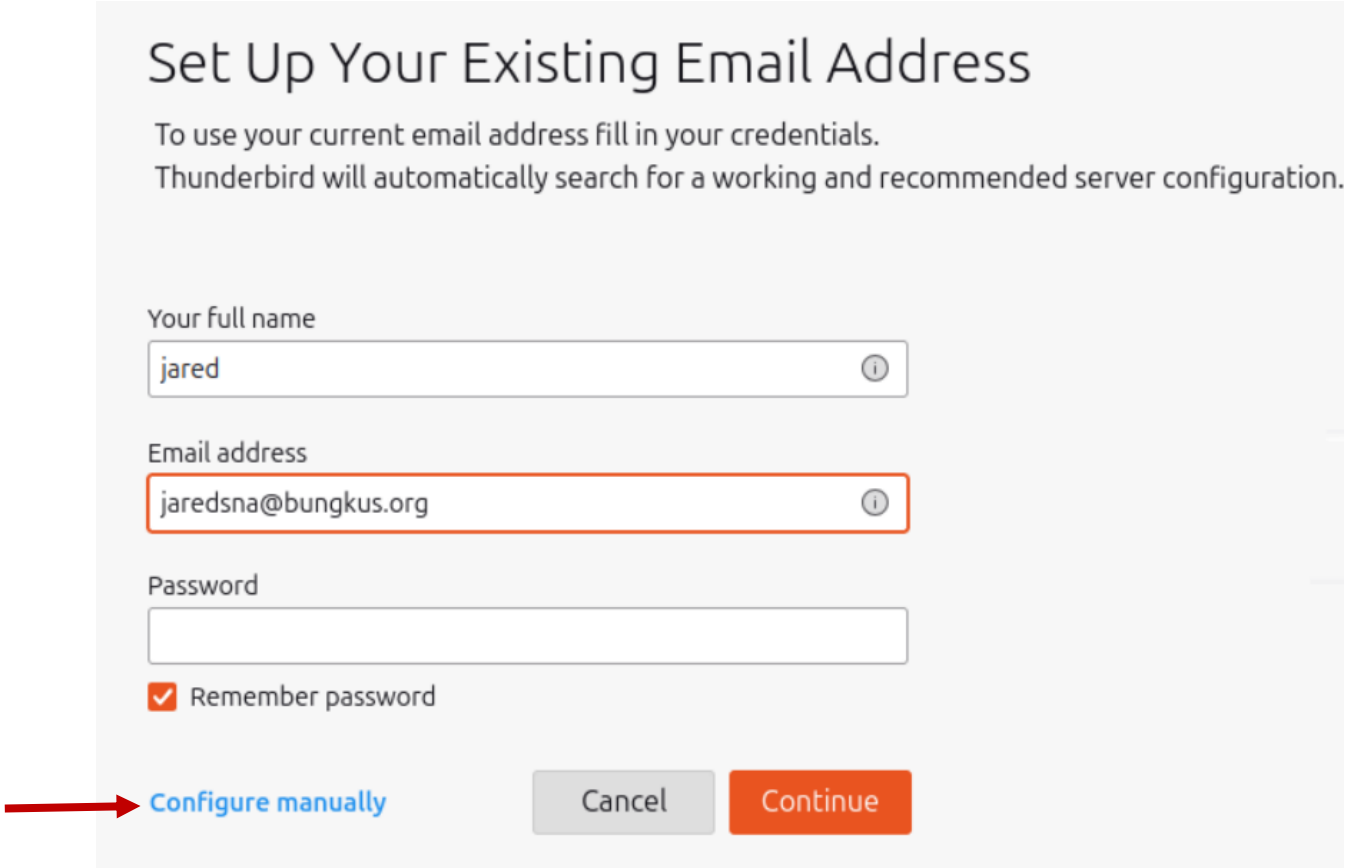
```
[jaredsna@jaredserver ~]$ sudo systemctl restart postfix
[jaredsna@jaredserver ~]$ sudo systemctl restart dovecot
^[[A^[[A^[[A[jaredsna@jaredserver ~]$ sudo firewall-cmd --reload
success
```

7.0 Mozilla Thunderbird Set-up

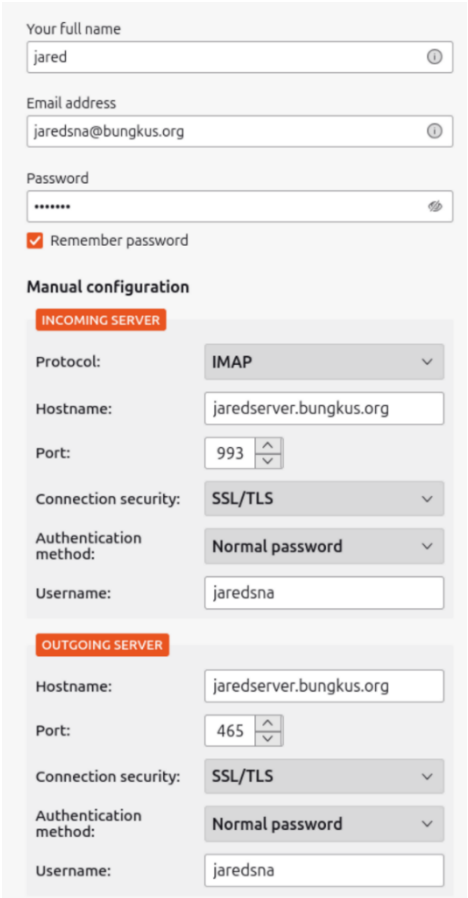
1. We install Thunderbird using the command sudo apt install thunderbird

```
snajq@jaredclient:~$ sudo apt install thunderbird
[sudo] password for snajq:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
thunderbird is already the newest version (2:1snap1-0ubuntu3).
The following packages were automatically installed and are no longer required:
  libllvm17t64 python3-netifaces
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
snajq@jaredclient:~$
```

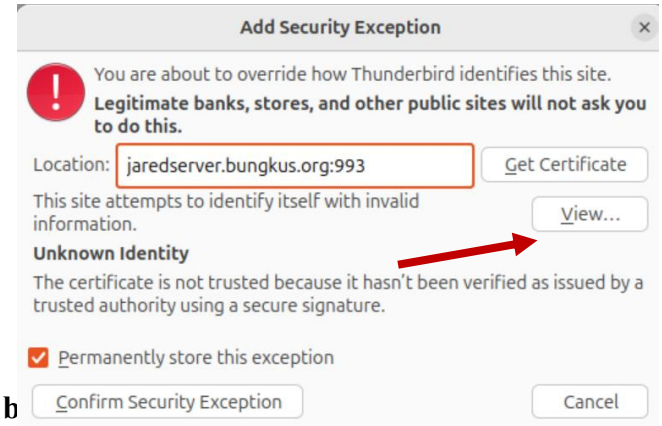
2. Then open Thunderbird and key in our full name and email address until a configure manually option becomes available. We then click on it.



3. We enter the details according to our configurations earlier.



- Protocol must be set to *IMAP*
- Hostname becomes our server's hostname *jaredserver.bungkus.org.*
- Port is selected for *993* for incoming server and *465* for outgoing server.
- Connection security is automatically changed to *SSL/TLS* when the port is selected to *993* or *465.*
- Authentication method is set to *Normal password.*
- The username must be the same as the Linux username, *jaredsna.*
- Click on *Done* after.
- An add security exception dialog will appear, this is because the certificate we have is self-signed and is trusted by Thunderbird's built-in list of Certificate Authorities. Hence, we need to confirm and allow this exception so thunderbird can trust this certificate.

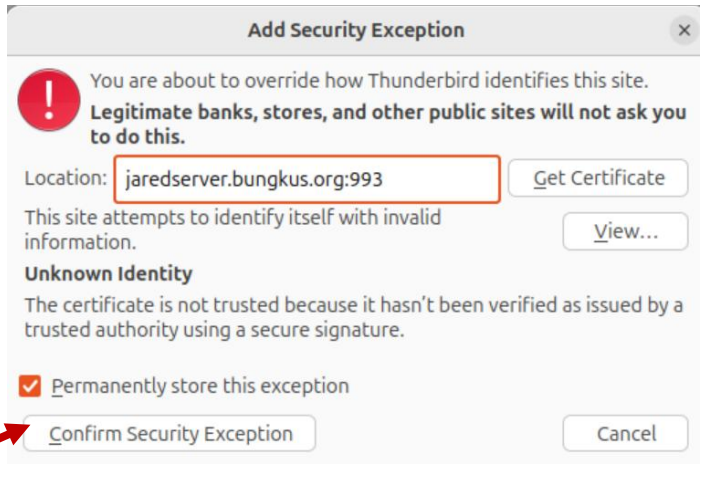


- Ensure the *permanently store this exception* option is ticked.
- Click view to check the certificate's details.

*Commands are b
Parameters & directories are in *italic*
Outputs are <u>underlined</u>*

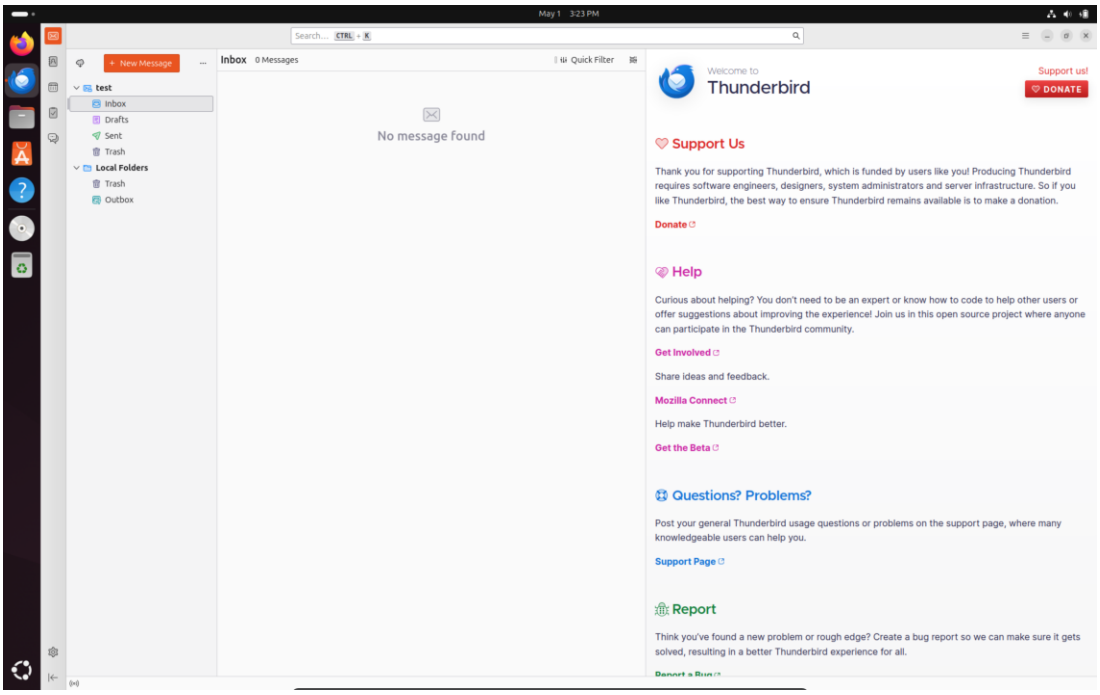
- The certificates information will be shown like this.

- Click *Confirm Security Exception*

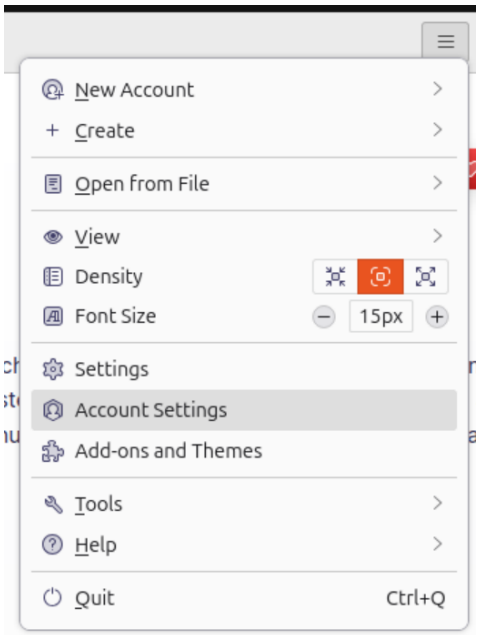


*Commands are **bolded**
Parameters & directories are in *italic*
Outputs are <u>underlined</u>*

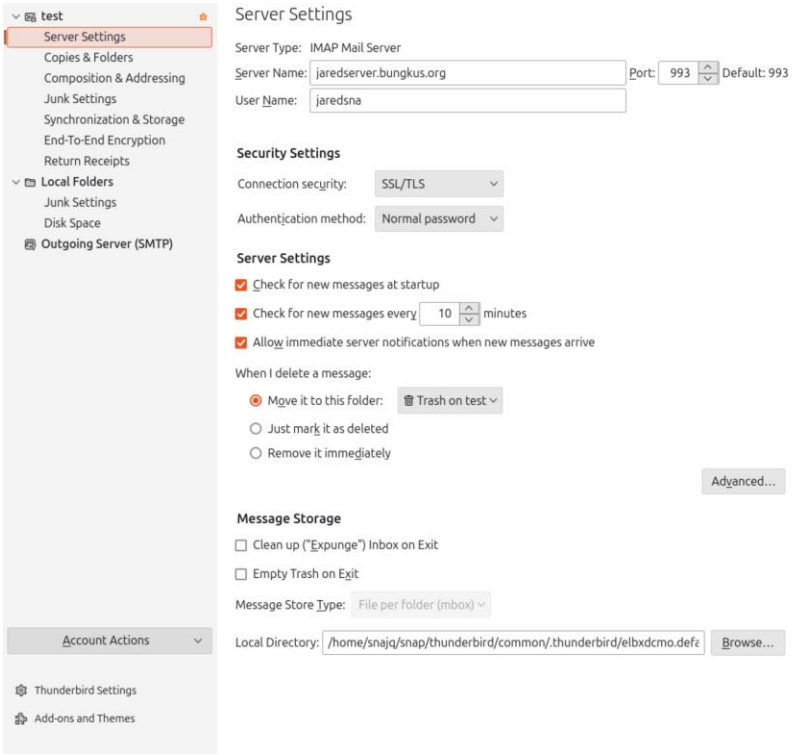
8. We now have successfully logged into Thunderbird.



9. Furthermore, we select the triple line icon on the top right and click on *Account Settings*

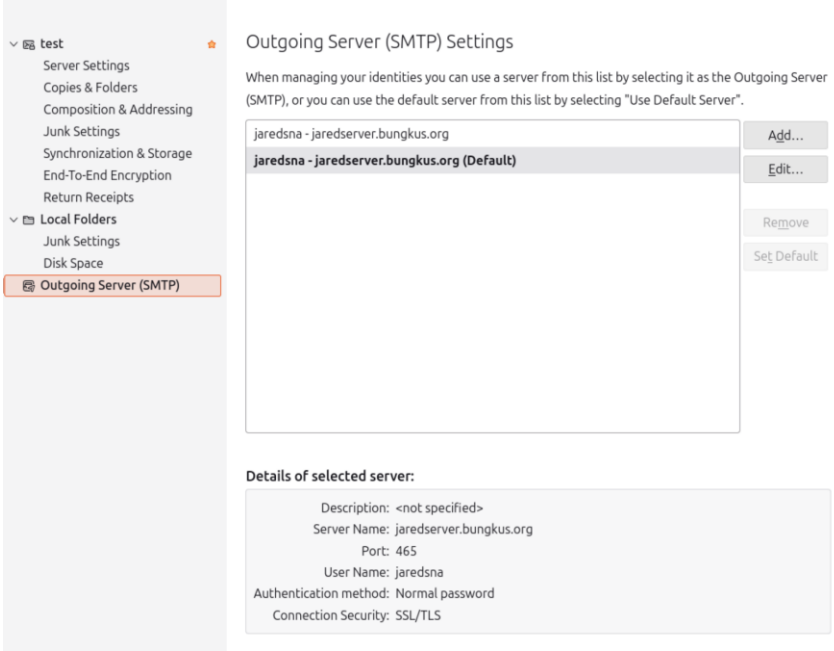


10. Click on server settings and check the configuration to ensure we are running on the correct settings.



*Commands are **bolded**
Parameters & directories are in *italic*
Outputs are <u>underlined</u>*

11. We then click on Outgoing Server (SMTP) option to check our outgoing server settings as well. We make sure the outgoing server is the same hostname as our Rocky server.



8.0 Proof of Successful and Secure Email between Client and Server

It is essential to provide reputable proof that the entire email communication system is functioning correctly and securely. This section presents evidence demonstrating that emails are successfully sent from the client and delivered to the intended mailbox on the server and that the communication channels used for both sending and receiving mail are encrypted using TLS/SSL.

8.1 Verifying Secured Incoming Mail

Firstly, we enter the command to let us view the established TLS connection on port 993, which is for incoming mail. As shown, there was a successful connection to the server jaredserver.bungkus.org (192.168.200.4).

- **openssl s_client -connect jaredserver.bungkus.org:993**

```
jaredsna@jaredserver:~ — openssl s_client -connect jaredserve...
[jaredsna@jaredserver ~]$ openssl s_client -connect jaredserver.bungkus.org:993
Connecting to 192.168.200.4
CONNECTED(00000003)
```

Then, we scroll down further until we can view the SSL session, this confirms the use of TLS and cipher suite, proving the connection is encrypted.

```
jaredsna@jaredserver:~
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: 3BDE44E5E0C140737AA91282858B613B0FB41327BCA01AF1DD20B44EE3779E3B
```

<u>Protocol: TLSv1.3</u>

TLS (Transport Layer Security) version that is being used is version 1.3.

<u>Cipher: TLS_AES_256_GCM_SHA384</u>

- Uses Advanced Encryption Standard with 256-bit key.
- Uses GCM, which is an encryption mode that provides confidentiality and data integrity.
- Uses Secured Hashing Algorithm (SHA) with 384 bits.

8.2 Verifying Secured Outgoing Mail

We enter the same command used to view the TLS connection for port 993, however, this time we change the port to port 465. Which is for outgoing mail. Here, we see yet again a successful connection to the server. We then scroll down again to view the SSL session that states the same protocol and cipher suite being used for the connection. Ensuring both the incoming and outgoing mail connections are secured and encrypted.

- **openssl s_client -connect jaredserver.bungkus.org:465**

```
jaredsna@jaredserver:~ — openssl s_client -connect jaredserve...
[jaredsna@jaredserver ~]$ openssl s_client -connect jaredserver.bungkus.org:465
Connecting to 192.168.200.4
CONNECTED(00000003)
```

```
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: FB20132C8CFC2B9C3132323D0C1DA369F6162DF3773DE5B5D2E82D46C901149F
```

8.3 Digital Certificate Verification

With the same command, we can scroll down to view the server certificate. This was the same certificate that was self-signed and requested earlier. Besides checking that the connection is encrypted, it is also necessary to verify the identity of the server. During the TLS/SSL handshake, the server's digital certificate serves this main purpose. The certificate binds the server's public key to its hostname, serving as a digital identity card.

- **openssl s_client -connect jaredserver.bungkus.org:465**

```
jaredsna@jaredserver:~
Certificate chain
 0 s:C=JP, ST=Hiroshima, L=Hiroshima, O=bungkus, OU=IT, CN=jaredserver.bungkus.org, emailAddress=test@bungkus.com
   i:C=JP, ST=Hiroshima, L=Hiroshima, O=bungkus, OU=IT, CN=jaredserver.bungkus.org, emailAddress=test@bungkus.com
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: Apr 28 08:30:49 2025 GMT; NotAfter: Apr 28 08:30:49 2026 GMT
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIEETCCAvmgAwIBAgIUAOxqDiG4zx4FD5d+EmUWIbFwhxgwDQYJKoZIhvcNAQEL
BQAwgZcxCzAJBgNVBAYTAkpQMRIwEAYDVQQIDAlIaXJvc2hpbWExEjAQBgNVBAcM
CUhpcm9zaGltYTEQMA4GA1UECgwHYnVuZ2t1czELMAkGA1UECwwCSVQxIDAeBgNV
BAMMF2phcmVkc2VydmVyLmJ1bmdrdXMub3JnMR8wHQYJKoZIhvcNAQkBFhB0ZXN0
QGJ1bmdrdXMuY29tMB4XDTI1MDQyODA4MzA0OVoXDTI2MDQyODA4MzA0OVowgZcx
CzAJBgNVBAYTAkpQMRIwEAYDVQQIDAlIaXJvc2hpbWExEjAQBgNVBAcMCUhpcm9z
aGltYTEQMA4GA1UECgwHYnVuZ2t1czELMAkGA1UECwwCSVQxIDAeBgNVBAMMF2ph
cmVkc2VydmVyLmJ1bmdrdXMub3JnMR8wHQYJKoZIhvcNAQkBFhB0ZXN0QGJ1bmdr
dXMuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAkpKmHdel+JLY
WL6TWbDnryMBymmSaiyjaK6hJCqH/+ilmex/7QDl3/wBAQjyAU4np3pySTvaKLav
BKI5kgjH401i57bgmxUbguCitHyCcueO9fxujm9tXzZ84IiW8ehrpa8/Pcs91wQL
q64GuEF9RO4Ce0m3Yyn/KkC+4mfE086dWJbXI4UxCec/qM4NzLe49+3wwlIh0E9o
9GFU7OFmHUP31WSIWcqzucdfjJ/bTFaC9ihDUAu2dCsplUxb/zNtHfVUa5kDy6yw
eIJDZOlsMI+4NcXukImyLORgk/PCC0ArTaH9FEoh5h6rfRyZRHqEFOqPFw5TbDom
uOE5RzDWFQIDAQABo1MwUTAdBgNVHQ4EFgQUFzxBJMvYg9PRzdJerHispL23U3Ew
HwYDVR0jBBgwFoAUFzxBJMvYg9PRzdJerHispL23U3EwDwYDVR0TAQH/BAUwAwEB
/zANBgkqhkiG9w0BAQsFAAOCAQEAHAFZa7ltjHSqMsyTZZWHGgA3jB4UbIzTFyL2
hAGT8yQs1eb4Y2X+x+1DXThSgFAo+uunh18sJJxvlPxdaX73SFiAqea1XBOJ8JLr
0aDdfvIvq2fGolKV7izhTw1OfWeD3BO1dk0Fg/BkeWsLfwOewGglPVxZJMb7+rnY
HQZ8Q17/4sw4yXhuUfNTtCR3cnvOFBaNqZYYWx4Ym4DPAcnYWU1njD6bfXWsXKcU
hMQ+Vur8RPF9lCu+DH9t9nxA6eK3jssMz3vn2CW9RqUM7xqN58cWQQyNW94sSu/X
wfv9r6kCNhUYRfFbJ08KdO9ImKoD7mVcsI4UzX8G+uvqBmjitw==
-----END CERTIFICATE-----
subject=C=JP, ST=Hiroshima, L=Hiroshima, O=bungkus, OU=IT, CN=jaredserver.bungkus.org, emailAddress=test@bungkus.com
issuer=C=JP, ST=Hiroshima, L=Hiroshima, O=bungkus, OU=IT, CN=jaredserver.bungkus.org, emailAddress=test@bungkus.com
---
```

*Commands are **bolded**
Parameters & directories are in *italic*
Outputs are <u>underlined</u>*

9.0 Troubleshooting – SASL Authentication Failure

Setting up a working mail server system was successful and involved different software, most notably Postfix for sending and receiving mail and Dovecot for mailbox access and user authentication. Despite following the initial configuration procedures, several issues arose that caused an unsuccessful mail delivery including the inability to send emails due to authentication failures and the failure to deliver emails to local mailboxes. This section describes the troubleshooting procedure used to identify the fundamental causes of these issues and take corrective action to establish a functional mail server setup.

9.0.1 Cause of Error

The cause of the initial authentication failure was a misconfiguration in Postfix in how it was attempting to use SASL authentication with Dovecot. Specifically, two parameters were incorrectly set:

- *smtpd_sasl_path* pointed to the wrong socket location.
- *smtpd_sasl_type* was set to *cyrus* instead of *dovecot*,

This caused Postfix to look for a local authentication method instead of using the Dovecot authentication socket as intended.

9.0.2 Troubleshooting Steps

1. Viewing Postfix's logs helped to get a better understanding of where in particular the error was coming from. Here, a misconfiguration with the SASL was highlighted in the log below which is why it was outputting <u>no such file or directory</u>.

Command used to view Postfix's logs: **sudo tail -f /var/log/maillog**

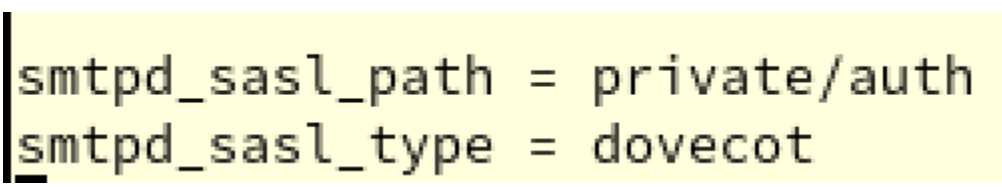
```
Apr 29 16:26:17 jaredserver postfix/smtpd[3231]: warning: unknown[192.168.200.80]: SASL PLAIN authentication failed: generic failure, sasl_username=jaredsna
Apr 29 16:26:17 jaredserver postfix/smtpd[3231]: warning: SASL authentication failure: cannot connect to saslauthd server: No such file or directory
Apr 29 16:26:17 jaredserver postfix/smtpd[3231]: warning: unknown[192.168.200.80]: SASL LOGIN authentication failed: generic failure, sasl_username=jaredsna
Apr 29 16:26:24 jaredserver postfix/smtpd[3231]: warning: SASL authentication failure: cannot connect to saslauthd server: No such file or directory
Apr 29 16:26:24 jaredserver postfix/smtpd[3231]: warning: SASL authentication failure: Password verification failed
Apr 29 16:26:24 jaredserver postfix/smtpd[3231]: warning: unknown[192.168.200.80]: SASL PLAIN authentication failed: generic failure, sasl_username=jaredsna
Apr 29 16:26:24 jaredserver postfix/smtpd[3231]: warning: SASL authentication failure: cannot connect to saslauthd server: No such file or directory
Apr 29 16:26:24 jaredserver postfix/smtpd[3231]: warning: unknown[192.168.200.80]: SASL LOGIN authentication failed: generic failure, sasl_username=jaredsna
Apr 29 16:26:25 jaredserver postfix/smtpd[3231]: lost connection after AUTH from unknown[192.168.200.80]
Apr 29 16:26:25 jaredserver postfix/smtpd[3231]: disconnect from unknown[192.168.200.80] ehlo=1 auth=0/6 commands=1/7
```

*Commands are **bolded**
Parameters & directories are in *italic*
Outputs are <u>underlined</u>*

2. Since the error was stating no such file or directory, it was clear that Postfix could not find/connect to its authentication backend socket.

3. Then, Postfix's main configuration file was accessed to change the path and type of SASL. Making sure it uses the socket provided by Dovecot.

4. **sudo postconf smtpd_sasl_path** was used to view which socket SASL was using. Initially it was using *smtpd* which was wrong, it needed to be changed to *private/auth*.



- *Smtpd_sasl_path* must be changed to "*private/auth*"
- *Smtpd_sasl_type* must be changed to "*dovecot"*

5. Restart Postfix and the error is resolved.

- **sudo systemctl restart postfix**

```
jaredsna@jaredserver:~
[jaredsna@jaredserver ~]$ sudo nano /etc/postfix/main.cf
[sudo] password for jaredsna:
[jaredsna@jaredserver ~]$ sudo systemctl restart postfix
[sudo] password for jaredsna:
[jaredsna@jaredserver ~]$
```

6. Now we can send emails without any authentication failure messages.

jared <jaredsna@bungkus.org> 9:41 PM
test

jared <jaredsna@bungkus.org> 9:40 PM
jhi

9.1 Troubleshooting – Emails Not Appearing in Inbox

Another problem encountered was the emails not updating in the inbox, no matter how many were sent. While sending emails from the client appeared to be successful after resolving the authentication issue, emails addressed to the local user (jaredsna@bungkus.org) were not being delivered to the intended mailbox on the server. This indicated that something was wrong with the local delivery.

9.1.1 Cause of Error

Postfix was not correctly configured to identify email addresses within its own domain (bungkus.org) as destinations for local delivery. Specifically, the *mydestination* parameter in the main configuration file *(/etc/postfix/main.cf)* did not include *$mydomain*, this prevented Postfix from recognizing these addresses as local recipients.

9.1.2 Troubleshooting Steps

1. We enter the command **sudo ls -l ~jaredsna/Maildir/new/**. As shown, the output was 0, meaning that no files were written in the first place.

```
[jaredsna@jaredserver ~]$ sudo ls -l ~jaredsna/Maildir/new/
total 0
```

2. We then entered the main configuration file, and it was visible that *$mydomain* was missing from the *mydestination* parameter. This was where the problem was.

```
jaredsna@jaredserver:~ — sudo nano /etc/postfix/main.cf
  GNU nano 5.6.1                /etc/postfix/main.cf                Modified
#
# See also below, section "REJECTING MAIL FOR UNKNOWN LOCAL USERS".
#
mydestination = $myhostname, localhost.$mydomain, localhost
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
#       mail.$mydomain, www.$mydomain, ftp.$mydomain
```

3. We add $mydomain at the last part so Postfix can recognize the email address.

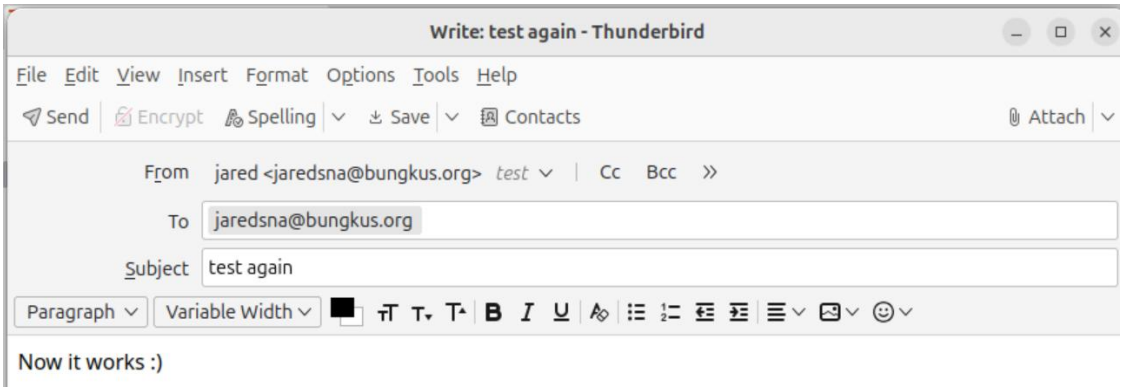
jaredsna@bungkus.org

```
jaredsna@jaredserver:~ — sudo nano /etc/postfix/main.cf
  GNU nano 5.6.1                    /etc/postfix/main.cf
# See also below, section "REJECTING MAIL FOR UNKNOWN LOCAL USERS".
#
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
#       mail.$mydomain, www.$mydomain, ftp.$mydomain
```

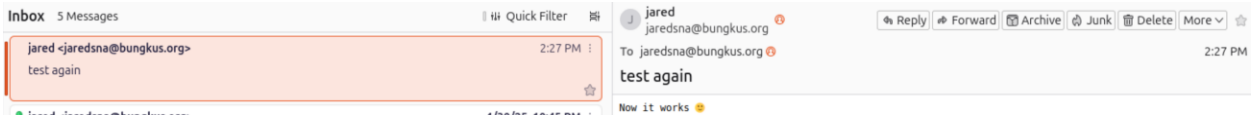
4. We save the file and exit. And restart postfix.

```
jaredsna@jaredserver:~
[jaredsna@jaredserver ~]$ sudo nano /etc/postfix/main.cf
[sudo] password for jaredsna:
[jaredsna@jaredserver ~]$ sudo nano /etc/postfix/main.cf
[jaredsna@jaredserver ~]$ sudo systemctl restart postfix
[sudo] password for jaredsna:
[jaredsna@jaredserver ~]$
```

5. Then, we tried to send another email to see if the problem persisted.

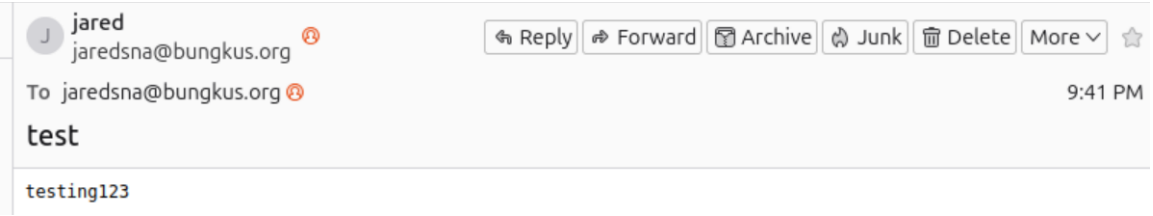


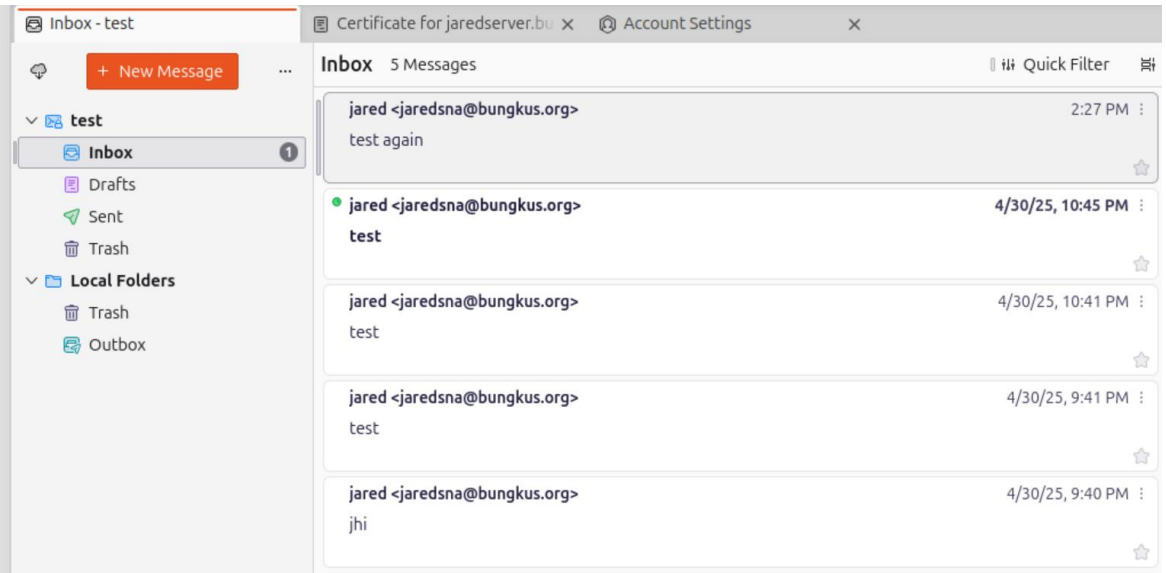
6. Now, the inbox can be updated, and we can see the email sent.



*Commands are **bolded**
Parameters & directories are in *italic*
Outputs are <u>underlined</u>*

10.0 Conclusion and Testing





The project has now successfully established a functional and secure email server environment as shown above using Postfix for mail transfer and Dovecot for mail delivery and authentication.

In conclusion, by carefully configuring these services and the Thunderbird client, the system is now capable of handling email communication for local users. The troubleshooting process, which involved the analysis of server logs and configuration files, was crucial in identifying and resolving issues related to client authentication and local mail delivery. Furthermore, verification steps demonstrated that the connections between the client and the server are secured using TLS/SSL encryption, ensuring the privacy and integrity of email traffic. The successful implementation and verification of these components confirm that the project objectives of setting up a working and secure mail platform have been met.

References

1. *Stenzel, G. (2023, January 12). How to install and configure Postfix. Red Hat.* *https://www.redhat.com/en/blog/install-configure-postfix*

2. *Stenzel, G. (2023a, January 12). How to install and configure Dovecot. Red Hat.* *https://www.redhat.com/en/blog/install-configure-dovecot*

3. *M, S. (2025, January 31). What is TLS? Understanding transport layer security and how it works. Hostinger Tutorials.* *https://www.hostinger.com/my/tutorials/what-is-tls*

4. *What is Postfix? Hosting Wikipedia*. (2021, April 12). Plesk. https://www.plesk.com/wiki/postfix/

5. *Chapter 9. Configuring and maintaining a Dovecot IMAP and POP3 server | Red Hat Product Documentation*. (n.d.). https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/8/html/deploying_different_types_of_servers/configuring-and-maintaining-a-dovecot-imap-and-pop3-server_deploying-different-types-of-servers#preparing-dovecot-to-use-virtual-users_setting-up-a-dovecot-server-with-pam-authentication

6. *What is Dovecot? Hosting Wikipedia*. (2021, April 12). Plesk. https://www.plesk.com/wiki/dovecot/

7. *4.324. thunderbird | Red Hat Product Documentation*. (n.d.). https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/6/html/6.2_technical_notes/thunderbird

*Commands are **bolded**
Parameters & directories are in *italic*
Outputs are underlined*