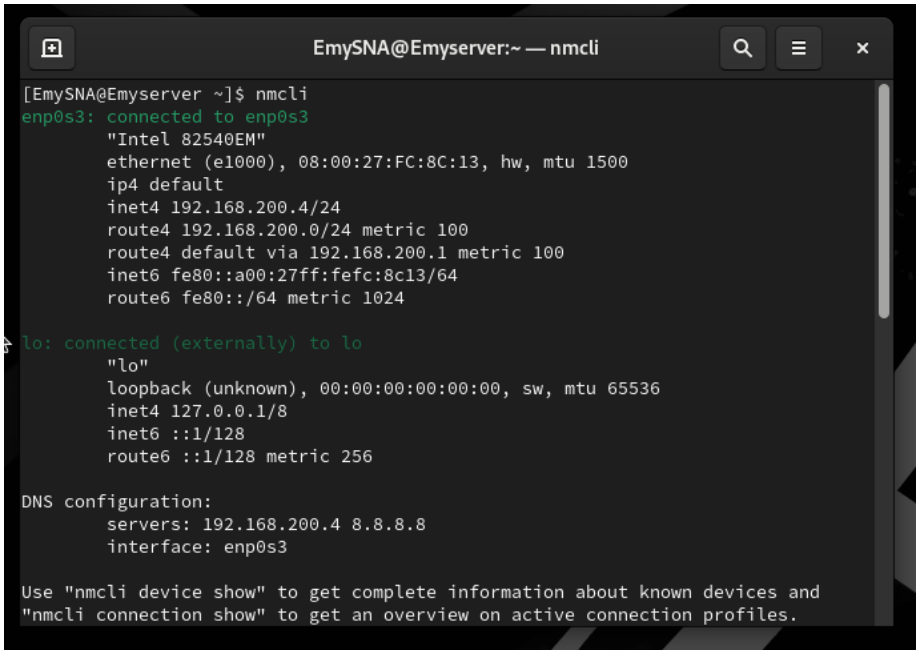
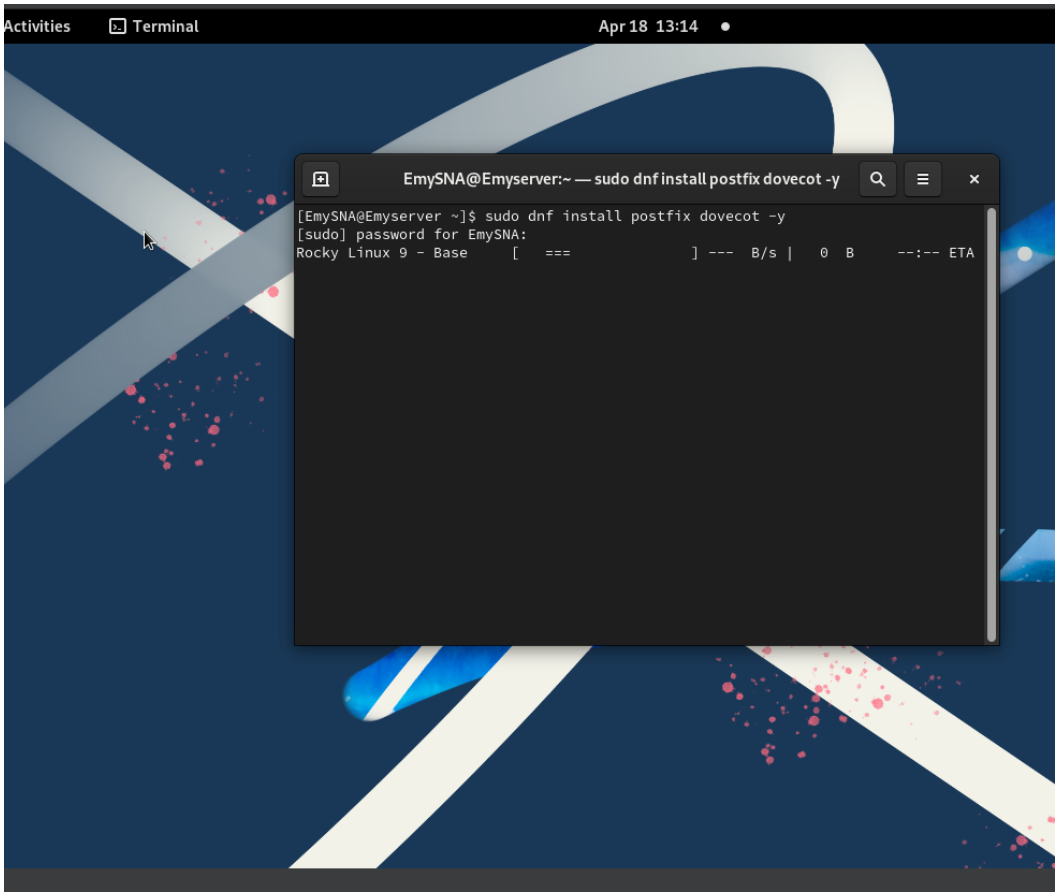
Contents

Configure SSL:

First step is to use nmcli to Identify and list out our server configuration:

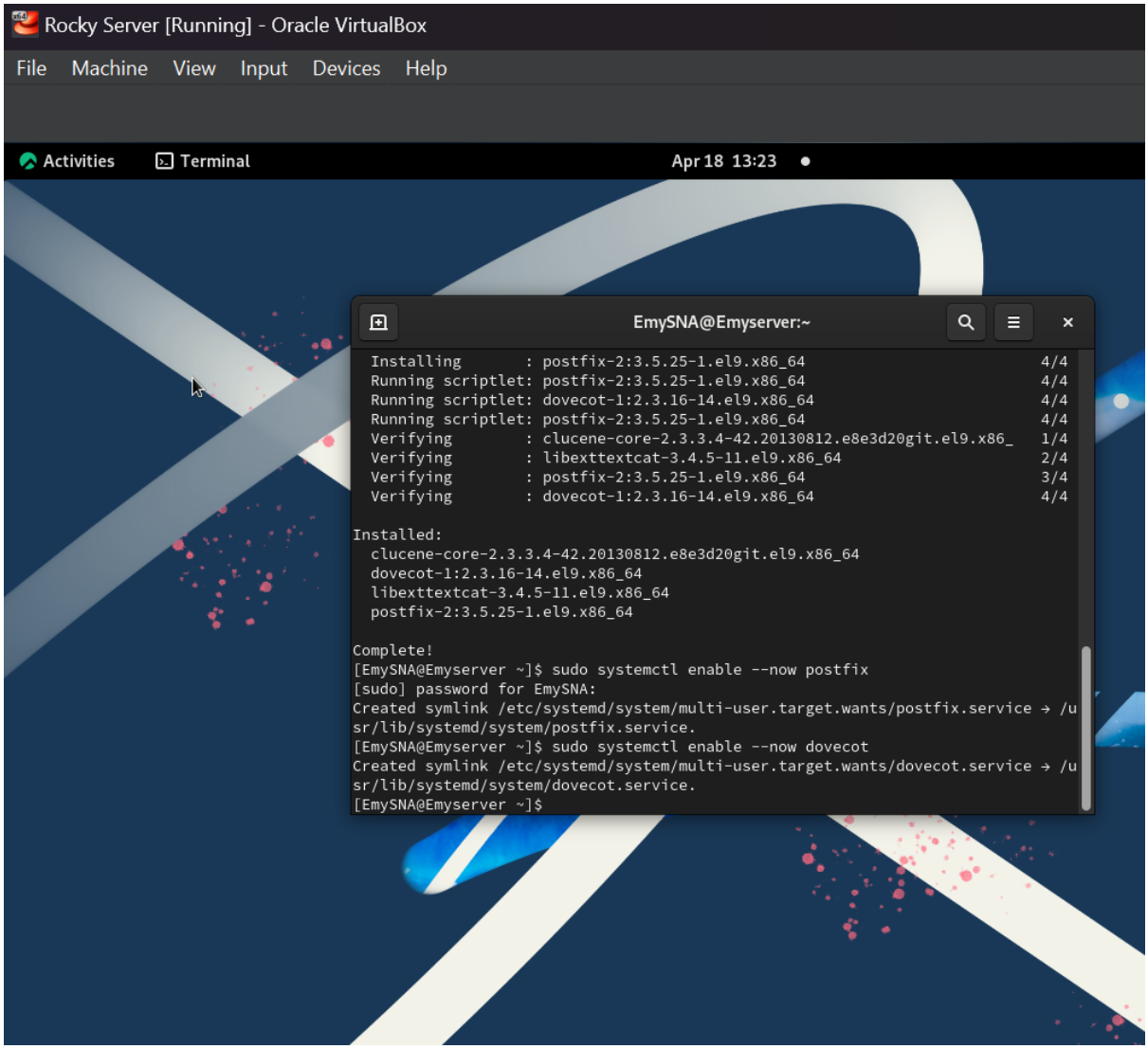


First thing, ensure the system is updated:



Using “sudo dnf install postfix dovecot -y” we’re Install postfix to use as a MTA (Mail Transfer Agent) to send and receive emails using SMTP (simple mail transfer protocol) as

well as Install dovecot as our mail delivery agent (MDA) software to handle IMAP for sending messages as a message transferring protocol to enable user to access mail on various d and the message transferring protocol POP3 to download message to our machine when connected to the internet and delete it from server since it's a message transferring protocol in case of unstable internet connection.



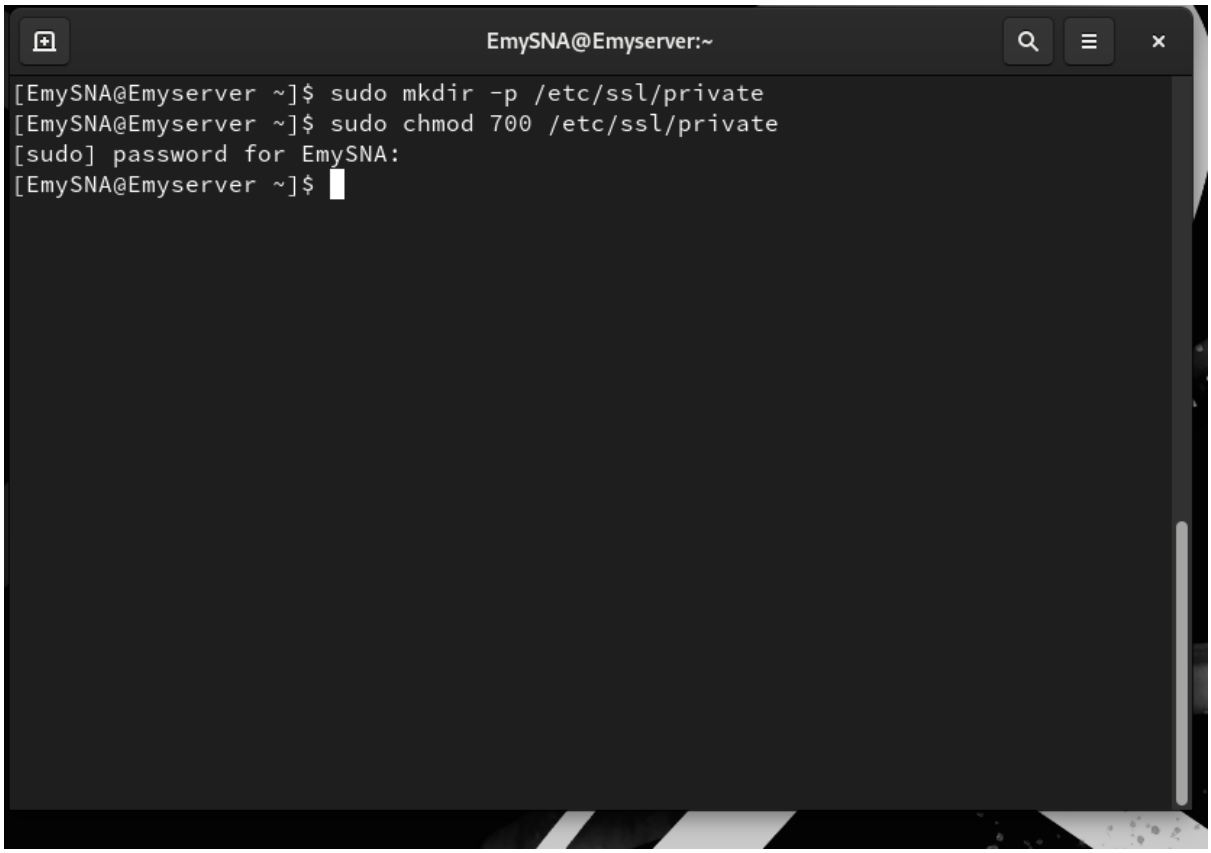
Use "sudo systemctl enable --now postfix

sudo systemctl enable --now dovecot"

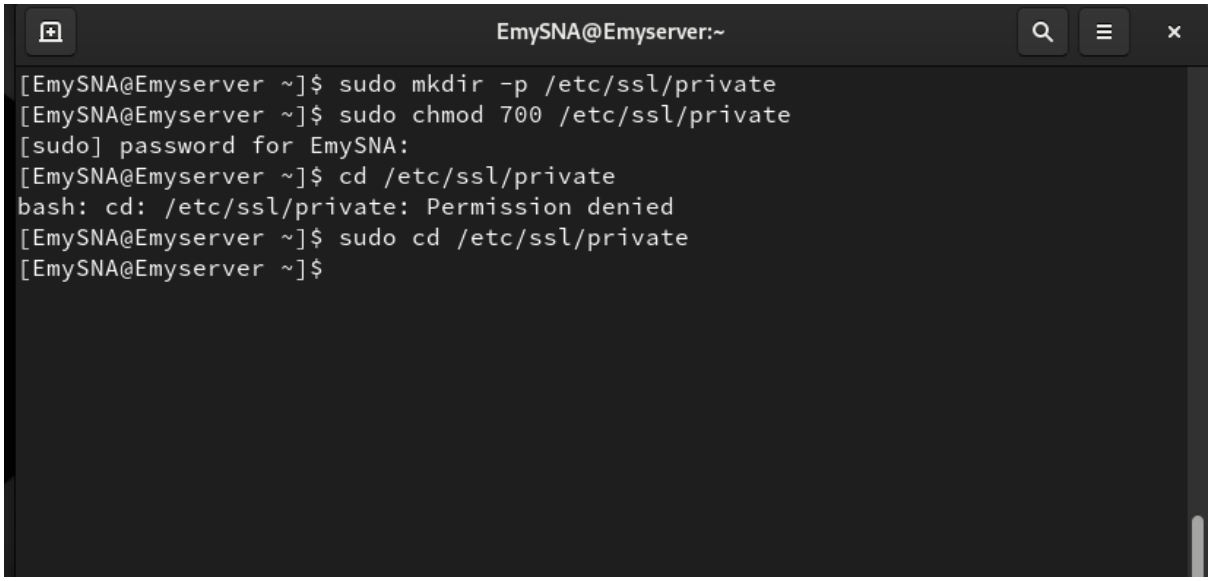
to ensure both services start immediately and ensure mail server will be active after reboot

next we create a directory to store the private SSL/TLS encryption keys securely in the system:

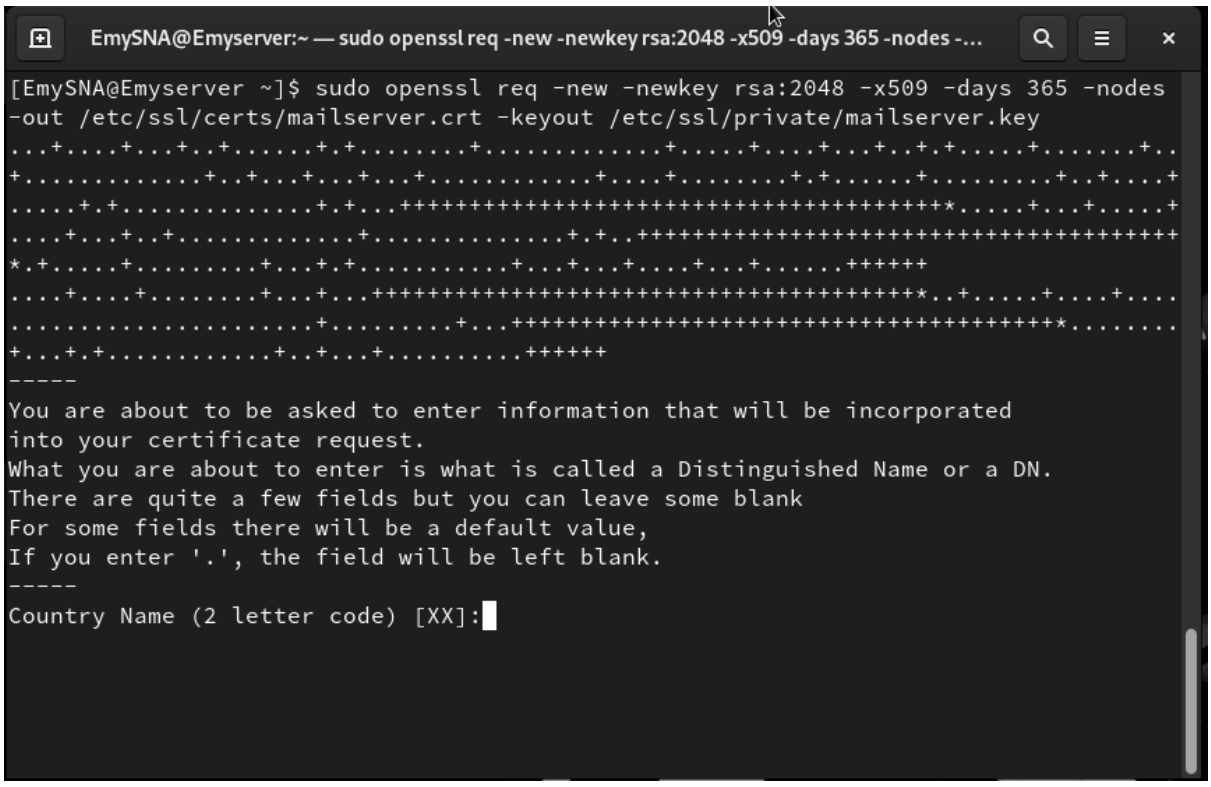
We then set permissions int the recently created directory where only the root (owner) can write (4), read(2) and execute(1) hence the 7, whereas group and others can't hence 0s:



We then change directories to the recently created one.



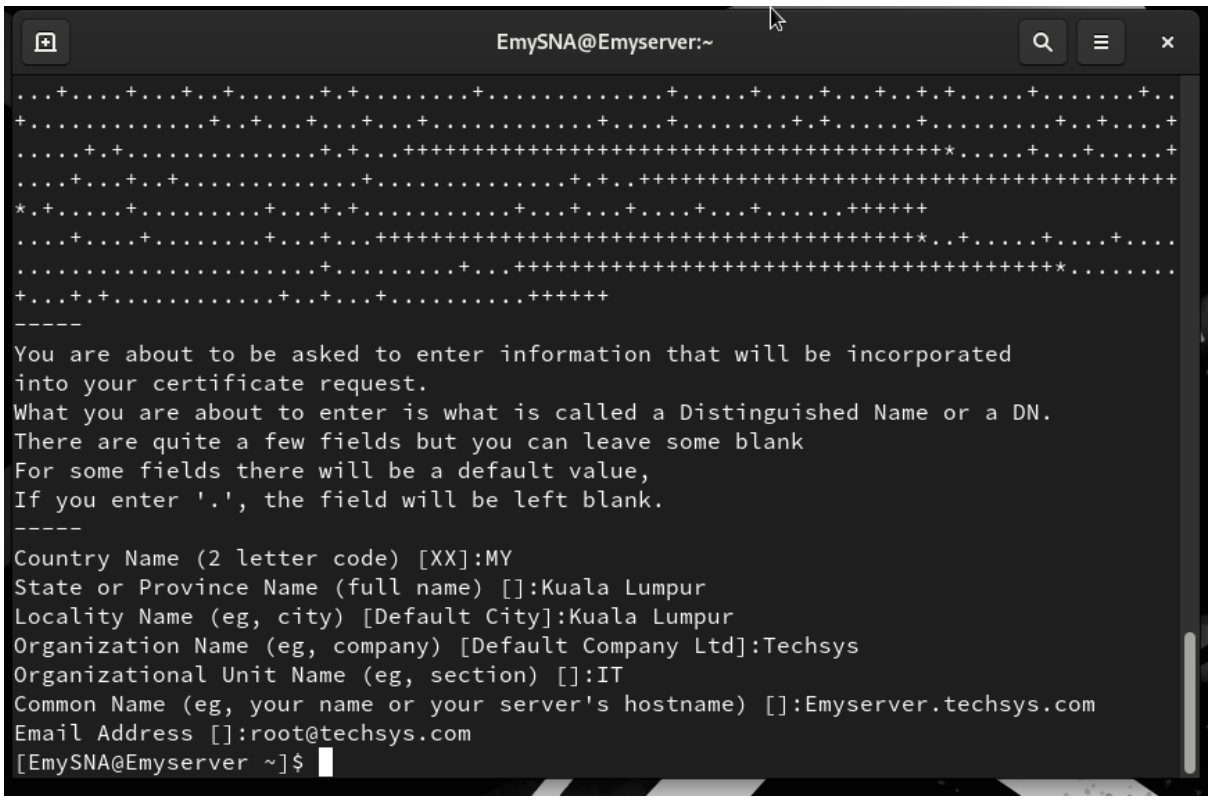
Then we generate a self-signed TLS certificate and private key to be used for mail servers like Postfix or Dovecot in testing or private deployments:



- new : generate a new certificate request
- newkey rsa:2048 : generate a new RSA(Rivest–Shamir–Adleman) 2048-bit key
- x509 : make self-signed certificate instead of CSR (certificate signing request)
- days 365 : sets certificate validity period to 365 days
- nodes : specifies the private key that is not to be encrypted

- out : defines the output path for the generated certificate
- keyout :define the output path for the generated private key

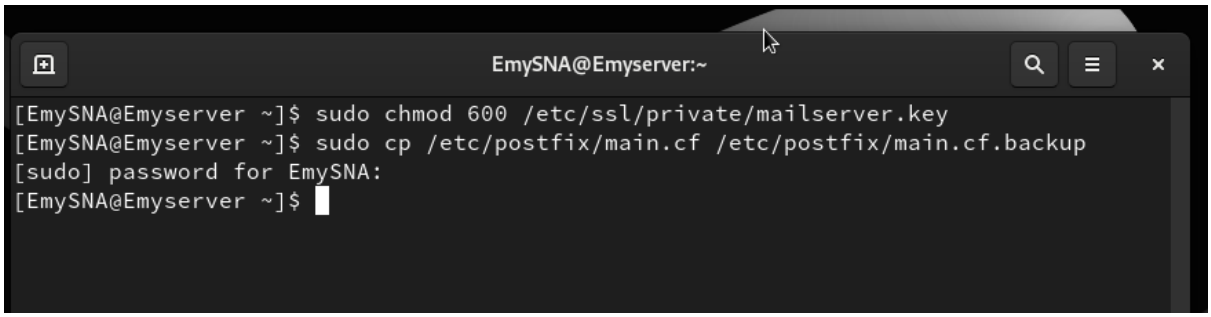
then the self-signed certificate information is filled:



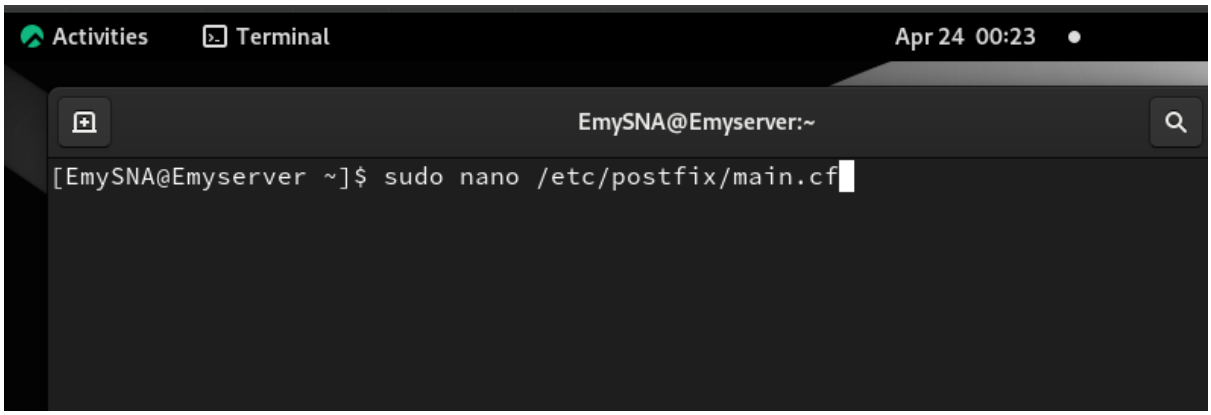
Set permission so only root can read and write the private key:

Postfix configuration

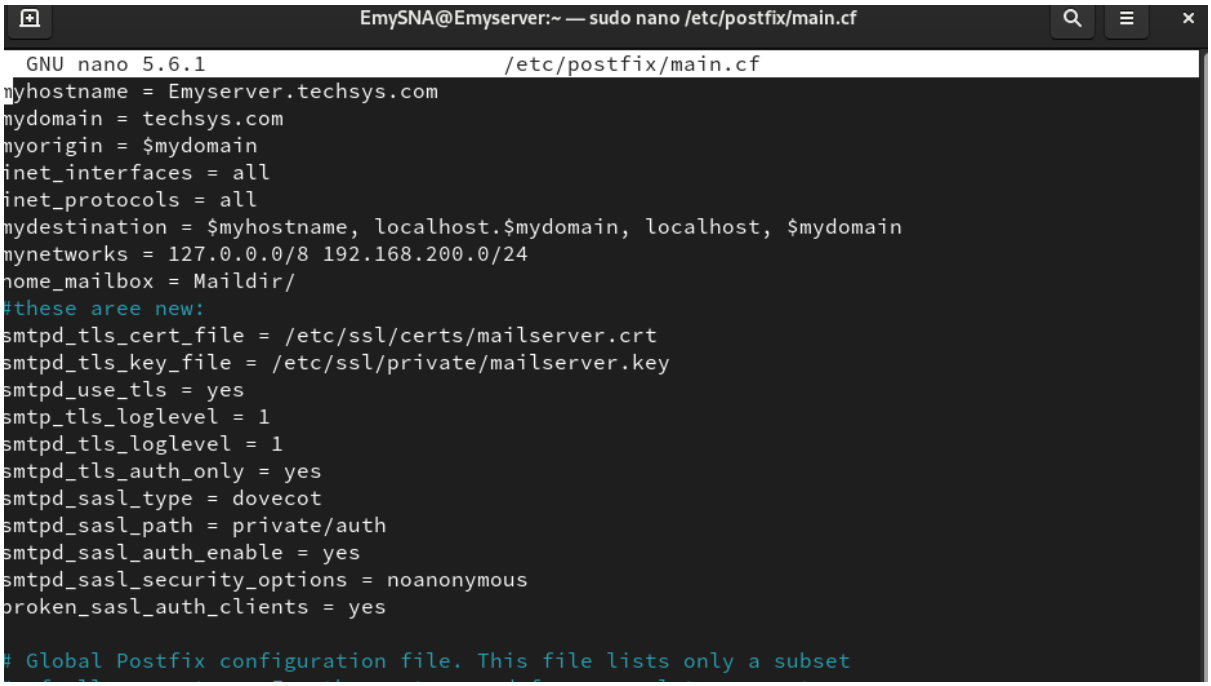
We use sudo as the original file Is owned by the root based on previous permissions to make a backup of postfix configuration file and stores it as “main.cf.backup” in the same directory



we then open a nano text editor with root privileges to edit hostname, TLS, domain…etc.:



We then modify the file to the following:



myhostname =Emyserver.techsys.org (hostname)

mydomain = tech.org (domain name)

myorigin = $mydomain (domain for outgoing emails, will be auto filled with the assigned domain)

inet_interfaces = all,

inet_protocols = all (all network interfaces and both ipv4 and IPv6 will be supports)

mydestination = $myhostname, localhost.$mydomain,localhost, $mydomain(domains this server will accept for mail purposes)

mynetworks = 127.0.0.0/8, 192.168.200.0/24 (IP addresses range allowed to send via server)

home_mailbox = Maildir/ (stores mail in “/Maildir/”) :

smtpd_tls_cert_file = /etc/ssl/certs/mailserver.crt (The location of the SSL certificate used to encrypt connections)

smtpd_tls_key_file = /etc/ssl/private/mailserver.key (private key that matches the certificate above)

smtpd_use_tls = yes (Turns on TLS encryption for incoming mail)

smtp_tls_loglevel = 1 (Log basic info about outgoing TLS connections to smtp)

smtpd_tls_loglevel = 1 (Log basic info about incoming TLS connections to smtpd)

smtpd_tls_auth_only = yes (allow login if the connection is encrypted only)

smtpd_sasl_type = dovecot (Use Dovecot to check login usernames and passwords)

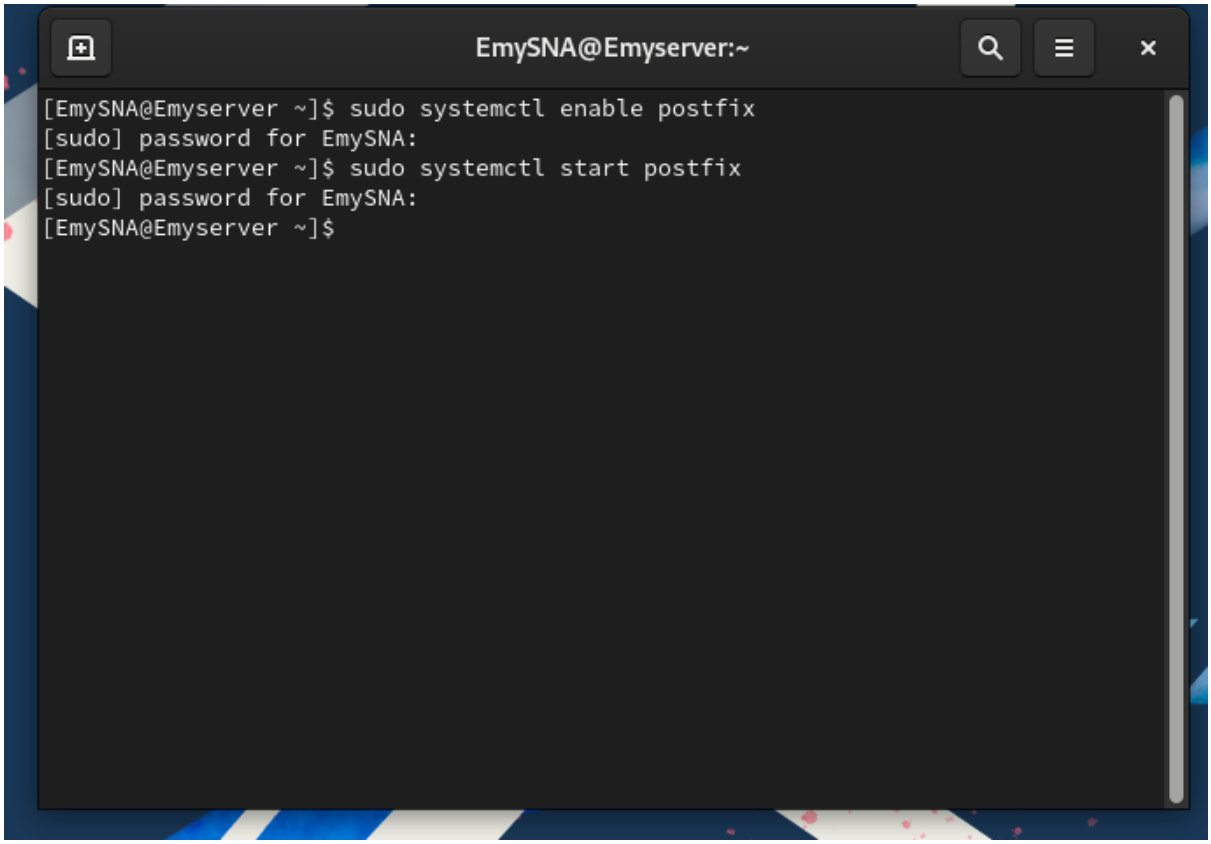
smtpd_sasl_path = private/auth (The internal socket Postfix uses totalks to Dovecot to check logins)

smtpd_sasl_auth_enable = yes (Enable login/authentication for users sending mail)

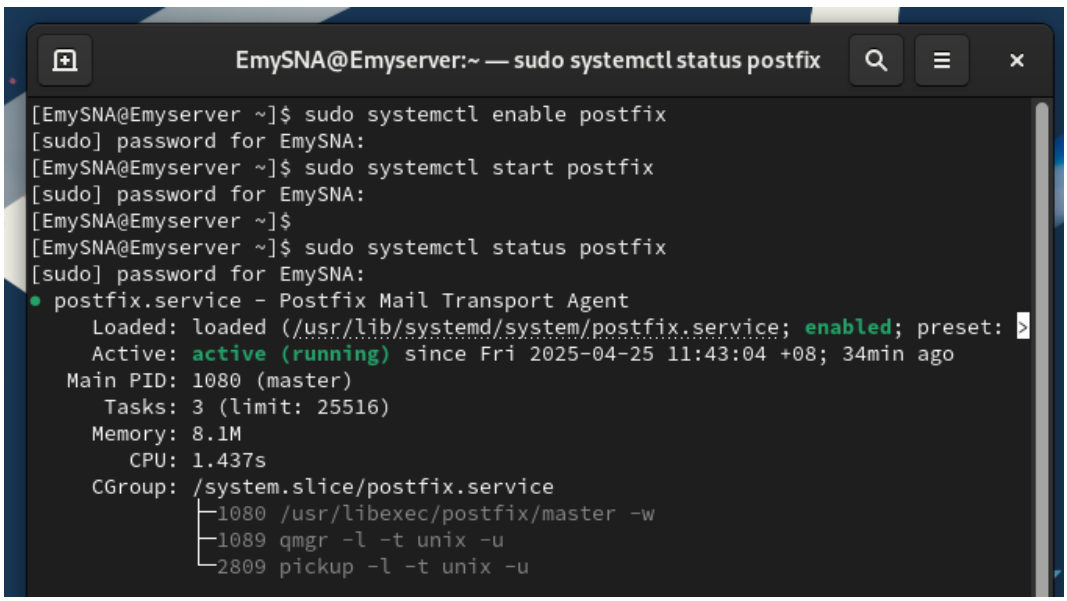
smtpd_sasl_security_options = noanonymous (reject logging in attempts without a username/password)

broken_sasl_auth_clients = yes (support old or buggy email apps that don’t follow SASL clients standards)

Enable and start postfix:

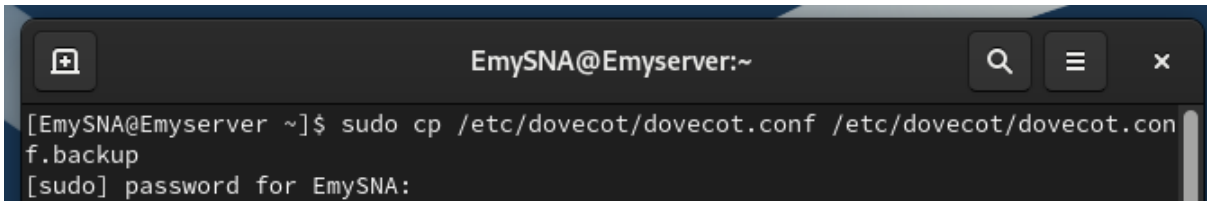


Checking postfix status to ensuer it’s enabled and active:

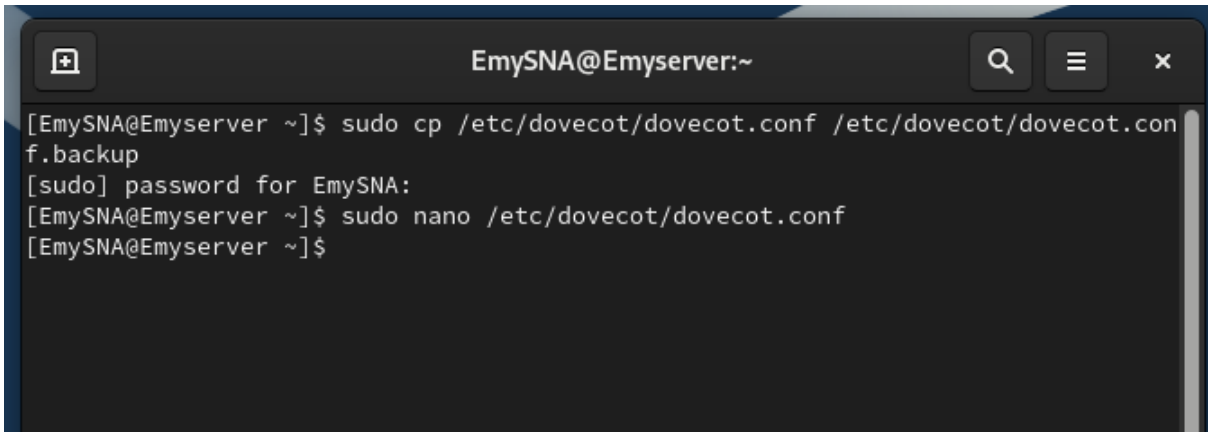


Dovecot configuration

Command to make a backup file of dovecot.conf :



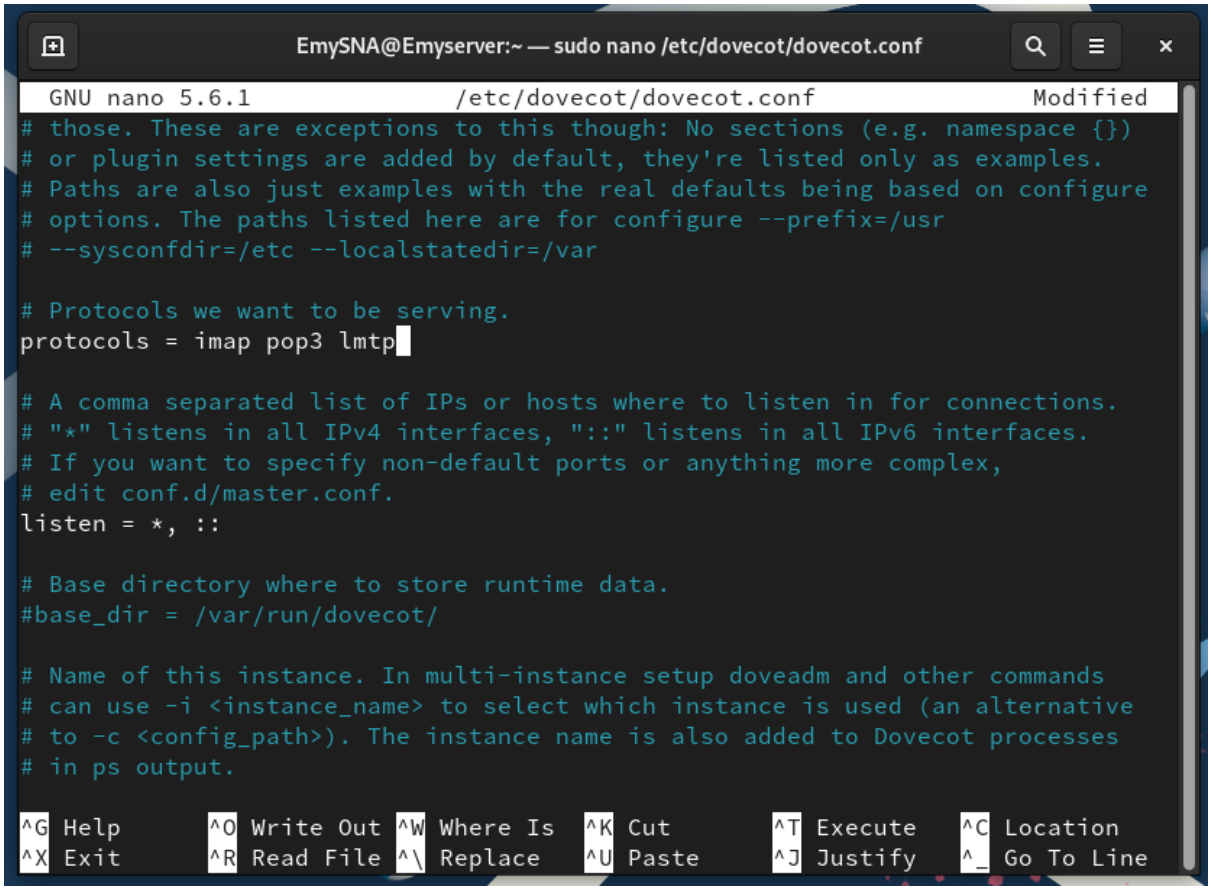
Open a text editor as root to edit dovecot configuration file which manages IMAP/POP3 mail.server:



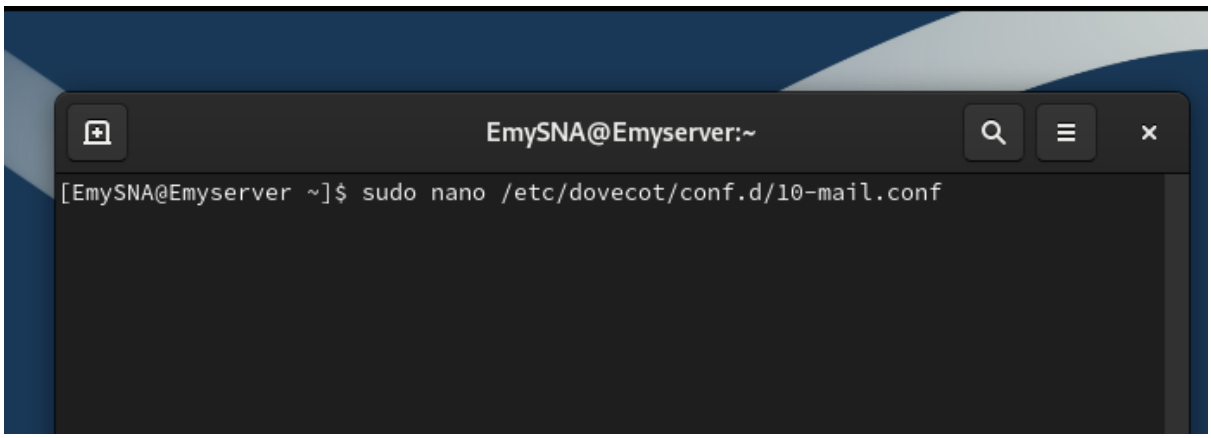
Uncomment the following lines:

Protocol = imap pop3 lmtp (to enable IMAP, POP3, LMTP)

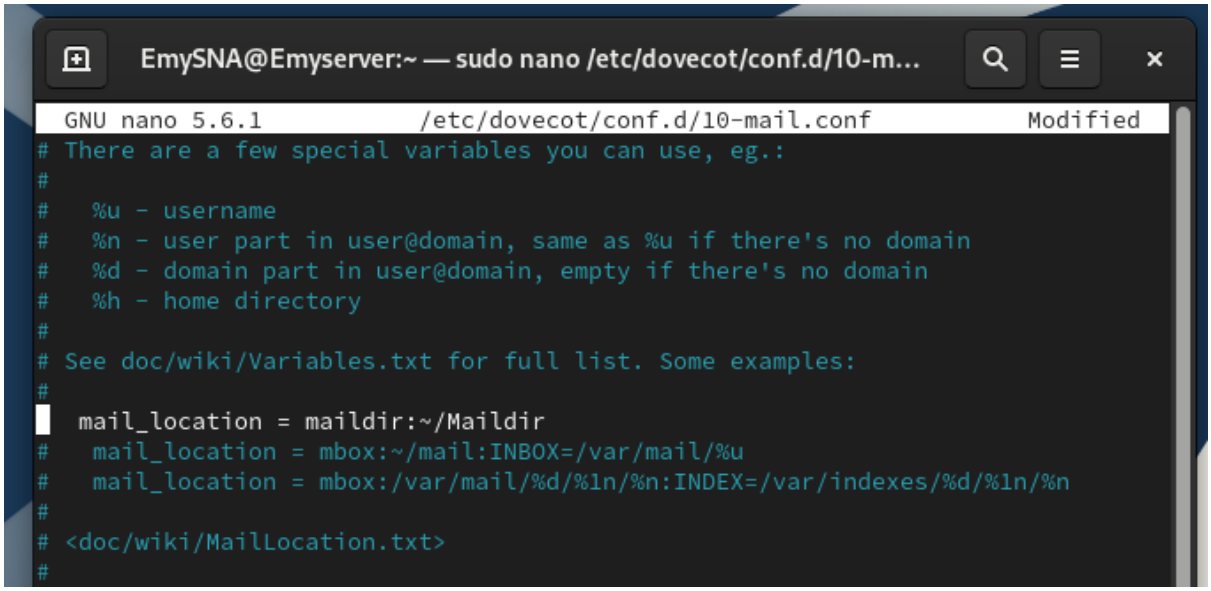
Listen = *, :: (listens on all IPv4, IPV6 address)



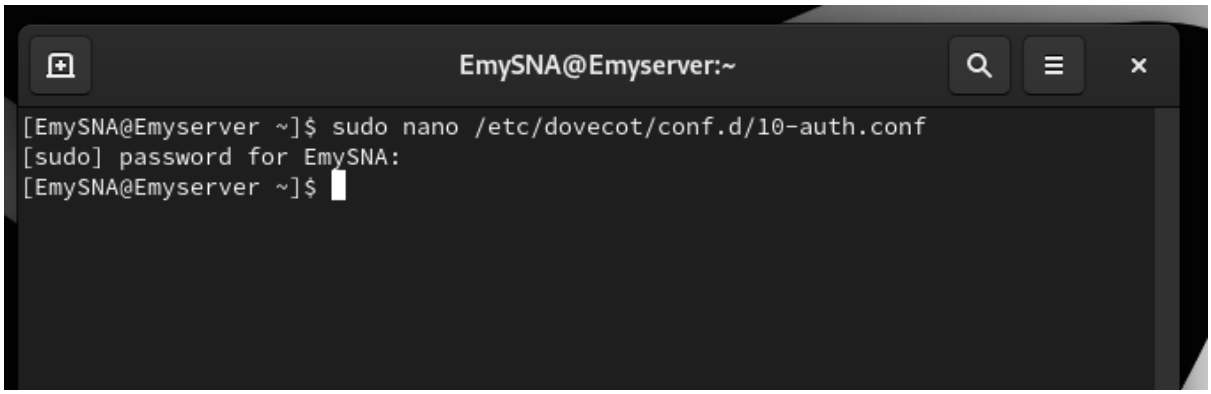
Open a text editor as root to Dovecot sub-configuration file for mail storage settings:



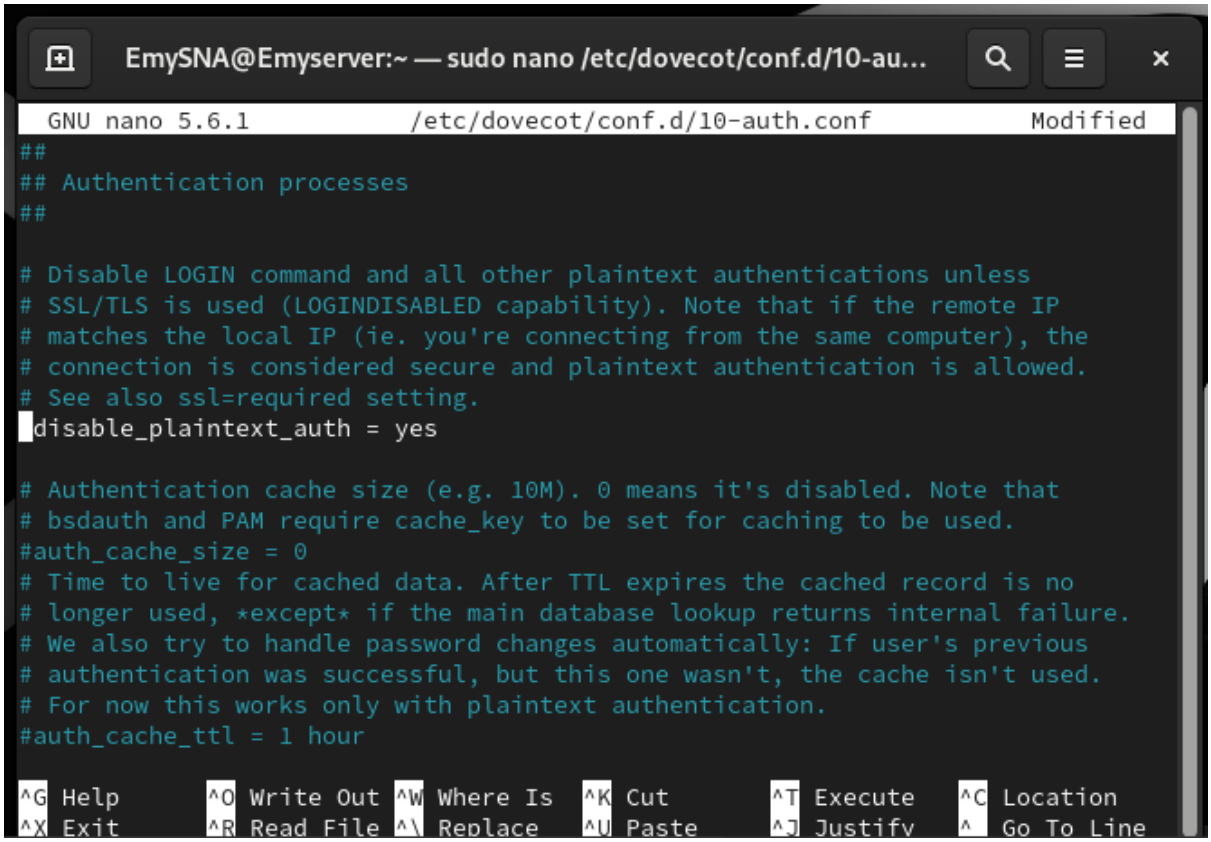
Uncomment the following the following comment that stores each user's email in a folder called 'Maildir' inside their own home directory:



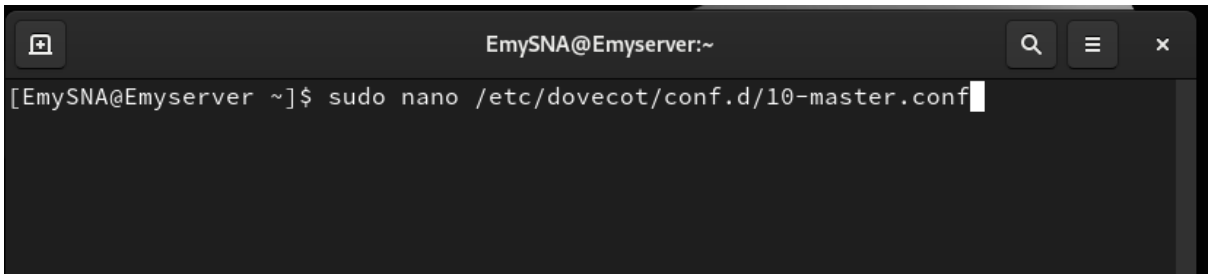
The following command uses text editor to open Dovecot text authentication settings file with root privileges:



Uncomment “disable_plaintext_auth=yes” to ensure passwords are protected unless using SSL as SSL encrypts the connection to ensure security:



Using a nano text editor to open a dovecot configuration file that runs authentication and communication with the system:



Service auth {} : this states the authentication service in dovecot

Within it the following lines are added,

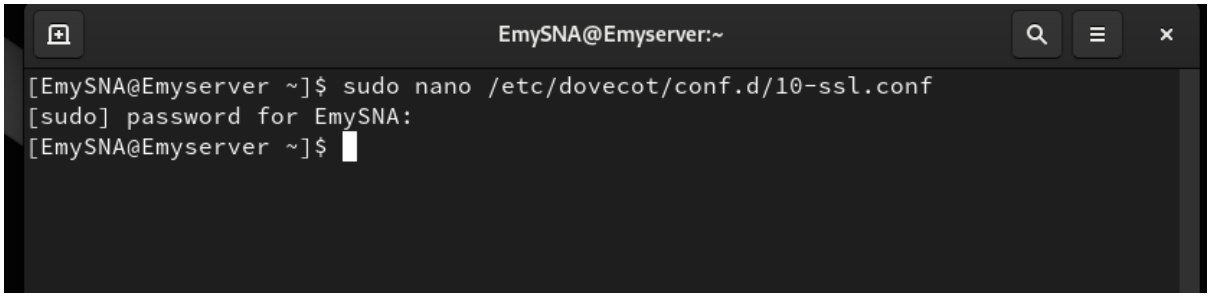
“unix_listener /va/spool/postfix/private/auth”: it creates a UNIX socket so postfix and dovecot can communicate

“mode = 0666”: set permissions so processes can read and write to socket

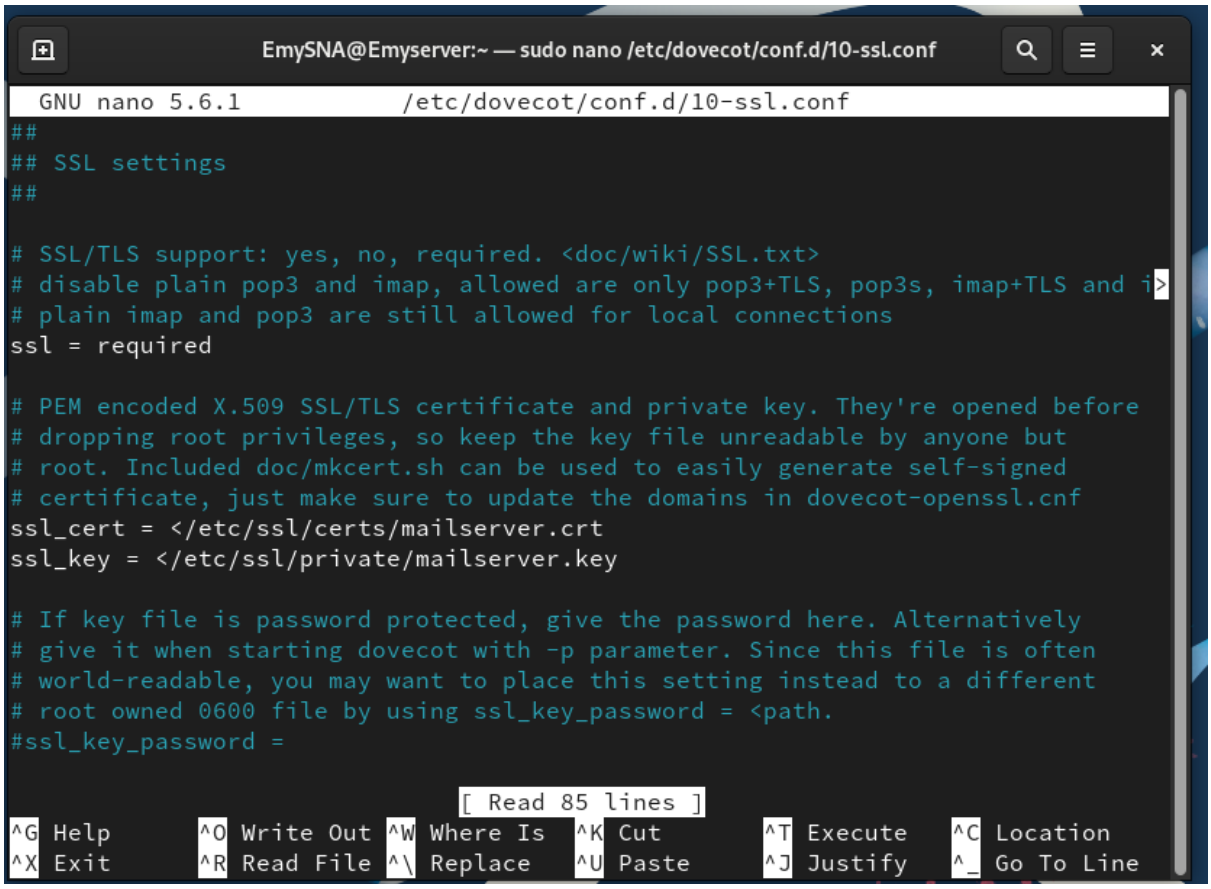
“User = postfix”: only user can use this socket file

“group = postfix": only group can use this file

Next the SSL/TLS settings file for dovecot is opened as root using nano text editor with root:



We ensure the following lines are uncommented:

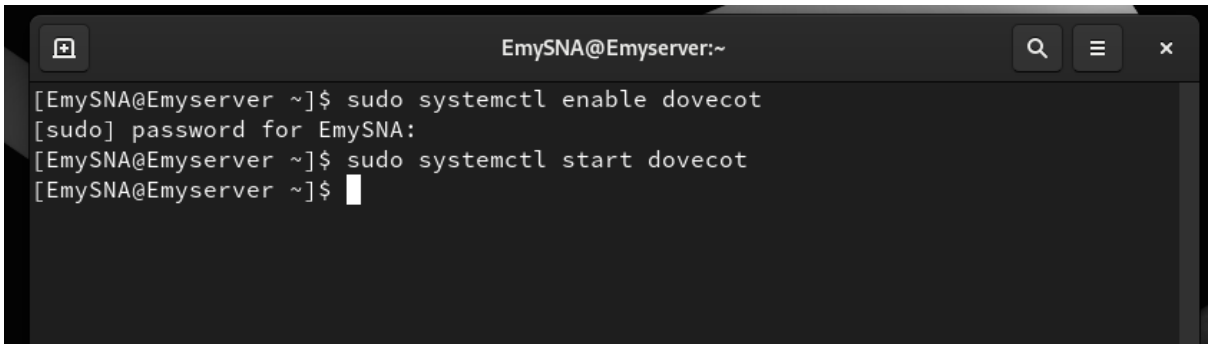


“ssl = required”: this will make dovecot only allow encrypted (SSL/TSL) connections

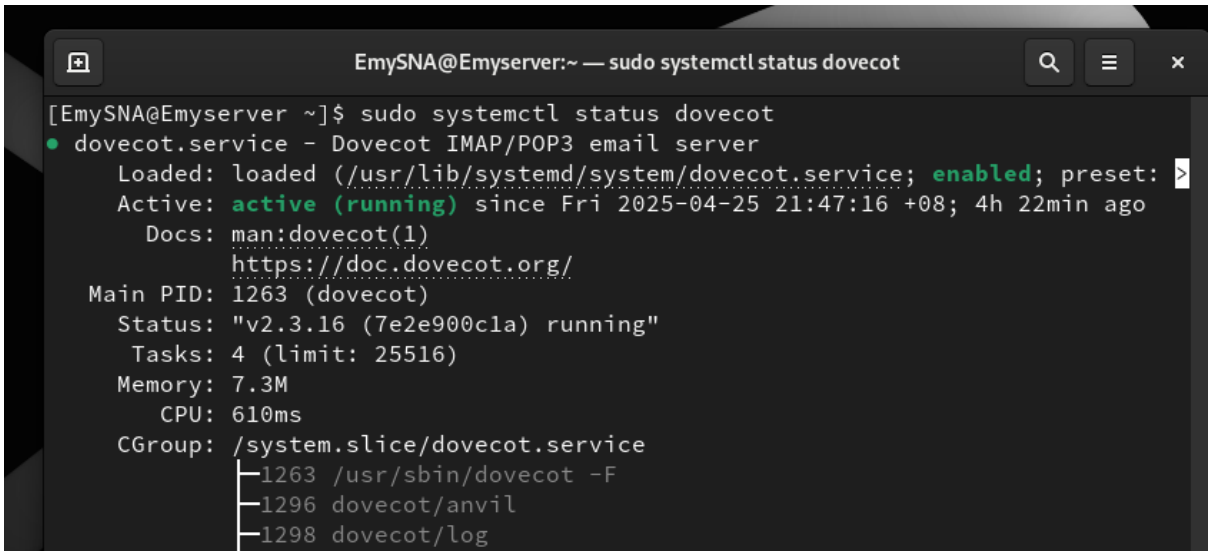
"ssl_cert = </etc/pki/dovecot/ssl/certs/dovecot.pem”: the public file, SSL certificate, which clients use to identify the server

“ssl_key = </etc/pki/dovecot/private/dovecot.key”: the private key, which is used to decrypt secure messages

Then enable and start dovecot services:



Then we ensure everything is running properly through checking the status:

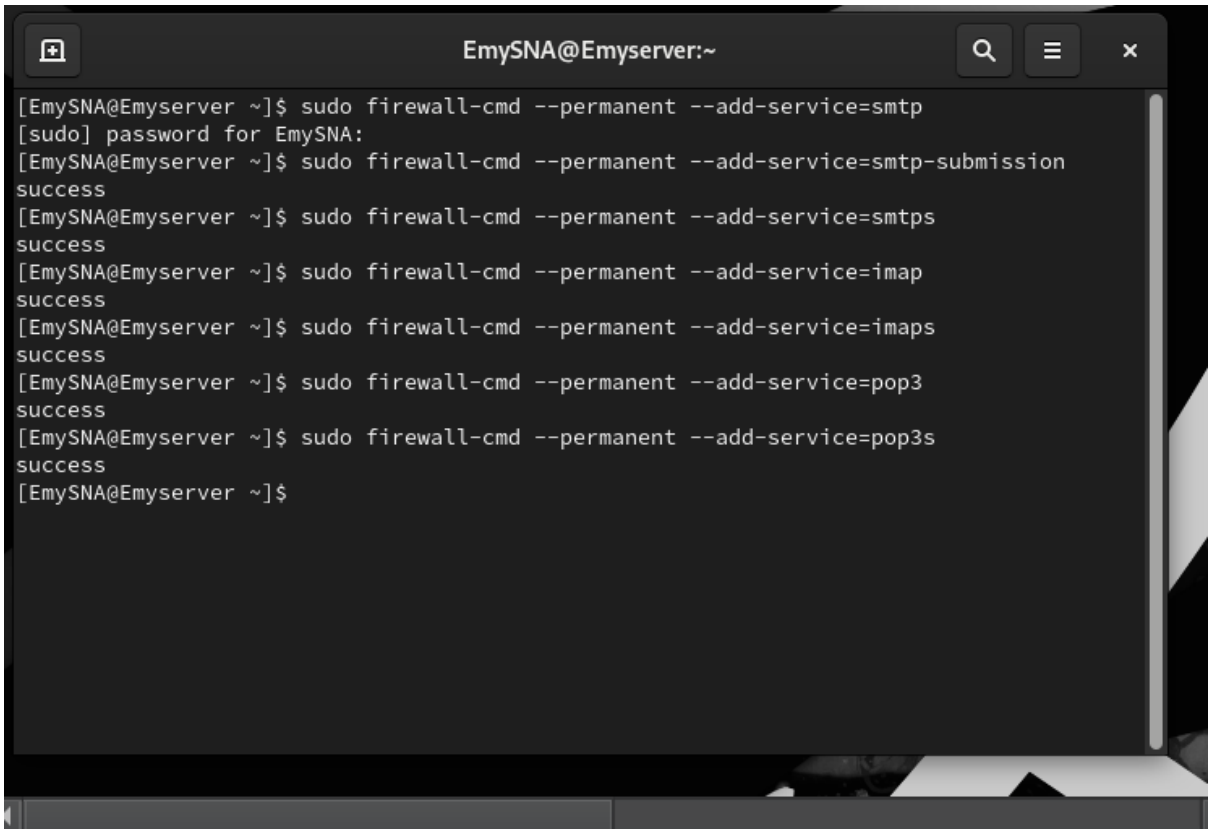


Firewall/Ports:

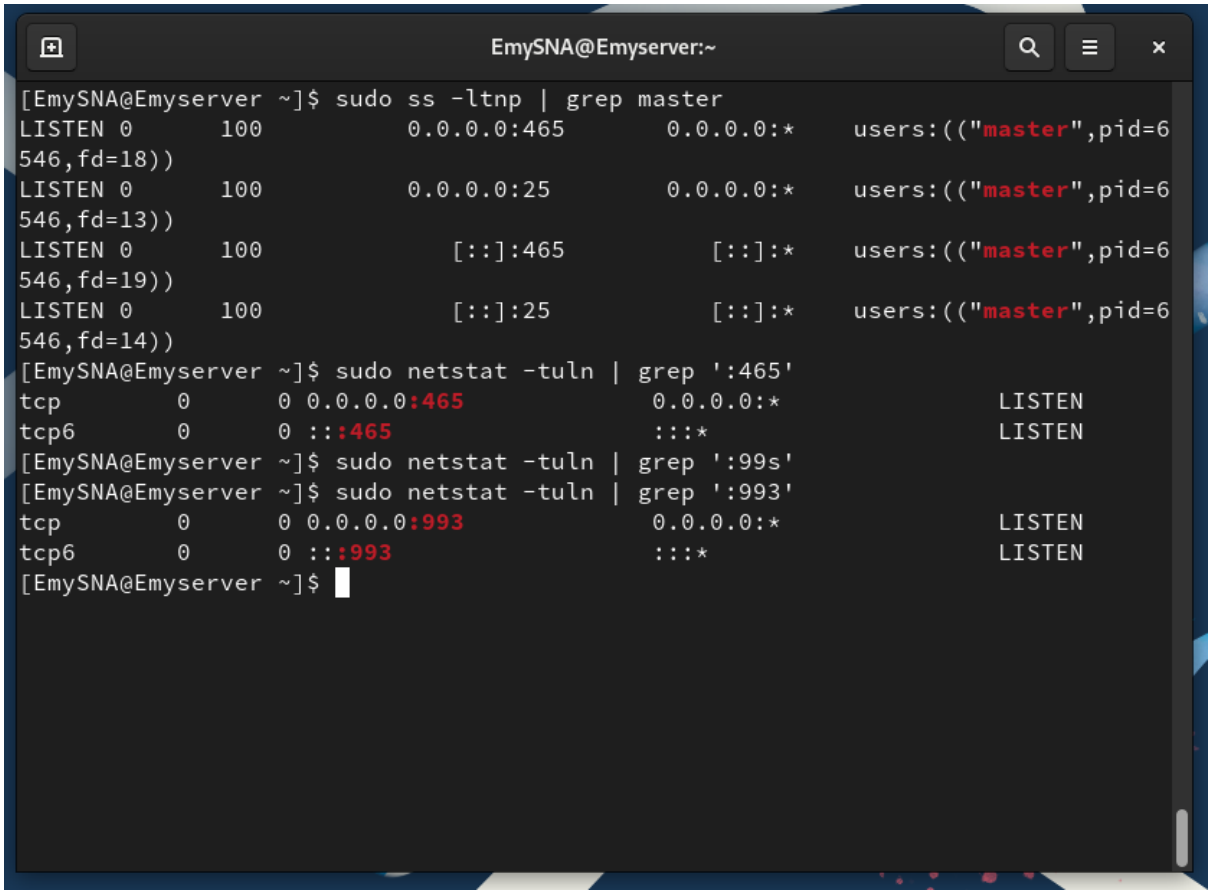
Then all traffic, sent or received through the ports, is ensured to go through the firewall

ports used and their use: SMTP for sending outgoing emails from server, SMTP-submission is to securely send mail from client, SMTPS is for sending mail over a secure and encrypted connection like SSL/TLS, IMAP is used to read mail without the need to downloading them,

IMAPS is to read emails without downloading them yet over a secure connection, POP enables users to download emails from server and remove them, and POP3S does the same job but securely, all to keep harmful data from intercepting the network connection.



We then check all active (listening) ports being used by posfix master process that manages the mail services such as SMTP, submission, etc.:



Ss: to show all active network sockets

-l: is to show the listening ports AKA the ones waiting for any sort of connection

-t: to show tcp connections only

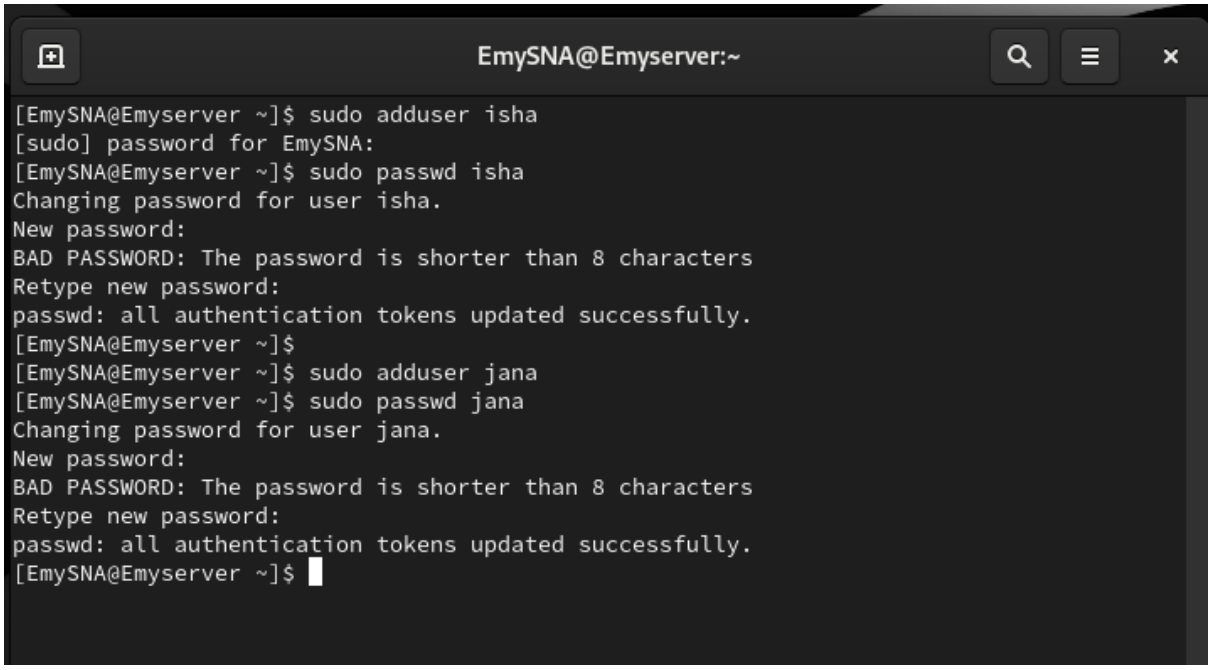
-n: to show addresses and ports not names

-p: to show processes and program using the port

| grep master: this is so only lines that mention ‘master’ (main controller process of postfix) show up.

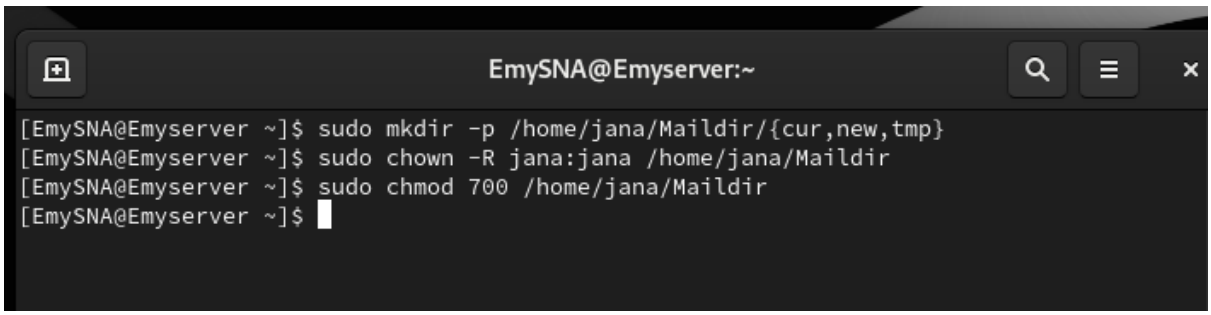
Testing

Then to test it we create users with different names and passwords using the following steps:

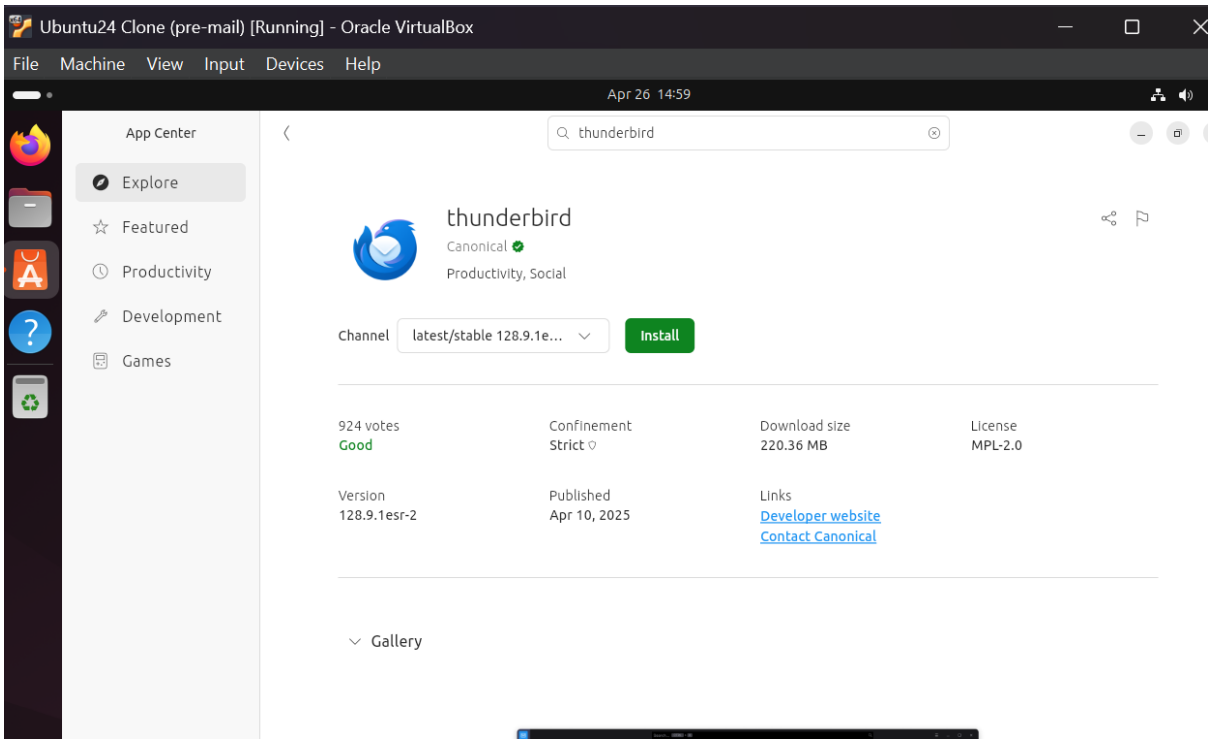


We then create a new directory for each, then change the ownership of the owner and group to the respective mail user, and give each respective user full access to read, write and execute using “chmod 700”:

Then we repeat the previous steps for the 2nd user:

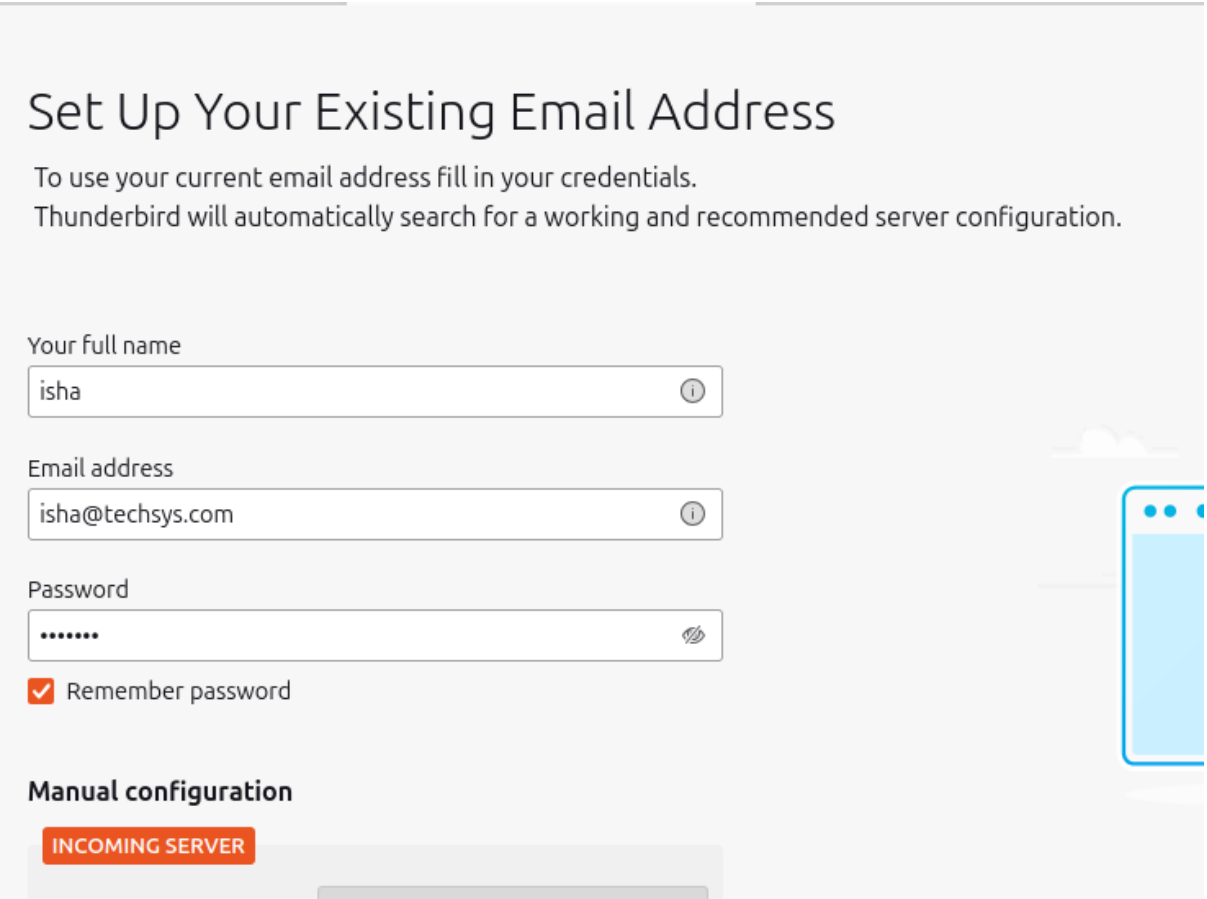


Install thunderbird on client machine (ubuntu)

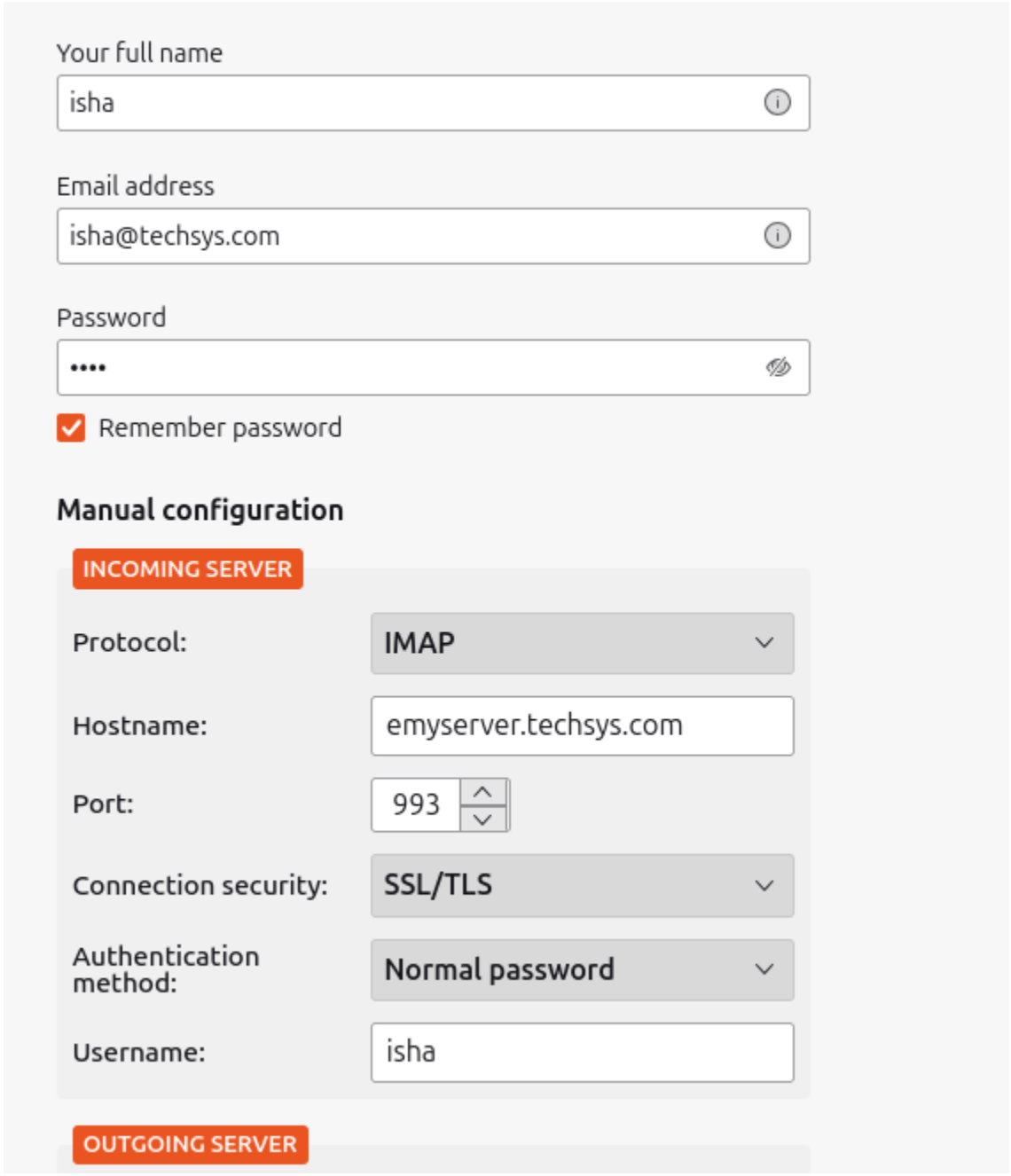


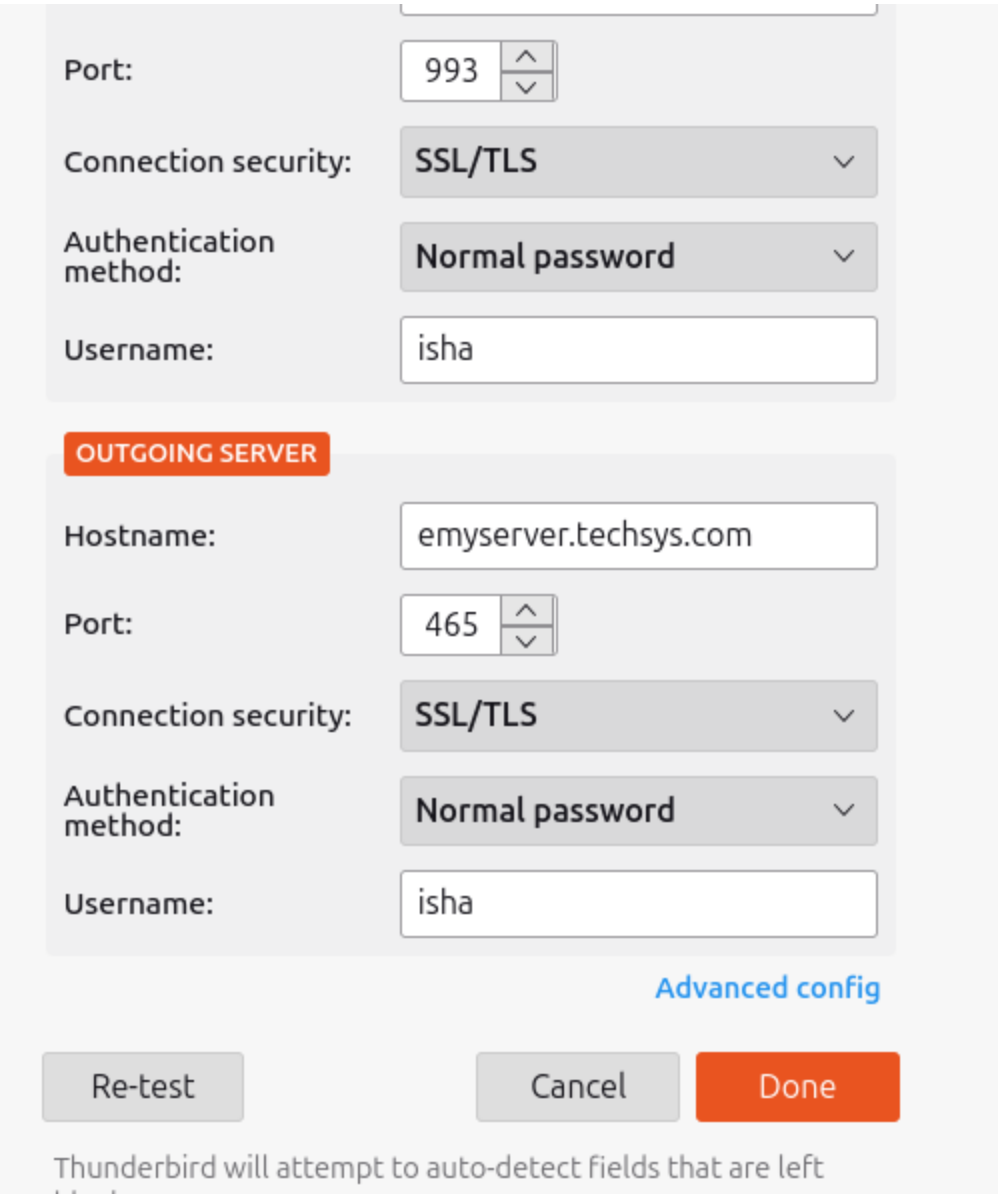
A problem arise as I'm trying to log in, steps to solve it in troubleshooting section, issue 1, we then continue from it to this next step:

Next I enter the details of created users:

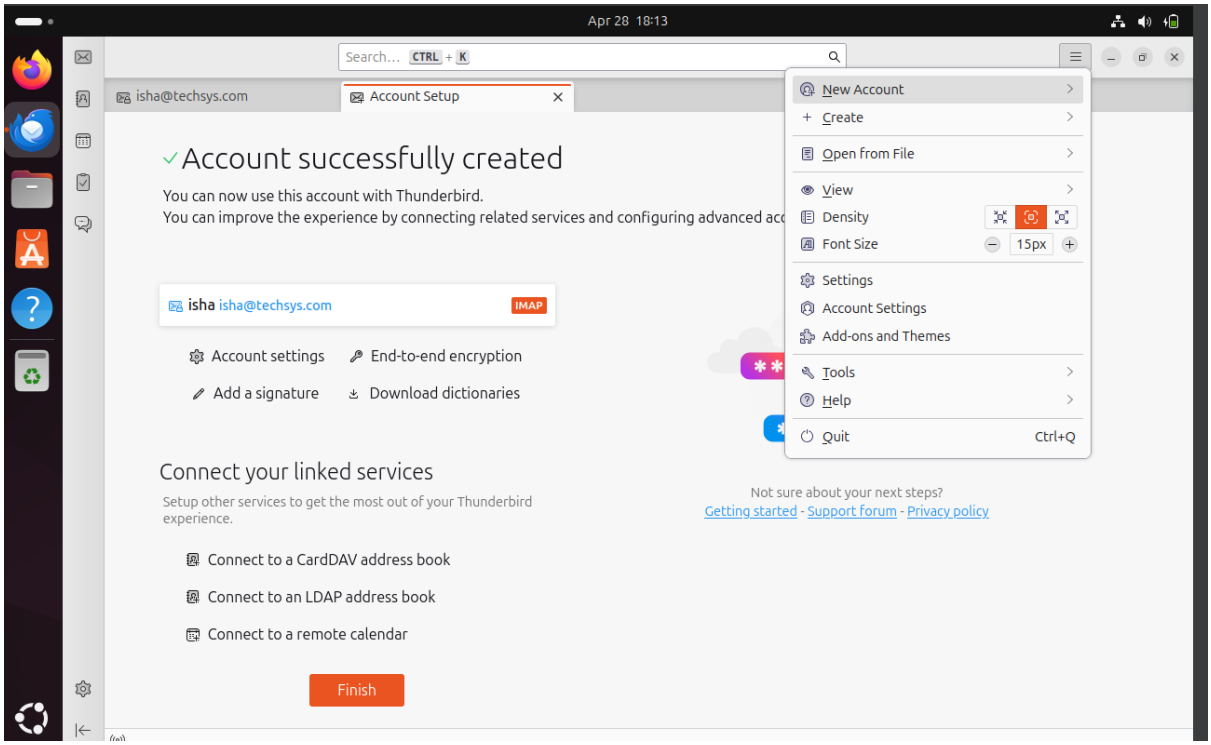


Then enter the configuration of user:

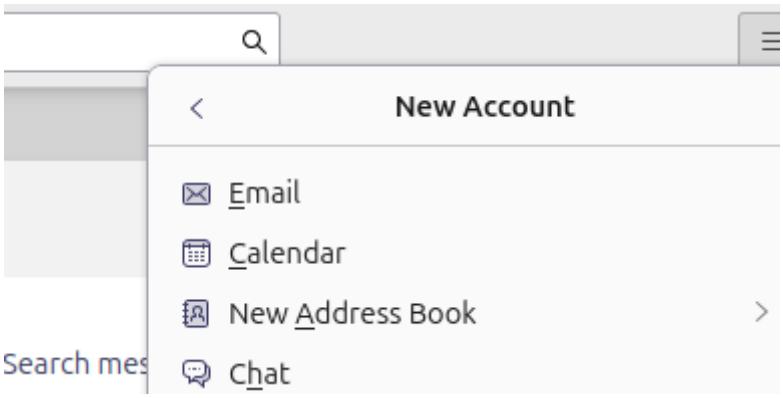

Your full name
isha
Email address
isha@techsys.com
Password
Remember password
Manual configuration
INCOMING SERVER
Protocol:
IMAP
Hostname:
emyserver.techsys.com
Port:
993
Connection security:
SSL/TLS
Authentication method:
Normal password
Username:
isha
OUTGOING SERVER



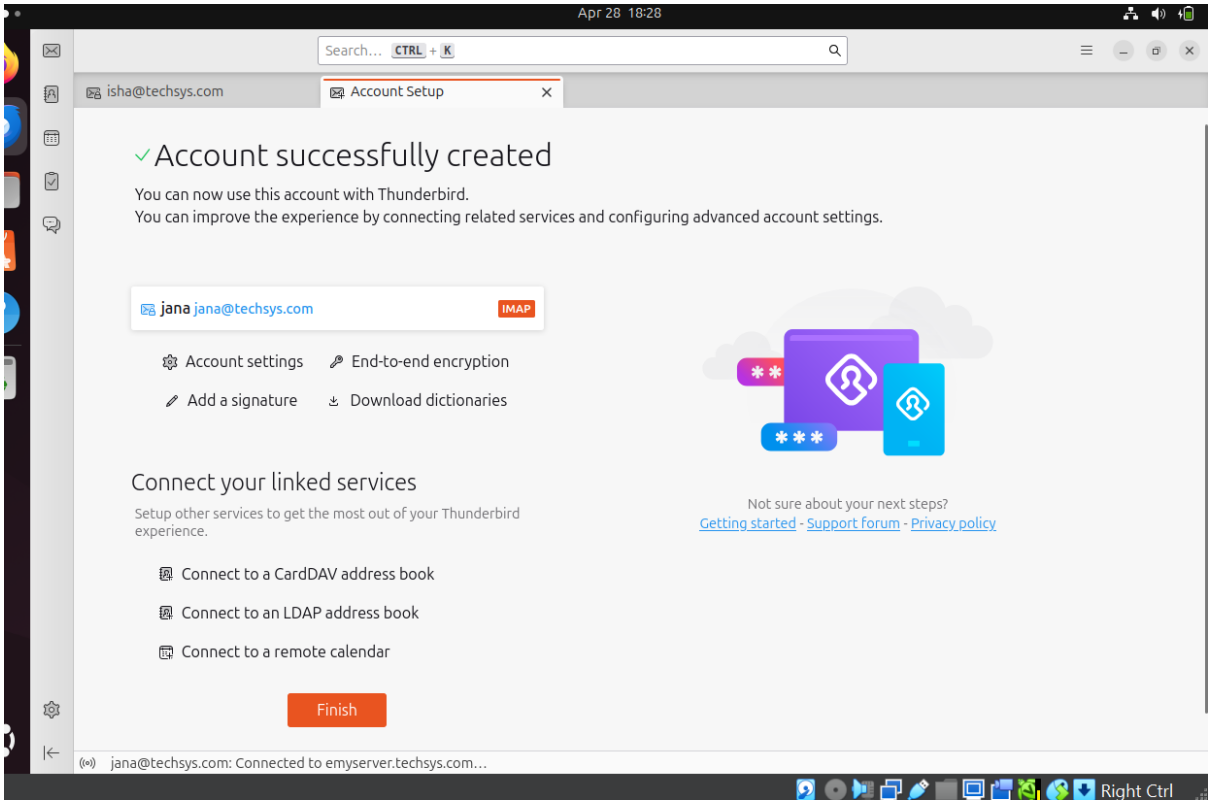
After logging in, time to log in into the 2nd user account by clicking the three lines and then "new account":



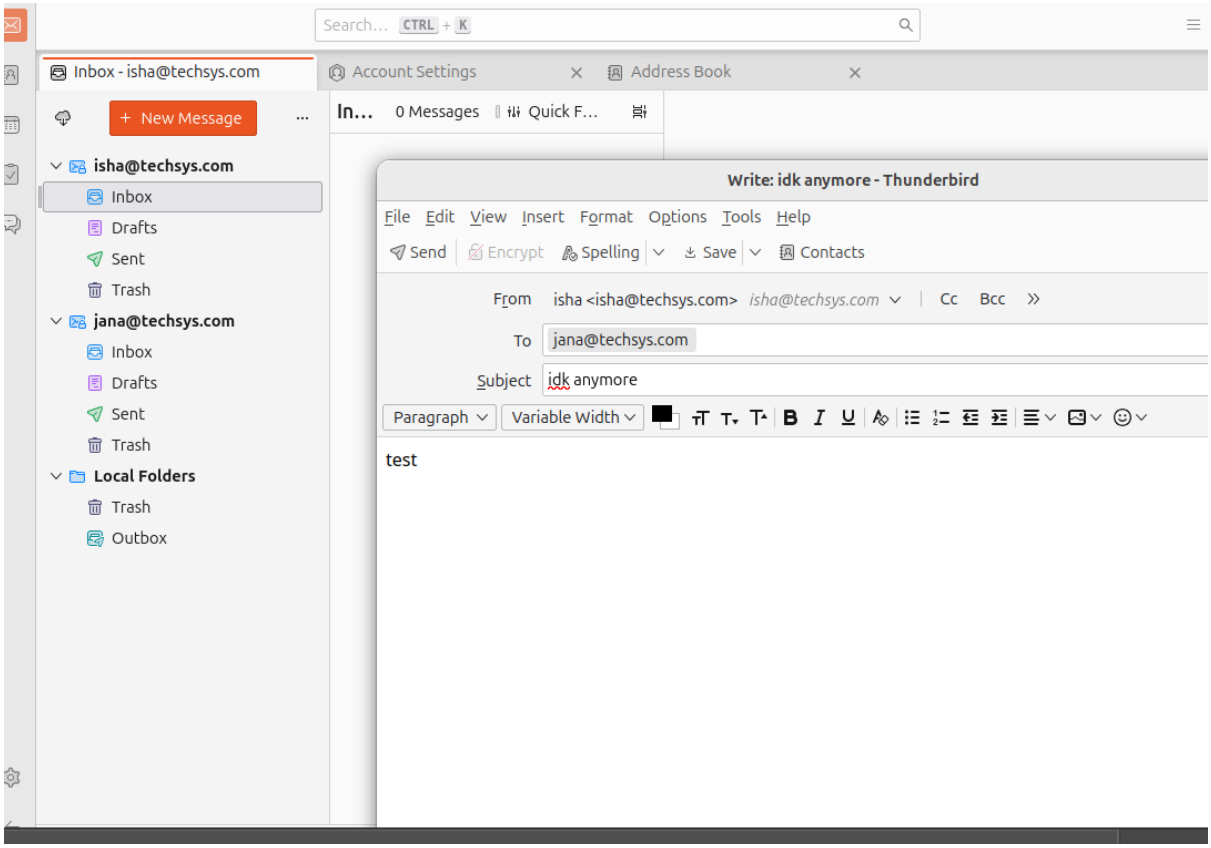
we then click “Email”



And repeat the previous steps for the 2nd user:

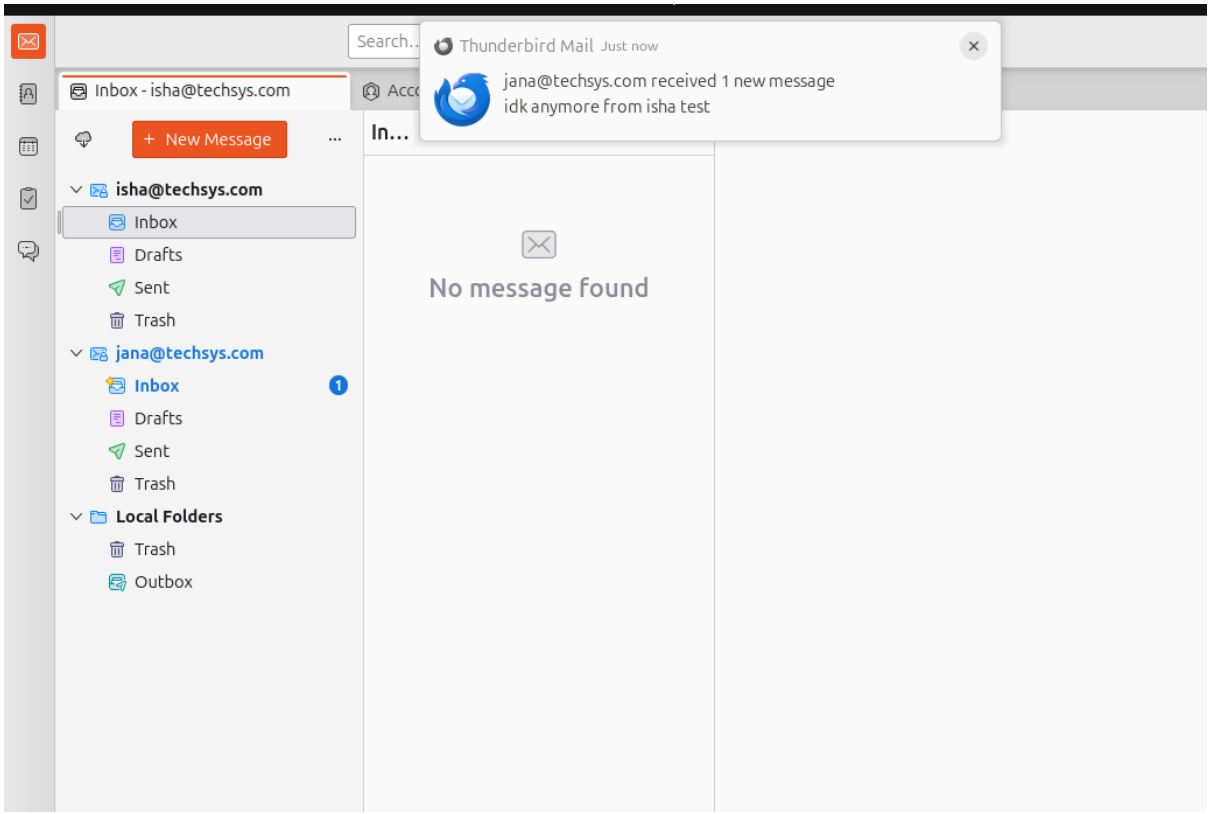


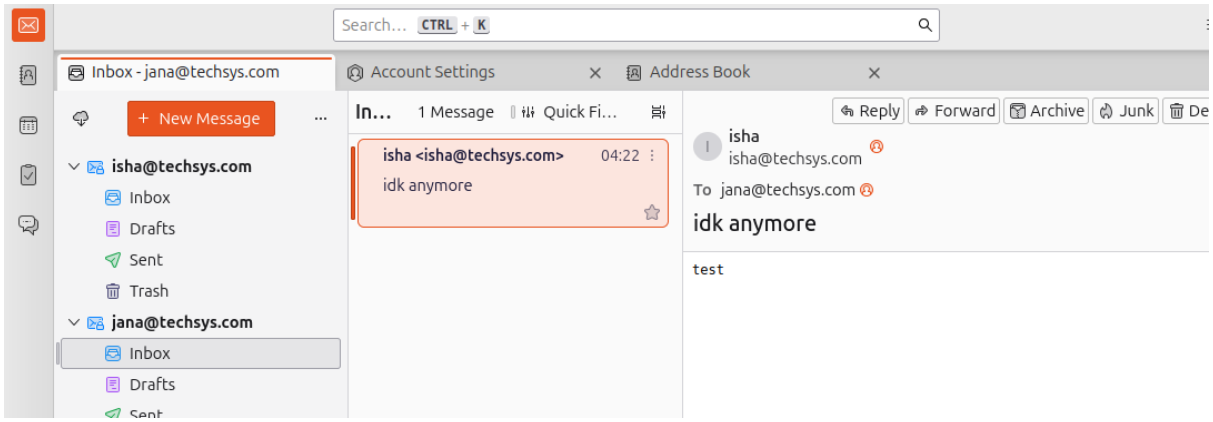
Now let's try sending an Email as a test, to send an email from "isha" user "jana" we click "+ New message" write the mail and then send it:



Another problem was faced in sending the mail, steps to solving it in troubleshooting section, issue 2. then we continue from it to this next step:

And the Email is successfully received by the recipient:



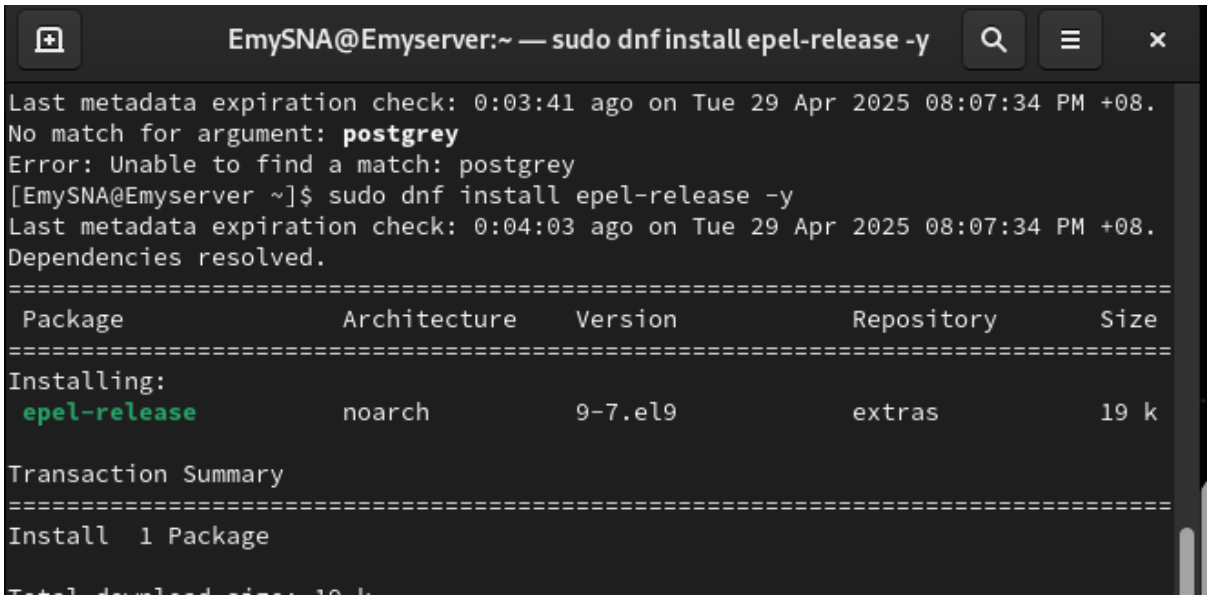


And with that we're done making a fully functioning email server.

Additional service:

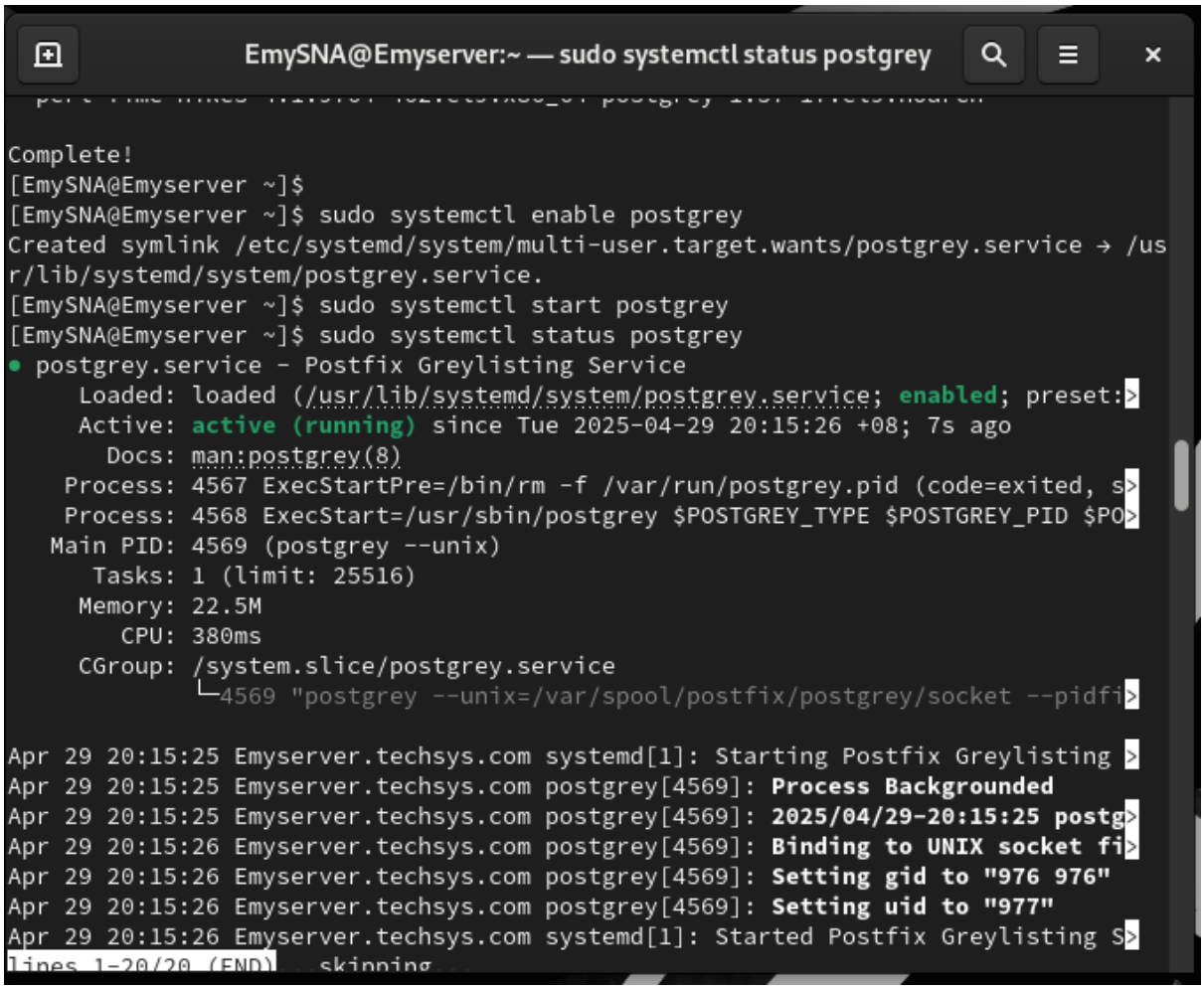
For the additional service I'll be installing a Postfix server called Postgrey mail which is used for greylisting, which is a method to filters spam by temporarily rejecting emails from unknown senders as spam sending server don't retry sending emails but legitimate mail servers do so after a short duration of time hence if the server retries sending mail it's regarded as legitimate mail and delivered normally on the mail server's second attempt.

As postgrey isn't found in my machine I must enable EPEL (extra packages for enterprise linux):

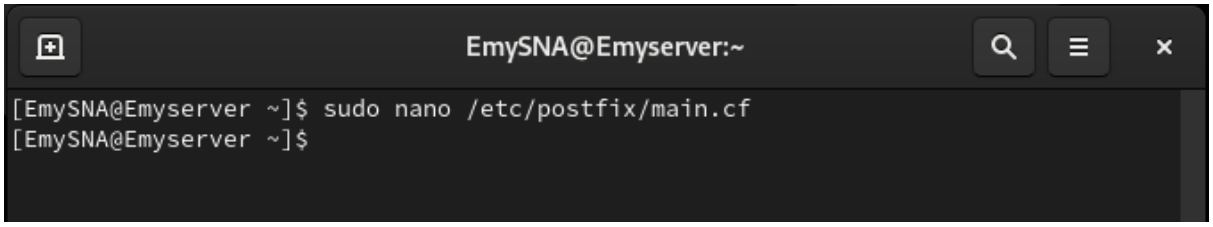


Then I installed postgrey:

Afterwards postgrey is enabled started then I checked it's status:



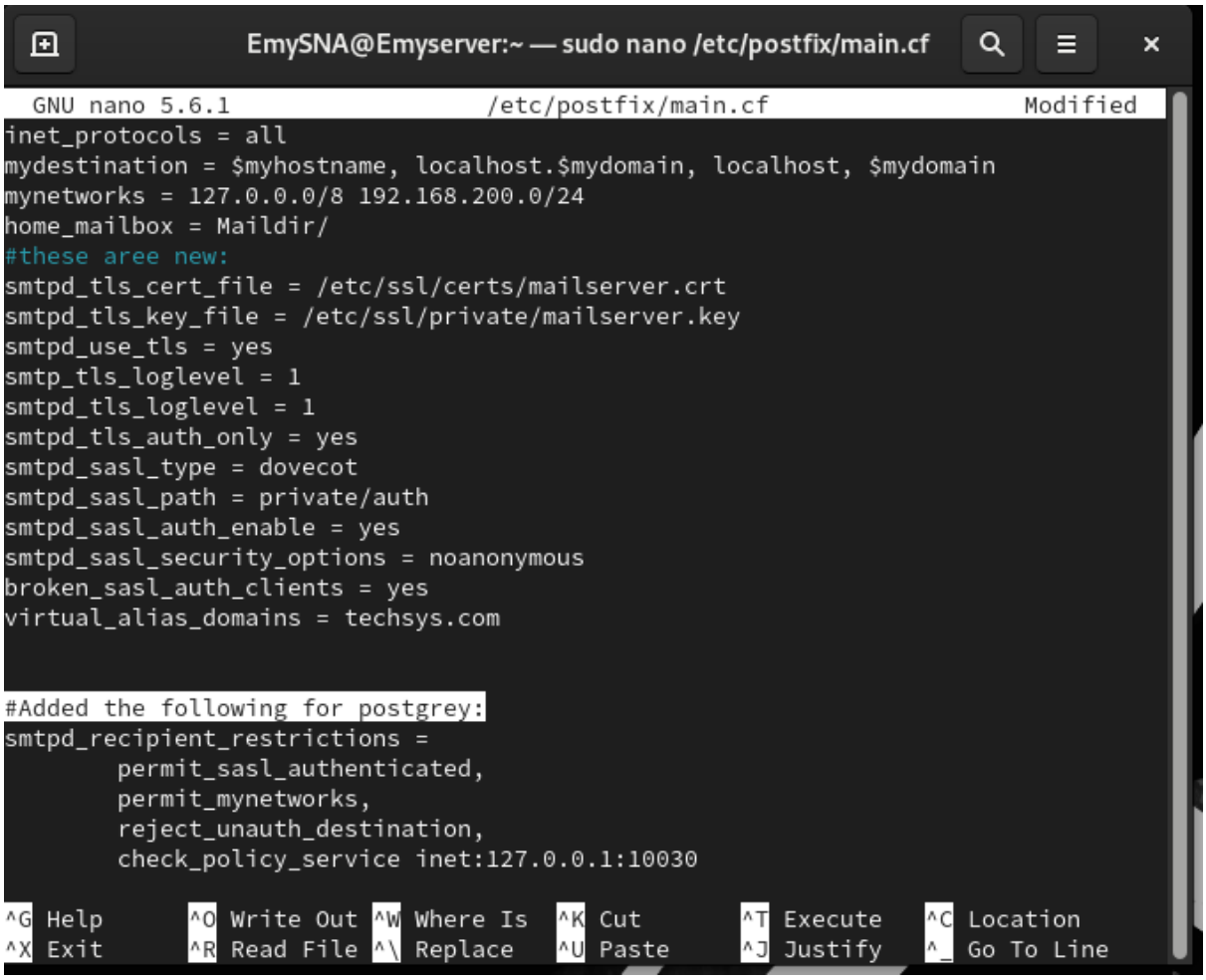
Next postgrey must be configured so we add it's configuration inside postfix/main.cf file:



The last lines in the following picture are add:

smtpd_recipient_restrictions (Defines a set of rules for handling incoming email) permit_sasl_authenticated(Allows email from authenticated users) permit_mynetworks (Allows email from trusted networks) reject_unauth_destination (rejects email if the destination is not authorized) check_policy_service inet:127.0.0.1:10030(This is used for

spam filtering or greylisting.)



smtpd_recipient_restrictions = (Defines the rules that are used to decide if incoming mail is accepted or rejected)

permit_sasl_authenticated (mail from SASL authenticated users is accepted)

permit_mynetworks (mail from IP addresses or subnets defined in “mynetwork” in “mail.cf” is accepted)

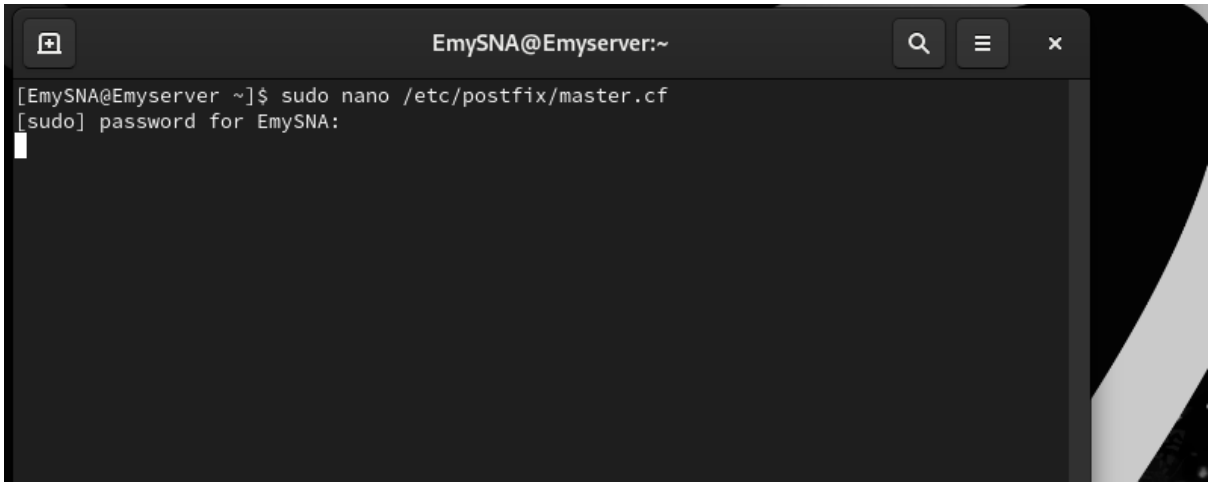
reject_unauth_destination (mail from IP addresses or subnets not defined or authorized in “mynetwork” in “mail.cf” is rejected)

check_policy_service inet:127.0.0.1:10030 (postfix socket specification tells postfix to connect to a policy server such as postgrey, that listens on port 10030, running on the same machine)

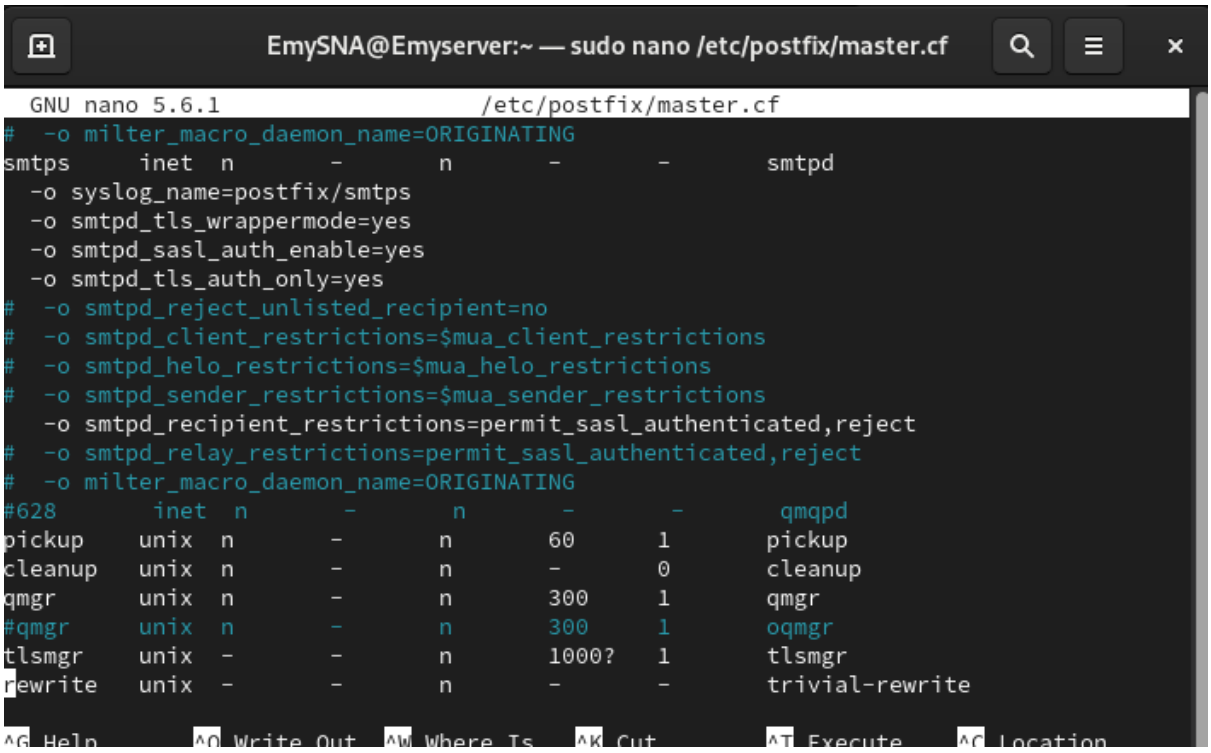
Troubleshooting:

Troubleshooting, issue 1:

First open master.cf file:



Uncomment/add the following lines:

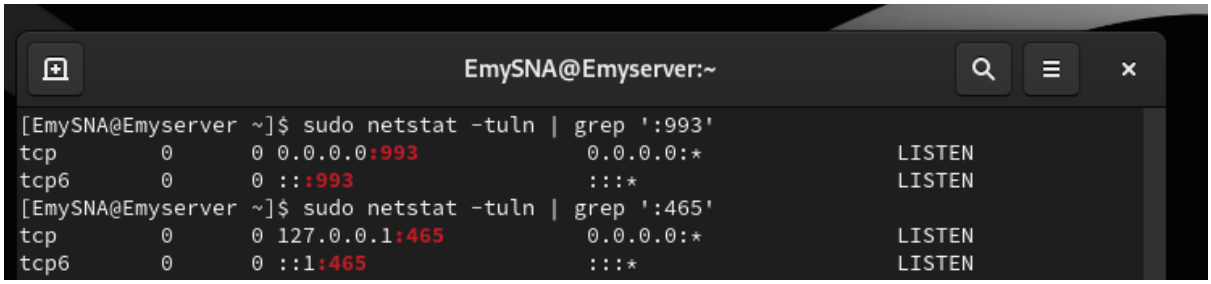


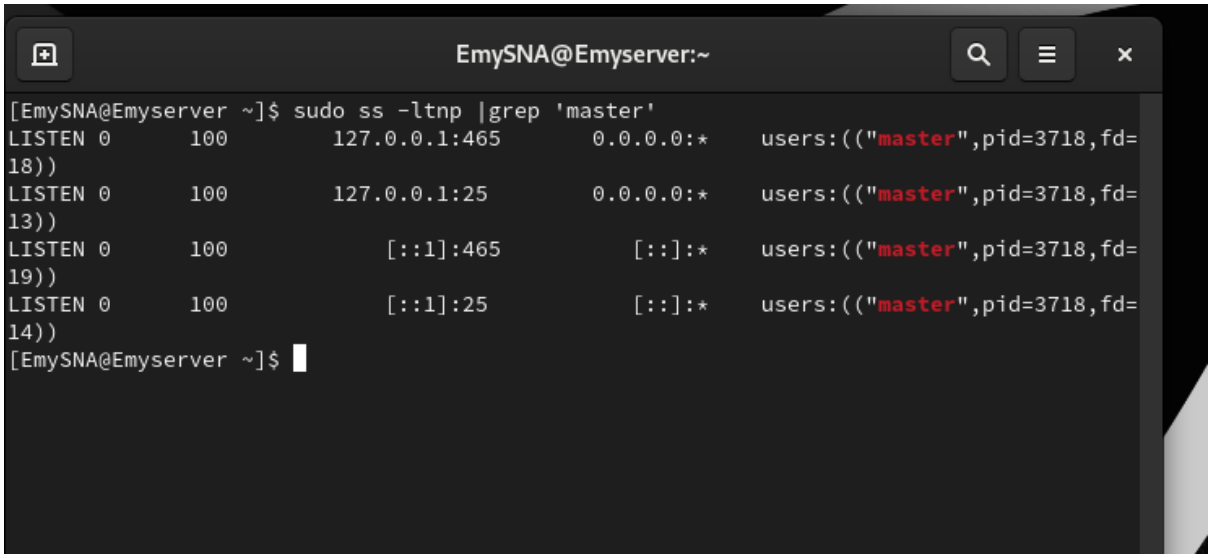
Uncommenting the first 4 lines in the picture to allow SMTP server to listen for encrypted connections on port 465 (SMTPS) over SSL/TLS to ensure secure communication between server and clients

then added “-o smtpd_tls_auth_only=yes” to make postfix use TLS for authentication and “permit_sasl_authenticated” after “-o smtpd_recipient_restrictions=” to enable only authenticated users can send emails through the postfix server

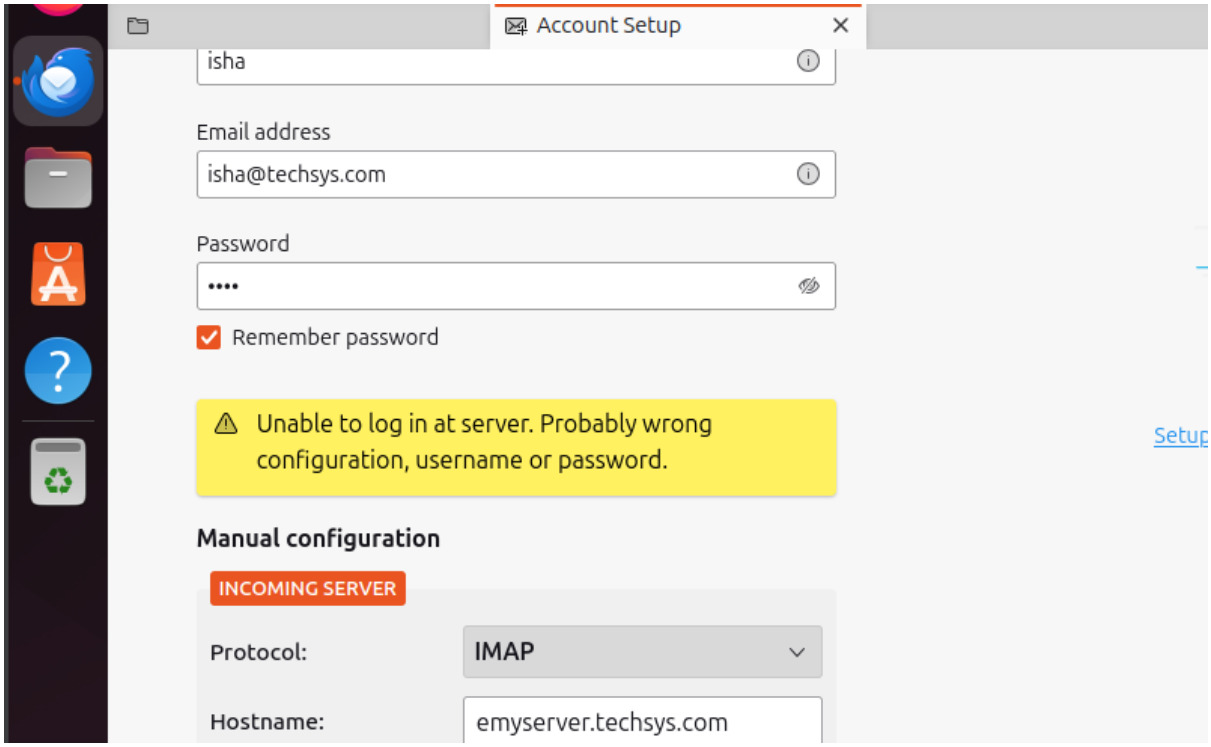
we then configure the firewall to allow secure traffic (SMTP/SMTPS) which sends and receives mail over SSL/TLS, make it permanent then reload it to apply changes and restart it to apply new settings:

Ensure all ports are waiting for a connection and listening:

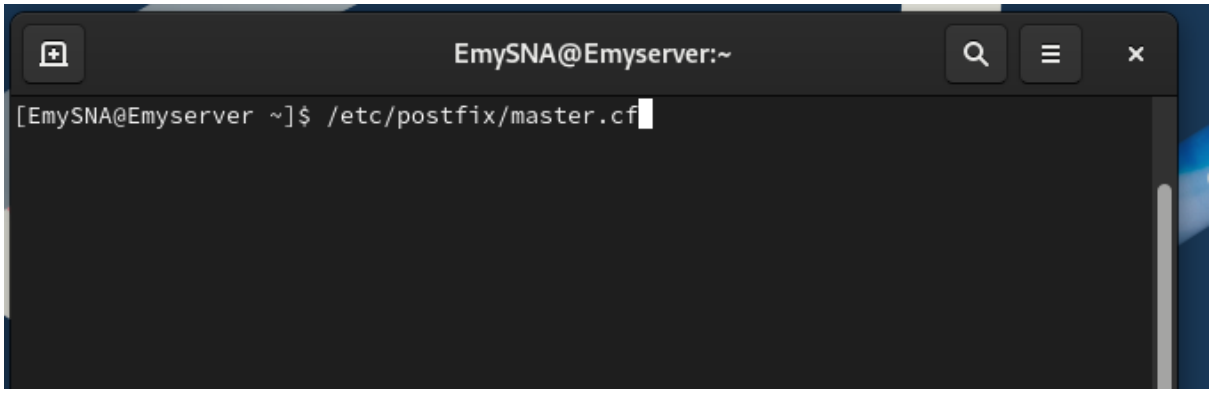


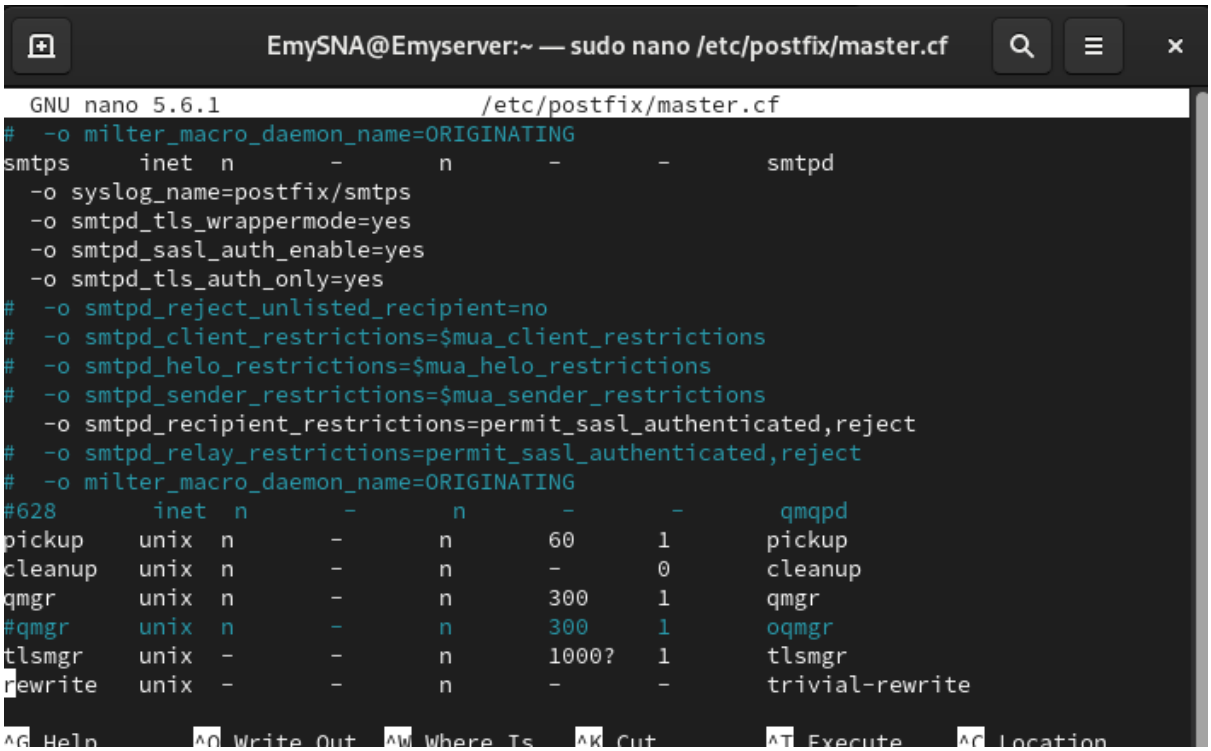


As the port listen to local host “127” I must check what’s binding it, as keeping it binded will make my thunderbird unable to properly connect with the mail server like this, so I must back track to fix it

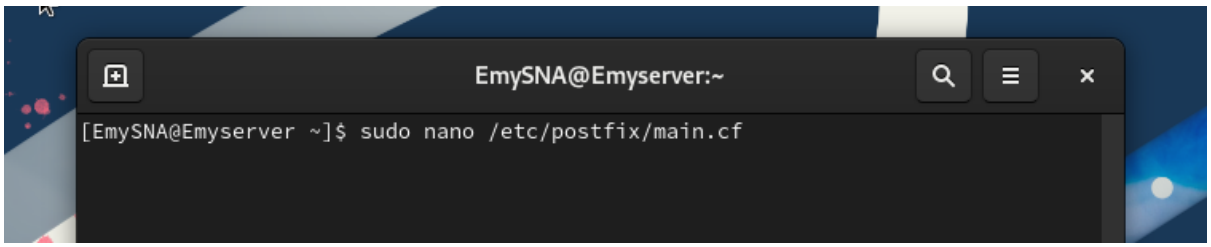


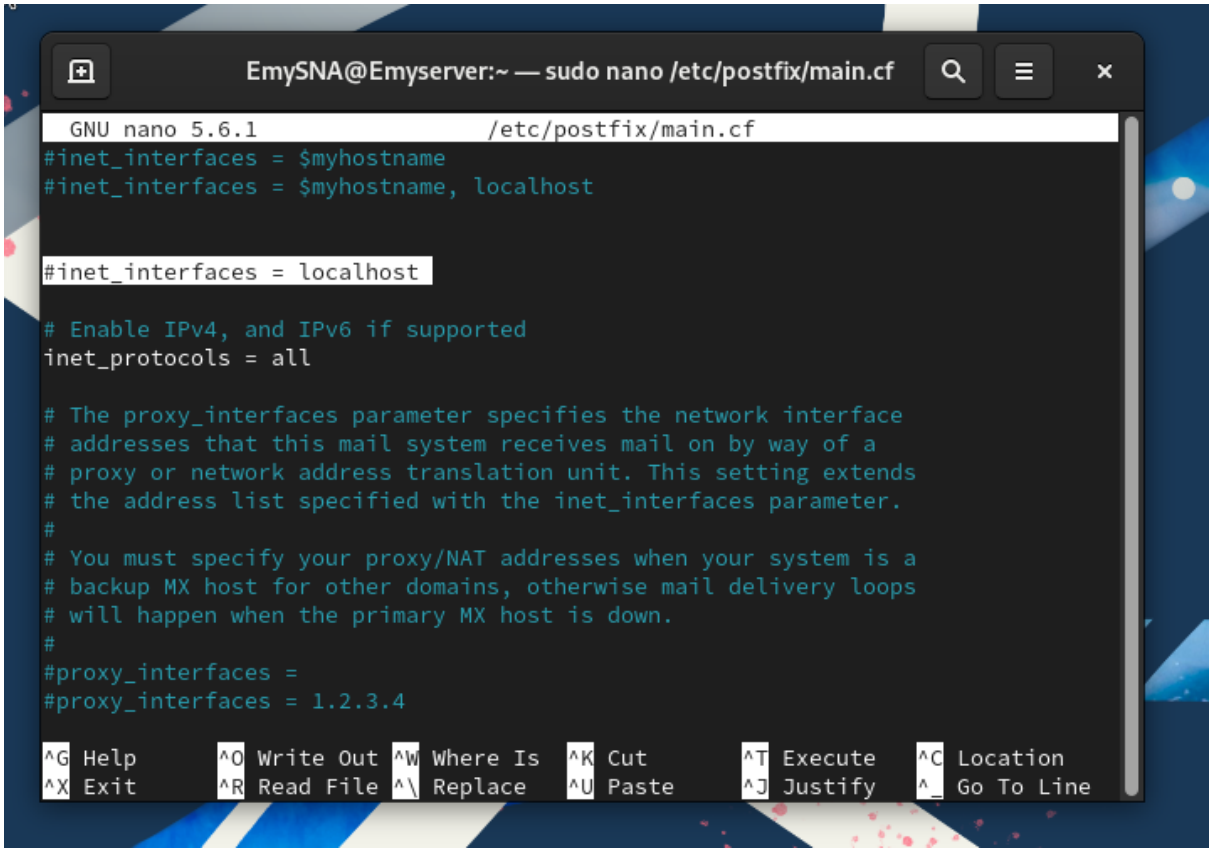
“/etc/postfic/master.cf” is checked and nothing is binding postfix to localhost or directly “127.0.0.1” :





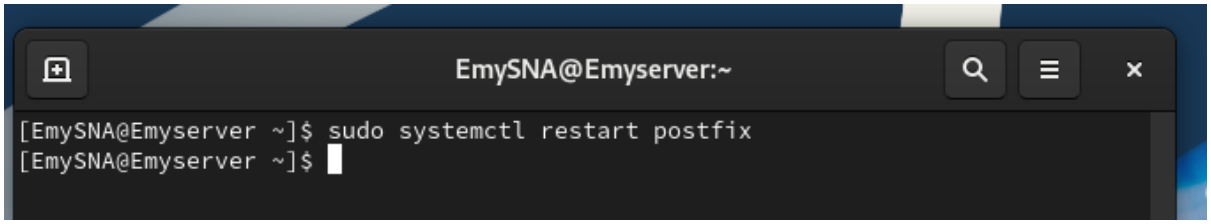
So next we check “main.cf” :



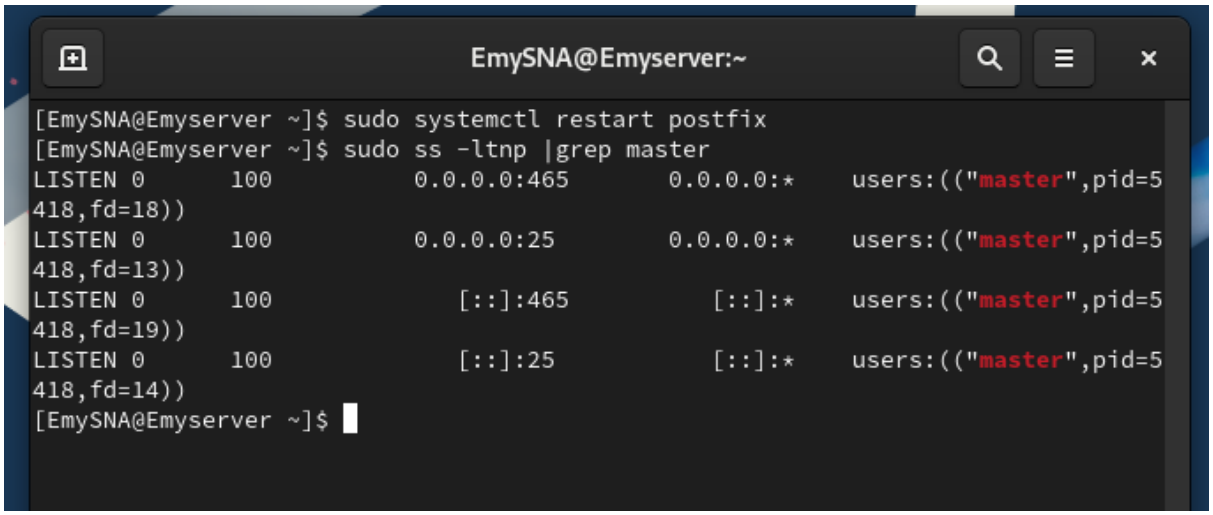


“inet_interfaces = localhost” was uncommented so we comment it un-bind it from port from listening to local host only.

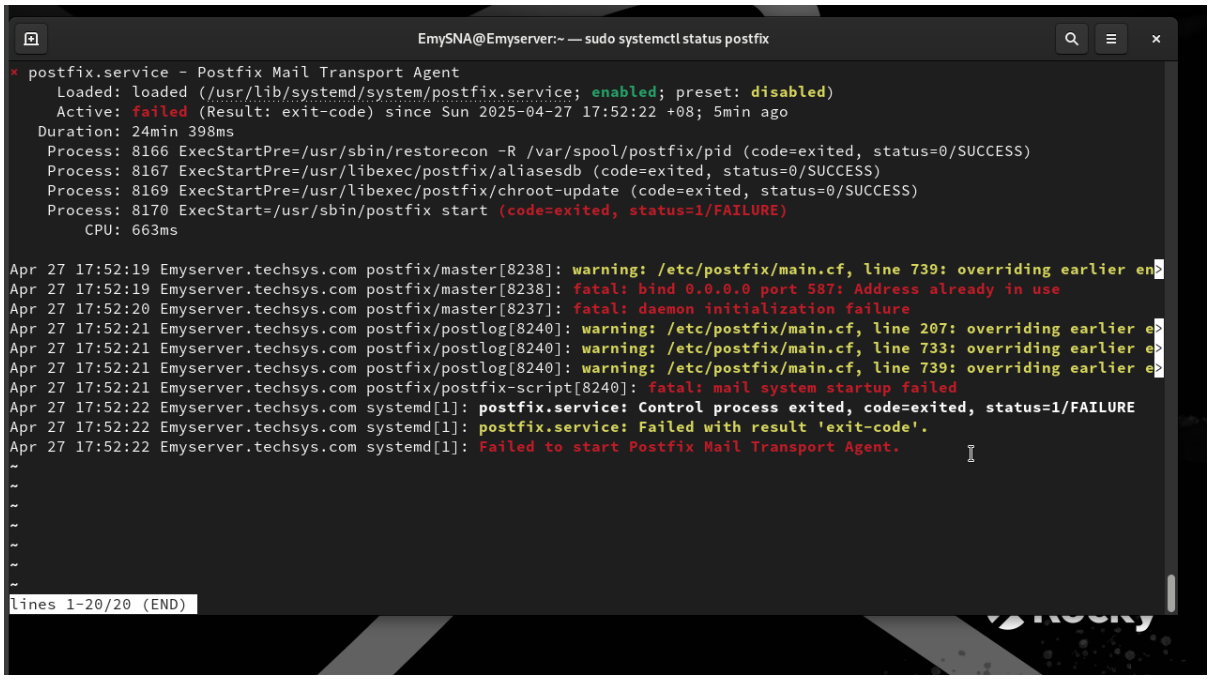
Then we save the new buffer before restarting postfix to activate new settings:



checked again and surely now it’s accepting both public and private Ips:

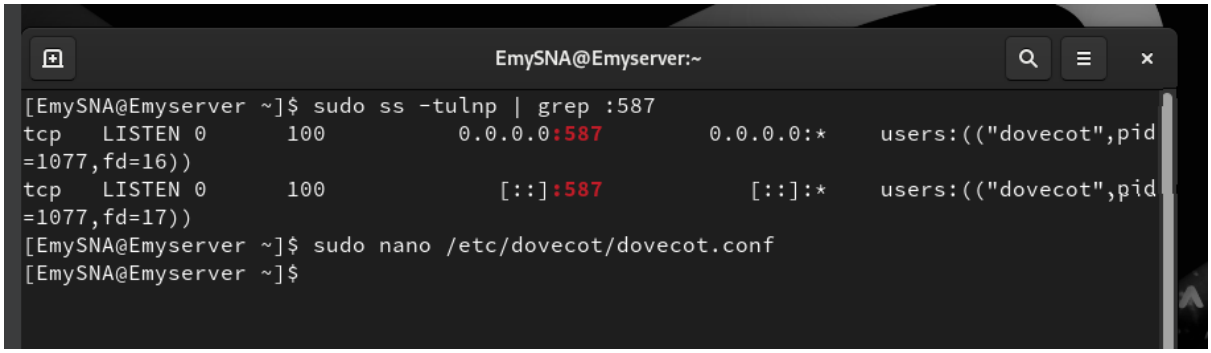


However, I couldn’t restart postfix earlier after some of the modifications

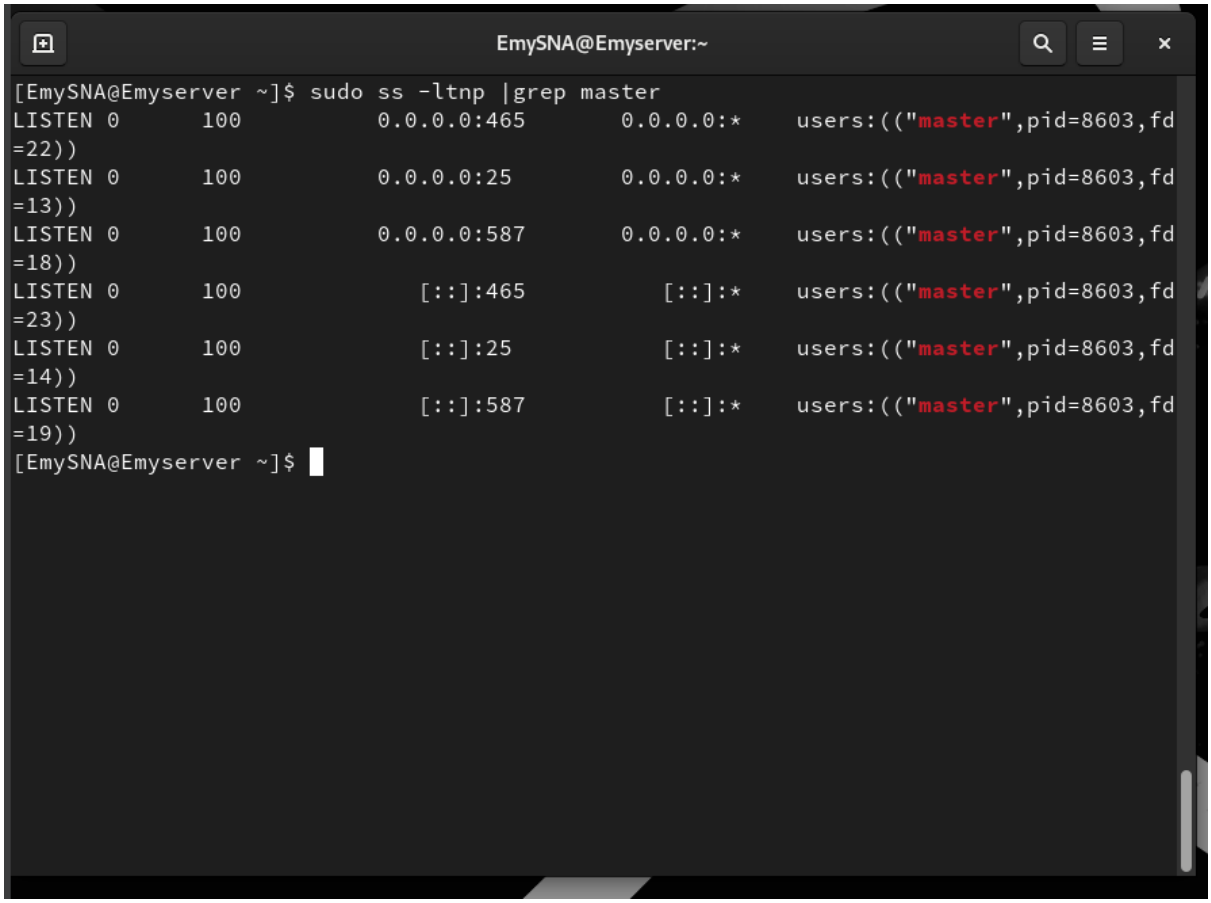


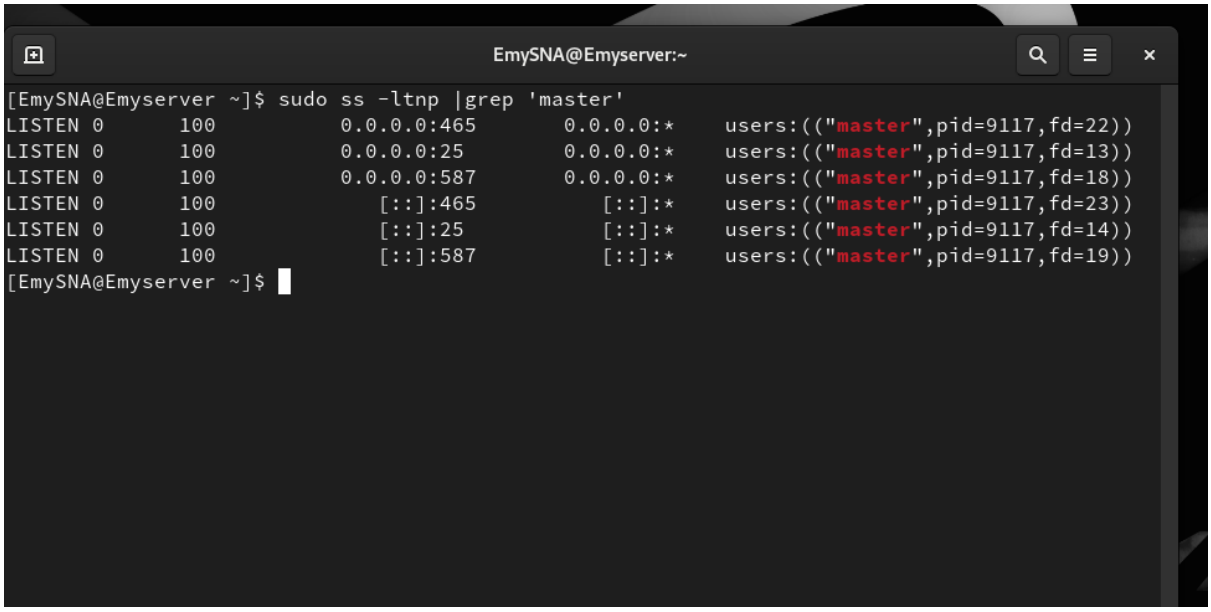
The problem was that postfix can’t bind to port 587 because dovecot already took it, so to disable dovecot listening on 587 I opened “10-master.conf” file and change the “port = 587” to 0:

Then restart dovecot and check who’s using it:

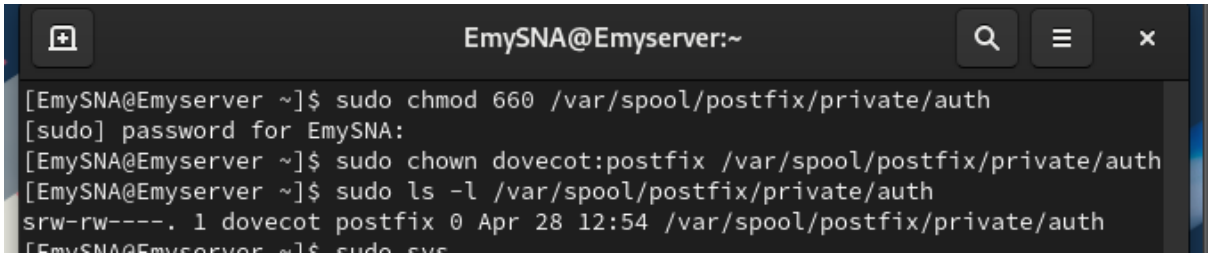


And check the rest just to as a safe measure:

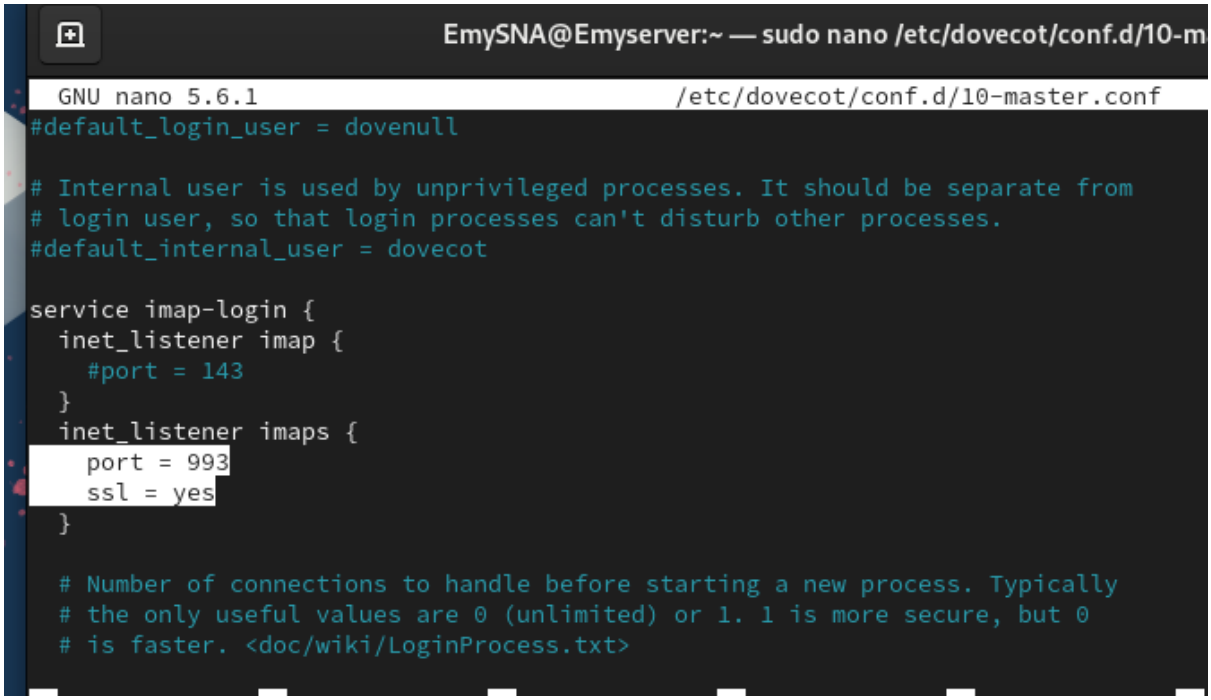




After trying to log in again the issue persisted so auth socket in dovecot permissions are changed so only the owner (dovecot) and group(postfix) can read and write it to communicate and verify the usernames and passwords (SASL auth), then change ownership of socket to dovecot since it created and group to postfix so it can access it to allow authenticated user sending mails:

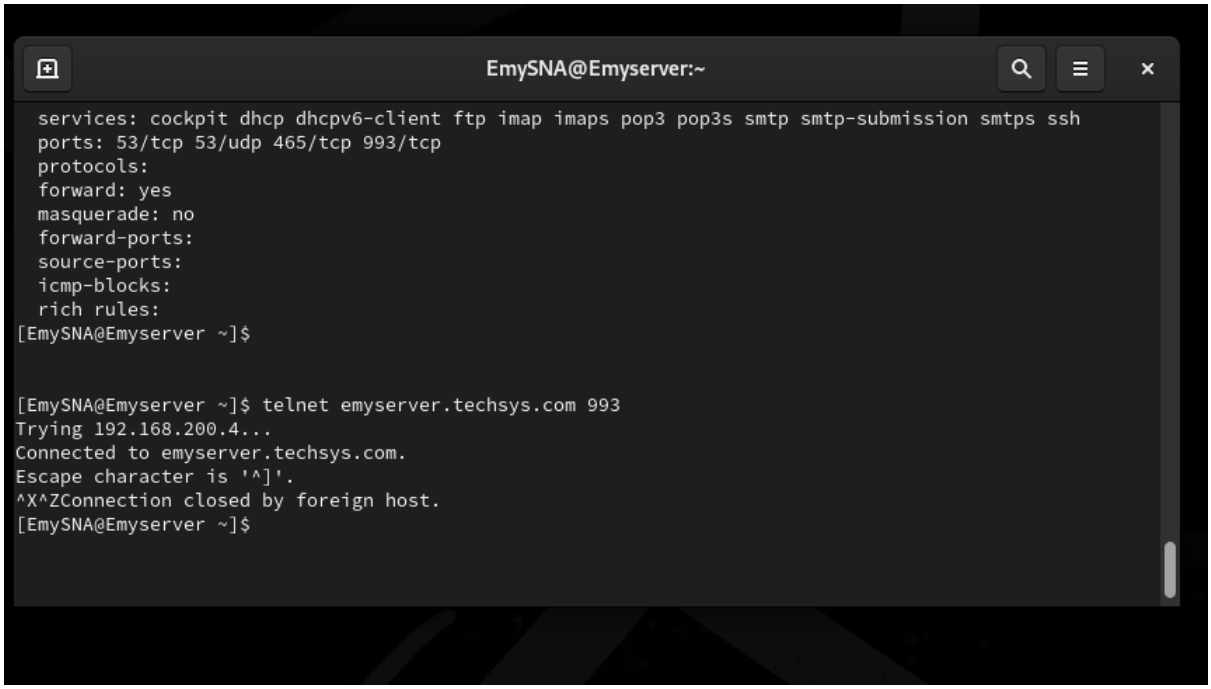


Uncomment the following two lines to enable dovecot to listen on port 993 for encrypted IMAPS for secure email retrieval to enable clients to securely receive their mail over SSL/TLS

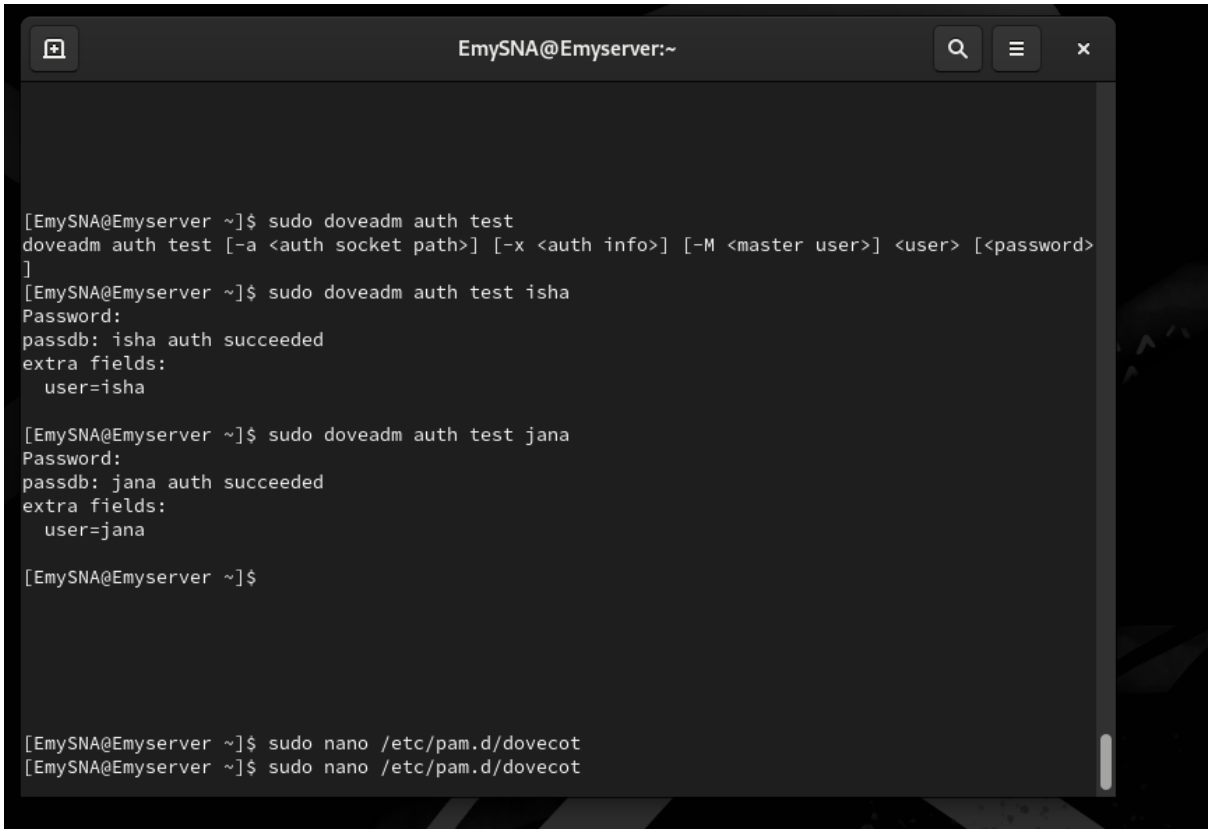


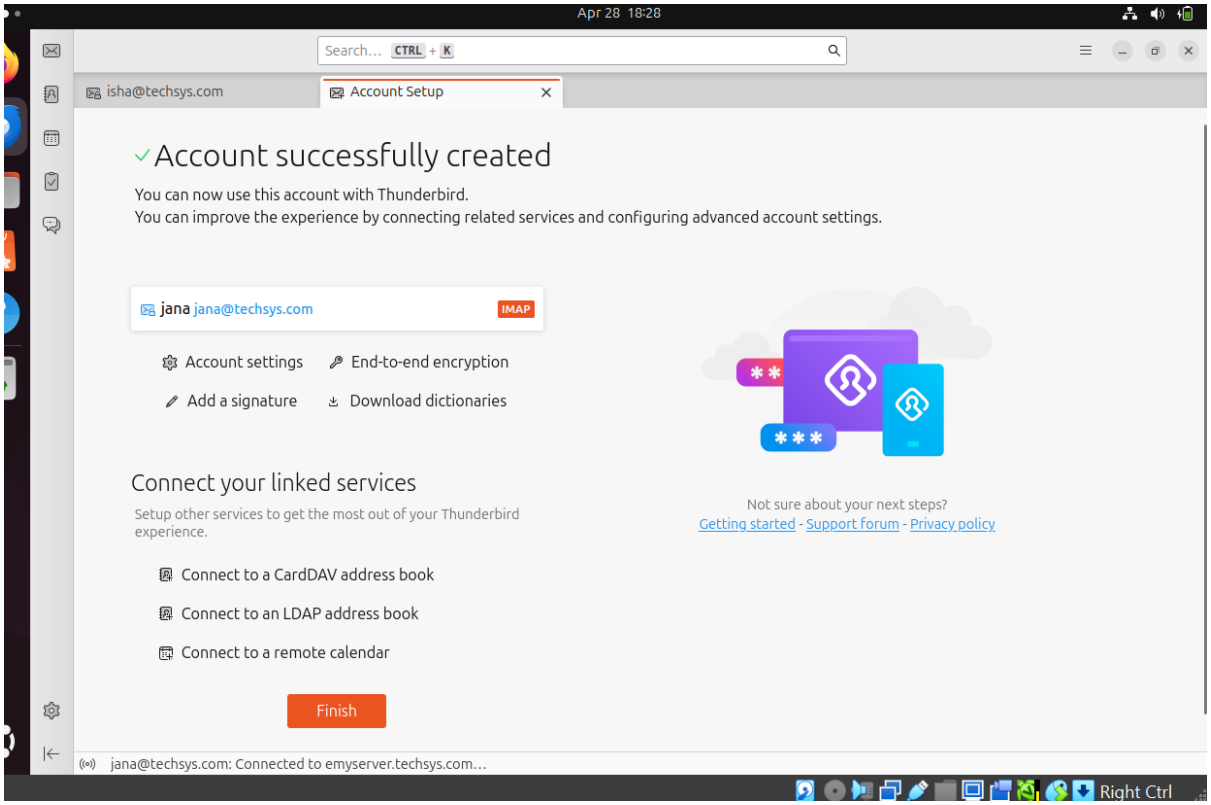
Then run "sudo firewall-cmd --list-all" to view all current firewall setting to check ports like 993 and 465 aren't blocked.

After that I tried to connect to port 993 using Telnet to test if IMAPS port is reachable and dovecot is listening, not being blocked by the firewall



Final to ensure that users can log in, dovecot's own auth system is used to test it to make sure authentication is working before trying from the actual mail client:

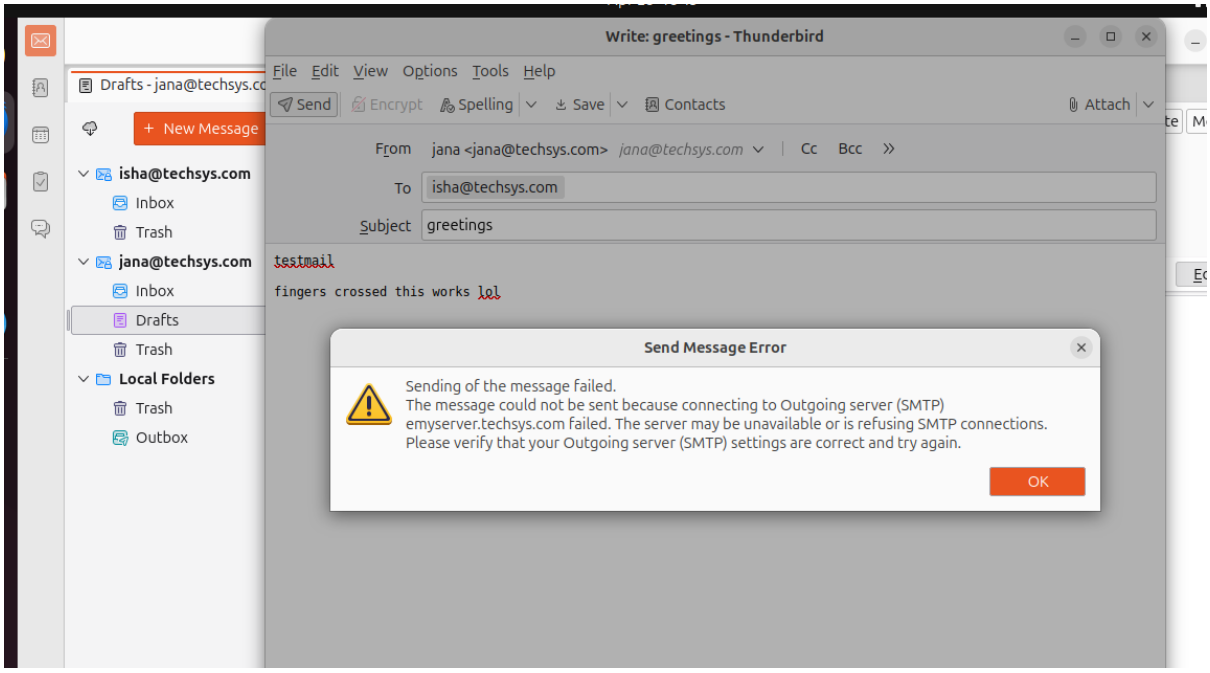




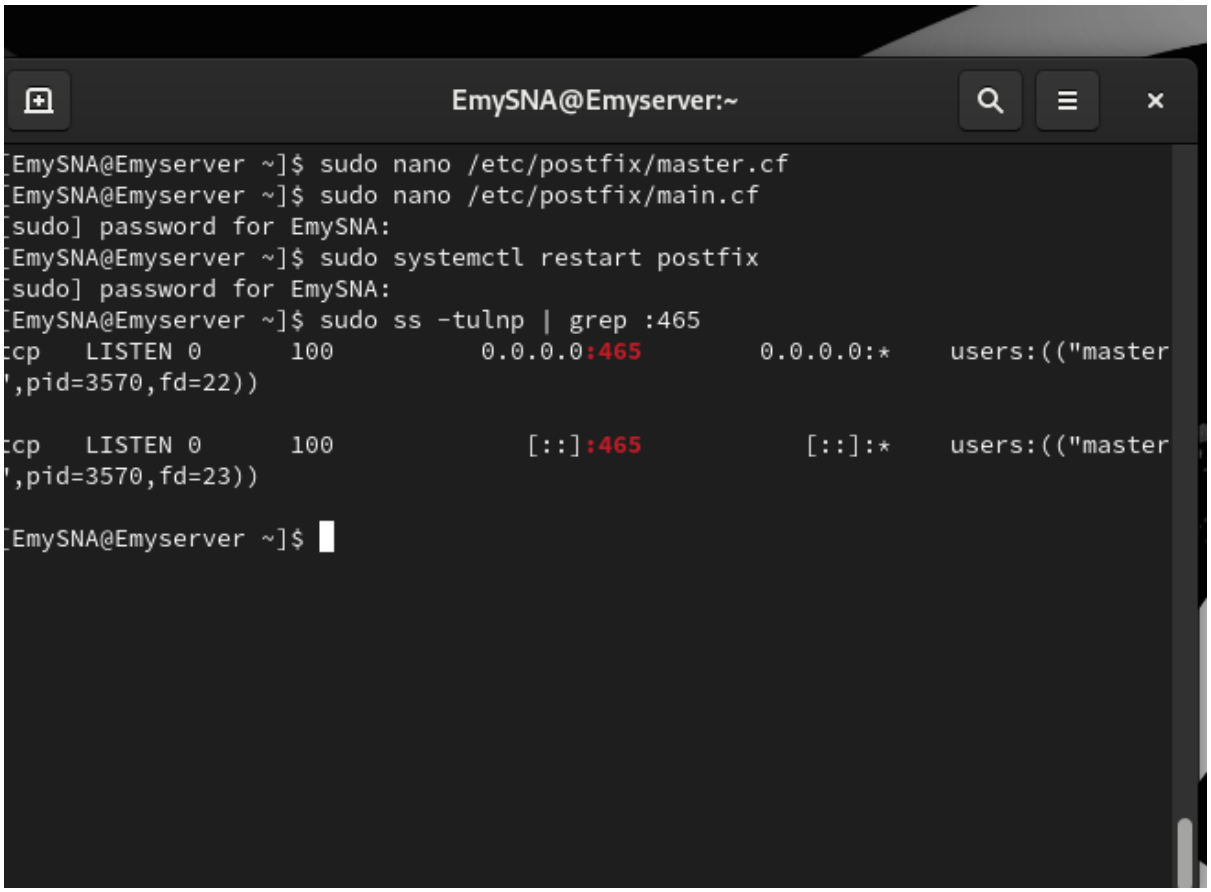
Which enabled me to log in and connect to the server.

Troubleshooting, issue 2:

The next problem I had to troubleshoot is that the client was unable to send emails because connecting to my outgoing server (SMTP) failed.



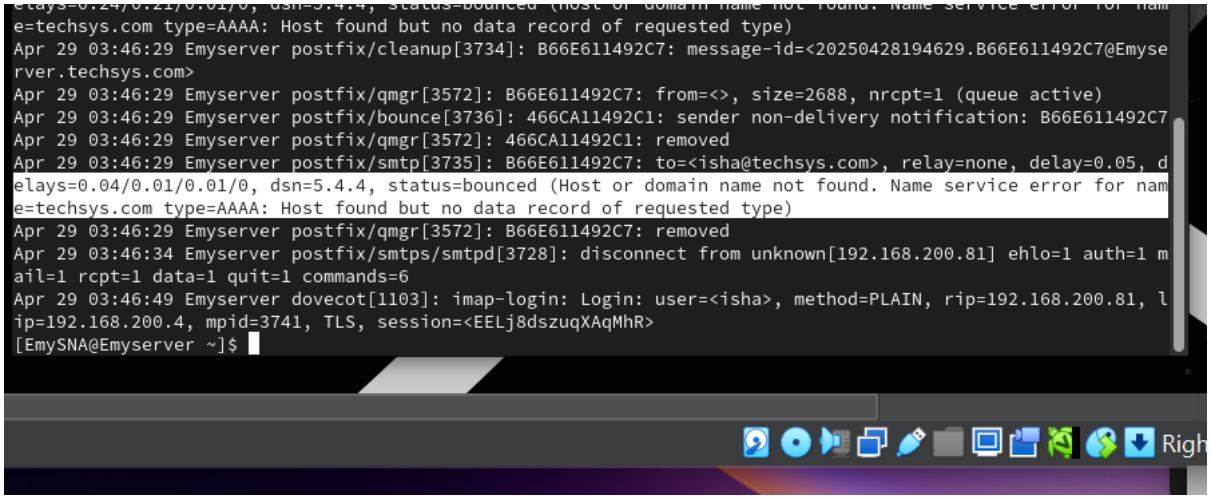
I double checked the “/etc/postfix/main.cf” to double-check configuration is correct with typos and it was I then double-checked postfix ports connection, and it wasn’t set to accept only Rocky’s local hosts as it previously was



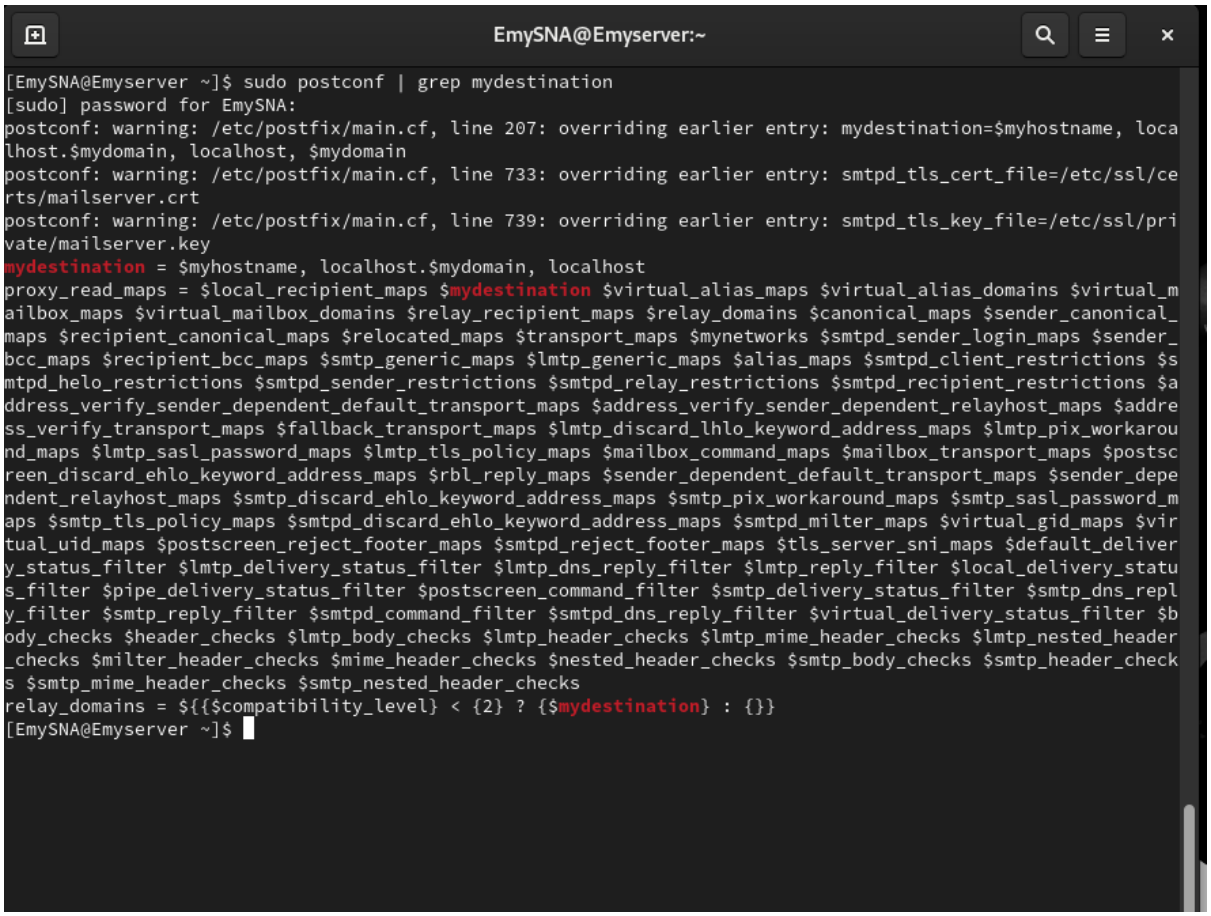
So, I ran “openssl s_client -connect emyserver.techsys.com:465 -crlf” to connect the mail server on port 465 using SSL/TLS to test SMTP commands.

The

The error shows the “status=bounced” which means it was sent but couldn’t be delivered/received and sent back to sender :

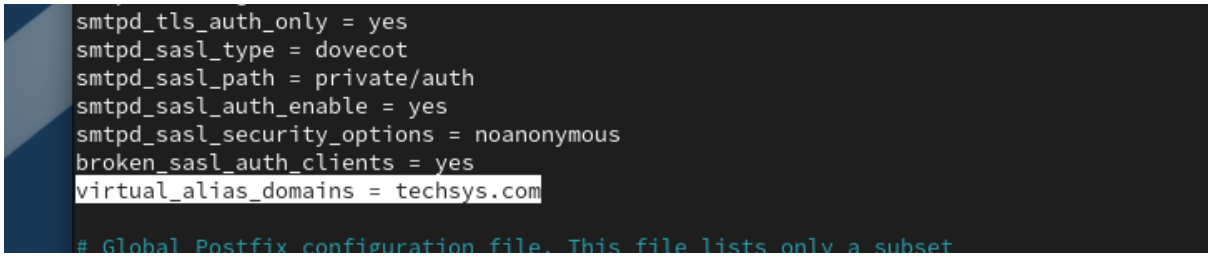


Then I ran “sudo postconf | grep mydestination” to check “mydestination” values to ensure the server identifies the proper domains that should receive the mail:



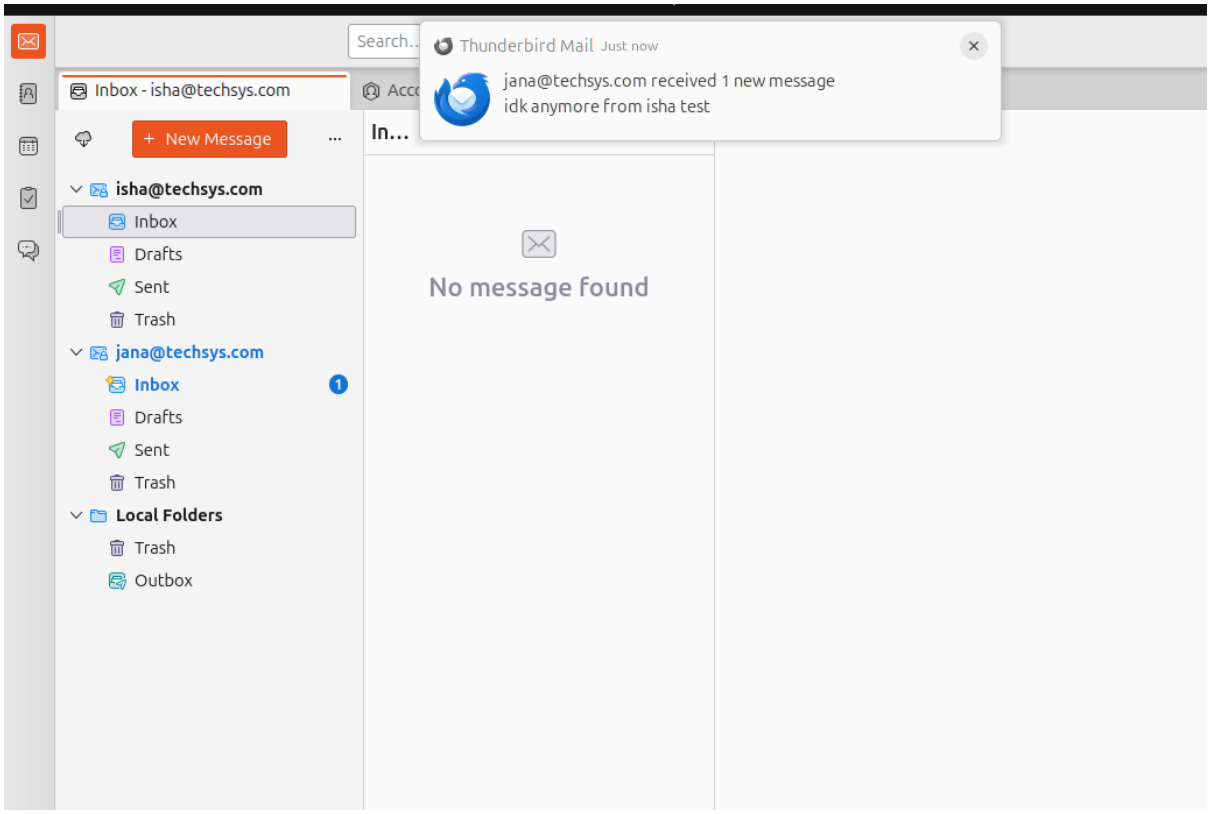
And based on the first few lines there has been an overriding.

So to have postfix direct mail to respective system user “virtual_alias_domains = techsys.com” must be added to “main.cf” so postfix treats “techsys.com” as local:



Then we add “sudo postconf | grep mydestination” to point to postfix the “/etc/postfix/virtual” file that has the virtual aliases of users so postfix knows which user to deliver mail to when giving an Email account:

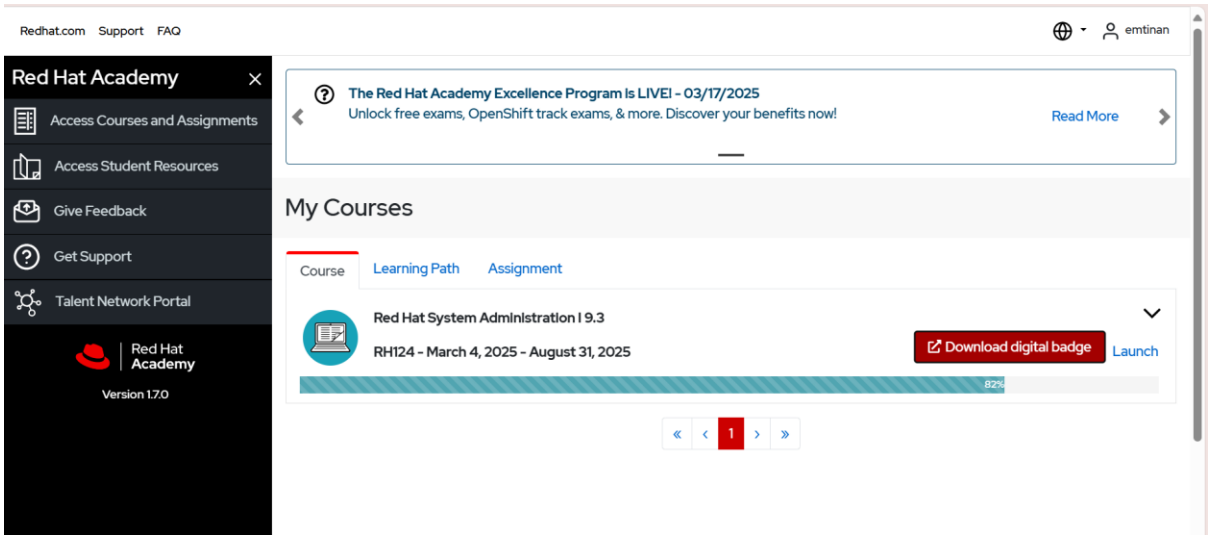
And now mails are properly sent:



Thunderbird now works and receives emails normally.

RedHat Academy Certificate/Progress:

Certificate wasn’t given upon course completion, but a badge was given:



RedHat Academy course completion badge:

Red Hat System Administration I (RH124 - RHA) - Ver. 9.3

ISSUED TO

Emtinan Ahmed

Issued on: 09 MAY 2025 | Issued by: Red Hat
Verify: https://www.credly.com/go/mOQBPTN9