

Table Of Contents

1.0 Introduction.....	6
Installation:.....	6
Rocky:.....	6
Ubuntu Installation.....	21
1. Creating the Ubuntu VM:.....	22
2. Installing Ubuntu:.....	24
3. Installation Options:.....	25
4. Choosing Installation Type:.....	25
5. Location and User Setup:.....	26
6. Completing the Installation:.....	27
Post-Installation and User Setup in Ubuntu.....	28
1. Logging In:.....	28
2. Skipping Unnecessary Steps:.....	28
Updating the System in Ubuntu.....	30
1. Opening the Terminal:.....	30
Ensuring 'sudo' Access.....	30
1. Modifying the 'sudoers' File:.....	30
2. Verifying the Update Process:.....	31
3. Exiting Root User:.....	31
Updating Software via the GUI.....	31
1. Updating Installed Programs:.....	31
Troubleshooting:.....	32
Error 1: Cannot install the best update candidate.....	32
Error 2: Troubleshooting Administrative Access.....	33
2.0 Setting up NAT Network.....	35
1. Checking IP Address on Ubuntu Client:.....	35
Installing Net-Tools.....	35
2. Checking IP Address on Rocky Server:.....	36
3. Creating a NAT Network in VirtualBox:.....	36
4. Assigning NAT Network to VMs:.....	37
5. Verifying IP Addresses:.....	37
3.0 Setting hostnames for both VMs.....	38
Setting Ubuntu Hostname:.....	38
2. Updating the Hosts File in Ubuntu:.....	38
3. Reboot and Test Hostname in Ubuntu:.....	39
Setting Rocky Server Hostname:.....	39
1. Setting Rocky Linux Hostname:.....	39

4.0 Setting Static IP Addresses:	41
Rocky:	41
1. Verify Operating System:	41
2. Update and Upgrade the System:	41
3. Check and Enable NetworkManager:	42
4. Check Current Network Configuration:	42
5. Check System Network Information:	43
6. Identify Network Connection Name:	43
6. Configure Static IP Address:	43
8. Restart NetworkManager:	44
9. Verify IP Configuration:	44
10. Additional Network Checks:	45
Ubuntu:	45
1. Update and Upgrade the System:	45
2. Check and Install NetworkManager:	46
3. Check the Current IP Address:	47
4. Check Network and Device Information:	47
5. Identify the Network Connection Name:	47
5.0 DNS	48
DNS Configuration in Rocky Linux	48
1. Installing DNS (BIND9):	48
2. Enable DNS to Start Automatically:	48
4. Edit the 'named.conf' File:	50
Configure DNS Settings	50
Add DNS Forwarders:	50
Configure DNS Zones:	51
Check Configuration for Syntax Errors:	51
Create Forward and Reverse Zone Files:	51
Create Reverse Zone File:	52
Restart the DNS Service:	52
Allow DNS Traffic Through the Firewall:	53
Set DNS in Resolv.conf:	53
Testing DNS:	54
DNS configuration in Ubuntu Client	55
Access Network Settings	55
1. Open Network Settings:	55
2. Configure DNS Addresses:	55
Install resolvconf	56
1. Open a Terminal:	56
2. Install resolvconf:	56
3. Configure resolvconf	56

Update DNS Resolver Configuration:.....	57
4. Modify NetworkManager.conf.....	57
Restart NetworkManager:.....	57
Verify DNS Configuration.....	57
TROUBLESHOOTING:.....	59
Error 1: Job for named.service failed.....	59
Error 2: server can't find vegeta.rockyserver.org: NXDOMAIN.....	61
6.0 DHCP Configuration on Rocky Linux.....	63
1. Install the DHCP Server.....	63
1. Open a Terminal on Rocky Linux:.....	63
2. Install DHCP Server Package:.....	63
2. Create a DHCP Configuration File.....	63
1. Create and Edit the DHCP Configuration File:.....	63
2. Setup DHCP Configuration:.....	64
3. Enable and Start the DHCP Service.....	64
4. Configure the Firewall.....	65
5. Update VirtualBox Network Settings.....	65
Verify Network Adapter and Connection ID.....	66
Check Network Connection Settings:.....	67
Troubleshooting:.....	69
Error 1:.....	69
7.0 Email Server:.....	70
Postfix Installation and Configuration:.....	70
Dovecot Installation and Configuration:.....	73
Thunderbird (Ubuntu Client):.....	75
Troubleshooting:.....	79
Error 1: Thunderbird failed to find the settings for the email account.....	79
Fixing the MX and A Records in DNS:.....	79
Firewall Configuration for Mail Services:.....	81
Error 2: Protocol Connection is unavailable.....	82
Error 3: Thunderbird was unable to match the User or Password to the ones stored in the Rocky User Database.....	83
Issue: Authentication Failure for User Kashave.....	83
Error 4: Thunderbird does not trust the self-signed certificate on Rocky Postfix.....	85
Steps to Export the Self-Signed Certificate:.....	85
Importing the Certificate in Thunderbird:.....	87
Error 5: Sasl Authentication not allowing thunderbird to verify the user password from rocky:.....	89
Solution: Install and Configure saslauthd on Rocky Linux.....	89
Ensuring Correct saslauthd Directory Configuration.....	90
Adding Postfix to saslauthd Group.....	90

Configuring Dovecot for SASL.....	91
8.0 Apache Webserver Setup.....	92
1. Checking for System Updates.....	92
2. Installing Apache (httpd).....	93
3. Starting and Enabling Apache Service.....	93
4. Configuring the Firewall for HTTP and HTTPS.....	94
5. Testing Apache HTTP Server.....	95
Configuring an Apache Virtual Host.....	96
1. Checking Server Information.....	96
2. Backing Up Apache Configuration Files.....	96
3. Navigating to the Apache Sites Directory.....	96
4. Creating a New Virtual Host File.....	96
5. Creating and Setting Permissions for the Website Directory.....	97
6. Adding Basic HTML to the Website Directory.....	97
7. Checking Apache Configuration and Restarting the Service.....	98
Testing the Apache Virtual Host.....	98
Testing on Rocky Linux.....	98
Testing on Ubuntu Client.....	100
9.0 SSL/TLS Configuration.....	101
1. Generate a New Private Key.....	101
2. Generate a Certificate Signing Request (CSR).....	101
3. Generate a Self-Signed Certificate.....	102
Update Postfix Configuration.....	102
Set SELinux Contexts for New Files.....	103
Configure Dovecot for SSL/TLS.....	104
Configure Apache for SSL/TLS.....	106
8. Test SSL/TLS Encryption.....	107
Testing Postfix SSL/TLS:.....	107
○ On Rocky Server:.....	107
Test On Ubuntu Client:.....	110
Testing Apache SSL/TLS:.....	111
● On Rocky Server:.....	111
Ubuntu Client:.....	113
Command-Line Testing:.....	115
● On Rocky Server:.....	115
On Ubuntu Client:.....	117
Troubleshooting SSL/TLS Configuration:.....	119
Verify Certificate File Paths.....	119
Update Dovecot Configuration.....	119
Update Postfix Configuration.....	120
Log Analysis.....	121

Verify SSL/TLS Configuration.....	121
Check Logs.....	123
Remove Private Key Password (Optional).....	123
Verify File Permissions.....	123
Restart Postfix.....	125
10.0 Additional Features:.....	126
10.1 Network File System (NFS) Configuration.....	126
Install NFS Server.....	126
Create Shared NFS Directory.....	126
Set Directory Ownership and Permissions.....	126
Set Directory Ownership.....	127
Set Directory Permissions.....	127
Recursively Set Permissions.....	127
Configure NFS Exports.....	127
Export the NFS Directory.....	128
Restart the NFS Server.....	128
Configure the Firewall.....	128
Enable the Firewall.....	128
Verify Firewall Status.....	129
NFS Client Configuration (Rocky Linux):.....	129
Install NFS Client.....	129
Create a Mount Point.....	130
Mount the NFS Share.....	130
Verify NFS Mount.....	130
Test Write Access.....	131
Verify on the NFS Server.....	131
Unmount and Remount NFS Share.....	131
Unmount the NFS Share.....	131
Verify Unmount.....	132
Remount the NFS Share.....	132
Verify the Remount:.....	132
Check the Contents of the Mounted Directory.....	132
Test Writing to the NFS Share After Remount.....	132
Confirm Test Writing:.....	133
Final Checks.....	133
Check Firewall on the Server.....	133
Check Logs for NFS Issues.....	133
10.2 Configuring SASL for Dovecot and Postfix.....	135
Ensuring Correct saslauthd Directory Configuration.....	135
Adding Postfix to saslauthd Group.....	136
Configuring Dovecot for SASL.....	137

1.0 Introduction

This project involves setting up a network administration environment using Rocky Linux and Ubuntu within VirtualBox. The main objective is to integrate essential services such as DNS, DHCP, email, web, and SSL/TLS encryption to create a fully functional and secure system suitable for a small organization. By leveraging both Rocky Linux and Ubuntu, the project demonstrates versatility in managing network services across different Linux distributions. Each component, from the installation of the operating systems to the configuration of services, will be documented thoroughly, providing insights into the process and overcoming any obstacles encountered.

Installation:

Rocky:

First, start the virtual box. Click the ‘machine’ dropdown menu and click on the “New...”. When done, the below will pop up. Ensure that

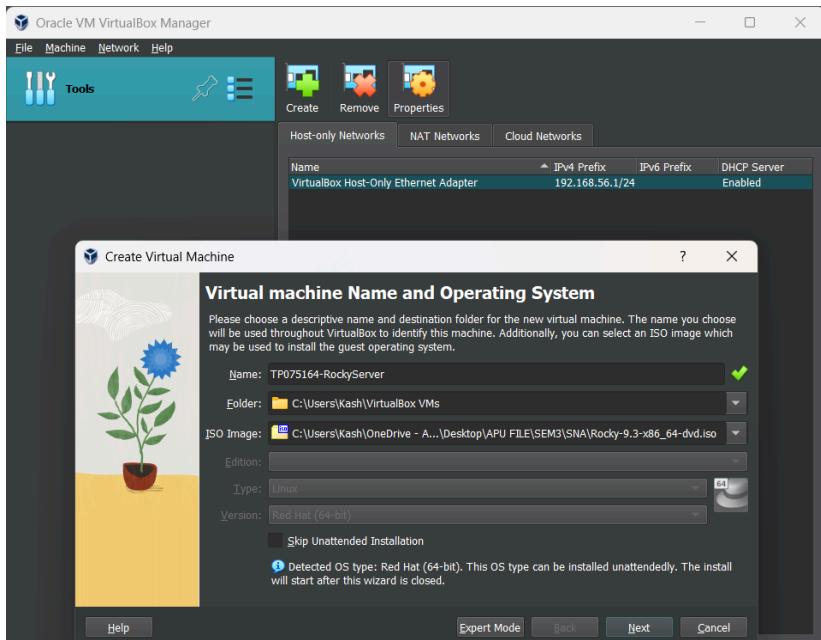
- ‘type’ is set to ‘Linux’
- ‘Version’ is set to ‘Red Hat (64-bits)’
- ‘Name’ of your choosing
- ‘ISO Image’ is the ISO file you downloaded

The name of my virtual Machine is ‘TP075164-RockyServer’ to allow for easier differentiating between the machines later on.

On the following page, fill in your username, along with your password. Be sure to remember it! Also ensure that the ‘Guest Additions’ checkbox has been clicked.

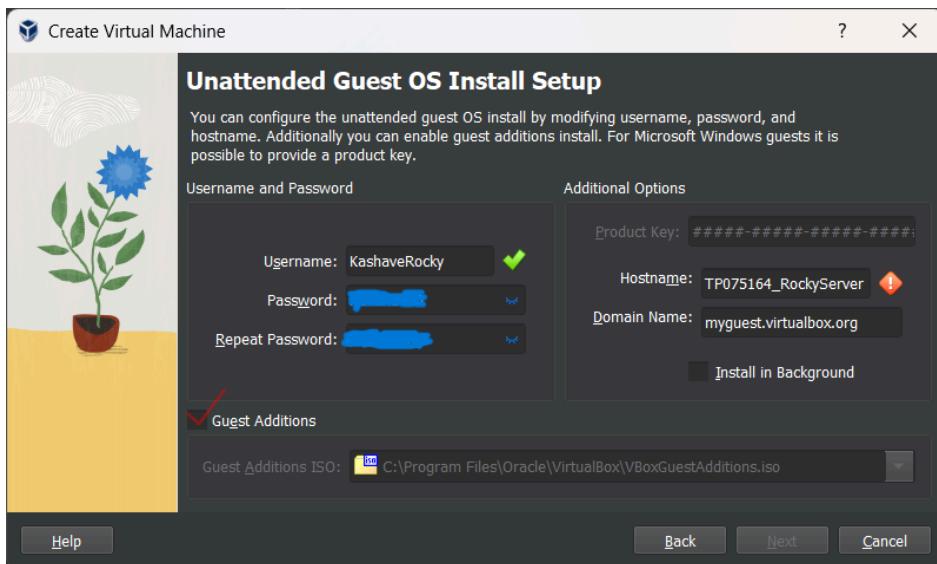
‘Guest Additions’ are software components installed inside a virtual machine to enhance its capabilities and make it function more like a physical computer. They bridge the gap between the virtual machine and the host operating system, offering features such as:

- **Shared Folders:** For easy file transfer between the host and guest operating systems.
- **Improved Mouse Integration:** For a smoother and more responsive mouse experience within the virtual machine.
- **Shared Clipboard:** To copy and paste text or files between the host and guest operating systems.



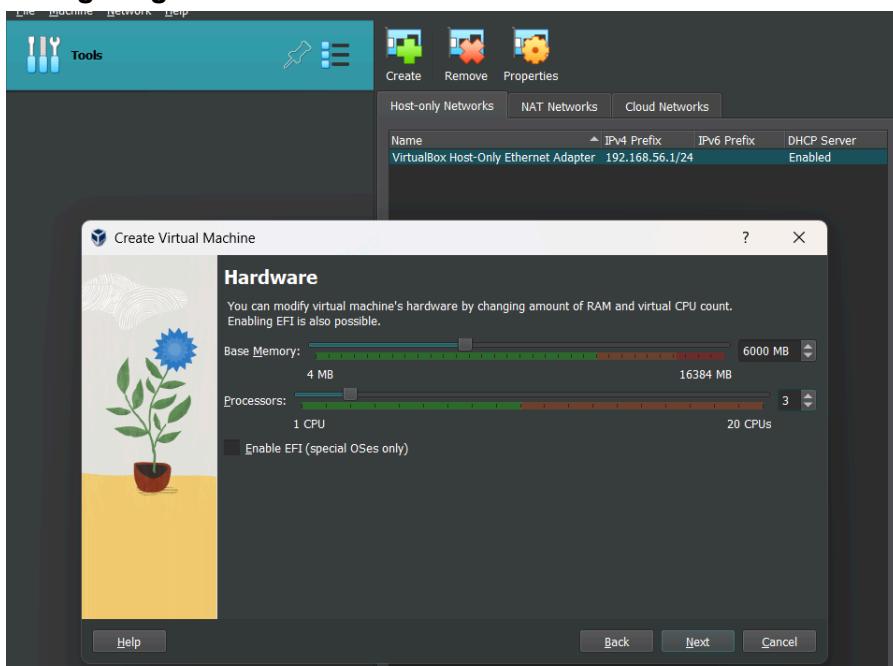
Creating the Virtual Machine:

- Open **VirtualBox** and click on the **Machine** menu, then select **New....**
- In the pop-up window:
 - Set the **Type** to **Linux**.
 - Set the **Version** to **Red Hat (64-bit)**.
 - Set the **Name** for the virtual machine to **TP075164-RockyServer** for easy identification.
 - Select the **ISO Image** downloaded for Rocky Linux.

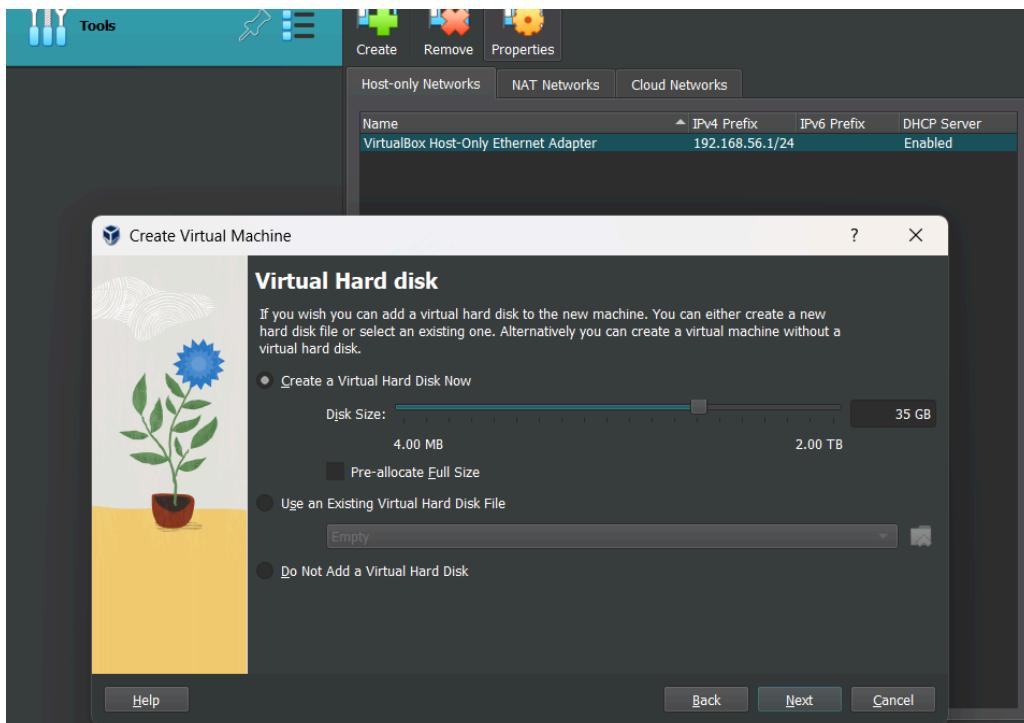


User Setup: For this installation, I created the username **KashaveRocky** to make it easier to identify the user on the system.

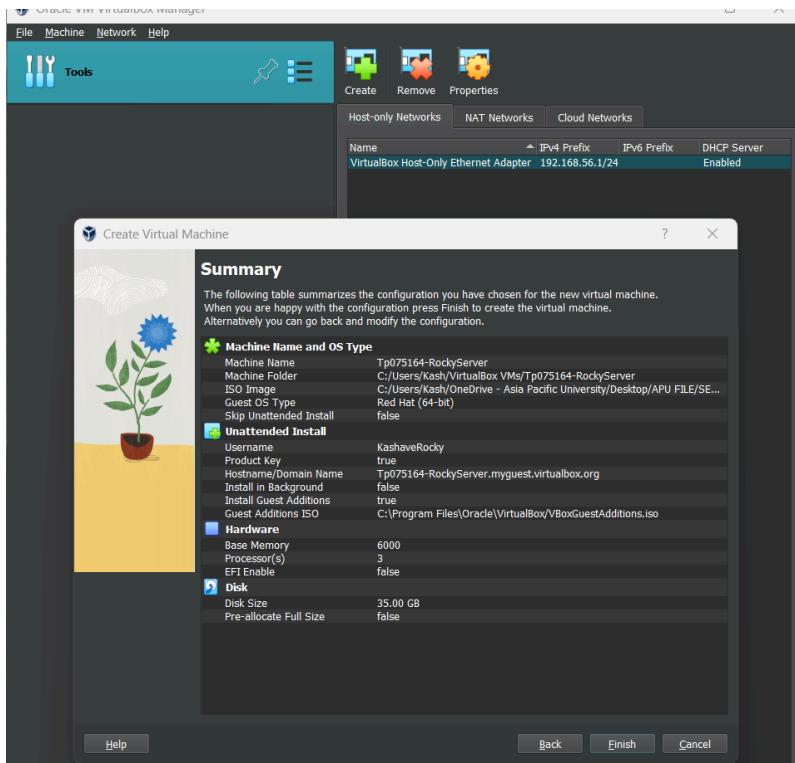
Configuring Resources:



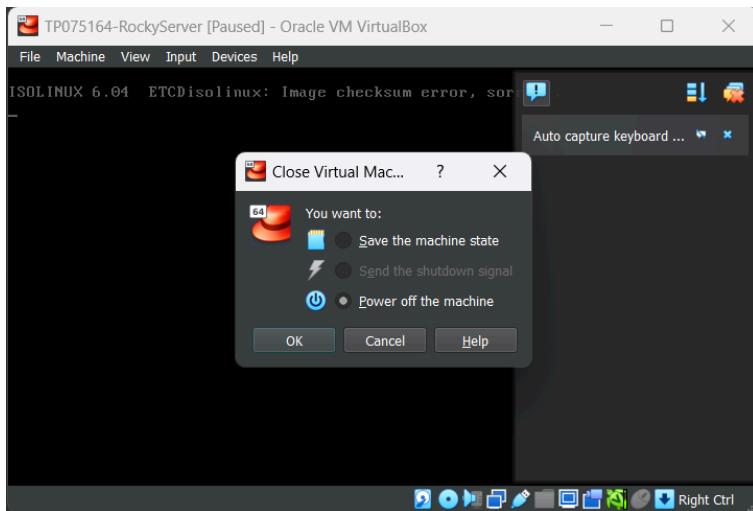
- **Memory and CPU:** I allocated **6000 MB** of base memory (RAM) and assigned **3 CPUs** to the virtual machine to ensure smooth performance during the installation and service configuration process.



- **Virtual Hard Drive:** A **35 GB** virtual hard disk was created for this assignment to accommodate the necessary operating system files and service configurations.



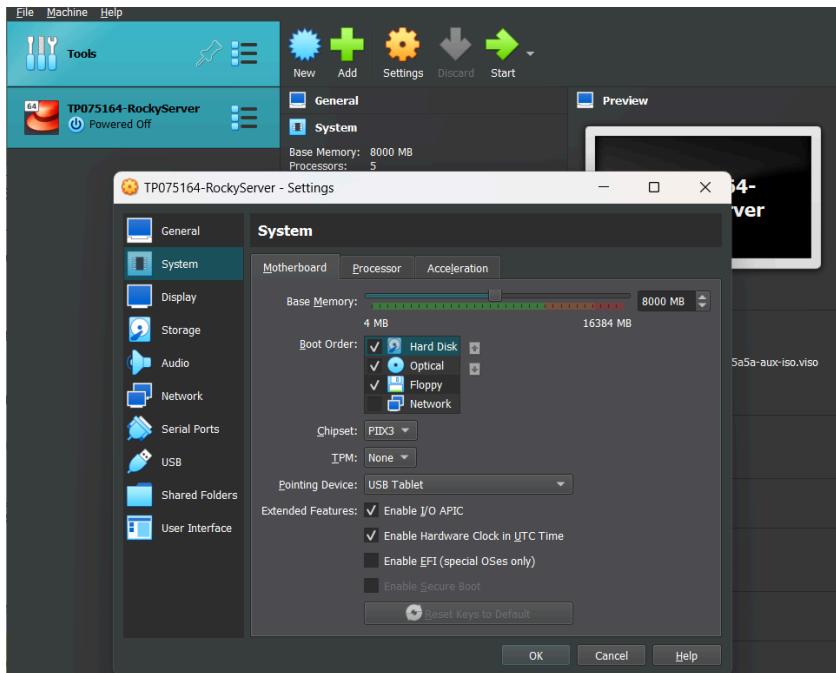
Finalizing VM Setup: After configuring the system, a summary page displayed all previously entered details, allowing me to double-check the configuration before proceeding. Once everything was confirmed as correct, I clicked **Finish** to complete the virtual machine setup.



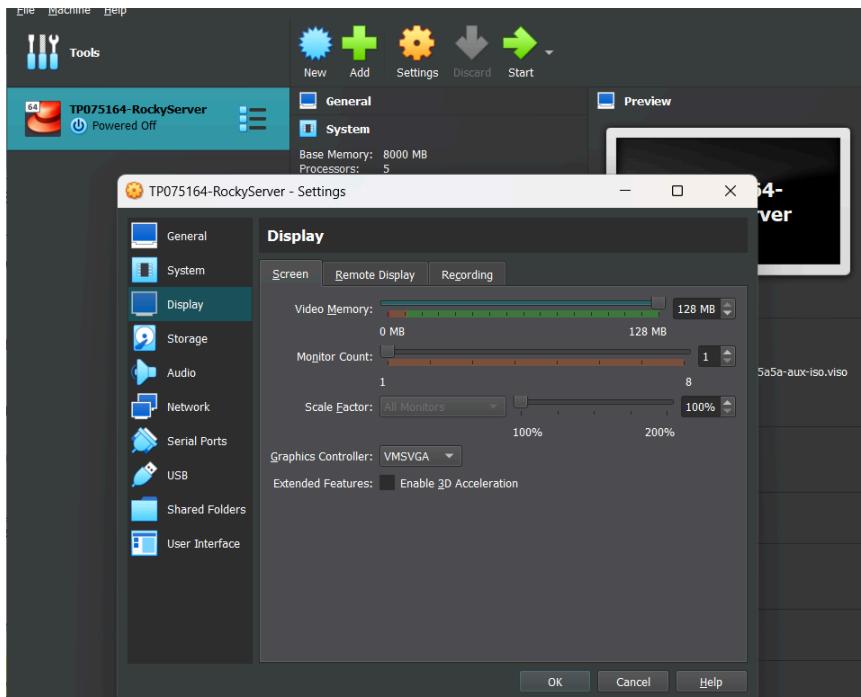
Starting the Virtual Machine:

- After setup, click on the machine and press the **Start** button located at the top of the VirtualBox window. Initially, the system displayed a **checksum error**.
 - To handle this, I selected **File > Close Machine** and ensured that **Power Off the Machine** was selected before clicking **OK**. This step ensures data integrity by preventing potential file corruption and gracefully shutting down the machine.

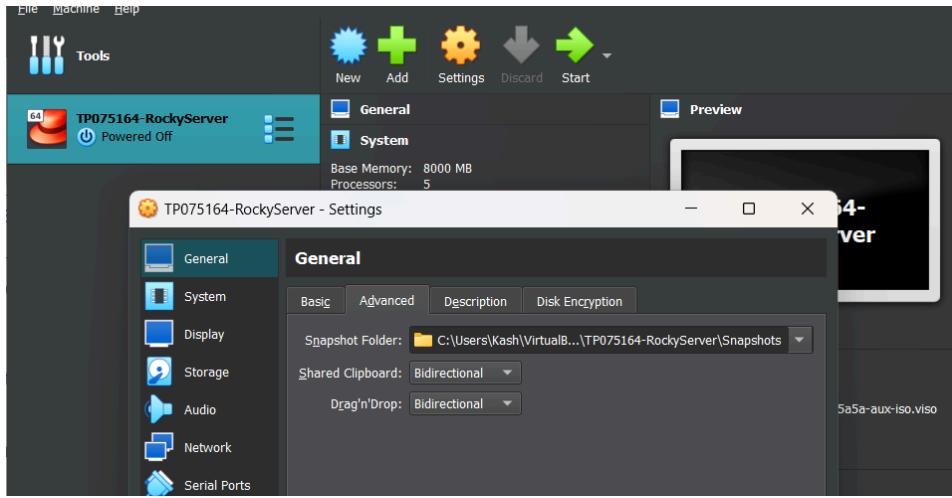
Adjusting Machine Settings: After shutting down the machine, I made further adjustments:



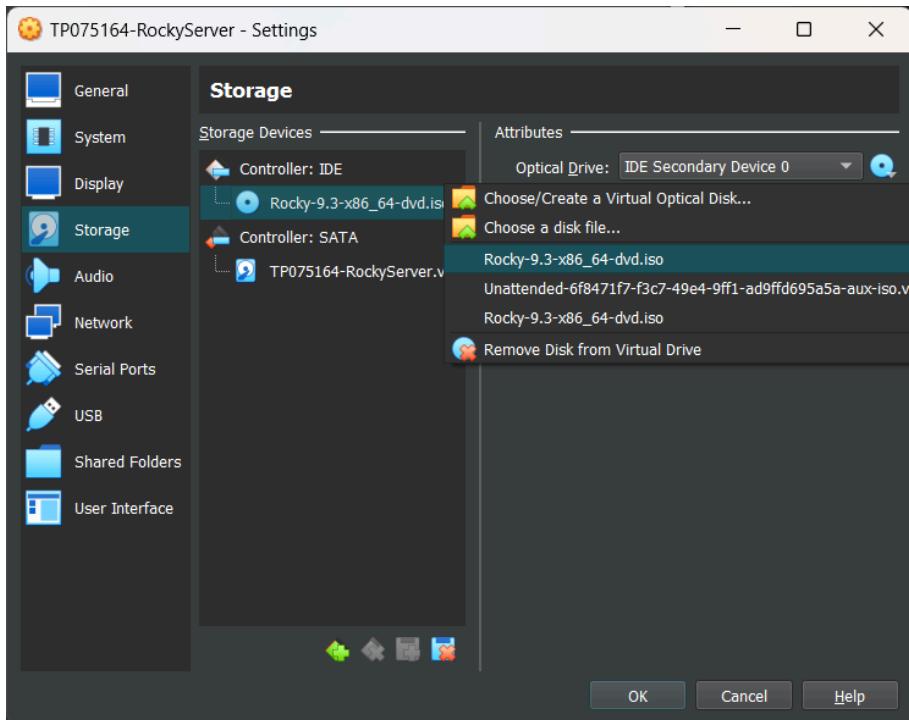
Pointing Device: I changed the pointing device to **USB Tablet** under **Settings > System** to improve mouse handling.



Video Memory: In **Settings > Display**, I set the **Video Memory** to the maximum of **128 MB** for optimal graphical performance.

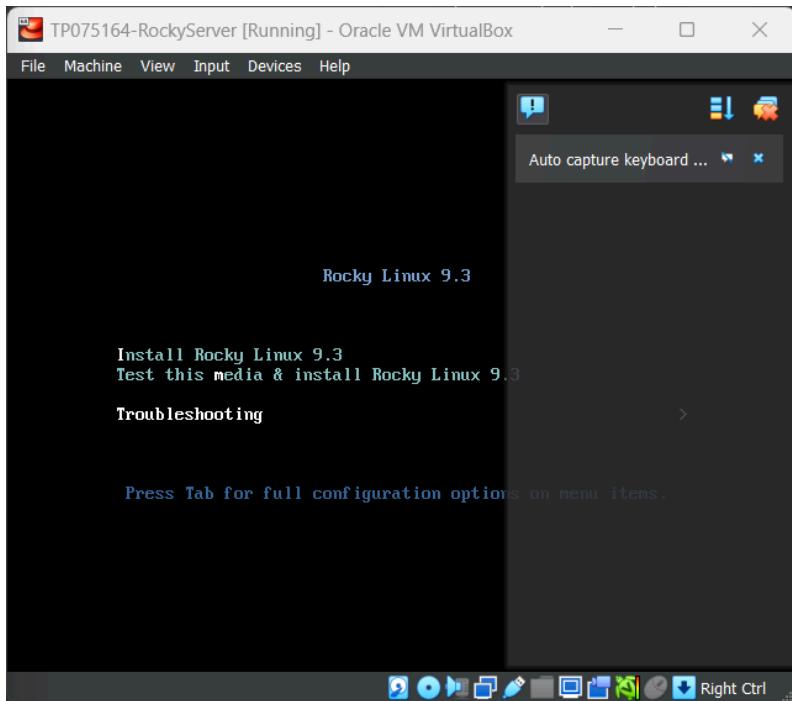


Clipboard and Drag'n'Drop: In **Settings > General > Advanced**, I enabled **Bidirectional** for both **Shared Clipboard** and **Drag'n'Drop**, allowing seamless file sharing and text copy-pasting between the host and guest environments.



Setting the ISO Image:

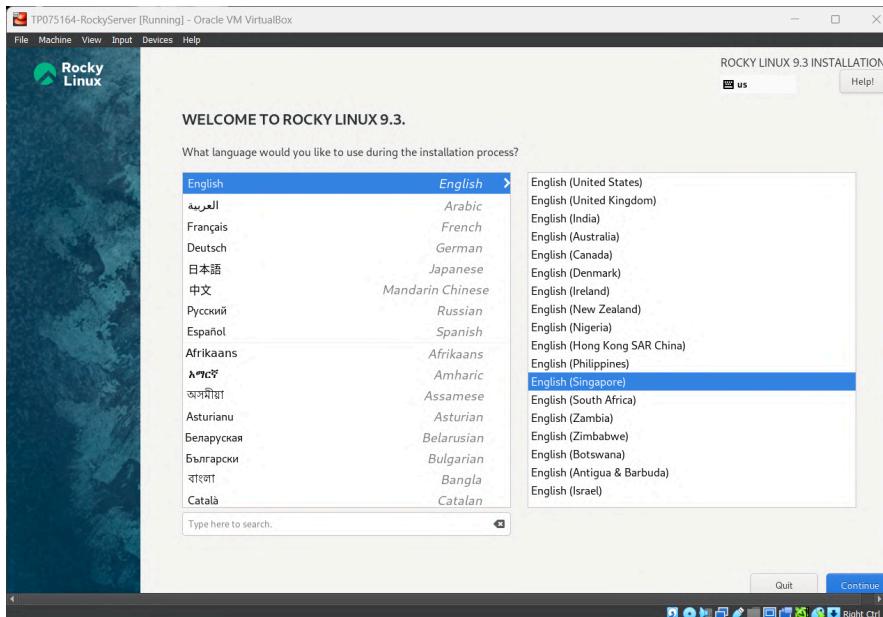
- In **Settings > Storage**, under **Controller: IDE**, I clicked on the untitled CD, then selected the **Optical Drive** option. From there, I chose the **Rocky Linux DVD ISO** used during the initial setup. This ensures the system uses the correct installation media.



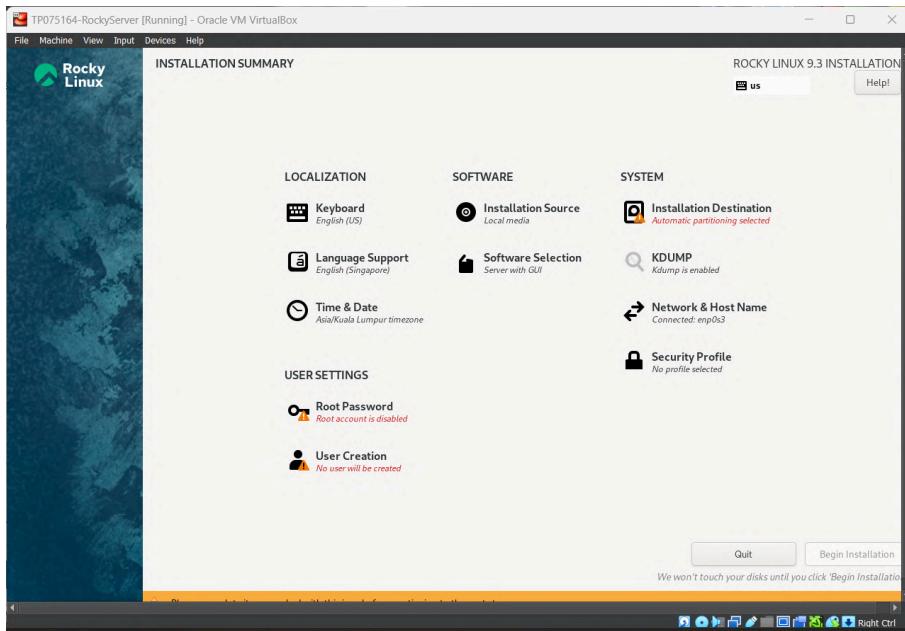
Installing Rocky Linux:

- With the configuration complete, I started the virtual machine again and selected **Install Rocky Linux 9.3** from the boot menu using the arrow keys.

Installation:

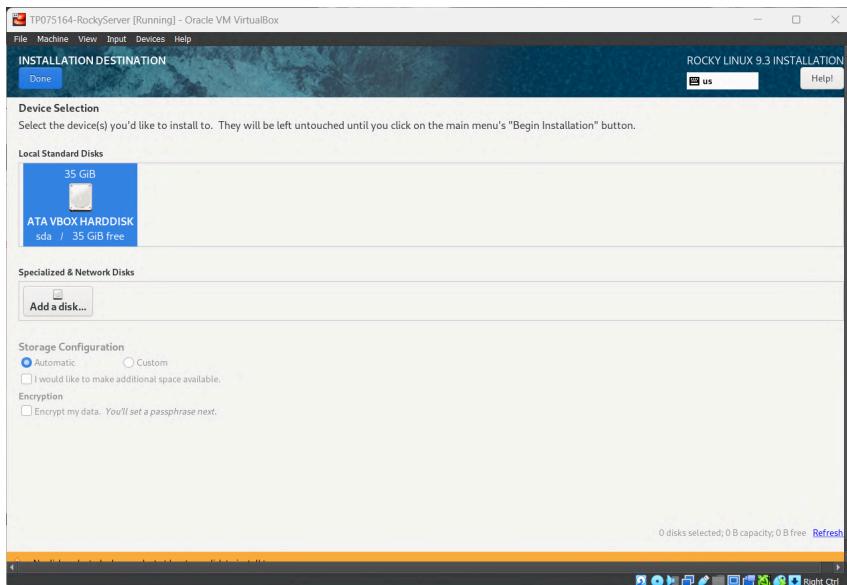


Selecting Language: After launching the installer, I am prompted to select my desired language. I chose **English (Singapore)** for this setup.



Resolving Pre-Installation Errors:

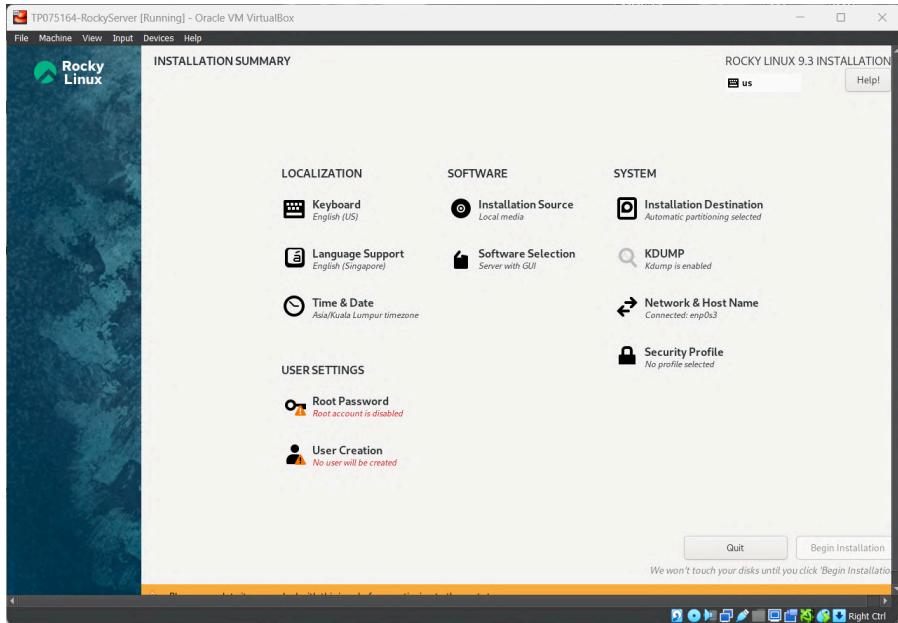
- On the next screen, there may see a few errors that need to be addressed before starting the installation. These errors typically relate to disk selection, root password, and user creation.



Selecting Installation Destination:

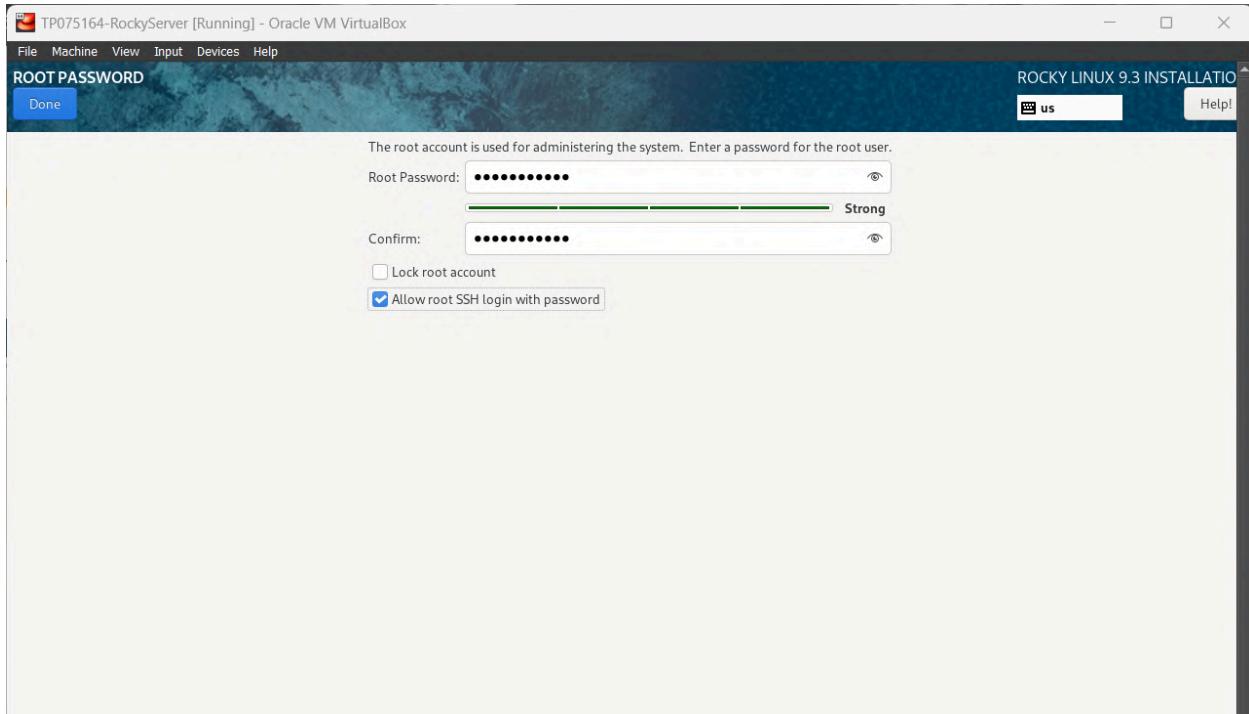
- Click on **Installation Destination**. I will see a list of available drives under **Local Standard Disks**. Ensure that the drive I intend to install Rocky Linux on is selected. There should be a **white tick on a black background** to indicate it is selected. If this is missing, the system will not recognize my selection. Once confirmed, click **Done** at the top left of the screen.

When the error no longer displays, I can move onto ‘root password’ and ‘user creation’.

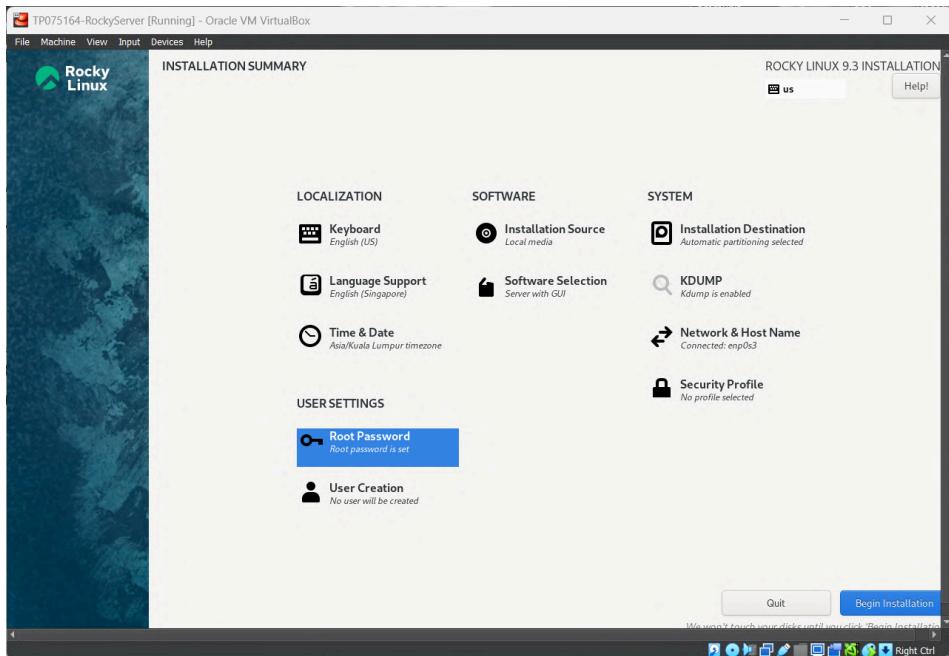


Setting Root Password and User Creation:

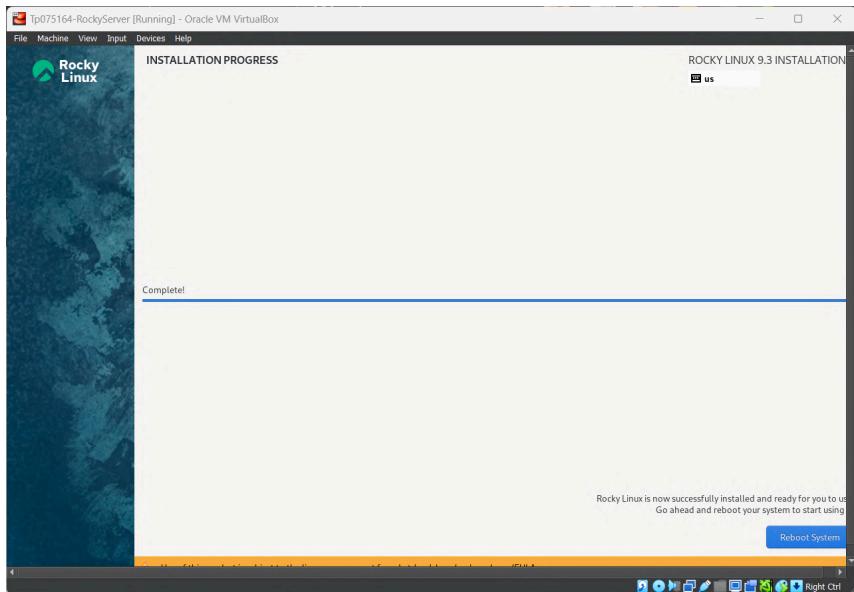
- With the installation destination set, click on **Root Password**.



- Create a strong root password, and make sure to enable **Allow root SSH login with password** for remote access later. Remember, this password is crucial for system administration tasks, so be sure to store it securely. Once done, click **Done**.



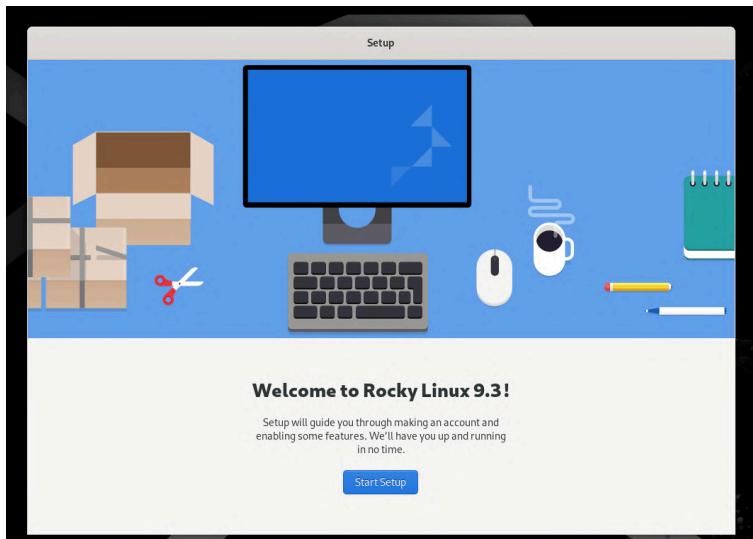
Beginning the Installation: Once all errors are resolved, you can begin the installation process by clicking the **Begin Installation** button at the bottom right. The installation process can take up to an hour, depending on system resources.



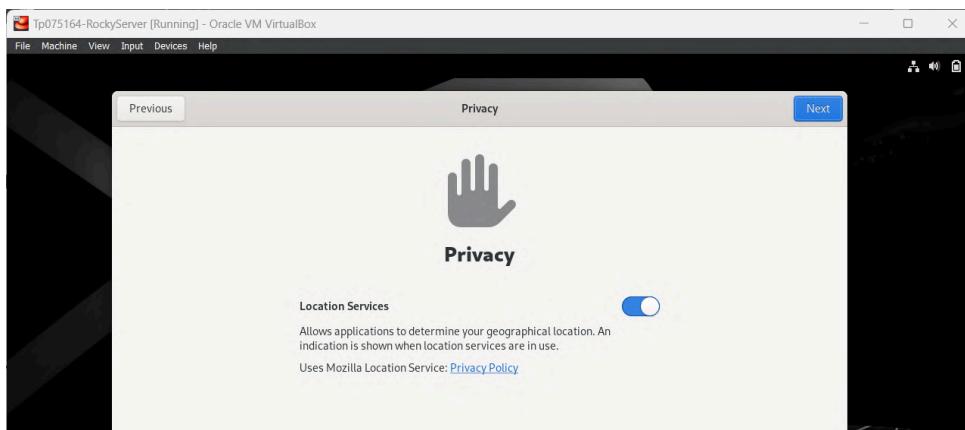
Handling Post-Installation Issues:

- After the installation completes, I encountered an issue where the **Reboot System** button did not work. To resolve this, I manually shut down the virtual machine by selecting **File > Close Machine** and ensuring **Power Off the Machine** was selected.
- After shutting down, go to **Settings > Storage** in VirtualBox, unmount the **Rocky DVD ISO** image, and then mount it again. This ensures the system boots from the installed OS rather than the ISO.

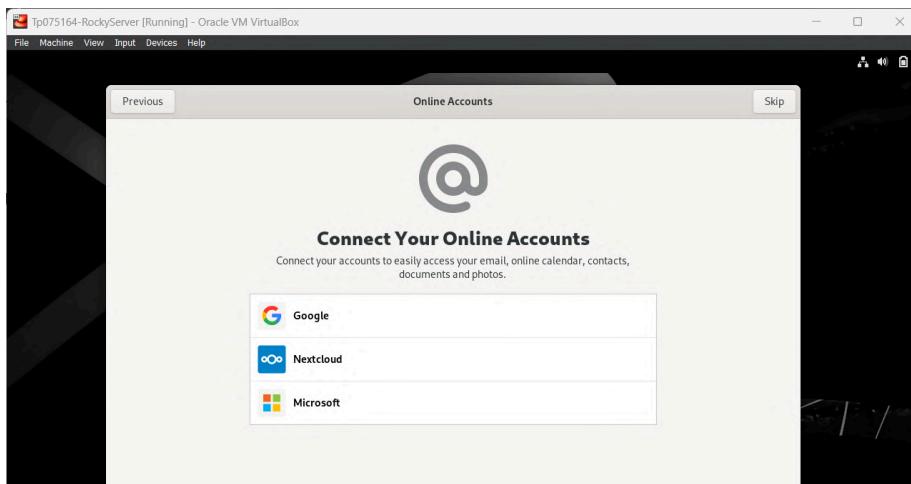
Post-Installation Setup:



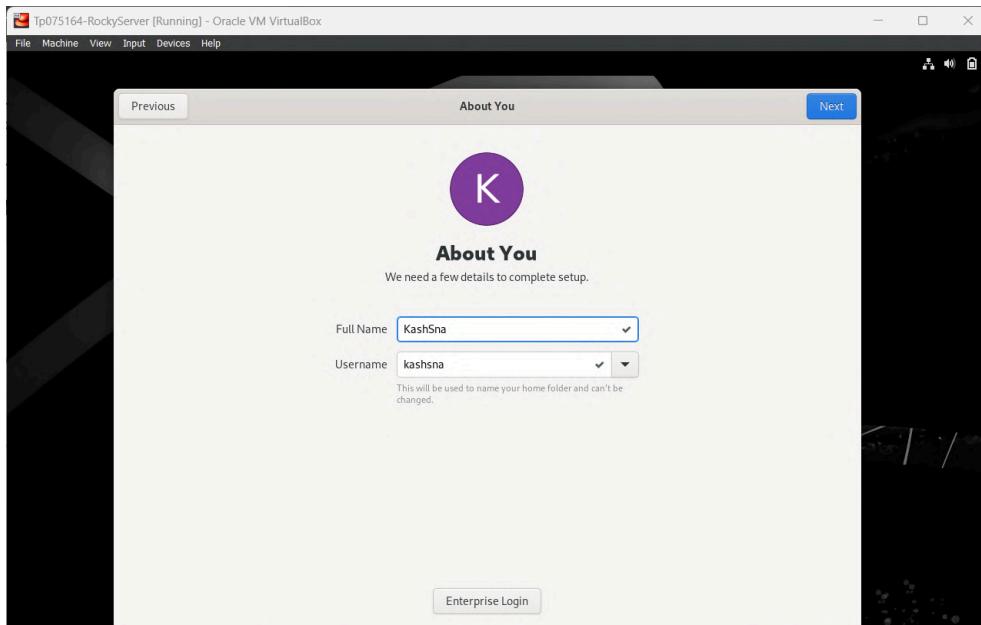
- After the installation, the system will prompt you to complete the setup process.



- Location Services:** Turn the **Location Services** on, and click **Next**.

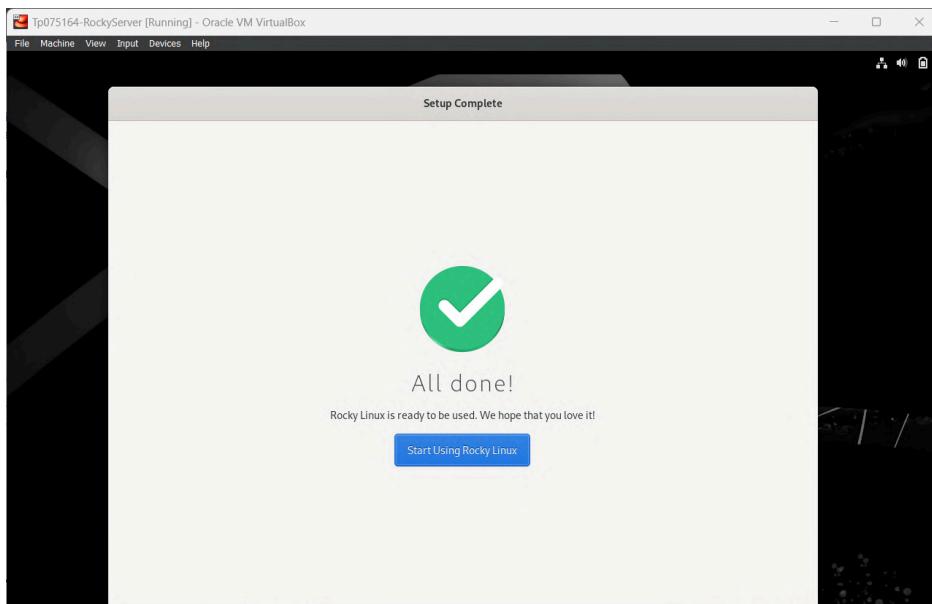


- I skipped the following page.



Setting Up the Login Username:

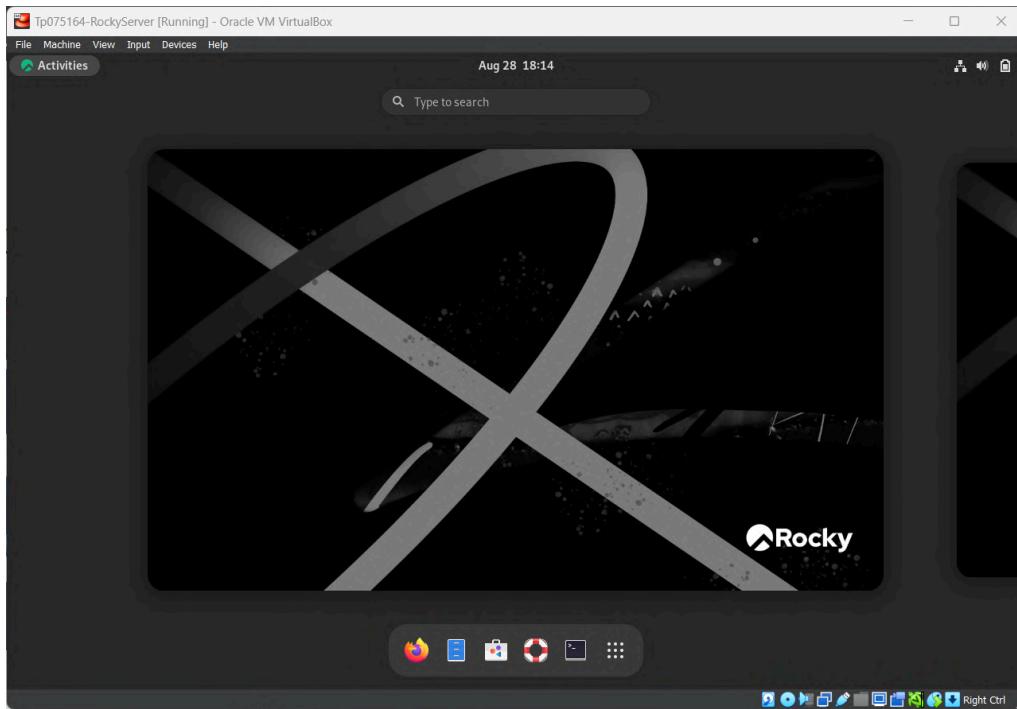
- I am prompted to create a username that will be used to log in to the system. Ensure that this username is something easy to remember, as it will be required for future access.
- Once the username is accepted, click **Next** to complete this step.



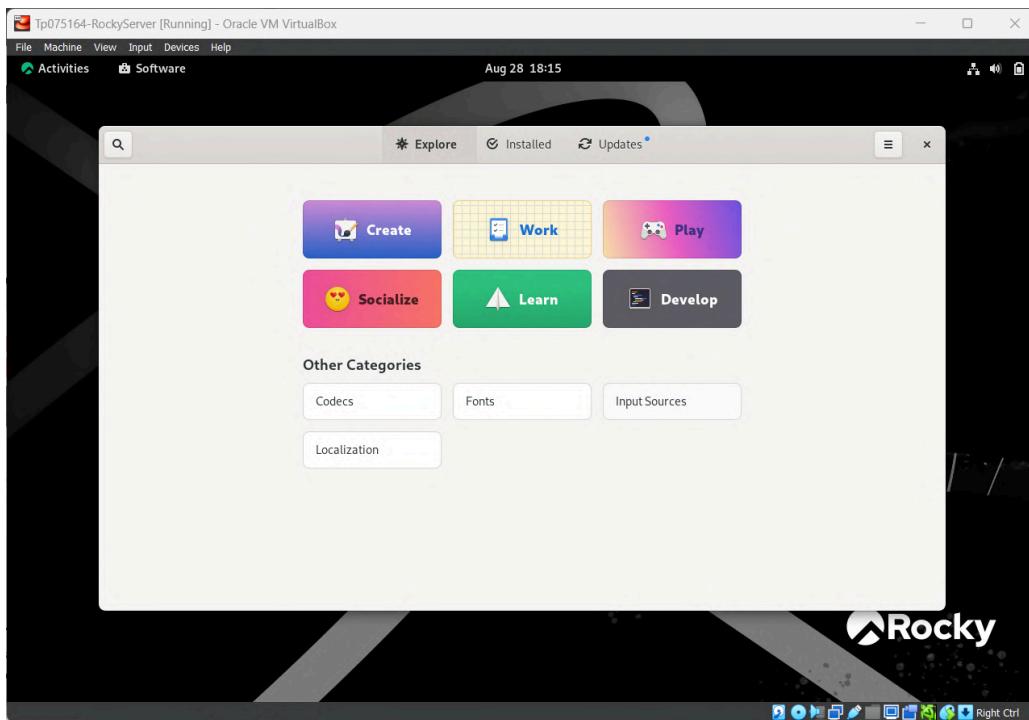
Completion of Setup:

- With the username configured, the setup is now complete! I can start using **Rocky Linux**.

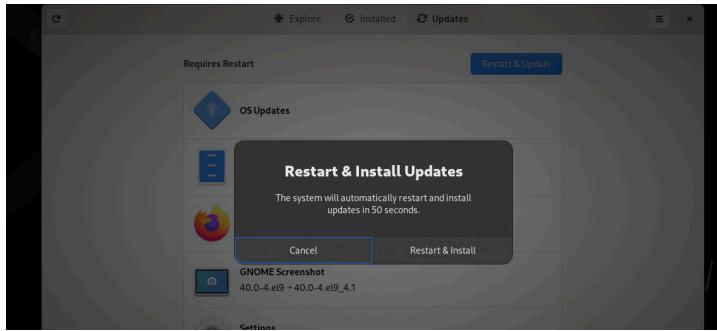
Updating the System:



- To ensure your system is up to date, click the **Activities** tab located in the top-left corner of the screen. In the search bar, type **Software**.

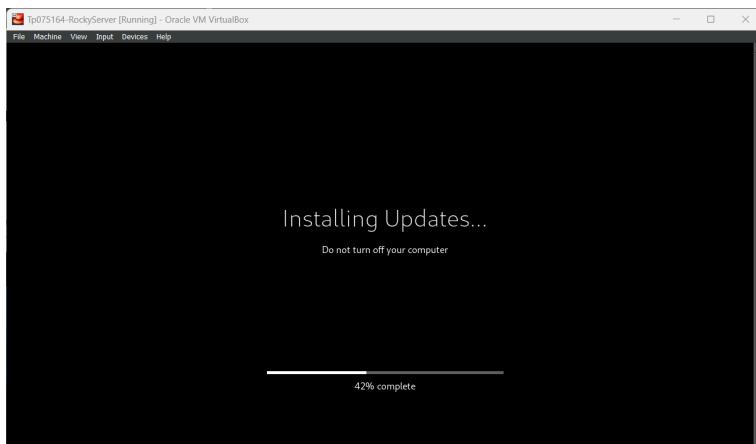


- Open the **Software** program, then navigate to the **Updates** tab.
- Click the **Download** button at the top-right corner to begin downloading any available updates.

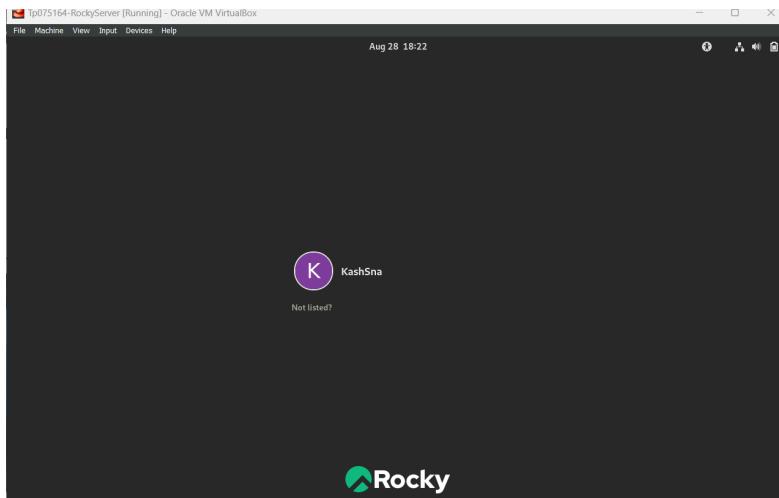


Restart and Update:

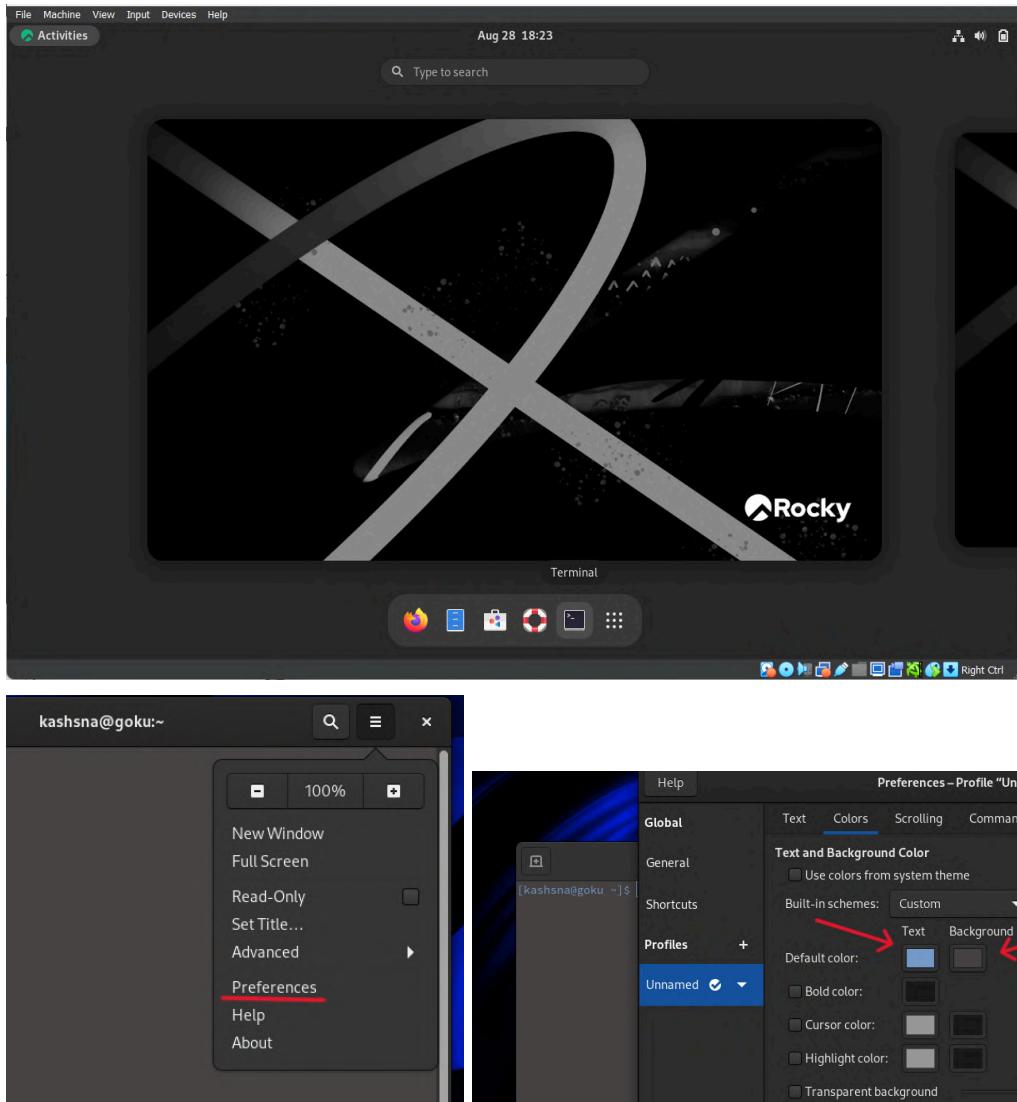
- Once the updates have downloaded, the button will change to **Restart and Update**. Click it to apply the updates.



- Important:** Do not allow your PC to enter sleep or hibernate mode during the update process, as this can result in a corrupted operating system. If the system becomes corrupted, you will need to reinstall Rocky Linux from the beginning.



Logging In: After the system restarts, the login screen will appear. Select your user account, enter your password, and you will be logged into the system, ready to continue with the configuration and setup.



Open terminal. Go to preferences and change the color of text and background.

Opening the Terminal:

- After the system has restarted from the initial update, click the **Activities** tab again and open the terminal.

Running the Update Command:

- To update all installed packages to their latest versions, enter the following command:
 - sudo dnf update -y***

Running the Update Command:

```
[kashsna@localhost ~]$ sudo dnf update -y
Rocky Linux 9 - BaseOS          2.2 MB/s | 2.3 MB    00:01
Rocky Linux 9 - AppStream       5.4 MB/s | 8.0 MB    00:01
Rocky Linux 9 - Extras          14 kB/s | 15 KB    00:01
Dependencies resolved.

=====
Package           Arch   Version        Repo      Size
=====
Installing:
kernel           x86_64 5.14.0-427.31.1.el9_4  baseos   5.0 M
Upgrading:
NetworkManager    x86_64 1:1.46.0-13.el9_4   baseos   2.3 M
NetworkManager-adsl x86_64 1:1.46.0-13.el9_4   baseos   35 k
NetworkManager-bluetooth x86_64 1:1.46.0-13.el9_4   baseos   61 k
NetworkManager-config-server noarch 1:1.46.0-13.el9_4   baseos   20 k
NetworkManager-libnm  x86_64 1:1.46.0-13.el9_4   baseos   1.8 M
NetworkManager-team  x86_64 1:1.46.0-13.el9_4   baseos   40 k
NetworkManager-tui   x86_64 1:1.46.0-13.el9_4   baseos   245 k
NetworkManager-wifi  x86_64 1:1.46.0-13.el9_4   baseos   83 k
NetworkManager-wwan  x86_64 1:1.46.0-13.el9_4   baseos   68 k
bpftool            x86_64 7.3.0-427.31.1.el9_4   baseos   5.7 M
```

- As we can see, all files are being updated successfully.

Upgrading the System:

- To ensure that all packages are upgraded to the latest versions, I used the following command:

```
[kashsna@localhost ~]$ sudo dnf upgrade -y
[sudo] password for kashsna:
Last metadata expiration check: 0:05:03 ago on Wed 28 Aug 2024 09:46:28 PM.
Dependencies resolved.
Nothing to do.
Complete!
[kashsna@localhost ~]$
```

The difference between the **update** and **upgrade** commands is as follows:

- sudo dnf update -y***: This command checks for updates but does not install them until the **dnf upgrade** command is run.
- sudo dnf upgrade -y***: This command checks for updates and automatically installs them, ensuring that the system is completely up-to-date.

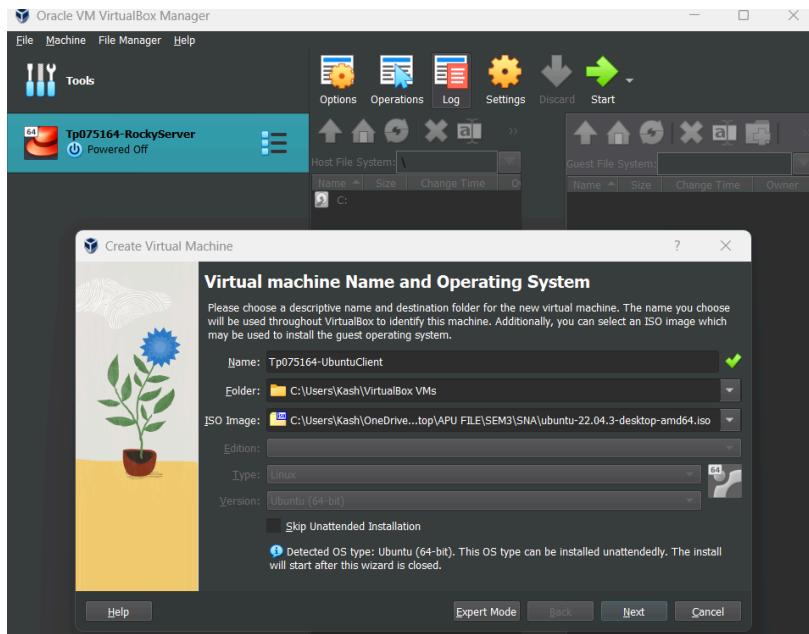
Ubuntu Installation

After setting up Rocky Linux as the server, we repeat a similar process to install Ubuntu as the client. Ubuntu is a popular Linux distribution known for its user-friendly interface and extensive

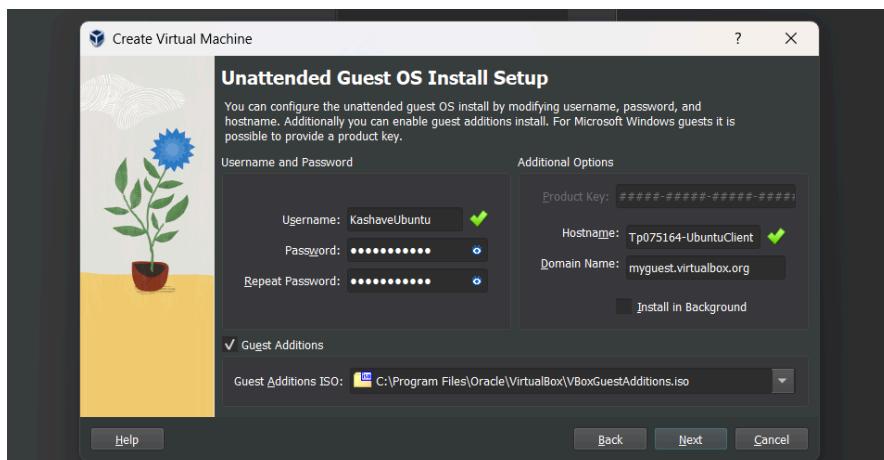
software support. In this project, Ubuntu serves as the client machine in the virtual environment. It is important to install and configure Ubuntu alongside Rocky Linux to simulate the interaction between a server and client, making this setup suitable for a small organizational network.

To distinguish between the virtual machines (VMs), I named this machine TP075164-UbuntuClient, making it easier to manage both VMs later.

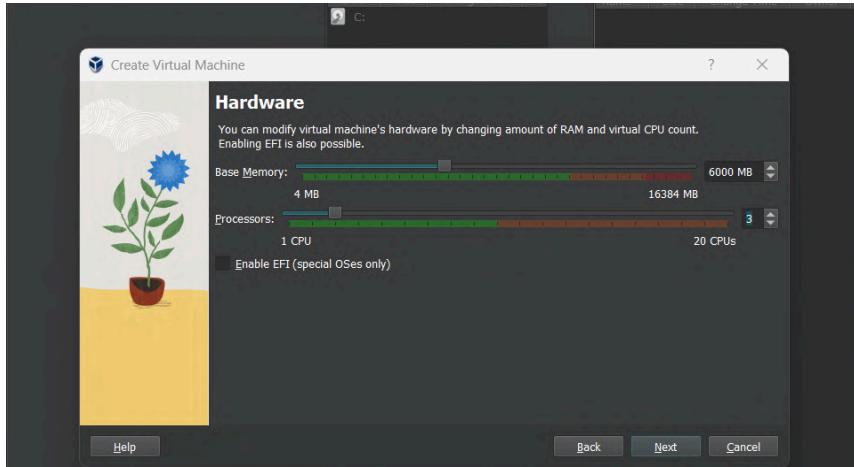
1. Creating the Ubuntu VM:



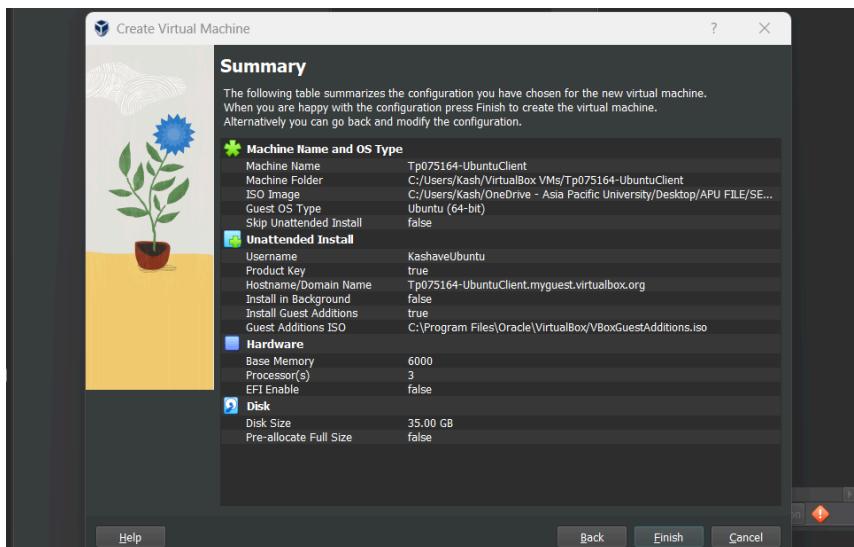
- Open **VirtualBox** and create a new virtual machine, similar to how we did for Rocky Linux.
- Set the **Type** to **Linux** and the **Version** to **Ubuntu (64-bit)**.
- Name the VM **TP075164-UbuntuClient** for easier identification.



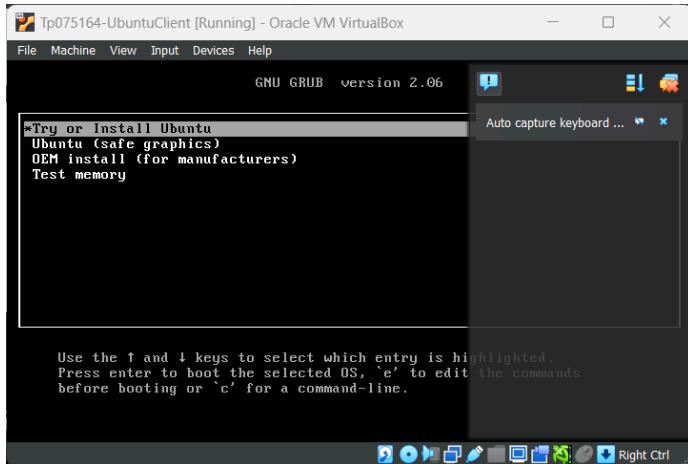
Next, input your username and password. Make sure to include **Guest Additions** during the setup, which allows features such as shared folders and better integration between the host and guest systems.



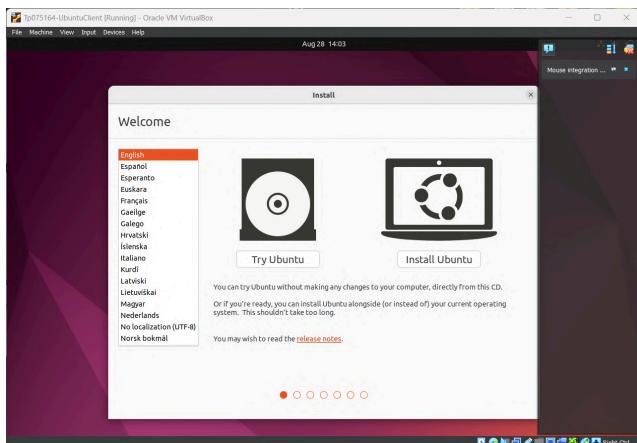
Allocate **memory** and **disk space** (e.g., **6000 MB RAM** and **35 GB hard drive**).



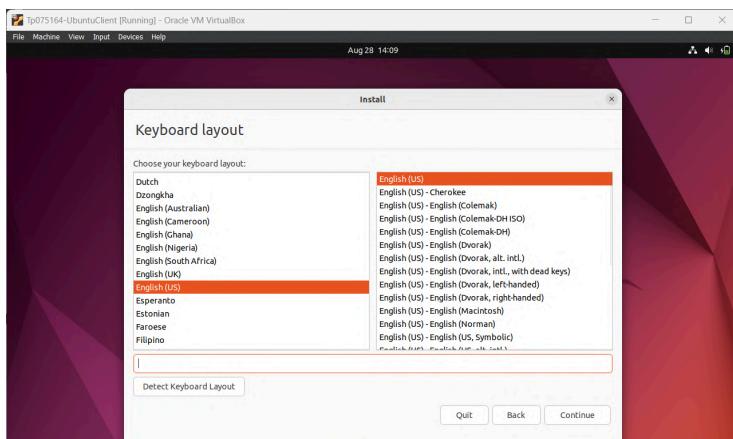
2. Installing Ubuntu:



- Start the VM, and on the boot screen, use the arrow keys to highlight **Try or Install Ubuntu** and press **Enter**.



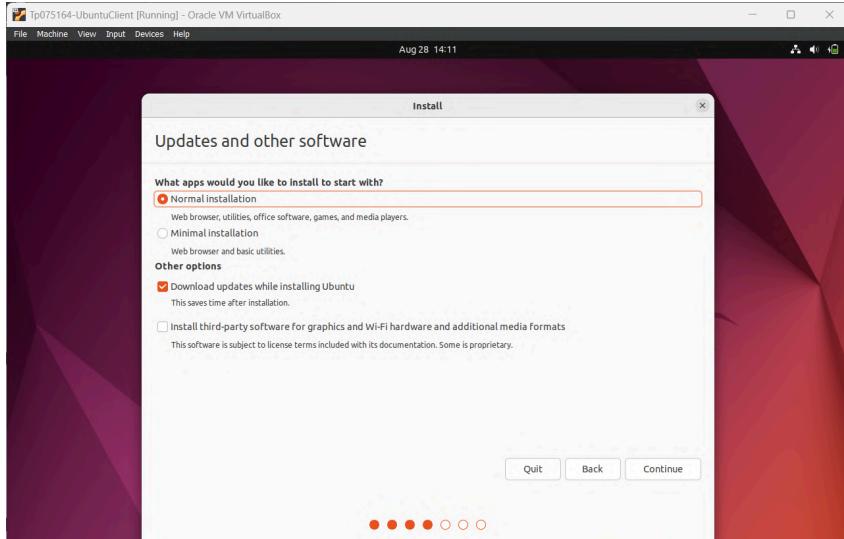
On the next page, click the **Install Ubuntu** button to begin the installation process.



Select Language: Choose your desired language for the installation. In my case, I selected **English**.

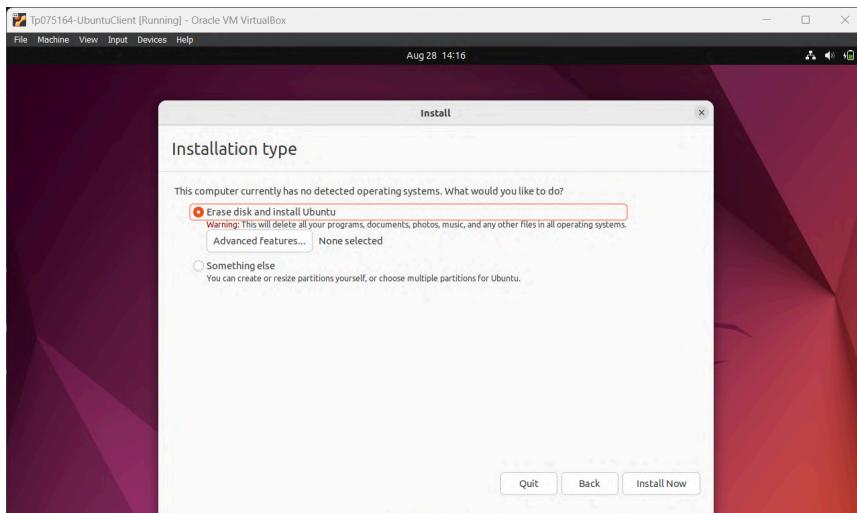
3. Installation Options:

The next page will provide you with 2 installation options. Pick ‘normal installation to get all the software and utilities. And make sure to check ‘Download updates while installing Ubuntu’ so that we won’t have to spend time updating later on.



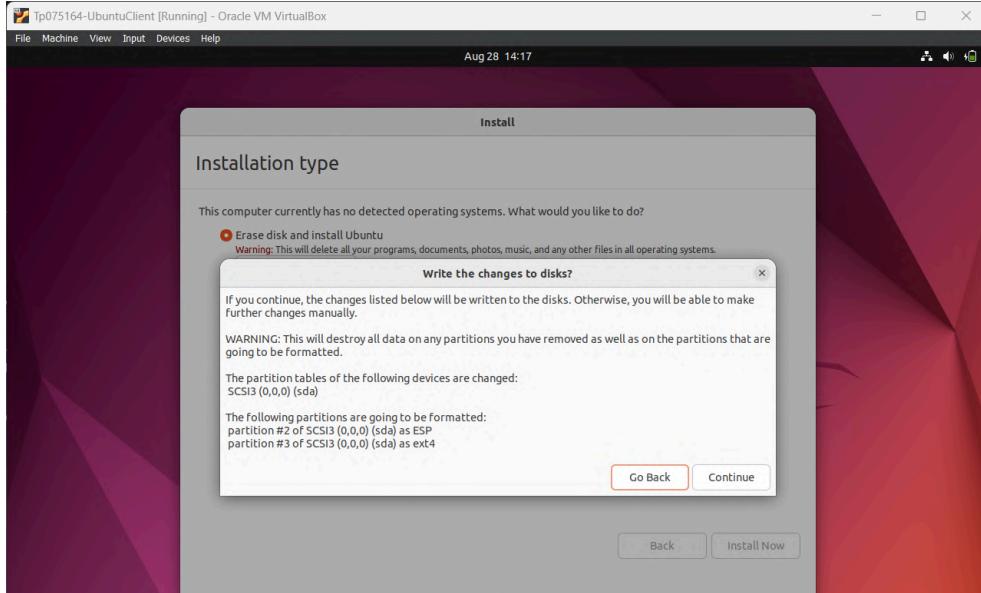
- On the following page, you are provided with two options for installation. Pick:
 - **Normal installation:** This includes all the standard utilities and software needed for typical usage, such as a web browser, office tools, and media players. It is recommended for this setup.
 - Ensure to check **Download updates while installing Ubuntu**, which saves time by ensuring the system is up-to-date after installation.

4. Choosing Installation Type:



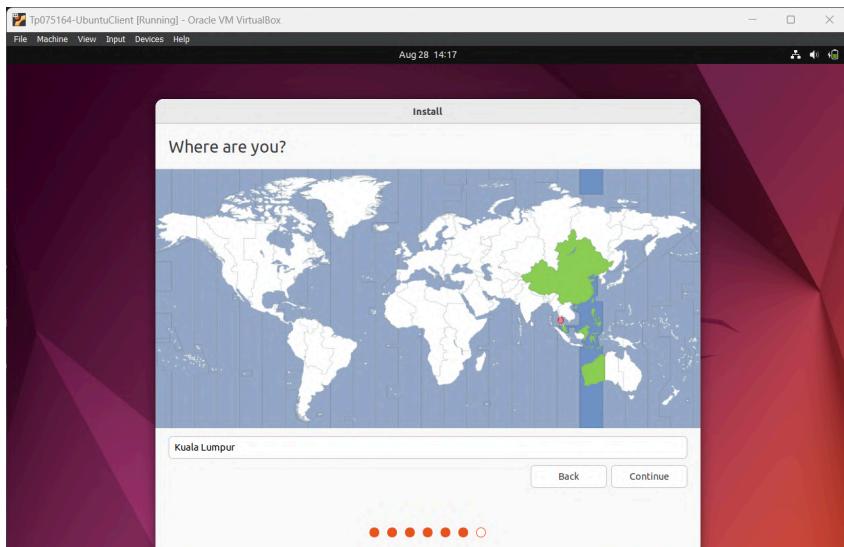
You will be prompted to choose an installation type. I selected **Erase disk and install Ubuntu**. This option clears the virtual disk and installs Ubuntu as the primary operating system.

- **Why choose this option?** Since this is a virtual machine, we don't need to worry about other data on the disk. Erasing the disk allows for a fresh, clean installation of Ubuntu, ensuring no conflicts with previous installations or data. It is safe to erase the virtual disk, as it only affects this virtual environment and not your actual hard drive.

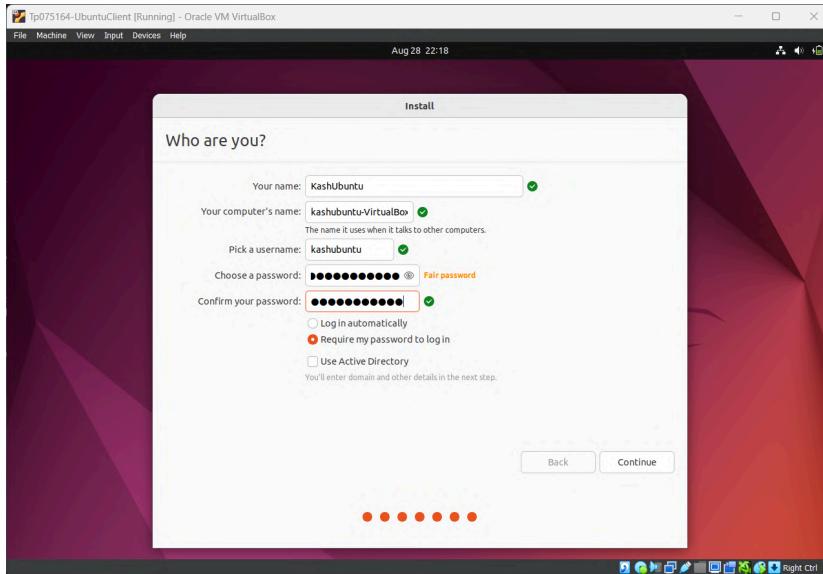


Click **Continue** to proceed with the installation.

5. Location and User Setup:



Next, select your location for time zone and regional settings, then click **Continue**.

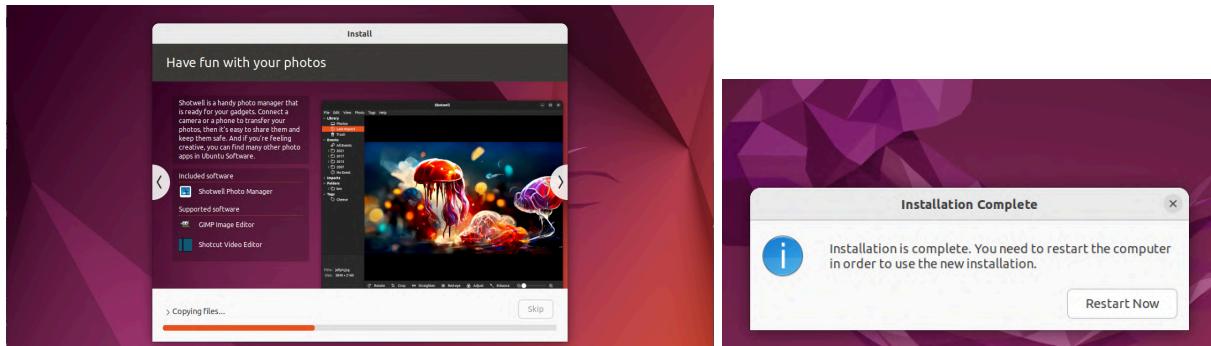


You will then be prompted to create a **username** and **password** for logging into the system.

- Make sure to check **Require my password to log in** for better security, ensuring that access to the system is password-protected.

Click **Continue** to finalize the user setup.

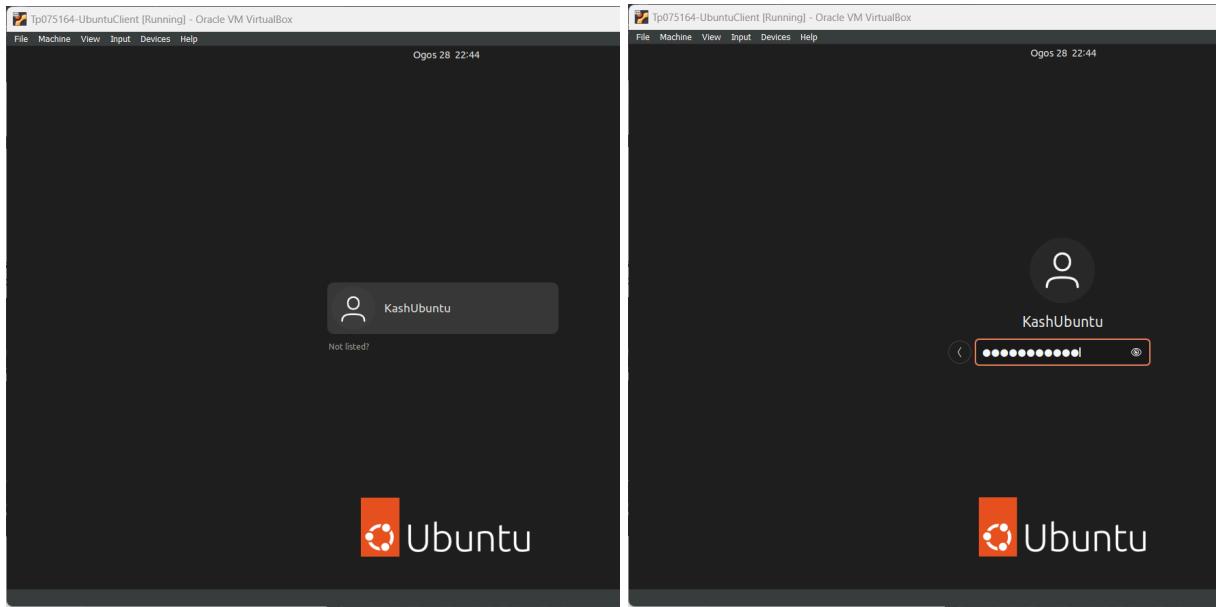
6. Completing the Installation:



- The installation process will now begin. This can take several minutes depending on your system's resources.
- Once the installation is complete, you will be prompted to **restart** the VM. Make sure to do this to complete the installation process and start using Ubuntu.

Post-Installation and User Setup in Ubuntu

1. Logging In:

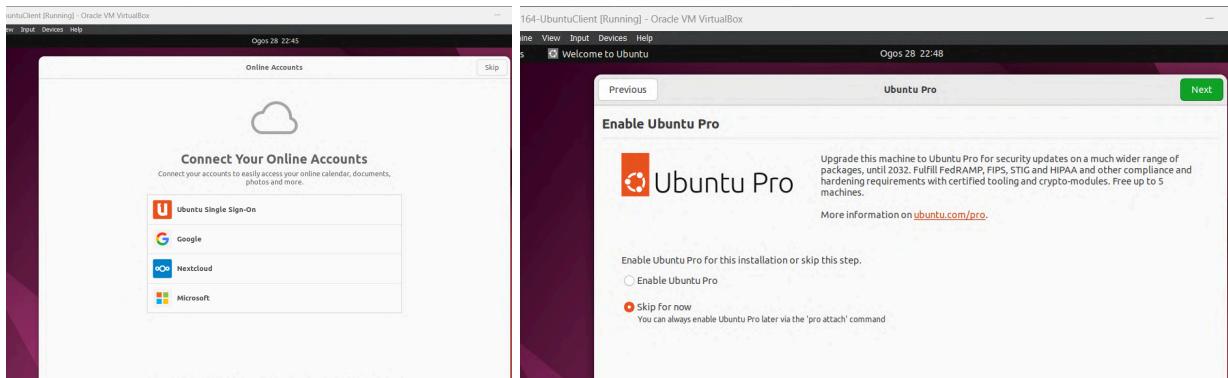


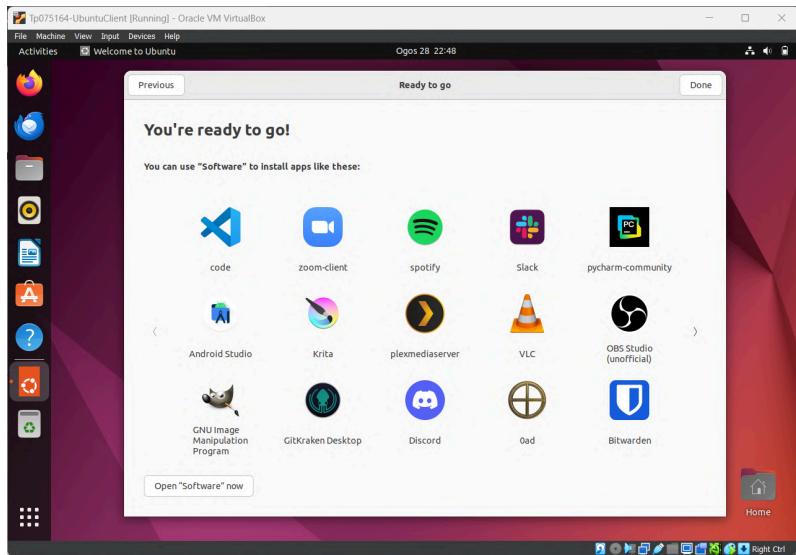
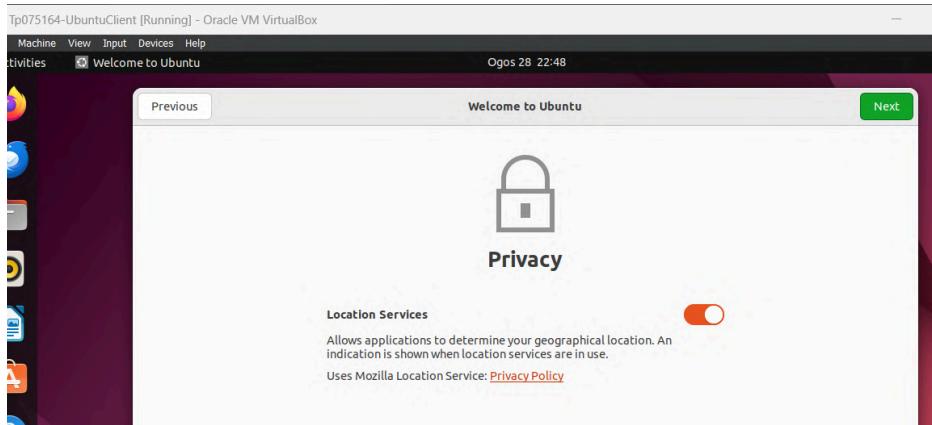
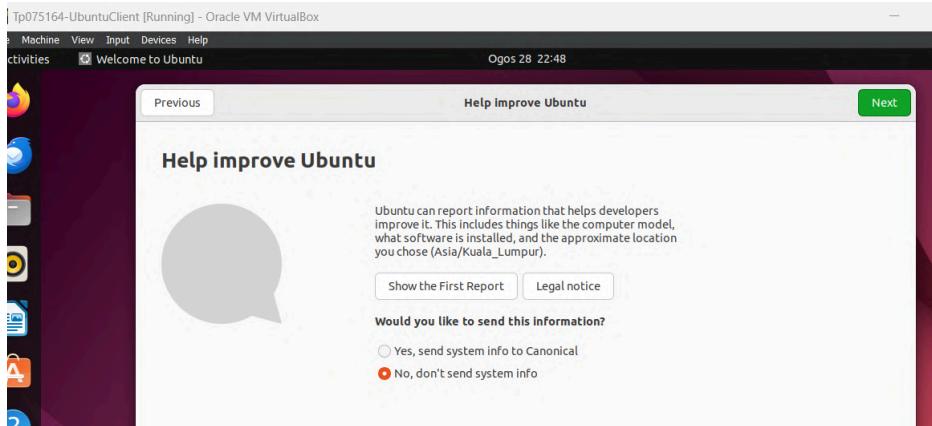
After restarting the system, you can log in using the **username** and **password** you created earlier. Make sure to enter the correct password to access the system.

2. Skipping Unnecessary Steps:

After logging in, you'll be prompted with several options:

- **Skip** connecting online accounts, as it's not necessary for this setup.
- When prompted for **Ubuntu Pro**, select **Skip for now** since we don't need this service.
- **Do not send system information**, but enable **location services** when prompted.



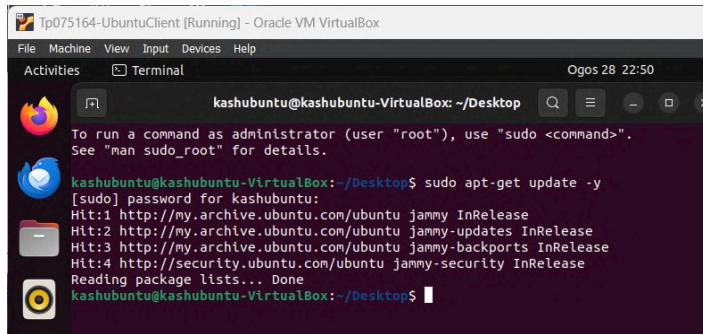


We don't need to install any additional applications at this moment, so click **Done** to complete the setup. Ubuntu is now fully installed!

Updating the System in Ubuntu

1. Opening the Terminal:

- Right-click on the desktop background and select **Open Terminal** to access the command line interface.



The screenshot shows a terminal window titled "Tp075164-UbuntuClient [Running] - Oracle VM VirtualBox". The window title bar includes "File", "Machine", "View", "Input", "Devices", "Help", "Activities", and "Terminal". The status bar at the bottom right says "Ogos 28 22:50". The terminal content shows the command "sudo apt-get update -y" being run, followed by a password prompt, and then a list of package sources being checked. The command "Reading package lists... Done" is visible at the end.

In the terminal, update the package repository information by running:

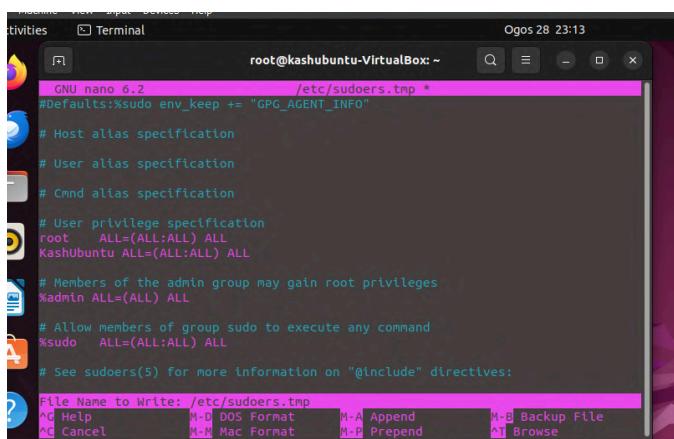
- ***sudo apt-get update -y***

You will be prompted to enter your password. In some cases, users might encounter an error stating that the user is not a **sudoer** (does not have administrative privileges). However, I did not encounter this error and the system immediately began updating the package information.

Ensuring 'sudo' Access

1. Modifying the 'sudoers' File:

To ensure that my user has **sudo** privileges, I edited the **/etc/sudoers** file using the **visudo** command. This file controls which users have administrative rights.



The screenshot shows a terminal window titled "root@kashubuntu-VirtualBox: ~". The window title bar includes "Activities", "Terminal", and "Ogos 28 23:13". The terminal content shows the command "root@kashubuntu-VirtualBox: ~" followed by the output of the "visudo" command. It displays the contents of the "/etc/sudoers" file, which includes standard sudoer directives like "# Host alias specification", "# User alias specification", "# Cmnd alias specification", and "# User privilege specification". A specific line for the root user is shown: "root ALL=(ALL:ALL) ALL". Below this, the "visudo" command prompt asks for a file name to write: "File Name to Write: /etc/sudoers.tmp". The bottom of the screen shows a menu with options like "Help", "Cancel", "DOS Format", "Mac Format", "Append", "Prepend", "Backup File", and "Browse".

First, locate the line that says:

- ***root ALL=(ALL:ALL) ALL***

Below this line, I added a new line for my user (**KashUbuntu**):

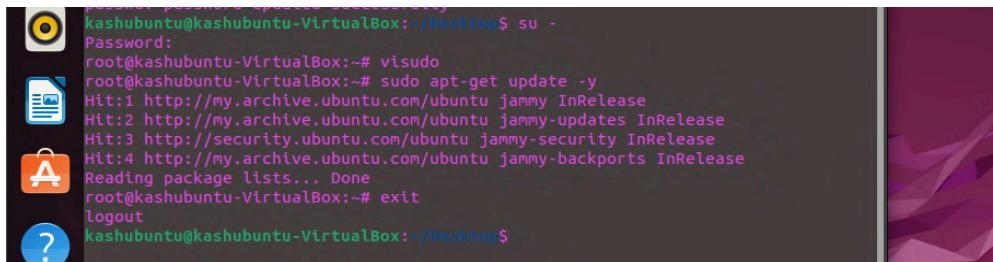
- ***KashUbuntu ALL=(ALL:ALL) ALL***

After making the change, save and exit by pressing:

- **Control + X, Y, then Enter.**

2. Verifying the Update Process:

I then ran the update command again:

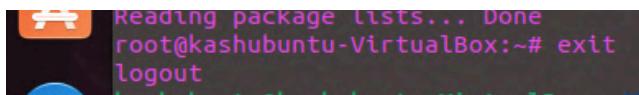


```
kashubuntu@kashubuntu-VirtualBox:~/Desktop$ su -
Password:
root@kashubuntu-VirtualBox:~# visudo
root@kashubuntu-VirtualBox:~# sudo apt-get update -y
Hit:1 http://my.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://my.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://my.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
root@kashubuntu-VirtualBox:~# exit
logout
kashubuntu@kashubuntu-VirtualBox:~/Desktop$
```

Since I was still logged in as the root user, I did not need to enter my password. The system updated successfully.

3. Exiting Root User:

After completing the updates, I used the **exit** command to return to the regular user account.

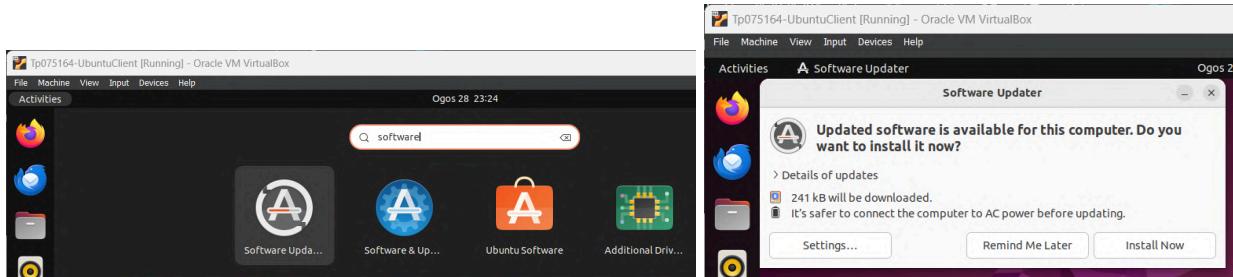


```
Reading package lists... Done
root@kashubuntu-VirtualBox:~# exit
logout
```

Updating Software via the GUI

1. Updating Installed Programs:

To update programs through the software interface, click on the Activities bar, type Software in the search box, and open the Software Program.



Follow the same steps as in Rocky Linux by navigating to the **Updates** tab and downloading the available updates.

Restarting the System:

- Once all updates are complete, restart Ubuntu by typing the **reboot** command in the terminal:

Troubleshooting:

Error 1: Cannot install the best update candidate.

```
[kashsna@localhost ~]$ sudo dnf update -y
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:
#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for kashsna:
Rocky Linux 9 - BaseOS           55 kB/s | 2.3 MB   00:42
Rocky Linux 9 - AppStream         5.9 MB/s | 8.0 MB   00:01
Rocky Linux 9 - Extras            19 kB/s | 15 kB    00:00
Error:
Problem 1: cannot install the best update candidate for package cups-1:2.3.3op2-21.el9.x86_64
- nothing provides cups-libs(x86-64) = 1:2.3.3op2-27.el9_4 needed by cups-1:2.3.3op2-27.el9_4.x86_64 from appstream
Problem 2: cannot install the best update candidate for package cups-client-1:2.3.3op2-21.el9.x86_64
- nothing provides cups-libs(x86-64) = 1:2.3.3op2-27.el9_4 needed by cups-client-1:2.3.3op2-27.el9_4.x86_64 from appstream
Problem 3: cannot install the best update candidate for package cups-ippool-1:2.3.3op2-21.el9.x86_64
```

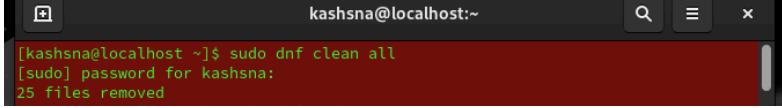
- This command initiates the process of checking for and downloading the latest updates for all installed packages. However, during this process, I encountered an error where the system was unable to install the "best update candidate" for several packages, including **cups**.

Understanding the 'Best Update Candidate' Error:

- The term "**best update candidate**" refers to the most appropriate version of a package for installation, which is determined by several factors:
 - **Compatibility:** The new version must be compatible with the system's hardware and software environment.
 - **Dependencies:** The package must not conflict with other installed packages.
 - **Repository Prioritization:** If multiple versions are available from different repositories, the one from a higher-priority repository will be chosen.
- Potential causes for this error include:
 - **Corrupt or incomplete package metadata:** This often happens if the system has cached corrupt metadata, preventing proper package installation.
 - **Network connectivity issues:** If the package manager cannot reach the repositories, it might fail to find the required packages.
 - **Temporary file system problems:** These can interfere with package installations.

Solving the Issue:

- To resolve this issue, I used the following command to clean the **DNF cache**:

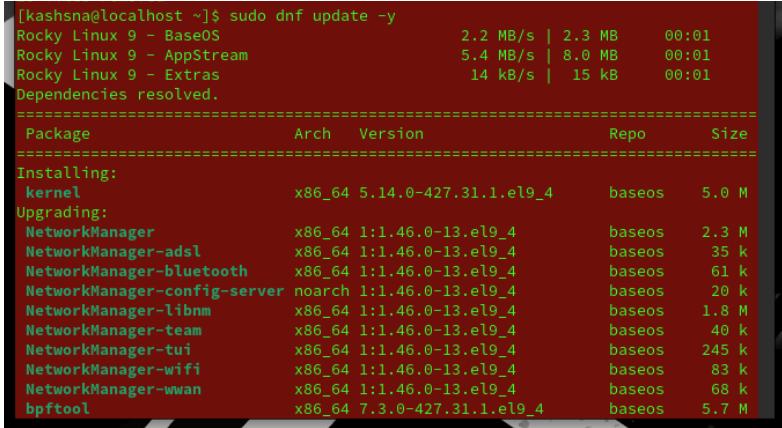


```
[kashsna@localhost ~]$ sudo dnf clean all
[sudo] password for kashsna:
25 files removed
```

- This command clears out all cached metadata and package data, which is useful if there are any corrupt or incomplete package files. In my case, it removed **25 files** from the cache.

Re-running the Update Command:

- After cleaning the cache, I ran the update command again:



```
[kashsna@localhost ~]$ sudo dnf update -y
Rocky Linux 9 - BaseOS           2.2 MB/s | 2.3 MB   00:01
Rocky Linux 9 - AppStream        5.4 MB/s | 8.0 MB   00:01
Rocky Linux 9 - Extras          14 kB/s | 15 KB   00:01
Dependencies resolved.

=====
Package          Arch    Version      Repo      Size
=====
Installing:
 kernel          x86_64  5.14.0-427.31.1.el9_4  baseos   5.0 M
Upgrading:
 NetworkManager   x86_64  1:1.46.0-13.el9_4   baseos   2.3 M
 NetworkManager-adsl x86_64  1:1.46.0-13.el9_4   baseos   35 k
 NetworkManager-bluetooth x86_64  1:1.46.0-13.el9_4   baseos   61 k
 NetworkManager-config-server noarch  1:1.46.0-13.el9_4   baseos   20 k
 NetworkManager-libnm   x86_64  1:1.46.0-13.el9_4   baseos   1.8 M
 NetworkManager-team   x86_64  1:1.46.0-13.el9_4   baseos   40 k
 NetworkManager-tui    x86_64  1:1.46.0-13.el9_4   baseos   245 k
 NetworkManager-wifi   x86_64  1:1.46.0-13.el9_4   baseos   83 k
 NetworkManager-wwan   x86_64  1:1.46.0-13.el9_4   baseos   68 k
 bpftrace          x86_64  7.3.0-427.31.1.el9_4  baseos   5.7 M
```

- This time, all files updated successfully without any errors, confirming that the issue was likely caused by corrupted or incomplete package metadata.

Error 2: Troubleshooting Administrative Access

Checking Root Access:

```
kashubuntu@kashubuntu-VirtualBox:~/Desktop$ su -  
Password:  
su: Authentication failure
```

- When attempting to switch to the **root** user using '**su -**', I encountered an "**authentication failure**" error, even when entering the correct password. To troubleshoot, I needed to check if my user account had **sudo** privileges.

Verifying 'sudo' Privileges:

```
su: Authentication failure  
kashubuntu@kashubuntu-VirtualBox:~/Desktop$ groups  
kashubuntu adm cdrom sudo dip plugdev lpadmin lxd sambashare
```

- I ran the **groups** command to check if my user was part of the **sudo** group. Since the **sudo** group was listed, it confirmed that my account had administrative privileges.

Enabling Root Account:

```
kashubuntu@kashubuntu-VirtualBox:~/Desktop$ sudo passwd root  
[sudo] password for kashubuntu:  
New password:  
Retype new password:  
passwd: password updated successfully
```

- The error suggested that the root account might be disabled. To enable it and set a password, I ran the **sudo passwd root** command.

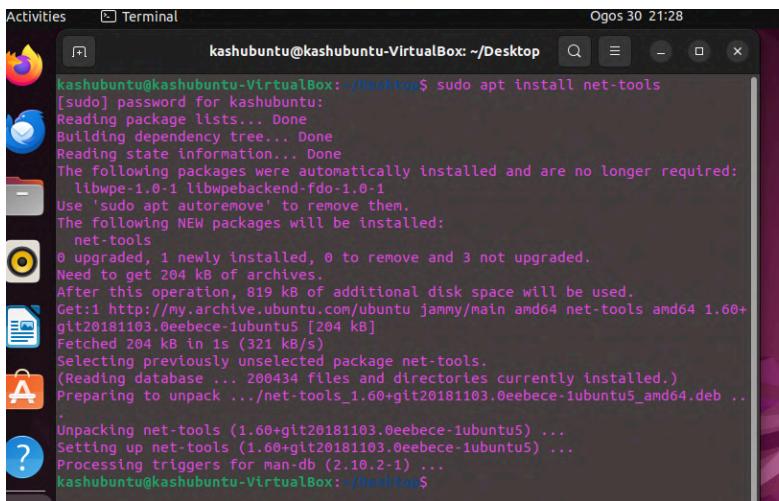
```
kashubuntu@kashubuntu-VirtualBox:~/Desktop$ su -  
Password:  
root@kashubuntu-VirtualBox:~#
```

After setting a new password for the root user, I could successfully switch to the root account using the **su -** command and the newly set password.

2.0 Setting up NAT Network

1. Checking IP Address on Ubuntu Client:

Installing Net-Tools

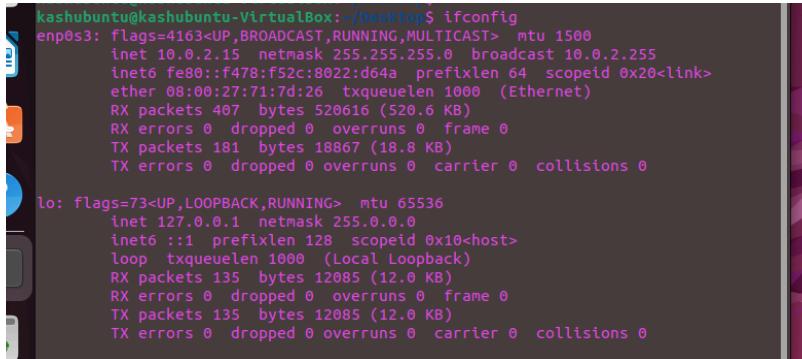


```
kashubuntu@kashubuntu-VirtualBox:~/Desktop$ sudo apt install net-tools
[sudo] password for kashubuntu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 204 kB of archives.
After this operation, 819 kB of additional disk space will be used.
Get:1 http://my.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Fetched 204 kB in 0s (321 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 200434 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
.
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Processing triggers for man-db (2.10.2-1) ...
kashubuntu@kashubuntu-VirtualBox:~/Desktop$
```

To begin, we need to install **net-tools** to use the **ifconfig** command. Run the following command in the terminal:

- **Sudo apt install net-tools**

After installation, use the **ifconfig** command to check the current IP address of the Ubuntu system:



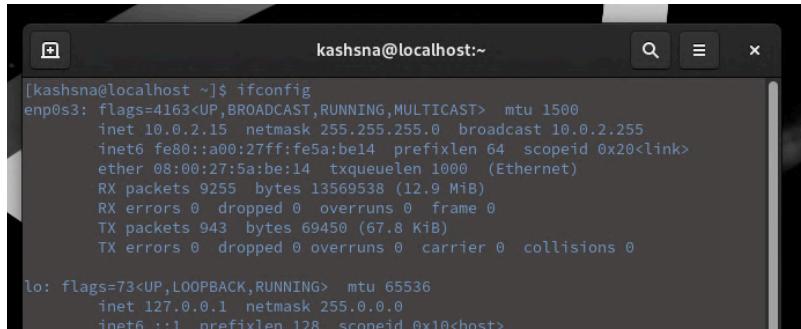
```
kashubuntu@kashubuntu-VirtualBox:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
  inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
    inet6 fe80::f478:f52c:8022:d64a  prefixlen 64  scopeid 0x20<link>
      ether 08:00:27:71:7d:26  txqueuelen 1000  (Ethernet)
        RX packets 407  bytes 520616 (520.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 181  bytes 18867 (18.8 KB)
        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
  inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
      loop  txqueuelen 1000  (Local Loopback)
        RX packets 135  bytes 12085 (12.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 135  bytes 12085 (12.0 KB)
        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

The default IP address, **10.0.2.15**, is assigned by the Virtual Machine's NAT interface.

2. Checking IP Address on Rocky Server:

Start the Rocky server and run **ifconfig** to check its IP address



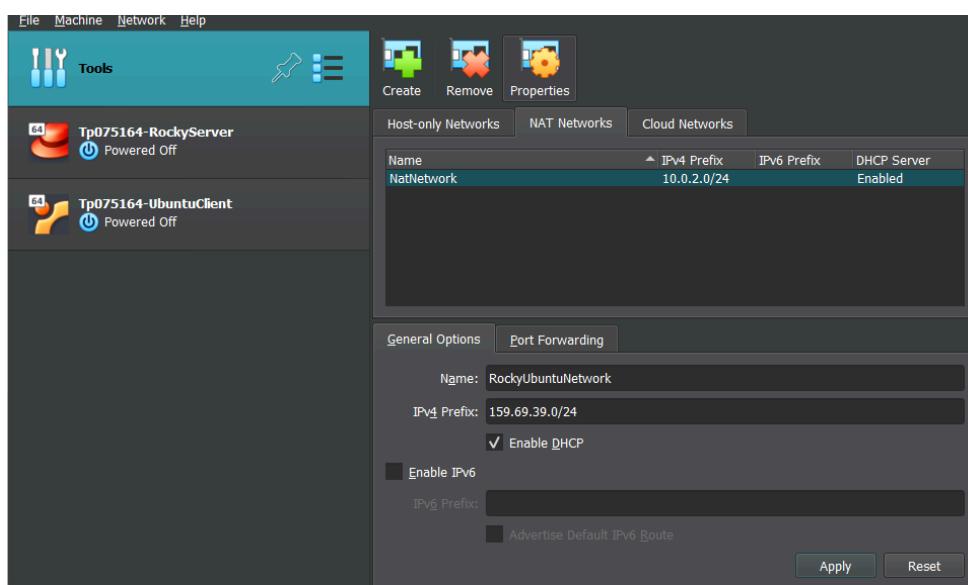
```
[kashsna@localhost ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
                inet6 fe80::a00:27ff:fe5a:be14  prefixlen 64  scopeid 0x20<link>
                    ether 08:00:27:5a:be:14  txqueuelen 1000  (Ethernet)
                        RX packets 9255  bytes 13569538 (12.9 MiB)
                        RX errors 0  dropped 0  overruns 0  frame 0
                        TX packets 943  bytes 69450 (67.8 KiB)
                        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
                inet6 ::1  prefixlen 128  scopeid 0x10<host>
```

As we can see, the Rocky server has the same default IP address as the Ubuntu client. Since both systems cannot share the same IP, we need to change the IP addresses.

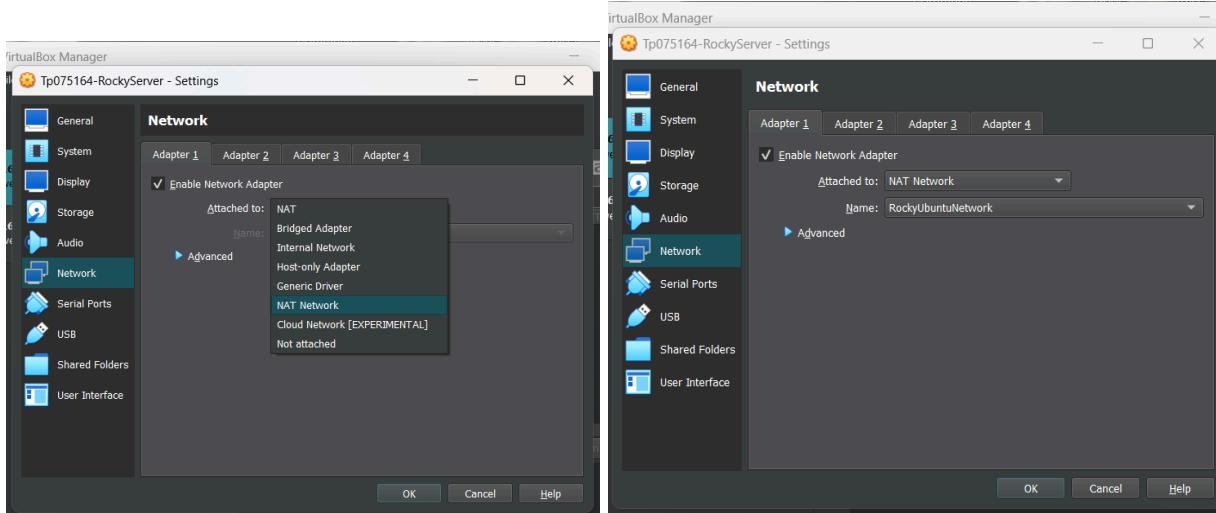
3. Creating a NAT Network in VirtualBox:

- Open **VirtualBox Manager**, go to **Tools**, and click on **NAT Networks**.
- Click **Create**. A box will appear at the bottom of the screen where you can rename the network. I named it **RockyUbuntuNetwork**.
- In the **IPv4 Prefix** box, specify the network subnet. I chose **159.69.39.0/24** to match my project's IP range.
- Enable the **DHCP** checkbox to automatically assign IP addresses to the systems.
- Once everything is set, click **Apply**.



4. Assigning NAT Network to VMs:

- Click on the Rocky Linux server in VirtualBox, go to **Settings**, and then click **Network**.
- Under **Adapter 1**, select **NAT Network** in the **Attached to** dropdown. Ensure that the network name is set to **RockyUbuntuNetwork**. Click **OK**.
- Repeat the same steps for the Ubuntu client.



5. Verifying IP Addresses:

Start the Rocky Linux and Ubuntu client and run **ifconfig** to check the updated IP address:

```
[kashsna@static ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 159.69.39.4 netmask 255.255.255.0 broadcast 159.69.39.255
        ...
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0

kashubuntu@kashubuntu-VirtualBox:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 159.69.39.5 netmask 255.255.255.0 broadcast 159.69.39.255
        ...
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
```

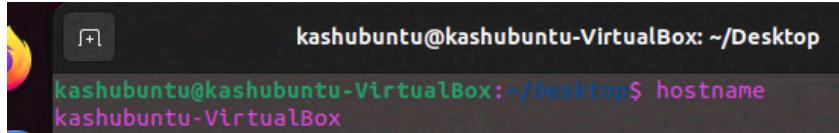
- The IP address in Rocky should now be **159.69.39.4/24**.
- The Ubuntu client should have a different IP, such as **159.69.39.5/24**.

Now, both systems are on the same network but have different IP addresses, which is essential for network communication.

3.0 Setting hostnames for both VMs

Setting Ubuntu Hostname:

- Open Ubuntu and check the current hostname by running `hostname`:



```
kashubuntu@kashubuntu-VirtualBox: ~/Desktop
kashubuntu@kashubuntu-VirtualBox:~/Desktop$ hostname
kashubuntu-VirtualBox
```

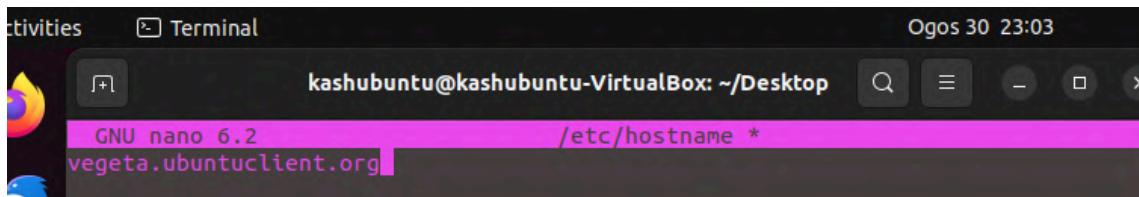
The current hostname is `kashubuntu-VirtualBox`. We will change this to a fully qualified domain name (FQDN).

To do this, edit the `/etc/hostname` file:



```
kashubuntu@kashubuntu-VirtualBox:~/Desktop$ sudo nano /etc/hostname
[sudo] password for kashubuntu: [REDACTED]
```

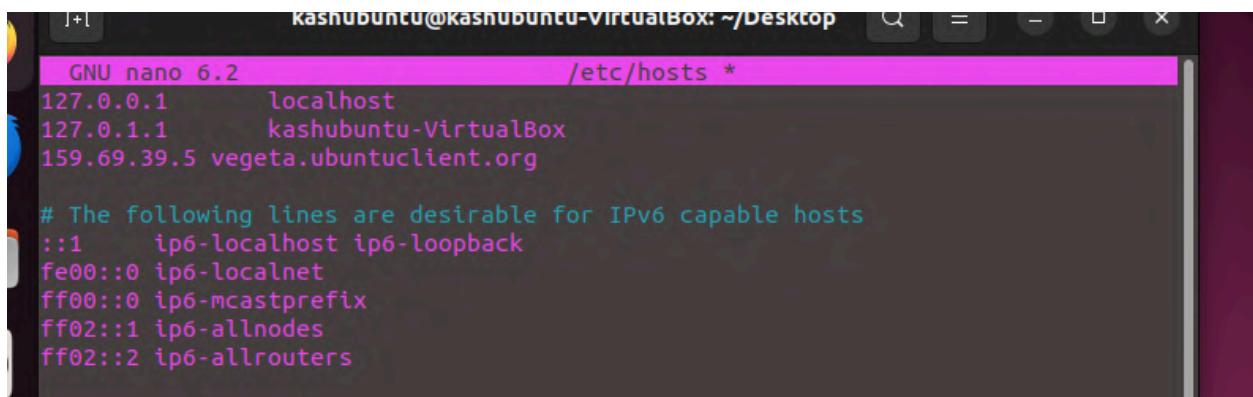
Replace the existing hostname with the desired FQDN. In this case, I used `vegeta.ubuntuclient.org`. Save the file by pressing **SHIFT + X**, then **Y**, and **Enter**.



```
Activities Terminal Ogos 30 23:03
kashubuntu@kashubuntu-VirtualBox: ~/Desktop
GNU nano 6.2 /etc/hostname *
vegeta.ubuntuclient.org
```

2. Updating the Hosts File in Ubuntu:

- Next, update the `/etc/hosts` file
- Add the IP address of the Ubuntu client followed by the new FQDN:
 - `159.69.39.5 vegeta.ubuntuclient.org`



```
J+ Kashubuntu@kashubuntu-VirtualBox: ~/Desktop
GNU nano 6.2 /etc/hosts *
127.0.0.1 localhost
127.0.1.1 kashubuntu-VirtualBox
159.69.39.5 vegeta.ubuntuclient.org

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

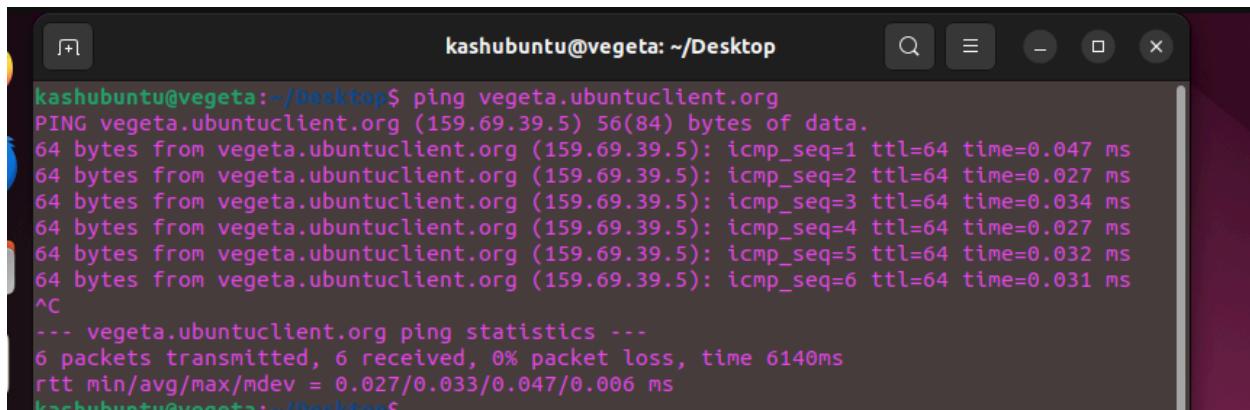
Save and exit using the same steps as before.

3. Reboot and Test Hostname in Ubuntu:

- Reboot the system:

Ping vegeta/ubuntuclient.org

We should receive replies from the client, confirming that the hostname has been successfully configured. Use **CTRL + C** to stop the ping.

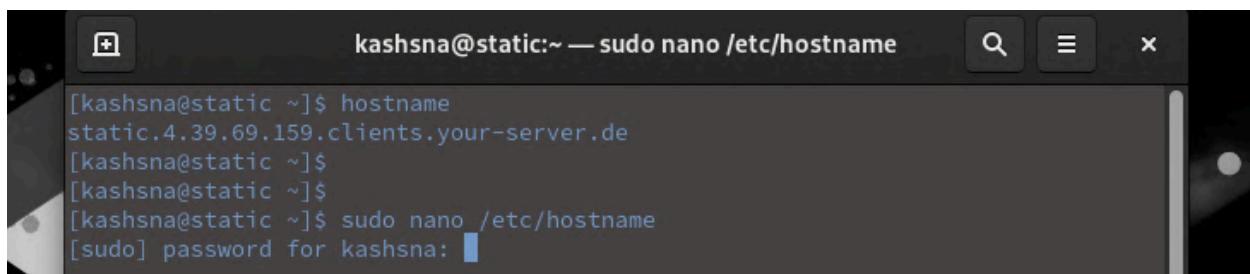


```
kashubuntu@vegeta:~/Desktop$ ping vegeta/ubuntuclient.org
PING vegeta/ubuntuclient.org (159.69.39.5) 56(84) bytes of data.
64 bytes from vegeta/ubuntuclient.org (159.69.39.5): icmp_seq=1 ttl=64 time=0.047 ms
64 bytes from vegeta/ubuntuclient.org (159.69.39.5): icmp_seq=2 ttl=64 time=0.027 ms
64 bytes from vegeta/ubuntuclient.org (159.69.39.5): icmp_seq=3 ttl=64 time=0.034 ms
64 bytes from vegeta/ubuntuclient.org (159.69.39.5): icmp_seq=4 ttl=64 time=0.027 ms
64 bytes from vegeta/ubuntuclient.org (159.69.39.5): icmp_seq=5 ttl=64 time=0.032 ms
64 bytes from vegeta/ubuntuclient.org (159.69.39.5): icmp_seq=6 ttl=64 time=0.031 ms
^C
--- vegeta/ubuntuclient.org ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 6140ms
rtt min/avg/max/mdev = 0.027/0.033/0.047/0.006 ms
kashubuntu@vegeta:~/Desktop$
```

Setting Rocky Server Hostname:

1. Setting Rocky Linux Hostname:

- Repeat the same steps on the Rocky Linux server.
- Check the current hostname



```
[kashsna@static ~]$ hostname
static.4.39.69.159.clients.your-server.de
[kashsna@static ~]$
[kashsna@static ~]$
[kashsna@static ~]$ sudo nano /etc/hostname
[sudo] password for kashsna: [REDACTED]
```

Edit the **/etc/hostname** file to change the hostname to **goku.rockyserver.org**:

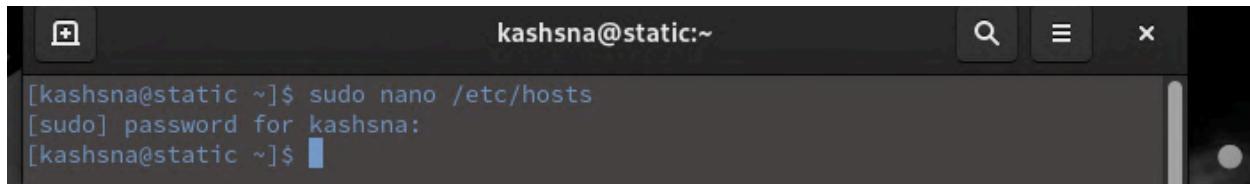


```
GNU nano 5.6.1          /etc/hostname          Modified
goku.rockyserver.org
```

Save and exit.

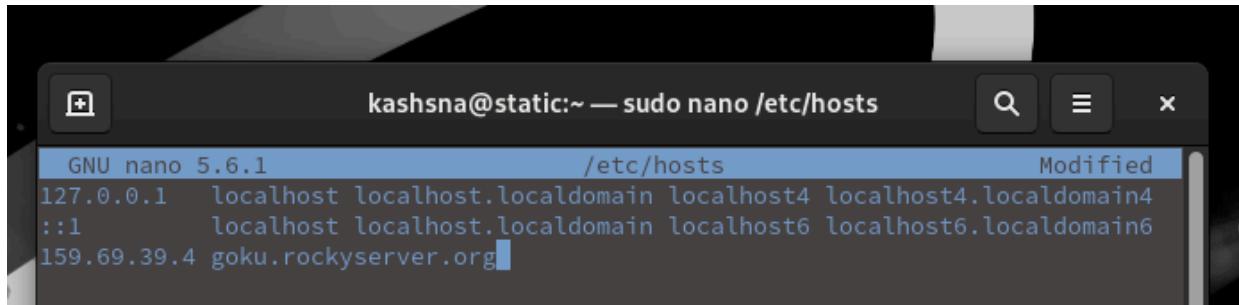
Updating the Hosts File in Rocky Linux:

- Similarly, update the **/etc/hosts** file to map the Rocky server's IP address to its FQDN:



```
[kashsna@static ~]$ sudo nano /etc/hosts
[sudo] password for kashsna:
[kashsna@static ~]$
```

Add the following line: **159.69.39.4 goku.rockyserver.org**



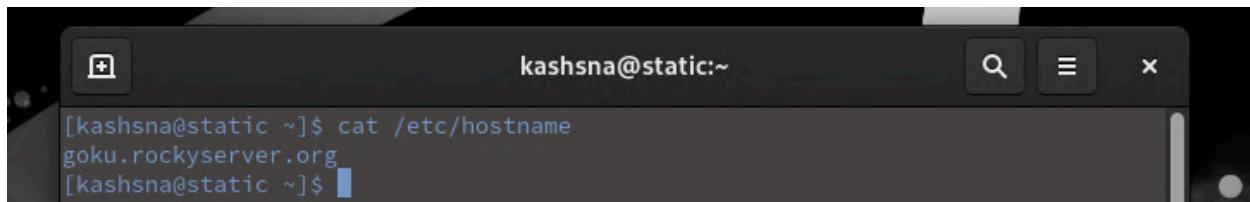
```
GNU nano 5.6.1           /etc/hosts          Modified
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
159.69.39.4 goku.rockyserver.org
```

Save and exit.

Reboot and Test Hostname in Rocky Linux:

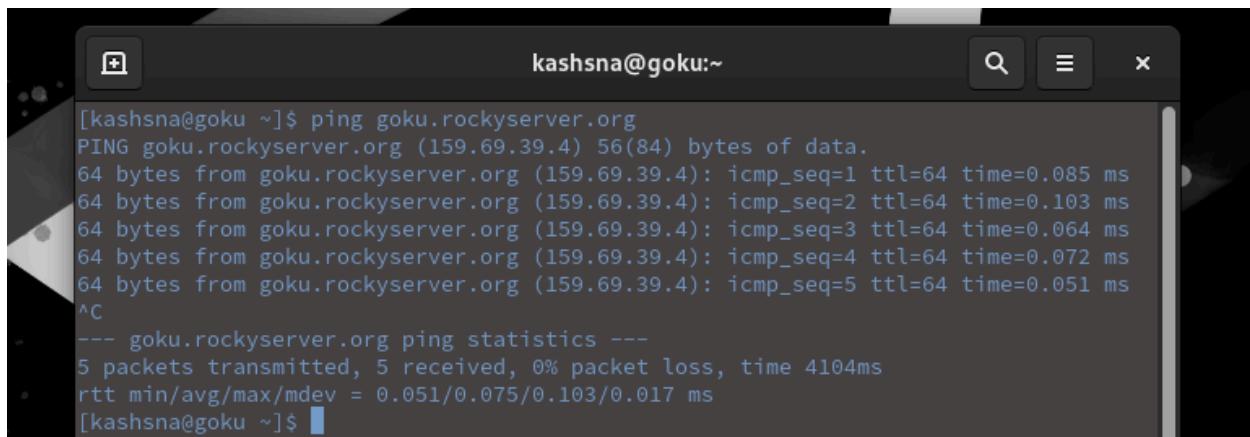
- Reboot the Rocky Linux server: **reboot**

After the reboot, check if the hostname has been saved:



```
[kashsna@static ~]$ cat /etc/hostname
goku.rockyserver.org
[kashsna@static ~]$
```

Now, test the hostname by pinging it



```
[kashsna@goku ~]$ ping goku.rockyserver.org
PING goku.rockyserver.org (159.69.39.4) 56(84) bytes of data.
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=1 ttl=64 time=0.085 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=2 ttl=64 time=0.103 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=3 ttl=64 time=0.064 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=4 ttl=64 time=0.072 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=5 ttl=64 time=0.051 ms
^C
--- goku.rockyserver.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4104ms
rtt min/avg/max/mdev = 0.051/0.075/0.103/0.017 ms
[kashsna@goku ~]$
```

As we can see, all packets are being transmitted and received without any loss. Therefore the hostname configuration was successful. Use **CTRL + C** to break the ping.

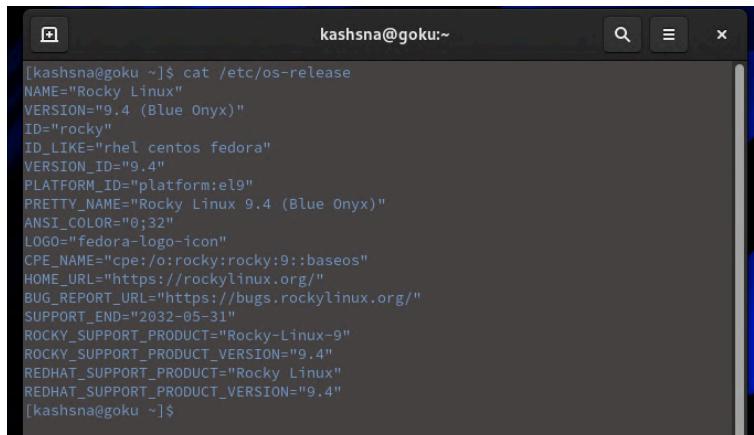
4.0 Setting Static IP Addresses:

Before configuring the DNS server, it's crucial to set up static IP addresses for both **Rocky Linux** and **Ubuntu**. Static IP addresses ensure that the systems retain the same IP address after reboots, which is essential for reliable network service communication.

Rocky:

1. Verify Operating System:

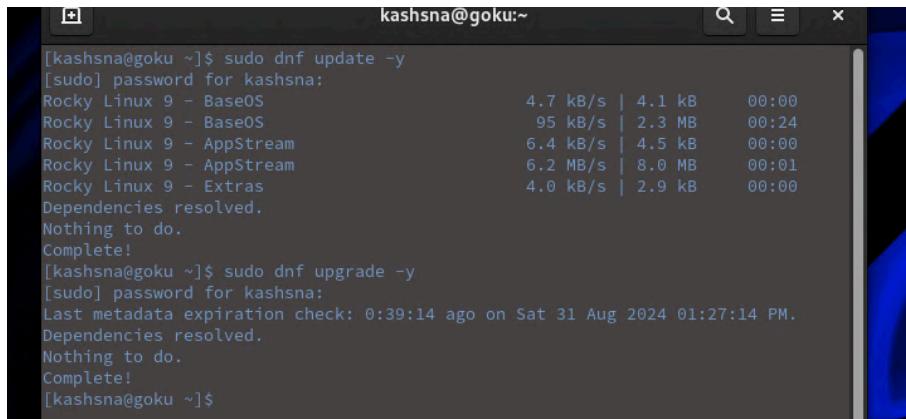
- Start by checking the operating system on the Rocky Linux server to ensure you're working on the correct machine. I used `cat /etc/os-release`



```
[kashsna@goku ~]$ cat /etc/os-release
NAME="Rocky Linux"
VERSION="9.4 (Blue Onyx)"
ID="rocky"
ID_LIKE="rhel centos fedora"
VERSION_ID="9.4"
PLATFORM_ID="platform:el9"
PRETTY_NAME="Rocky Linux 9.4 (Blue Onyx)"
ANSI_COLOR="#32CD32"
LOGO="fedora-logo-icon"
CPE_NAME="cpe:/o:rocky:rocky:9::baseos"
HOME_URL="https://rockylinux.org/"
BUG_REPORT_URL="https://bugs.rockylinux.org/"
SUPPORT_END="2032-05-31"
ROCKY_SUPPORT_PRODUCT="Rocky-Linux-9"
ROCKY_SUPPORT_PRODUCT_VERSION="9.4"
REDHAT_SUPPORT_PRODUCT="Rocky Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="9.4"
[kashsna@goku ~]$
```

2. Update and Upgrade the System:

- Before proceeding with IP configuration, it's best to update and upgrade the system:



```
[kashsna@goku ~]$ sudo dnf update -y
[sudo] password for kashsna:
Rocky Linux 9 - BaseOS          4.7 kB/s | 4.1 kB    00:00
Rocky Linux 9 - BaseOS          95 kB/s | 2.3 MB   00:24
Rocky Linux 9 - AppStream        6.4 kB/s | 4.5 kB    00:00
Rocky Linux 9 - AppStream        6.2 MB/s | 8.0 MB   00:01
Rocky Linux 9 - Extras          4.0 kB/s | 2.9 kB    00:00
Dependencies resolved.
Nothing to do.
Complete!
[kashsna@goku ~]$ sudo dnf upgrade -y
[sudo] password for kashsna:
Last metadata expiration check: 0:39:14 ago on Sat 31 Aug 2024 01:27:14 PM.
Dependencies resolved.
Nothing to do.
Complete!
[kashsna@goku ~]$
```

After the update is complete, reboot the system to apply any necessary changes:



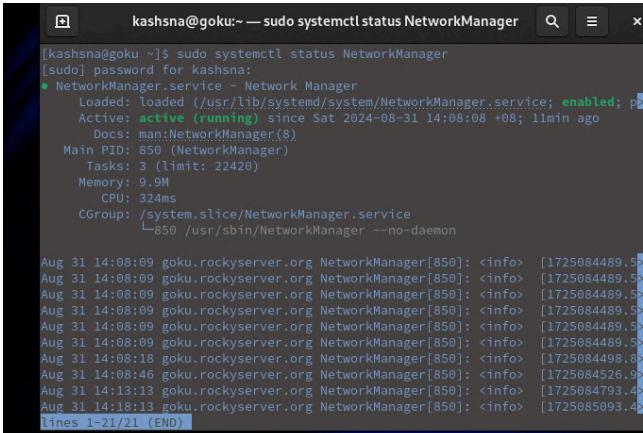
```
[kashsna@goku ~]$ sudo reboot
```

3. Check and Enable NetworkManager:

- To configure the static IP, **NetworkManager** must be installed and running. NetworkManager is a tool that manages network connections and devices.
- Check if NetworkManager is installed with: `sudo rpm -q NetworkManager`

Enable and start the service if it's not already running. Use `systemctl status` to ensure it is running:

1. `sudo systemctl enable NetworkManager`
2. `sudo systemctl start NetworkManager`
3. `sudo systemctl status NetworkManager`



The screenshot shows a terminal window with two panes. The top pane displays the command `sudo systemctl status NetworkManager` and its output, which shows the service is active (running) since the previous day. The bottom pane shows the command `sudo rpm -q NetworkManager` and its output, confirming the package is installed.

```
[kashsna@goku ~]$ sudo systemctl status NetworkManager
[sudo] password for kashsna:
● NetworkManager.service - Network Manager
  Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; pre
  Active: active (running) since Sat 2024-08-31 14:08:08 +08; 1min ago
    Docs: man:NetworkManager(8)
    Main PID: 850 (NetworkManager)
      Tasks: 3 (limit: 22420)
        Memory: 9.9M
          CPU: 324ms
        CGroup: /system.slice/NetworkManager.service
           └─ 850 /usr/sbin/NetworkManager --no-daemon

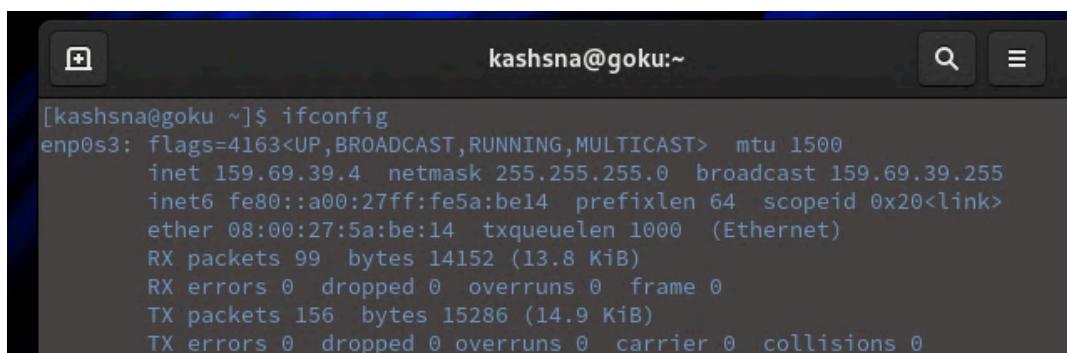
Aug 31 14:08:09 goku.rockyserver.org NetworkManager[850]: <info> [1725084489.5]
Aug 31 14:08:18 goku.rockyserver.org NetworkManager[850]: <info> [1725084498.8]
Aug 31 14:08:46 goku.rockyserver.org NetworkManager[850]: <info> [1725084526.9]
Aug 31 14:13:13 goku.rockyserver.org NetworkManager[850]: <info> [1725084793.4]
Aug 31 14:18:13 goku.rockyserver.org NetworkManager[850]: <info> [1725085093.4]
lines 1-21/21 (END)
```

```
[kashsna@goku ~]$ sudo rpm -q NetworkManager
[sudo] password for kashsna:
NetworkManager-1.46.0-13.el9_4.x86_64
[kashsna@goku ~]$ sudo systemctl enable NetworkManager
[kashsna@goku ~]$ sudo systemctl start NetworkManager
[kashsna@goku ~]$
```

If the service is running successfully, we can proceed with configuring the static IP.

4. Check Current Network Configuration:

- Use `ifconfig` to display the current IP address and the network interface details:



The screenshot shows a terminal window displaying the output of the `ifconfig` command. It lists the `enp0s3` interface with its IP address (159.69.39.4), netmask (255.255.255.0), broadcast address (159.69.39.255), and other configuration details like MAC address and queueing discipline.

```
[kashsna@goku ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
          inet 159.69.39.4  netmask 255.255.255.0  broadcast 159.69.39.255
              inet6 fe80::a00:27ff:fe5a:be14  prefixlen 64  scopeid 0x20<link>
                ether 08:00:27:5a:be:14  txqueuelen 1000  (Ethernet)
                  RX packets 99  bytes 14152 (13.8 KiB)
                  RX errors 0  dropped 0  overruns 0  frame 0
                  TX packets 156  bytes 15286 (14.9 KiB)
                  TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Note down the IP address (e.g., **159.69.39.4**) and the network interface name (e.g., **enp0s3**). We will use these details in the upcoming steps.

5. Check System Network Information:

For additional information on the network configuration, I use these commands:

1. `sudo nmcli general status`
2. `sudo nmcli device status`

```
[kashsna@goku ~]$ sudo nmcli general status
[sudo] password for kashsna:
STATE      CONNECTIVITY WIFI-HW  WIFI      WWAN-HW  WWAN
connected   full        missing  enabled   missing  enabled
[kashsna@goku ~]$ sudo nmcli device status
DEVICE    TYPE      STATE           CONNECTION
enp0s3    ethernet  connected      enp0s3
lo        loopback  connected (externally)  lo
[kashsna@goku ~]$
```

6. Identify Network Connection Name:

To identify the network connection file that we will modify to set the static IP, run:

```
[kashsna@goku ~]$ sudo ls -l /etc/NetworkManager/system-connections/
[sudo] password for kashsna:
total 4
-rw-----. 1 root root 229 Aug 28 17:37 enp0s3.nmconnection
[kashsna@goku ~]$
```

Look for the file that corresponds to your network interface (in my case, **enp0s3.nmconnection**). This file contains the configuration settings for the network interface card (NIC), which manages network and internet connectivity.

6. Configure Static IP Address:

Open the connection file for editing:

```
[root@goku ~]# nano /etc/NetworkManager/system-connections/enp0s3.nmconnection
```

Locate the **[ipv4]** section and modify or add the following details:

```
[ipv4]
address=159.69.39.4/24,159.69.39.1
dns=8.8.8.8;8.8.4.4;159.69.39.4
ignore-auto-dns=true
method=manual
```

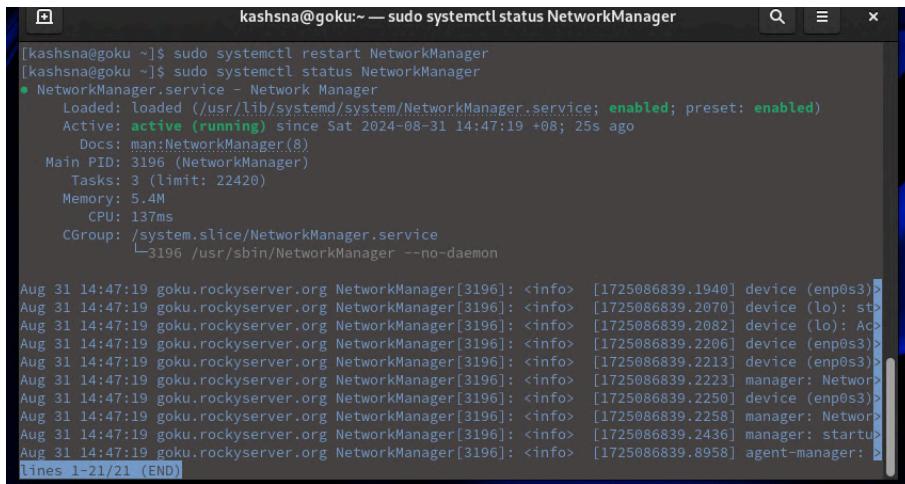
Explanation of the IP configuration:

- **159.69.39.4/24** is the static IP address for your server with a subnet mask of **255.255.255.0** (indicated by **/24**).
- **159.69.39.1** is the gateway (or router) that connects your network to the internet.
- **dns=8.8.8.8;8.8.4.4;159.69.39.4** specifies the DNS servers. In this case, the first two DNS addresses are Google DNS servers (8.8.8.8 and 8.8.4.4), and the last is the IP address of your Rocky Linux server.

8. Restart NetworkManager:

After making changes to the network settings, restart **NetworkManager** to apply the new configuration:

- `sudo systemctl restart NetworkManager`
- `sudo systemctl status NetworkManager`

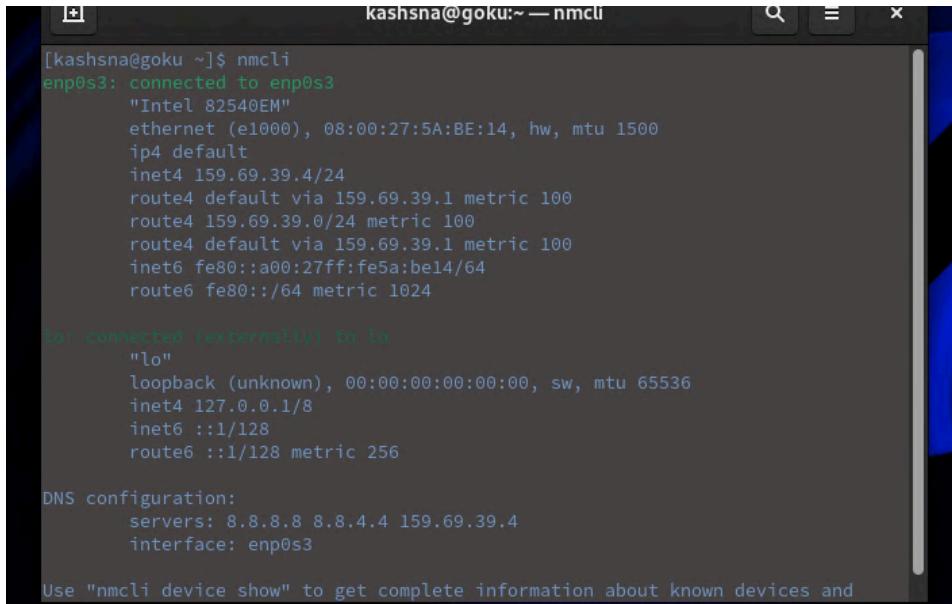


```
[kashsna@goku ~]$ sudo systemctl restart NetworkManager
[kashsna@goku ~]$ sudo systemctl status NetworkManager
● NetworkManager.service - Network Manager
  Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; preset: enabled)
  Active: active (running) since Sat 2024-08-31 14:47:19 +08; 25s ago
    Docs: man:NetworkManager(8)
   Main PID: 3196 (NetworkManager)
     Tasks: 3 (limit: 22420)
    Memory: 5.4M
       CPU: 137ms
      CGroup: /system.slice/NetworkManager.service
              └─ 3196 /usr/sbin/NetworkManager --no-daemon

Aug 31 14:47:19 goku.rockyserver.org NetworkManager[3196]: <info> [1725086839.1940] device (enp0s3)>
Aug 31 14:47:19 goku.rockyserver.org NetworkManager[3196]: <info> [1725086839.2070] device (lo): st>
Aug 31 14:47:19 goku.rockyserver.org NetworkManager[3196]: <info> [1725086839.2082] device (lo): Ac>
Aug 31 14:47:19 goku.rockyserver.org NetworkManager[3196]: <info> [1725086839.2206] device (enp0s3)>
Aug 31 14:47:19 goku.rockyserver.org NetworkManager[3196]: <info> [1725086839.2213] device (enp0s3)>
Aug 31 14:47:19 goku.rockyserver.org NetworkManager[3196]: <info> [1725086839.2223] manager: Netw>
Aug 31 14:47:19 goku.rockyserver.org NetworkManager[3196]: <info> [1725086839.2250] device (enp0s3)>
Aug 31 14:47:19 goku.rockyserver.org NetworkManager[3196]: <info> [1725086839.2258] manager: Netw>
Aug 31 14:47:19 goku.rockyserver.org NetworkManager[3196]: <info> [1725086839.2436] manager: startu>
Aug 31 14:47:19 goku.rockyserver.org NetworkManager[3196]: <info> [1725086839.8958] agent-manager: >
lines 1-21/21 (END)
```

9. Verify IP Configuration:

- Use `nmcli` to check if the new static IP has been applied:



```
[kashsna@goku ~]$ nmcli
enp0s3: connected to enp0s3
  "Intel 82540EM"
  ethernet (e1000), 08:00:27:5A:BE:14, hw, mtu 1500
  ip4 default
  inet4 159.69.39.4/24
  route4 default via 159.69.39.1 metric 100
  route4 159.69.39.0/24 metric 100
  route4 default via 159.69.39.1 metric 100
  inet6 fe80::a00:27ff:fe5a:be14%enp0s3
  route6 fe80::/64 metric 1024

lo: connected (externally) to lo
  "lo"
  loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
  inet4 127.0.0.1/8
  inet6 ::1/128
  route6 ::1/128 metric 256

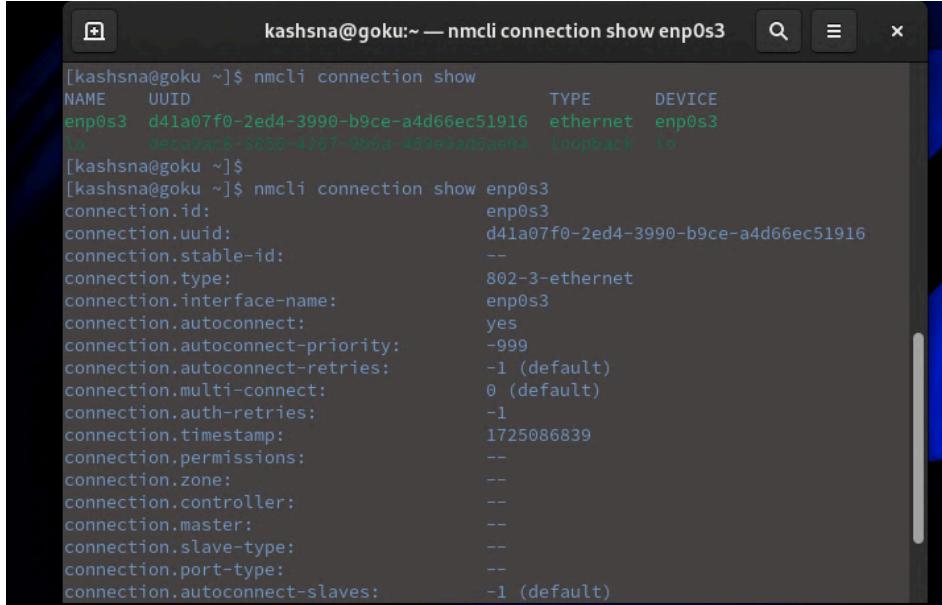
DNS configuration:
  servers: 8.8.8.8 8.8.4.4 159.69.39.4
  interface: enp0s3

Use "nmcli device show" to get complete information about known devices and
```

10. Additional Network Checks:

You can further inspect network settings with the following commands

1. *nmcli connection show*
2. *nmcli connection show enp0s3* (may differ based on the systems network interface name)

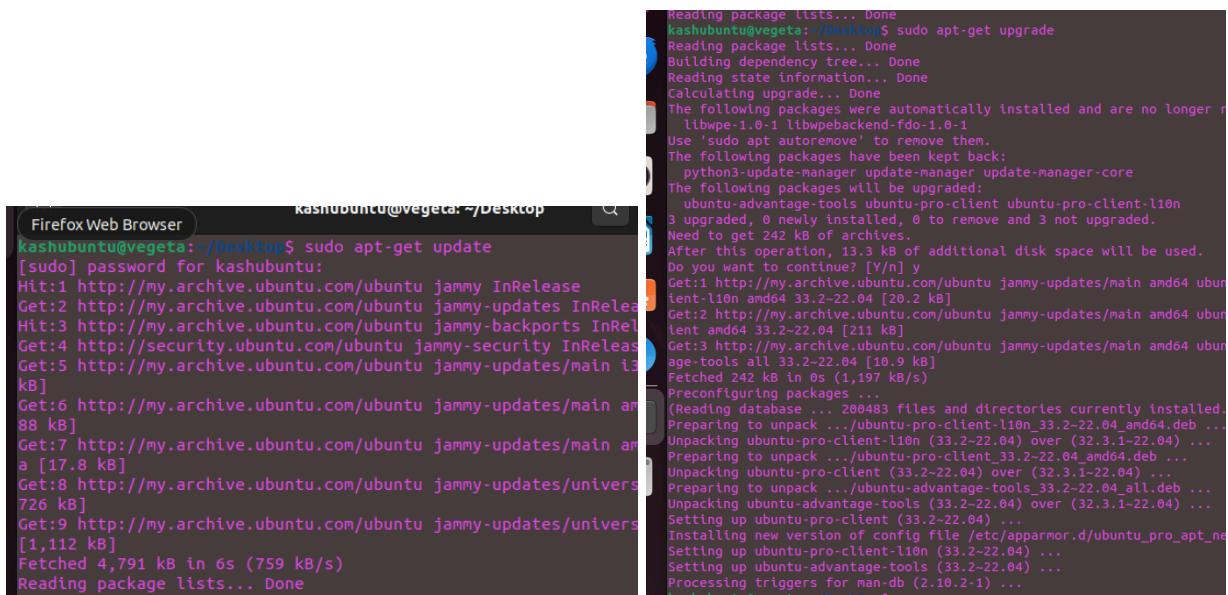


```
[kashsna@goku ~]$ nmcli connection show enp0s3
NAME UUID           TYPE      DEVICE
enp0s3 d41a07f0-2ed4-3990-b9ce-a4d66ec51916 ethernet enp0s3
lo     deca9a8-3356-4267-9b5a-499e93d6aa04 loopback lo
[kashsna@goku ~]$ [kashsna@goku ~]$ nmcli connection show enp0s3
connection.id:          enp0s3
connection.uuid:         d41a07f0-2ed4-3990-b9ce-a4d66ec51916
connection.stable-id:    --
connection.type:         802-3-ethernet
connection.interface-name: enp0s3
connection.autoconnect:   yes
connection.autoconnect-priority: -999
connection.autoconnect-retries: -1 (default)
connection.multi-connect: 0 (default)
connection.auth-retries:  -1
connection.timestamp:    1725086839
connection.permissions:  --
connection.zone:         --
connection.controller:   --
connection.master:       --
connection.slave-type:   --
connection.port-type:    --
connection.autoconnect-slaves: -1 (default)
```

Ubuntu:

1. Update and Upgrade the System:

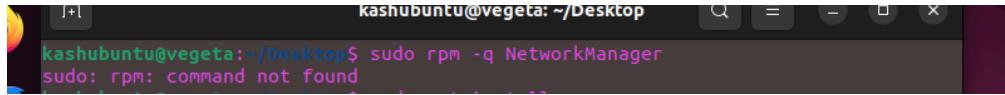
Before configuring a static IP, it's always a good practice to update and upgrade the system:



```
kashubuntu@vegeta:~/Desktop$ sudo apt-get update
[sudo] password for kashubuntu:
Hit:1 http://my.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://my.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://my.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 http://my.archive.ubuntu.com/ubuntu jammy-updates/main i386
  [88 kB]
Get:6 http://my.archive.ubuntu.com/ubuntu jammy-updates/main armhf
  [17.8 kB]
Get:8 http://my.archive.ubuntu.com/ubuntu jammy-updates/universe
  [726 kB]
Get:9 http://my.archive.ubuntu.com/ubuntu jammy-updates/universe
  [1,112 kB]
Fetched 4,791 kB in 6s (759 kB/s)
Reading package lists... Done
kashubuntu@vegeta:~/Desktop$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following packages have been kept back:
  python3-update-manager update-manager update-manager-core
The following packages will be upgraded:
  ubuntu-advantage-tools ubuntu-pro-client ubuntu-pro-client-l10n
3 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Need to get 242 kB of archives.
After this operation, 13.3 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://my.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ubuntu-l10n amd64 33.2-22.04 [20.2 kB]
Get:2 http://my.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ubuntu-l10n amd64 33.2-22.04 [211 kB]
Get:3 http://my.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ubuntu-advantage-tools all 33.2-22.04 [10.9 kB]
Fetched 242 kB in 0s (1,197 kB/s)
Preconfiguring packages ...
(Reading database ... 200483 files and directories currently installed.)
Preparing to unpack .../ubuntu-pro-client-l10n_33.2-22.04_amd64.deb ...
Unpacking ubuntu-pro-client-l10n (33.2-22.04) over (32.3.1-22.04) ...
Preparing to unpack .../ubuntu-pro-client_33.2-22.04_amd64.deb ...
Unpacking ubuntu-pro-client (33.2-22.04) over (32.3.1-22.04) ...
Preparing to unpack .../ubuntu-advantage-tools_33.2-22.04_all.deb ...
Unpacking ubuntu-advantage-tools (33.2-22.04) over (32.3.1-22.04) ...
Setting up ubuntu-pro-client (33.2-22.04) ...
Installing new version of config file /etc/apparmor.d/ubuntu_pro_apt_ne
Setting up ubuntu-pro-client-l10n (33.2-22.04) ...
Setting up ubuntu-advantage-tools (33.2-22.04) ...
Processing triggers for man-db (2.10.2-1) ...
```

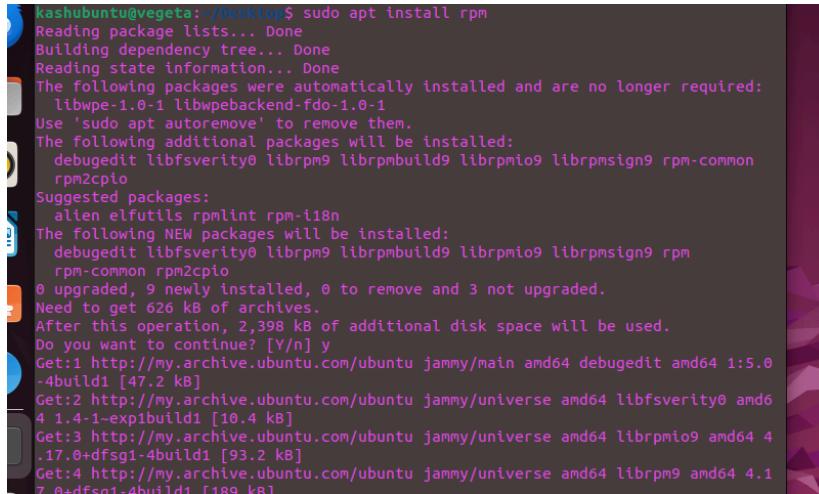
2. Check and Install NetworkManager:

- In Ubuntu, we use **NetworkManager** to manage network settings. First, check if NetworkManager is installed:



```
kashubuntu@vegeta:~/Desktop$ sudo rpm -q NetworkManager
sudo: rpm: command not found
```

If you encounter the error "**sudo: rpm: command not found**", it means that **rpm** (which is commonly used in Red Hat-based systems like Rocky Linux) is not installed on Ubuntu. To resolve this, install the **rpm** package:

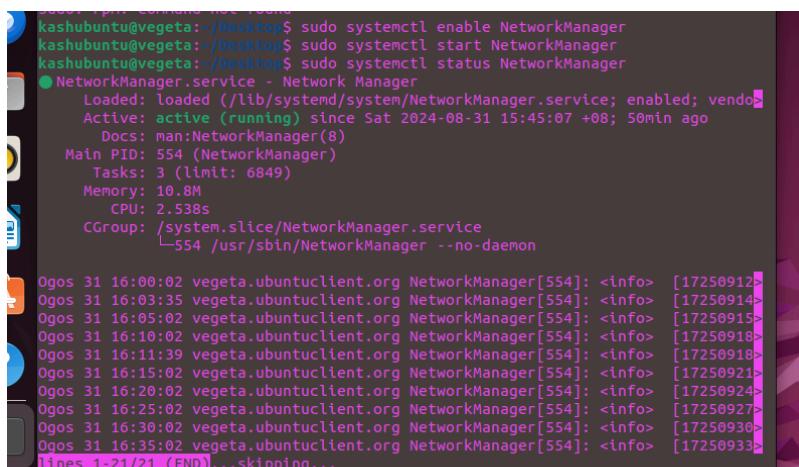


```
kashubuntu@vegeta:~/Desktop$ sudo apt install rpm
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  debugedit libfsverity0 librpm9 librpmbuild9 librpmio9 librpmssign9 rpm-common
  rpm2cpio
Suggested packages:
  alien eutils rpmlint rpm-i18n
The following NEW packages will be installed:
  debugedit libfsverity0 librpm9 librpmbuild9 librpmio9 librpmssign9 rpm
  rpm-common rpm2cpio
0 upgraded, 9 newly installed, 0 to remove and 3 not upgraded.
Need to get 626 kB of archives.
After this operation, 2,398 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://my.archive.ubuntu.com/ubuntu jammy/main amd64 debugedit amd64 1:5.0
-4build1 [47.2 kB]
Get:2 http://my.archive.ubuntu.com/ubuntu jammy/universe amd64 libfsverity0 amd6
4 1.4-1-exhibit1 [10.4 kB]
Get:3 http://my.archive.ubuntu.com/ubuntu jammy/universe amd64 librpmio9 amd64 4
.17.0+dfsg1-4build1 [93.2 kB]
Get:4 http://my.archive.ubuntu.com/ubuntu jammy/universe amd64 librpm9 amd64 4.1
7.0+dfsg1-4build1 [189 kB]
```

Once **rpm** is installed, proceed with enabling, starting, and checking the status of **NetworkManager**:

- sudo rpm -q NetworkManager**
- sudo systemctl enable NetworkManager**
- sudo systemctl start NetworkManager**
- sudo systemctl status NetworkManager**

These commands ensure that **NetworkManager** is running and ready for managing network settings.



```
kashubuntu@vegeta:~/Desktop$ sudo systemctl enable NetworkManager
kashubuntu@vegeta:~/Desktop$ sudo systemctl start NetworkManager
kashubuntu@vegeta:~/Desktop$ sudo systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor
   Active: active (running) since Sat 2024-08-31 15:45:07 +08; 50min ago
     Docs: man:NetworkManager(8)
     Main PID: 554 (NetworkManager)
        Tasks: 3 (limit: 6849)
       Memory: 10.8M
          CPU: 2.538s
        CGroup: /system.slice/NetworkManager.service
                   └─554 /usr/sbin/NetworkManager --no-daemon

Ogos 31 16:00:02 vegeta/ubuntuclient.org NetworkManager[554]: <info> [17250912>
Ogos 31 16:03:35 vegeta/ubuntuclient.org NetworkManager[554]: <info> [17250914>
Ogos 31 16:05:02 vegeta/ubuntuclient.org NetworkManager[554]: <info> [17250915>
Ogos 31 16:10:02 vegeta/ubuntuclient.org NetworkManager[554]: <info> [17250918>
Ogos 31 16:11:39 vegeta/ubuntuclient.org NetworkManager[554]: <info> [17250918>
Ogos 31 16:15:02 vegeta/ubuntuclient.org NetworkManager[554]: <info> [17250921>
Ogos 31 16:20:02 vegeta/ubuntuclient.org NetworkManager[554]: <info> [17250924>
Ogos 31 16:25:02 vegeta/ubuntuclient.org NetworkManager[554]: <info> [17250927>
Ogos 31 16:30:02 vegeta/ubuntuclient.org NetworkManager[554]: <info> [17250930>
Ogos 31 16:35:02 vegeta/ubuntuclient.org NetworkManager[554]: <info> [17250933>
lines 1-21/21 (END)... skipping...
```

3. Check the Current IP Address:

- Use the ***sudo nmcli*** command to view the current IP address and network details:

```
kashubuntu@vegeta:~/Desktop$ sudo nmcli
[sudo] password for kashubuntu:
enp0s3: connected to Wired connection 1
    "Intel 82540EM"
    ethernet (e1000), 08:00:27:71:7D:26, hw, mtu 1500
    ip4 default
    inet4 159.69.39.5/24
        route4 159.69.39.0/24 metric 100
        route4 169.254.0.0/16 metric 1000
        route4 default via 159.69.39.1 metric 100
    inet6 fe80::f478:f52c:8022:6d4a/64
        route6 fe80::/64 metric 1024

lo: unmanaged
    "lo"
    loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536

DNS configuration:
    servers: 192.168.0.1
    interface: enp0s3

Use "nmcli device show" to get complete information about known devices and
"nmcli connection show" to get an overview on active connection profiles.

Consult nmcli(1) and nmcli-examples(7) manual pages for complete usage details.
Lines 1-23... skipping...
```

4. Check Network and Device Information:

To check for other network settings and configurations, we use the following commands:

- Check the general network status: ***sudo nmcli general status***
- Check the status of network devices: ***sudo nmcli device status***
- List all available network connections: ***sudo connection show***

```
kashubuntu@vegeta:~/Desktop$ sudo nmcli general status
STATE      CONNECTIVITY WIFI-HW WIFI      WWAN-HW WWAN
connected   full        enabled  enabled  enabled  enabled
kashubuntu@vegeta:~/Desktop$ sudo nmcli device status
DEVICE      TYPE      STATE      CONNECTION
enp0s3      ethernet  connected  Wired connection 1
lo          loopback  unmanaged  --
kashubuntu@vegeta:~/Desktop$ nmcli connection show
NAME           UUID                                  TYPE      DEVICE
Wired connection 1 984577e1-60c1-3e49-a952-1715f01a62c9  ethernet  enp0s3
kashubuntu@vegeta:~/Desktop$
```

5. Identify the Network Connection Name:

The next step is to identify the connection file that corresponds to the network interface you want to configure. Use this command to list the connection files:

- ***Sudo ls -l /etc/NetworkManager/system-connections***

```
kashubuntu@vegeta:~/Desktop$ sudo ls -l /etc/NetworkManager/system-connections
total 0
```

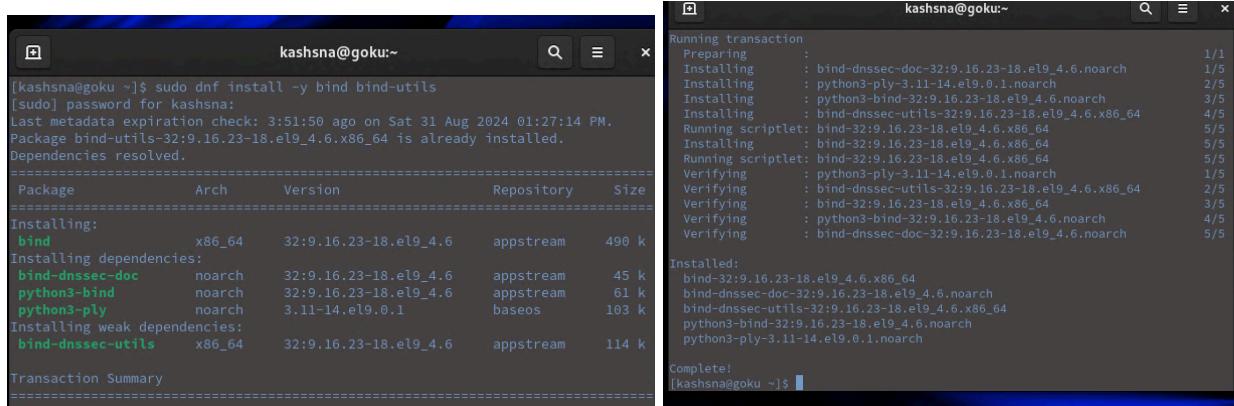
5.0 DNS

DNS Configuration in Rocky Linux

1. Installing DNS (BIND9):

First, we need to install **BIND9**, a widely used DNS server software, on the Rocky Linux system. Open the terminal and run the following command:

- ***sudo dnf install -y bind bind-utils***



```
[kashsna@goku ~]$ sudo dnf install -y bind bind-utils
[sudo] password for kashsna:
Last metadata expiration check: 3:51:50 ago on Sat 31 Aug 2024 01:27:14 PM.
Package bind-utils-32:9.16.23-18.el9_4.6.x86_64 is already installed.
Dependencies resolved.

=====
Package      Arch    Version       Repository  Size
=====
Installing:
bind          x86_64  32:9.16.23-18.el9_4.6   appstream   490 k
Installing dependencies:
bind-dnssec-doc noarch  32:9.16.23-18.el9_4.6   appstream   45 k
python3-bind   noarch  32:9.16.23-18.el9_4.6   appstream   61 k
python3-ply    noarch  3.11-14.el9_0.1        baseos     103 k
Installing weak dependencies:
bind-dnssec-utils x86_64  32:9.16.23-18.el9_4.6   appstream   114 k

Transaction Summary
=====

Running transaction
Preparing : bind-dnssec-doc-32:9.16.23-18.el9_4.6.noarch           1/1
Installing : python3-ply-3.11-14.el9_0.1.noarch                   2/5
Installing : python3-bind-32:9.16.23-18.el9_4.6.noarch             3/5
Installing : bind-dnssec-utils-32:9.16.23-18.el9_4.6.x86_64       4/5
Running scriptlet: bind-32:9.16.23-18.el9_4.6.x86_64            5/5
Installing : bind-32:9.16.23-18.el9_4.6.x86_64                  5/5
Running scriptlet: bind-32:9.16.23-18.el9_4.6.x86_64            5/5
Verifying  : python3-ply-3.11-14.el9_0.1.noarch                 1/5
Verifying  : bind-dnssec-utils-32:9.16.23-18.el9_4.6.x86_64      2/5
Verifying  : bind-32:9.16.23-18.el9_4.6.x86_64                  3/5
Verifying  : python3-bind-32:9.16.23-18.el9_4.6.noarch            4/5
Verifying  : bind-dnssec-doc-32:9.16.23-18.el9_4.6.noarch          5/5

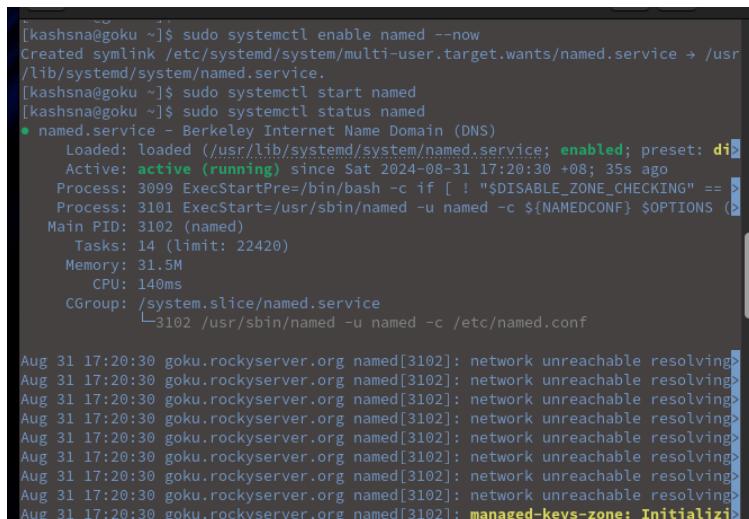
Installed:
bind-32:9.16.23-18.el9_4.6.x86_64
bind-dnssec-doc-32:9.16.23-18.el9_4.6.noarch
bind-dnssec-utils-32:9.16.23-18.el9_4.6.x86_64
python3-bind-32:9.16.23-18.el9_4.6.noarch
python3-ply-3.11-14.el9_0.1.noarch

Complete!
[kashsna@goku ~]$
```

2. Enable DNS to Start Automatically:

- To ensure that the DNS server starts automatically after every reboot, enable and start the **named** service (the DNS service):
 - ***sudo systemctl enable named --now***
 - ***sudo systemctl start named***
 - ***sudo systemctl status named*** (check the status of service)

The **--now** flag starts the service immediately after enabling it. ***systemctl status*** is used to confirm the service is running.



```
[kashsna@goku ~]$ sudo systemctl enable named --now
Created symlink /etc/systemd/system/multi-user.target.wants/named.service → /usr/lib/systemd/system/named.service.
[kashsna@goku ~]$ sudo systemctl start named
[kashsna@goku ~]$ sudo systemctl status named
● named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-08-31 17:20:30 +08; 35s ago
     Process: 3099 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" == >
     Process: 3101 ExecStart=/usr/sbin/named -u named -c ${NAMEDCONF} $OPTIONS <
   Main PID: 3102 (named)
     Tasks: 14 (limit: 22420)
    Memory: 31.5M
       CPU: 140ms
      CGroup: /system.slice/named.service
              └─3102 /usr/sbin/named -u named -c /etc/named.conf

Aug 31 17:20:30 goku.rockyserver.org named[3102]: network unreachable resolving
Aug 31 17:20:30 goku.rockyserver.org named[3102]: managed-kevs-zone: Initializi
```

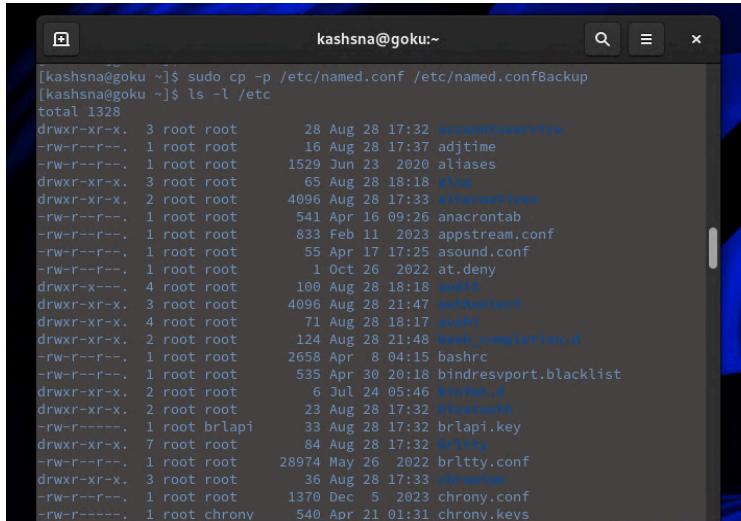
Verify the 'named.conf' File:

- The DNS configuration file is **named.conf**, located in the **/etc** directory. Use the following command to list the file:

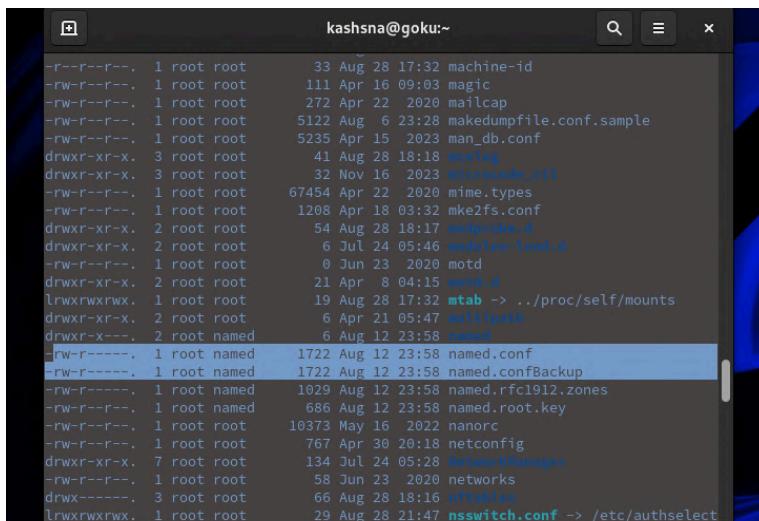
```
[kashsna@goku ~]$ ll /etc/named.conf
-rw-r-----. 1 root named 1722 Aug 12 23:58 /etc/named.conf
```

It's always good to create a backup of the file before making any changes. Run the following command:

- sudo cp -p /etc/named.conf /etc/named.confBackup**
 - The **-p** flag preserves the original file's permissions and ownership. You can verify both files are saved by listing the contents of the **/etc** directory.
- ls -l /etc**
 - list the contents of the etc directory to check if named.conf and named.confBackup are saved



```
[kashsna@goku ~]$ sudo cp -p /etc/named.conf /etc/named.confBackup
[kashsna@goku ~]$ ls -l /etc
total 1328
drwxr-xr-x. 3 root root      28 Aug 28 17:32 accountservice
-rw-r--r--. 1 root root     16 Aug 28 17:37 adjtime
-rw-r--r--. 1 root root   1529 Jun 23 2020 aliases
drwxr-xr-x. 3 root root     65 Aug 28 18:18 alsa
drwxr-xr-x. 2 root root   4096 Aug 28 17:33 alternatives
-rw-r--r--. 1 root root    541 Apr 16 09:26 anacrontab
-rw-r--r--. 1 root root   833 Feb 11 2023 appstream.conf
-rw-r--r--. 1 root root    55 Apr 17 17:25 asound.conf
-rw-r--r--. 1 root root     1 Oct 26 2022 at.deny
drwxr-xr-x. 4 root root   100 Aug 28 18:18 audit
drwxr-xr-x. 3 root root   4096 Aug 28 21:47 authselect
drwxr-xr-x. 4 root root    71 Aug 28 18:17 avahi
drwxr-xr-x. 2 root root   124 Aug 28 21:48 bash_completion.d
-rw-r--r--. 1 root root   2658 Apr  8 04:15 bashrc
-rw-r--r--. 1 root root   535 Apr 30 20:18 bindresvport.blacklist
drwxr-xr-x. 2 root root     6 Jul 24 05:46 binfmt.d
drwxr-xr-x. 2 root root   23 Aug 28 17:32 bluetooth
-rw-r-----. 1 root brlapi   33 Aug 28 17:32 brlapi.key
drwxr-xr-x. 7 root root   84 Aug 28 17:32 brltty
-rw-r--r--. 1 root root  28974 May 26 2022 brltty.conf
drwxr-xr-x. 3 root root   36 Aug 28 17:33 chromium
-rw-r--r--. 1 root root  1370 Dec  5 2023 chromium.conf
-rw-r-----. 1 root chromium 540 Apr 21 01:31 chromium.keys
```



```
[kashsna@goku ~]$ ls -l /etc
total 1328
-r--r--r--. 1 root root      33 Aug 28 17:32 machine-id
-rw-r--r--. 1 root root     111 Apr 16 09:03 magic
-rw-r--r--. 1 root root    272 Apr 22 2020 mailcap
-rw-r--r--. 1 root root   5122 Aug  6 23:28 makedumpfile.conf.sample
-rw-r--r--. 1 root root   5235 Apr 15 2023 man_db.conf
drwxr-xr-x. 3 root root     41 Aug 28 18:18 misc
drwxr-xr-x. 3 root root    32 Nov 16 2023 microcode_ctl
-rw-r--r--. 1 root root  67454 Apr 22 2020 mime.types
-rw-r--r--. 1 root root   1208 Apr 18 03:32 mke2fs.conf
drwxr-xr-x. 2 root root     54 Aug 28 18:17 modprobe.d
drwxr-xr-x. 2 root root     6 Jul 24 05:46 modules-load.d
-rw-r--r--. 1 root root     0 Jun 23 2020 motd
drwxr-xr-x. 2 root root    21 Apr  8 04:15 mount.d
lrwxrwxrwx. 1 root root    19 Aug 28 17:32 mtab -> ./proc/self/mounts
drwxr-xr-x. 2 root root     6 Apr 21 05:47 multipath
drwxr-x---. 2 root named   6 Aug 12 23:58 named
drw-r-----. 1 root named  1722 Aug 12 23:58 named.conf
-rw-r-----. 1 root named  1722 Aug 12 23:58 named.confBackup
-rw-r-----. 1 root named 1029 Aug 12 23:58 named.rfc1912.zones
-rw-r-----. 1 root named   686 Aug 12 23:58 named.root.key
-rw-r-----. 1 root root  10373 May 16 2022 nanorc
-rw-r--r--. 1 root root   767 Apr 30 20:18 netconfig
drwxr-xr-x. 7 root root   134 Jul 24 05:28 NetworkManager
-rw-r--r--. 1 root root   58 Jun 23 2020 networks
drwxr-xr-x. 3 root root   66 Aug 28 18:16 nftables
lrwxrwxrwx. 1 root root  29 Aug 28 21:47 nsswitch.conf -> /etc/authselect
```

As we can see, the backup files have been created.

4. Edit the 'named.conf' File:

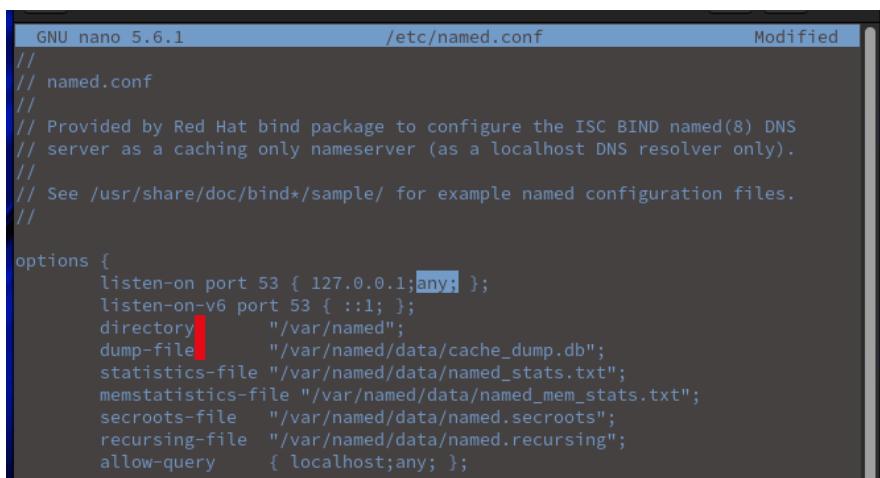
I logged in as the root user to avoid entering `sudo` before every command:

Open the `named.conf` file for editing:

```
[kashsna@goku ~]$ su -
Password:
[root@goku ~]# nano /etc/named.conf
[root@goku ~]#
```

Configure DNS Settings:

- In the `named.conf` file, locate the lines: `listen-on-port 53` and `allow-query`
- Modify this to allow all devices to connect to the DNS server on port 53 by `any` in the brackets.



```
GNU nano 5.6.1          /etc/named.conf          Modified
//                         ...
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
    listen-on port 53 { 127.0.0.1;any; };
    listen-on-v6 port 53 { ::1; };
    directory    "/var/named";
    dump-file    "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    secroots-file   "/var/named/data/named.secroots";
    recursing-file  "/var/named/data/named.recurse";
    allow-query     { localhost;any; };
```

Add DNS Forwarders:

- To add external DNS servers as forwarders, insert the following section into the `options` block in `named.conf`:

```
forwarders {
    159.69.39.4;
    8.8.8.8;
    8.8.4.4;
};
```

These IP addresses represent Google's public DNS servers, ensuring that any DNS query the server cannot resolve locally is forwarded to them.

Configure DNS Zones:

- Scroll down to the **zone “.” IN** section, and after this, define two new zones: one for **forward** and one for **reverse** lookup:
 - The **forward zone** resolves domain names to IP addresses.
 - The **reverse zone** resolves IP addresses to domain names.

I added the following lines:

```
root@goku:/var/named
GNU nano 5.6.1          /etc/named.conf      Modified
zone "." IN {
    type hint;
    file "named.ca";
};

zone "rockyserver.org" IN {
    type master;
    file "fwd.rockyserver.db";
    allow-update { none; };
};

zone "39.69.159.in-addr.arpa" IN {
    type master;
    file "rvs.39.69.159.db";
    allow-update { none; };
};

include "/etc/named.rfc1912.zones";
```

The **forward zone (*rockyserver.org*)** will use the file **fwd.rockyserver.org.db** to store its records.

The **reverse zone (*39.69.159.in-addr.arpa*)** will use **rvs.39.69.159.db** to handle reverse lookups. This zone name represents the reverse of the IP subnet.

Check Configuration for Syntax Errors:

- After editing the **named.conf** file, run the following command to check for any syntax errors:

```
[root@goku ~]# named-checkconf
[root@goku ~]#
```

Since no errors are reported, I can proceed with the next steps.

Create Forward and Reverse Zone Files:

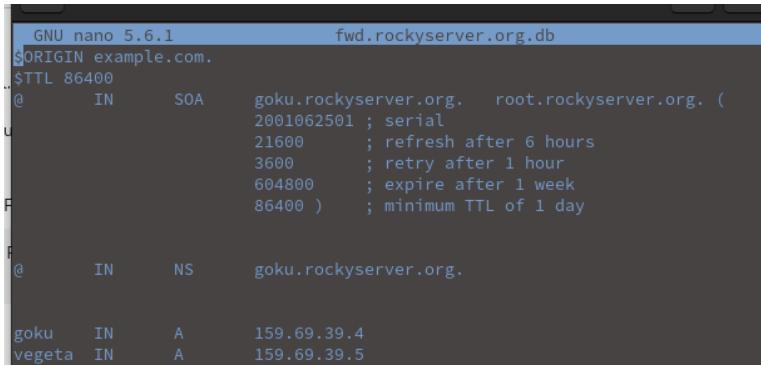
- Change the directory to **/var/named** where zone files are stored:

```
[root@goku ~]#
[root@goku ~]# cd /var/named
[root@goku named]#
```

Use the following command to create the forward zone file: **nano fwd.rockyserver.org.db**

```
[root@goku named]# nano fwd.rockyserver.org.db
```

Copy the template content from the **Red Hat** link and adjust accordingly:



```
GNU nano 5.6.1          fwd.rockyserver.org.db
$ORIGIN example.com.
$TTL 86400
@       IN      SOA    goku.rockyserver.org.  root.rockyserver.org. (
                        2001062501 ; serial
                        21600      ; refresh after 6 hours
                        3600       ; retry after 1 hour
                        604800     ; expire after 1 week
                        86400 )    ; minimum TTL of 1 day

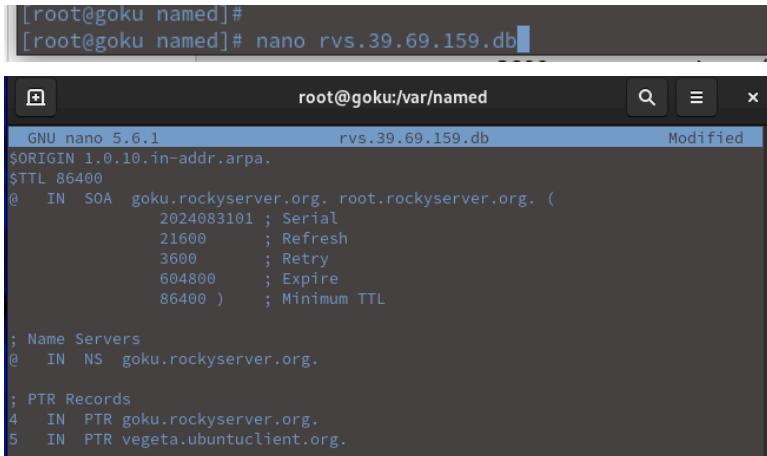
@       IN      NS     goku.rockyserver.org.

goku   IN      A      159.69.39.4
vegeta IN      A      159.69.39.5
```

Save and exit the file.

Create Reverse Zone File:

- Now create the reverse zone file: **nano rvs.39.69.159.db**



```
[root@goku named]#
[root@goku named]# nano rvs.39.69.159.db
root@goku:/var/named
```

```
GNU nano 5.6.1          rvs.39.69.159.db          Modified
$ORIGIN 1.0.10.in-addr.arpa.
$TTL 86400
@       IN      SOA    goku.rockyserver.org.  root.rockyserver.org. (
                        2024083101 ; Serial
                        21600      ; Refresh
                        3600       ; Retry
                        604800     ; Expire
                        86400 )    ; Minimum TTL

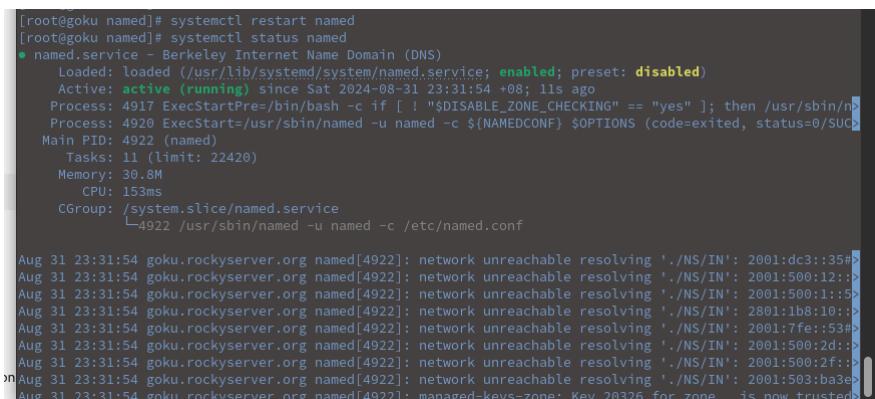
; Name Servers
@       IN      NS     goku.rockyserver.org.

; PTR Records
4       IN      PTR    goku.rockyserver.org.
5       IN      PTR    vegeta.ubuntuclient.org.
```

This file associates the IP addresses **159.69.39.4** and **159.69.39.5** with their respective domain names.

Restart the DNS Service:

- After creating the zone files, restart the DNS service



```
[root@goku named]# systemctl restart named
[root@goku named]# systemctl status named
● named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-08-31 23:31:54 +08; 11s ago
     Process: 4917 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" == "yes" ]; then /usr/sbin/n
    Process: 4920 ExecStart=/usr/sbin/named -u named -c ${NAMEDCONF} ${OPTIONS} (code=exited, status=0/SUC
 Main PID: 4922 (named)
   Tasks: 11 (limit: 22420)
    Memory: 30.8M
      CPU: 153ms
     CGroup: /system.slice/named.service
             └─4922 /usr/sbin/named -u named -c /etc/named.conf

Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:dc3::35>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:500:12::>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:500:1::5>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2801:1b8:10::>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:7fe:53:>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:500:2d::>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:500:2f::>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:503:ba3e:>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: managed-keys-zone: Key 20326 for zone . is now trusted
```

Allow DNS Traffic Through the Firewall:

- Enable DNS (port 53) to pass through both UDP and TCP traffic in the firewall
- I then reload the firewall to save all the changes.

```
[root@goku named]# firewall-cmd --permanent --add-port=53/tcp
success
[root@goku named]# firewall-cmd --permanent --add-port=53/udp
success
[root@goku named]# firewall-cmd --reload
success
[root@goku named]#
```

Set DNS in Resolv.conf:

- Finally, edit the **/etc/resolv.conf** file to point to the Rocky Linux DNS server

```
[root@goku ~]# nano /etc/resolv.conf
```

- Add the following line: **nameserver 159.69.39.4**

```
root@goku:~#
GNU nano 5.6.1          /etc/resolv.conf
# Generated by NetworkManager
search rockyserver.org
nameserver 159.69.39.4
nameserver 8.8.8.8
```

Now that the server has DNS and is operational, we need to change the DNS server to our Rocky Server using the network manager from when we assign static IP addresses.

Change the dns to

- dns=159.69.39.4;8.8.8.8;

```
[kashnsnaggoku ~]$ 
[kashnsnaggoku ~]$ sudo nano /etc/NetworkManager/system-connections/enp0s3.nmconnection
[kashnsnaggoku ~]$
```

```
root@goku:~#
GNU nano 5.6.1          /etc/NetworkManager/system-connections/enp0s3.nmconnection      Modified
[connection]
id=enp0s3
uuid=d41a07f0-2ed4-3990-b9ce-a4d66ec51916
type=ethernet
autoconnect-priority=-999
interface-name=enp0s3
timestamp=1724837378

[ethernet]

[ipv4]
address1=159.69.39.4/24,159.69.39.1
dns=159.69.39.4;8.8.8.8;
ignore-auto-dns=true
method=manual
```

Now restart the DNS server again to save the changes.

```
[root@goku ~]#
[root@goku ~]# sudo systemctl restart named
```

Testing DNS:

- To verify the DNS server is functioning correctly, use **nslookup**..

```
[root@goku ~]# nslookup goku.rockyserver.org
Server:      159.69.39.4
Address:     159.69.39.4#53

Name:   goku.rockyserver.org
Address: 159.69.39.4

[root@goku ~]# nslookup vegeta.rockyserver.org
Server:      159.69.39.4
Address:     159.69.39.4#53

Name:   vegeta.rockyserver.org
Address: 159.69.39.5
```

When we lookup both the goku(server) and vegeta(client), we get the correct IP address

- Another method we can use is to ping the server (goku.rockyserver.org) and the client (vegeta.rockyserver.org).

```
[root@goku ~]# ping goku.rockyserver.org
PING goku.rockyserver.org (159.69.39.4) 56(84) bytes of data.
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=1 ttl=64 time=0.098 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=2 ttl=64 time=0.078 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=3 ttl=64 time=0.191 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=4 ttl=64 time=0.056 ms
^C
--- goku.rockyserver.org ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3085ms
rtt min/avg/max/mdev = 0.056/0.105/0.191/0.051 ms
[root@goku ~]# ping vegeta.rockyserver.org
PING vegeta.rockyserver.org (159.69.39.5) 56(84) bytes of data.
64 bytes from vegeta.ubuntuclient.org (159.69.39.5): icmp_seq=1 ttl=64 time=4.58 ms
64 bytes from vegeta.ubuntuclient.org (159.69.39.5): icmp_seq=2 ttl=64 time=1.98 ms
64 bytes from vegeta.ubuntuclient.org (159.69.39.5): icmp_seq=3 ttl=64 time=2.46 ms
64 bytes from vegeta.ubuntuclient.org (159.69.39.5): icmp_seq=4 ttl=64 time=4.75 ms
^C
--- vegeta.rockyserver.org ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3060ms
rtt min/avg/max/mdev = 1.977/3.441/4.753/1.236 ms
[root@goku ~]#
```

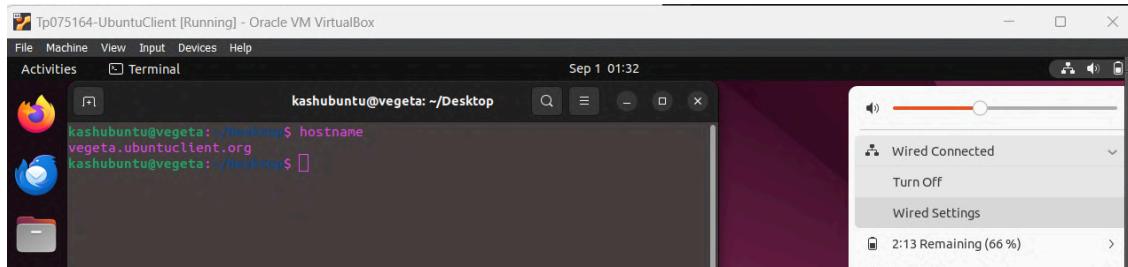
As we can see, both the server and client can be pinged successfully, guaranteeing The DNS server works on Rocky.

DNS configuration in Ubuntu Client

Access Network Settings

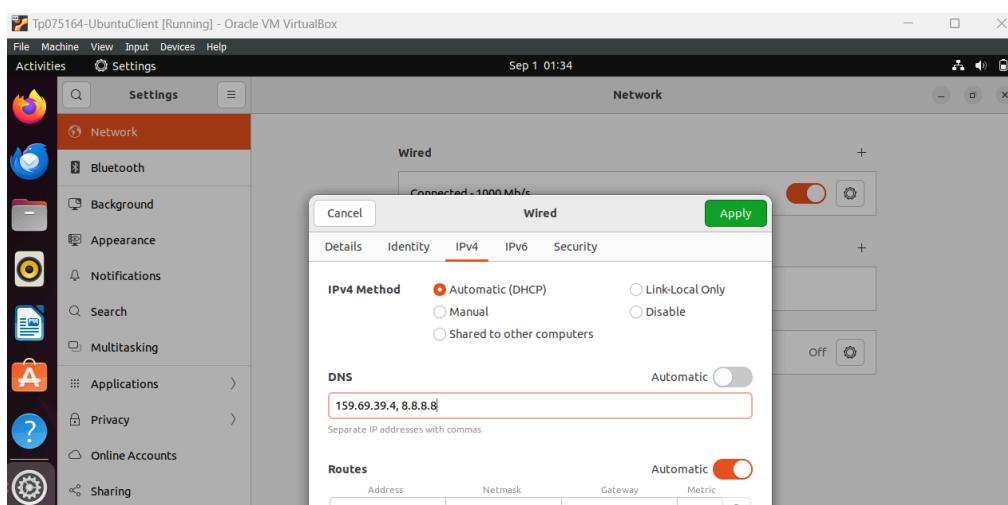
1. Open Network Settings:

- Click the network icon located in the top right corner of your Ubuntu desktop.
- From the dropdown menu, select '**Wired Connected**' to open the wired network settings.
- Click '**Wired Settings**' to access the network configuration interface.



2. Configure DNS Addresses:

- In the network settings menu, click '**Network**' to view the network connections.
- Locate '**Connected-1000MB/s**' under the '**Wired**' section and click the **gear icon** next to it to open its settings.
- In the settings pop-up, navigate to the **IPv4** tab.
- Enter the DNS addresses in the **DNS** field. For example, add the Rocky Server's DNS IP and Google's DNS IP (159.69.39.4 and 8.8.8.8).
- Disable the **Automatic** toggle for DNS to prevent it from being overwritten by DHCP settings.
- Click '**Apply**' to save the changes.



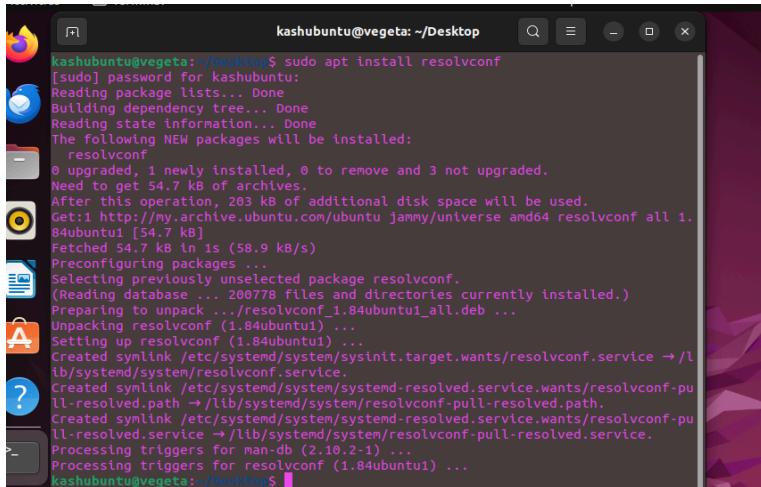
Install `resolvconf`

1. Open a Terminal:

- Press **`Ctrl+Alt+T`** to open a terminal window.

2. Install `resolvconf`:

- Execute the following command to install `resolvconf`, which helps manage DNS resolver configurations: **`sudo apt-get install resolvconf`**



```
kashubuntu@vegeta:~/Desktop$ sudo apt install resolvconf
[sudo] password for kashubuntu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  resolvconf
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 54.7 kB of archives.
After this operation, 203 kB of additional disk space will be used.
Get:1 http://my.archive.ubuntu.com/ubuntu jammy/universe amd64 resolvconf all 1.84ubuntu1 [54.7 kB]
Fetched 54.7 kB in 0s (58.9 kB/s)
Preconfiguring packages ...
Selecting previously unselected package resolvconf.
(Reading database ... 200778 files and directories currently installed.)
Preparing to unpack .../resolvconf_1.84ubuntu1_all.deb ...
Unpacking resolvconf (1.84ubuntu1) ...
Setting up resolvconf (1.84ubuntu1) ...
Created symlink /etc/systemd/system/syntest.target.wants/resolvconf.service → /lib/systemd/system/resolvconf.service.
Created symlink /etc/systemd/system/systemd-resolved.service.wants/resolvconf-pull-resolved.path → /lib/systemd/system/resolvconf-pull-resolved.path.
Created symlink /etc/systemd/system/systemd-resolved.service.wants/resolvconf-pull-resolved.service → /lib/systemd/system/resolvconf-pull-resolved.service.
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for resolvconf (1.84ubuntu1) ...
kashubuntu@vegeta:~/Desktop$
```

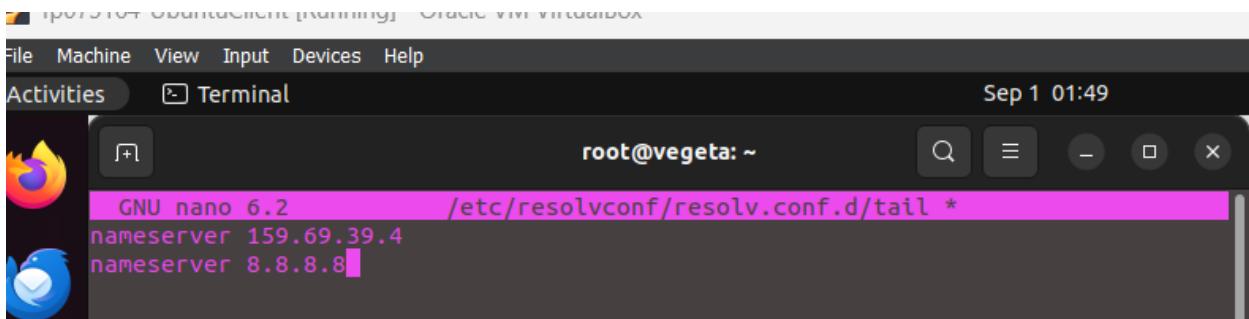
3. Configure `resolvconf`

1. Edit the `resolv.conf` Configuration:

- Open the `resolv.conf` file to add DNS server IP addresses:

```
kashubuntu@vegeta:~/Desktop$ sudo nano /etc/resolvconf/resolv.conf.d/tail
kashubuntu@vegeta:~/Desktop$ sudo resolvconf -u
```

I add the following lines to specify the DNS servers:



```
root@vegeta: ~
GNU nano 6.2          /etc/resolvconf/resolv.conf.d/tail *
nameserver 159.69.39.4
nameserver 8.8.8.8
```

Save the file and exit the editor (**`Ctrl+O, Enter, Ctrl+X`**).

Update DNS Resolver Configuration:

- Apply the changes made to `resolvconf` by running: '`sudo resolvconf -u`'

```
kashubuntu@vegeta:~/Desktop$ sudo resolvconf -u
```

4. Modify NetworkManager.conf

1. Edit NetworkManager Configuration:

- Open the NetworkManager configuration file for editing:

```
root@vegeta:~# nano /etc/NetworkManager/NetworkManager.conf
```

- Locate the `[main]` section and set the DNS option: `dns=none`



Save the file and exit the editor

Restart NetworkManager:

- Apply the configuration changes by restarting NetworkManager:

```
kashubuntu@vegeta:~/Desktop$ sudo systemctl restart NetworkManager
```

Verify DNS Configuration

1. Check `resolv.conf`:

- Confirm that the DNS settings are correctly applied by viewing the `resolv.conf` file:

```
kashubuntu@vegeta:~/Desktop$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
# 127.0.0.53 is the systemd-resolved stub resolver.
# run "systemd-resolve --status" to see details about the actual nameservers.

nameserver 127.0.0.53
options edns0 trust-ad
nameserver 159.69.39.4
nameserver 8.8.8.8
kashubuntu@vegeta:~/Desktop$
```

We can see that it lists the DNS servers I configured.

Test DNS Resolution:

- Use ***nslookup*** to test if the DNS server can resolve hostnames:
- Additionally we can use other methods such as ***ping***, ***dig***, or ***dig -x***.

```
kashubuntu@vegeta:~/Desktop$ ping goku.rockyserver.org
PING goku.rockyserver.org (159.69.39.4) 56(84) bytes of data.
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=1 ttl=64 time=0.833 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=2 ttl=64 time=0.770 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=3 ttl=64 time=0.899 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=4 ttl=64 time=2.05 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=5 ttl=64 time=1.13 ms
^C
--- goku.rockyserver.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4012ms
rtt min/avg/max/mdev = 0.770/1.134/2.049/0.473 ms
kashubuntu@vegeta:~/Desktop$ nslookup vegeta.rockyserver.org
Server:     127.0.0.53
Address:   127.0.0.53#53

Non-authoritative answer:
Name:  vegeta.rockyserver.org
Address: 159.69.39.5

kashubuntu@vegeta:~/Desktop$ nslookup goku.rockyserver.org
Server:     127.0.0.53
Address:   127.0.0.53#53

Non-authoritative answer:
Name:  goku.rockyserver.org
Address: 159.69.39.4
```



```
kashubuntu@vegeta:~/Desktop$ ping vegeta.rockyserver.org
PING vegeta.rockyserver.org (159.69.39.5) 56(84) bytes of data.
64 bytes from vegeta.ubuntuclient.org (159.69.39.5): icmp_seq=1 ttl=64 time=0.016 ms
64 bytes from vegeta.ubuntuclient.org (159.69.39.5): icmp_seq=2 ttl=64 time=0.032 ms
64 bytes from vegeta.ubuntuclient.org (159.69.39.5): icmp_seq=3 ttl=64 time=0.086 ms
64 bytes from vegeta.ubuntuclient.org (159.69.39.5): icmp_seq=4 ttl=64 time=0.045 ms
64 bytes from vegeta.ubuntuclient.org (159.69.39.5): icmp_seq=5 ttl=64 time=0.025 ms
^C
--- vegeta.rockyserver.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4117ms
rtt min/avg/max/mdev = 0.016/0.040/0.086/0.024 ms
kashubuntu@vegeta:~/Desktop$
```



```
kashubuntu@vegeta:~/Desktop$ dig goku.rockyserver.org
; <>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <>> goku.rockyserver.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16601
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;goku.rockyserver.org.      IN      A
;;
;; ANSWER SECTION:
goku.rockyserver.org.  6304    IN      A      159.69.39.4
;;
;; Query time: 2 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sun Sep 01 04:52:54 +08 2024
;; MSG SIZE  rcvd: 65
```



```
kashubuntu@vegeta:~/Desktop$ dig vegeta.rockyserver.org
; <>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <>> vegeta.rockyserver.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9336
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;vegeta.rockyserver.org.      IN      A
;;
;; ANSWER SECTION:
vegeta.rockyserver.org.  6099    IN      A      159.69.39.5
;;
;; Query time: 6 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sun Sep 01 04:53:46 +08 2024
;; MSG SIZE  rcvd: 67
```



```
kashubuntu@vegeta:~/Desktop$ dig -x goku.rockyserver.org
; <>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <>> -x goku.rockyserver.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 19438
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;org.rockyserver.goku.in-addr.arpa. IN  PTR
;;
;; AUTHORITY SECTION:
in-addr.arpa.      3600    IN      SOA     b.in-addr-servers.arpa. nstld.iana.org
. 2022094929 1800 900 604800 3600
;;
;; Query time: 1577 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sun Sep 01 04:54:24 +08 2024
;; MSG SIZE  rcvd: 130
```



```
kashubuntu@vegeta:~/Desktop$ dig -x vegeta.rockyserver.org
; <>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <>> -x vegeta.rockyserver.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 35576
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;org.rockyserver.vegeta.in-addr.arpa. IN      PTR
;;
;; AUTHORITY SECTION:
in-addr.arpa.      3417    IN      SOA     b.in-addr-servers.arpa. nstld.iana.org
. 2022094929 1800 900 604800 3600
;;
;; Query time: 22 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sun Sep 01 04:54:57 +08 2024
;; MSG SIZE  rcvd: 132
```

TROUBLESHOOTING:

Error 1: Job for named.service failed

I encountered the following error when trying to restart the DNS service. This means there was an error when I was creating the zones and their respective files.

```
[kashsna@goku ~]$ systemctl restart named  
Job for named.service failed because the control process exited with error code.  
See "systemctl status named.service" and "journalctl -xeu named.service" for det-  
ails.  
[kashsna@goku ~]$
```

I used 'systemctl status named.service' to find more details about why the service failed. As seen it shows that the zone 39.69.159.in-addr.arpa/IN: not loaded due to errors

I then use checkzone to find out the exact issue. As below, the issue is that there are 0 SOA records, there are no NS records.

```
[root@goku named]# ls -l /var/named/rvs.39.69.159.db
-rw-r--r--. 1 named named 494 Aug 31 22:47 /var/named/rvs.39.69.159.db
[root@goku named]# named-checkzone 39.69.159.in-addr.arpa /var/named/rvs.39.69.159.db
/var/named/rvs.39.69.159.db:3: ignoring out-of-zone data (1.0.10.in-addr.arpa)
/var/named/rvs.39.69.159.db:10: ignoring out-of-zone data (1.0.10.in-addr.arpa)
/var/named/rvs.39.69.159.db:12: ignoring out-of-zone data (4.1.0.10.in-addr.arpa)
/var/named/rvs.39.69.159.db:13: ignoring out-of-zone data (5.1.0.10.in-addr.arpa)
zone 39.69.159.in-addr.arpa/IN: has 0 SOA records
zone 39.69.159.in-addr.arpa/IN: has no NS records
zone 39.69.159.in-addr.arpa/IN: not loaded due to errors.
[root@goku named]#
```

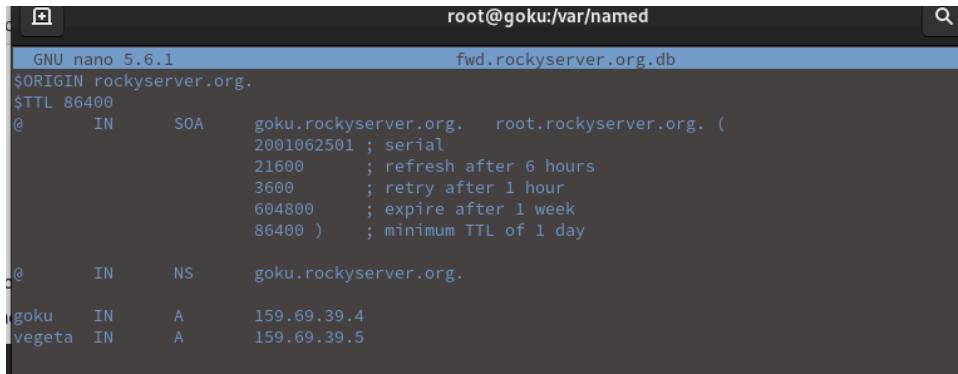
Enter the 'rvs.39.69.159.db' file to check the errors. After the \$ORIGIN I did not change the path to match my zone. So I changed it from '1.0.10' to '39.69.159' to match the reverse of the IP.

```
[+] root@goku:/var/named
GNU nano 5.6.1
$ORIGIN 39.69.159.in-addr.arpa.
$TTL 86400
@ IN SOA goku.rockyserver.org. root.rockyserver.org. (
    2024083101 ; Serial
    21600      ; Refresh
    3600       ; Retry
    604800     ; Expire
    86400 )    ; Minimum TTL

; Name Servers
@ IN NS goku.rockyserver.org.

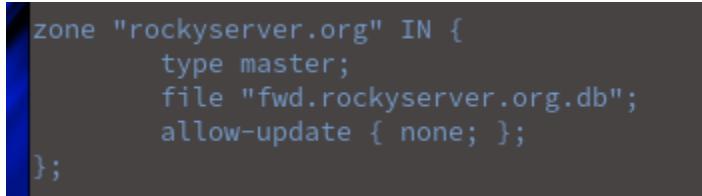
; PTR Records
4 IN PTR goku.rockyserver.org.
5 IN PTR ubuntuclient.vegeta.ubuntuclient.org.
```

In the fwd file, the \$ORIGIN domain was not changed to match the domain in the zone either which caused another error. So change it to ‘rockyserver.org.’ to match the domain.



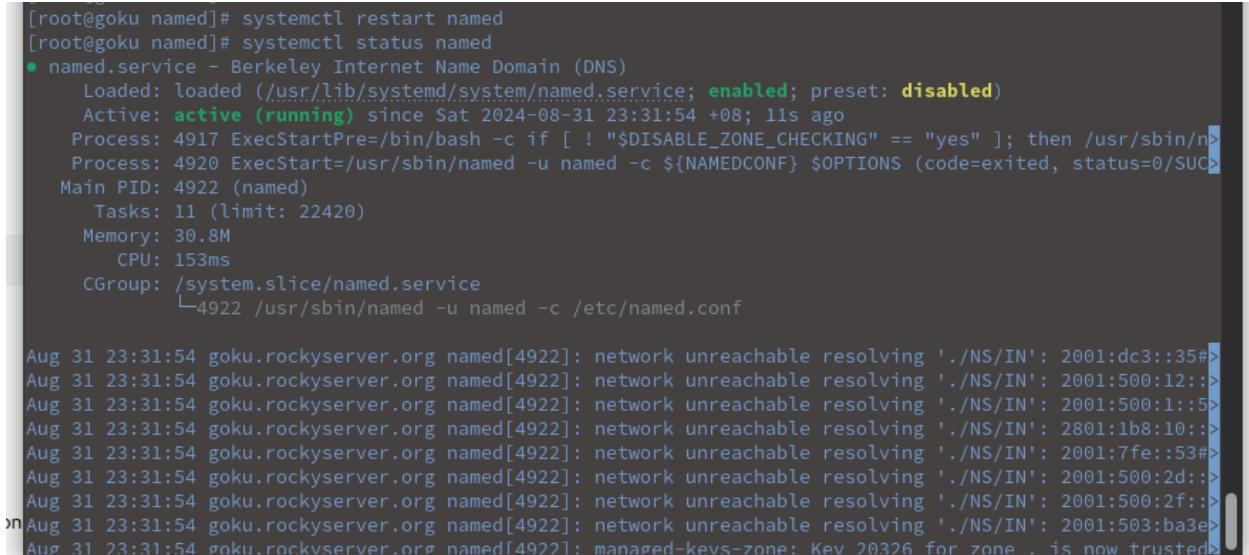
```
root@goku:/var/named
GNU nano 5.6.1
fwd.rockyserver.org.db
$ORIGIN rockyserver.org.
$TTL 86400
@ IN SOA goku.rockyserver.org. root.rockyserver.org. (
    2001062501 ; serial
    21600      ; refresh after 6 hours
    3600       ; retry after 1 hour
    604800     ; expire after 1 week
    86400      ; minimum TTL of 1 day
)
@ IN NS goku.rockyserver.org.
goku IN A 159.69.39.4
vegeta IN A 159.69.39.5
```

Another issue was that in the ‘/etc/named.conf’ file, I initially left it as “fwd.rockyserver.db” instead of “fwd.rockyserver.org.db”. This caused it to search for a non-existing file instead of the one we configured for the forward zone.



```
zone "rockyserver.org" IN {
    type master;
    file "fwd.rockyserver.org.db";
    allow-update { none; };
};
```

Now when we run ‘systemctl restart named’ there is no error that occurred. Next run ‘systemctl status named’.



```
[root@goku named]# systemctl restart named
[root@goku named]# systemctl status named
● named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-08-31 23:31:54 +08; 11s ago
     Process: 4917 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" == "yes" ]; then /usr/sbin/named -c ${NAMEDCONF} $OPTIONS (code=exited, status=0/SUCCESS)
     Process: 4920 ExecStart=/usr/sbin/named -u named -c ${NAMEDCONF} $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 4922 (named)
     Tasks: 11 (limit: 22420)
    Memory: 30.8M
       CPU: 153ms
      CGroup: /system.slice/named.service
              └─4922 /usr/sbin/named -u named -c /etc/named.conf

Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:dc3::35#>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:500:12::#>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:500:1::5#>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2801:1b8:10::#>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:7fe::53#>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:500:2d::#>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:500:2f::#>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: network unreachable resolving './NS/IN': 2001:503:ba3e#>
Aug 31 23:31:54 goku.rockyserver.org named[4922]: managed-kevs-zone: Key 20326 for zone . is now trusted#>
```

Error 2: server can't find `vegeta.rockyserver.org`: NXDOMAIN

One method to test if the DNS Rocky Server is working is to use ‘nslookup’

However I came across an error where the server can't find the client or server: NXDOMAIN

```
kashubuntu@vegeta:~/Desktop$ nslookup vegeta.rockyserver.org
;; communications error to 127.0.0.53#53: timed out
Server:      127.0.0.53
Address:     127.0.0.53#53

** server can't find vegeta.rockyserver.org: NXDOMAIN
```

Initially I assumed the issue lied in the way I set up the DNS Rocky Server in the Ubuntu Client. So I decided to verify the DNS settings using the nmcli tool.

- Using the command: `nmcli dev show | grep DNS`

I could verify that the correct DNS servers (which I entered previously) are listed.

```
kashubuntu@vegeta:~/Desktop$ nmcli dev show | grep DNS
[IP4.DNS[1]]:                         159.69.39.4
[IP4.DNS[2]]:                         8.8.8.8
kashubuntu@vegeta:~/Desktop$
```

Then I noticed in the resolv.conf file, an additional line that is ‘options edns0 trust-ad’. I used Sudo nano /etc/resolv.conf to edit the file and remove that line.

```
kashubuntu@vegeta:~/Desktop$ sudo nano /etc/resolv.conf
[REDACTED]
GNU nano 6.2                               kashubuntu@vegeta: ~/Desktop
/etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
# 127.0.0.53 is the systemd-resolved stub resolver.
# run "systemd-resolve --status" to see details about the actual nameservers.

nameserver 127.0.0.53
options edns0 trust-ad
nameserver 159.69.39.4
nameserver 8.8.8.8

kashubuntu@vegeta:~/Desktop$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
# 127.0.0.53 is the systemd-resolved stub resolver.
# run "systemd-resolve --status" to see details about the actual nameservers.

nameserver 127.0.0.53
nameserver 159.69.39.4
nameserver 8.8.8.8
```

However when it continued to still give errors, I assumed there was an error with the firewall in Rocky. I changed the permission to allow DNS service through using

- `Sudo firewall-cmd –add-service=dns –permanent` (Allow DNS through the firewall)
- `Sudo firewall-cmd –reload` (Reload the firewall to save changes)

```
[root@goku ~]# sudo firewall-cmd --add-service=dns --permanent
success
[root@goku ~]# sudo firewall-cmd --reload
success
[root@goku ~]#
```

I then listed all the allowed services in the firewall to ensure that dns has permission.

```
[root@goku ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
    services: cockpit dhcpcv6-client dns ssh
    ports: 53/tcp 53/udp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

And ran ***journalctl -xeu named.service*** to ensure there were no errors in the DNS files.

```
[root@goku ~]#
[root@goku ~]# journalctl -xeu named.service
Sep 01 04:22:36 goku.rockyserver.org named[3216]: network unreachable resolving>
Sep 01 04:22:36 goku.rockyserver.org named[3216]: network unreachable resolving>
Sep 01 04:22:36 goku.rockyserver.org named[3216]: zone localhost/IN: loaded ser>
Sep 01 04:22:36 goku.rockyserver.org named[3216]: all zones loaded
Sep 01 04:22:36 goku.rockyserver.org named[3216]: running
Sep 01 04:22:36 goku.rockyserver.org systemd[1]: Started Berkeley Internet Name>
  Subject: A start job for unit named.service has finished successfully
  Defined-By: systemd
  Support: https://wiki.rockylinux.org/rocky/support

  A start job for unit named.service has finished successfully.

  The job identifier is 2813.
Sep 01 04:22:36 goku.rockyserver.org named[3216]: managed-keys-zone: Key 20326 >
Sep 01 04:22:37 goku.rockyserver.org named[3216]: resolver priming query comple>
Sep 01 04:25:05 goku.rockyserver.org named[3216]: timed out resolving 'ab.chatg>
Sep 01 04:25:05 goku.rockyserver.org named[3216]: timed out resolving 'ab.chatg>
Sep 01 04:25:05 goku.rockyserver.org named[3216]: timed out resolving 'ab.chatg>
Sep 01 04:25:05 goku.rockyserver.org named[3216]: timed out resolving 'ab.chatg>
Sep 01 04:25:05 goku.rockyserver.org named[3216]: validating ab.chatgpt.com/AAA>
Sep 01 04:25:05 goku.rockyserver.org named[3216]: validating ab.chatgpt.com/A: >
Sep 01 04:25:09 goku.rockyserver.org named[3216]: validating chatgpt.com/A: no >
Sep 01 04:25:09 goku.rockyserver.org named[3216]: validating chatgpt.com/AAAA: >
```

Ubuntu is now able to successfully look up the client and server through DNS.

```
kashubuntu@vegeta:~/Desktop$ nslookup vegeta.rockyserver.org
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:  vegeta.rockyserver.org
Address: 159.69.39.5

kashubuntu@vegeta:~/Desktop$ nslookup goku.rockyserver.org
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:  goku.rockyserver.org
Address: 159.69.39.4
```

6.0 DHCP Configuration on Rocky Linux

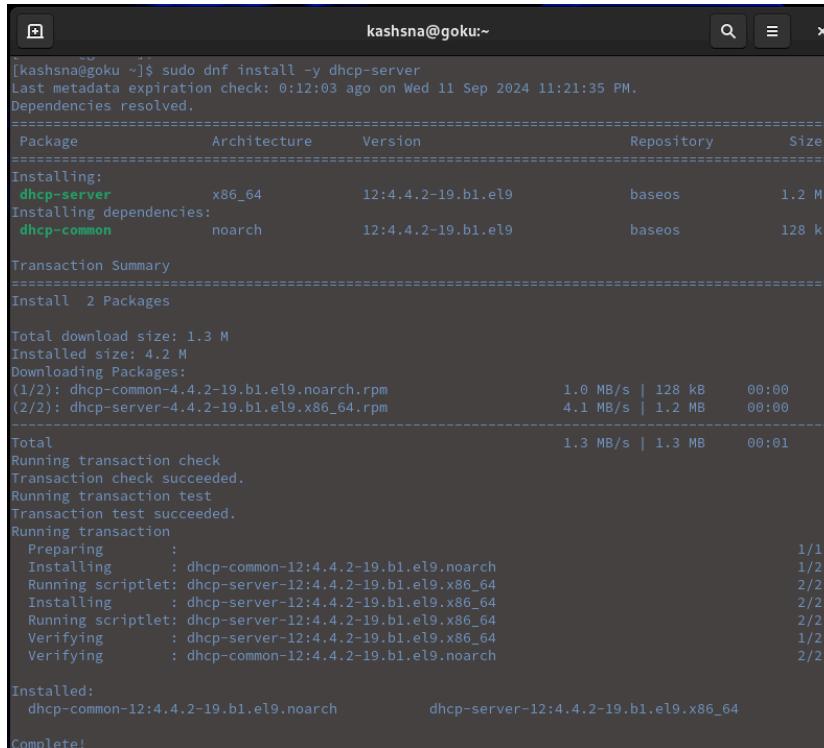
1. Install the DHCP Server

1. Open a Terminal on Rocky Linux:

- Access the terminal on your Rocky Linux server.

2. Install DHCP Server Package:

- Run the following command to install the DHCP server: ***sudo dnf install dhcp-server***



```
[kashsna@goku ~]$ sudo dnf install -y dhcp-server
Last metadata expiration check: 0:12:03 ago on Wed 11 Sep 2024 11:21:35 PM.
Dependencies resolved.
=====
Transaction Summary
=====
Install  2 Packages

Total download size: 1.3 M
Installed size: 4.2 M
Downloading Packages:
(1/2): dhcp-common-4.4.2-19.b1.el9.noarch.rpm           1.0 MB/s | 128 kB   00:00
(2/2): dhcp-server-4.4.2-19.b1.el9.x86_64.rpm          4.1 MB/s | 1.2 MB   00:00
Total                                         1.3 MB/s | 1.3 MB   00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :                                                 1/1
  Installing : dhcp-common-12:4.4.2-19.b1.el9.noarch      1/2
  Running scriptlet: dhcp-server-12:4.4.2-19.b1.el9.x86_64 2/2
  Installing : dhcp-server-12:4.4.2-19.b1.el9.x86_64      2/2
  Running scriptlet: dhcp-server-12:4.4.2-19.b1.el9.x86_64 2/2
  Verifying   : dhcp-server-12:4.4.2-19.b1.el9.x86_64      1/2
  Verifying   : dhcp-common-12:4.4.2-19.b1.el9.noarch      2/2

Installed:
  dhcp-common-12:4.4.2-19.b1.el9.noarch      dhcp-server-12:4.4.2-19.b1.el9.x86_64

Complete!
```

2. Create a DHCP Configuration File

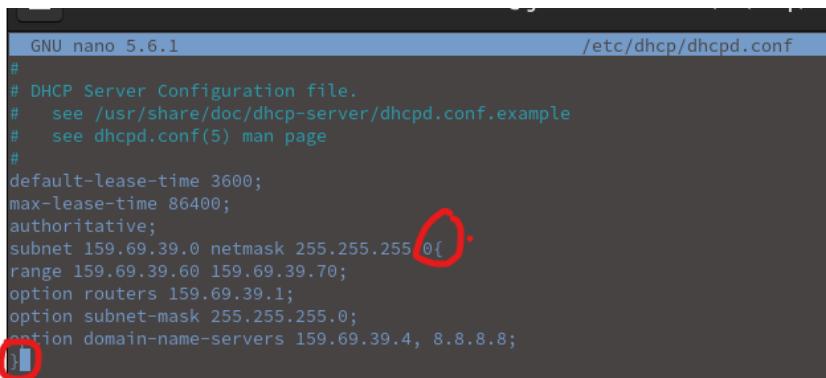
1. Create and Edit the DHCP Configuration File:

- Open the DHCPd configuration file for editing:

```
[kashsna@goku ~]$ sudo nano /etc/dhcp/dhcpd.conf
```

2. Setup DHCP Configuration:

- I added the following configuration to the file, adjusting the IP address range as below:
 - default -lease-time 3600;
 - Max-lease-time 86400;
 - authoritative;
 - Subnet 159.69.39.0 netmask 255.255.255.0 {
 - Range 159.69.39.60 159.69.39.70; (Can be any range)
 - Option routers 159.69.39.1;
 - Option subnet-mask 255.255.255.0;
 - Option domain-name-servers 159.69.39.4, 8.8.8.8;
 - }

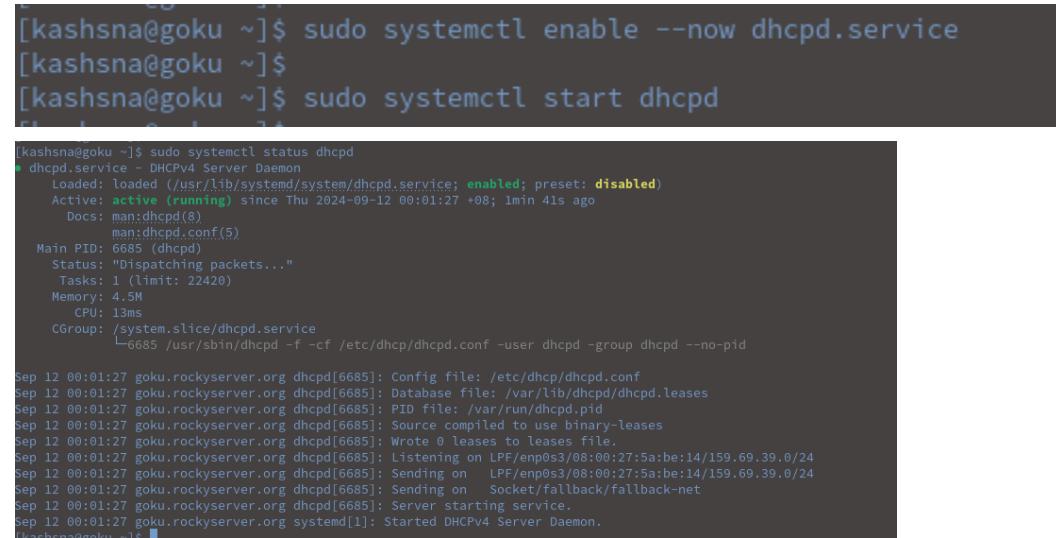


```
GNU nano 5.6.1                               /etc/dhcp/dhcpd.conf
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp-server/dhcpd.conf.example
#   see dhcpcd.conf(5) man page
#
default-lease-time 3600;
max-lease-time 86400;
authoritative;
subnet 159.69.39.0 netmask 255.255.255.0 {
    range 159.69.39.60 159.69.39.70;
    option routers 159.69.39.1;
    option subnet-mask 255.255.255.0;
    option domain-name-servers 159.69.39.4, 8.8.8.8;
}
```

Save the file and exit the editor

3. Enable and Start the DHCP Service

Enable the DHCP Service to start on Boot with the following command and start the service:



```
[kashsna@goku ~]$ sudo systemctl enable --now dhcpcd.service
[kashsna@goku ~]$
[kashsna@goku ~]$ sudo systemctl start dhcpcd
[kashsna@goku ~]$

[kashsna@goku ~]$ sudo systemctl status dhcpcd
● dhcpcd.service - DHCPv4 Server Daemon
   Loaded: loaded (/usr/lib/systemd/system/dhcpcd.service; enabled; preset: disabled)
     Active: active (running) since Thu 2024-09-12 00:01:27 +08; 1min 41s ago
       Docs: man:dhcpcd(8)
              man:dhcpcd.conf(5)
     Main PID: 6685 (dhcpcd)
        Status: "Dispatching packets..."
          Tasks: 1 (limit: 22420)
         Memory: 4.5M
            CPU: 13ms
           CGroup: /system.slice/dhcpcd.service
                   └─6685 /usr/sbin/dhcpcd -f -cf /etc/dhcp/dhcpcd.conf -user dhcpcd -group dhcpcd --no-pid

Sep 12 00:01:27 goku.rockyserver.org dhcpcd[6685]: Config file: /etc/dhcp/dhcpcd.conf
Sep 12 00:01:27 goku.rockyserver.org dhcpcd[6685]: Database file: /var/lib/dhcpcd/dhcpcd.leases
Sep 12 00:01:27 goku.rockyserver.org dhcpcd[6685]: PID file: /var/run/dhcpcd.pid
Sep 12 00:01:27 goku.rockyserver.org dhcpcd[6685]: Source compiled to use binary-leases
Sep 12 00:01:27 goku.rockyserver.org dhcpcd[6685]: Wrote 0 leases to leases file.
Sep 12 00:01:27 goku.rockyserver.org dhcpcd[6685]: Listening on LPF/epn0s3/08:00:27:5a:be:14/159.69.39.0/24
Sep 12 00:01:27 goku.rockyserver.org dhcpcd[6685]: Sending on  LPF/epn0s3/08:00:27:5a:be:14/159.69.39.0/24
Sep 12 00:01:27 goku.rockyserver.org dhcpcd[6685]: Sending on  Socket/fallback/fallback-net
Sep 12 00:01:27 goku.rockyserver.org dhcpcd[6685]: Server starting service.
Sep 12 00:01:27 goku.rockyserver.org systemd[1]: Started DHCPv4 Server Daemon.
```

Check the status of DHCP service and we can see that it is running.

4. Configure the Firewall

1. Allow DHCP Traffic Through the Firewall

This command allows DHCP client requests through the firewall.

```
kashsna@goku ~]$ sudo firewall-cmd --permanent --add-service=dhcp  
success  
kashsna@goku ~]$ sudo firewall-cmd --reload  
success
```

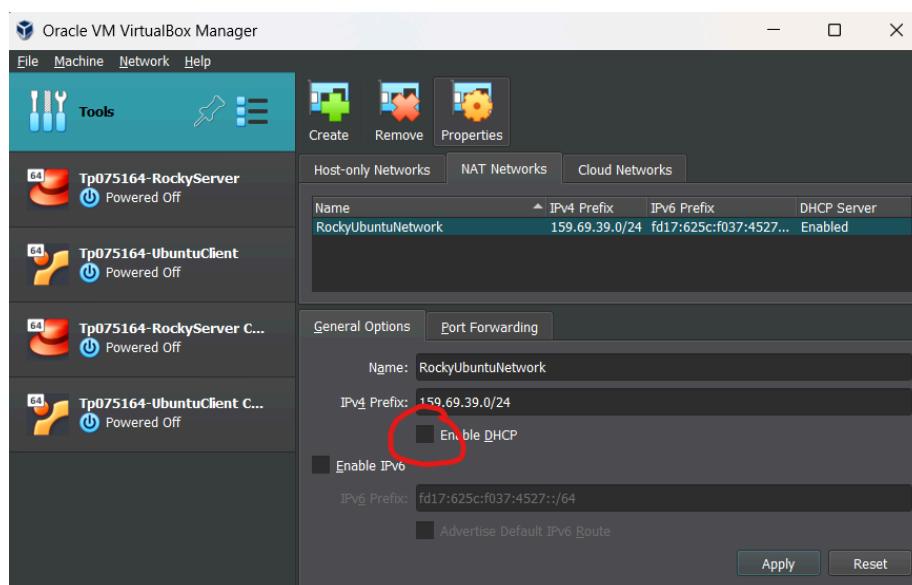
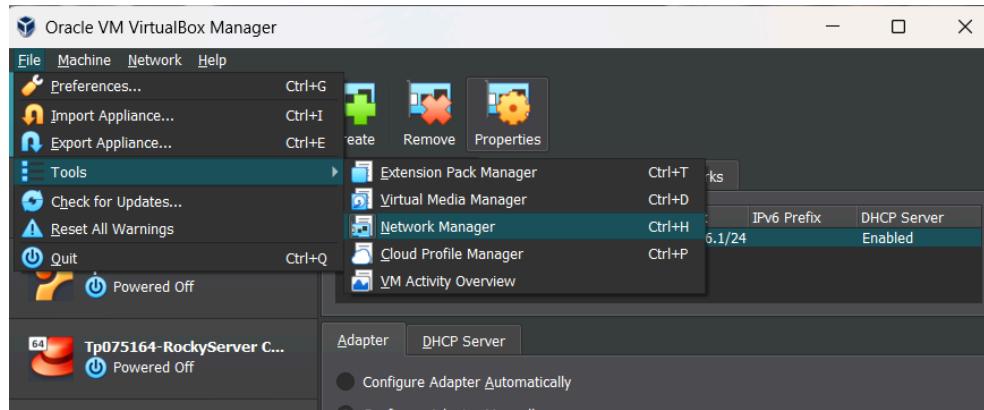
5. Update VirtualBox Network Settings

1. Shutdown Rocky Server:

- Power off the Rocky Server.

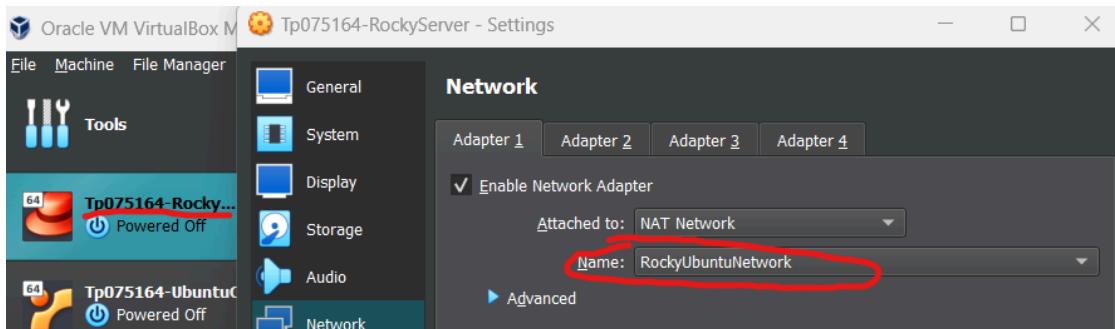
2. Modify VirtualBox Network Settings:

- Open VirtualBox Manager.
- Navigate to **File -> Tools -> Network Manager**.

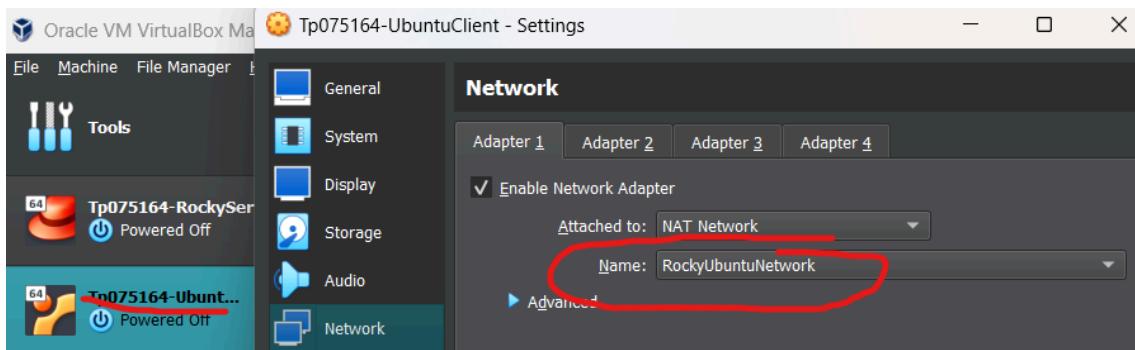


- Locate the NAT network configuration and uncheck **Enable DHCP** for that network.

Ensure that the Rocky Server is still connected to the same NAT Network.



Similarly, check that the Ubuntu client is connected to the same NAT Network.



Verify Network Adapter and Connection ID

1. Use `nmcli` to Verify Network Adapter:

- Open a terminal on Rocky and run. We can see the IP address is: **159.69.39.4** and the connection name is **`enp0s3`**

```
[kashsna@goku ~]$ nmcli
enp0s3: connected to enp0s3
    "Intel 82540EM"
    ethernet (e1000), 08:00:27:5A:BE:14, hw, mtu 1500
    ip4 default
    inet4 159.69.39.4/24
    route4 159.69.39.0/24 metric 100
    route4 default via 159.69.39.1 metric 100
    inet6 fe80::a00:27ff:fe5a:be14/64
    route6 fe80::/64 metric 1024

lo: connected (externally) to lo
    "lo"
    loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
    inet4 127.0.0.1/8
    inet6 ::1/128
    route6 ::1/128 metric 256

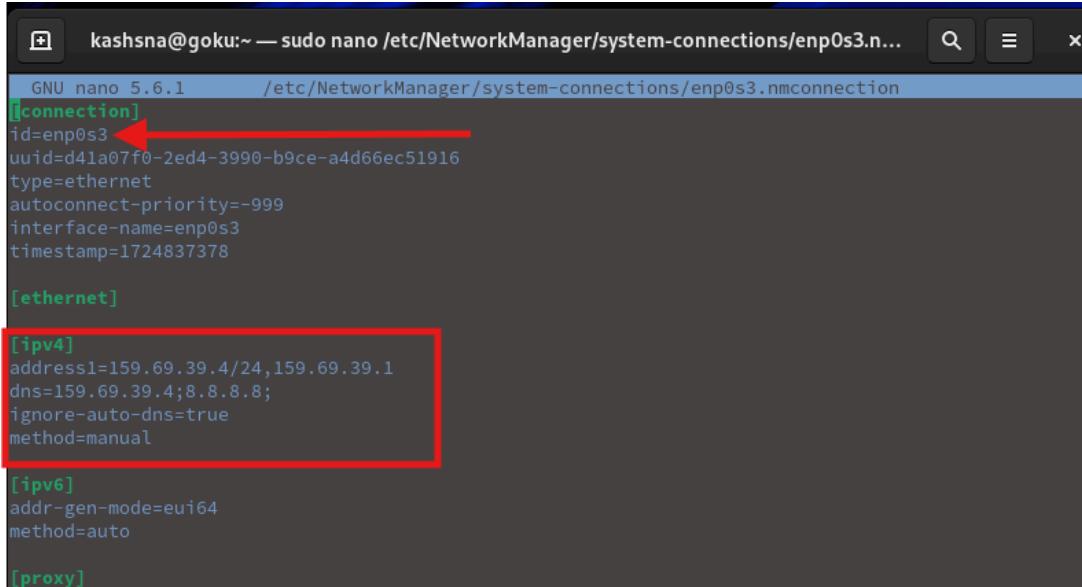
DNS configuration:
    servers: 159.69.39.4 8.8.8.8
    interface: enp0s3
```

Check Network Connection Settings:

- Open the network connection settings file:

```
[kashsna@goku ~]$ sudo nano /etc/NetworkManager/system-connections/enp0s3.nmconnection
```

Enter the file and verify that the **ID** matches the connection name from the **nmcli** command, and that the **IPv4** section is correctly configured for DHCP.



```
GNU nano 5.6.1      /etc/NetworkManager/system-connections/enp0s3.nmconnection
[connection]
id=enp0s3
uuid=d41a07f0-2ed4-3990-b9ce-a4d66ec51916
type=ethernet
autoconnect-priority=-999
interface-name=enp0s3
timestamp=1724837378

[ethernet]

[ipv4]
address1=159.69.39.4/24,159.69.39.1
dns=159.69.39.4;8.8.8.8;
ignore-auto-dns=true
method=manual

[ipv6]
addr-gen-mode=eui64
method=auto

[proxy]
```

Then I restart the NetworkManager and check the Status to see it is active and running.

```
[kashsna@goku ~]$ sudo systemctl restart NetworkManager
[kashsna@goku ~]$
[kashsna@goku ~]$ sudo systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-09-12 00:31:30 +08; 11s ago
     Docs: man:NetworkManager(8)
     Main PID: 3076 (NetworkManager)
        Tasks: 4 (limit: 22420)
       Memory: 5.4M
          CPU: 99ms
        CGroup: /system.slice/NetworkManager.service
                   └─3076 /usr/sbin/NetworkManager --no-daemon

Sep 12 00:31:30 goku.rockyserver.org NetworkManager[3076]: <info> [1726072290.2298] device (lo):>
Sep 12 00:31:30 goku.rockyserver.org NetworkManager[3076]: <info> [1726072290.2362] device (lo):>
Sep 12 00:31:30 goku.rockyserver.org NetworkManager[3076]: <info> [1726072290.2395] device (enp0s3):>
Sep 12 00:31:30 goku.rockyserver.org NetworkManager[3076]: <info> [1726072290.2479] device (enp0s3):>
Sep 12 00:31:30 goku.rockyserver.org NetworkManager[3076]: <info> [1726072290.2497] device (enp0s3):>
Sep 12 00:31:30 goku.rockyserver.org NetworkManager[3076]: <info> [1726072290.2527] manager: NetManager:>
Sep 12 00:31:30 goku.rockyserver.org NetworkManager[3076]: <info> [1726072290.2567] device (enp0s3):>
Sep 12 00:31:30 goku.rockyserver.org NetworkManager[3076]: <info> [1726072290.2603] manager: NetManager:>
Sep 12 00:31:30 goku.rockyserver.org NetworkManager[3076]: <info> [1726072290.2648] manager: static:>
Sep 12 00:31:30 goku.rockyserver.org NetworkManager[3076]: <info> [1726072290.8087] agent-managed:>

[kashsna@goku ~]$ sudo systemctl enable --now dhcpd.service
[sudo] password for kashsna:
[kashsna@goku ~]$
```

Enable the DHCP service to start on boot.

Now we open Ubuntu to check if the DHCP has assigned the IP address. Run ***ifconfig*** and we can see that the Ubuntu IP address has changed to **159.69.39.60** which is in the range we specified previously.

```
kashubuntu@vegeta: ~/Desktop
kashubuntu@vegeta: ~/Desktop
enp0s3: connected to Wired connection 1
    "Intel 82540EM"
    ethernet (e1000), 08:00:27:71:7D:26, hw, mtu 1500
    ip4 default
    inet4 159.69.39.60/24
        route4 159.69.39.0/24 metric 100
        route4 169.254.0.0/16 metric 1000
        route4 default via 159.69.39.1 metric 100
    inet6 fe80::f478:f52c:8022:d64a/64
        route6 fe80::/64 metric 1024

lo: unmanaged
    "lo"
    loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536

DNS configuration:
    servers: 159.69.39.4 8.8.8.8
    interface: enp0s3
```

We ping the Rocky Server from Ubuntu with the new IP address. As seen below, the ping works, therefore our DHCP is successful.

```
kashubuntu@vegeta: ~/Desktop$ 
kashubuntu@vegeta: ~/Desktop$ ping goku.rockyserver.org
PING goku.rockyserver.org (159.69.39.4) 56(84) bytes of data.
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=1 ttl=64 time=1.61 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=2 ttl=64 time=2.41 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=3 ttl=64 time=2.75 ms
64 bytes from goku.rockyserver.org (159.69.39.4): icmp_seq=4 ttl=64 time=2.29 ms
^C
--- goku.rockyserver.org ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3016ms
rtt min/avg/max/mdev = 1.612/2.266/2.753/0.414 ms
kashubuntu@vegeta: ~/Desktop$
```

Troubleshooting:

Error 1:

There is an error that occurred when I enable the DHCPd service.

```
[kashsna@goku ~]$ sudo systemctl enable --now dhcpd.service
[sudo] password for kashsna:
Created symlink /etc/systemd/system/multi-user.target.wants/dhcpd.service → /usr/lib/systemd/system/dhcpd.service.
Job for dhcpd.service failed because the control process exited with error code.
See "systemctl status dhcpd.service" and "journalctl -xeu dhcpd.service" for details.
[kashsna@goku ~]$
```

Run journalctl -xeu dhcpd.service to get a more detailed version of the error

```
[kashsna@goku ~]$ journalctl -xeu dhcpd.service
Sep 11 23:55:15 goku.rockyserver.org dhcpd[6531]: For info, please visit https://www.isc.org/software/dhcp/
Sep 11 23:55:15 goku.rockyserver.org dhcpd[6531]: /etc/dhcp/dhcpd.conf line 91 expecting left brace.
Sep 11 23:55:15 goku.rockyserver.org dhcpd[6531]: subnet 159.69.39.0 netmask 255.255.255.0;
Sep 11 23:55:15 goku.rockyserver.org dhcpd[6531]: ^
Sep 11 23:55:15 goku.rockyserver.org dhcpd[6531]: Configuration file errors encountered -- exiting
Sep 11 23:55:15 goku.rockyserver.org dhcpd[6531]: This version of ISC DHCP is based on the release available
Sep 11 23:55:15 goku.rockyserver.org dhcpd[6531]: on ftp.isc.org. Features have been added and other changes
Sep 11 23:55:15 goku.rockyserver.org dhcpd[6531]: have been made to the base software release in order to make
Sep 11 23:55:15 goku.rockyserver.org dhcpd[6531]: it work better with this distribution.
Sep 11 23:55:15 goku.rockyserver.org dhcpd[6531]: Please report issues with this software via:
Sep 11 23:55:15 goku.rockyserver.org dhcpd[6531]: https://bugs.rockylinux.org/
Sep 11 23:55:15 goku.rockyserver.org dhcpd[6531]: exiting...
Sep 11 23:55:15 goku.rockyserver.org systemd[1]: dhcpd.service: Main process exited, code=exited, status=1/FAILURE
Subject: Unit process exited
Defined-By: systemd
Support: https://wiki.rockylinux.org/rocky/support

An ExecStart= process belonging to unit dhcpd.service has exited.

The process' exit code is 'exited' and its exit status is 1.
Sep 11 23:55:15 goku.rockyserver.org systemd[1]: dhcpd.service: Failed with result 'exit-code'.
Subject: Unit failed
Defined-By: systemd
Support: https://wiki.rockylinux.org/rocky/support

The unit dhcpd.service has entered the 'failed' state with result 'exit-code'.
Sep 11 23:55:15 goku.rockyserver.org systemd[1]: Failed to start DHCPv4 Server Daemon.
Subject: A start job for unit dhcpd.service has failed
Defined-By: systemd
Support: https://wiki.rockylinux.org/rocky/support

A start job for unit dhcpd.service has finished with a failure.
```

We can see that the error is due to a missing left brace in the dhcpd configuration file.



The image shows two side-by-side terminal windows. The left window displays the original `/etc/dhcp/dhcpd.conf` file with syntax errors. The right window shows the corrected file where the missing left brace is highlighted with a red circle.

```
GNU nano 5.6.1
/etc/dhcp/dhcpd.conf
#
# DHCP Server Configuration file.
# see /usr/share/doc/dhcp-server/dhcpd.conf.example
# see dhcpd.conf(5) man page
#
default-lease-time 3600;
max-lease-time 86400;
authoritative;
subnet 159.69.39.0 netmask 255.255.255.0;
range 159.69.39.60 159.69.39.70;
option routers 159.69.39.1;
option subnet-mask 255.255.255.0;
option domain-name-servers 159.69.39.4, 8.8.8.8;
```

```
GNU nano 5.6.1
/etc/dhcp/dhcpd.conf
#
# DHCP Server Configuration file.
# see /usr/share/doc/dhcp-server/dhcpd.conf.example
# see dhcpd.conf(5) man page
#
default-lease-time 3600;
max-lease-time 86400;
authoritative;
subnet 159.69.39.0 netmask 255.255.255.0{;
range 159.69.39.60 159.69.39.70;
option routers 159.69.39.1;
option subnet-mask 255.255.255.0;
option domain-name-servers 159.69.39.4, 8.8.8.8;
```

The left image is the original configuration I made. As we can see I had mistakenly left out the braces that are suppose to surround the range and options lines.

```
[kashsna@goku ~]$ sudo systemctl enable --now dhcpd.service
[sudo] password for kashsna:
[kashsna@goku ~]$
```

Now the enable of the dhcp service is working.

7.0 Email Server:

Postfix Installation and Configuration:

Postfix Installation:

- To begin setting up the email server, I installed Postfix on Rocky Linux by running the command: `sudo dnf install postfix`

```
[kashsna@goku ~]$ sudo dnf install postfix
Last metadata expiration check: 0:00:27 ago on Thu 12 Sep 2024 11:54:42 PM.
Dependencies resolved.
=====
Package           Architecture Version       Repository      Size
=====
Installing:
 postfix          x86_64      2:3.5.9-24.el9    appstream     1.4 M
=====
Transaction Summary
=====
Install 1 Package

Total download size: 1.4 M
Installed size: 4.4 M
Is this ok [y/N]: y
=====
Is this ok [y/N]: y
Downloading Packages:
postfix-3.5.9-24.el9.x86_64.rpm          9.8 MB/s | 1.4 MB   00:00
=====
Total                                         1.7 MB/s | 1.4 MB   00:00
=====
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :                                                 1/1
  Running scriptlet: postfix-2:3.5.9-24.el9.x86_64          1/1
  Installing  : postfix-2:3.5.9-24.el9.x86_64              1/1
  Running scriptlet: postfix-2:3.5.9-24.el9.x86_64          1/1
  Verifying   : postfix-2:3.5.9-24.el9.x86_64              1/1
  Installed   : postfix-2:3.5.9-24.el9.x86_64              1/1
=====
Complete!
=====
Transaction summary (use "dnf history" to see detailed info):
  Installed: postfix-2:3.5.9-24.el9.x86_64
```

Starting and Enabling Postfix Service:

- After installation, it's crucial to ensure that the Postfix service is running and will start automatically upon boot. This is done using:
 - `sudo systemctl start postfix`
 - `sudo systemctl enable postfix`
 - `sudo systemctl status postfix`(**status** command will verify that Postfix is active and running.)

```
[kashsna@goku ~]$ sudo systemctl start postfix
[sudo] password for kashsna:
[kashsna@goku ~]$ 
[kashsna@goku ~]$ sudo systemctl enable postfix
Created symlink /etc/systemd/system/multi-user.target.wants/postfix.service → /u
ser/lib/systemd/system/postfix.service.
[kashsna@goku ~]$ 
[kashsna@goku ~]$ sudo systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
   Loaded: loaded (/usr/lib/systemd/system/postfix.service; enabled; preset: en
abled)
   Active: active (running) since Fri 2024-09-13 00:05:56 +08; 31s ago
     Main PID: 4207 (master)
        Tasks: 3 (limit: 22420)
       Memory: 4.3M
          CPU: 449ms
         CGroup: /system.slice/postfix.service
                 └─4207 /usr/libexec/postfix/master -w
                   ├ 4208 pickup -l -t unix -u
                   ├ 4209 qmgr -l -t unix -u
                   └─4205 postfix-script[4205]: starting the

Sep 13 00:05:55 goku.rockyserver.org systemd[1]: Starting Postfix Mail Transport
Sep 13 00:05:55 goku.rockyserver.org postfix/postfix-script[4205]: starting the
Sep 13 00:05:56 goku.rockyserver.org postfix/master[4207]: daemon started -- ve
Sep 13 00:05:56 goku.rockyserver.org systemd[1]: Started Postfix Mail Transport

[kashsna@goku ~]$
```

Editing Postfix Main Configuration File:

- I modified the Postfix configuration file located at **/etc/postfix/main.cf** using:

```
[kashsna@goku ~]$ sudo nano /etc/postfix/main.cf
```

The following lines were uncommented and edited to configure the mail server:

```
GNU nano 5.6.1          /etc/postfix/main.cf
#
myhostname = goku.rockyserver.org
#myhostname = virtual.domain.tld

mydomain = rockyserver.org

myorigin = $myhostname
#myorigin = $mydomain

inet_interfaces = all
#inet_interfaces = all

#mydestination = $myhostname, localhost.$mydomain, localhost
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
#      mail.$mydomain, www.$mydomain, ftp.$mydomain

#relay_domains = $mydestination

#home_mailbox = mailbox
home_mailbox = Maildir/
```

These above settings define the mail server's identity, domain, and the format of user mailboxes. **Maildir** stores emails in a directory structure, making it ideal for handling larger volumes of emails.

Additional Configuration:

- At the end of the configuration file, I added specific lines (shown in the image below) to further customize Postfix. These might include settings for security, performance, or relay configurations based on your project.

```
GNU nano 5.6.1          /etc/postfix/main.cf          Modified
shlib_directory = /usr/lib64/postfix

#Enable SMTPS (SMTP over SSL/TLS)
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes

#Enable Dovecot SASL authentication
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_tls_security_level = may
#smtpd_recipient_restrictions = permit_sasl_authenticated, permit_mynetworks, reject_unauth_destination
smtpd_use_tls = yes
smtp_tls_security_level = may
```

Modifying Master.cf File:

- Next, I accessed the `/etc/postfix/master.cf` file and uncommented the required lines (refer to the image). This file defines how different Postfix services, such as SMTP, operate.

```
GNU nano 5.6.1                               /etc/postfix/master.cf                         Modified
# service type  private unpriv  chroot  wakeup  maxproc command + args
#          (yes)   (yes)    (no)    (never) (100)
# =====
smtp      inet  n      -       n      -       -       smtpd ←
#smtp     inet  n      -       n      -       1       postscreen
#smtpd    pass  -      -       n      -       -       smtpd
#dnsblog  unix  -      -       n      -       0       dnsblog
#tlsproxy  unix  -      -       n      -       0       tlsproxy
#submission inet n      -       n      -       -       smtpd
# -o syslog_name=postfix/submission
-o smtpd_tls_security_level=encrypt ←
# -o smtpd_sasl_auth_enable=yes
# -o smtpd_tls_auth_only=yes
# -o smtpd_reject_unlisted_recipient=no
# -o smtpd_client_restrictions=$mua_client_restrictions
# -o smtpd_helo_restrictions=$mua_helo_restrictions
# -o smtpd_sender_restrictions=$mua_sender_restrictions
# -o smtpd_recipient_restrictions=
-o smtpd_relay_restrictions=permit_sasl_authenticated,reject ←
-o milter_macro_daemon_name=ORIGINATING
smtps     inet  n      -       n      -       -       smtpd ←
# -o syslog_name=postfix/smtps
-o smtpd_tls_wrappermode=yes ←
-o smtpd_sasl_auth_enable=yes
# -o smtpd_reject_unlisted_recipient=no
# -o smtpd_client_restrictions=$mua_client_restrictions
# -o smtpd_helo_restrictions=$mua_helo_restrictions
# -o smtpd_sender_restrictions=$mua_sender_restrictions
# -o smtpd_recipient_restrictions=
# -o smtpd_relay_restrictions=permit_sasl_authenticated,reject ←
# -o milter_macro_daemon_name=ORIGINATING
```

Restarting Postfix:

- After all changes, Postfix was restarted to apply the new configurations:

```
[kashsna@goku ~]$ sudo systemctl restart postfix
[sudo] password for kashsna:
[kashsna@goku ~]$
```

Dovecot Installation and Configuration:

Installing Dovecot:

- To manage incoming emails, I installed **Dovecot**, an IMAP/POP3 server, using:

```
[kashsna@goku ~]$ sudo dnf install dovecot
Last metadata expiration check: 0:21:46 ago on Fri 13 Sep 2024 12:33:43 AM.
Dependencies resolved.
=====
 Package      Arch    Version           Repository  Size
=====
Installing:
 dovecot      x86_64  1:2.3.16-11.el9_4.1      appstream  4.7 M
Installing dependencies:
 clucene-core  x86_64  2.3.3.4-42.20130812.e8e3d20git.el9  appstream  585 k
 libexttextcat x86_64  3.4.5-11.el9            appstream  209 k

Transaction Summary
=====
Install 3 Packages

Total download size: 5.5 M
Installed size: 20 M
Is this ok [y/N]: y
```

Editing the Dovecot Main Configuration File:

- I opened the main Dovecot configuration file located at **/etc/dovecot/dovecot.conf** and added/uncommented these lines:

```
[kashsna@goku ~]$ sudo nano /etc/dovecot/dovecot.conf
[kashsna@goku ~]$

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

GNU nano 5.6.1          /etc/dovecot/dovecot.conf
#quota = mysql:/etc/dovecot/dovecot-dict-sql.conf.ext
}

# Most of the actual configuration gets included below. The
# first sorted by their ASCII value and parsed in that order
# in filenames are intended to make it easier to understand
!include conf.d/*.conf

# A config file can also tried to be included without giving
# it's not found:
!include_try local.conf

mail_location = maildir:~/Maildir
auth_mechanisms = plain login
```

This sets up the IMAP and POP3 protocols, which allow email clients to retrieve messages from the server. The **mail_location** directive points to the user's **Maildir** directory.

Configuring 10-mail.conf:

- In the **/etc/dovecot/conf.d/10-mail.conf** file, I added the same **mail_location** directive:

```
[kashsna@goku ~]$ sudo nano /etc/dovecot/conf.d/10-mail.conf
GNU nano 5.6.1                               /etc/dovecot/conf.d/10-mail.conf
# See doc/wiki/Variables.txt for full list. Some examples:
#
mail_location = maildir:~/Maildir
mail_location = mbox:~/mail.TNBOX=/var/mail/%u
```

Configuring 10-auth.conf:

- I then edited the **/etc/dovecot/conf.d/10-auth.conf** file to disable plaintext authentication and enable the desired mechanisms:
 - These settings ensure secure authentication when logging into the email server.

```
[kashsna@goku ~]$ sudo nano /etc/dovecot/conf.d/10-auth.conf
GNU nano 5.6.1                               /etc/dovecot/conf.d/10-auth.conf
# See also ssl=required setting.
disable_plaintext_auth = no

# NOTE: See also disable_plaintext_auth
auth_mechanisms = plain login
```

Configuring 10-master.conf:

- I then modified the **10-master.conf** file as shown in the image, which configures specific services like the mail delivery system (LMTP or local delivery). This ensures that Dovecot integrates properly with Postfix.

```
[kashsna@goku ~]$ sudo nano /etc/dovecot/conf.d/10-master.conf
GNU nano 5.6.1                               /etc/dovecot/conf.d/10-master.conf
}

service auth {
    unix_listener /var/spool/postfix/private/auth {
        mode = 0660
        user = postfix
        group = postfix
    }
}
```

Restarting Dovecot:

- After making these changes, I restarted Dovecot to apply the new settings:

```
[kashsna@goku ~]$
[kashsna@goku ~]$ sudo systemctl restart dovecot
```

Creating a User for Email:

- To test the setup, I added a user on the Rocky server. This user will receive and send emails:

```
[kashsna@goku ~]$  
[kashsna@goku ~]$ sudo useradd -m -s /sbin/nologin Kashave  
[sudo] password for kashsna:  
[kashsna@goku ~]$ sudo passwd Kashave  
Changing password for user Kashave.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.
```

The **-m** flag creates a home directory for the user, and **/sbin/nologin** ensures the user cannot log in via SSH.

Thunderbird (Ubuntu Client):

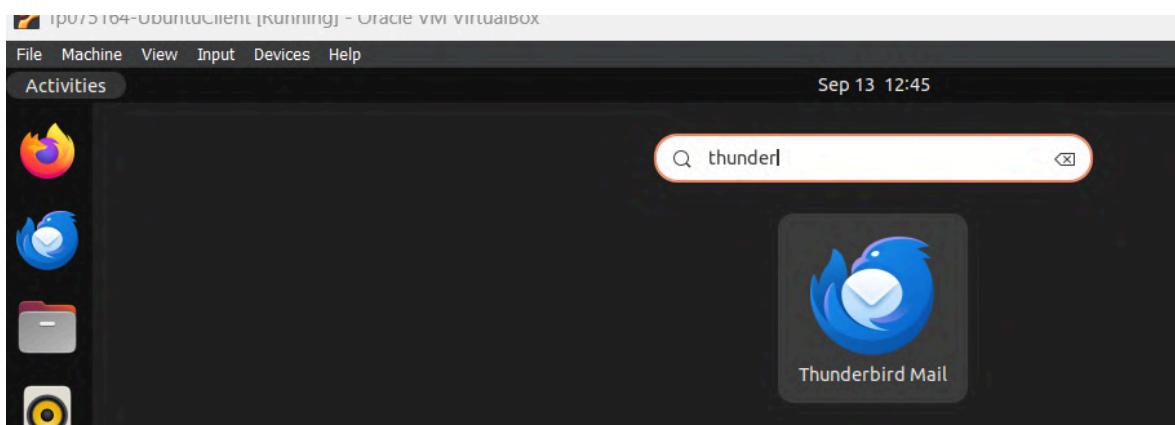
Installing Thunderbird:

- To test the email server, I installed **Thunderbird** on the Ubuntu client using:

```
kashubuntu@vegeta:~/Desktop$ sudo apt-get install thunderbird  
[sudo] password for kashubuntu:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
thunderbird is already the newest version (1:115.15.0+build1-0ubuntu0.22.04.1).  
thunderbird set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.  
kashubuntu@vegeta:~/Desktop$
```

Opening Thunderbird:

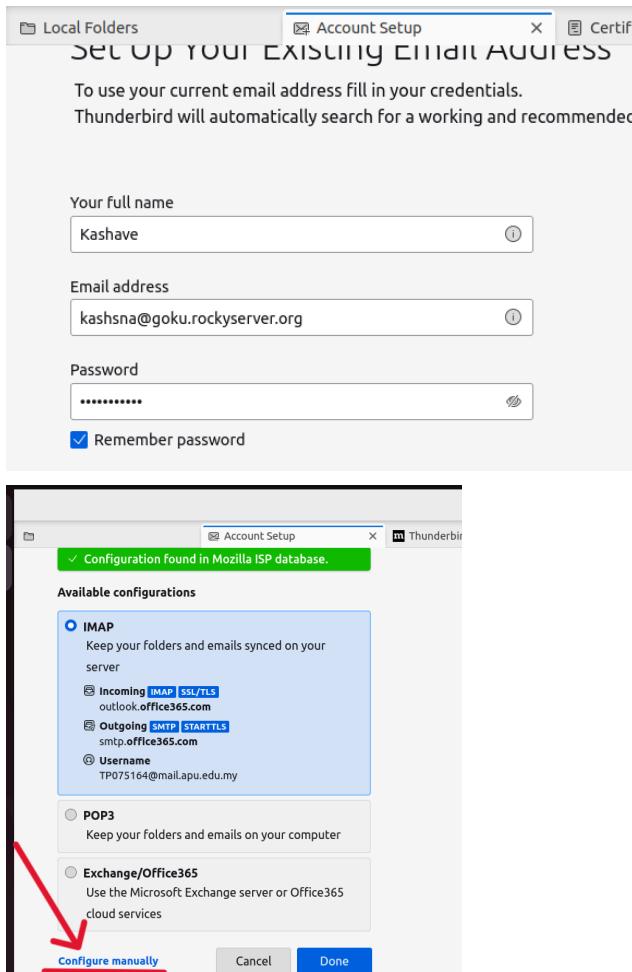
- Once installed, I launched Thunderbird by searching for it in the **Activities** menu. The software prompted me to set up an existing account.



Configuring Thunderbird Email Account:

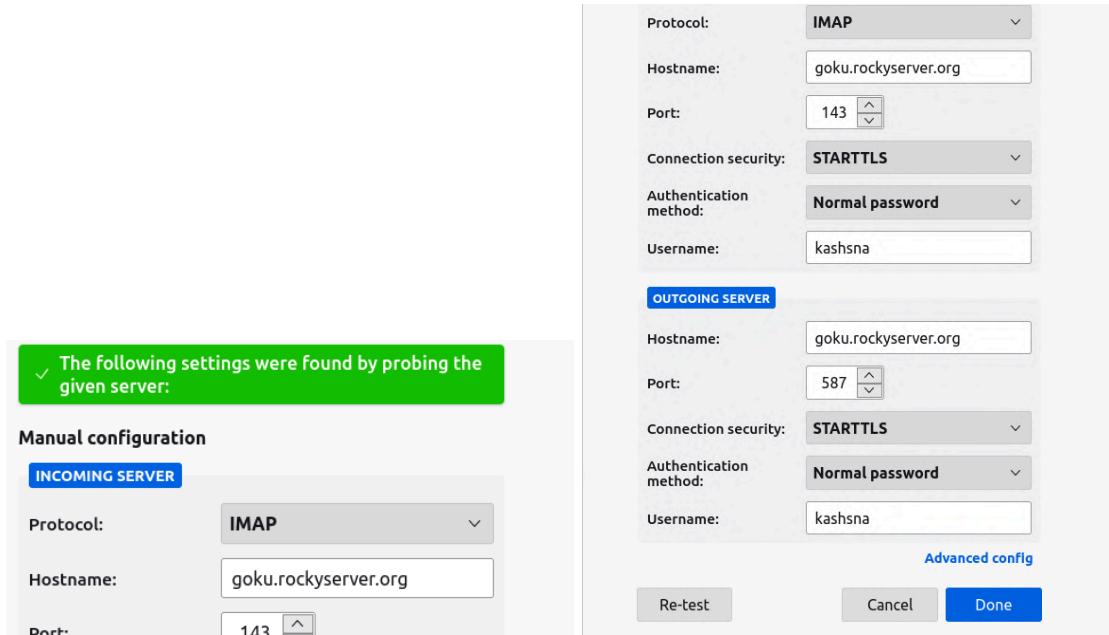
- I set up the email account with the following details:
 - Name:** Any name (e.g., "Kashave").
 - Email Address:** The format is `user@hostname` of the server. In my case, it's `kashsna@goku.rockyserver.org`.
 - Password:** The password I set for the user **Kashave** in Rocky Linux.

Under activities tab, search for thunderbird app



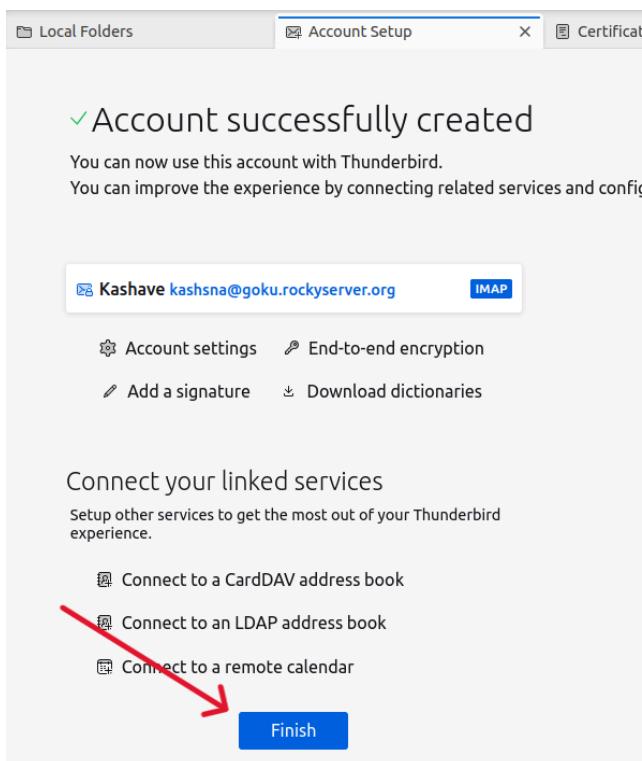
Manual Server Configuration:

- I clicked on **Configure Manually** to adjusted the server settings:
 - Incoming Server:** IMAP, Port **143 (STARTTLS)**, Authentication method: **Normal Password**.
 - Outgoing Server:** Port **587 (STARTTLS)**, Authentication method: **Normal Password**.
 - Username:** The email username (`kashsna` in my case).



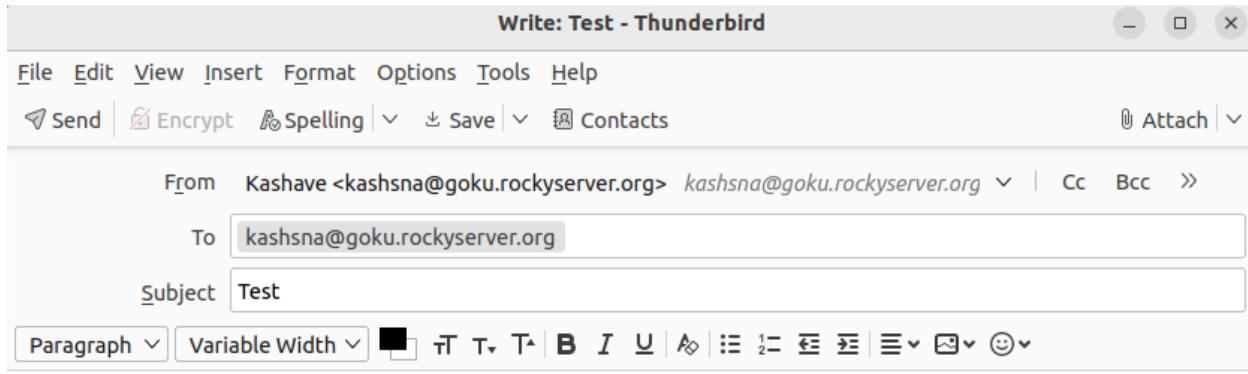
Account Creation Success:

- After entering these details, I clicked **Done** to create the account. Thunderbird confirmed the setup was successful, and I clicked **Finish** to proceed.

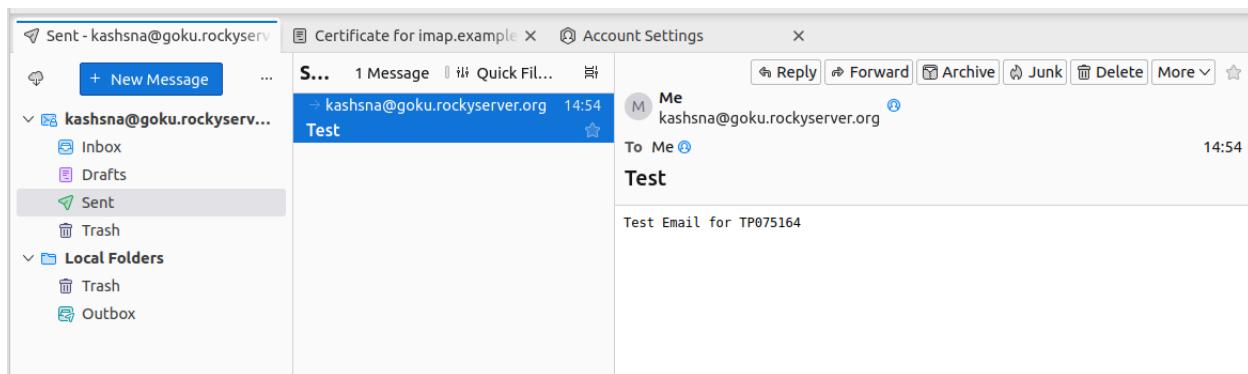


Sending an Email:

- To test the configuration, I composed a new email addressed to myself (kashsna@goku.rockyserver.org) and sent it.

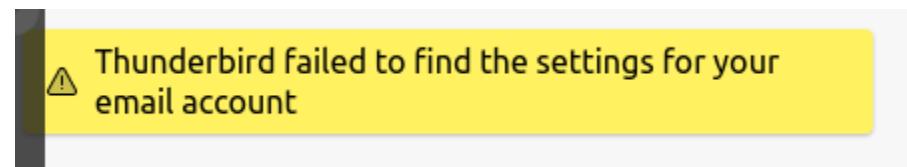


After navigating to the **Sent** folder, I verified that the email was sent successfully, as it appeared in the sent messages as seen in the image below:



Troubleshooting:

Error 1: Thunderbird failed to find the settings for the email account.



Main Issue: Missing DNS MX Records

- Reason for the Error:

- When attempting to set up Thunderbird, I received an error after clicking **Re-test** for the manual email configuration. This happened because the **MX (Mail Exchanger) records** were not configured in the DNS forward zone file, which is necessary for mail servers to route emails.

Fixing the MX and A Records in DNS:

- Open the DNS Forward Zone File:

- I needed to edit the **forward zone file** to add the missing **MX** and **A** records for mail routing. This ensures that emails can be sent and received via the configured email server.
- First, I opened the forward zone file (**fwd.rockyserver.org.db**) with:

```
[root@goku ~]# nano /var/named/rockyserver.org.zone
```

Adding MX and A Records:

- I added the following **MX record** to route emails through the mail server:
 - @ IN MX 10 goku.rockyserver.org.
 - This line tells the DNS system that the domain **rockyserver.org** handles email by routing it through **goku.rockyserver.org** with a priority of **10**.
- Next, I added the **A record** for the mail server:
 - mail IN A 159.69.39.4
 - This **A record** maps the hostname **goku.rockyserver.org** to its IP address **159.69.39.4**, which allows email clients like Thunderbird to resolve the mail server correctly.

```

GNU nano 5.6.1
$ORIGIN rockyserver.org.
$TTL 86400
@ IN SOA goku.rockyserver.org. root.rockyserver.org. (
    2001062503 ; serial
    21600       ; refresh after 6 hours
    3600        ; retry after 1 hour
    604800      ; expire after 1 week
    86400 )     ; minimum TTL of 1 day

@ IN NS goku.rockyserver.org.

goku IN A 159.69.39.4
vegeta IN A 159.69.39.5
@ IN MX 10 goku.rockyserver.org.

mail IN A 159.69.39.4

```

Increasing the Serial Number:

- Before exiting the file, I incremented the **serial number** at the top of the forward zone file to reflect that changes had been made. This is critical for DNS zone updates.

Restarting DNS (named):

- After making the necessary changes, I restarted the DNS service to apply the new settings:

```
[root@goku ~]#
[root@goku ~]# systemctl restart named
```

Verifying DNS Changes:

- To ensure the DNS changes were saved correctly, I used the following command to check the **MX record**:

```

[root@goku ~]# dig @localhost MX rockyserver.org
; <>> DiG 9.16.23-RH <>> @localhost MX rockyserver.org
; (2 servers found)
; global options: +cmd
; Got answer:
; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 19953
; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 545132838a73412d010000066e5a74c652c223bef3a24ba (good)
; QUESTION SECTION:
;rockyserver.org.          IN      MX

;; ANSWER SECTION:
rockyserver.org.      86400   IN      MX      10 goku.rockyserver.org.

;; ADDITIONAL SECTION:
goku.rockyserver.org. 86400   IN      A       159.69.39.4

;; Query time: 2 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sat Sep 14 23:10:04 +08 2024
;; MSG SIZE  rcvd: 109

```

In the Answer section, the MX record for rockyserver.org pointed to goku.rockyserver.org with a priority of 10, confirming that the DNS changes were applied successfully.

I also verified the **A record**:

```
[root@goku ~]# dig @localhost A mail.rockyserver.org

; <>> DiG 9.16.23-RH <>> @localhost A mail.rockyserver.org
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 14641
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 72e6ef9869200a6a010000066e5a76acd6470b88243dedb (good)
;; QUESTION SECTION:
mail.rockyserver.org.      IN      A

;; ANSWER SECTION: ←
mail.rockyserver.org. 86400  IN      A      159.69.39.4

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sat Sep 14 23:10:34 +08 2024
;; MSG SIZE rcvd: 93
```

The **A record** resolved to the correct IP address, **159.69.39.4**.

Firewall Configuration for Mail Services:

1. Allowing SMTP Port Through Firewall:

- Once the DNS records were correctly configured, I encountered another issue: the firewall was blocking the **SMTP port (Port 25)**, preventing mail from being sent or received. To fix this, I allowed the **SMTP** service through the firewall:

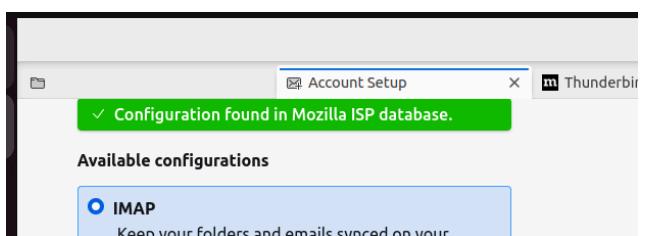
```
[root@goku ~]# sudo firewall-cmd --add-service=smtp --permanent
success
[root@goku ~]# firewall-cmd --reload
success
```

Allowing IMAP/IMAPS Ports:

- Similarly, I needed to allow the **IMAP** and **IMAPS** ports to enable incoming mail services:

```
[root@goku ~]# sudo firewall-cmd --add-service=imap --permanent
success
[root@goku ~]# firewall-cmd --add-service=imaps --permanent
success
[root@goku ~]# firewall-cmd --reload
success
```

Now thunderbird is able to verify the settings for our account.



Error 2: Protocol Connection is unavailable.

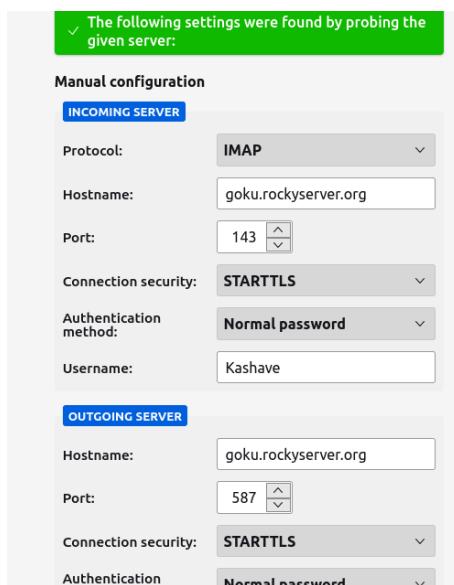
Configuration could not be verified. If your username and password are correct, it's likely that the server administrator has disabled the selected configuration for your account. Try selecting another protocol.

Reason for the Error:

- This error occurred because I initially configured Thunderbird to use **Port 993** (SSL/TLS) for the **incoming server** and **Port 465** (SSL/TLS) for the **outgoing server**. While these ports are standard for encrypted connections, the email server was not configured to support SSL/TLS on these ports.

Changing to STARTTLS:

- I modified the port settings to use **Port 143 (STARTTLS)** for the **incoming server** and **Port 587 (STARTTLS)** for the **outgoing server**.
 - Incoming server (IMAP):** Port 143 (STARTTLS).
 - Outgoing server (SMTP):** Port 587 (STARTTLS).
- After making these changes, Thunderbird was able to verify the email settings, and the connection was successfully established.



STARTTLS is a protocol that upgrades an existing connection to a secure one using TLS. It's often preferred over SSL/TLS on ports 993 and 465 for better compatibility with servers like Postfix and Dovecot, especially when using self-signed certificates or basic encryption, which may be what caused the error.

Error 3: Thunderbird was unable to match the User or Password to the ones stored in the Rocky User Database

 Unable to log in at server. Probably wrong configuration, username or password.

Issue: Authentication Failure for User Kashave

- Reason for the Error:

- When trying to log in to the email account using the **Kashave** user, Thunderbird outputted an authentication failure. Dovecot is unable to verify the user's credentials against the Rocky server's user database.

Check Dovecot Permissions for User Kashave:

- First, I verified if Dovecot had the necessary permissions to access the **Kashave** user's mailbox. The error message indicated that the **Kashave** authentication failed

```
[root@goku ~]# dovecadm auth test Kashave
Password:
passdb: Kashave auth failed
extra fields:
  user=kashave
  original_user=Kashave
[root@goku ~]#
```

Creating Mail Directory:

- To make sure the user's mailbox existed, I manually created the **Maildir** directory for **Kashave**:

```
[root@goku ~]# mkdir -p /Kashave/Kashave/Maildir
```

Setting Directory Permissions:

- After creating the directory, I ensured that the correct ownership and permissions were set:

```
[root@goku ~]# chown -R Kashave:Kashave /home/Kashave/Maildir
```

This command ensures that the **Kashave** user has full ownership over the **Maildir**, allowing Dovecot to properly manage incoming and outgoing mail.

Login Issue with nologin Shell:

- While trying to troubleshoot further, I attempted to access the **root** account for the user **Kashave**. However, the system returned an error stating the account was currently unavailable. Upon reviewing the user configuration, I realized that I had initially created the **Kashave** user with the **nologin** shell, which prevents the user from logging in interactively:

```
[root@goku ~]# su - Kashave
This account is currently not available.
```

```
[root@goku ~]# grep Kashave /etc/passwd
Kashave:x:1001:1001::/home/Kashave:/sbin/nologin
```

I changed the shell to **/bin/bash** to allow login access, but the authentication issue persisted, and **Kashave** was still unauthorized..

```
[root@goku ~]# sudo usermod -s /bin/bash Kashave
[root@goku ~]# dovecadm auth test Kashave
Password:
passdb: Kashave auth failed
extra fields:
  user=kashave
  original_user=Kashave
```

Testing with **kashsna** User:

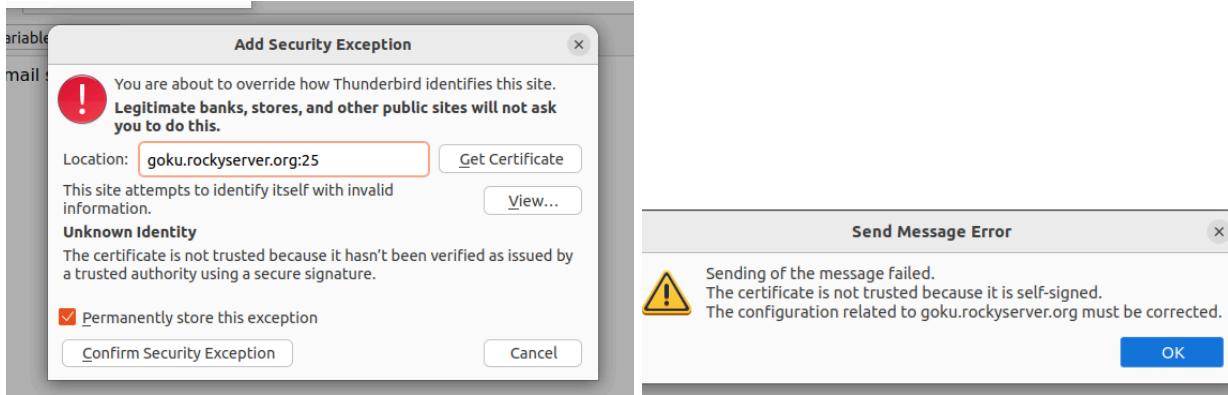
- I decided to test the setup using the **kashsna** user, which I had created during the Rocky Linux installation. This user was already part of the system's user base, and it successfully authenticated with Dovecot and Thunderbird.

```
[root@goku ~]# dovecadm auth test kashsna
Password:
passdb: kashsna auth succeeded
extra fields:
  user=kashsna
```

Conclusion:

The fact that the **kashsna** user was able to authenticate suggests that Dovecot had issues recognizing newly created users, like **Kashave**, for authentication. This might be due to a misconfiguration in how Dovecot accesses the system's user database or some delay in syncing new users for email authentication.

Error 4: Thunderbird does not trust the self-signed certificate on Rocky Postfix.



Reason for the Error:

- When I attempted to send an email via Thunderbird, it returned an error indicating that the certificate was not trusted. The message stated, "*The certificate is not trusted because it hasn't been verified as issued by a trusted authority using a secure signature.*" Even after checking the box for "Permanently store this exception," the email was still not sent, and the error persisted.

I decided to export the self-signed certificate from the Rocky Linux server to the Ubuntu client and upload it to Thunderbird as a trusted certificate.

Steps to Export the Self-Signed Certificate:

1. Finding the Self-Signed Certificate:

- To locate the self-signed certificate used by **Postfix**, I ran the following command on the Rocky Linux server: 'find / -type f \(-name "*.pem" -o -name "*.crt" -o -name "*.key" \) 2>/dev/null'

```
[root@goku ~]# find / -type f \(-name "*.pem" -o -name "*.crt" -o -name "*.key" \) 2>/dev/null
/run/named/session.key
/etc/pki/tls/certs/postfix.pem ←
/etc/pki/tls/private/postfix.key
/etc/pki/ca-trust/extracted/openssl/ca-bundle.trust.crt
/etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem
/etc/pki/ca-trust/extracted/pem/email-ca-bundle.pem
/etc/pki/ca-trust/extracted/pem/objsign-ca-bundle.pem
/etc/pki/ca-trust/extracted/pem/directory-hash/ACCVRAIZ1.pem
/etc/pki/ca-trust/extracted/pem/directory-hash/AC_RAIZ_FNMT-RCM.pem
/etc/pki/ca-trust/extracted/pem/directory-hash/COMODO_Certification_Authority.pem
/etc/pki/ca-trust/extracted/pem/directory-hash/AC_RAIZ_FNMT-RCM_SERVIDORES_SEGUROS.pem
/etc/pki/ca-trust/extracted/pem/directory-hash/ANF_Secure_Server_Root_CA.pem
/etc/pki/ca-trust/extracted/pem/directory-hash/COMODO_ECC_Certification_Authority.pem
/etc/pki/ca-trust/extracted/pem/directory-hash/Actalis_Authentication_Root_CA.pem
/etc/pki/ca-trust/extracted/pem/directory-hash/Certum_EC-384_CA.pem
```

This command searches for files with the **.pem**, **.crt**, and **.key** extensions, which are typically used for SSL/TLS certificates and keys.

2. Preparing for Certificate Transfer:

- Before exporting the certificate to my Ubuntu client, I needed to set up **SSH** on the client. Running `systemctl status ssh` showed that the SSH service was not installed:

```
[root@vegeta ~]# systemctl status ssh  
Unit ssh.service could not be found.
```

3. Installing OpenSSH Server on Ubuntu:

- I installed the **OpenSSH server** on the Ubuntu client to allow secure file transfer using **SCP**:

```
[root@vegeta ~]# sudo apt install openssh-server  
Reading package lists... Done
```

4. Opening Rocky Firewall for SSH:

- I opened **Port 22** (used for SSH) on the Rocky server's firewall and reloaded the firewall to save changes:

```
[root@goku ~]# sudo firewall-cmd --add-port=22/tcp --permanent  
success  
[root@goku ~]# sudo firewall-cmd --reload  
success
```

5. Exporting the Certificate Using SCP:

- Once the SSH server was installed and the Rocky firewall was configured, I used **SCP** to securely copy the **postfix.pem** certificate from the Rocky server to my Ubuntu client:

```
[root@goku ~]# scp /etc/pki/tls/certs/postfix.pem kashubuntu@159.69.39.60:~  
kashubuntu@159.69.39.60's password:  
postfix.pem                                              100% 1415     800.2KB/s   00:00  
[root@goku ~]#
```

This command transfers the **postfix.pem** file to the home directory of the **kashubuntu** user on the Ubuntu client.

6. Verifying the Transfer:

- I listed the contents of the home directory on the Ubuntu client to confirm that the certificate was successfully transferred:

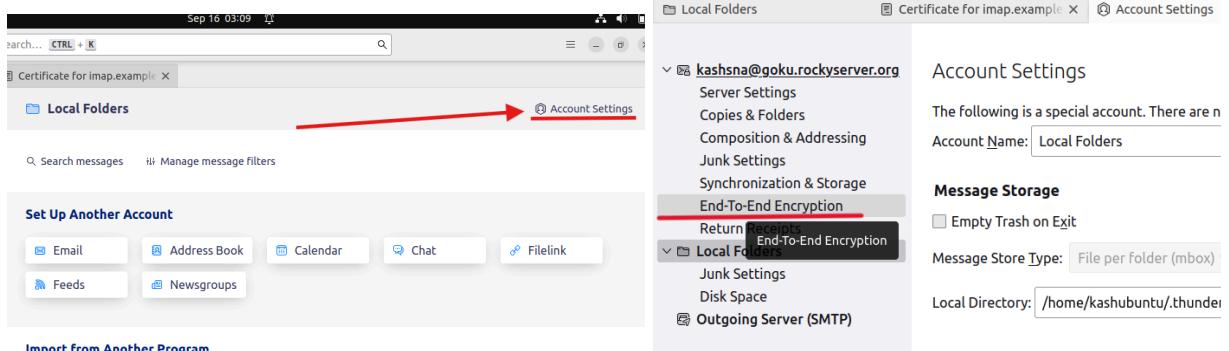
```
kashubuntu@vegeta:~/Desktop$ ls -l /home/kashubuntu  
total 40  
drwxr-xr-x 2 kashubuntu kashubuntu 4096 Ogos 28 22:45 Desktop  
drwxr-xr-x 2 kashubuntu kashubuntu 4096 Ogos 28 22:45 Documents  
drwxr-xr-x 2 kashubuntu kashubuntu 4096 Ogos 28 22:45 Downloads  
drwxr-xr-x 2 kashubuntu kashubuntu 4096 Ogos 28 22:45 Music  
drwxr-xr-x 2 kashubuntu kashubuntu 4096 Ogos 28 22:45 Pictures  
-rw-r--r-- 1 kashubuntu kashubuntu 1415 Sep 16 03:02 postfix.pem  
drwxr-xr-x 2 kashubuntu kashubuntu 4096 Ogos 28 22:45 Public  
drwx----- 4 kashubuntu kashubuntu 4096 Sep 1 01:44 snap  
drwxr-xr-x 2 kashubuntu kashubuntu 4096 Ogos 28 22:45 Templates  
drwxr-xr-x 2 kashubuntu kashubuntu 4096 Ogos 28 22:45 Videos  
kashubuntu@vegeta:~/Desktop$
```

The **postfix.pem** certificate appeared in the directory, confirming a successful export.

Importing the Certificate in Thunderbird:

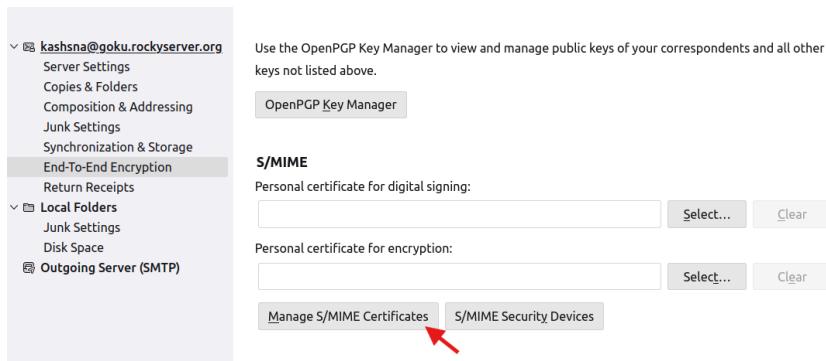
Access Thunderbird Account Settings:

- I opened **Thunderbird**, navigated to **Account Settings**, and scrolled to the **End-to-End Encryption** section.



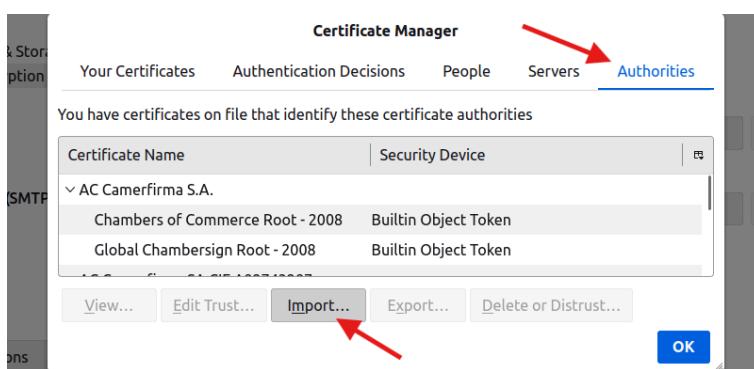
Managing S/MIME Certificates:

- In the **S/MIME** section, I clicked **Manage S/MIME Certificates**, which opens the **Certificate Manager**.

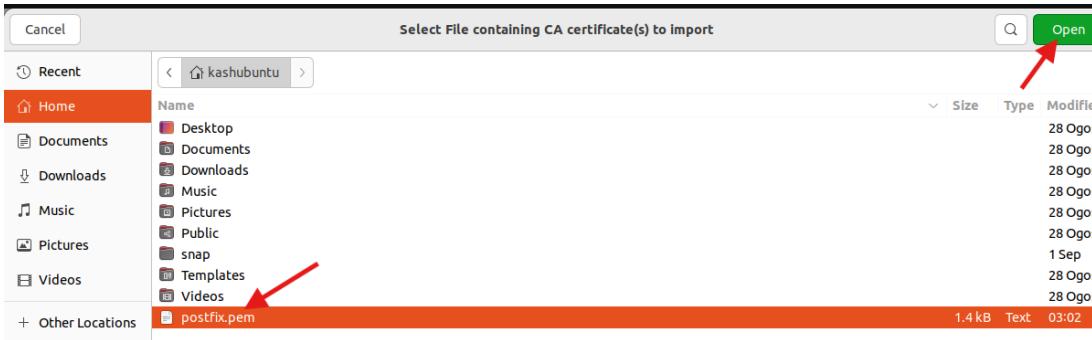


Importing the Certificate:

- In the **Certificate Manager**, I switched to the **Authorities** tab and clicked **Import**.

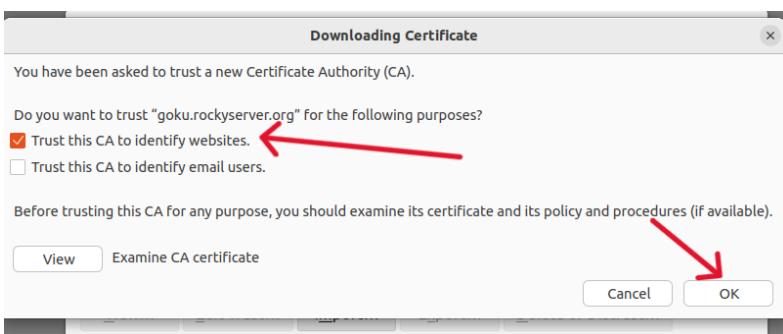


- Thunderbird opened the home directory of **kashubuntu**, where I selected the **postfix.pem** certificate and clicked **Open**.



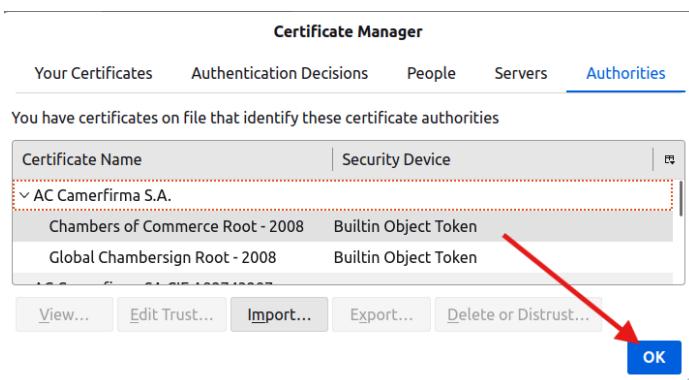
Trusting the Certificate:

- When importing the certificate, Thunderbird prompted two checkboxes. I ticked the box for:
 - **Trust this CA to identify websites**
- After confirming, I clicked **OK**.



Verifying the Certificate:

- After the certificate was successfully imported, I returned to the **Certificate Manager** to ensure the **postfix.pem** certificate was listed under the trusted authorities.



Error Resolved:

- With the self-signed certificate now trusted by Thunderbird, the error disappeared, and I was able to send emails without issues.

Error 5: Sasl Authentication not allowing thunderbird to verify the user password from rocky:

Reason for the Error:

- While trying to authenticate via Thunderbird, I encountered the error: "SASL authentication failure: cannot connect to saslauthd server: no such file or directory." This indicated that Postfix was unable to connect to the **SASL authentication daemon (saslauthd)**, which is required to verify user passwords.

Solution: Install and Configure saslauthd on Rocky Linux

1. Checking Postfix Mail Log for SASL Errors:

- First, I checked the **maillog** for more information on the SASL error:

```
[kashsna@goku ~]$ sudo tail -f /var/log/maillog | grep sasl
Sep 16 11:43:01 goku postfix/submission/smtpd[3257]: warning: SASL authentication failure: cannot connect to saslauthd server: No such file or directory
Sep 16 11:43:01 goku postfix/submission/smtpd[3257]: warning: SASL authentication failure: cannot connect to saslauthd server: No such file or directory
```

This returned the error: "**SASL authentication failure: cannot connect to saslauthd server: no such file or directory.**"

Installing Cyrus SASL:

- Upon investigating the status of the **saslauthd** service, I discovered that it was not installed. I installed the necessary packages using the following command:
bash

```
[kashsna@goku ~]$ sudo dnf install cyrus-sasl cyrus-sasl-plain cyrus-sasl-scram
Rocky Linux 9 - BaseOS           2.6 kB/s | 4.1 kB    00:01
Rocky Linux 9 - AppStream        3.2 kB/s | 4.5 kB    00:01
Rocky Linux 9 - Extras          2.9 kB/s | 2.9 kB    00:01
Package cyrus-sasl-plain-2.1.27-21.el9.x86_64 is already installed.
Dependencies resolved.
=====
Transaction Summary
=====
Install 2 Packages

Total download size: 99 k
Installed size: 191 k
Is this ok [y/N]:
```

These packages provide SASL (Simple Authentication and Security Layer) support for Postfix.

Enabling and Starting saslauthd:

- After installation, I enabled and started the **saslauthd** service:

```
[kashsna@goku ~]$
[kashsna@goku ~]$ sudo systemctl enable saslauthd
Created symlink /etc/systemd/system/multi-user.target.wants/saslauthd.service → /usr/lib/systemd/system/saslauthd.service.
[kashsna@goku ~]$
[kashsna@goku ~]$ sudo systemctl start saslauthd
[sudo] password for kashsna:
```

Configuring Postfix to Use SASL:

- I then needed to configure **Postfix** to use SASL for authentication. I opened the `/etc/postfix/main.cf` file and added the following lines:

```
[kashsna@goku ~]$ sudo vi /etc/postfix/main.cf
[...]
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
"/etc/postfix/main.cf" 750L, 29902B
```

These settings enable SASL authentication in Postfix and define the path where Postfix will communicate with **Dovecot** for user authentication.

Ensuring Correct saslauthd Directory Configuration

1. Verifying saslauthd Directory:

- Next, I checked the configuration in the `/etc/sysconfig/saslauthd` file to ensure that the directory was correctly set. This file specifies where the **saslauthd** socket should be created.

```
[kashsna@goku ~]$ sudo vi /etc/sysconfig/saslauthd
[...]
# Directory in which to place saslauthd's
# socket. This directory must already exist.
SOCKETDIR=/var/run/saslauthd
```

Restarting saslauthd and Postfix:

- After ensuring the configuration was correct, I restarted both **saslauthd** and **Postfix** to apply the changes:

bash

```
[kashsna@goku ~]$ sudo systemctl restart saslauthd
[kashsna@goku ~]$ sudo systemctl restart postfix
```

Checking saslauthd Directory:

- To confirm that the **saslauthd** directory existed and was properly configured, I ran:

```
[kashsna@goku ~]$ ls -l /var/run/saslauthd
total 4
srwxrwxrwx. 1 root root 0 Sep 16 12:09 mux
-rw-----. 1 root root 0 Sep 16 12:09 mux.accept
-rw-----. 1 root root 5 Sep 16 12:09 saslauthd.pid
```

Since the directory does not exist, Postfix will not be able to communicate with **saslauthd**.

Adding Postfix to saslauthd Group

1. Adding Postfix to saslauthd Group:

- Postfix requires access to the **saslauthd** directory to perform authentication. First, I attempted to add **Postfix** to the **saslauthd** group:

```
[kashsna@goku ~]$ sudo usermod -aG saslauthd postfix
usermod: group 'saslauthd' does not exist
```

However, the system returned an error indicating that the **saslauthd** group did not exist.

Creating the saslauthd Group:

- I created the **saslauthd** group using: **sudo groupadd saslauthd**
- After creating the group, I reran the command to add **Postfix** to the group:

```
[kashsna@goku ~]$ sudo groupadd saslauthd
[kashsna@goku ~]$ sudo usermod -aG saslauthd postfix
[kashsna@goku ~]$
```

Setting Permissions for the saslauthd Directory:

- After adding Postfix to the group, I set the correct permissions for the **saslauthd** directory:

```
[kashsna@goku ~]$ sudo chgrp saslauthd /var/run/saslauthd
[kashsna@goku ~]$ sudo chmod 750 /var/run/saslauthd
```

Restarting Services:

- Once again, I restarted **saslauthd** and **Postfix** to apply the changes:

```
[kashsna@goku ~]$ systemctl restart saslauthd
[kashsna@goku ~]$ sudo systemctl restart postfix
[kashsna@goku ~]$
```

Verifying Permissions:

- I verified that the **saslauthd** directory had the correct permissions:

```
[kashsna@goku ~]$ ls -ld /var/run/saslauthd
drwxr-x---. 2 root saslauthd 100 Sep 16 12:25 /var/run/saslauthd
```

The output showed that the directory was now correctly configured with the appropriate group ownership and permissions.

Configuring Dovecot for SASL

1. Editing Dovecot Configuration (10-master.conf):

- To ensure Dovecot properly supports SASL authentication, I opened the **/etc/dovecot/conf.d/10-master.conf** file. I added the **unix listener** configuration under **auth-userdb{}** and **auth-master{}**:

```
[kashsna@goku ~]$ sudo vi /etc/dovecot/conf.d/10-master.conf
```

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

These settings ensure that Postfix can communicate with Dovecot for authentication, and the correct permissions are applied to the **auth-master** socket.

Restarting Dovecot and Postfix:

- After modifying the configuration, I restarted **Dovecot** and **Postfix**:

```
[root@goku ~]# systemctl restart postfix
[root@goku ~]# systemctl restart dovecot
[root@goku ~]#
```

After configuring SASL with Dovecot and Postfix, I tested sending and receiving emails via Thunderbird, and the SASL authentication worked as expected.

8.0 Apache Webserver Setup

1. Checking for System Updates

- Before installing **Apache** on Rocky Linux, I ensured the system was up-to-date by running the update command:

```
[kashsna@goku ~]$ sudo dnf update -y
[sudo] password for kashsna:
Last metadata expiration check: 5:12:50 ago on Mon 16 Sep 2024 04:30:25 PM.
Dependencies resolved.
Nothing to do.
Complete!
[kashsna@goku ~]$ sudo dnf upgrade -y
Last metadata expiration check: 5:13:02 ago on Mon 16 Sep 2024 04:30:25 PM.
Dependencies resolved.
Nothing to do.
Complete!
[kashsna@goku ~]$
```

2. Installing Apache (httpd)

- To install **Apache**, I used the following command:

```
[kashsna@goku ~]$ sudo dnf install httpd -y
Last metadata expiration check: 5:14:33 ago on Mon 16 Sep 2024 04:30:25 PM.
Dependencies resolved.
=====
Package           Architecture Version      Repository  Size
=====
Installing:
httpd            x86_64      2.4.57-11.el9_4.1    appstream  44 k
Installing dependencies:
apr              x86_64      1.7.0-12.el9_3     appstream  122 k
apr-util         x86_64      1.6.1-23.el9      appstream  94 k
apr-util-bdb     x86_64      1.6.1-23.el9      appstream  12 k
httpd-core       x86_64      2.4.57-11.el9_4.1    appstream  1.4 M
httpd-filesystem noarch     2.4.57-11.el9_4.1    appstream  11 k
httpd-tools      x86_64      2.4.57-11.el9_4.1    appstream  79 k
rocky-logos-httpd noarch     90.15-2.el9      appstream  24 k
Installing weak dependencies:
apr-util-openssl x86_64      1.6.1-23.el9      appstream  14 k
mod_http2        x86_64      2.0.26-2.el9_4     appstream  162 k
mod_lua          x86_64      2.4.57-11.el9_4.1    appstream  58 k
Transaction Summary
=====
Install 11 Packages

Total download size: 2.0 M
Installed size: 6.0 M
Downloading Packages:
(1/11): rocky-logos-httpd-90.15-2.el9.noarch.rpm   40 kB/s | 24 kB  00:00
(2/11): mod_lua-2.4.57-11.el9_4.1.x86_64.rpm      85 kB/s | 58 kB  00:00
(3/11): httpd-tools-2.4.57-11.el9_4.1.x86_64.rpm  113 kB/s | 79 kB  00:00
(4/11): httpd-filesystem-2.4.57-11.el9_4.1.noarch.r 271 kB/s | 11 kB  00:00
(5/11): httpd-2.4.57-11.el9_4.1.x86_64.rpm       255 kB/s | 44 kB  00:00
(6/11): apr-util-bdb-1.6.1-23.el9.x86_64.rpm      182 kB/s | 12 kB  00:00
(7/11): apr-util-openssl-1.6.1-23.el9.x86_64.rpm   108 kB/s | 14 kB  00:00
(8/11): apr-util-1.6.1-23.el9.x86_64.rpm        628 kB/s | 94 kB  00:00
(9/11): apr-1.7.0-12.el9_3.x86_64.rpm        668 kB/s | 122 kB 00:00
(10/11): mod_http2-2.0.26-2.el9_4.x86_64.rpm    488 kB/s | 162 kB 00:00
(11/11): httpd-core-2.4.57-11.el9_4.1.x86_64.rpm 1.8 MB/s | 1.4 MB 00:00
```

After installation, I checked the Apache version to confirm it was installed correctly:

```
[kashsna@goku ~]$ sudo httpd -v
Server version: Apache/2.4.57 (Rocky Linux)
Server built:  Aug 5 2024 00:00:00
```

3. Starting and Enabling Apache Service

- Once Apache was installed, I started and enabled the **httpd** service so that it starts automatically on boot:

```
[kashsna@goku ~]$ sudo systemctl start httpd
[kashsna@goku ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service →
/usr/lib/systemd/system/httpd.service.
```

```
[kashsna@goku ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset)
   Active: active (running) since Mon 2024-09-16 21:50:22 +08; 4min 6s ago
     Docs: man:httpd.service(8)
 Main PID: 4170 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; B>
      Tasks: 177 (limit: 22420)
     Memory: 42.1M
        CPU: 605ms
      CGroup: /system.slice/httpd.service
              └─4170 /usr/sbin/httpd -DFOREGROUND
                  ├─4171 /usr/sbin/httpd -DFOREGROUND
                  ├─4172 /usr/sbin/httpd -DFOREGROUND
                  ├─4173 /usr/sbin/httpd -DFOREGROUND
                  ├─4174 /usr/sbin/httpd -DFOREGROUND

[kashsna@goku ~]$
```

The **status** command showed that Apache was up and running successfully.

4. Configuring the Firewall for HTTP and HTTPS

- I configured the firewall to allow **HTTP** and **HTTPS** traffic:

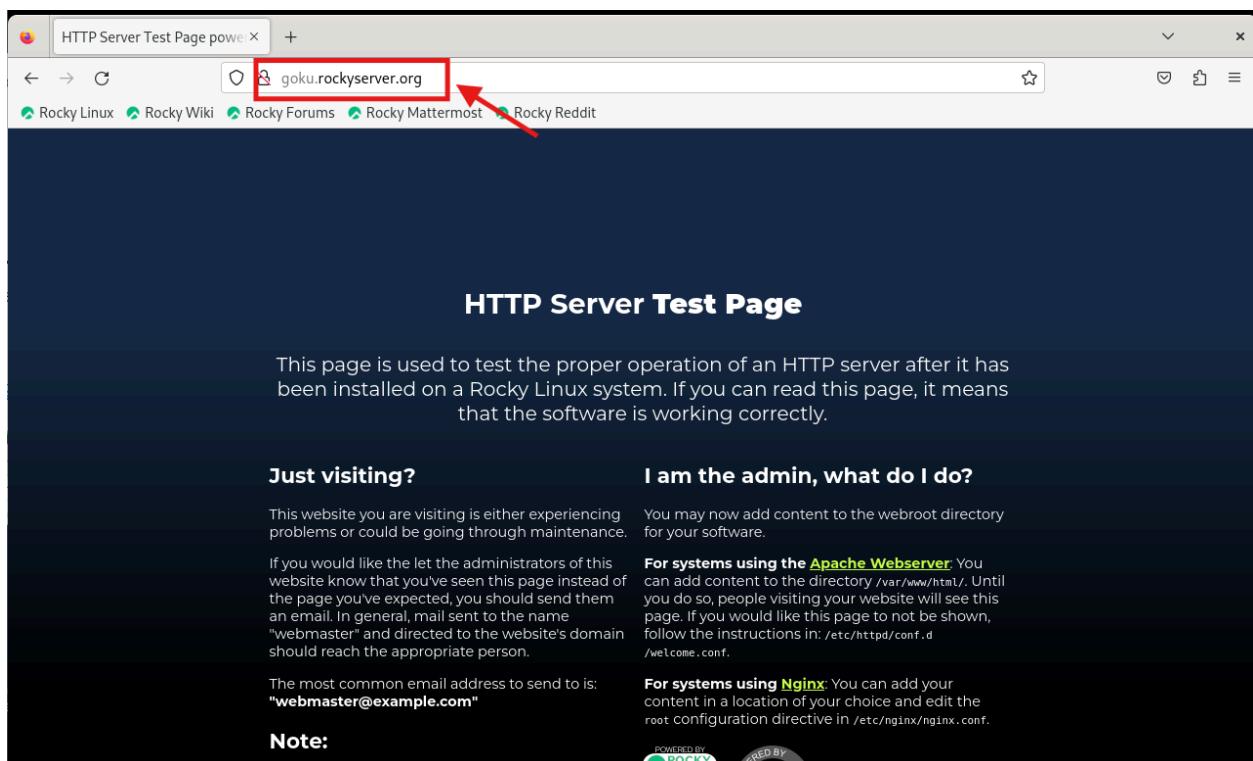
```
[kashsna@goku ~]$ sudo firewall-cmd --permanent --add-service=http
success
[kashsna@goku ~]$ sudo firewall-cmd --permanent --add-service=https
success
[kashsna@goku ~]$ sudo firewall-cmd --reload
success
```

To verify that **HTTP** and **HTTPS** were allowed, I listed all the firewall services:

```
[kashsna@goku ~]$ sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcp dhcpcv6-client dns http https imap imaps smtp ssh
  ports: 53/tcp 53/udp 993/tcp 465/tcp 80/tcp 443/tcp 587/tcp 143/tcp 25/tcp
  22/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[kashsna@goku ~]$
```

5. Testing Apache HTTP Server

- After configuring Apache, I tested the default Apache HTTP server page by entering either the **hostname** or **IP address** of the Rocky server into a browser.
 - When navigating to the **IP address** or **hostname**, I was successfully taken to the **Apache HTTP Server Test Page**, indicating that Apache was correctly set up and running.



Configuring an Apache Virtual Host

1. Checking Server Information

- Before setting up a virtual host, I verified the **IP address** and **hostname** of the Rocky server:

```
[kashsna@goku ~]$ hostname  
goku.rockyserver.org  
[kashsna@goku ~]$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 159.69.39.4 brd 159.69.39.255 netmask 255.255.255.0  
        ether 08:00:27:5a:be:14 txqueuelen 1000 (Ethernet)  
        RX packets 2639 bytes 2215164 (2.1 MiB)  
        RX errors 0 dropped 0 overruns 0 frame 0
```

2. Backing Up Apache Configuration Files

- To avoid losing any important configurations, I backed up the default Apache configuration files:

```
[kashsna@goku ~]$ sudo cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.bak  
[sudo] password for kashsna:
```

I confirmed the backup by listing the contents of the directory:

```
[kashsna@goku ~]$ cd /etc/httpd/conf  
[kashsna@goku conf]$ ls -ltr  
total 40  
-rw-r--r--. 1 root root 12005 Aug  9 00:28 httpd.conf  
-rw-r--r--. 1 root root 13430 Aug  9 00:29 magic  
-rw-r--r--. 1 root root 12005 Sep 16 22:14 httpd.conf.bak  
[kashsna@goku conf]$
```

3. Navigating to the Apache Sites Directory

- I then navigated to the **sites-available** directory (or **/etc/httpd/conf.d** for Apache):

```
[root@goku ~]#  
[root@goku ~]# cd /etc/httpd/conf.d/  
[root@goku conf.d]#
```

4. Creating a New Virtual Host File

- I created a new virtual host file to define the settings for the new domain:

```
[root@goku conf.d]# nano TP075164.rockyserver.org.conf
```

In the file, I added the following configuration:

```
GNU nano 5.6.1          TP075164.rockyserver.org.conf      Modified
<VirtualHost *:80>
    ServerAdmin admin@rockyserver.org
    ServerName TP075164.rockyserver.org
    DocumentRoot /var/www/html/TP075164.rockyserver.org
    ErrorLog /var/log/httpd/TP075164.rockyserver.org-error.log
    CustomLog /var/log/httpd/TP075164.rockyserver.org-access.log combined
</VirtualHost>
```

Explanation:

- **ServerAdmin**: The email address of the server admin.
- **DocumentRoot**: The directory where website files are stored.
- **ServerName**: The domain name for the virtual host.
- **ErrorLog** and **CustomLog**: Paths to log files.

5. Creating and Setting Permissions for the Website Directory

- I created a directory to store the website files for the new domain using ***sudo mkdir***
- I set the proper ownership and permissions for the directory using ***sudo chown***

```
[root@goku conf.d]# sudo mkdir -p /var/www/html/TP075164.rockyserver.org
[root@goku conf.d]#
[root@goku conf.d]# sudo chown -R $USER:$USER /var/www/html/TP075164.rockyserver.org
```

Create the directory where our website files will be stored: *mkdir*

Set proper permissions: *chown*

```
[root@goku conf.d]# sudo ls -l /var/www/html/TP075164.rockyserver.org/index.html
ls: cannot access '/var/www/html/TP075164.rockyserver.org/index.html': No such file or
directory
```

Verify if permission is set. However we see that there is no such file or directory when we run it.

6. Adding Basic HTML to the Website Directory

- Enter the ***/var/www/html/TP075164.rockyserver.org/index.html*** file.
- I added a simple HTML file to ensure the directory wasn't empty:

```
[root@goku conf.d]#
[root@goku conf.d]# sudo nano /var/www/html/TP075164.rockyserver.org/index.html
```

I added the following HTML content:

```
GNU nano 5.6.1          /var/www/html/TP075164.rockyserver.org/index.html      Modified
<html>
<head>
    <title>Welcome to TP075164.rockyserver.org</title>
</head>
<body>
    <h1>Hello World! The virtual host for Apache is working!</h1>
</body>
</html>
```

Now we can see the permissions for the file.

```
[root@goku conf.d]# sudo ls -l /var/www/html/TP075164.rockyserver.org/index.html
-rw-r--r--. 1 root root 150 Sep 16 22:58 /var/www/html/TP075164.rockyserver.org/index.html
```

7. Checking Apache Configuration and Restarting the Service

- I checked for any syntax errors in the Apache configuration files:

```
[root@goku conf.d]# sudo apachectl configtest
Syntax OK
```

- After verifying there were no syntax errors, I restarted the Apache service to apply the changes and checked the status:

```
[root@goku conf.d]# systemctl restart httpd
[root@goku conf.d]# systemctl status httpd
● httpd.service - The Apache HTTP Server
    Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
    Active: active (running) since Mon 2024-09-16 23:09:05 +08; 57s ago
      Docs: man:httpd.service(8)
   Main PID: 3960 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0"
        Tasks: 177 (limit: 22420)
      Memory: 31.9M
        CPU: 233ms
       CGroup: /system.slice/httpd.service
               └─3960 /usr/sbin/httpd -DFOREGROUND
                  ├─3961 /usr/sbin/httpd -DFOREGROUND
                  ├─3962 /usr/sbin/httpd -DFOREGROUND
                  ├─3963 /usr/sbin/httpd -DFOREGROUND
                  ├─3964 /usr/sbin/httpd -DFOREGROUND

Sep 16 23:09:05 goku.rockyserver.org httpd[3960]: Server configured, listening on: port 80
Sep 16 23:09:05 goku.rockyserver.org systemd[1]: Started The Apache HTTP Server.

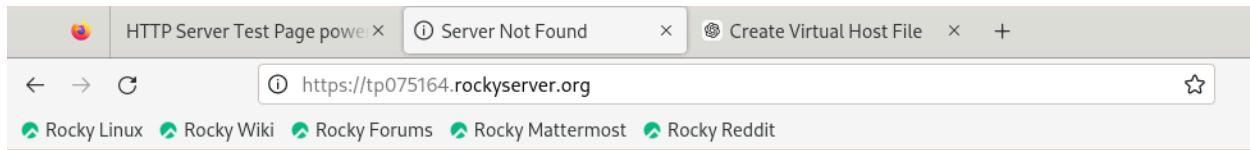
[root@goku conf.d]#
```

Testing the Apache Virtual Host

Testing on Rocky Linux

TROUBLESHOOTING:

- I tested the virtual host by entering the domain name **TP075164.rockyserver.org** in a browser. However, I received an error stating that the **server was not found**.



Hmm. We're having trouble finding that site.

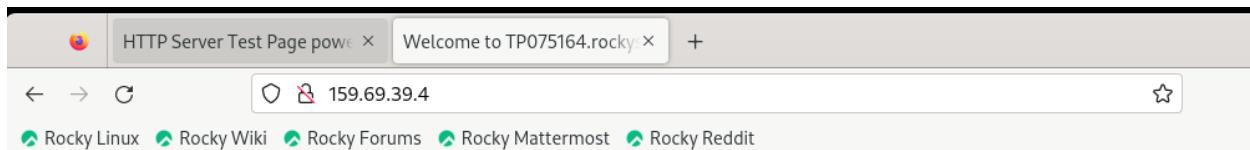
We can't connect to the server at tp075164.rockyserver.org.

If you entered the right address, you can:

- Try again later
- Check your network connection
- Check that Firefox has permission to access the web (you might be connected but behind a firewall)

[Try Again](#)

When I tried accessing the website using the server's **IP address**, the site loaded successfully, indicating a DNS misconfiguration for the domain name.



Hello World! The virtual host for Apache is working!

To fix this, I added the virtual host's **IP address** and **domain name** to the DNS configuration file on the Rocky server.

```
GNU nano 5.6.1                               /var/named/fwd.rockyserver.org.db

@      IN      NS      goku.rockyserver.org.

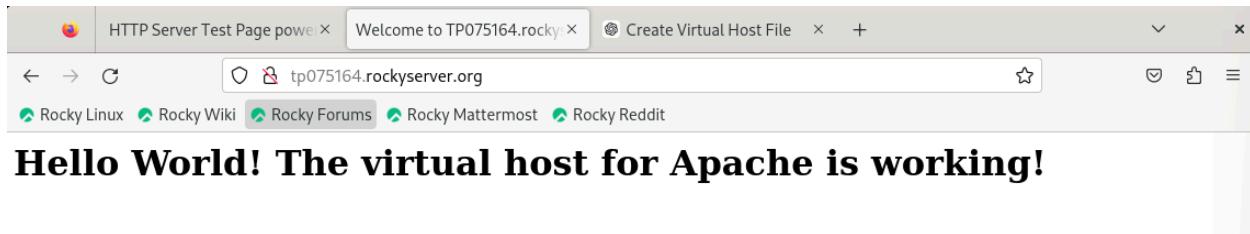
goku   IN      A       159.69.39.4
vegeta IN      A       159.69.39.5

@      IN      MX      10      goku.rockyserver.org.

mail   IN      A       159.69.39.4

TP075164          IN      A       159.69.39.4
```

After making the DNS changes, the website successfully loaded when using the **virtual hostname**.



Testing on Ubuntu Client

- I then tested the website on the Ubuntu client by entering **TP075164.rockyserver.org** in the browser. The website loaded successfully, confirming that the virtual host and DNS were configured properly.



9.0 SSL/TLS Configuration

1. Generate a New Private Key

1. Open a Terminal on Rocky Linux:

- Run sudo dnf install openssl. In my case it has already been installed.

```
[root@goku ~]# sudo dnf install openssl
Last metadata expiration check: 2:04:32 ago on Mon 16 Sep 2024 09:50:37 PM.
Package openssl-1:3.0.7-27.el9.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@goku ~]#
```

Run the following command to generate a new private key:

- openssl genpkey -algorithm RSA -out /etc/pki/tls/private/new_server.key -aes256**

```
[root@goku ~]# openssl genpkey -algorithm RSA -out /etc/pki/tls/private/server2.key -aes256
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
[root@goku ~]#
```

- I am prompted to enter a passphrase for the private key.

2. Generate a Certificate Signing Request (CSR)

1. Create a CSR:

- Execute the following command to generate a CSR:

openssl req -new -key /etc/pki/tls/private/new_server.key -out /etc/pki/tls/server2.csr

```
[root@goku ~]# openssl req -new -key /etc/pki/tls/private/server2.key -out /etc/pki/tls/server2.csr
Enter pass phrase for /etc/pki/tls/private/server2.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]:MY
State or Province Name (full name) []:Kuala Lumpur
Locality Name (eg, city) [Default City]:Kuala Lumpur
Organization Name (eg, company) [Default Company Ltd]:APU
Organizational Unit Name (eg, section) []:CS
Common Name (eg, your name or your server's hostname) []:goku.rockyserver.org
Email Address []:kashna@goku.rockyserver.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:Raikiri1000
An optional company name []:APU
[root@goku ~]#
```

Fill in the required information when prompted. This information will be included in the certificate.

3. Generate a Self-Signed Certificate

1. Create the Certificate:

- Run the following command to generate a self-signed certificate:

```
openssl x509 -req -days 365 -in /etc/pki/tls/server2.csr -signkey
```

```
/etc/pki/tls/private/new_server.key -out /etc/pki/tls/certs/server2.crt
```

```
[root@goku ~]# openssl x509 -req -days 365 -in /etc/pki/tls/server2.csr -signkey /etc/pki/tls/private/server2.key -out /etc/pki/tls/certs/server2.crt
Enter pass phrase for /etc/pki/tls/private/server2.key:
Certificate request self-signature ok
subject=C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
[root@goku ~]#
```

I set the ‘-days’ parameter to 365 days(1 year)

Update Postfix Configuration

1. Edit `/etc/postfix/main.cf`:

- Open the configuration file:

```
[root@goku ~]# nano /etc/postfix/main.cf
```

Update the following settings with the new certificate paths:

```
GNU nano 5.6.1                               /etc/postfix/main.cf
#
smtpd_tls_cert_file = /etc/pki/tls/certs/server2.crt

# The full pathname of a file with the Postfix SMTP server RSA private key.
# in PEM format. The private key must be accessible without a password,
# i.e. it must not be encrypted.
#
smtpd_tls_key_file = /etc/pki/tls/private/server2_no_password.key
```

Edit `/etc/postfix/master.cf`:

- Open the configuration file:

```
[root@goku ~]# nano /etc/postfix/master.cf
```

Ensure the **submission** service is configured as follows:

```
GNU nano 5.6.1          /etc/postfix/master.cf
#dnsblog    unix  -      n      -      0      dnsblog
#tlsproxy   unix  -      n      -      0      tlsproxy
submission inet n      -      n      -      -      smtpd
-o syslog_name=postfix/submission
-o smtpd_tls_security_level=encrypt
-o smtpd_sasl_auth_enable=yes
-o smtpd_tls_cert_file=/etc/pki/tls/certs/server2.crt
-o smtpd_tls_key_file=/etc/pki/tls/private/server2_no_password.key
-o smtpd_tls_auth_only=yes
# -o smtpd_reject_unlisted_recipient=no
# -o smtpd_client_restrictions=$mua_client_restrictions
# -o smtpd_helo_restrictions=$mua_helo_restrictions
# -o smtpd_sender_restrictions=$mua_sender_restrictions
# -o smtpd_recipient_restrictions=
-o smtpd_relay_restrictions=permit_sasl_authenticated,reject_unauth_destination
-o smtpd_client_restrictions=permit_mynetworks,permit_sasl_authenticated,reject_unauth_destination
-o milter_macro_daemon_name=ORIGINATING
```

Restart Postfix:

- **Apply the configuration changes by restarting Postfix**

```
[root@goku ~]# systemctl restart postfix
[root@goku ~]# systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
   Loaded: loaded (/usr/lib/systemd/system/postfix.service; enabled; preset: en>
   Active: active (running) since Wed 2024-09-18 22:00:16 +08; 39min ago
     Process: 3627 ExecStartPre=/usr/sbin/restorecon -R /var/spool/postfix/pid (>
     Process: 3629 ExecStartPre=/usr/libexec/postfix/aliasesdb (code=exited, sta>
     Process: 3631 ExecStartPre=/usr/libexec/postfix/chroot-update (code=exited,>
     Process: 3632 ExecStart=/usr/sbin/postfix start (code=exited, status=0/SUCC>
   Main PID: 3700 (master)
     Tasks: 3 (limit: 22420)
   Memory: 3.2M
     CPU: 509ms
   CGroup: /system.slice/postfix.service
           └─3700 /usr/libexec/postfix/master -w
             ├─3701 pickup -l -t unix -u
             ├─3702 qmgr -l -t unix -u

Sep 18 22:00:15 goku.rockyserver.org systemd[1]: Starting Postfix Mail Transport Agent...
Sep 18 22:00:16 goku.rockyserver.org postfix/postfix-script[3698]: starting the Postfix mail system...
Sep 18 22:00:16 goku.rockyserver.org postfix/master[3700]: daemon started -- version 3.6.1
Sep 18 22:00:16 goku.rockyserver.org systemd[1]: Started Postfix Mail Transport Agent.
```

Set SELinux Contexts for New Files

1. Add SELinux Contexts:

- Run the following commands to set the correct SELinux contexts:

```
[root@goku ~]# sudo semanage fcontext -a -t cert_t "/etc/pki/tls/private(.*?)?"  
sudo semanage fcontext -a -t cert_t "/etc/pki/tls/certs(.*?)?"
```

Restore Contexts:

- Apply the contexts to the directories

```
[root@goku ~]#  
[root@goku ~]# sudo restorecon -Rv /etc/pki/tls/private  
sudo restorecon -Rv /etc/pki/tls/certs  
[root@goku ~]#
```

Configure Dovecot for SSL/TLS

1. Edit Dovecot SSL Configuration:

- Open the Dovecot SSL configuration file:

```
[root@goku ~]# sudo nano /etc/dovecot/conf.d/10-ssl.conf
```

Update the settings with the new certificate paths:

```
GNU nano 5.6.1           /etc/dovecot/conf.d/10-ssl.conf      Mo  
ssl = yes  
  
# PEM encoded X.509 SSL/TLS certificate and private key. They're opened  
# 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.c  
ssl_cert = </etc/pki/tls/certs/server2.crt  
ssl_key = </etc/pki/tls/private/server2_no_password.key
```

Restart Dovecot:

- Restart the Dovecot service to apply changes:

```
[root@goku ~]# systemctl restart dovecot  
[root@goku ~]#
```

Check the status of Dovecot:

```
[root@goku ~]# systemctl status dovecot
● dovecot.service - Dovecot IMAP/POP3 email server
  Loaded: loaded (/usr/lib/systemd/system/dovecot.service; enabled; preset: >
  Active: active (running) since Wed 2024-09-18 22:00:21 +08; 38min ago
    Docs: man:dovecot(1)
           https://doc.dovecot.org/
   Process: 3709 ExecStartPre=/usr/libexec/dovecot/prestartscript (code=exited)
 Main PID: 3716 (dovecot)
   Status: "v2.3.16 (7e2e900c1a) running"
     Tasks: 4 (limit: 22420)
    Memory: 5.1M
       CPU: 76ms
      CGroup: /system.slice/dovecot.service
              └─3716 /usr/sbin/dovecot -F
                  ├─3717 dovecot/anvil
                  ├─3718 dovecot/log
                  ├─3719 dovecot/config

Sep 18 22:00:21 goku.rockyserver.org systemd[1]: Starting Dovecot IMAP/POP3 ema>
Sep 18 22:00:21 goku.rockyserver.org dovecot[3716]: master: Dovecot v2.3.16 (7e>
Sep 18 22:00:21 goku.rockyserver.org systemd[1]: Started Dovecot IMAP/POP3 emai>
```

Configure Apache for SSL/TLS

1. Install and Enable SSL Module:

- Install the SSL module for Apache:

```
[kashsna@goku ~]$ sudo dnf install mod_ssl
[sudo] password for kashsna:
Sorry, try again.
[sudo] password for kashsna:
Rocky Linux 9 - BaseOS           5.4 kB/s | 4.1 kB    00:00
Rocky Linux 9 - AppStream         5.8 kB/s | 4.5 kB    00:00
Rocky Linux 9 - Extras           2.9 kB/s | 2.9 kB    00:01
Dependencies resolved.
=====
 Package      Architecture Version       Repository      Size
 =====
 Installing:
 mod_ssl      x86_64        1:2.4.57-11.el9_4.1   appstream      108 k

Transaction Summary
=====
Install 1 Package

Total download size: 108 k
Installed size: 268 k
Is this ok [y/N]:
```

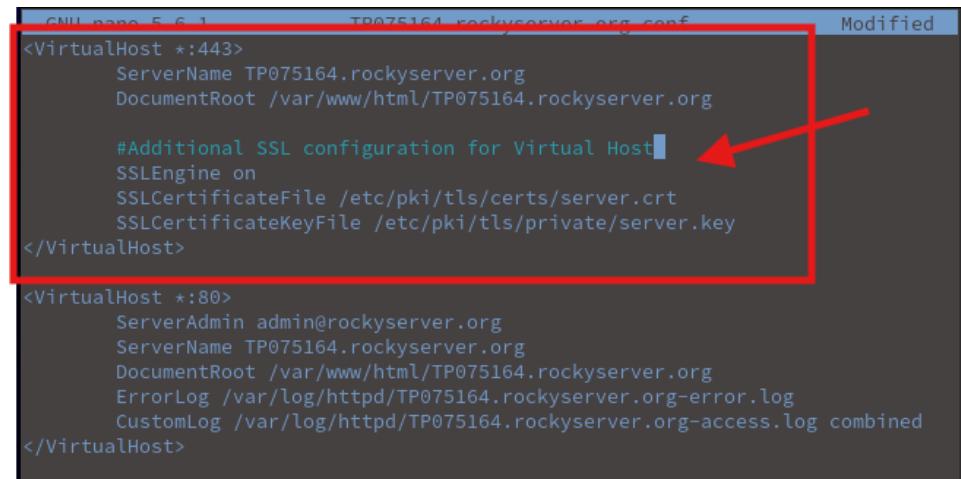
Enable the SSL module and configure it to handle HTTPS requests..

Update Apache SSL Configuration:

- Open the Apache SSL configuration file:

```
[kashsna@goku ~]$ cd /etc/httpd/conf.d/
[kashsna@goku conf.d]$ sudo nano TP075164.rockyserver.org.conf
[sudo] password for kashsna:
```

Add the following configuration lines to point to the new certificate files:



```
GNU nano 5.6.1          TP075164.rockyserver.org.conf          Modified
<VirtualHost *:443>
    ServerName TP075164.rockyserver.org
    DocumentRoot /var/www/html/TP075164.rockyserver.org

    #Additional SSL configuration for Virtual Host
    SSLEngine on
    SSLCertificateFile /etc/pki/tls/certs/server.crt
    SSLCertificateKeyFile /etc/pki/tls/private/server.key
</VirtualHost>

<VirtualHost *:80>
    ServerAdmin admin@rockyserver.org
    ServerName TP075164.rockyserver.org
    DocumentRoot /var/www/html/TP075164.rockyserver.org
    ErrorLog /var/log/httpd/TP075164.rockyserver.org-error.log
    CustomLog /var/log/httpd/TP075164.rockyserver.org-access.log combined
</VirtualHost>
```

Set Permissions for Certificate Files:

- Ensure the certificate files have appropriate permissions:

```
[root@goku ~]#  
[root@goku ~]# mkdir /etc/pki/tls  
mkdir: cannot create directory '/etc/pki/tls': File exists  
[root@goku ~]#  
[root@goku ~]# sudo chmod 600 /etc/pki/tls/private/server.key  
[root@goku ~]# sudo chmod 644 /etc/pki/tls/certs/server.crt  
[root@goku ~]#  
[root@goku ~]# sudo chown root:root /etc/pki/tls/private/server.key  
[root@goku ~]# sudo chown root:root /etc/pki/tls/certs/server.crt
```

Restart Apache:

- Restart the Apache service to apply changes:

```
[root@goku ~]#  
[root@goku ~]# sudo systemctl restart httpd
```

Check the status of Apache to ensure it is running without errors:

```
[root@goku ~]# sudo systemctl status httpd  
● httpd.service - The Apache HTTP Server  
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)  
  Active: active (running) since Tue 2024-09-17 12:34:36 +08; 54s ago  
    Docs: man:httpd.service(8)  
   Main PID: 3358 (httpd)  
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"  
        Tasks: 177 (limit: 22420)  
       Memory: 34.7M  
         CPU: 439ms  
        CGroup: /system.slice/httpd.service  
                  ├─3358 /usr/sbin/httpd -DFOREGROUND  
                  ├─3359 /usr/sbin/httpd -DFOREGROUND  
                  ├─3360 /usr/sbin/httpd -DFOREGROUND  
                  ├─3361 /usr/sbin/httpd -DFOREGROUND  
                  ├─3362 /usr/sbin/httpd -DFOREGROUND  
  
Sep 17 12:34:36 goku.rockyserver.org systemd[1]: Starting The Apache HTTP Server...  
Sep 17 12:34:36 goku.rockyserver.org httpd[3358]: Server configured, listening on: port 443, port 80  
Sep 17 12:34:36 goku.rockyserver.org systemd[1]: Started The Apache HTTP Server.
```

8. Test SSL/TLS Encryption

Testing Postfix SSL/TLS:

- On Rocky Server:

- Run the following command to check if Postfix is correctly using SSL/TLS for email transmission:

```
[root@goku ~]# openssl s_client -connect TP075164.rockyserver.org:587 -starttls smtp
```

Review the output to ensure the SSL/TLS handshake completes successfully.

Results from the testing

```
[root@goku ~]# openssl s_client -connect goku.rockyserver.org:587 -starttls smtp
CONNECTED(0x0000003)
depth=0 C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
verify error:num=18:self-signed certificate
verify return:1
depth=0 C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
verify return:1
-----
Certificate chain
  0 s:C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
    1:i:C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
      a:KEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
      v:NotBefore: Sep 17 08:33:51 2024 GMT; NotAfter: Sep 17 08:33:51 2025 GMT
-----
Server certificate
-----BEGIN CERTIFICATE-----
MIIDTCARuUCFCDIg00z3t1nv1kGqeaU005xUMA0GCSqGSTb3DQEBCwJAMIGi
MQswCQYDVQQGEwNWTEVBMGA1UEAwMS3NbGEgTHVtcHvYMRUwEwDVQQHDAxL
diwSYSBMdwIwdXKxDKbgVBAoMAoFOVTELMAKgA1UECwvCQ1MxHTAbgnVBBAM
Fgdva3Ucm9ja3lZZX2ZXub3JnMSwQYKoZThvchNAQkBfHrYXnc025hGdv
a3Ucm9ja3lZZX2ZXub3JnB4XD10MDkxNzA4M2M1MVoXDT1IMDkxNzA4M2M1
MVowga1xczAJBgVBAyTAKLZMRUwEwDVQQUADxLwFysBMDwIwdXifXTBtgNV
BACDEt1WxhiExibXb1cjemMAOGA1UECgwQBVBMQswCQYDVQQLDAJDUZedB8g
A1UEAwu29rd5yb2NreXlcn2lc5vcmcXKzApBgkghk1G9w0BCQEWHGtch2hZ
bmFAZ29rd5yb2NreXlcn2lc5vcmcXggE1MA0GCSqGSIb3DQEBAQAA14BDwAw
ggEKAoIBAQCT77FTKMe775Se6pJHFqn+blMavSMGRt+Wwpua38dKcsbV86dFgu
0pPjeZnpPwW/1Szllk0m2jMAAA2G5nWkocrlP/eq0UK6FWu6sx35peIDYw+gfHzE7
nld8vdMr9rBtBbbqmTlqsuxfJ/Qa74Rah41sJmIj+ZgPRkoJbP9j6MRq2n1Bh
rtcs8g2EM9r+0od0ls9sqz50XALQSHPg93RcylpPmHWozxND81c8zogzcwcqIm
yR9yT1IG2aQN1NDT3rnFTWPAnhBR8UDmMuu8q21svx7oxdBhodhAk5RCW9
kg10grDCnVn1x59XjV956rD0zKSbfz0AgMBAAEwDQYJKoZIhvcNAQELBQA0DggEB
A3Xt7b0nfs/au/p4RV2RSFVpj1ZSM0bn8pNleooHZ6ltkU00okg5vrcrJasvK
BEI+SLj0J0NP+VOIt8QH0Gkwxe2Fz4UF43Nvaln6mK57EcacnhIeSuA85B4K
ZDhGMp77yAoC1cfuk01k0mgCNjNMMaN1xIAiywhtpt0TEK2+bLWXObg7qyBlf3a
gSwylqw0386RG/hrstK2+Gnpp7mfse4ZactasRDKhCUvsytoyXSJWuBkjAE/q
SmqkWJz0bjunaf7v91Fb152Aw@o7DNms4JRWm7xZZkyR36VLy0GyN9bHtiDTwadw
3rs19A060734x9jEkke7-
-----END CERTIFICATE-----
subject=C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
subject=C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
issuer=C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
-----
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
-----
SSL handshake has read 1788 bytes and written 437 bytes
Verification error: self-signed certificate
-----
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 18 (self-signed certificate)
-----
250 CHUNKING
-----
Post-Handshake New Session Ticket arrived:
SSL-Session:
  Protocol : TLSv1.3
  Cipher  : TLS_AES_256_GCM_SHA384
  Session-ID: D0A9D0B46FFB193C139FEA5A8E021BFFFDBB87FAA57845E7F2AC55E3D53725F8
  Session-ID-ctx: Resumption PSK: D055D35642C2837CDDACF432D3D96CBFC7B00F218D388A0242C353E52BFCF6353F40D1771A3008C3F8AF930759CCCC51
  PSK identity: None
  PSK identity hint: None
  SRP username: None
  TLS session ticket lifetime hint: 7200 (seconds)
  TLS session ticket:
  0000 - 10 b0 e9 e5 29 d0 ac 6d-29 a1 22 c1 6b 4b f0 03  ....).,kk.,.
  0010 - 53 12 e2 24 87 a9 88 22-02 ee 7d 01 9f f9 ec a1  S,$..,"}.....
  0020 - 01 c6 79 5d a3 52 4b cd-a6 51 fa 23 3f 90 36 fb  ..y].RK.Q.?.6.
  0030 - 5c 20 20 6c 4b 6c-01 00 01 00 00 7d-7b-7c 2-6-3-4-5-6-
```

```

0000 - 10 b0 e9 e5 29 d0 ac 6d-29 a1 22 c1 6b 4b f0 03 ....) ..m)." .kK..
0010 - 53 12 e2 24 87 a9 88 22-02 ee 7d 01 9f f9 ec a1 S..$....").....
0020 - 01 c6 79 5d a3 52 4b cd-a6 51 fa a2 3f 90 36 fb ..y].RK..Q..7.6.
0030 - 6a 82 20 fc da 66 e1 90-04 6c 91 62 69 73 c7 b7 j...f...l.bis..
0040 - 9a d9 4a da 29 7c 85 4c-f8 c2 06 ee c8 44 d3 e1 ..N.)|..L.....D..
0050 - 39 56 55 64 88 98 38 36-54 48 b2 26 4b 9a e4 a5 9VUD..86TH.&K...
0060 - 23 1f lc 6a 89 11 98 9b-85 c6 38 6c b4 44 74 09 #..j.....8L.Dt.
0070 - 77 b9 bc a8 c6 b6 a5 3d-56 c7 d9 ef 74 a7 73 65 w.....=v...t.se
0080 - 2d 2a 24 2f 5f cc cc f2-9b b8 70 87 c6 ff ae ea -$/_....p...
0090 - 2b 74 53 c3 99 df 88 8e-45 8c ce 78 0c 4e b7 a0 +tS....E..X.N..
00a0 - 78 c5 6b 39 13 1a d2 62-a7 c9 4e 1f 5c 77 1b 83 x.k9...b..N.\w..
00b0 - e9 d5 b4 53 18 9a a1 39-24 de ac f6 ac 97 fd e1 ...S....S9.....
00c0 - 05 5c c8 5c 50 76 cd f5-f3 81 5c cb a2 4e b7 5c .\.\Pv....\..N.\

Start Time: 1726565406
Timeout : 7200 (sec)
Verify return code: 18 (self-signed certificate)
Extended master secret: no
Max Early Data: 0
---
read R BLOCK
421 4.4.2 goku.rockyserver.org Error: timeout exceeded
closed
[root@goku ~]# 

```

Test On Ubuntu Client:

- Use `openssl s_client` to verify the certificate:

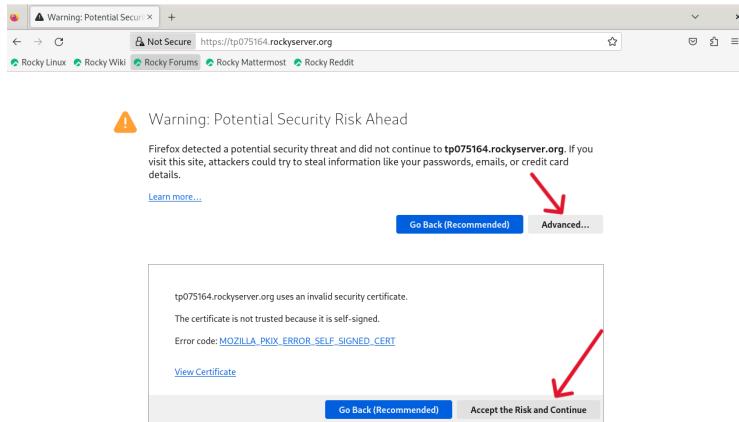
```
kashubuntu@vegeta:~/Desktop$ openssl s_client -connect TP075164.rockyserver.org:587 -starttls smtp
CONNECTED(00000003)
depth=0 C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
verify error:num=18:self-signed certificate
verify return:1
depth=0 C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
verify return:1
...
Certificate chain
0 s:C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
    i:C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
        a:KEY; rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
        v:NotBefore: Sep 17 08:33:51 2024 GMT; NotAfter: Sep 17 08:33:51 2025 GMT
...
Server certificate
-----BEGIN CERTIFICATE-----
MIIDzTCARUCFCCDjgLoQz3ta1nvkCqo@U005xUMA0GCSqGS1b3DQEBCwUAMIGL
M0swcNQYDVQGEWJNMITEVBMGA1UECAwMS3vhDgEgTHvtChVvMrUwEwVDVQOHdAx1
dIuFSySBMd1wdX1KxDAXKBnVBAoMA0FOVTLEMAGA1UECwxC01MxHTAbBgNBAMM
Fdgv3UucmJa3lZzX122XX1ub31nMsSwkQjJKoZT1hvcNaQkBFlxxYNoc25hQcdv
a3UicmJa3lZzX122XX1ub31nB4XTD10MdkxzIz44H21MVoxDT1MDKnxZ4A2MzM1
M沃ga1xZaJbgNBAYTAk1ZMRUlwewDVQ0TD1Ax1dwFsYSBMDw1wdxIxFTATBgnV
BachDEt1NxhiEx1bX81JEMMa0GA1UECp9DQVBWQsWQDQQLADJU2EdMBsG
A1UAvwIlZ29rds5yb2reXnlcn2lciSvcmcxzApBqkjhkLG9wBQCEMHGthc2hz
bnFAZ29rds5yb2reXnlcn2lciSvcmcwqgElMA0GCSqGS1b3DQEBAQUAIAIBDwAw
ggEKAoIBAQCT77FIKMe775seopJHMqrn+BuIaySWIGRt+wWOpua4Bdkc5cb/86DFGu
SPPeZ+pwM/1SzLk0e3JAAA2GSmnk0e1P/e0lJK6fVu6s.x3Pe1DWw+gfHzE7
ndvdwOr9r1tEbbqmkT1qsoxfL/9a74rh41s3n1j-ZgPRk0a2p916RqZn18hs
rTcs8g2EM3r+E0d01s9q7ss0XALQShp9JRCyxljpMMH0zzX08V1c8ZoQczwceqln
yR9vV1ITGzAQN10DT3x1rFHTWPnAnnM8B0DmMu8qZlsvx7oxd2hodhAk5RCW9
kg10gjDCnVnIX59XVjC956rD02kSBf0ZagjBAAEwQjJKoZ1hvcmNAELBQAdggEB
AJXt7bonfs/.au/p4RV2RSFVpj1Z3HDbnqANL0eo0Z1tkU0po0kq5vvrcRjasVK
BEI-sLjO3DNp+VoIt8Qh0GokwxoeFzufJ3NmvaLn0mKS7Eccahziesuax8584K
ZDhcMp7yao1cfcfuk01koqkq5vrrcRjasVK
gsw1q03808RC/nF5TZK2+hnnpp?/fSe42actasM0DRKLU5vtoyxSJsu008Kjaet/q
Sn0kw1zbhjunaF7/y91Fb152AwRoZT0Ns4JRNh7Z7kr36vLYOGYn9bhHtLDTwadw
3rs19A0CD7J4x9jEkke3w3d8=
-----END CERTIFICATE-----
subject=C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
issuer=C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
...
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
...
SSL handshake has read 1788 bytes and written 439 bytes
Verification error: self-signed certificate
...
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 18 (self-signed certificate)
...
250 CHUNKING
...
Post-Handshake New Session Ticket arrived:
SSL-Session:
Protocol : TLSv1.3
Cipher  : TLS_AES_256_GCM_SHA384
Session-ID: 31A0D13881D376C44F794655DCB88665A7BAF5B20804EC1B95AEC6998A200EC
Session-ID-ctx:
Resumption PSK: 713F646F46A7DB47A9AB4684660D81B06A31006B6372DDE73BF4018A4ABB88555C94D25E3CBF1ECDDED36CBD2BD6CB32
PSK Identity: None
PSK Identity hint: None
SRP username: None
TLS session ticket lifetime hint: 7200 (seconds)
TLS session ticket:
0000 - b2 a2 40 d7 he f5 08 29-ef a1 b5 87 d7 8c 98 87 + .0C ...
0000 - b2 a2 40 d7 be f5 08 29-ef a1 b5 87 d7 8c 98 87 + .@G,...)..... .
0010 - 32 73 c9 42 43 09 59 6b-5d eb 70 4d ac 4c 20 F5 +.8C.YK].pM.L .
0020 - 9f c1 41 d0 eb 79 73 81-98 b4 a1 17 3f 98 11 ce ..A..ys.....?...
0030 - da 9c d7 80 9f cf ef 7f-7a 87 91 11 6e 75 f2 ed .....z...nu..
0040 - 1b 05 49 99 c5 7b c1 8a-7d a8 18 7e c2 44 24 48 ..I[...].~.D*@
0050 - 25 c6 06 83 6f be 33 94-b6 a8 62 3b e6 a2 49 14 %..o..3..b..I.
0060 - ed f5 cd 81 fb eg 61 8c-1f bf 0e b6 7c b1 78 74 .....a.....|.xt
0070 - 6c 66 d7 7f 02 70 8f a0-82 8c 10 8e 27 de e6 9c ln...p...,'...
0080 - ba 9e f3 7b 44 56 83 30-9c e7 56 21 96 ac f9 73 ...[Dv..0..Vi...s
0090 - 0b a0 d8 21 32 e8 44-5f b4 58 20 43 5d 7b p2 k..12..M...X C].
00a0 - 7b 9a e6 44 07 10 4d 94-88 b8 17 2a 24 2d cb 77 {..0..M...*$-.w
00b0 - 2b 63 22 54 5c 65 bf 52-55 b8 8c c1 c3 84 07 e1 +c"Te.RU.....
00c0 - e0 92 9a 39 17 a9 73 f9-9c 8d 57 d1 e7 7f 68 74 ...9.s...W...ht

Start Time: 1726568622
Timeout : 7200 (sec)
Verify return code: 18 (self-signed certificate)
Extended master secret: no
Max Early Data: 0
...
read R BLOCK
^C
kashubuntu@vegeta:~/Desktop$
```

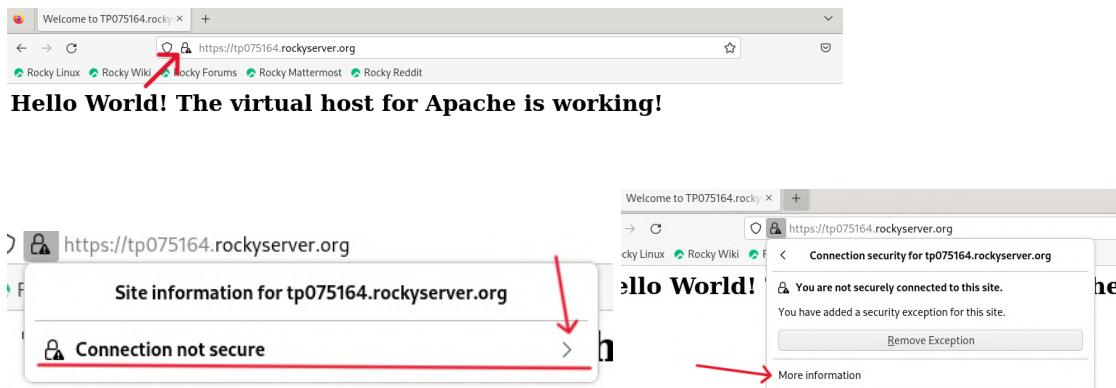
Testing Apache SSL/TLS:

- **On Rocky Server:**

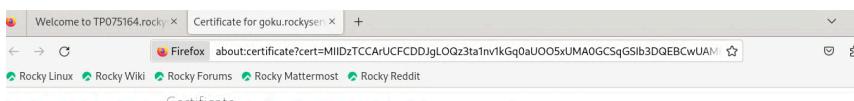
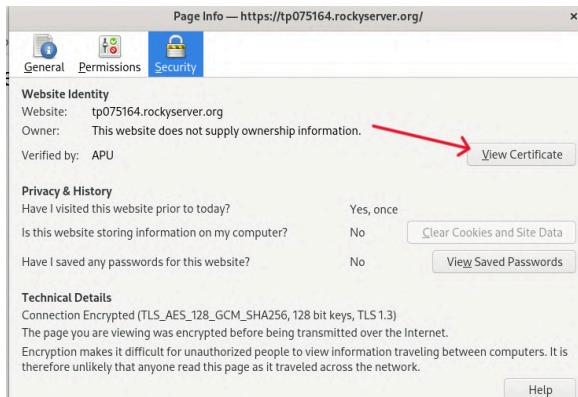
- Open a web browser and navigate to your website using HTTPS:
<https://servertp076161.sna.tp076161.com>.



Click on the padlock icon in the browser's address bar and select “**Connection Secure**”, then click “**More Information**”.



In the “**Security**” tab, click “**View Certificate**” to see the certificate details.



Certificate

goku.rockyserver.org

Subject Name	
Country	MY
State/Province	Kuala Lumpur
Locality	Kuala Lumpur
Organization	APU
Organizational Unit	CS
Common Name	goku.rockyserver.org
Email Address	kashsna@goku.rockyserver.org

Issuer Name	
Country	MY
State/Province	Kuala Lumpur
Locality	Kuala Lumpur
Organization	APU
Organizational Unit	CS
Common Name	goku.rockyserver.org
Email Address	kashsna@goku.rockyserver.org

Validity	
Not Before	Tue, 17 Sep 2024 08:33:51 GMT
Not After	Wed, 17 Sep 2025 08:33:51 GMT

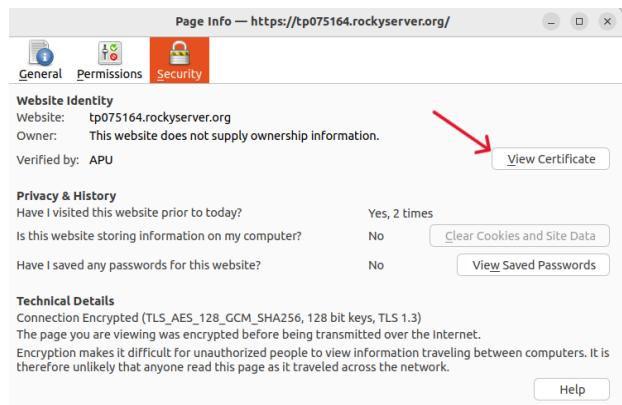
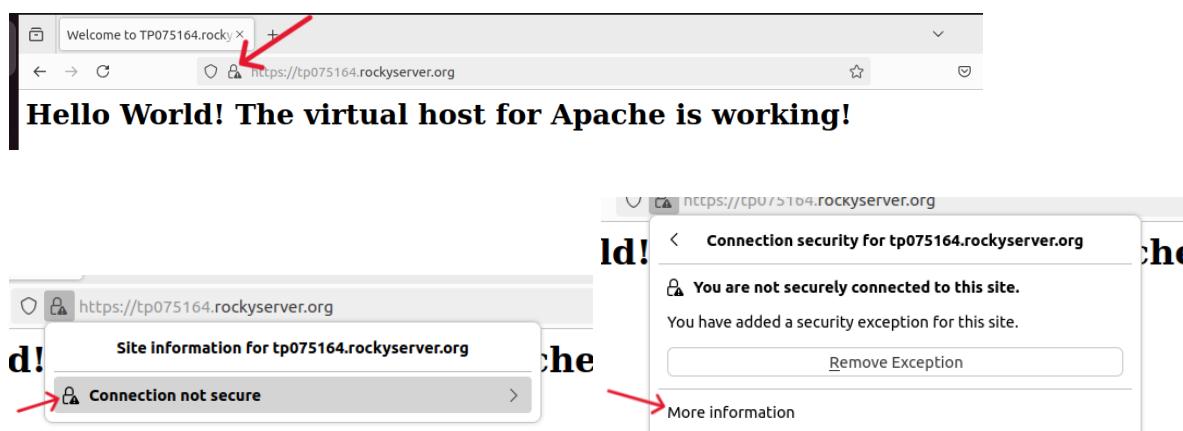
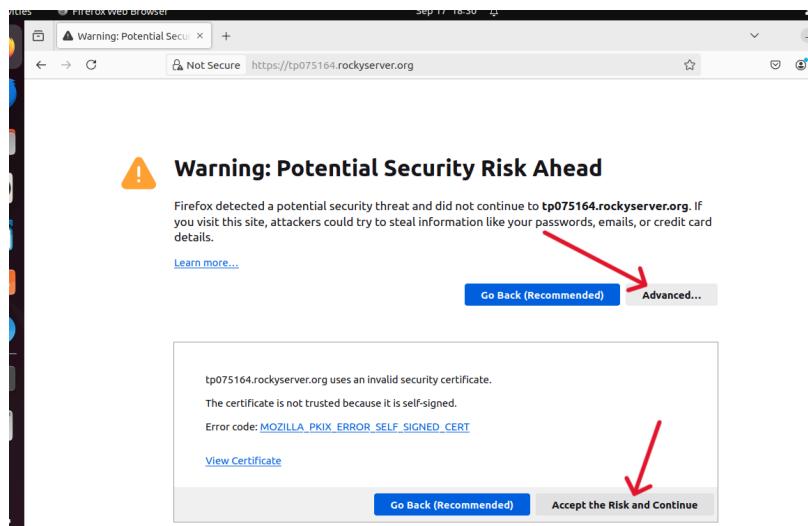
Public Key Info	
Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	BB:EC:52:0A:34:C7:BB:4F:9E:5E:EA:92:47:16:A9:FE:6D:43:1A:BD:23:06:46:D...

Miscellaneous	
Serial Number	20:C3:26:02:CE:43:3D:ED:6B:59:EF:D6:41:AA:D1:A5:0E:3B:9C:54
Signature Algorithm	SHA-256 with RSA Encryption
Version	1
Download	PEM (cert) PEM (chain)

Fingerprints	
SHA-256	05:D2:7E:0C:50:5F:B9:98:37:DE:26:91:8C:D9:00:56:92:6D:13:99:3F:0F:4D:25...
SHA-1	47:23:7B:D8:4A:74:99:55:08:2D:53:E7:BB:68:CF:FC:D1:22:44:F2

Ubuntu Client:

- Follow the same steps as above to view the SSL/TLS certificate in the browser.



Firefox about:certificate?cert=MIIDzTCCArUCFCDDJgLOQz3ta1nv1kGq0aUOO5xUMA0GCSqGSib3D ☆

Certificate

goku.rockyserver.org

Subject Name

Country	MY
State/Province	Kuala Lumpur
Locality	Kuala Lumpur
Organization	APU
Organizational Unit	CS
Common Name	goku.rockyserver.org
Email Address	kashsna@goku.rockyserver.org

Issuer Name

Country	MY
State/Province	Kuala Lumpur
Locality	Kuala Lumpur
Organization	APU
Organizational Unit	CS
Common Name	goku.rockyserver.org
Email Address	kashsna@goku.rockyserver.org

Validity

Email Address: kashsna@goku.rockyserver.org

Validity

Not Before: Tue, 17 Sep 2024 08:33:51 GMT
Not After: Wed, 17 Sep 2025 08:33:51 GMT

Public Key Info

Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	BB:EC:52:0A:34:C7:BB:4F:9E:5E:EA:92:47:16:A9:FE:6D:43:1A:BD:23:06:46:DF...

Miscellaneous

Serial Number	20:C3:26:02:CE:43:3D:ED:6B:59:EF:D6:41:AA:D1:A5:0E:3B:9C:54
Signature Algorithm	SHA-256 with RSA Encryption
Version	1
Download	PEM (cert) PEM (chain)

Command-Line Testing:

- **On Rocky Server:**
 - Verify the web server's SSL/TLS setup with:

```
openssl s_client -connect servertp076161.sna.tp076161.com:443
```

```
[root@goku ~]# openssl s_client -connect TP075164.rockyserver.org:443
CONNECTED(0x00000003)
depth=0 C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
verify error:num=18:self-signed certificate
verify return:1
depth=0 C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
verify return:1
---
Certificate chain
  0 s:C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
    1 :C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
      a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
      v:NotBefore: Sep 17 08:33:51 2024 GMT; NotAfter: Sep 17 08:33:51 2025 GMT
  ...
Server certificate
-----BEGIN CERTIFICATE-----
MIIDzTCARUCFCDDJgLoQz3taInv1kGq0aU005xUMA0GCSqGSIB3DQEBCwUAMIGi
M0swQYDVQGEwJNWTTEVMBMGIAUECwMs3VhBgEgTHVtcHVyMRUwEwYDVQQHDAxL
dwFySBSMdWlwdxIxDDAKBgNBVAoM0FQVTELMAKgA1UECwQzC1MxHTAbBgNVBAM
Fdova3ucm9ja3lZKJ2ZXIub3JnMSswkQYJKoZIhvncNAQBFhxrvXNo25hQGdv
a3uUcm9ja3lZKJ2ZXIub3JnMB4XDTI0MDkxIz4AMzMLMVoXTDTI1MDkxNzA4MzI
MV0wgaIxCzaJ8gNBAYATk1ZMRUwEwYDVQGIDAxlwdFwSYSBMdWlwdxIxFTATBgNV
DwEDEtVxHIEixbxBlc1EMMAoGA1UECgwQDVBFM0wQzCQYDVQGQDAAJDUzEdMBsG
ALUEAwuU29rds5yb2NrExXlnlCz1c5vcmcxxzApBgkqhkiGw0BQCWEHgthc2hz
ggFAZ29rds5yb2NrExXlnlCz1c5vcmcwgE1MA0GCSqGSIB3DQEBAQUAIAIBdwAw
ggEKAo0BACQ77FIKNMe775Se6pJHFgn+biUavSMGRt+WwQpuas4BdKcsbV86DFGu
6pPjeZnfwWlIsZllk0m3jMAAA25nWkoCrIP/eoUK6FWu6osx35peIDwyrghTzE7
6pvdwM0r9BTBEBBQmkTlsqsfL/Qa74Rah4isMjI+ZgPRkoJBp9j6MRq2n1Bhs
rTcS&g2EM3r+EOd01s9qZs50XALQSH9PjzCqxy1pPmWozxDN8V1c8ZoqzcwqcLm
y99yvtiIGzAnQN1D73xrFMTPmaAnnBMr8UDmMuuaq2isvx70xdBhodhAk5RCWw9
kgi6grCnVn1X59XVjC956'D02zSBFozAgMBAAEwD0YJKoZIhvncNAQELBQAdggEB
A3jxt7bonfs/au/p4RV2RSRFVpj iZSMDBn8pNLeeohZ6ltKUooookq5vvrcRjsAvK
BEIiSL0J030NP+VOitbQGh0Gkw6e2F4UF43Nwla1n6mK57ECCahzIeSuax85B4K
ZDhgMp77yAocICfu0k1k6MgCNJNMaiNx18tAywhTpToTEK2+bLB8WxDbgTqyBlf3a
gSwlylQW0386RG/hrStZK2+Gnnp7mFsE4ZactasmRDkHCUSvtoyXSJWu0BkjAe/q
SmqkWz2objunaF7v91Fb152Aw867DNMs4RjWm7xZkyR36vLY0Gyn9bHtiDTwaDw
3rs19a0GD734x9jEkke3wd#
-----END CERTIFICATE-----
subject=C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
3rs19a0GD734x9jEkke3wd#
-----END CERTIFICATE-----
subject=C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
issuer=C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 1537 bytes and written 408 bytes
Verification error: self-signed certificate
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 18 (self-signed certificate)
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
  Protocol : TLSv1.3
  Cipher   : TLS_AES_256_GCM_SHA384
  Session-ID: A8F4486DFABC45D048336D2B99A2E621445E99F3AEB88B68AF336539C95FA1A7
  Session-ID-ctx:
  Resumption PSK: BCAC420B73E5DD7D87D25C3041985D9B15B127D7EA8563E5D74CDC4D9730428EB41DCF64C9336AD7CF9E6B80365FE0F
  PSK identity: None
  PSK identity hint: None
  SRP username: None
  TLS session ticket lifetime hint: 300 (seconds)
  TLS session ticket:
  0000 - e1 fd 01 c1 59 82 20 40-9a bc 36 87 ab 61 94 70  ...Y. .6..a.p
  0010 - dc 3d c8 5a 19 e3 f2 79-fb 8d d7 82 51 9e 35 ca  ..^. /y....Q.5.
  0020 - d8 b9 c4 e8 a7 4b 3b 9f-ec 6b 64 14 65 31 ff 93  ....K;.lkde1..
  0030 - a6 1f 27 bb bd 00 7e 4c-4b bf 8e 28 29 e9 c7  ' _ x K /

```

```

0040 - 0b 92 9c 88 99 d9 fd 37-e3 a4 51 03 6a eb 7d 17 .....7..Q.j.}.
0050 - 1a b3 f4 5e 67 a3 fe 42-49 3b fe f3 c8 f4 b0 56 ...^g..BI;....V
0060 - f8 00 54 20 d7 f6 e9 49-6d b5 55 ef f3 00 da 99 ..T ...Im.U. ....
0070 - c9 dc 30 ca 11 53 9a f0-64 a0 95 bd 12 62 e5 30 ..0..S..d....b.0
0080 - 00 f2 91 58 41 17 23 74-68 8a 52 ef 56 7f 85 a2 ...XA.#th.R.V...
0090 - 50 2f a2 3e f8 b8 a1 0f-81 53 99 46 3f 2f 69 e7 P/,>.....S.F?/i.
00a0 - 8b f8 b4 d1 e2 9c 47 a0-43 0f dd 5b 98 01 88 6d .....G.C..[...m
00b0 - 0c 50 89 b8 e9 e3 53 74-95 e8 a8 37 03 bf 7f e4 .P....St....7....
00c0 - 9e 38 5a 9e c3 2b 90 a7-1d 3f 6d 04 53 07 90 73 .8Z..+...?m.S..s
00d0 - e7 0a 4e 8f 8b 43 f3 86-c2 e8 c5 11 e1 d3 c1 ac ..N..C..... .
00e0 - 08 4d 46 af 54 eb a0 7d-60 ad 29 48 4c 16 78 6a .MF.T..)`.)HL.xj
00f0 - 8d 07 8a f2 c2 93 d2 ba-32 98 4e 34 47 8b 95 04 .....2.N4G...
Start Time: 1726569151
Timeout : 7200 (sec)
Verify return code: 18 (self-signed certificate)
Extended master secret: no
Max Early Data: 0
---
read R BLOCK
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
Protocol : TLSv1.3
Cipher : TLS_AES_256_GCM_SHA384
Session-ID: 0E934836B85E80D508CD798B7F9B1FEA21E086C662B8BF337D8754C747256DA0
Session-ID-ctx:
Resumption PSK: B3583600F7CAA0AF3D189F2FEA60F18F46BF647424E68084746AA8A1395C1A9A0509E090EEABD556FBF6D6D5196BDCA6
PSK identity: None
PSK identity hint: None
SRP username: None
TLS session ticket lifetime hint: 300 (seconds)
TLS session ticket:
0000 - e1 fd 01 c1 59 82 20 40-9a bc 36 87 ab 61 94 70 .....Y. @..6..a.p
0010 - e9 74 2a 87 a0 4a 20 ec-e7 ba 8d 4a 69 2f 24 61 .t*..J ....Ji/$a
0020 - e0 d3 cc ca e0 27 52 ca-90 f9 94 f2 c7 86 18 1d .....'R. .....
0030 - 08 7e 73 60 4b 52 a3 1c-80 29 be d1 8d 8d ab 78 .~s'KR...)....x
0040 - 09 72 46 0c cb 64 5d 72-65 57 ba 74 e3 1a 0b 80 .rF..d]reW.t. ....
0050 - 98 3f 69 4e 90 fc 0f ba-29 b3 b0 21 9e 8b 93 c8 .?IN....)!.....
0060 - 39 c5 18 df h1 1f 4b db-9f 0f f1 13 5a cf ef 3c 9.....K.....Z..<
0066 - 39 c5 18 df b1 1f 4b db-9f 0f f1 13 5a cf ef 3c 9.....K.....Z..<
0070 - 83 f8 6c 2f d0 e9 f0 25-6b 6c af 8e 7e b7 06 ae ..l/...%kl.,~...
0080 - 72 a3 a7 06 a3 28 c6 40-87 33 4a 7f 86 25 05 9f r....(@.3J.%..
0090 - e5 1c 9c ba c0 61 b0 4c-00 f1 c9 00 94 fb d7 15 .....a.L. .....
00a0 - 7f 83 18 95 c8 3c c6 7a-87 26 a6 96 4e 00 bb 04 .....<.z.&..N...
00b0 - a1 49 a5 22 55 23 34 8c-77 04 e4 3b d1 f3 80 b8 .I."U#4.w..j. ....
00c0 - 59 f7 1d 6f 38 e9 lc 82-4e 16 7e 46 e5 d4 d7 c4 Y..o8...N..~F. ....
00d0 - df d7 ad ba fe 04 b0 09-44 44 c8 d4 02 38 f9 ...j....DDC..8.
00e0 - eb 5e 6e 8f le 91 44 61-a1 f0 04 ac 85 ec 42 d4 .^n...Da.....B.
00f0 - f1 51 19 fd 77 6d 29 4a-32 51 35 14 b0 35 fa e6 .Q..wm)J2Q5..5..
Start Time: 1726569151
Timeout : 7200 (sec)
Verify return code: 18 (self-signed certificate)
Extended master secret: no
Max Early Data: 0
---
read R BLOCK
C
root@goku ~]# 
```

On Ubuntu Client:

- Use the same `openssl` command to test SSL/TLS:

```
openssl s_client -connect servertp076161.sna.tp076161.com:443
```

```
kashubuntu@vegeta:~/Desktop$ openssl s_client -connect TP075164.rockyserver.org:443
CONNECTED(00000003)
depth=0 C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
verify error:num=18:self-signed certificate
verify return:1
depth=0 C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
verify return:1
...
Certificate chain
  0 s:C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
    i:C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
      a:KEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
      v:NotBefore: Sep 17 08:33:51 2024 GMT; NotAfter: Sep 17 08:33:51 2025 GMT
...
Server certificate
-----BEGIN CERTIFICATE-----
MIIDzTCCArUCFCDDJgLOQz3tainvkGq0aU005xUMA0CCSgGS1b3DQEBCwUAMIGi
MQSwCQYDVQGWEJNNTEVMBMGA1UECAwMS3vhbGEgHTVtcHvMRUweWYDVOQHQDAxL
dWfSYSBMdW1wdXixDDAKBgNVBAoMA0FQVTELMAKGA1UECwCQ1MxHTAbBgNVBAMM
FGdva3Uucm9ja3lZJZKJZXIub3JnMSkwQJKoZIhvcNAQkBFhxryXNoc25hQdva
a3Uucm9ja3lZJZKJZXIub3JnMB4XTD10MDkxNzA4MzM1MVoxDTI1MDkxNzA4MzM1
MVowgaIxczAJBgNVBAYTAKIZMRUwEwYDVQQIDaxLdwFYSBMDw1wdXiFTATBqNV
BACMDet1YWhxIE1xb81cjEMMAoGA1UECgwDQVBVMQswCQYDVQQLDAJDUzEdMBsG
A1UEAwUZ29rdSSyb2NreXNlcnzLci5vcmcxKzApBgkqhkiG9w0BCQEWHGthc2hz
bmFAZ29rdSSyb2NreXNlcnzLci5vcmcwggiMA0GCsG1b3DQEBAQUAA1BDwAw
ggEKAoIBAQCC77FKNMe7T55e6pJHFqn+buMavSMGRt+lwQpuas4BdKcsbV86DFGu
6PjeZ+nPwW/isZllK0m3jMAAA2G5nwkoCrIP/eoUK6FWu0sx35peIDYw+gfHzE7
nld8vdmOr9BtbEqmk1qsxuFL/qat74rah41sJmlj+zgPRkojbP9j6Mrq2n1Bhs
rTC8g2EM3r+E0d01s9qZs50XALQSHgP93RcyxlppMhMozxND8V1c8Zoqczwqlm
yR9yVtIIGzaQNQ1DT3xrxFNTWPmAAnBMR8UDMuu8q2isvx7oxdBhodhAk5RCW9
kgIgrDCnVn1X59XjC956rD02K8bf0zAgMBAAEwQYJKoZIhvcNAQELBQA0dgEB
AJxt7b0fs/au+p4RV2RSFVpjizSMDbn8pNleohZ61tku0ookqj5vrrCrjasvK
BEI+SLJOJ0NP+VOIt8QGhOGkwx6e2F4zU43NwvaIn6mKS7EccahzIeSuAx85B4K
ZDHGmp77yAocIfcuK01k6McCNJNMMaNIx8IAywhtPtoTEK2+BL8WDbgTqyBLf3a
gSkylqwo386RG/hrStZK2+Gnpp7fFsE4ZactasmRDKHCUsvtoyXSJWu0Bkjaf/q
SmqkWJz0bjunaF7v91Fb152Aw8o7DNmS4JRWm7ZZkyR36vLY0GyN9bHtiDTwaDw
...
gSwlyLqwo386RG/hrStZK2+Gnpp7fFsE4ZactasmRDKHCUsvtoyXSJWu0Bkjaf/q
SmqkWJz0bjunaF7v91Fb152Aw8o7DNmS4JRWm7ZZkyR36vLY0GyN9bHtiDTwaDw
3rs19A0GD7J4x9jEkke3wd8=
-----END CERTIFICATE-----
subject=C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
issuer=C = MY, ST = Kuala Lumpur, L = Kuala Lumpur, O = APU, OU = CS, CN = goku.rockyserver.org, emailAddress = kashsna@goku.rockyserver.org
...
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
...
SSL handshake has read 1537 bytes and written 406 bytes
Verification error: self-signed certificate
...
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 18 (self-signed certificate)
...
...
Post-Handshake New Session Ticket arrived:
SSL-Session:
  Protocol : TLSv1.3
  Cipher   : TLS_AES_256_GCM_SHA384
  Session-ID: 9F4EB01AB294F428FAD02DA5BBDFC472625F973AE283435CF35B003776BD4A65
  Session-ID-ctx:
  Resumption PSK: 688143277E44C2D1CF20F987C0789334360A68E3825A852C426684736274C6595A5F41E044632A6A6799B68554AF88B0
  PSK identity: None
  PSK identity hint: None
  SRP username: None
  TLS session ticket lifetime hint: 300 (seconds)
```

```

SRP username: None
TLS session ticket lifetime hint: 300 (seconds)
TLS session ticket:
0000 - e1 fd 01 c1 59 82 20 40-9a bc 36 87 ab 61 94 70 ....Y. @..6..a.p
0010 - ca 0c a6 41 84 9b 33 36-ea da 97 de 9d ce d4 1c ...A..36.....
0020 - 92 e5 a7 a1 ac 16 90 83-20 44 d2 24 0c 45 f8 91 ..... D.$.E..
0030 - 5d 95 52 99 f7 a7 d5 04-62 07 10 8d 6d 6f 43 e7 ].R....b..moc.
0040 - 78 75 7a 17 4d e5 96 d5-fd 11 a7 c9 c9 24 aa KASHAVE SATHYELL A/L SATHYVELL (TP075164@mail.apu.edu.my) is signed in
0050 - 38 db e2 00 e9 8f 16 86-4d 49 23 39 b1 2c ea ..z...v...q...p...
0060 - 33 37 ac 0e f5 40 67 0d-4f f2 0c 22 2a 7b f8 f1 37...@g..0.. "*{..
0070 - d7 19 34 42 21 ff 39 46-1b 46 9a 73 26 39 6e 56 ..4B!..9F.F.s&9nV
0080 - 5d 17 97 8b 64 95 8f 6f-d3 9f ec bf 00 39 d3 83 ]....d..o.....9..
0090 - a6 fa e4 7c 39 3c 99 72-8c 0f 78 9f 7b 14 b8 6a ...|9<.r..x.{..j
00a0 - e6 08 ec 78 a4 0a d9 15-4b 9c 99 d6 46 3b 0f ba ...x....K...F;..
00b0 - 03 ae 6a be f5 a8 7e 73-7a 93 f9 1a 95 f5 1e 53 ..j....~sz.....S
00c0 - 32 1e 15 e6 3a c6 fd e4-c3 33 5f d0 21 6c 67 30 2.....3_.!lg0
00d0 - cc 27 cf 29 d7 37 b8 43-77 b1 4b cd e9 70 45 d5 .').).7.Cw.K..pE.
00e0 - 7e 27 08 6d 80 04 45 22-1b e9 9e 0f 61 61 26 a4 -.'.m..E"....aa&.
00f0 - f8 2d 63 9c 63 7f 3f fa-0a 6d 4c 1b 99 3a 4d e1 ..c.c.?..mL..iM.

Start Time: 1726569506
Timeout : 7200 (sec)
Verify return code: 18 (self-signed certificate)
Extended master secret: no
Max Early Data: 0
---
read R BLOCK
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
  Protocol : TLSv1.3
  Cipher   : TLS_AES_256_GCM_SHA384
  Session-ID: 2F3067C2DA9EBE5F3511FFD6B9C1B4F503D8C40C76F0EB8FDD79DCC15AE3CF87
  Session-ID-ctx:
  Resumption PSK: D74FB073E27D783873696E6C61862768E10D41E3BECBA9691E7FE59E96B138A6A029F1E537C8F8C182F767BF9B84CDB5
  PSK identity: None
  PSK identity hint: None
  PSK ...
PSK identity: None
PSK identity hint: None
SRP username: None
TLS session ticket lifetime hint: 300 (seconds)
TLS session ticket:
0000 - e1 fd 01 c1 59 82 20 40-9a bc 36 87 ab 61 94 70 ....Y. @..6..a.p
0010 - f5 eb 8d b9 cc df 22 94-b1 bd f1 27 1c 48 e5 9e ....."....'H..
0020 - ea b0 7a 30 54 bb 56 a1-71 50 96 58 45 a1 56 69 ..z0T.V.qP.XE.Vl
0030 - 83 30 2b f6 21 3d 7e 4b-e0 a8 1a 1b f5 fd 06 3a .0+.!=~K.....;
0040 - 42 54 c6 17 36 9a 21 51-56 92 c8 96 72 6d 0c 80 BT..6.!QV..rm..
0050 - eb b5 5d d0 a5 50 45 de-4a 5e e1 bf ec ff 96 8c ..]..PE.J^.....
0060 - f3 96 71 ec 81 ee 23 53-8c 2c fa 15 d4 71 50 ef ..q...#S.,...qP.
0070 - 9c d5 d8 9f 0f 26 7a 63-94 2d 82 e5 47 d3 87 e6 .....&zC.-.G...
0080 - b3 fc 33 b6 eb f8 76 0b-1b 33 24 fc be b6 30 46 ..3...v..3$...0F
0090 - 92 a1 0b f9 0d 82 89 5f-74 85 d6 2e 2b 4b 04 fd ....._t...+K..
00a0 - e1 bf bf df d5 ff 4e 12-3a ef d2 1c da f1 6a a7 .....N.....J.
00b0 - 12 82 ca 31 fa 93 45 c5-f8 31 9e 3f ce 69 84 73 ..1..E..1.?i.s
00c0 - 45 da a2 0c 3b 64 a5 24-c1 71 3a 64 6d 15 e9 67 E...;d.$.q:dm..g
00d0 - 52 38 29 b2 0c e1 58 44-b7 c8 05 73 bc c2 fe 5c R8)...XD...s...\
00e0 - a7 5e dc 51 48 21 11 11-f6 e6 0f 2d d3 6b 01 4f ..^QH!.....k.0
00f0 - 32 47 97 53 45 33 e5 74-a4 f0 be 9e 41 e5 25 cc 2G.SE3.t....A.%.

Start Time: 1726569506
Timeout : 7200 (sec)
Verify return code: 18 (self-signed certificate)
Extended master secret: no
Max Early Data: 0
---
read R BLOCK
^C
kashubuntu@vegeta:~/desktop$ 

```

Troubleshooting SSL/TLS Configuration:

Systemctl restart dovecot failed.

```
[root@goku ~]# systemctl restart dovecot
Job for dovecot.service failed because the control process exited with error code.
See "systemctl status dovecot.service" and "journalctl -xeu dovecot.service" for details.
```

Verify Certificate File Paths

1. Check Certificate File Path:

- Verify that the new certificate file exists and has the correct path:

```
[root@goku ~]# ls -l /etc/pki/tls/certs/server2.crt
-rw-r--r--. 1 root root 1379 Sep 17 16:33 /etc/pki/tls/certs/server2.crt
```

Check Private Key File Path:

- Verify that the new private key file exists and has the correct path:

```
[root@goku ~]# ls -l /etc/pki/tls/private/server2.key
-rw-----. 1 root root 1874 Sep 17 16:29 /etc/pki/tls/private/server2.key
```

Update Dovecot Configuration

1. Open the Dovecot SSL Configuration File:

- Edit the configuration file:

```
[root@goku ~]# sudo nano /etc/dovecot/conf.d/10-ssl.conf
```

Ensure the Correct Paths are Set:

- Update the configuration with the new paths:

```
[root@goku ~]# systemctl restart dovecot
[root@goku ~]#
```

Check Logs:

- Inspect the Dovecot logs for any errors related to SSL/TLS:

```
[root@goku ~]# sudo tail -f /var/log/maillog
Sep 17 12:55:19 goku postfix/submission/smtpd[4185]: warning: TLS library problem: error:10080002:BIO routines::system lib:crypto/bio/bss_file.c:300:
Sep 17 12:55:19 goku postfix/submission/smtpd[4185]: warning: TLS library problem: error:0A080002:SSL routines::system lib::ssl_rsa.c:448:
Sep 17 12:55:19 goku postfix/submission/smtpd[4185]: connect from goku.rockyserver.org[159.69.39.4]
Sep 17 12:55:19 goku postfix/submission/smtpd[4185]: lost connection after STARTTLS from goku.rockyserver.org[159.69.39.4]
Sep 17 12:55:19 goku postfix/cleanup[4189]: 3029D13D5F3: message-id=<20240917045519.3029D13D5F3@goku.rockyserver.org>
Sep 17 12:55:19 goku postfix/submission/smtpd[4185]: disconnect from goku.rockyserver.org[159.69.39.4] ehlo=1 starttls=0/1 commands=1/2
Sep 17 12:55:19 goku postfix/qmgr[4183]: 3029D13D5F3: from=<double-bounce@goku.rockyserver.org>, size=1225, nrcpt=1 (queue active)
Sep 17 12:55:19 goku postfix/local[4191]: 3029D13D5F3: to=<root@rockyserver.org>, orig_to=<postmaster>, relay=local, delays=0.08, delays=0.04/0.01/0/0.03, dsn=5.2.0, status=bounced (cannot update mailbox /var/mail/root for user root, unable to create lock file /var/mail/root.lock: No such file or directory)
Sep 17 12:55:19 goku postfix/bounce[4192]: warning: 3029D13D5F3: undeliverable postmaster notification discarded
Sep 17 12:55:19 goku postfix/qmgr[4183]: 3029D13D5F3: removed
^C
```

Double Check Permissions:

- Ensure the certificate and private key files have the correct permissions:

```
[root@goku ~]# sudo ls -l /etc/pki/tls/certs/server.crt /etc/pki/tls/private/server.key
-rw-r--r--. 1 root root 1375 Sep 17 00:09 /etc/pki/tls/certs/server.crt
-rw-----. 1 root root 1704 Sep 17 00:00 /etc/pki/tls/private/server.key
[root@goku ~]#
```

Check Certificate and Key Match:

- Verify that the certificate and private key match:

```
[root@goku ~]# openssl rsa -noout -modulus -in /etc/pki/tls/private/server.key | openssl md5
MD5(stdin)= bcdc183e801b131b1530a6bb7270e969
[root@goku ~]# openssl x509 -noout -modulus -in /etc/pki/tls/certs/server.crt | openssl md5
MD5(stdin)= bcdc183e801b131b1530a6bb7270e969
[root@goku ~]#
```

The output from both commands are the same which means they match.

Update Postfix Configuration

1. Edit Postfix Configuration:

o Edit /etc/postfix/master.cf

```
GNU nano 5.6.1                               /etc/postfix/master.cf                         Modified
-o smtpd_tls_security_level=encrypt
-o smtpd_sasl_auth_enable=yes
-o smtpd_tls_cert_file=/etc/pki/tls/server.crt
-o smtpd_tls_key_file=/etc/pki/tls/private/server.key
-o smtpd_tls_auth_only=yes
# -o smtpd_reject_unlisted_recipient=no
# -o smtpd_client_restrictions=$mua_client_restrictions
```

Previously I had not added the ‘private’ in the private key path in the postfix/master.cf file.

```
GNU nano 5.6.1                               /etc/postfix/master.cf                         Modified
smtp      inet  n      -      n      -      -      smtpd
#smtp     inet  n      -      n      -      1      postscreen
#smtpd    pass  -      -      n      -      -      smtpd
#dnsblog  unix  -      -      n      -      0      dnsblog
#tlsproxy  unix  -      -      n      -      0      tlsproxy
submission inet n      -      n      -      -      smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_tls_cert_file=/etc/pki/tls/server.crt
  -o smtpd_tls_key_file=/etc/pki/tls/private/server.key
  -o smtpd_tls_auth_only=yes
# -o smtpd_reject_unlisted_recipient=no
# -o smtpd_client_restrictions=$mua_client_restrictions
# -o smtpd_helo_restrictions=$mua_helo_restrictions
# -o smtpd_sender_restrictions=$mua_sender_restrictions
```

Ensure that the `master.cf` file has the correct settings for the `submission` service

Run `sudo tail -f /var/log/maillog`

```
[root@goku ~]# sudo tail -f /var/log/maillog
Sep 17 17:08:53 goku postfix/submission/smtpd[7107]: lost connection after STARTTLS from goku.rockyserver.org[159.69.39.4]
Sep 17 17:08:53 goku postfix/cleanup[7111]: message-id=<20240917090853.3F9B31BE501@goku.rockyserver.org>
Sep 17 17:08:53 goku postfix/qmgr[6633]: 3F9B31BE501: from=<double-bounce@goku.rockyserver.org>, size=1074, nrcpt=1 (queue active)
Sep 17 17:08:53 goku postfix/submission/smtpd[7107]: disconnect from goku.rockyserver.org[159.69.39.4] ehlo=1 starttls=0/1 commands=1/2
Sep 17 17:08:53 goku postfix/local[7113]: 3F9B31BE501: to=<root@rockyserver.org>, orig_to=<postmaster>, relay=local, delay=0.04, delays=0.02/0.01/0.01, dsn=5.2.0, status=bounced (cannot update mailbox /var/mail/root for user root. unable to create lock file /var/mail/root.lock: No such file or directory)
Sep 17 17:08:53 goku postfix/bounce[7114]: warning: 3F9B31BE501: undeliverable postmaster notification discarded
Sep 17 17:08:53 goku postfix/qmgr[6633]: 3F9B31BE501: removed
Sep 17 17:12:13 goku postfix/anvil[7110]: statistics: max connection rate 1/60s for (submission:159.69.39.4) at Sep 17 17:08:53
Sep 17 17:12:13 goku postfix/anvil[7110]: statistics: max connection count 1 for (submission:159.69.39.4) at Sep 17 17:08:53
Sep 17 17:12:13 goku postfix/anvil[7110]: statistics: max cache size 1 at Sep 17 17:08:53
^C
[root@goku ~]#
```

The logs indicate several issues that may impact the functionality of Postfix and the ability to successfully retrieve the certificate information with `openssl`. Here's a breakdown of the log entries and their implications:

Log Analysis

Lost Connection After STARTTLS

`Sep 17 17:08:53 goku postfix/submission/smtpd[7107]: lost connection after STARTTLS from goku.rockyserver.org[159.69.39.4]`

`Sep 17 17:08:53 goku postfix/submission/smtpd[7107]: disconnect from goku.rockyserver.org[159.69.39.4] ehlo=1 starttls=0/1 commands=1/2`

- **Issue:** The connection is lost after the STARTTLS command.
- **Implication:** This suggests that the STARTTLS handshake is failing, which could be due to issues with SSL/TLS configuration or the client failing to handle the certificate properly.

Verify SSL/TLS Configuration

Make sure your SSL/TLS configuration in `main.cf` is correct:

- **Paths:** Verify the paths to the certificate and key files.
- **Permissions:** Ensure that Postfix has read permissions for the certificate and key files.

`sudo ls -l /etc/pki/tls/certs/server2.crt`

`sudo ls -l /etc/pki/tls/private/server2.key`

```
[root@goku ~]# ls -l /etc/pki/tls/certs/server2.crt
-rw-r--r--. 1 root root 1379 Sep 17 16:33 /etc/pki/tls/certs/server2.crt
[root@goku ~]# ls -l /etc/pki/tls/private/server2.key
-rw-----. 1 root root 1874 Sep 17 16:29 /etc/pki/tls/private/server2.key
[root@goku ~]#
```

Openssl connection still not working.

```
[root@goku ~]# openssl s_client -connect goku.rockyserver.org:587 -starttls smtp
CONNECTED(00000003)
0FB32FA0B7F000:error:0A00010B:SSL routines:ssl3_get_record:wrong version number:ssl/record/ssl3_record.c:354:
---
no peer certificate available
---
No client certificate CA names sent
---
SSL handshake has read 280 bytes and written 357 bytes
Verification: OK
---
New, (NONE), Cipher is (NONE)
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)
---
[root@goku ~]#
```

Check for Syntax Errors:

- Validate Postfix configuration:

The warnings from **postfix check** indicate that there is a typo or incorrect configuration in your ***main.cf*** file. Specifically, it appears that you have a repeated incorrect parameter name: ***smtpd_tls_protocols*** instead of ***smtpd_tls_protocols***. Remove the duplicate from the /etc/postfix/master.cf file.

Check Logs

Continue to monitor Postfix logs for any issues:

```
sudo tail -f /var/log/maillog
```

```
Sep 17 17:23:59 goku postfix/submission/smtpd[7574]: warning: TLS library problem: error:04800068:PEM routines::bad password read:crypto/pem/pem_pkey.c:155:  
Sep 17 17:23:59 goku postfix/submission/smtpd[7574]: warning: TLS library problem: error:0A080009:SSL routines::PEM Lib:ssl/ssl_rsa.c:384:  
Sep 17 17:23:59 goku postfix/submission/smtpd[7574]: connect from goku.rockyserver.org[159.69.39.4]  
Sep 17 17:23:59 goku postfix/submission/smtpd[7574]: lost connection after STARTTLS from goku.rockyserver.org[159.69.39.4]  
Sep 17 17:23:59 goku postfix/cleanup[7578]: message-id=<20240917092359.76A11140F0C@goku.rockyserver.org>  
Sep 17 17:23:59 goku postfix/submission/smtpd[7574]: disconnect from goku.rockyserver.org[159.69.39.4] ehlo=1 starttls=0/1 commands=1/2  
Sep 17 17:23:59 goku postfix/qmgr[7571]: 76A11140F0C: from=<double-bounce@goku.rockyserver.org>, size=1124, nrcpt=1 (queue active)  
Sep 17 17:23:59 goku postfix/local[7580]: 76A11140F0C: to=<root@rockyserver.org>, orig_to=<postmaster>, relay=local, delay=0.04, delays=0.02/0.01/0/0.01, dsn=  
5.2.0, status=bounced (cannot update mailbox /var/mail/root for user root. unable to create lock file /var/mail/root.lock: No such file or directory)  
Sep 17 17:23:59 goku postfix/bounce[7581]: warning: 76A11140F0C: undeliverable postmaster notification discarded  
Sep 17 17:23:59 goku postfix/qmgr[7571]: 76A11140F0C: removed  
^C
```

Remove Private Key Password (Optional)

I decided to remove the password from the private key (which is generally less secure). I used the following command:

```
openssl rsa -in /etc/pki/tls/private/server2.key -out /etc/pki/tls/private/server2_no_password.key
```

```
[root@goku ~]# openssl rsa -in /etc/pki/tls/private/server2.key -out /etc/pki/tls/private/server2_no_password.key  
Enter pass phrase for /etc/pki/tls/private/server2.key:  
writing RSA key  
[root@goku ~]#
```

Verify File Permissions

Ensure that the private key file has the correct permissions and ownership:

```
root@goku ~]#  
root@goku ~]# sudo chown root:root /etc/pki/tls/private/server2.key  
root[goku ~]# chmod 600 /etc/pki/tls/private/server2.key  
root@goku ~]#
```

Check Postfix TLS Settings:

- Verify TLS-related parameters in Postfix configuration:

```
sudo postconf | grep tls
```

```
[root@goku ~]# sudo postconf | grep tls
lsmtp_enforce_tls = no
lsmtp_sasl_tls_security_options = $lsmtp_sasl_security_options
lsmtp_sasl_tls_verified_security_options = $lsmtp_sasl_tls_security_options
lsmtp_starttls_timeout = 300s
lsmtp_tls_CAfile =
lsmtp_tls_CApath =
lsmtp_tls_block_early_mail_reply = no
lsmtp_tls_cert_file =
lsmtp_tls_chain_files =
lsmtp_tls_ciphers = medium
lsmtp_tls_connection_reuse = no
lsmtp_tls_dcrt_file =
lsmtp_tls_dkey_file = $lsmtp_tls_dcrt_file
lsmtp_tls_eccert_file =
lsmtp_tls_eckey_file = $lsmtp_tls_eccert_file
lsmtp_tls_enforce_peername = yes
lsmtp_tls_exclude_ciphers =
lsmtp_tls_fingerprint_cert_match =
lsmtp_tls_fingerprint_digest = md5
lsmtp_tls_force_insecure_host_tlsa_lookup = no
lsmtp_tls_key_file = $lsmtp_tls_cert_file
lsmtp_tls_loglevel = 0
lsmtp_tls_mandatory_ciphers = medium
lsmtp_tls_mandatory_exclude_ciphers =
lsmtp_tls_mandatory_protocols = !SSLv2, !SSLv3
lsmtp_tls_note_starttls_offer = no
lsmtp_tls_per_site =
lsmtp_tls_policy_maps =
lsmtp_tls_protocols = !SSLv2, !SSLv3
lsmtp_tls_scrt_verifydepth = 9
lsmtp_tls_secure_cert_match = nexthop
lsmtp_tls_security_level =
lsmtp_tls_servername =
lsmtp_tls_session_cache_database =
lsmtp_tls_session_cache_timeout = 3600s
lsmtp_tls_trust_anchor_file =



tlsproxy.client.fingerprint.digest = $ssmtp_tls_fingerprint_digest
tlsproxy.client.key_file = $ssmtp_tls_key_file
tlsproxy.client.level = $ssmtp_tls_security_level
tlsproxy.client.loglevel = $ssmtp_tls_loglevel
tlsproxy.client.loglevel_parameter = smtp_tls_loglevel
tlsproxy.client.per_site = $ssmtp_tls_per_site
tlsproxy.client.policy = $ssmtp_tls_policy_maps
tlsproxy.client.scrt_verifydepth = $ssmtp_tls_scrt_verifydepth
tlsproxy.client.use_tls = $smtp_use_tls
tlsproxy.enforce_tls = $smtpd_enforce_tls
tlsproxy.service_name = tlsproxy
tlsproxy.tls_CAfile = $smtpd_tls_CAfile
tlsproxy.tls_CApath = $smtpd_tls_CApath
tlsproxy.tls_always_issue_session_ids = $ssmtpd_tls_always_issue_session_ids
tlsproxy.tls_ask_ccert = $ssmtpd_tls_ask_ccert
tlsproxy.tls_ccert_verifydepth = $ssmtpd_tls_ccert_verifydepth
tlsproxy.tls_cert_file = $ssmtpd_tls_cert_file
tlsproxy.tls_chain_files = $ssmtpd_tls_chain_files
tlsproxy.tls_ciphers = $ssmtpd_tls_ciphers
tlsproxy.tls_dcrt_file = $ssmtpd_tls_dcrt_file
tlsproxy.tls_dh1024_param_file = $ssmtpd_tls_dh1024_param_file
tlsproxy.tls_dh512_param_file = $ssmtpd_tls_dh512_param_file
tlsproxy.tls_dkey_file = $ssmtpd_tls_dkey_file
tlsproxy.tls_eccert_file = $ssmtpd_tls_eccert_file
tlsproxy.tls_eckey_file = $ssmtpd_tls_eckey_file
tlsproxy.tls_eecdh_grade = $ssmtpd_tls_eecdh_grade
tlsproxy.tls_exclude_ciphers = $ssmtpd_tls_exclude_ciphers
tlsproxy.tls_fingerprint_digest = $ssmtpd_tls_fingerprint_digest
tlsproxy.tls_key_file = $ssmtpd_tls_key_file
tlsproxy.tls_loglevel = $ssmtpd_tls_loglevel
tlsproxy.tls_mandatory_ciphers = $ssmtpd_tls_mandatory_ciphers
tlsproxy.tls_mandatory_exclude_ciphers = $ssmtpd_tls_mandatory_exclude_ciphers
tlsproxy.tls_mandatory_protocols = $ssmtpd_tls_mandatory_protocols
tlsproxy.tls_protocols = $ssmtpd_tls_protocols
tlsproxy.tls_req_ccert = $ssmtpd_tls_req_ccert
tlsproxy.tls_security_level = $ssmtpd_tls_security_level
tlsproxy.use_tls = $ssmtpd_use_tls
tlsproxy_watchdog_timeout = 10s
```

The output of `postconf | grep tls` shows that several TLS-related parameters are not set for the SMTP service on port 587. This is likely the reason you're encountering issues with TLS and the `wrong version number` error.

Verify the permissions of the regenerated certificate and key files:

```
[root@goku ~]# sudo chown root:root /etc/pki/tls/certs/server2.crt
sudo chown root:root /etc/pki/tls/private/server2_no_password.key
sudo chmod 644 /etc/pki/tls/certs/server2.crt
sudo chmod 600 /etc/pki/tls/private/server2_no_password.key
[root@goku ~]#
```

Restart Postfix

After making these changes, restart Postfix:

```
[root@goku ~]# systemctl restart postfix
[root@goku ~]#
```

After this I had No more issues and the openssl connection is working.

10.0 Additional Features:

10.1 Network File System (NFS) Configuration

This section covers the step-by-step process to configure an NFS server on Ubuntu and set up a client on Rocky Linux. This setup allows file sharing between the two systems.

Install NFS Server

```
kashubuntu@vegeta:~/Desktop$ sudo apt install nfs-kernel-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  keyutils libevent-core-2.1-7 libnfsidmap1 nfs-common rpcbind
Suggested packages:
  open-iscsi watchdog
The following NEW packages will be installed:
  keyutils libevent-core-2.1-7 libnfsidmap1 nfs-common nfs-kernel-server
  rpcbind
0 upgraded, 6 newly installed, 0 to remove and 3 not upgraded.
Need to get 615 kB of archives.
After this operation, 2,235 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://my.archive.ubuntu.com/ubuntu jammy/main amd64 libevent-core-2.1-7 amd64 2.1.12-stable-1build3 [93.9 kB]
Get:2 http://my.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnfsidmap1 amd64 1:2.6.1-1ubuntu1.2 [42.9 kB]
Get:3 http://my.archive.ubuntu.com/ubuntu jammy/main amd64 rpcbind amd64 1:2.6.2-2build1 [46.6 kB]
Get:4 http://my.archive.ubuntu.com/ubuntu jammy/main amd64 keyutils amd64 1:0.1-1ubuntu3 [50.4 kB]
Get:5 http://my.archive.ubuntu.com/ubuntu jammy-updates/main amd64 nfs-common amd64 1:2.6.1-1ubuntu1.2 [241 kB]
Get:6 http://my.archive.ubuntu.com/ubuntu jammy-updates/main amd64 nfs-kernel-server amd64 1:2.6.1-1ubuntu1.2 [140 kB]
Fetched 615 kB in 1 (1,068 kB/s)
Selecting previously unselected package libevent-core-2.1-7:amd64.
(Reading database ... 203698 files and directories currently installed.)
Preparing to unpack .../0-libevent-core-2.1-7_2.1.12-stable-1build3_amd64.deb ...
Unpacking libevent-core-2.1-7:amd64 (2.1.12-stable-1build3) ...
Selecting previously unselected package libnfsidmap1:amd64.
Preparing to unpack .../1-libnfsidmap1_1x3a2.6.1-1ubuntu1.2_amd64.deb ...
Unpacking libnfsidmap1:amd64 (1:2.6.1-1ubuntu1.2) ...
Selecting previously unselected package rpcbind.
Preparing to unpack .../2-rpcbind_1.2.6.2-2build1_amd64.deb ...
Unpacking rpcbind (1.2.6.2-2build1) ...
Selecting previously unselected package keyutils.
Preparing to unpack .../3-keyutils_1.6.1-2ubuntu3_amd64.deb ...
Unpacking keyutils (1.6.1-2ubuntu3) ...
Selecting previously unselected package nfs-common.
```

First, install the NFS server on your Ubuntu system with the following command:

```
sudo apt install nfs-kernel-server
```

This installs the necessary NFS kernel modules and services to manage file sharing.

Create Shared NFS Directory

Next, create the directory that will be shared with the NFS clients:

```
kashubuntu@vegeta:~/Desktop$ sudo mkdir /srv/nfs_share
[sudo] password for kashubuntu:
kashubuntu@vegeta:~/Desktop$
```

- **Directory: /srv/nfs_share**
 - The `/srv` directory is a standard location for server data, making it appropriate for the shared folder.
- **/srv/nfs_share:** This directory will serve as the NFS share. The `/srv` directory is commonly used for server data, making it the appropriate location for the shared folder.

Set Directory Ownership and Permissions

```
kashubuntu@vegeta:~/Desktop$ sudo chown nobody:nogroup /srv/nfs_share
kashubuntu@vegeta:~/Desktop$ sudo chmod 777 /srv/nfs_share
```

Set Directory Ownership

- Change the ownership of the shared directory to ensure that no individual user has control over it, reducing security risks
- **`chown nobody:nogroup`**: This command changes the ownership of the directory to the nobody user and the nogroup group, which is a common security practice for shared directories. It ensures that no individual user on the server has ownership of the NFS directory, reducing potential security risks.

Set Directory Permissions

- Set full permissions on the directory to allow all users to read, write, and execute files:
- **`chmod 777`**: This command gives all users (owner, group, others) full access to the directory. It allows reading, writing, and executing files for every user. This is necessary to avoid permission issues across different NFS clients, ensuring smooth access.

Recursively Set Permissions

```
kashubuntu@vegeta:~/Desktop$ sudo chmod -R 777 /srv/nfs_share
```

If the directory contains subdirectories or files, apply the permission settings recursively:

- **`-R`**: This flag ensures that the permission settings apply to all files and subdirectories inside `/srv/nfs_share`. This is particularly important if new files are added in the future, as they will inherit the correct permissions.

Configure NFS Exports

To allow clients to access the shared directory, configure the NFS exports by editing the `/etc/exports` file:
Modify the `/etc/exports` File

```
kashubuntu@vegeta:~/Desktop$ sudo nano /etc/exports
```

```
GNU nano 6.2                                     /etc/exports *
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/srv/nfs_share *(rw,sync,no_subtree_check)
```

Add the above line to define the NFS share and its access permissions:

- **159.69.39.4**: The IP address of the client allowed to access the NFS share. In this case, it is your Rocky Linux client.
- **`rw`**: Grants read and write access to the client.

- **sync**: Ensures that changes made by the client are written to the disk immediately, which improves data integrity.
- **no_subtree_check**: Disables subtree checking for better performance. This option prevents NFS from checking the permissions of parent directories, which can slow down operations.

Export the NFS Directory

After configuring the /etc/exports file, make the NFS share available to clients by running:

```
kashubuntu@vegeta:~/Desktop$ sudo exportfs -a
kashubuntu@vegeta:~/Desktop$
```

This command exports all shared directories specified in the /etc/exports file, making them available to NFS clients.

Restart the NFS Server

To apply the new configuration, restart the NFS server:

```
kashubuntu@vegeta:~/Desktop$ sudo systemctl restart nfs-kernel-server
kashubuntu@vegeta:~/Desktop$
```

- This ensures that the NFS service is running and the new configuration is applied.

Configure the Firewall

Allow NFS traffic through the firewall by opening the NFS port (2049) for the client's IP:

```
kashubuntu@vegeta:~/Desktop$ sudo ufw allow from any to any port nfs
Rules updated
Rules updated (v6)
kashubuntu@vegeta:~/Desktop$
```

- This command enables NFS traffic from the client's IP address (159.69.39.4). The default port for NFS is 2049.

Enable the Firewall

Ensure that the firewall is active to protect your server from unauthorized access.

```
kashubuntu@vegeta:~/Desktop$ sudo ufw enable
Firewall is active and enabled on system startup
kashubuntu@vegeta:~/Desktop$
```

- This activates the firewall, ensuring that only allowed traffic, including NFS, is permitted.

Verify Firewall Status

Check if the firewall is active and configured to allow NFS traffic:

```
kashubuntu@vegeta:~/Desktop$ sudo ufw status
Status: active

To                         Action      From
--                         --          --
2049                       ALLOW       Anywhere
2049 (v6)                  ALLOW       Anywhere (v6)
```

- Look for the rule that allows access through port 2049. This ensures that the firewall is correctly configured.

NFS Client Configuration (Rocky Linux):

Install NFS Client

On the Rocky Linux client, install the NFS utilities required to interact with the NFS server:

```
[root@goku ~]# sudo dnf install nfs-utils
Last metadata expiration check: 1:23:11 ago on Tue 17 Sep 2024 08:59:49 PM.
Dependencies resolved.
=====
Transaction Summary
=====
Install 8 Packages

Total download size: 830 k
Installed size: 2.1 M
Is this ok [y/N]: y
Downloading Packages:
(1/8): libverto-libbev-0.3.2-3.el9.x86_64.rpm           161 kB/s | 13 kB   00:00
(2/8): libbev-4.33-5.el9.x86_64.rpm                   596 kB/s | 52 kB   00:00
(3/8): rpcbind-1.2.6-7.el9.x86_64.rpm                583 kB/s | 56 kB   00:00
(4/8): gssproxy-0.8.4-6.el9.x86_64.rpm              2.2 MB/s | 108 kB  00:00
(5/8): ssasd-nfs-idmap-2.9.4-6.el9.4.1.x86_64.rpm    517 kB/s | 41 kB   00:00
(6/8): nfs-utils-2.5.4-26.el9.4.x86_64.rpm          5.3 MB/s | 429 kB  00:00
(7/8): libnfsidmap-2.5.4-26.el9.4.x86_64.rpm        1.4 MB/s | 59 kB   00:00
(8/8): keyutils-1.6.3-1.el9.x86_64.rpm               850 kB/s | 72 kB   00:00
=====
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
```

```

Running transaction
Preparing : 
Installing : libnfsidmap-1:2.5.4-26.el9_4.x86_64 1/1
Installing : keyutils-1.6.3-1.el9.x86_64 1/8
Installing : libev-4.33-5.el9.x86_64 2/8
Installing : libverto-libev-0.3.2-3.el9.x86_64 3/8
Installing : gssproxy-0.8.4-6.el9.x86_64 4/8
Running scriptlet: gssproxy-0.8.4-6.el9.x86_64 5/8
Running scriptlet: rpcbind-1.2.6-7.el9.x86_64 5/8
Installing : rpcbind-1.2.6-7.el9.x86_64 6/8
Running scriptlet: rpcbind-1.2.6-7.el9.x86_64 6/8
Created symlink /etc/systemd/system/multi-user.target.wants/rpcbind.service → /usr/lib/systemd/system/rpcbind.service.
Created symlink /etc/systemd/system/sockets.target.wants/rpcbind.socket → /usr/lib/systemd/system/rpcbind.socket.

Running scriptlet: nfs-utils-1:2.5.4-26.el9_4.x86_64 7/8
Installing : nfs-utils-1:2.5.4-26.el9_4.x86_64 7/8
Running scriptlet: nfs-utils-1:2.5.4-26.el9_4.x86_64 7/8
Installing : sssd-nfs-idmap-2.9.4-6.el9_4.1.x86_64 8/8
Running scriptlet: sssd-nfs-idmap-2.9.4-6.el9_4.1.x86_64 8/8
Verifying : libverto-libev-0.3.2-3.el9.x86_64 1/8
Verifying : rpcbind-1.2.6-7.el9.x86_64 2/8
Verifying : libev-4.33-5.el9.x86_64 3/8
Verifying : sssd-nfs-idmap-2.9.4-6.el9_4.1.x86_64 4/8
Verifying : gssproxy-0.8.4-6.el9.x86_64 5/8
Verifying : nfs-utils-1:2.5.4-26.el9_4.x86_64 6/8
Verifying : libnfsidmap-1:2.5.4-26.el9_4.x86_64 7/8
Verifying : keyutils-1.6.3-1.el9.x86_64 8/8

Installed:
  gssproxy-0.8.4-6.el9.x86_64           keyutils-1.6.3-1.el9.x86_64           libev-4.33-5.el9.x86_64           libnfsidmap-1:2.5.4-26.el9_4.x86_64
  libverto-libev-0.3.2-3.el9.x86_64      nfs-utils-1:2.5.4-26.el9_4.x86_64      rpcbind-1.2.6-7.el9.x86_64       sssd-nfs-idmap-2.9.4-6.el9_4.1.x86_64

Complete!
[root@goku ~]#

```

- This package provides the tools needed to mount NFS shares and interact with the NFS server.

Create a Mount Point

Next, create a directory on the client system that will serve as the mount point for the NFS share:

```
[root@goku ~]# sudo mkdir /mnt/nfs_share
```

- */mnt/nfs_share*: This is where the Ubuntu server's shared directory will be mounted.

Mount the NFS Share

Mount the NFS share from the Ubuntu server to the Rocky Linux client.

```
[root@goku ~]# sudo mount -t nfs 159.69.39.60:/srv/nfs_share /mnt/nfs_share
```

- **159.69.39.60:/srv/nfs_share**: The NFS share on the Ubuntu server.
- **/mnt/nfs_share**: The local mount point on the Rocky Linux client.

Verify NFS Mount

To ensure the NFS share is mounted, check the mounted file systems:

```
[root@goku ~]# df -h | grep nfs
159.69.39.60:/srv/nfs_share 34G   14G   19G  42% /mnt/nfs_share
[root@goku ~]#
```

- This shows the mounted NFS share along with its available disk space.

Test Write Access

To confirm that the client can write to the NFS share, create a test file.

```
[root@goku ~]# touch /mnt/nfs_share/test_file
```

- This creates a new file in the mounted directory, which resides on the Ubuntu server.

Verify on the NFS Server

Navigate to the NFS share on the Ubuntu server and check for the presence of the test file.

```
kashubuntu@vegeta:~/Desktop$ cd /srv/nfs_share
kashubuntu@vegeta:/srv/nfs_share$
```



```
kashubuntu@vegeta:/srv/nfs_share$ ls -l
total 0
-rw-r--r-- 1 nobody nogroup 0 Sep 17 22:29 test_file
kashubuntu@vegeta:/srv/nfs_share$
```

- **cd /srv/nfs_share:** The cd (change directory) command is used to navigate to the NFS shared directory, `/srv/nfs_share`, on the Ubuntu server. This is necessary because the `test_file` was created in this specific directory from the Rocky Linux client. By changing to the correct directory, you can
- **ls -l:** The ls -l command is used to list the contents of the current directory in long format. This format shows detailed information about each file, including its permissions, ownership, size, and timestamp. Using this command helps confirm not only the presence of the `test_file` but also that the file was correctly written with the appropriate permissions and ownership settings. This level of detail ensures that the NFS server and client are communicating properly.

Unmount and Remount NFS Share

This step is crucial to ensure the stability of the NFS connection. We will first unmount the NFS share from the Rocky Linux client, verify the unmount, and then remount it to confirm the connection can be re-established without issues.

Unmount the NFS Share

To unmount the NFS share, we use the following command:

```
[root@goku ~]# sudo umount /mnt/nfs_share
```

This will disconnect the client from the shared folder on the Ubuntu server.

Verify Unmount

Check if the directory is successfully unmounted.

```
[root@goku ~]# df -h | grep nfs
[root@goku ~]#
[root@goku ~]# cd /mnt/nfs_share
[root@goku nfs_share]# ls -l
total 0
[root@goku nfs_share]#
```

There should be no output, confirming that the NFS share has been successfully unmounted. Additionally, navigating to the /mnt/nfs_share directory should show it as empty, ensuring the connection to the Ubuntu server is no longer active, as shown above.

Remount the NFS Share

Now, we will remount the NFS share to re-establish the connection between the Rocky Linux client and the Ubuntu server. This is done with the same command used earlier:

```
[root@goku ~]#
[root@goku ~]# sudo mount -t nfs 159.69.39.60:/srv/nfs_share /mnt/nfs_share
[root@goku ~]#
```

Verify the Remount:

To confirm the share is mounted again, we check the mounted file systems:

```
[root@goku ~]#
[root@goku ~]# df -h | grep nfs
159.69.39.60:/srv/nfs_share 34G 14G 19G 42% /mnt/nfs_share
[root@goku ~]#
```

As we can see it is successful, as the NFS share is listed, showing it is mounted at */mnt/nfs_share*.

Check the Contents of the Mounted Directory

To ensure the remount worked correctly, we list the contents of the mounted directory:

```
[root@goku ~]#
[root@goku ~]# ls -l /mnt/nfs_share
total 0
-rw-r--r--. 1 nobody nobody 0 Sep 17 22:29 test_file
[root@goku ~]#
```

The test_file that was created earlier on the Rocky Linux client should still be visible, confirming that the remount is functioning properly and the files are accessible again.

Test Writing to the NFS Share After Remount

To further verify that the NFS share is fully operational, we will create a new test file after the remount:

```
[root@goku ~]#
[root@goku ~]# touch /mnt/nfs_share/test_file_2
[root@goku ~]#
```

Confirm Test Writing:

After remounting, we then check the Ubuntu server to confirm that the new file was successfully created. Navigate to the shared directory on the Ubuntu server and list the contents:

```
root@vegeta:~# ls -l /srv/nfs_share
total 0
-rw-r--r-- 1 nobody nogroup 0 Sep 17 22:29 test_file
-rw-r--r-- 1 nobody nogroup 0 Sep 17 23:13 test_file_2
root@vegeta:~#
```

The presence of `test_file_2` verifies that the Rocky Linux client can still write to the NFS share even after the remount, ensuring the stability of the connection.

Final Checks

Check Firewall on the Server

Ensure the NFS server firewall is still configured to allow traffic through port 2049.

```
root@vegeta:~#
root@vegeta:~# sudo ufw status | grep 2049
2049                         ALLOW      Anywhere
2049 (v6)                     ALLOW      Anywhere (v6)
root@vegeta:~#
```

On the ubuntu server, ensure that port 2049 is allowed through the firewall. This verifies that the NFS traffic can pass through the firewall. Since it is properly configured, we can see the rule allowing access to port 2049 displayed in the output.

Check Logs for NFS Issues

Check the system logs for any NFS-related issues.

```
[root@goku ~]# journalctl -xe | grep nfs
Sep 17 16:36:34 goku.rockyserver.org kernel: SELinux: policy capability genfs_seclabel_symlinks=1
Sep 17 16:36:35 goku.rockyserver.org kernel: SELinux: policy capability genfs_seclabel_symlinks=1
Sep 17 22:22:51 goku.rockyserver.org sudo[8611]:    root : TTY=pts/3 ; PWD=/root ; USER=root ; COMMAND=/bin/dnf install nfs-utils
Sep 17 22:23:13 goku.rockyserver.org useradd[8743]: new user: name=rpcuser, UID=29, GID=29, home=/var/lib/nfs, shell=/sbin/nologin, from=none
Subject: A start job for unit var-lib-nfs-rpc_pipefs.mount has begun execution.
A start job for unit var-lib-nfs-rpc_pipefs.mount has begun execution.
Subject: A start job for unit var-lib-nfs-rpc_pipefs.mount has finished successfully.
A start job for unit var-lib-nfs-rpc_pipefs.mount has finished successfully.
Subject: A start job for unit nfs-client.target has finished successfully.
A start job for unit nfs-client.target has finished successfully.
Sep 17 22:25:31 goku.rockyserver.org sudo[9775]:    root : TTY=pts/3 ; PWD=/root ; USER=root ; COMMAND=/bin/mkdir /mnt/nfs_share
Sep 17 22:27:11 goku.rockyserver.org sudo[9778]:    root : TTY=pts/3 ; PWD=/root ; USER=root ; COMMAND=/bin/mount -t nfs 159.69.39.60:/srv/nfs_share /mnt/nfs_share
Sep 17 22:59:52 goku.rockyserver.org sudo[9899]:    root : TTY=pts/3 ; PWD=/root ; USER=root ; COMMAND=/bin/umount /mnt/nfs_share
Sep 17 22:59:52 goku.rockyserver.org systemd[1]: mnt-nfs_share.mount: Deactivated successfully.
The unit mnt-nfs_share.mount has successfully entered the 'dead' state.
Sep 17 23:06:34 goku.rockyserver.org sudo[9967]:    root : TTY=pts/3 ; PWD=/root ; USER=root ; COMMAND=/bin/mount -t nfs 159.69.39.60:/srv/nfs_share /mnt/nfs_share
Sep 17 23:06:34 goku.rockyserver.org nfshead[9981]: setting /mnt/nfs_share readahead to 128
[root@goku ~]#
```

Review system logs for any NFS-related issues. Based on the journalctl log output, the NFS operations completed successfully:

- NFS utilities installed:
`sudo[8611]: root : COMMAND=/bin/dnf install nfs-utils`
- NFS share mounted:
`sudo[9778]: root : COMMAND=/bin/mount -t nfs 159.69.39.60:/srv/nfs_share /mnt/nfs_share`
- Successful unmount and remount:
`systemd[1]: mnt-nfs_share.mount: Deactivated successfully.`

The NFS configuration has been successfully completed with no issues found in the logs. All tests (mounting, writing, unmounting, and remounting) were conducted smoothly, verifying proper functionality between the Rocky Linux client and the Ubuntu NFS server. The setup is stable and ready for production use.

10.2 Configuring SASL for Dovecot and Postfix

Installing Cyrus SASL:

- Upon investigating the status of the **saslauthd** service, I discovered that it was not installed. I installed the necessary packages using the following command:
bash

```
[kashsna@goku ~]$ sudo dnf install cyrus-sasl cyrus-sasl-plain cyrus-sasl-scram
Rocky Linux 9 - BaseOS           2.6 kB/s | 4.1 kB    00:01
Rocky Linux 9 - AppStream        3.2 kB/s | 4.5 kB    00:01
Rocky Linux 9 - Extras          2.9 kB/s | 2.9 kB    00:01
Package cyrus-sasl-plain-2.1.27-21.el9.x86_64 is already installed.
Dependencies resolved.
=====
Package           Architecture Version      Repository  Size
=====
Installing:
cyrus-sasl       x86_64      2.1.27-21.el9   baseos     71 k
cyrus-sasl-scram x86_64      2.1.27-21.el9   baseos     28 k

Transaction Summary
=====
Install 2 Packages

Total download size: 99 k
Installed size: 191 k
Is this ok [y/N]:
```

These packages provide SASL (Simple Authentication and Security Layer) support for Postfix.

Enabling and Starting saslauthd:

- After installation, I enabled and started the **saslauthd** service:

```
[kashsna@goku ~]$
[kashsna@goku ~]$ sudo systemctl enable saslauthd
Created symlink /etc/systemd/system/multi-user.target.wants/saslauthd.service → /usr/lib/systemd/system/saslauthd.service.
[...]
[...]
[kashsna@goku ~]$ sudo systemctl start saslauthd
[...]
```

Configuring Postfix to Use SASL:

- I then needed to configure **Postfix** to use SASL for authentication. I opened the **/etc/postfix/main.cf** file and added the following lines:

```
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
"/etc/postfix/main.cf" 750L, 29902B
```

These settings enable SASL authentication in Postfix and define the path where Postfix will communicate with **Dovecot** for user authentication.

Ensuring Correct saslauthd Directory Configuration

2. Verifying saslauthd Directory:

- Next, I checked the configuration in the **/etc/sysconfig/saslauthd** file to ensure that the directory was correctly set. This file specifies where the **saslauthd** socket should be created.

```
[kashsna@goku ~]$ sudo vi /etc/sysconfig/saslauthd  
# Directory in which to place saslauthd's  
# on. This directory must already exist.  
SOCKETDIR=/var/run/saslauthd
```

Restarting saslauthd and Postfix:

- After ensuring the configuration was correct, I restarted both **saslauthd** and **Postfix** to apply the changes:
bash

```
[kashsna@goku ~]$ sudo systemctl restart saslauthd  
[kashsna@goku ~]$ sudo systemctl restart postfix
```

Checking saslauthd Directory:

- To confirm that the **saslauthd** directory existed and was properly configured, I ran:

```
[kashsna@goku ~]$ ls -l /var/run/saslauthd  
total 4  
srwxrwxrwx. 1 root root 0 Sep 16 12:09 mux  
-rw-----. 1 root root 0 Sep 16 12:09 mux.accept  
-rw-----. 1 root root 5 Sep 16 12:09 saslauthd.pid
```

Since the directory does not exist, Postfix will not be able to communicate with **saslauthd**.

Adding Postfix to saslauthd Group

2. Adding Postfix to saslauthd Group:

- Postfix requires access to the **saslauthd** directory to perform authentication.
First, I attempted to add **Postfix** to the **saslauthd** group:

```
[kashsna@goku ~]$ sudo usermod -aG saslauthd postfix  
usermod: group 'saslauthd' does not exist
```

However, the system returned an error indicating that the **saslauthd** group did not exist.

Creating the saslauthd Group:

- I created the **saslauthd** group using: **sudo groupadd saslauthd**
- After creating the group, I reran the command to add **Postfix** to the group:

```
[kashsna@goku ~]$ sudo groupadd saslauthd  
[kashsna@goku ~]$ sudo usermod -aG saslauthd postfix
```

Setting Permissions for the saslauthd Directory:

- After adding Postfix to the group, I set the correct permissions for the **saslauthd** directory:

```
[kashsna@goku ~]$ sudo chgrp saslauthd /var/run/saslauthd  
[kashsna@goku ~]$ sudo chmod 750 /var/run/saslauthd
```

Restarting Services:

- Once again, I restarted **saslauthd** and **Postfix** to apply the changes:

```
[kashsna@goku ~]$ systemctl restart saslauthd
[kashsna@goku ~]$ sudo systemctl restart postfix
[kashsna@goku ~]
```

Verifying Permissions:

- I verified that the **saslauthd** directory had the correct permissions:

```
[kashsna@goku ~]$ ls -ld /var/run/saslauthd
drwxr-x---. 2 root saslauthd 100 Sep 16 12:25 /var/run/saslauthd
[kashsna@goku ~]
```

The output showed that the directory was now correctly configured with the appropriate group ownership and permissions.

Configuring Dovecot for SASL

2. Editing Dovecot Configuration (10-master.conf):

- To ensure Dovecot properly supports SASL authentication, I opened the **/etc/dovecot/conf.d/10-master.conf** file. I added the **unix listener** configuration under **auth-userdb{}** and **auth-master{}**:

```
[kashsna@goku ~]$ sudo vi /etc/dovecot/conf.d/10-master.conf
```

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

These settings ensure that Postfix can communicate with Dovecot for authentication, and the correct permissions are applied to the **auth-master** socket.

Restarting Dovecot and Postfix:

- After modifying the configuration, I restarted **Dovecot** and **Postfix**:

```
[root@goku ~]# systemctl restart postfix
[root@goku ~]# systemctl restart dovecot
[root@goku ~]
```

After configuring SASL with Dovecot and Postfix, I tested sending and receiving emails via Thunderbird, and the SASL authentication worked as expected.

