Internet & Network Services: Assignment 2

- Daniel Sheehan
- R00204226

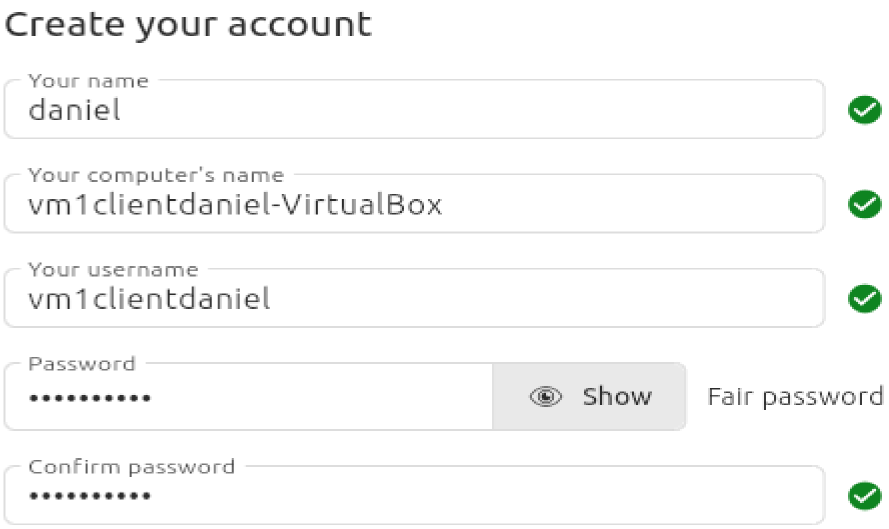
Introduction:

In this report I will be going through the steps of implementing a range of network services for a company named CALMnet. To do this I will be using Oracle VirtualBox to act as the system administrator and set up and install Ubuntu based Virtual Machines consisting of 1 DHCP client and 2 servers with a variety of essential services installed. All of these machines will be operating under the subnet of **10.0.26.0/24.**

The services I will be configuring include a web server with Ruby on Rails, DNS (primary and secondary), DHCP, SSH, an email server, iSCSI storage, network printing (CUPS) and lastly an open source network management system. I will be documenting this process with screenshots and a final reflection on the experience and outcomes of the assignment.

VM Name:	OS:	Role:	Network Adapter:	IP Address:
VM 1	Ubuntu Desktop	DHCP Client	Internal Network	DHCP
VM 2	Ubuntu Server	Main Server	Internal Network	Static (10.0.26.12)
VM 3	Ubuntu Server	Infrastructure Server	Internal Network	Static (10.0.26.13)

VM 1 Login:



Create your account

Your name: daniel ✓

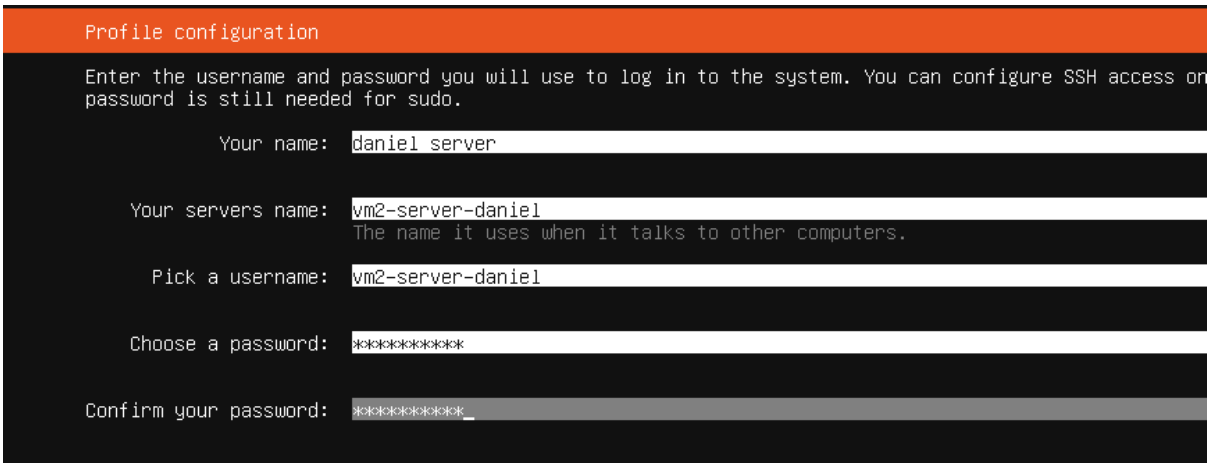
Your computer's name: vm1clientdaniel-VirtualBox ✓

Your username: vm1clientdaniel ✓

Password: •••••••••• Fair password

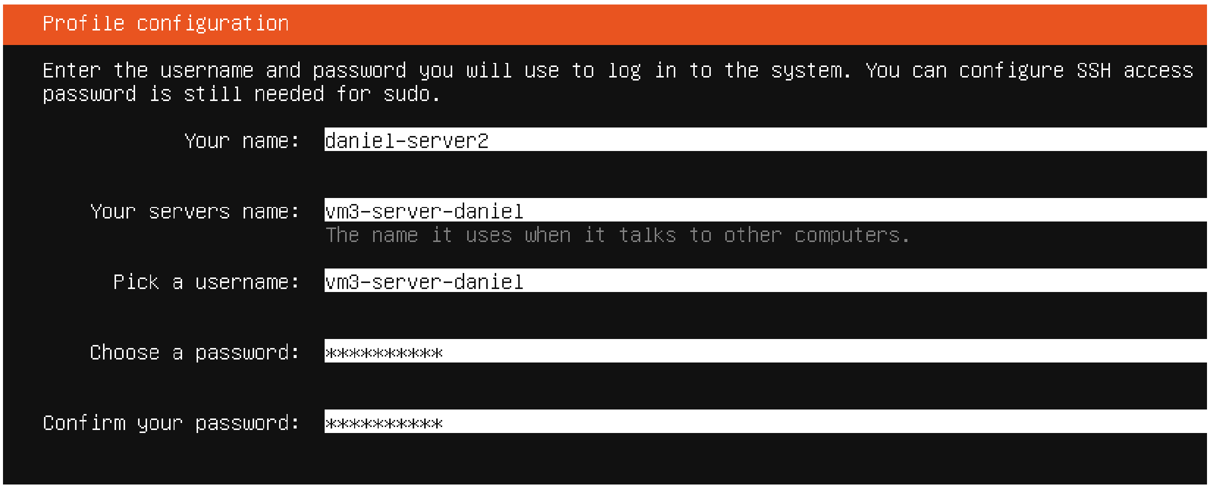
Confirm password: •••••••••• ✓

VM 2 Login:



```
Profile configuration

Enter the username and password you will use to log in to the system. You can configure SSH access on
password is still needed for sudo.

           Your name: daniel server

   Your servers name: vm2-server-daniel
                      The name it uses when it talks to other computers.

     Pick a username: vm2-server-daniel

   Choose a password: ***********

Confirm your password: ***********_
```

VM 3 Login:



```
Profile configuration

Enter the username and password you will use to log in to the system. You can configure SSH access
password is still needed for sudo.

           Your name: daniel-server2

   Your servers name: vm3-server-daniel
                      The name it uses when it talks to other computers.

     Pick a username: vm3-server-daniel

   Choose a password: ***********

Confirm your password: ***********
```

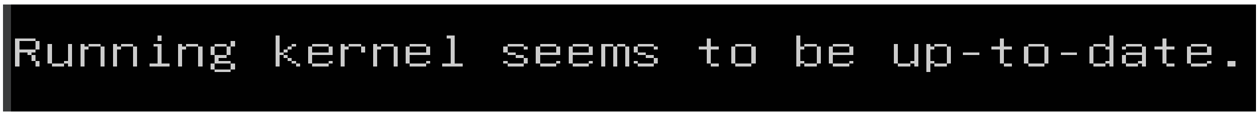
After all of my Virtual machines were set up and installed I ensured they were set to NAT in order to install updates using:

Sudo apt install -y

I followed this up with:

Sudo apt upgrade -y

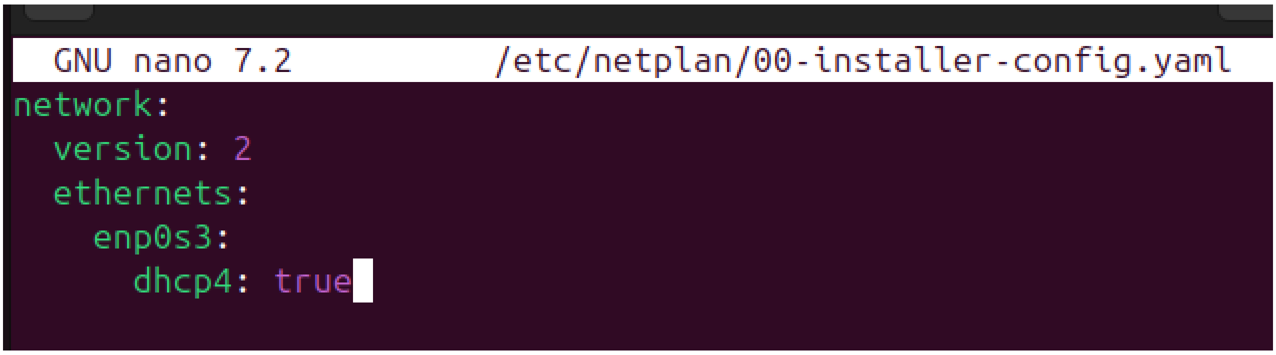
This then allowed me to upgrade all currently installed packages on these new machines ensuring they were all secure and fully up to date in order to limit the number of bugs and errors encountered throughout the installation process.



```
Running kernel seems to be up-to-date.
```

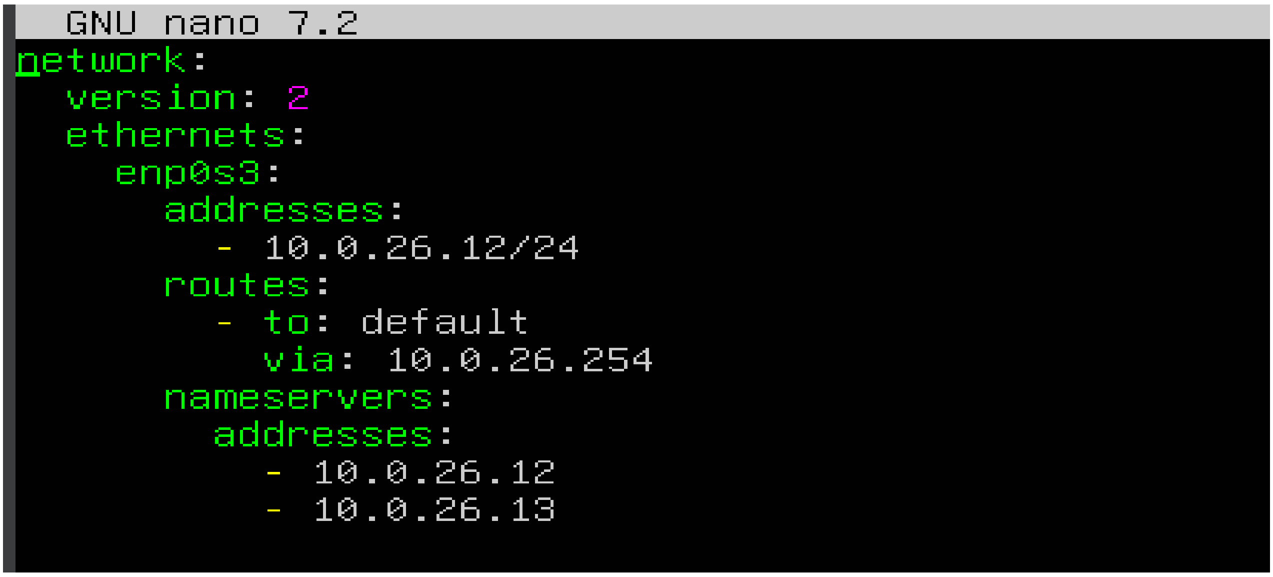
Network Layout:

VM1 Client netplan file:



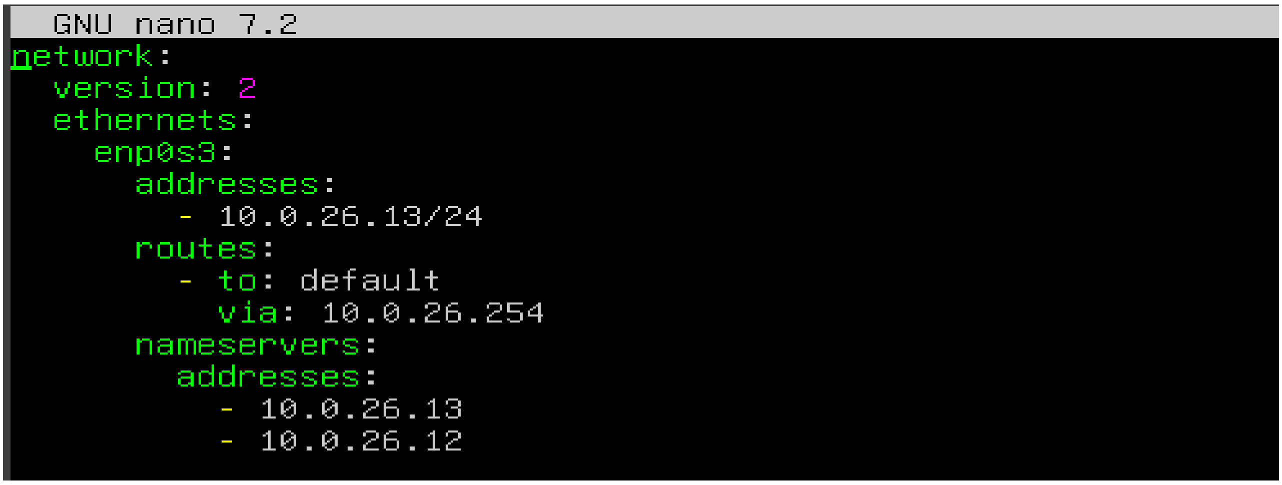
```
  GNU nano 7.2              /etc/netplan/00-installer-config.yaml
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
```

VM2 Server netplan file:



```
  GNU nano 7.2
network:
  version: 2
  ethernets:
    enp0s3:
      addresses:
        - 10.0.26.12/24
      routes:
        - to: default
          via: 10.0.26.254
      nameservers:
        addresses:
          - 10.0.26.12
          - 10.0.26.13
```

VM3 Server netplan file:

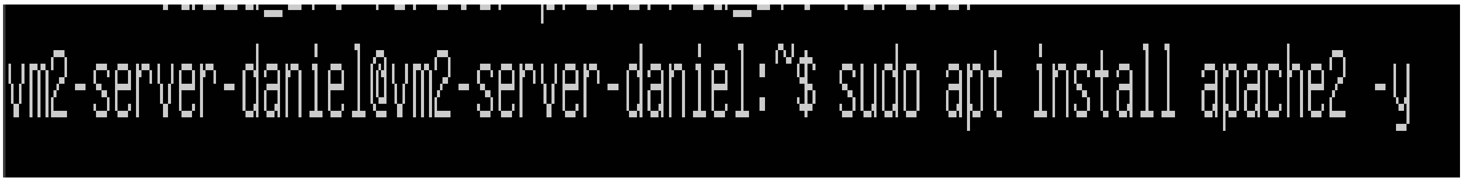


```
  GNU nano 7.2
network:
  version: 2
  ethernets:
    enp0s3:
      addresses:
        - 10.0.26.13/24
      routes:
        - to: default
          via: 10.0.26.254
      nameservers:
        addresses:
          - 10.0.26.13
          - 10.0.26.12
```

Apache Server & Ruby on Rails:[i]

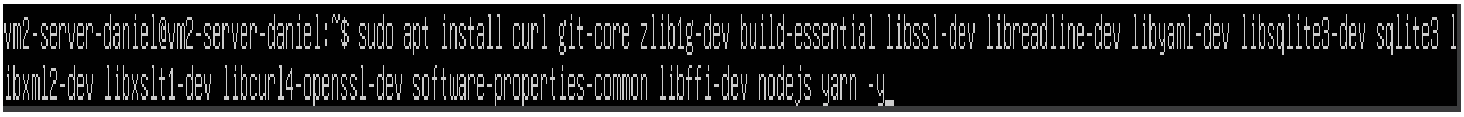
To begin I opted to start with the setup and installation of an Apache Server that will be running Ruby on Rails on Virtual Machine 2 which is going to act as my main server.

I began by typing: **sudo apt install apache2 -y** , this allowed me to install the Apache http server which will be the main web server used to host my ruby on rails app. Before carrying out this installation I had to switch back from internal network to NAT interface to access the internet.



```
vm2-server-daniel@vm2-server-daniel:~$ sudo apt install apache2 -y
```

Next, I install the dependencies needed to support Ruby on Rails, this included things such as developer tools, libraries and package managers that I would need to build and run my rails app. These packages mainly had compilers, SSL libraries and JavaScript tools like Node.js and Yarn which are needed for Rails.



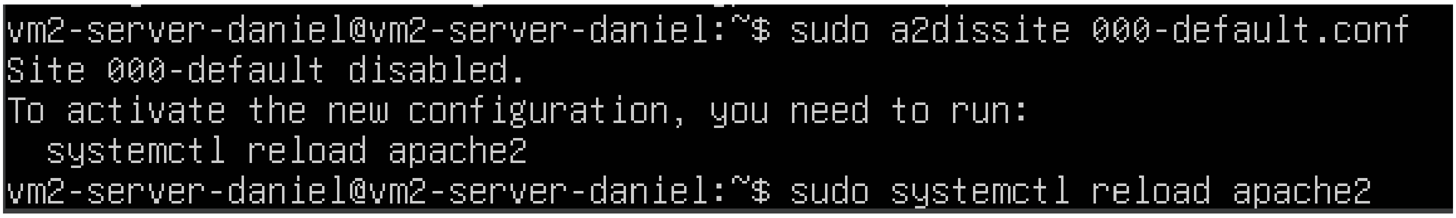
```
vm2-server-daniel@vm2-server-daniel:~$ sudo apt install curl git-core zlib1g-dev build-essential libssl-dev libreadline-dev libyaml-dev libsqlite3-dev sqlite3 l
ibxml2-dev libxslt1-dev libcurl4-openssl-dev software-properties-common libffi-dev nodejs yarn -y_
```

Disabling default Apache site: Before I began to set up my own Rails app being served on Apache; I decided to disable the default Apache virtual host configuration file by running:

Sudo a2dissite 000-default.conf

The reason I did this was so Apache wouldn't go to the default page and instead I could set a custom virtual host which would point to my Rails app instead. After all this I reloaded Apache to apply the changes:

Sudo systemctl reload apache2

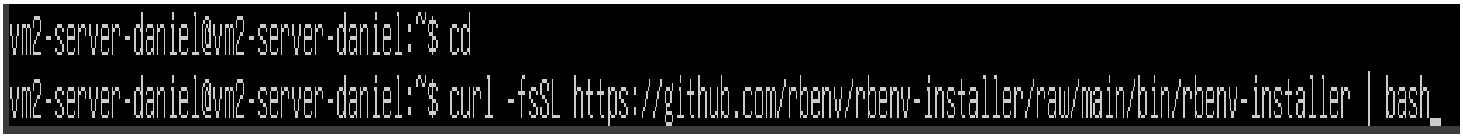


```
vm2-server-daniel@vm2-server-daniel:~$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
vm2-server-daniel@vm2-server-daniel:~$ sudo systemctl reload apache2
```

To install Ruby and manage it I decided to use **rbenv**, I downloaded and ran the official installation script from github using:

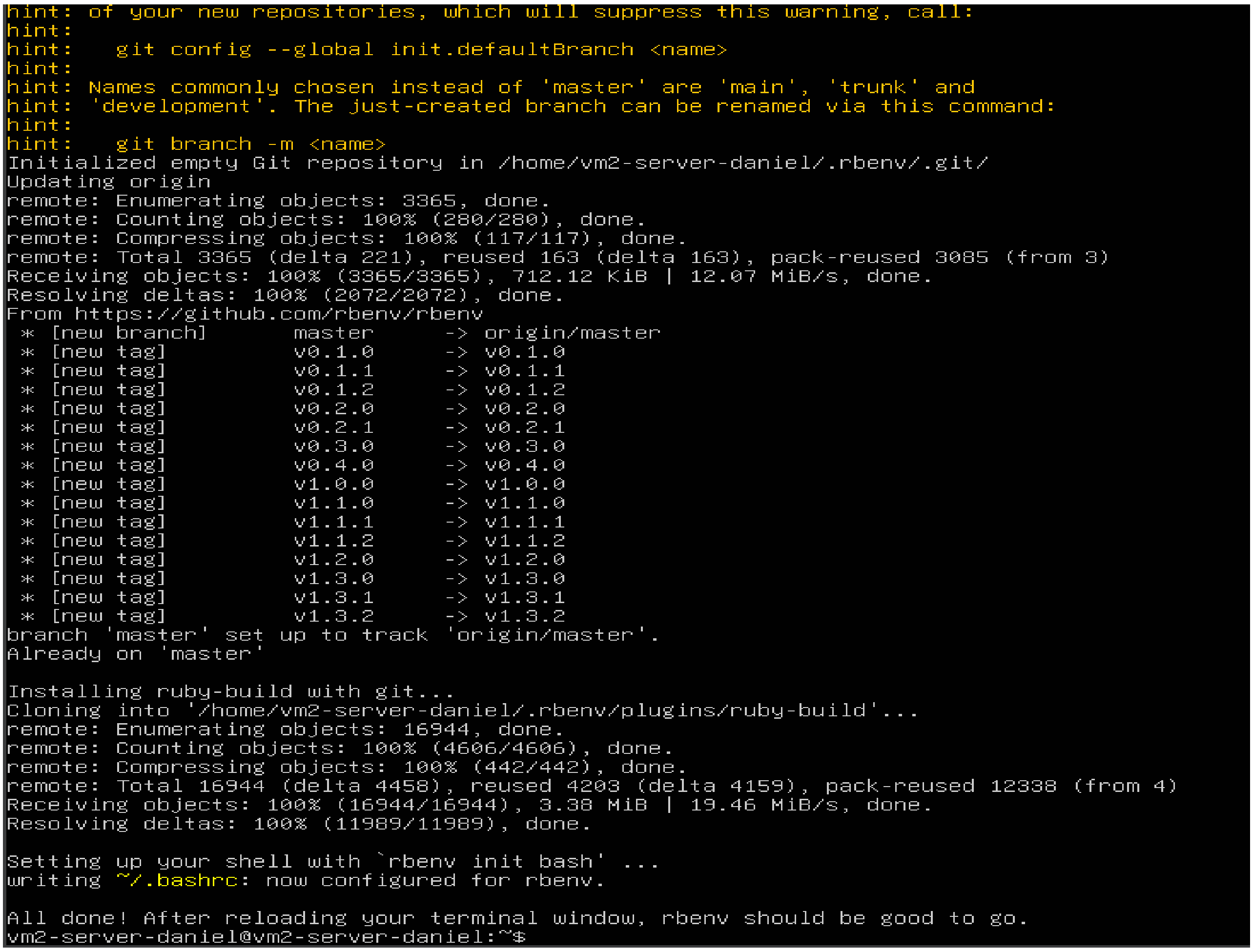
Curl -fsSL https://github.com/rbenv/rbenv-installer/raw/main/bin/rbenv-installer | bash

This let me install rbenv and ruby-build letting me easily install and switch between different versions of Ruby for my Rails app.



```
vm2-server-daniel@vm2-server-daniel:~$ cd
vm2-server-daniel@vm2-server-daniel:~$ curl -fsSL https://github.com/rbenv/rbenv-installer/raw/main/bin/rbenv-installer | bash_
```

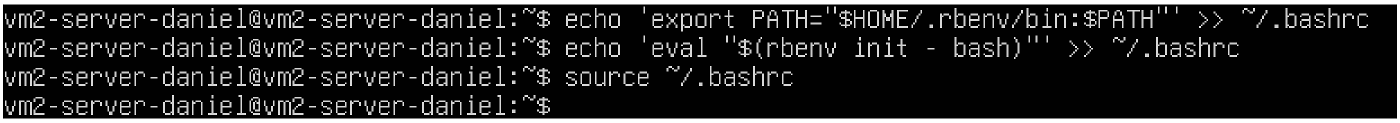
Next the rbenv installer cloned any of the files and plugins it needed including ruby-build, It then set up .bashhrc so I could use rbenv to install and manager Ruby versions easily.



```
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/vm2-server-daniel/.rbenv/.git/
Updating origin
remote: Enumerating objects: 3365, done.
remote: Counting objects: 100% (280/280), done.
remote: Compressing objects: 100% (117/117), done.
remote: Total 3365 (delta 221), reused 163 (delta 163), pack-reused 3085 (from 3)
Receiving objects: 100% (3365/3365), 712.12 KiB | 12.07 MiB/s, done.
Resolving deltas: 100% (2072/2072), done.
From https://github.com/rbenv/rbenv
 * [new branch]      master     -> origin/master
 * [new tag]         v0.1.0     -> v0.1.0
 * [new tag]         v0.1.1     -> v0.1.1
 * [new tag]         v0.1.2     -> v0.1.2
 * [new tag]         v0.2.0     -> v0.2.0
 * [new tag]         v0.2.1     -> v0.2.1
 * [new tag]         v0.3.0     -> v0.3.0
 * [new tag]         v0.4.0     -> v0.4.0
 * [new tag]         v1.0.0     -> v1.0.0
 * [new tag]         v1.1.0     -> v1.1.0
 * [new tag]         v1.1.1     -> v1.1.1
 * [new tag]         v1.1.2     -> v1.1.2
 * [new tag]         v1.2.0     -> v1.2.0
 * [new tag]         v1.3.0     -> v1.3.0
 * [new tag]         v1.3.1     -> v1.3.1
 * [new tag]         v1.3.2     -> v1.3.2
branch 'master' set up to track 'origin/master'.
Already on 'master'

Installing ruby-build with git...
Cloning into '/home/vm2-server-daniel/.rbenv/plugins/ruby-build'...
remote: Enumerating objects: 16944, done.
remote: Counting objects: 100% (4606/4606), done.
remote: Compressing objects: 100% (442/442), done.
remote: Total 16944 (delta 4458), reused 4203 (delta 4159), pack-reused 12338 (from 4)
Receiving objects: 100% (16944/16944), 3.38 MiB | 19.46 MiB/s, done.
Resolving deltas: 100% (11989/11989), done.

Setting up your shell with `rbenv init bash' ...
writing ~/.bashrc: now configured for rbenv.

All done! After reloading your terminal window, rbenv should be good to go.
vm2-server-daniel@vm2-server-daniel:~$
```

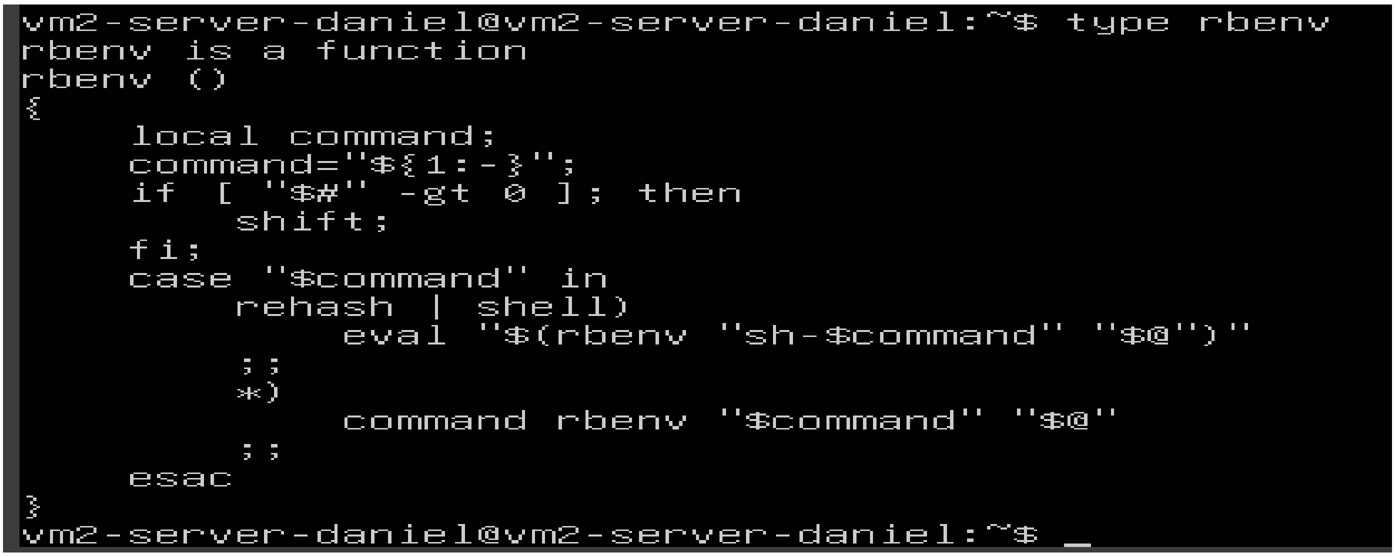
After this I updated .bashhrc to include rbenv in my path so it would use it for new sessions, I then ran:

Source ~/.bashrc so the changes could be applied.



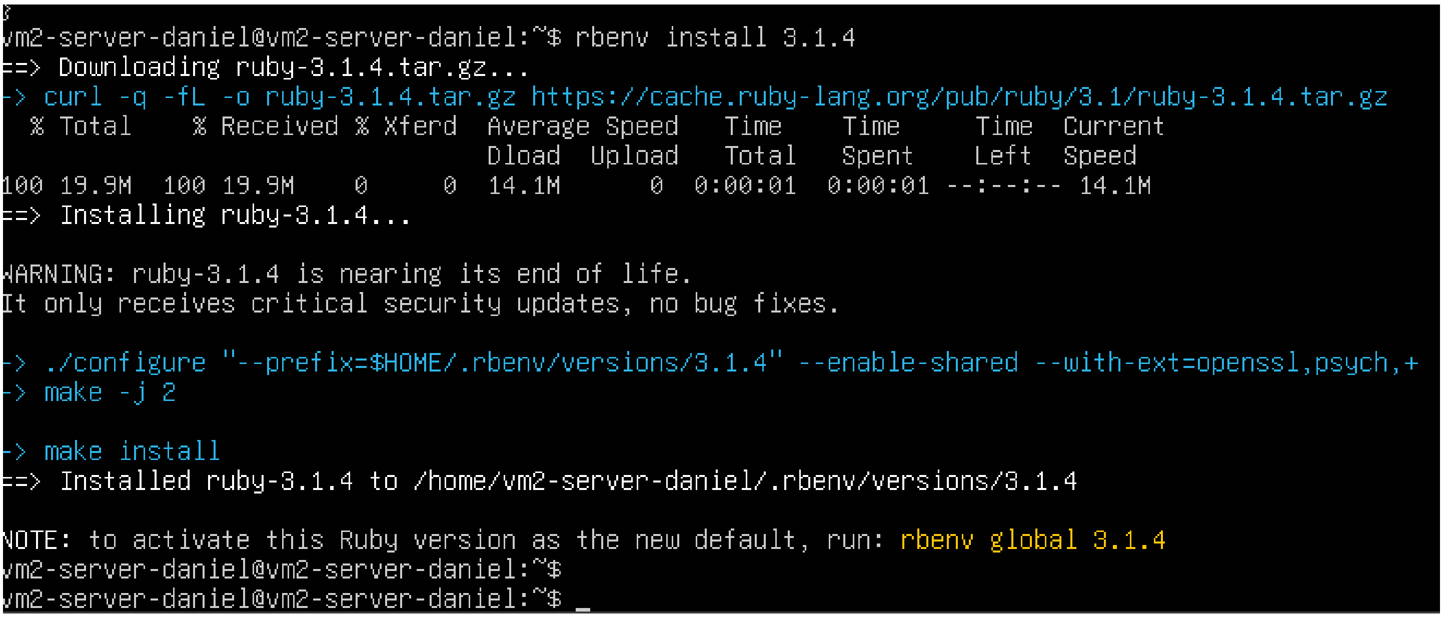
```
vm2-server-daniel@vm2-server-daniel:~$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc
vm2-server-daniel@vm2-server-daniel:~$ echo 'eval "$(rbenv init - bash)"' >> ~/.bashrc
vm2-server-daniel@vm2-server-daniel:~$ source ~/.bashrc
vm2-server-daniel@vm2-server-daniel:~$
```

I verified that rbenv was set up by running: **type rbenv** , It showed as a shell function telling me that my installation and path config worked correctly.



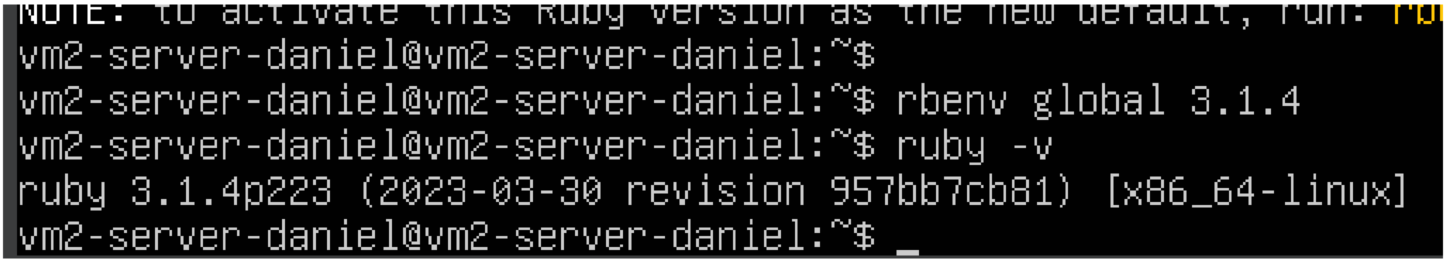
```
vm2-server-daniel@vm2-server-daniel:~$ type rbenv
rbenv is a function
rbenv ()
{
    local command;
    command="${1:-}";
    if [ "$#" -gt 0 ]; then
        shift;
    fi;
    case "$command" in
        rehash | shell)
            eval "$(rbenv "sh-$command" "$@")"
        ;;
        *)
            command rbenv "$command" "$@"
        ;;
    esac
}
vm2-server-daniel@vm2-server-daniel:~$ _
```

I installed Ruby version 3.1.4 using **rbenv install 3.14** and confirmed it was installed[ii].



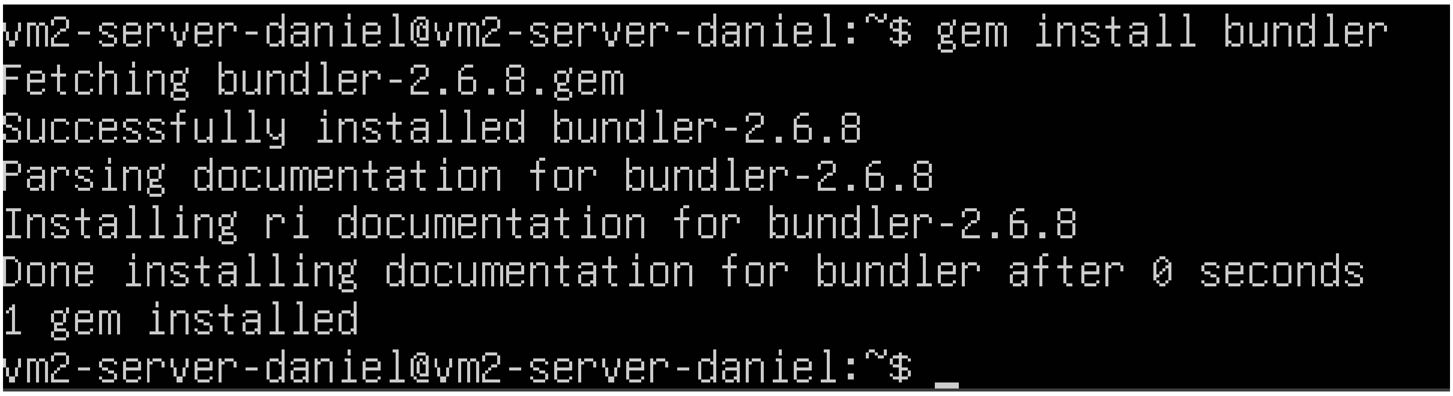
```
vm2-server-daniel@vm2-server-daniel:~$ rbenv install 3.1.4
==> Downloading ruby-3.1.4.tar.gz...
-> curl -q -fL -o ruby-3.1.4.tar.gz https://cache.ruby-lang.org/pub/ruby/3.1/ruby-3.1.4.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 19.9M  100 19.9M    0     0  14.1M      0  0:00:01  0:00:01 --:--:-- 14.1M
==> Installing ruby-3.1.4...

WARNING: ruby-3.1.4 is nearing its end of life.
It only receives critical security updates, no bug fixes.

-> ./configure "--prefix=$HOME/.rbenv/versions/3.1.4" --enable-shared --with-ext=openssl,psych,+
-> make -j 2

-> make install
==> Installed ruby-3.1.4 to /home/vm2-server-daniel/.rbenv/versions/3.1.4

NOTE: to activate this Ruby version as the new default, run: rbenv global 3.1.4
vm2-server-daniel@vm2-server-daniel:~$
vm2-server-daniel@vm2-server-daniel:~$ _
```

Next I set Ruby 3.1.4 as the global version using **rbenv global 3.1.4** and confirmed it with **ruby -v** This showed me it was active and working.



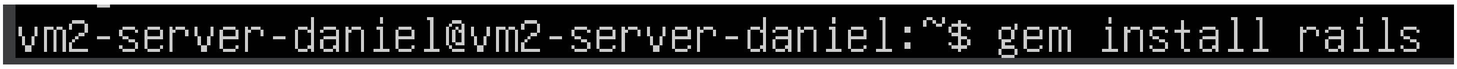
```
NOTE: to activate this Ruby version as the new default, run: rb
vm2-server-daniel@vm2-server-daniel:~$
vm2-server-daniel@vm2-server-daniel:~$ rbenv global 3.1.4
vm2-server-daniel@vm2-server-daniel:~$ ruby -v
ruby 3.1.4p223 (2023-03-30 revision 957bb7cb81) [x86_64-linux]
vm2-server-daniel@vm2-server-daniel:~$ _
```

After that I installed Bunder using **gem install bundler**, which completed successfully, This is needed to manage Ruby gem dependencies for my Rails application. It wouldn't work otherwise.



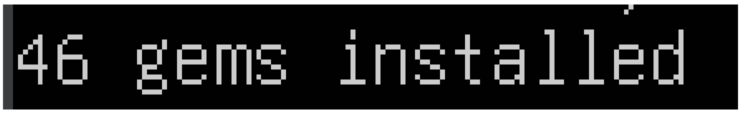
```
vm2-server-daniel@vm2-server-daniel:~$ gem install bundler
Fetching bundler-2.6.8.gem
Successfully installed bundler-2.6.8
Parsing documentation for bundler-2.6.8
Installing ri documentation for bundler-2.6.8
Done installing documentation for bundler after 0 seconds
1 gem installed
vm2-server-daniel@vm2-server-daniel:~$ _
```

I then ran **gem install rails** to install the Ruby on Rails framework so I can host my web application on the server.



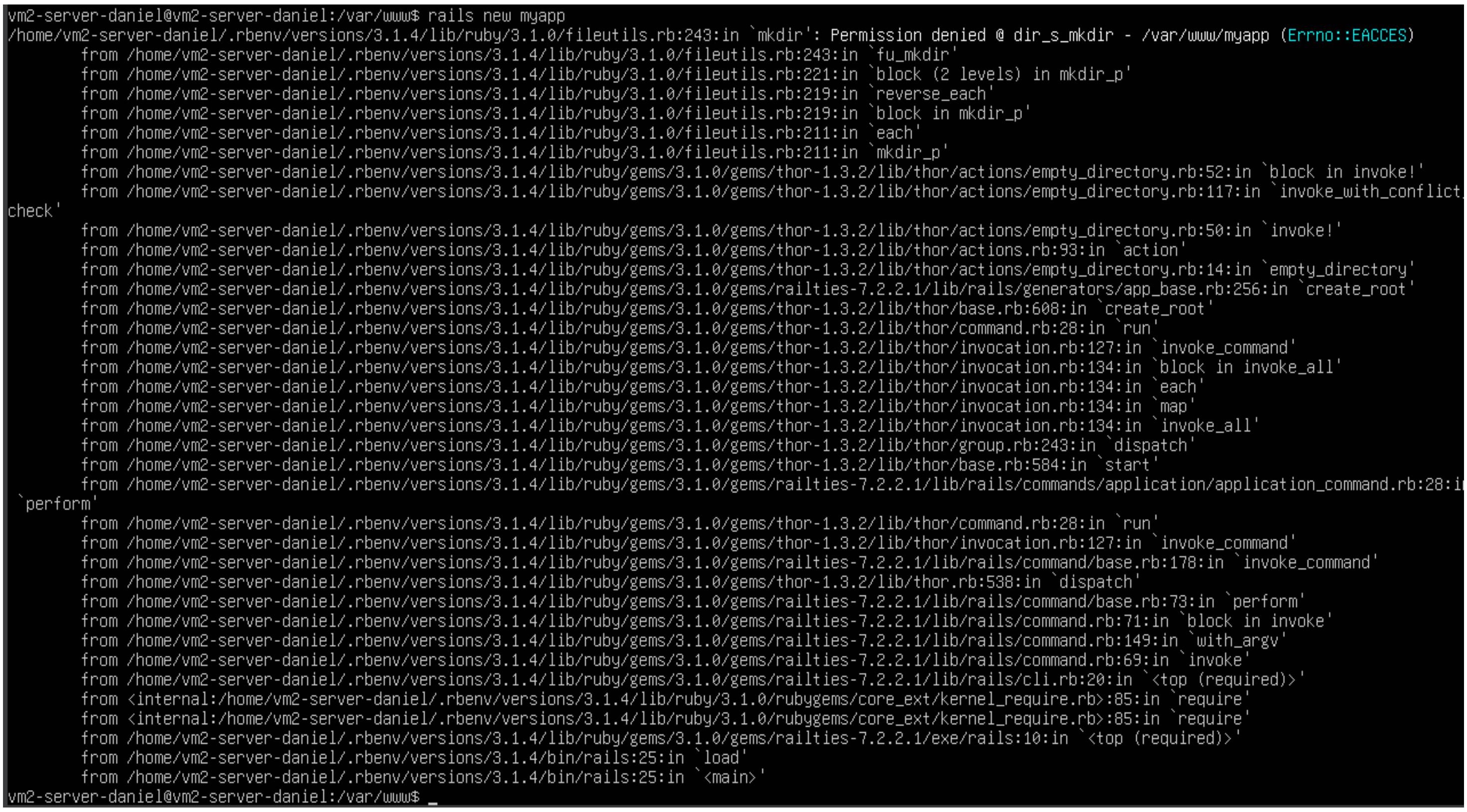
```
vm2-server-daniel@vm2-server-daniel:~$ gem install rails
```

After this I was met with the message **46 gems installed** telling me Rails was installed successfully.



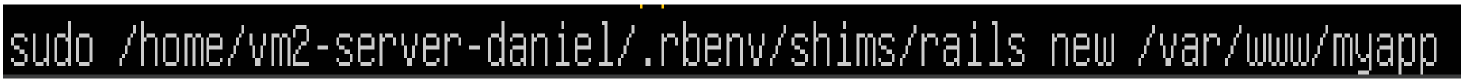
```
46 gems installed
```

To create the Rails app I used **rails new myapp** inside of the /var/www directory but I was met with an error the first time as I didn't use elevated privileges.



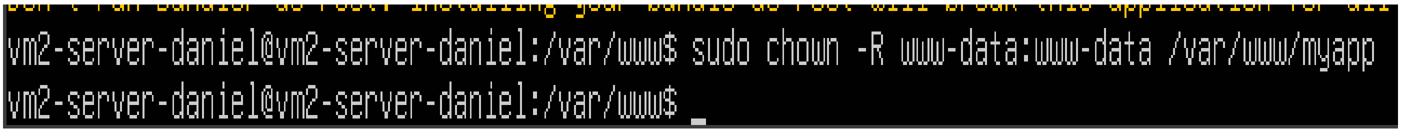
```
vm2-server-daniel@vm2-server-daniel:/var/www$ rails new myapp
/home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/3.1.0/fileutils.rb:243:in `mkdir': Permission denied @ dir_s_mkdir - /var/www/myapp (Errno::EACCES)
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/3.1.0/fileutils.rb:243:in `fu_mkdir'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/3.1.0/fileutils.rb:221:in `block (2 levels) in mkdir_p'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/3.1.0/fileutils.rb:219:in `reverse_each'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/3.1.0/fileutils.rb:219:in `block in mkdir_p'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/3.1.0/fileutils.rb:211:in `each'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/3.1.0/fileutils.rb:211:in `mkdir_p'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/actions/empty_directory.rb:52:in `block in invoke!'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/actions/empty_directory.rb:117:in `invoke_with_conflict
check'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/actions/empty_directory.rb:50:in `invoke!'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/actions.rb:93:in `action'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/actions/empty_directory.rb:14:in `empty_directory'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/railties-7.2.2.1/lib/rails/generators/app_base.rb:256:in `create_root'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/base.rb:608:in `create_root'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/command.rb:28:in `run'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/invocation.rb:127:in `invoke_command'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/invocation.rb:134:in `block in invoke_all'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/invocation.rb:134:in `each'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/invocation.rb:134:in `map'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/invocation.rb:134:in `invoke_all'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/group.rb:243:in `dispatch'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/base.rb:584:in `start'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/railties-7.2.2.1/lib/rails/commands/application/application_command.rb:28:i
 `perform'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/command.rb:28:in `run'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor/invocation.rb:127:in `invoke_command'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/railties-7.2.2.1/lib/rails/command/base.rb:178:in `invoke_command'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/thor-1.3.2/lib/thor.rb:538:in `dispatch'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/railties-7.2.2.1/lib/rails/command/base.rb:73:in `perform'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/railties-7.2.2.1/lib/rails/command.rb:71:in `block in invoke'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/railties-7.2.2.1/lib/rails/command.rb:149:in `with_argv'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/railties-7.2.2.1/lib/rails/command.rb:69:in `invoke'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/railties-7.2.2.1/lib/rails/cli.rb:20:in `<top (required)>'
        from <internal:/home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/3.1.0/rubygems/core_ext/kernel_require.rb>:85:in `require'
        from <internal:/home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/3.1.0/rubygems/core_ext/kernel_require.rb>:85:in `require'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/lib/ruby/gems/3.1.0/gems/railties-7.2.2.1/exe/rails:10:in `<top (required)>'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/bin/rails:25:in `load'
        from /home/vm2-server-daniel/.rbenv/versions/3.1.4/bin/rails:25:in `<main>'
vm2-server-daniel@vm2-server-daniel:/var/www$ _
```

When I got that working I used sudo to run the Rails generator to create the application in /var/www/myapp. This created the full directory for my Rails app.



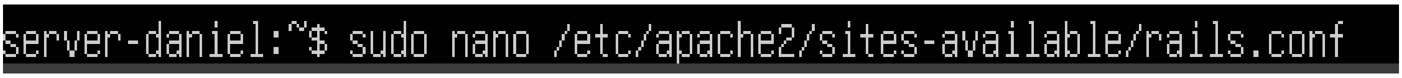
```
sudo /home/vm2-server-daniel/.rbenv/shims/rails new /var/www/myapp
```

Next I changed the ownership of my Rails app to www-data , This is because Apache runs under that user and makes it so the server is able to use the application files properly.



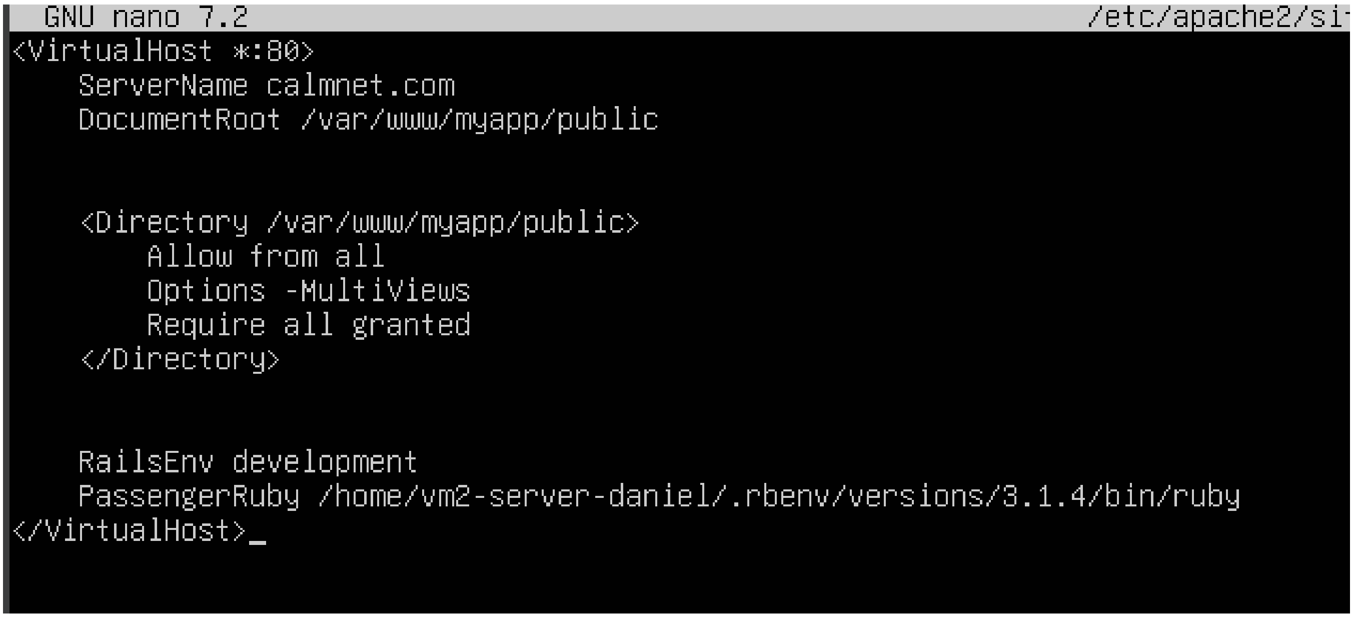
```
Don't run Bundler as root. Installing your bundle as root will break this application for all
vm2-server-daniel@vm2-server-daniel:/var/www$ sudo chown -R www-data:www-data /var/www/myapp
vm2-server-daniel@vm2-server-daniel:/var/www$ _
```

After that I made a new apache virtual host file named rails.conf so Apache would serve my Rails app from /var/www/myapp/public



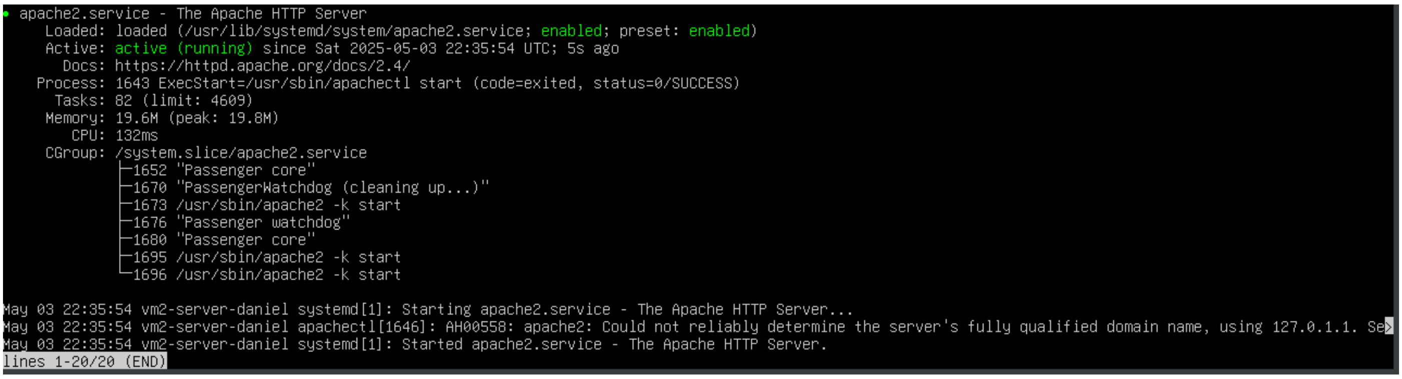
```
server-daniel:~$ sudo nano /etc/apache2/sites-available/rails.conf
```

Inside the rails.conf file I set the ServerName variable to calmnet.com and specified the document root path to match my rails app. Lastly, I saved and exited to save the changes I made to the config file.



```
  GNU nano 7.2                                               /etc/apache2/si
<VirtualHost *:80>
    ServerName calmnet.com
    DocumentRoot /var/www/myapp/public


    <Directory /var/www/myapp/public>
        Allow from all
        Options -MultiViews
        Require all granted
    </Directory>


    RailsEnv development
    PassengerRuby /home/vm2-server-daniel/.rbenv/versions/3.1.4/bin/ruby
</VirtualHost>_
```

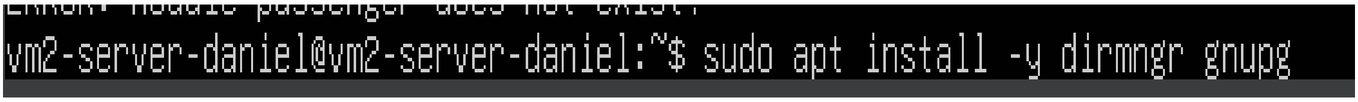
After enabling the Rails site and reloading Apache I confirmed that the apache2 service was running successfully showing it was ready to host my Ruby on Rails app.



```
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-05-03 22:35:54 UTC; 5s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 1643 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
      Tasks: 82 (limit: 4609)
     Memory: 19.6M (peak: 19.8M)
        CPU: 132ms
     CGroup: /system.slice/apache2.service
             ├─1652 "Passenger core"
             ├─1670 "PassengerWatchdog (cleaning up...)"
             ├─1673 /usr/sbin/apache2 -k start
             ├─1676 "Passenger watchdog"
             ├─1680 "Passenger core"
             ├─1695 /usr/sbin/apache2 -k start
             └─1696 /usr/sbin/apache2 -k start

May 03 22:35:54 vm2-server-daniel systemd[1]: Starting apache2.service - The Apache HTTP Server...
May 03 22:35:54 vm2-server-daniel apachectl[1646]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Se>
May 03 22:35:54 vm2-server-daniel systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-20/20 (END)
```

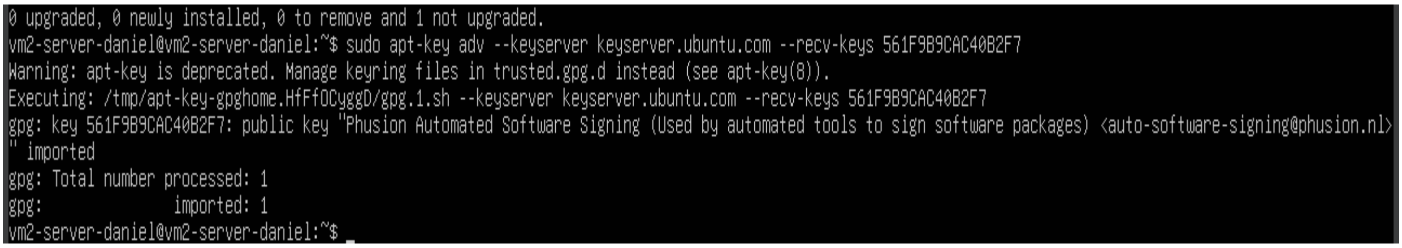
It was at this point I realised that I ran into an error as I was missing the Passenger Module required for my Rails app to work properly on Apache.

To fix this I installed **dirmngr** and **gnupg** so I could add the Phusion Passenger repository and GPG key that I was missing.



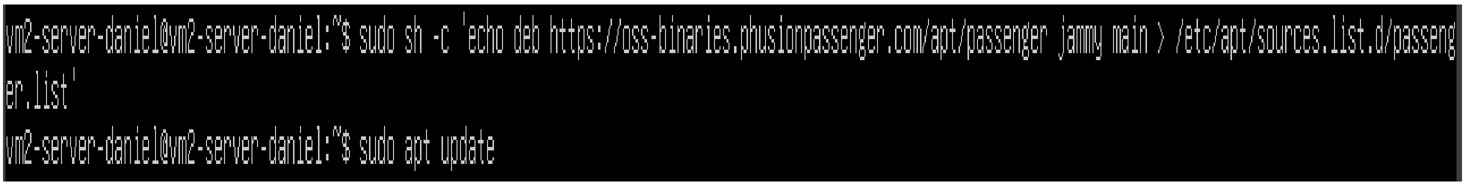
```
ERROR: Module passenger does not exist!
vm2-server-daniel@vm2-server-daniel:~$ sudo apt install -y dirmngr gnupg
```

Next I added the Phusion Passenger GPG key to allow me to safely install and add packages from their repositories.



```
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
vm2-server-daniel@vm2-server-daniel:~$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 561F9B9CAC40B2F7
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
Executing: /tmp/apt-key-gpghome.HfFfOCyggD/gpg.1.sh --keyserver keyserver.ubuntu.com --recv-keys 561F9B9CAC40B2F7
gpg: key 561F9B9CAC40B2F7: public key "Phusion Automated Software Signing (Used by automated tools to sign software packages) <auto-software-signing@phusion.nl>
" imported
gpg: Total number processed: 1
gpg:               imported: 1
vm2-server-daniel@vm2-server-daniel:~$ _
```

I then was ready to add the passenger module from the repositories and ran sudo apt update after it was finished so the packages would be refreshed.

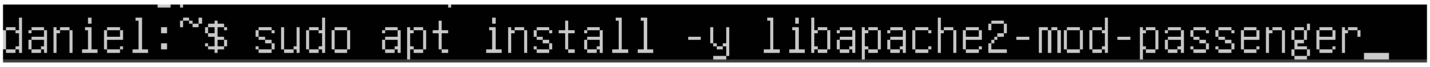


```
vm2-server-daniel@vm2-server-daniel:~$ sudo sh -c 'echo deb https://oss-binaries.phusionpassenger.com/apt/passenger jammy main > /etc/apt/sources.list.d/passeng
er.list'
vm2-server-daniel@vm2-server-daniel:~$ sudo apt update
```

I installed the passenger Apache module using the command:

Sudo apt install -y libapache2-mod-passenger

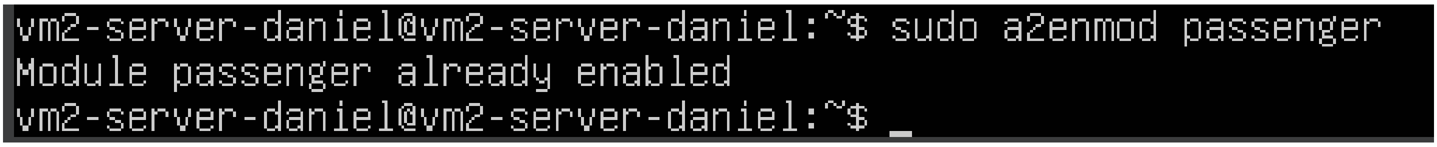
This passenger module finally made it so my Apache server could properly host my Ruby app.



```
daniel:~$ sudo apt install -y libapache2-mod-passenger_
```

I enabled the passenger module using: **sudo a2enmod passenger**

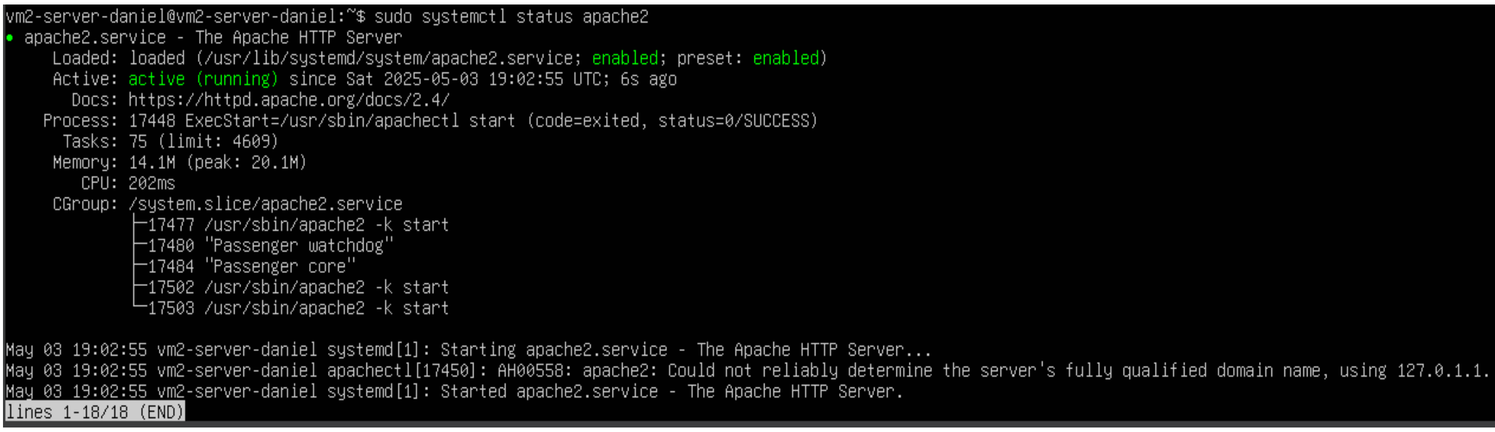
I could then see it was already enabled so I knew it was working properly now.



```
vm2-server-daniel@vm2-server-daniel:~$ sudo a2enmod passenger
Module passenger already enabled
vm2-server-daniel@vm2-server-daniel:~$ _
```

Just to double check I just ran: **sudo systemctl status apache2**

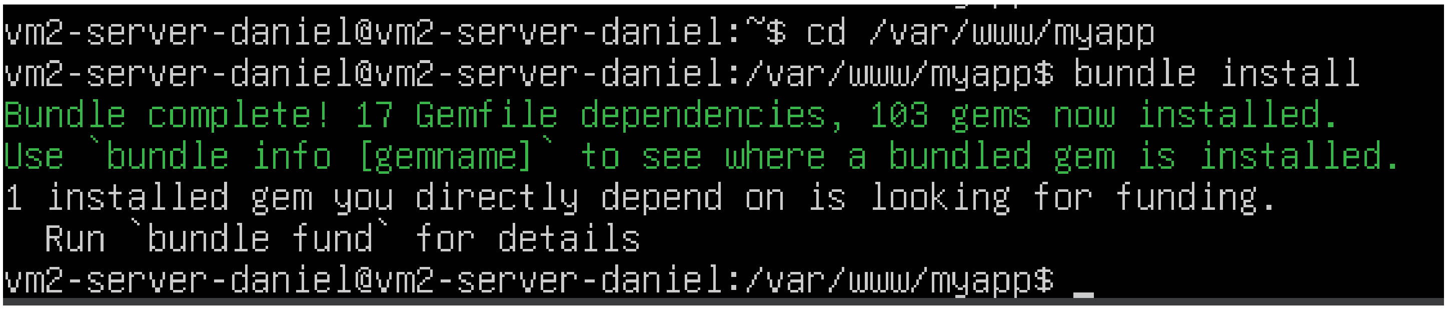
This showed me Apache was active and now running with no issues.



```
vm2-server-daniel@vm2-server-daniel:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-05-03 19:02:55 UTC; 6s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 17448 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
      Tasks: 75 (limit: 4609)
     Memory: 14.1M (peak: 20.1M)
        CPU: 202ms
     CGroup: /system.slice/apache2.service
             ├─17477 /usr/sbin/apache2 -k start
             ├─17480 "Passenger watchdog"
             ├─17484 "Passenger core"
             ├─17502 /usr/sbin/apache2 -k start
             └─17503 /usr/sbin/apache2 -k start

May 03 19:02:55 vm2-server-daniel systemd[1]: Starting apache2.service - The Apache HTTP Server...
May 03 19:02:55 vm2-server-daniel apachectl[17450]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1.
May 03 19:02:55 vm2-server-daniel systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-18/18 (END)
```

Once my Rails app was made I went to the app's directory and ran: **bundle install**

This installed all dependencies that were in the gem file, The bundle completed with 103 gems installed meaning my app was ready to run now.



```
vm2-server-daniel@vm2-server-daniel:~$ cd /var/www/myapp
vm2-server-daniel@vm2-server-daniel:/var/www/myapp$ bundle install
Bundle complete! 17 Gemfile dependencies, 103 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
1 installed gem you directly depend on is looking for funding.
  Run `bundle fund` for details
vm2-server-daniel@vm2-server-daniel:/var/www/myapp$ _
```

To check if the Apache server was correctly hosting my Rails app I ran:

Curl http://localhost

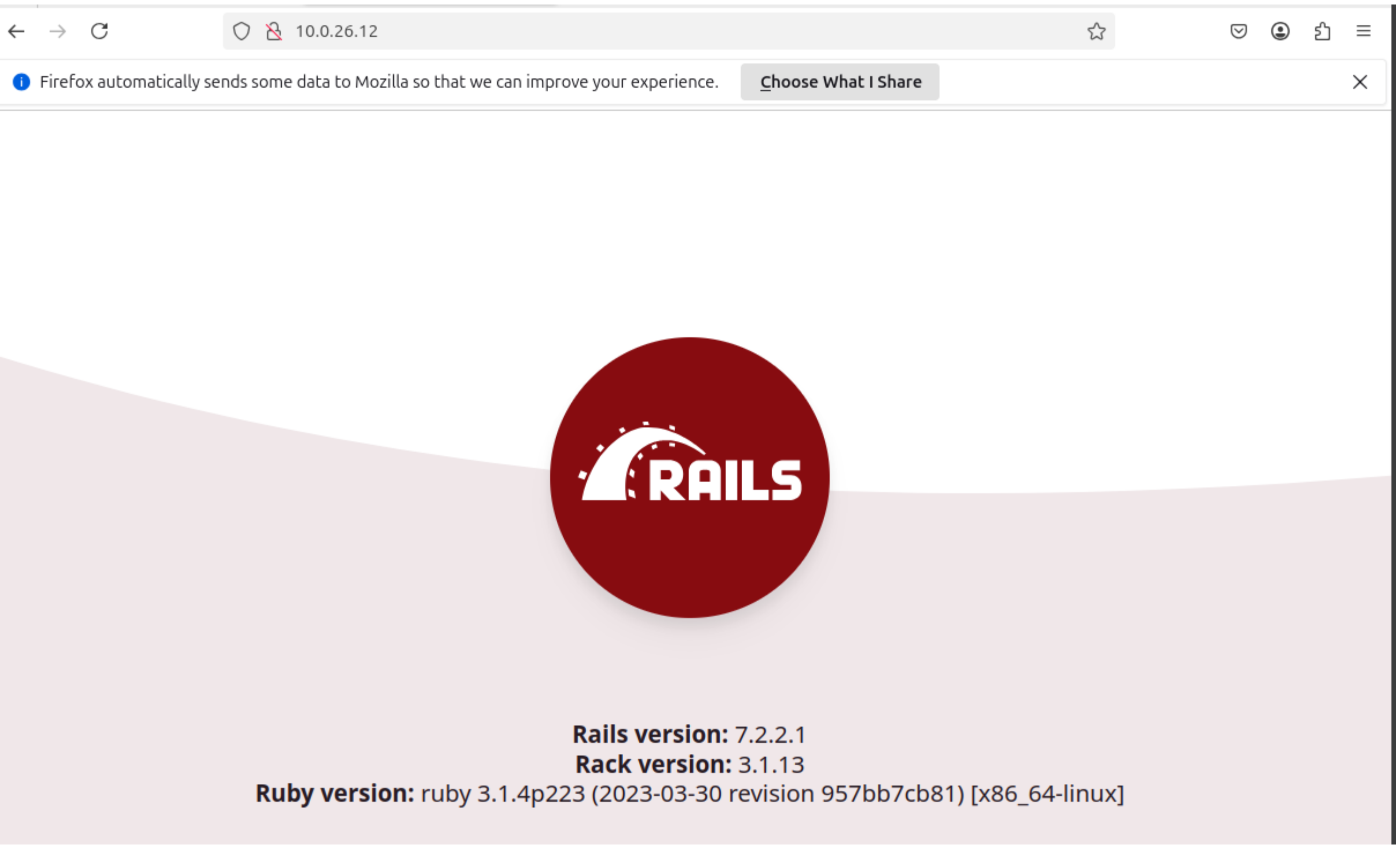
This would let me confirm the app was accessible locally from the server's side

```
vm2-server-daniel@vm2-server-daniel:~$ curl http://localhost_
```

The output of the curl command confirmed the Ruby on Rails welcome page was being shown. I could see Rails, Rack and Ruby version details being shown in the HTML output, This told me that after much difficulty the Rails app was up and running under Apache with the passenger module.

```
3NzEuNDQxMDQ2NCAyLjQ4ODI3NjFsLjEwMTM5MzYuNDQ5OTQwNi0yLjcxMjIwMDEtLjk3MTg3MTctLjAzMzgzNDctLjIxMjE2MTgtLjA2NjA0MjEtLjQ3NTU4NDMtLjA2MTE2MDEtLjU0MTIxOTVjLS4wNDgwMjg
1LS40Nzc0NjAyLS4wODc1ODE0LTEuMDE5OTAwMy0uMTEwMTgzMS0xLjYxODg0NDR6bTMxLjUwNjcyMzktNy42NjE5NjUyaC0xLjUxNDMxMTdjLS45NDAwMjM4IDAtMS4yMzkxMjI0LjQwOTc3Mi0xLjMzNDI5MDE
uNjcwNTM2bC0uMDI4MjI2Ni4wOTUwMzM5Yy0uMDAzMjM3LjAxNDA3My0uMDA1ODI2Ni4wMjcxNTI2LS4wMDc4OTgyLjAzOTA3MzJsLS4wMDgxNTczLjA3MTUyMzktLjAwMDEyOTUgMy45MTUzODY0aDQuMTM2MTA
1MWwtLjAwMTg1MzMtMy45MzQ3Njk0LS4wMTAyNjEyLS4wNjY5OTEzLS4wMjU3MjA3LS4wOTg3MjQxYy0uMDgzNTM1Ny0uMjU5MTUwNy0uMzUwOTEzNS0uNjkxMDY4Ni0xLjIwNTI1NjUtLjY5MTA2ODZ6bS01MC4
0OTIxMjQyLjMzOTAyNSAyLjU5OTE5MTcuOTQ5MjctLjQwODY0MS45NTE1ODM5Yy0uMjEyNDg4Mi40OTk0NjQ0LS40Mjc3Nzk2IDEuMDE0ODE0Mi0uNjAzMzU3NSAxLjQ1Nzc0NWwtLjExODA4NDkuMzAyODcxMi0
yLjU5OTE5MTctLjk0OTI3LjEzMjM2NjItLjM0NTMwMzMuMjU4Mzg4OS0uNjM5MjAzNC4zMDU3MTExLS43MjgxOTc0LjMyNTQ4NzUtLjc1MzYyNDRjLjAzNjI0MzgtLjA4Mjc5NDMuMDcyMzQzNy0uMTY0ODgyMy4
xMDgxMjk3LS4yNDU4NzE2em0zNS40NDUxMjA5LS4xNDM0NDQ5aC0zLjQ1Njg0Mzl2My42NTg4NjczaDMuNDM0Mzk2OGwuMDU0NzEwNi0uMDI1MzkyLjA4NjU5ODQtLjA0ODg5ODMuMTE0NzUzNi0uMDc4NjgyMmM
uMjkyNjQyOC0uMjIxMTQ0OC43MzE2MDcxLS43MTQ5Nzk3LjczMTYwNzEtMS42ODc2ODQ3cy0uNDI4OTg3OC0xLjQ1NjU2MzQtLjcxNDk3OTctMS42NzEwNTczbC0uMTEyMTQ1NS0uMDc2MDc0MS0uMDg0NjMwMy0
uMDQ2OTMwMXptLTE1LjQ0MjY0NTYtLjc2MDYyMTggMS42MjczMjAxIDEuMjg4Mjk1MWMtLjE4MDgxMzQuNzA1MTcyLS4zMTgyMzE1IDEuNDEwMzQ0LS40MTIyNTQ1IDIuMTE1NTE2bC0uMDYyMzgwNi41Mjg4Nzk
tMS44MzA3MzUtMS40NDY1MDY3Yy4xODA4MTM0LS44MTM2Ni4zODQyMjg0LTEuNjQ5OTIxNy42NzgwNS0yLjQ4NjE4MzR6bTQuMDAwNDk1MS02LjMwNTg2NTEgMS4wMTcwNzUgMS41MzY5MTM0Yy0uMzk3Nzg5My4
0MTU4NzA3LS43NjY2NDg1LjgzMTc0MTMtMS4wOTUwMDU1IDEuMjcwNzU2MWwtLjIzODQ5MjguMzMzOTYyMy0xLjA4NDg4MDEtMS42MjczMjAxYy40MDY4My0uNTE5ODM4My44ODE0NjUxLTEuMDM5Njc2NyAxLjQ
wMTMwMzQtMS41MTQzMTE3em0tMTYuMTgyNzkzNi0zLjM0NTA0NjcgMS42MDQ3MTgzIDEuNDAxMzAzNGMtLjQwNjgzLjQyMzc4MTItLjgwMDk0NjUuODcyOTg5NC0xLjE3MjgxNDYgMS4zMjg1NTQybC0uMzY0MDk
4Ny40NTY5Nzc1LTEuNzQwMzI4NC0xLjQ5MTcxMDFjLjUxOTgzODMtLjU2NTA0MTYgMS4wODQ4ODMtMS4xMzAwODMzIDEuNjcyNTIzNS0xLjY5NTEyNDlsLjA4MzgxMTEtLjA3NDA5M2wxLjQ5MTcxMDEgMS43NDAzMjg0
3MTAxYy0uNTI0MzQ3MS40NTIwMzM4LTEuMDA2MzI0My45MTk1MzM2LTEuNDQ2Nzg1NiAxLjQwMzg4MjZsLS4yNjAyNTg0LjI5NTY0OTd6bTcuMzkzNjIzNi02LjU4NDgxMDRjLS45MDQwNjY4LjU2NTA0MTYtMS43OTU4NDg1IDEuMjc3MjM2Ni0yLjY3OTE0NDcgMS45OTcyOTk4bC0uMjI2NDQyLjE4NDU5OTNjLS4wOTA0MDY3
uNDk3MjM2NiAxLjQ5MTcxMDEgMS43NDAzMjg0Yy4yNjAyMjE5LS4yMzkwOTQ5LjUyMTc4MTYtLjQ3MTk0NzguNzg0ODk2LTEuMjMwMzE3OGwuMzk3Nzg1MS4zNDk0MTM0LjA1MjA4NDQtLjk1NjA4MDQuMjExNDE1My0xLjg2OTgxOTZjLjA2MTk1NTEtLjQ2NDM3NTIuMTM5MDMyLTEuMDI4MTgyNi4yMzI4NDIyLTEuNTYwNzEwMjR
2NyAxLjQyNjk4OS0xLjU3NDE1NDNjLTEuMDk3MTI0Ni0uMTM1MTM2LTIuMjk2ODA5Mi0uMjk2MDI2OC0zLjEzOTEyNDktLjQ2NDU4MjMtLjQwMzU0NDguLjc0MDg1NjktLjg4MzY0My0xLjUyOTc0MTQtMS4zNjI1MjE2LTIuMjg4NTc3MmMtLjI0ODY1NTgtLjM5NDA4MzgtLjQ5MDgwOTgtLjc3NzMyOTItLjcyMDcyMjctMS4xNDQ3MDc1
1OTUxNWMtLjQ0MDc2MzIuMjUwMzcyOS0uODc1NjM5Mi41MTE5NzQxLTEuMzA1NjE4OC43ODM0OTM1bC0uNDQ1NzAyNy4yOTA0MjE4LjA5MDQwNjcuMTQxMzk3OGwuODc1ODQyOC4xMzI4MzM3LjI2MTg3MjkgMS45NjYyODUzYy41NzYzODgyLS4zMjk1ODk4IDEuMTgzNjI4Ni0uNjU0NTc3NyAxLjgxMzgwMTktLjk2NDI3NDkuMDE4OTA3LjA0MjE3NjYuMDM3OTA3Ni4wODQ2NjQ3LjA1NzAyNjcuMTI3MjQ3NQ
4NC0uNjc4MDUgMS4zMjgzMzIxLTEuMzgyODg1NGwxLjE0MDIwNDUtMS4yMjk0NzIxLjE2NzUzMjctLjE3MzQ0MDUuNzEwNzU3NS0uNzMwNDQxNGwtLjk5MzQ4NjItLjgzMjU0NDJjLS4yNzU1MzA2LjI3Mzk2MDMtLjU1MDk4OTkuNTQ4NTgwMi0uODIzNzY0Ni44MjMzOTg1bC0uNDA2OTIyMi40MTUxOTk3LS4yMjAzNjkxLS4yMjQ2OTg4em0tMjEuMDU5NjM1Ni0xNC4xMzE5NTEyYzAtLjI5Mzk4MzIuMDE2MDE4OC0uNTg1MzE1Mi4wNDY2Nzk0LS44NzQ3MzUz
uMDk5MzA0OC4zMjg0OTE1LjE4ODk2OS42NjEzODg2LjI2NDY1MDYuOTk4MzY5OWwxLjk0NzU1MDEtLjQ1MDA4MDdjLS4wODEwMDk0LS4zNTg3NzA0LS4xNzQ1MjE1LS43MTQzNzI0LS4yODEyMzI2LTEuMDY0OTQ4NmwtLjMyNzcwODMtLjk5MDQwNTgtMS44NjgyMjczLjcwMzIwNDUuMjY3NjEzMS43NjU0ODM5LjQ0MTM1NjMtLjEyNTE0OTh6bS0uNjg1ODE0MS4yODk3MjM2Yy0uMTcxMzEzNC4zMzkzNzM1LS4zNDY1ODMuNjY1OTg1Mi0uNTI0ODgwNi45ODI5MjE1
uNTU3NTA3OHptLTguNTQzNDMwMS0yLjgyNTIwODQuODMzMzAxNy0yLjAxNjcwNzdjLS43OTYyNzU3LS4zMTQ0NzY3LTEuNjEyNDM3NC0uNTkyOTI0NC0yLjQ1MTkxMDktLjgwNTgxNTRsLS4zMzU1MDcyLS4wNzk0NTU1LS40NzA4NTYyIDEuOTEyNDUwMy45NjQ1MzY4LjI0NDU3MjEuMTE4OTQ5Mi4wMzIzMzk1YzEuMDkxMzkwNy4zMTE1MzAzIDIuMTQwNzcxMi43MjU2ODA3IDMuMTQ5NDMxNC4yNDg0MjkyLjQ1NzY1MzgtLjcxMjkyOTkgMS40NjI1MjM1LTEuNDI1MDgyM
zMTMtLjQyOTM0MTctMS4zMzM4MDI4LS44NTkzMjIzLTIuMDQyMDMxOTRjLjE2NjI0NjQtLjU1ODU2OC45MzgwNTQ4LS4xMDgyMjY3IDEuNTg3NjM5OC0xLjE4Njc0NDR6bS0xMy4yNzc3NTAxLTE0LjAwMTMxNzVjLjAxMjgwODQuMDUyOTA2LS4wNDIzNTEzLjE3NzUzNTguMDI5NDMwMi4yOTU3MDMzbC0uNjg1NTIxLjU1NDg3MDYuNzk5MjE1Ny4zMDEzNDM0LjI4MjE1NTgtLjQxODEzNzQ3LjUwNDU3NTcuNjEzMjcyOGwtLjQ5NzA3ODcuNDE0MDQ4MmMuMDQ3NDg1NS4wMjcwNjU5LjA5MzU5Mjk
zMTMtLjQyOTM0MTctMS4zMzM4MDI4LjQ5MDA1MjgtLjk4NDQzMzctMS40NjE4NTI1LTEuOTg2MDA1MS0yLjYxNjc5ODYtMi45MTc3NzQ2bC0uMzQ0MzQ5OC0uMjczMTUzLTEuMTg2NDc2OSAxLjYwNDcxODNjLS45NzE5ODA2LS43NzgxMTk4LTEuOTc5NDMxLTEuNDU0NTA0OS0zLjAyMTkwNDMtMi4wMTQ5MjQ1bC0uNDI1NjgzLS4yMTk1MDgtLjg3OTA4NjYgMS44Mjg1NTIzLjk4MzExODIuNTI5NTkxNC4wODg4NTA4LjA1NjU3NzljLjkxODQzMTMuNTQ3MzE2OSAxLjg0MjIwMzYgMS4xOTgyMjQ0IDIuNzU0NDkzMyAxLjk1MDA1ODR6bTEuMjgxMjk4NS0xLjI2OTU0ODdjLS41NDUzNzM2LS40NDMyMzQ4LTEuMTE1NDIzOC0uODQ2NTU1NC0xLjcwMTE4OTgtMS4yMDU2Njc3
zMTMtLjQyOTM0MTctMS4zMzM4MDI4LjQ5MDA1MjgtLjk4NDQzMzctMS40NjE4NTI1LTEuOTg2MDA1MS0yLjYxNjc5ODYtMi45MTc3NzQ2bC0uMzQ0MzQ5OC0uMjczMTUzLTEuMTg2NDc2OSAxLjYwNDcxODNjLjE2MjA3OTQuMTE5ODE2LjMyMDY2MTIuMjQzMTk2Ny40NzU4MTI4LjM3MDExMzdsLjM1MDAwNjQuMjk2MjE4OXptLTIuNTQ0OTU4Ny0yLjA0ODA5MDRjLS42OTk5MTI3LS40MzY1OTU1LTEuNDQyMjc0Ny0uODE0MDEzMi0yLjIxNzIyMzctMS4xMjA0MzcybC0uMzk0MDczMy0uMTQ3MjM1NC0uNjE0NzY1NyAxLjg5MjE0MDYuOTU3MzI3Ny4zODQ1NzA4LjA1MDQ1NTEuMDE3NjUyNmMuNjg4OTc5MS4yNzIzMDY4IDEuMzQ5MzEwNC42MDY1MDAzIDEuOTcyMDc4NS45OTM0OTEybC4zMzIyMzg5LjIyMDIxOTF6bS00LjQ4NzMzNzgtMi4wMjI3Mzk4Yy0uOTEzMjc3Ni0uMjMxNzM5LTEuODUyMjk1Ni0uMzk4NzA0LTIuODEyOTQ2OC0uNDg5ODA1NA
zMTMtLjQyOTM0MTctMS4zMzM4MDI4LjQ5MDA1MjgtLjk4NDQzMzctMS40NjE4NTI1LTEuOTg2MDA1MS0yLjYxNjc5ODYtMi45MTc3NzQ2bC0uMzQ0MzQ5OC0uMjczMTUzLTEuMTg2NDc2OSAxLjYwNDcxODNjLjE2MjA3OTQuMTE5ODE2LjMyMDY2MTIuMjQzMTk2Ny40NzU4MTI4LjM3MDExMzdsLjM1MDAwNjQuMjk2MjE4OXptLTIuNTQ0OTU4Ny0yLjA0ODA5MDRjLS42OTk5MTI3LS40MzY1OTU1LTEuNDQyMjc0Ny0uODE0MDEzMi0yLjIxNzIyMzctMS4xMjA0MzcybC0uMzk0MDczMy0uMTQ3MjM1NC0uNjE0NzY1NyAxLjg5MjE0MDYuOTU3MzI3Ny4zODQ1NzA4LjA1MDQ1NTEuMDE3NjUyNmMuNjg4OTc5MS4yNzIzMDY4IDEuMzQ5MzEwNC42MDY1MDAzIDEuOTcyMDc4NS45OTM0OTEybC4zMzIyMzg5LjIyMDIxOTF6bTE0LjA0OTcxNzMtMTMuNzM4MDY1MmMtLjQxNDU5NDEuMDc4OTAxNS0uODYyNDc5NS4xNjE1MjQ1LTEuMzQ2MTQwMi4yMzk4ODE5LjA3ODUyMzQuODMzNDg2NS4xNTQ4ODU4IDEuNjk4MTQ3NS4yMjg5OTAzIDIuNTk0MTcxOWwtMS45Njg5MzMyLjE1NzQ2NDJjLS4wNTk0NzQ1LS44NDExNDctLjEyMjM3MDMtMS42NDk3MDI4LS4xODk1NzI0LTIuNDIzMjg4Mm
zMTMtLjQyOTM0MTctMS4zMzM4MDI4LS44NTkzMjIzLTIuMDQyMDMxOTRjLjE2NjI0NjQtLjU1ODU2OC45MzgwNTQ4LS4xMDgyMjY3IDEuNTg3NjM5OC0xLjE4Njc0NDR6bS0xMy4yNzc3NTAxLTE0LjAwMTMxNzVjLjAxMjgwODQuMDUyOTA2LS4wNDIzNTEzLjE3NzUzNTguMDI5NDMwMi4yOTU3MDMzbC0uNjg1NTIxLjU1NDg3MDYuNzk5MjE1Ny4zMDEzNDM0LjI4MjE1NTgtLjQxODEzNzQ3LjUwNDU3NTcuNjEzMjcyOGwtLjQ5NzA3ODcuNDE0MDQ4MmMuMDQ3NDg1NS4wMjcwNjU5LjA5MzU5MjkuMDU1MTA5MS4xMzkwNzYxLjA4NDAwNjRsMS4wNjMyNzAxLTEuNjg4ODk1MWMuNjg0MzExNy40MDg1MjU1IDEuMzQxODM1OC44NTc3NzQ4IDEuOTY2NjA4NCAxLjM0NTQ1NzJsLjM3MzAwNjIuMzAwNzI1M3ptMTQuMDQ5NzE3My0xMy43MzgwNjUyYy0uNDE0NTk0MS4wNzg5MDE1LS44NjI0Nzk1LjE2MTUyNDUtMS4zNDYxNDAyLjIzOTg4MTl6
zMTMtLjQyOTM0MTctMS4zMzM4MDI4LS44NTkzMjIzLTIuMDQyMDMxOTRjLjE2NjI0NjQtLjU1ODU2OC45MzgwNTQ4LS4xMDgyMjY3IDEuNTg3NjM5OC0xLjE4Njc0NDR6bS0xMy4yNzc3NTAxLTE0LjAwMTMxNzVjLjAxMjgwODQuMDUyOTA2LS4wNDIzNTEzLjE3NzUzNTguMDI5NDMwMi4yOTU3MDMzbC0uNjg1NTIxLjU1NDg3MDYuNzk5MjE1Ny4zMDEzNDM0LjI4MjE1NTgtLjQxODEzNzQ3LjUwNDU3NTcuNjEzMjcyOGwtLjQ5NzA3ODcuNDE0MDQ4MmMuMDQ3NDg1NS4wMjcwNjU5LjA5MzU5MjkuMDU1MTA5MS4xMzkwNzYxLjA4NDAwNjRsMS4wNjMyNzAxLTEuNjg4ODk1MWMuNjg0MzExNy40MDg1MjU1IDEuMzQxODM1OC44NTc3NzQ4IDEuOTY2NjA4NCAxLjM0NTQ1NzJsLjM3MzAwNjIuMzAwNzI1M3ptMTQuMDQ5NzE3My0xMy43MzgwNjUyYy0uNDE0NTk0MS4wNzg5MDE1LS44NjI0Nzk1LjE2MTUyNDUtMS4zNDYxNDAyLjIzOTg4MTl6
zMTMtLjQyOTM0MTctMS4zMzM4MDI4LS44NTkzMjIzLTIuMDQyMDMxOTRjLjE2NjI0NjQtLjU1ODU2OC45MzgwNTQ4LS4xMDgyMjY3IDEuNTg3NjM5OC0xLjE4Njc0NDR6bS0xMy4yNzc3NTAxLTE0LjAwMTMxNzVjLjAxMjgwODQuMDUyOTA2LS4wNDIzNTEzLjE3NzUzNTguMDI5NDMwMi4yOTU3MDMzbC0uNjg1NTIxLjU1NDg3MDYuNzk5MjE1Ny4zMDEzNDM0LjI4MjE1NTgtLjQxODEzNzQ3LjUwNDU3NTcuNjEzMjcyOGwtLjQ5NzA3ODcuNDE0MDQ4MmMuMDQ3NDg1NS4wMjcwNjU5LjA5MzU5MjkuMDU1MTA5MS4xMzkwNzYxLjA4NDAwNjRsMS4wNjMyNzAxLTEuNjg4ODk1MWMuNjg0MzExNy40MDg1MjU1IDEuMzQxODM1OC44NTc3NzQ4IDEuOTY2NjA4NCAxLjM0NTQ1NzJsLjM3MzAwNjIuMzAwNzI1M3ptMTQuMDQ5NzE3My0xMy43MzgwNjUyYy0uNDE0NTk0MS4wNzg5MDE1LS44NjI0Nzk1LjE2MTUyNDUtMS4zNDYxNDAyLjIzOTg4MTl6
```

At the end when testing on my DHCP client machine I went to the browser and searched **10.0.26.12** which was the IP of my Server hosting Apache. I was able to see the Ruby on Rails welcome page, this confirmed to me that my was successfully being hosted through my Apache server machine on the network.



Firefox automatically sends some data to Mozilla so that we can improve your experience.

Rails version: 7.2.2.1
Rack version: 3.1.13
Ruby version: ruby 3.1.4p223 (2023-03-30 revision 957bb7cb81) [x86_64-linux]

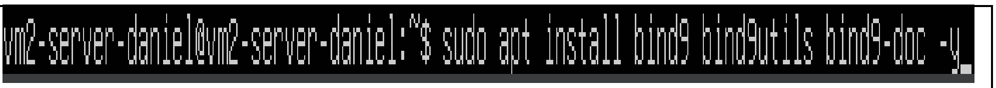
Installing & Configuring BIND9 (Master & Slave DNS Server) [iii]

Now that Apache is set up on the VM 2 Server I will next move on to setting up a Master Server on VM2 as well as a slave server in VM3.[iv]

To begin configuring DNS I installed BND9 DNS server on VM2 (my main server)

Sudo apt install bind9 bind9utils bind9-doc-y

This installed BIND9 as well as it’s utilities and documentation so I can properly manage DNS zones and configs

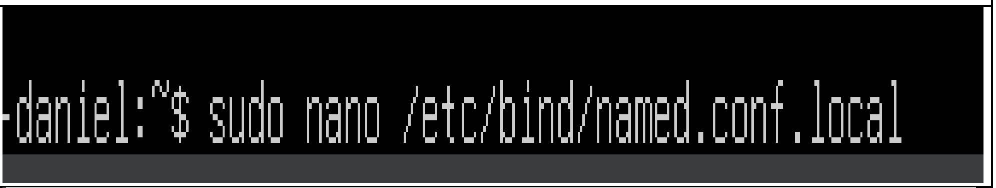


```
vm2-server-daniel@vm2-server-daniel:~$ sudo apt install bind9 bind9utils bind9-doc -y_
```

This server (VM2) was configured as the Primary DNS Server (Master)

After installing BIND9 , I opened the DNS local config file:

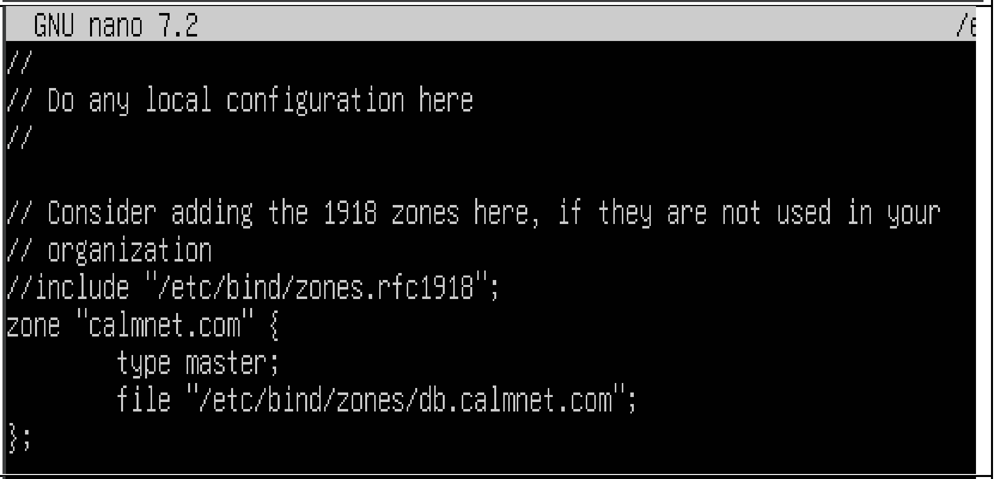
Sudo nano /etc/bind/named.conf.local



```
-daniel:~$ sudo nano /etc/bind/named.conf.local
```

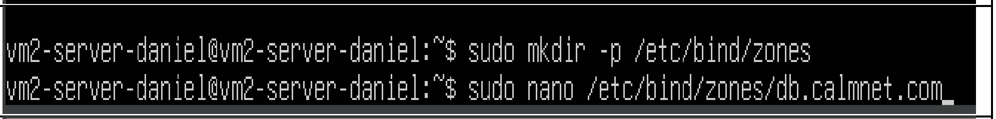
Inside the named.conf.local file I added a forward lookup zone for **calmnet.com** and set the zone type to **Master.** This just made it so the DNS server holds the records for the calmnet.com domain as the main server.

I then pointed the zone data file to the path **/etc/bind/zones/db.calmnet.com**



```
  GNU nano 7.2                                                          /e
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "calmnet.com" {
        type master;
        file "/etc/bind/zones/db.calmnet.com";
};
```

Here I created the necessary zone file:



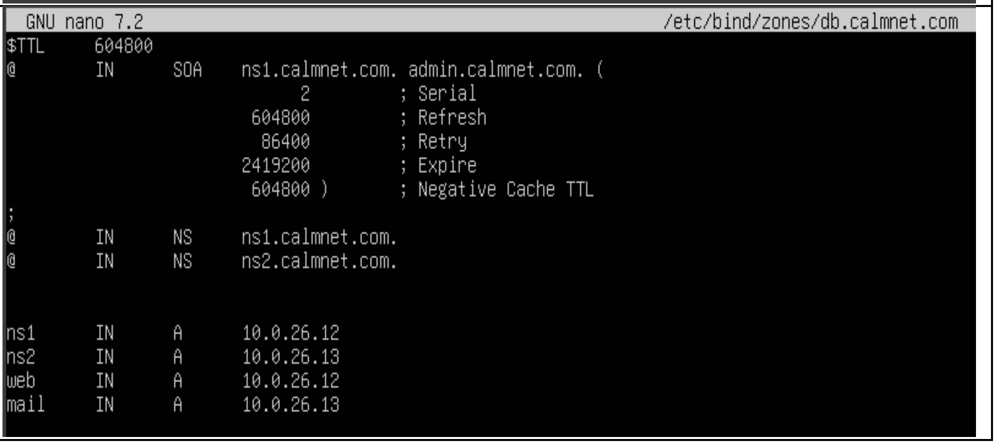
```
vm2-server-daniel@vm2-server-daniel:~$ sudo mkdir -p /etc/bind/zones
vm2-server-daniel@vm2-server-daniel:~$ sudo nano /etc/bind/zones/db.calmnet.com_
```

In the new Zone file, I set up the DNS forward zone for my calmnet.com domain as follows:

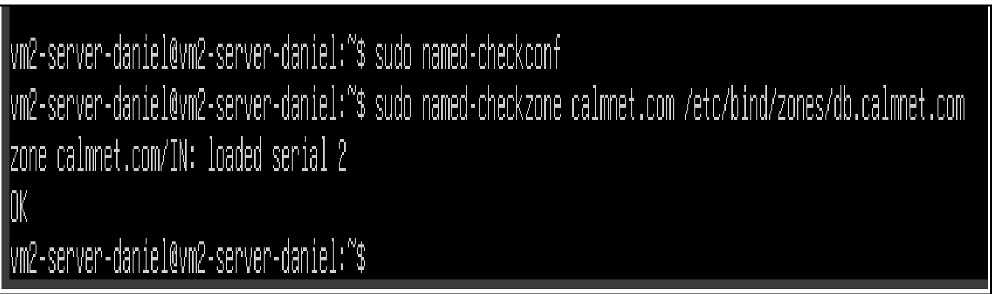
Start of Authority (SOA) = ns1.calmnet.com

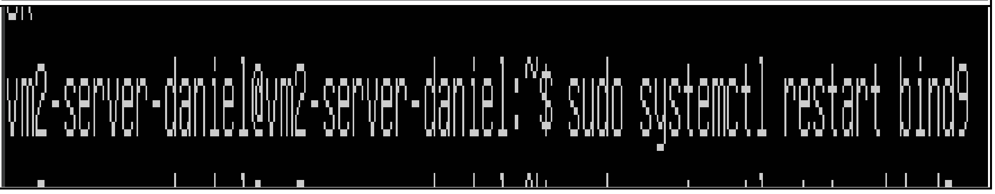
Admin contact = admin.calmnet.com

NS records = ns1 and ns2



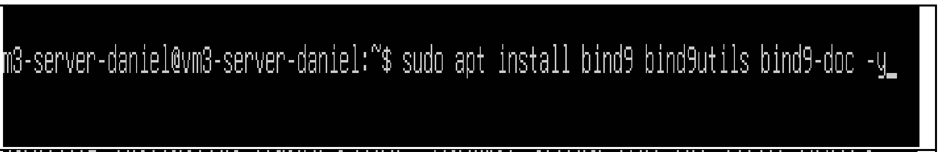
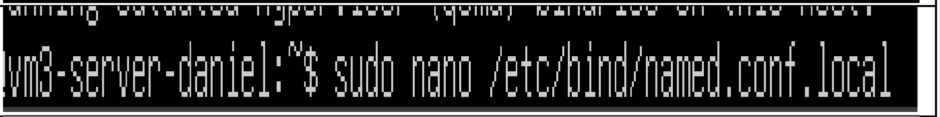
```
  GNU nano 7.2                                          /etc/bind/zones/db.calmnet.com
$TTL    604800
@       IN      SOA     ns1.calmnet.com. admin.calmnet.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.calmnet.com.
@       IN      NS      ns2.calmnet.com.


ns1     IN      A       10.0.26.12
ns2     IN      A       10.0.26.13
web     IN      A       10.0.26.12
mail    IN      A       10.0.26.13
```

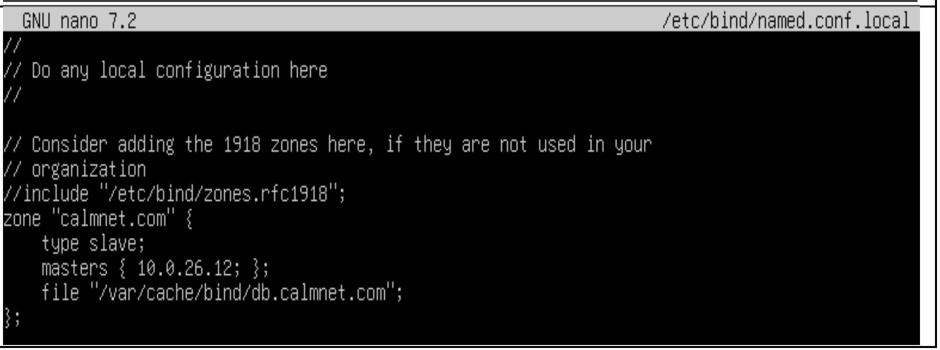
Once my zone file was done, I ran: **named-checkconf** and **named-checkzone calmnet.com /etc/bind/zones/db.calmnet.com**
This let me see if my config files were okay.
	
	`vm2-server-daniel@vm2-server-daniel:~$ sudo named-checkconf`
	`vm2-server-daniel@vm2-server-daniel:~$ sudo named-checkzone calmnet.com /etc/bind/zones/db.calmnet.com`
	`zone calmnet.com/IN: loaded serial 2`
	`OK`
	`vm2-server-daniel@vm2-server-daniel:~$`

After validating all this I restarted the BIND9 service to apply the changes so the new zone data and settings were loaded into the DNS server	 `vm2-server-daniel@vm2-server-daniel:~$ sudo systemctl restart bind9`

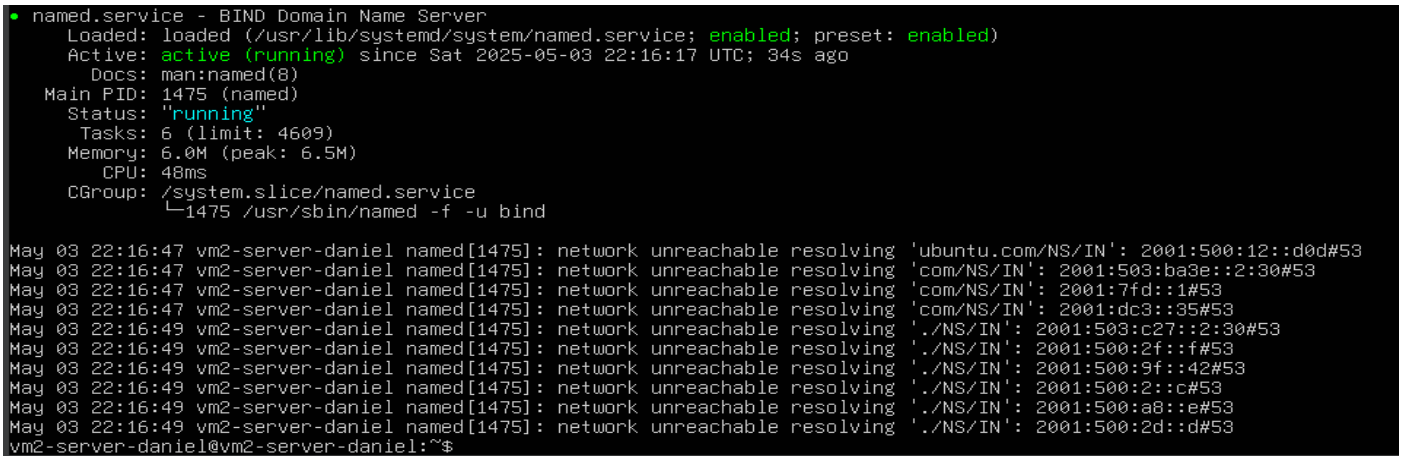
The Primary DNS server was now successfully set up on VM2

Next I was ready to set up the Slave (Secondary DNS Server) on VM 3.

To set up my secondary DNS server on VM3 I installed bind9 with it's utilities and documentation	 `m3-server-daniel@vm3-server-daniel:~$ sudo apt install bind9 bind9utils bind9-doc -y_`
On VM3 I edited the named.conf.local file using: **Sudo nano /etc/bind/named.conf.local**	 `vm3-server-daniel:~$ sudo nano /etc/bind/named.conf.local`

Here I added the slave zone entry for calmnet.com and set the type to slave, I also gave it the zone file location and IP address of the Primary/Master DNS Server (**10.0.26.12)** so it can access the zone data	
	`GNU nano 7.2                    /etc/bind/named.conf.local`
	`//`
	`// Do any local configuration here`
	`//`
	
	`// Consider adding the 1918 zones here, if they are not used in your`
	`// organization`
	`//include "/etc/bind/zones.rfc1918";`
	`zone "calmnet.com" {`
	`    type slave;`
	`    masters { 10.0.26.12; };`
	`    file "/var/cache/bind/db.calmnet.com";`
	`};`

After saving the zone file config I restarted BIND on server 3 this time using **sudo systemctl restart bind9** , I then ran **sudo systemctl status bind9** to confirm that it was active and running. The output showed me that DNS was up and running correctly confirming that my Primary and Secondary DNS servers were set up correctly.

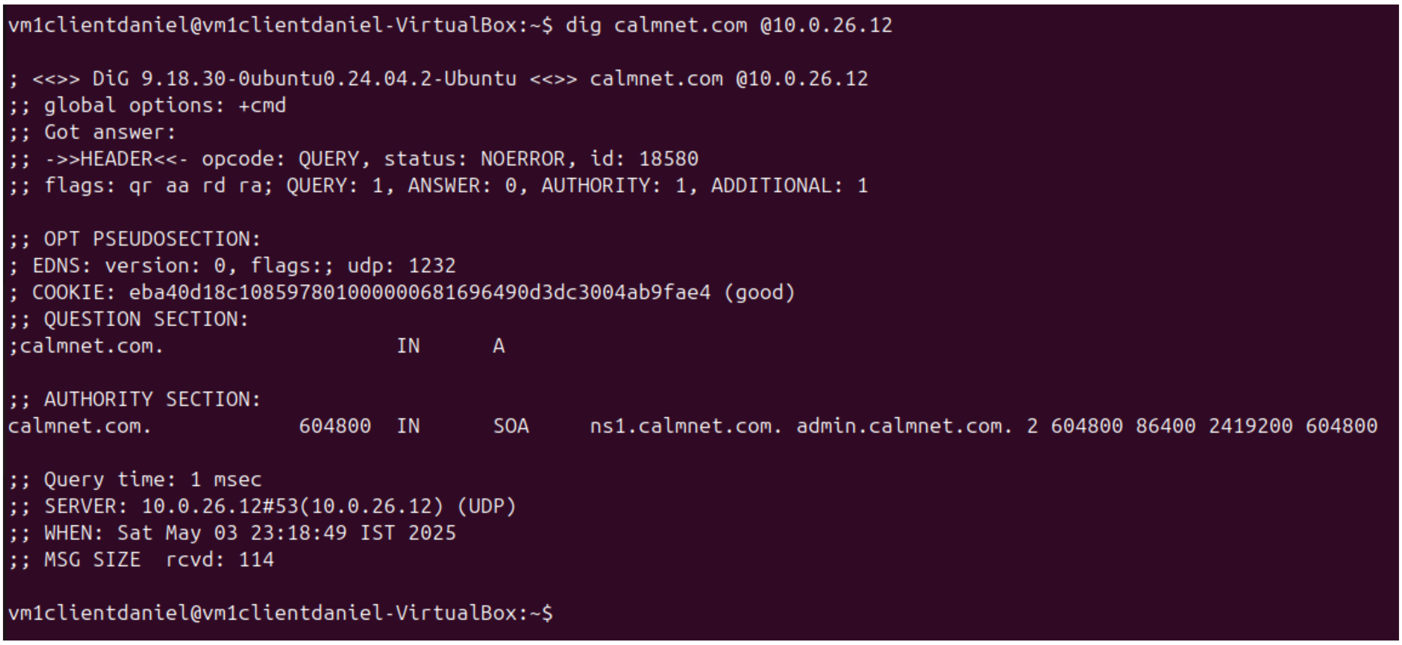


```
● named.service - BIND Domain Name Server
     Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-05-03 22:16:17 UTC; 34s ago
       Docs: man:named(8)
   Main PID: 1475 (named)
     Status: "running"
      Tasks: 6 (limit: 4609)
     Memory: 6.0M (peak: 6.5M)
        CPU: 48ms
     CGroup: /system.slice/named.service
             └─1475 /usr/sbin/named -f -u bind

May 03 22:16:47 vm2-server-daniel named[1475]: network unreachable resolving 'ubuntu.com/NS/IN': 2001:500:12::d0d#53
May 03 22:16:47 vm2-server-daniel named[1475]: network unreachable resolving 'com/NS/IN': 2001:503:ba3e::2:30#53
May 03 22:16:47 vm2-server-daniel named[1475]: network unreachable resolving 'com/NS/IN': 2001:7fd::1#53
May 03 22:16:47 vm2-server-daniel named[1475]: network unreachable resolving 'com/NS/IN': 2001:dc3::35#53
May 03 22:16:49 vm2-server-daniel named[1475]: network unreachable resolving './NS/IN': 2001:503:c27::2:30#53
May 03 22:16:49 vm2-server-daniel named[1475]: network unreachable resolving './NS/IN': 2001:500:2f::f#53
May 03 22:16:49 vm2-server-daniel named[1475]: network unreachable resolving './NS/IN': 2001:500:9f::42#53
May 03 22:16:49 vm2-server-daniel named[1475]: network unreachable resolving './NS/IN': 2001:500:2::c#53
May 03 22:16:49 vm2-server-daniel named[1475]: network unreachable resolving './NS/IN': 2001:500:a8::e#53
May 03 22:16:49 vm2-server-daniel named[1475]: network unreachable resolving './NS/IN': 2001:500:2d::d#53
vm2-server-daniel@vm2-server-daniel:~$
```

Now that this was finished I switched back to internal network using my static server IP's and used the DHCP client to test if I could query the primary and secondary DNS servers, I first attempted this on the Primary (Master) DNS Server using:

dig calmnet.com @10.0.26.12

Working DNS (Primary) On Client :

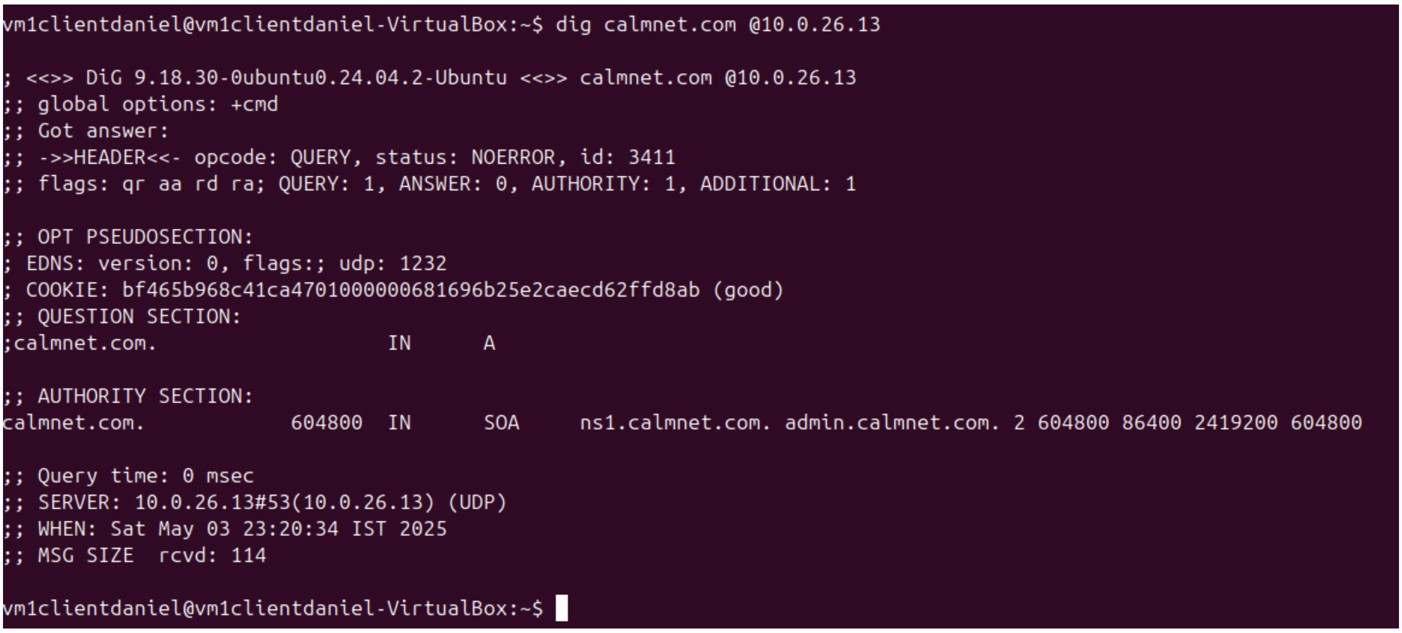


```
vm1clientdaniel@vm1clientdaniel-VirtualBox:~$ dig calmnet.com @10.0.26.12

; <<>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <<>> calmnet.com @10.0.26.12
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18580
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: eba40d18c108597801000000681696490d3dc3004ab9fae4 (good)
;; QUESTION SECTION:
;calmnet.com.                   IN      A

;; AUTHORITY SECTION:
calmnet.com.            604800  IN      SOA     ns1.calmnet.com. admin.calmnet.com. 2 604800 86400 2419200 604800

;; Query time: 1 msec
;; SERVER: 10.0.26.12#53(10.0.26.12) (UDP)
;; WHEN: Sat May 03 23:18:49 IST 2025
;; MSG SIZE  rcvd: 114

vm1clientdaniel@vm1clientdaniel-VirtualBox:~$
```

I carried out the same test again but this time for the secondary (slave) DNS server:

dig calmnet.com @10.0.26.13

Working DNS (Secondary) On Client :



```
vm1clientdaniel@vm1clientdaniel-VirtualBox:~$ dig calmnet.com @10.0.26.13

; <<>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <<>> calmnet.com @10.0.26.13
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3411
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: bf465b968c41ca4701000000681696b25e2caecd62ffd8ab (good)
;; QUESTION SECTION:
;calmnet.com.                   IN      A

;; AUTHORITY SECTION:
calmnet.com.            604800  IN      SOA     ns1.calmnet.com. admin.calmnet.com. 2 604800 86400 2419200 604800

;; Query time: 0 msec
;; SERVER: 10.0.26.13#53(10.0.26.13) (UDP)
;; WHEN: Sat May 03 23:20:34 IST 2025
;; MSG SIZE  rcvd: 114

vm1clientdaniel@vm1clientdaniel-VirtualBox:~$ 
```

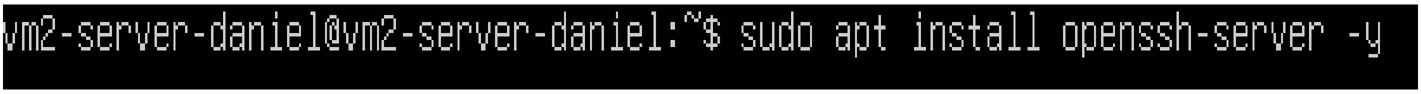
With this I have now successfully set up and configured primary and secondary DNS servers using my 2 server VM's.

SSH Server Install & Configuration:[v]

To allow remote access to my server I installed OpenSSH server package by running:

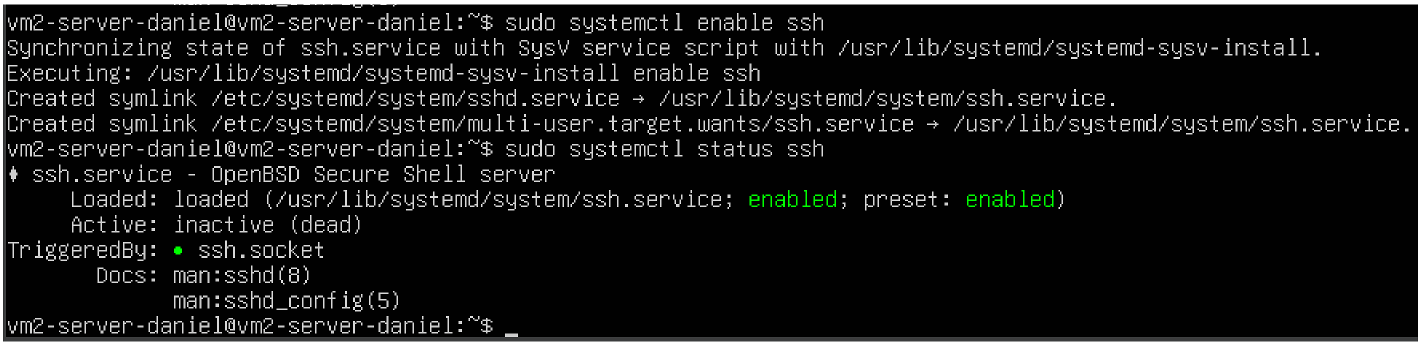
sudo apt install openssh-server -y

This enabled SSH access



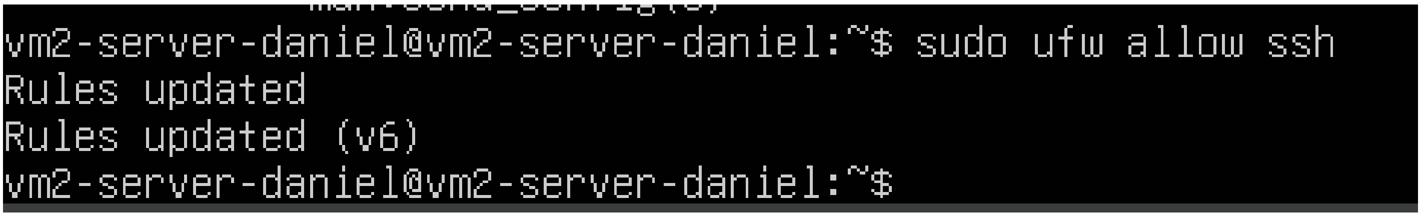
```
vm2-server-daniel@vm2-server-daniel:~$ sudo apt install openssh-server -y
```

After this I enabled SSH as a service by running: **sudo systemctl enable ssh** , I then checked the status and I was shown that it was enabled



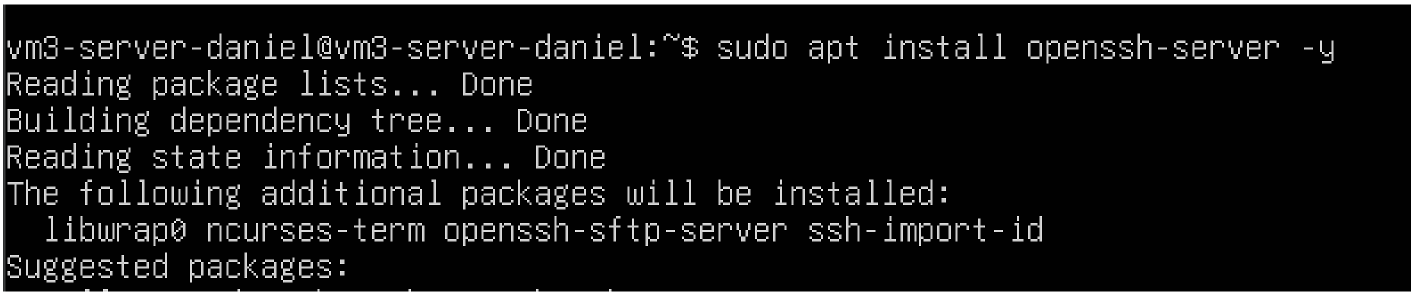
```
vm2-server-daniel@vm2-server-daniel:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /usr/lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/lib/systemd/system/ssh.service.
vm2-server-daniel@vm2-server-daniel:~$ sudo systemctl status ssh
○ ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: inactive (dead)
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
vm2-server-daniel@vm2-server-daniel:~$ _
```

Next I allowed SSH through the firewall[vi] with: **sudo ufw allow ssh**



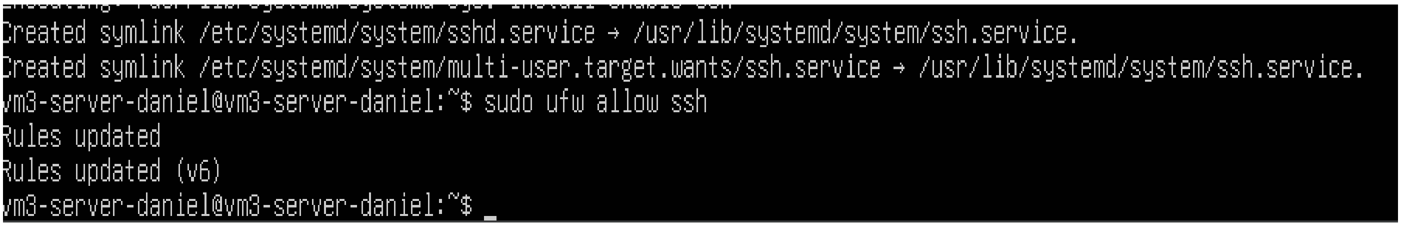
```
vm2-server-daniel@vm2-server-daniel:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
vm2-server-daniel@vm2-server-daniel:~$
```

Next I repeated the setup process on VM3 so both servers were running SSH as a service: **sudo apt install openssh-server -y** , I now enabled it on VM3 like I did on VM2 and checked it's status.



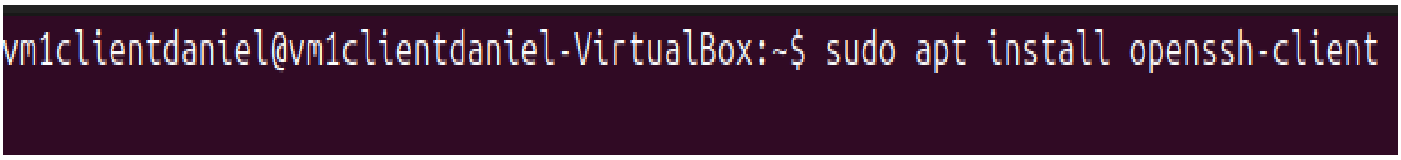
```
vm3-server-daniel@vm3-server-daniel:~$ sudo apt install openssh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libwrap0 ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
```

On VM3 I then allowed SSH through the firewall like I did previously for VM2.



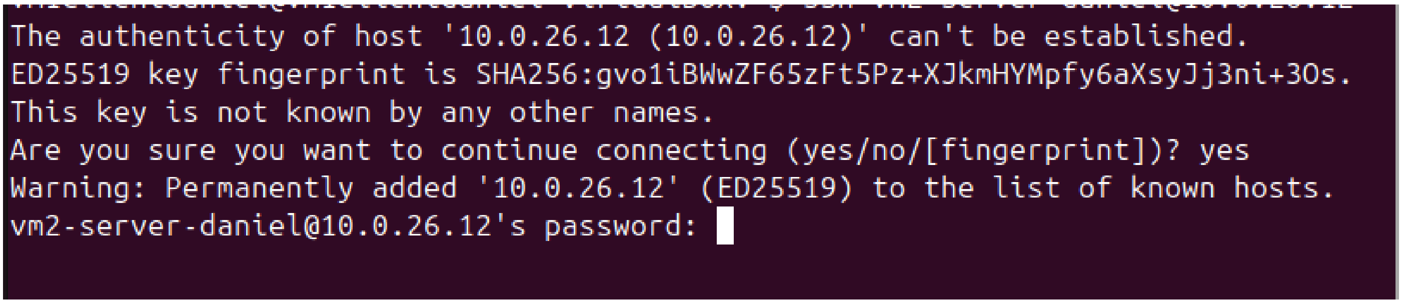
```
Created symlink /etc/systemd/system/sshd.service → /usr/lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/lib/systemd/system/ssh.service.
vm3-server-daniel@vm3-server-daniel:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
vm3-server-daniel@vm3-server-daniel:~$ _
```

While VM1 was on NAT I then installed openssh so I could try and connect it to both servers using SSH.



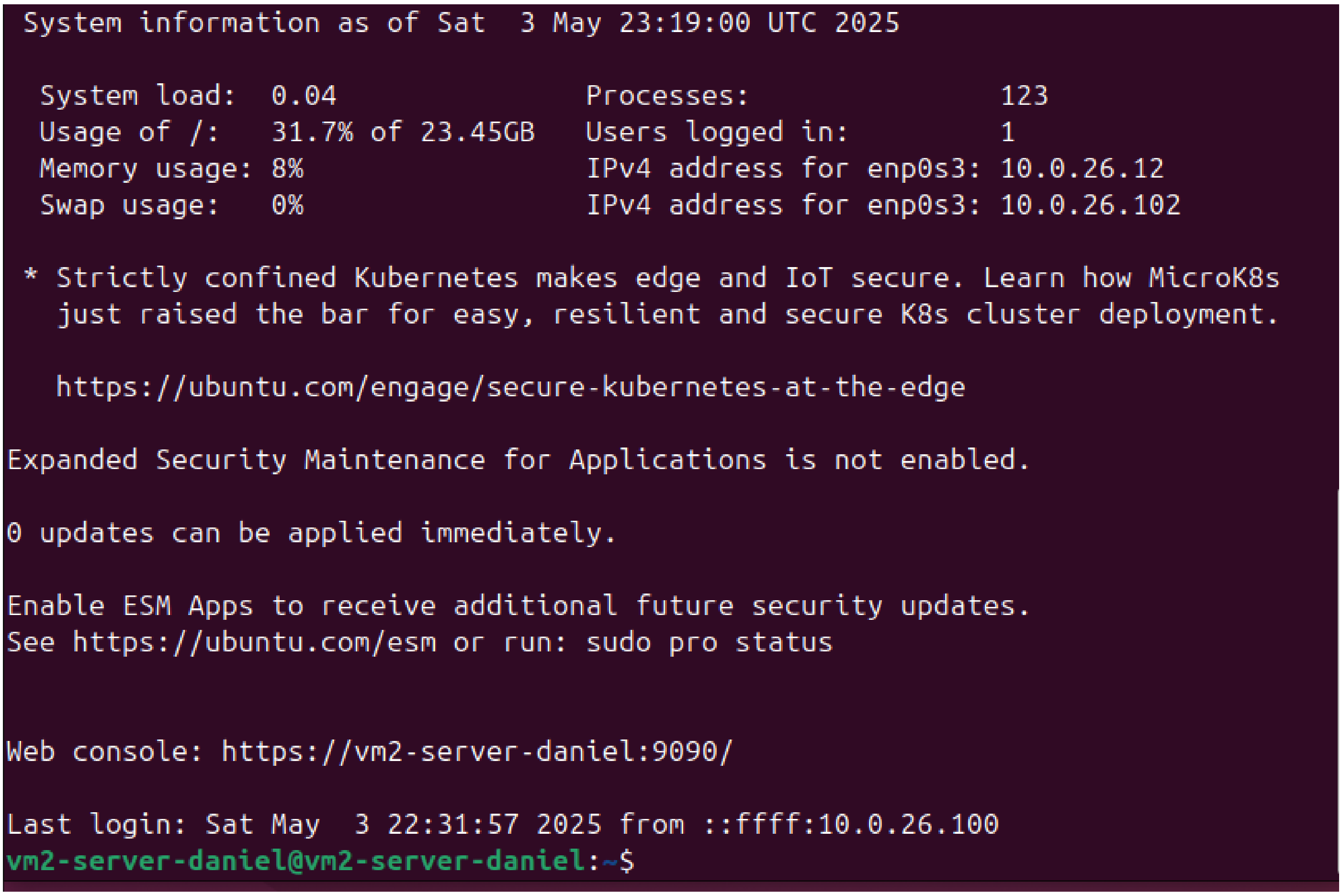
```
vm1clientdaniel@vm1clientdaniel-VirtualBox:~$ sudo apt install openssh-client
```

I then switched vm1 client back to internal network to begin testing SSH, Once I was on internal network I connected to the VM 2server using: **ssh <u>vm2-server-daniel@10.0.26.12</u>** , I accepted the server’s key and entered the password for the user of server 2 on my client machine.



```
The authenticity of host '10.0.26.12 (10.0.26.12)' can't be established.
ED25519 key fingerprint is SHA256:gvo1iBWwZF65zFt5Pz+XJkmHYMpfy6aXsyJj3ni+3Os.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.26.12' (ED25519) to the list of known hosts.
vm2-server-daniel@10.0.26.12's password:
```

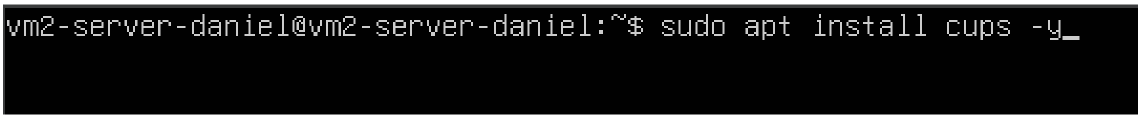
I successfully logged in to VM2 through SSH on my VM1 client confirming that SSH was working correctly.



```
 System information as of Sat  3 May 23:19:00 UTC 2025

  System load:  0.04               Processes:             123
  Usage of /:   31.7% of 23.45GB   Users logged in:       1
  Memory usage: 8%                 IPv4 address for enp0s3: 10.0.26.12
  Swap usage:   0%                 IPv4 address for enp0s3: 10.0.26.102

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Web console: https://vm2-server-daniel:9090/

Last login: Sat May  3 22:31:57 2025 from ::ffff:10.0.26.100
vm2-server-daniel@vm2-server-daniel:~$
```

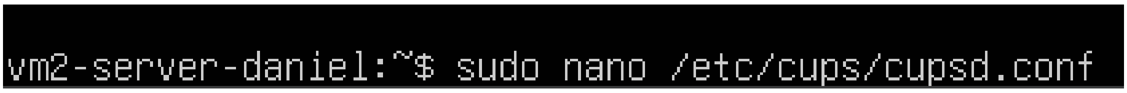
Installing CUPS Print Server:[vii]

To set up printing on my server I installed CUPS printing service using the command **sudo apt install cups -y**



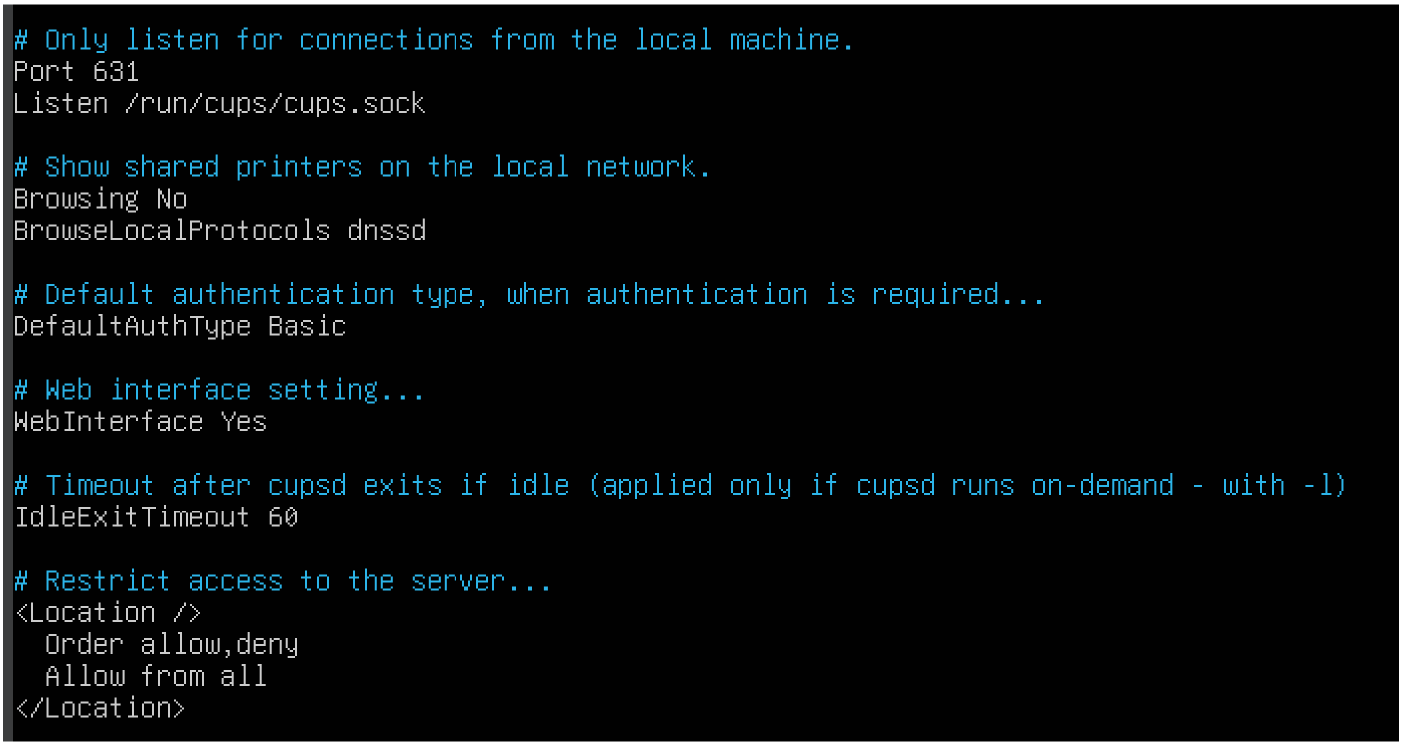
```
vm2-server-daniel@vm2-server-daniel:~$ sudo apt install cups -y_
```

To allow remote access I had to edit the configuration file using: **sudo nano /etc/cups/cupsd.conf** , as this file controls access to the CUPS service.



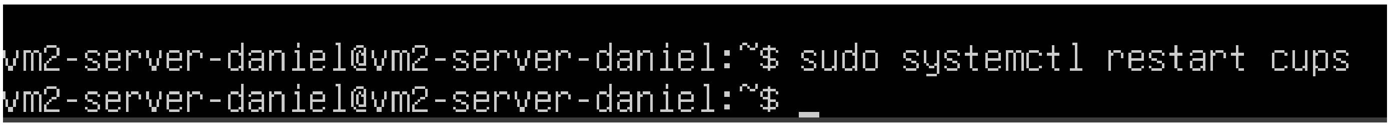
```
vm2-server-daniel:~$ sudo nano /etc/cups/cupsd.conf
```

In the config file I enabled the CUPS web interface by setting WebInterface to yes, I also changed the config file to include "Allow from all" so any client on my network can access the printing service. I also made sure CUPS was set up to listen on port 631



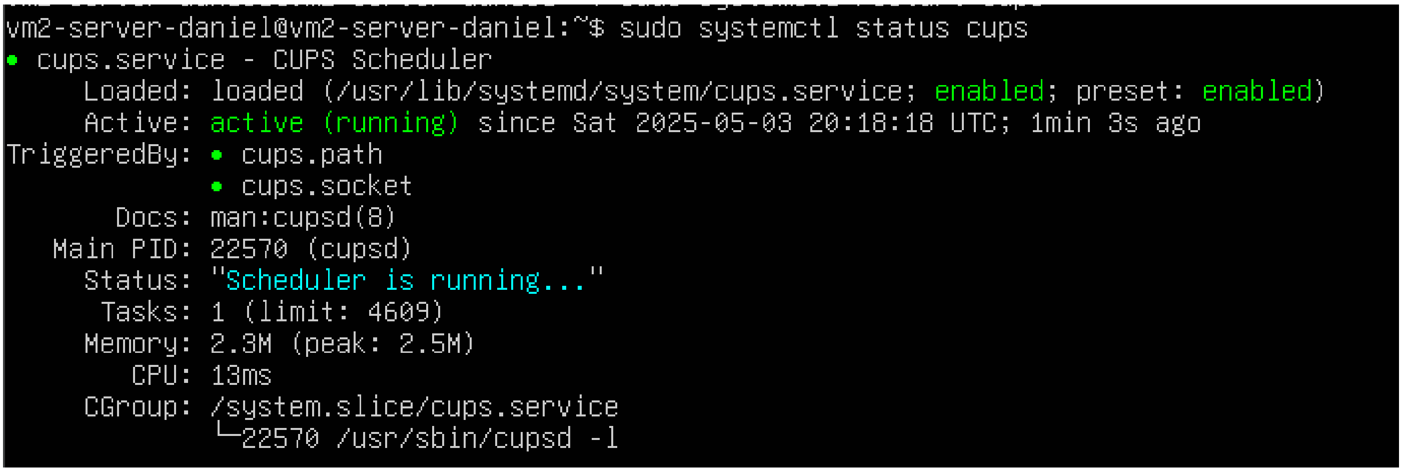
```
# Only listen for connections from the local machine.
Port 631
Listen /run/cups/cups.sock

# Show shared printers on the local network.
Browsing No
BrowseLocalProtocols dnssd

# Default authentication type, when authentication is required...
DefaultAuthType Basic

# Web interface setting...
WebInterface Yes

# Timeout after cupsd exits if idle (applied only if cupsd runs on-demand - with -l)
IdleExitTimeout 60

# Restrict access to the server...
<Location />
  Order allow,deny
  Allow from all
</Location>
```

After saving my changes and exiting the document I ran **sudo systemctl restart cups** , so the new configuration would be loaded and CUPS was running as a service.



```
vm2-server-daniel@vm2-server-daniel:~$ sudo systemctl restart cups
vm2-server-daniel@vm2-server-daniel:~$ _
```

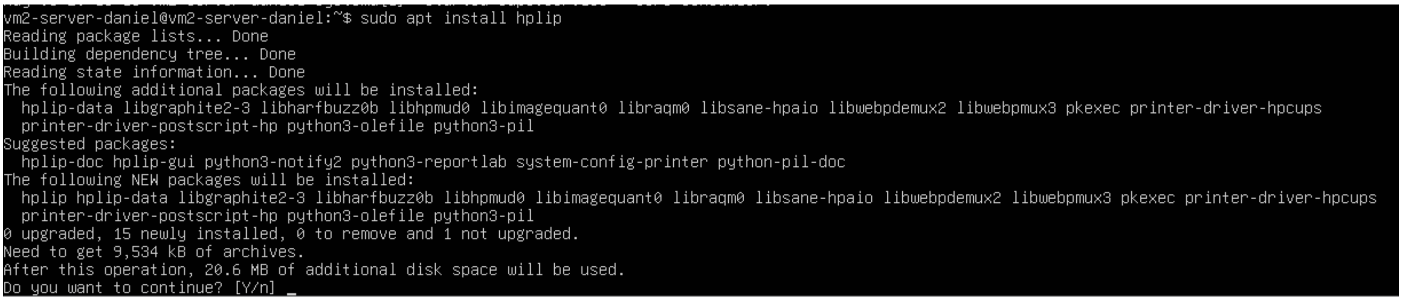
To double check that CUPS was running and active I ran: **sudo systemctl status cups**



```
vm2-server-daniel@vm2-server-daniel:~$ sudo systemctl status cups
● cups.service - CUPS Scheduler
     Loaded: loaded (/usr/lib/systemd/system/cups.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-05-03 20:18:18 UTC; 1min 3s ago
TriggeredBy: ● cups.path
             ● cups.socket
       Docs: man:cupsd(8)
   Main PID: 22570 (cupsd)
     Status: "Scheduler is running..."
      Tasks: 1 (limit: 4609)
     Memory: 2.3M (peak: 2.5M)
        CPU: 13ms
     CGroup: /system.slice/cups.service
             └─22570 /usr/sbin/cupsd -l
```

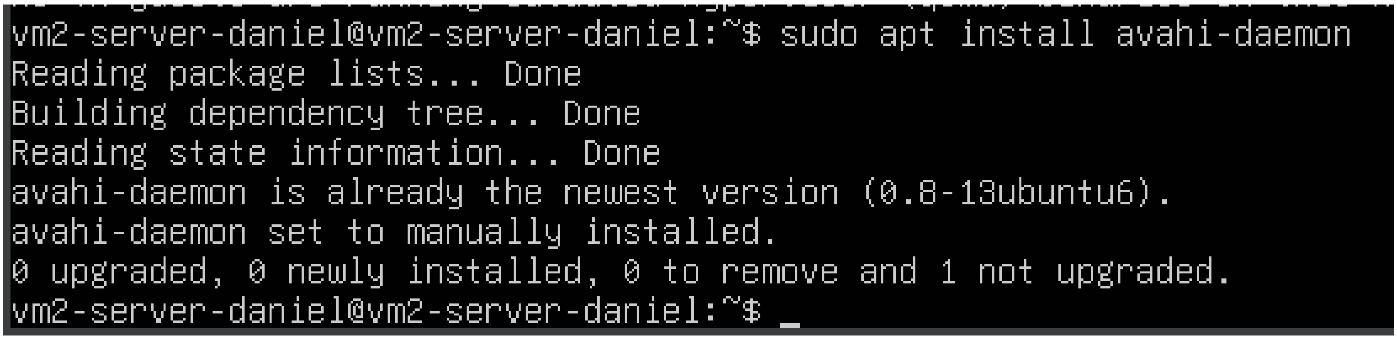
I decided to install HPLIP (HP Linux Imaging and Printing) package using **sudo apt install hplip**

The reason I did this is I wanted to make sure that the drivers[viii] and features for managing HP printers was available making it so printing from these would be free from errors.



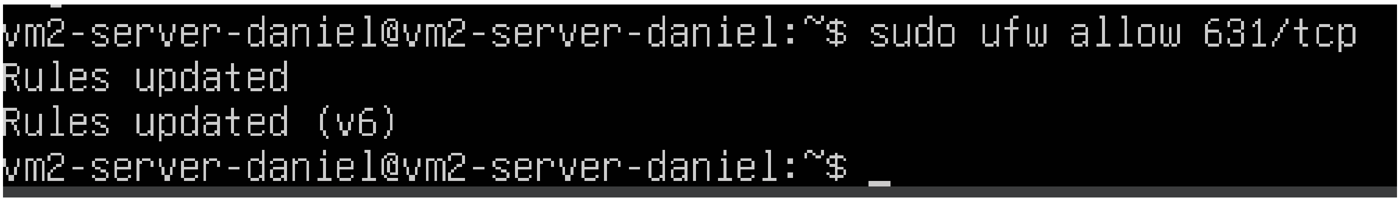
```
vm2-server-daniel@vm2-server-daniel:~$ sudo apt install hplip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  hplip-data libgraphite2-3 libharfbuzz0b libhpmud0 libimagequant0 libraqm0 libsane-hpaio libwebpdemux2 libwebpmux3 pkexec printer-driver-hpcups
  printer-driver-postscript-hp python3-olefile python3-pil
Suggested packages:
  hplip-doc hplip-gui python3-notify2 python3-reportlab system-config-printer python-pil-doc
The following NEW packages will be installed:
  hplip hplip-data libgraphite2-3 libharfbuzz0b libhpmud0 libimagequant0 libraqm0 libsane-hpaio libwebpdemux2 libwebpmux3 pkexec printer-driver-hpcups
  printer-driver-postscript-hp python3-olefile python3-pil
0 upgraded, 15 newly installed, 0 to remove and 1 not upgraded.
Need to get 9,534 kB of archives.
After this operation, 20.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

I also chose to install avahi daemon so CUPS could show shared printers on the local network, This way client machines could detect available printers.



```
vm2-server-daniel@vm2-server-daniel:~$ sudo apt install avahi-daemon
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
avahi-daemon is already the newest version (0.8-13ubuntu6).
avahi-daemon set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
vm2-server-daniel@vm2-server-daniel:~$ _
```

Lastly I chose to allow port 631 through the firewall to avoid any difficulties accessing CUPS

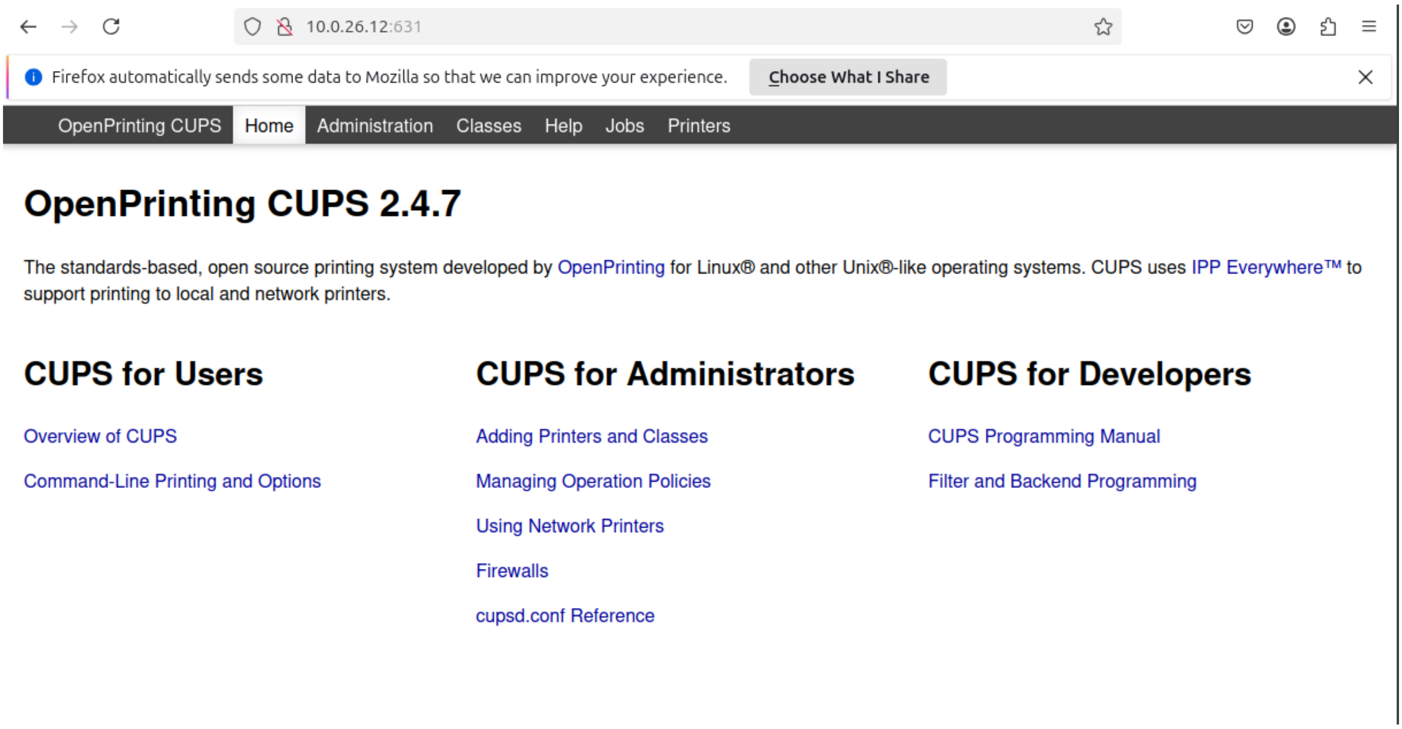


```
vm2-server-daniel@vm2-server-daniel:~$ sudo ufw allow 631/tcp
Rules updated
Rules updated (v6)
vm2-server-daniel@vm2-server-daniel:~$ _
```

To confirm that CUPS was set up correctly I opened the web browser on my DHCP client on the internal network and went to:

http://10.0.26.12:631

This let me access server 2's cups printing service page running on port 631 on my client machine. With this I now knew that my print server was online and able to be accessed without any issues across my network. Here anyone on the network was able to add printers among many other features.



OpenPrinting CUPS 2.4.7

The standards-based, open source printing system developed by OpenPrinting for Linux® and other Unix®-like operating systems. CUPS uses IPP Everywhere™ to support printing to local and network printers.

CUPS for Users

Overview of CUPS

Command-Line Printing and Options

CUPS for Administrators

Adding Printers and Classes

Managing Operation Policies

Using Network Printers

Firewalls

cupsd.conf Reference

CUPS for Developers

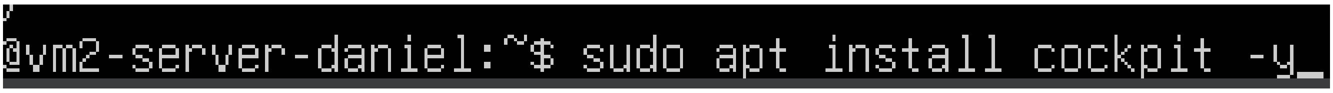
CUPS Programming Manual

Filter and Backend Programming

Installing Cockpit Management Server (Server 2):[ix]

For my management server I chose to use cockpit management interface as it is web based and would be easy to navigate and use on my client machine once set up. To begin I installed the cockpit service using:

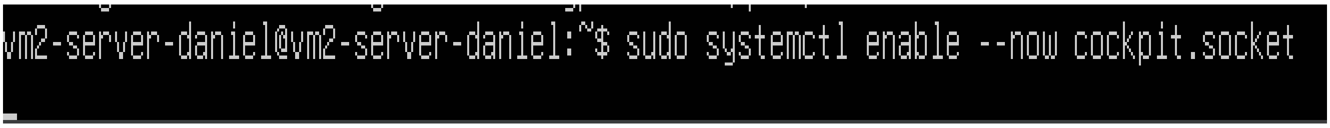
Sudo apt install cockpit -y



```
@vm2-server-daniel:~$ sudo apt install cockpit -y_
```

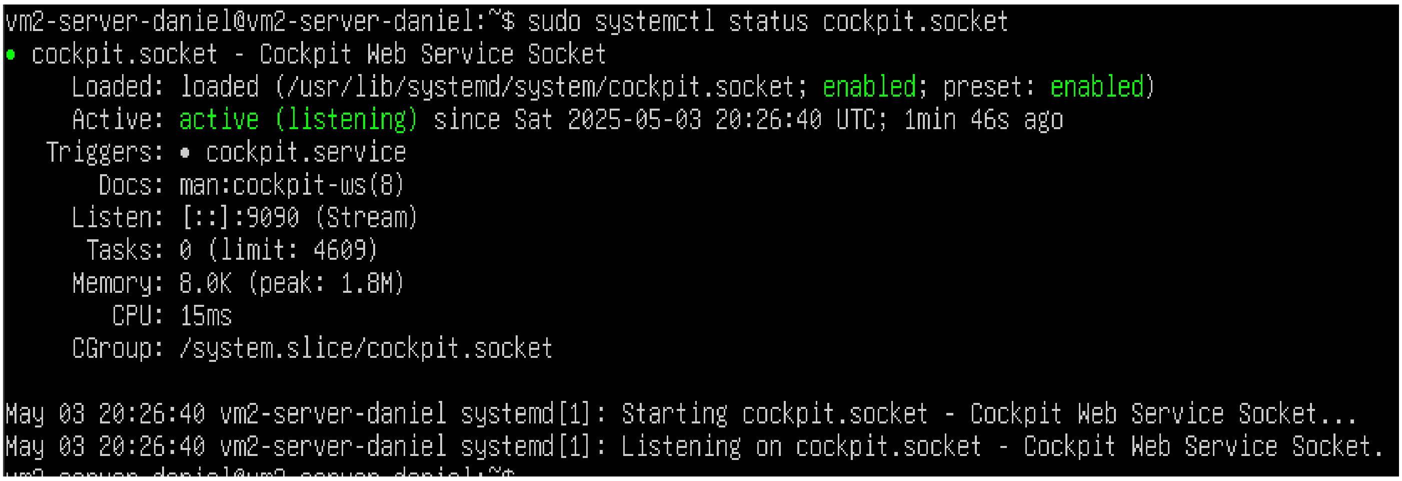
After Cockpit service was installed , I enabled and started it using:

Sudo systemctl enable –now cockpit.socket



```
vm2-server-daniel@vm2-server-daniel:~$ sudo systemctl enable --now cockpit.socket
_
```

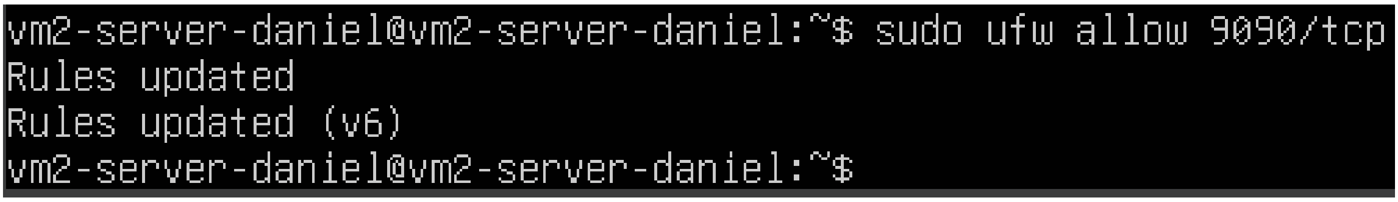
I then checked if it was enabled and saw it was running and listening on port 9090



```
vm2-server-daniel@vm2-server-daniel:~$ sudo systemctl status cockpit.socket
• cockpit.socket - Cockpit Web Service Socket
     Loaded: loaded (/usr/lib/systemd/system/cockpit.socket; enabled; preset: enabled)
     Active: active (listening) since Sat 2025-05-03 20:26:40 UTC; 1min 46s ago
   Triggers: • cockpit.service
       Docs: man:cockpit-ws(8)
     Listen: [::]:9090 (Stream)
      Tasks: 0 (limit: 4609)
     Memory: 8.0K (peak: 1.8M)
        CPU: 15ms
     CGroup: /system.slice/cockpit.socket

May 03 20:26:40 vm2-server-daniel systemd[1]: Starting cockpit.socket - Cockpit Web Service Socket...
May 03 20:26:40 vm2-server-daniel systemd[1]: Listening on cockpit.socket - Cockpit Web Service Socket.
```

For this reason I allowed access on the firewall to port 9090

Sudo ufw allow 9090/tcp

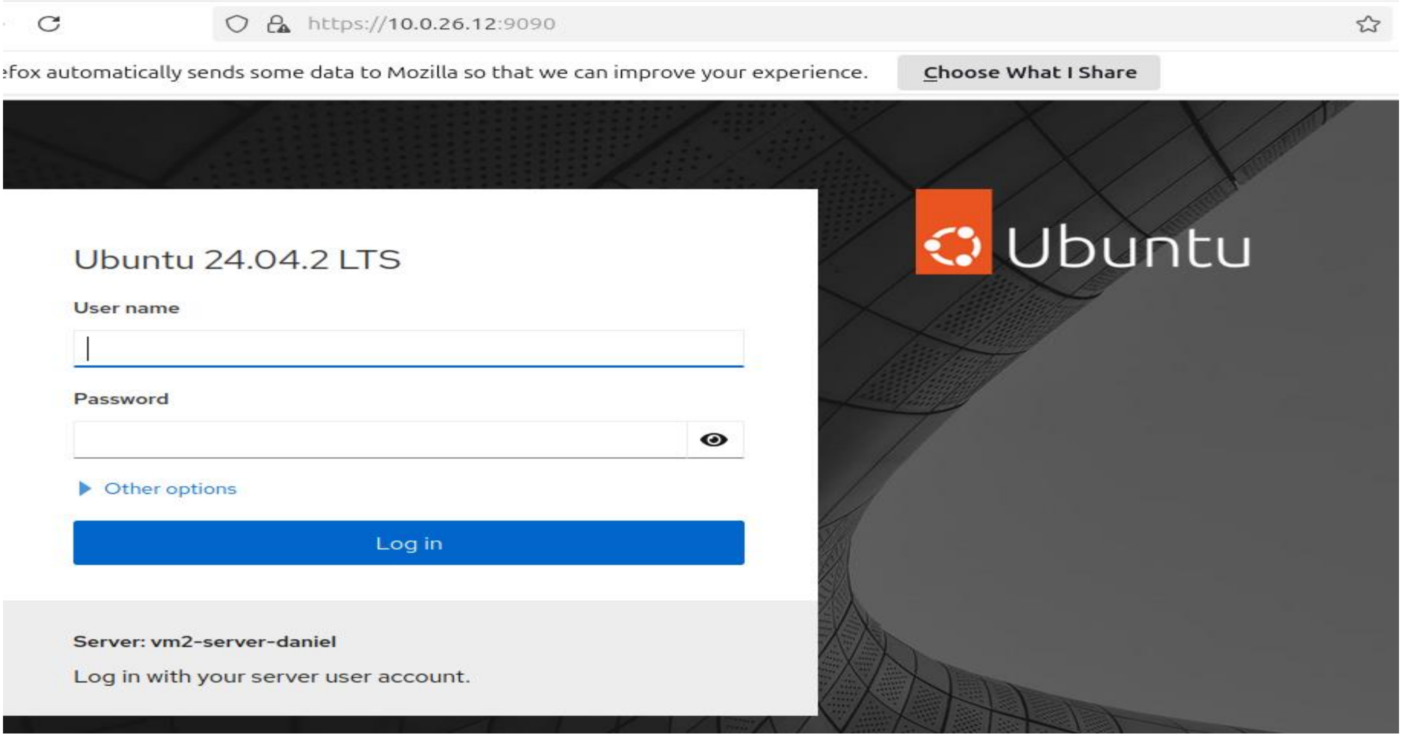


```
vm2-server-daniel@vm2-server-daniel:~$ sudo ufw allow 9090/tcp
Rules updated
Rules updated (v6)
vm2-server-daniel@vm2-server-daniel:~$
```

To confirm that Cockpit was set up correctly I opened the web browser on my DHCP client set to the internal network and went to:

https://10.0.26.12:9090

This let me access Server VM2's cockpit management service page running on port 9090 on my client machine. With this I now knew that my management server was running and able to be accessed without any issues by admins on my network.



https://10.0.26.12:9090

≀fox automatically sends some data to Mozilla so that we can improve your experience.

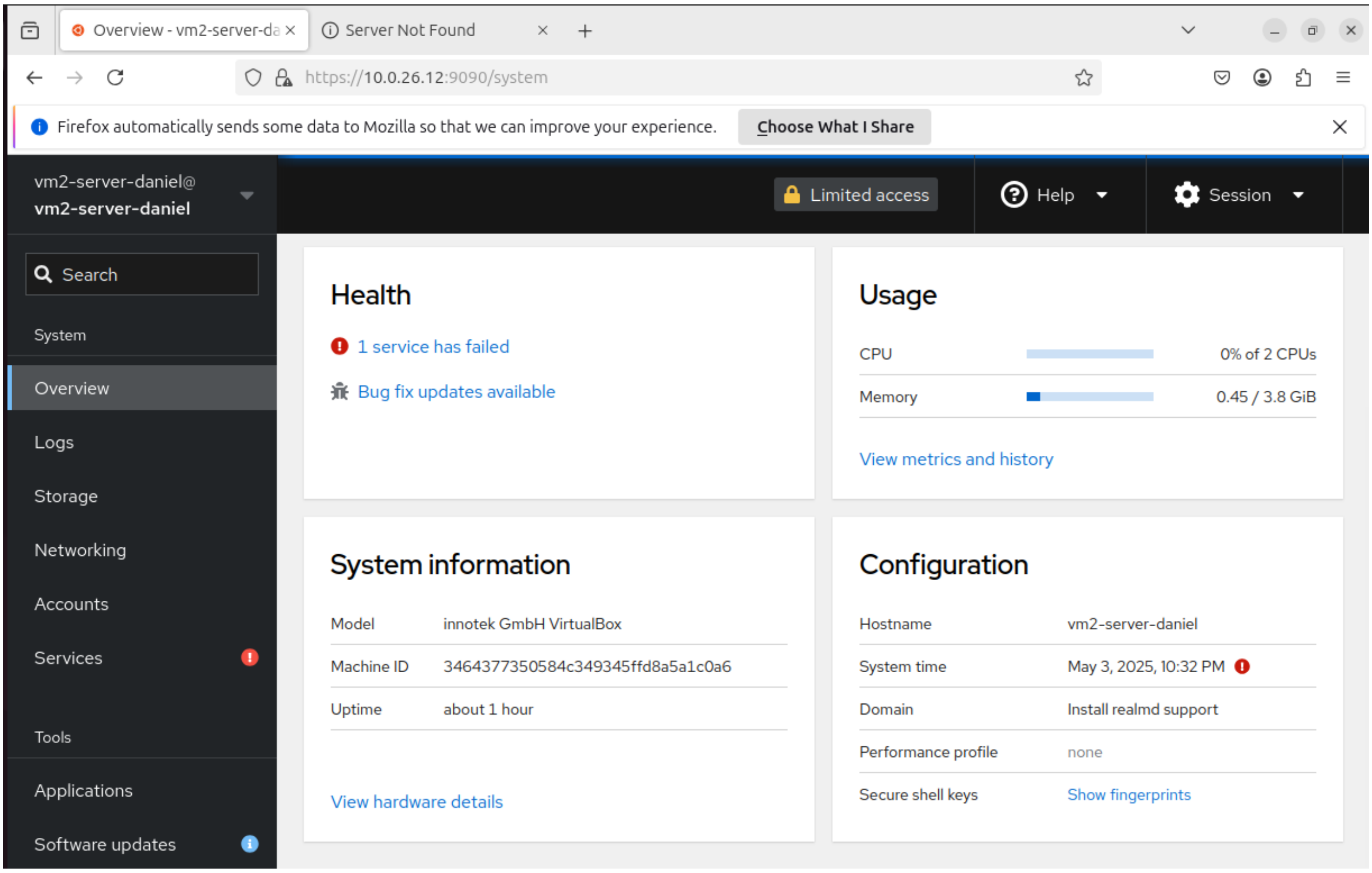
Ubuntu 24.04.2 LTS

User name

Password

Server: vm2-server-daniel

Log in with your server user account.



Overview - vm2-server-da
Server Not Found

https://10.0.26.12:9090/system

Firefox automatically sends some data to Mozilla so that we can improve your experience.

vm2-server-daniel@
vm2-server-daniel

Health

1 service has failed

Bug fix updates available

Usage

CPU	0% of 2 CPUs
Memory	0.45 / 3.8 GiB

View metrics and history

System information

Model	innotek GmbH VirtualBox
Machine ID	3464377350584c349345ffd8a5a1c0a6
Uptime	about 1 hour

View hardware details

Configuration

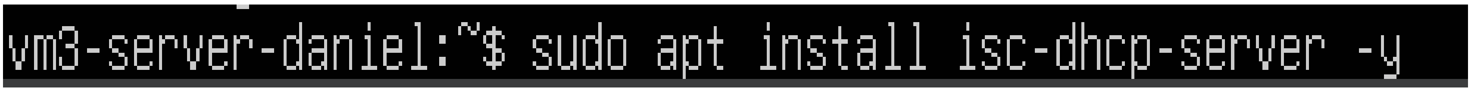
Hostname	vm2-server-daniel
System time	May 3, 2025, 10:32 PM
Domain	Install realmd support
Performance profile	none
Secure shell keys	Show fingerprints

Installing DHCP Server (VM3 Server):[x]

To automate IP address assignment to my client on my network, I configured my VM3 server as a DHCP server. To start I installed ISC DHCP server package using:

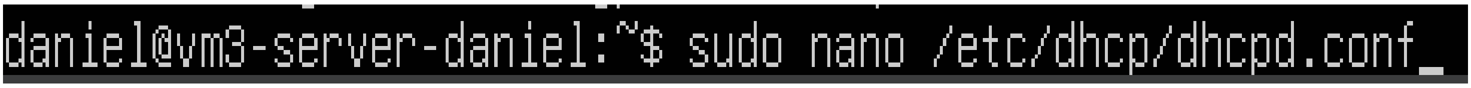
Sudo apt install isc-dhcp-server -y

This package makes it so I have what I need to assign dynamic IP addresses to clients in my 10.0.26.0/24 subnet



```
vm3-server-daniel:~$ sudo apt install isc-dhcp-server -y
```

After everything was installed I navigated to the configuration file using: **sudo nano /etc/dhcp/dhcp.conf**



```
daniel@vm3-server-daniel:~$ sudo nano /etc/dhcp/dhcpd.conf_
```

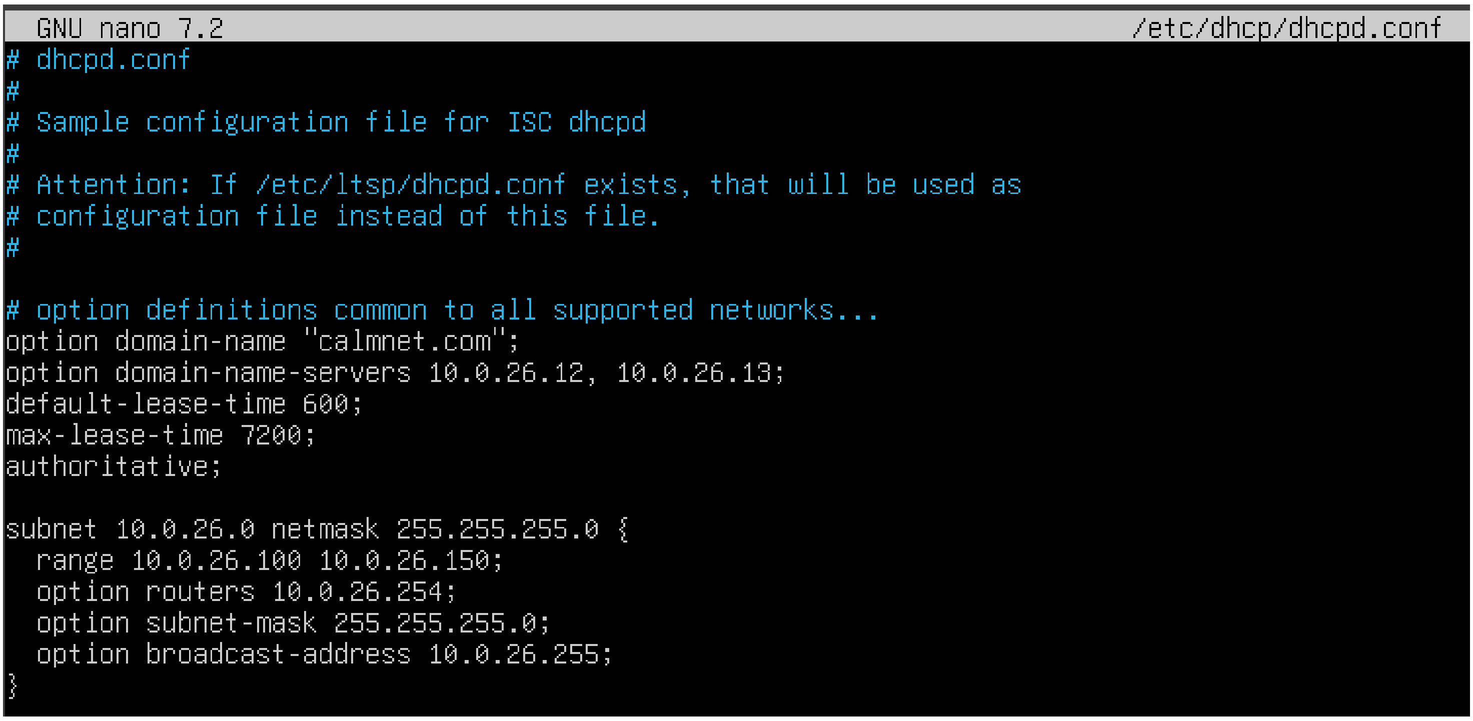
I edited the DHCP config file to set the ranges and IP's used on my calmnet.com internal network , I listened both my DNS servers : 10.0.26.12 and 10.0.26.13,

After that I then created a subnet block for 10.0.26.0/24 and also set:

IP Range: 10.0.26.100 to 10.0.26.150

Default Gateway: 10.0.26.254

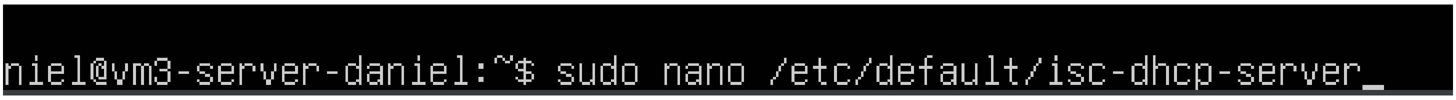
And finally a subnet mask and broadcast address



```
  GNU nano 7.2                                                          /etc/dhcp/dhcpd.conf
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#

# option definitions common to all supported networks...
option domain-name "calmnet.com";
option domain-name-servers 10.0.26.12, 10.0.26.13;
default-lease-time 600;
max-lease-time 7200;
authoritative;

subnet 10.0.26.0 netmask 255.255.255.0 {
  range 10.0.26.100 10.0.26.150;
  option routers 10.0.26.254;
  option subnet-mask 255.255.255.0;
  option broadcast-address 10.0.26.255;
}
```

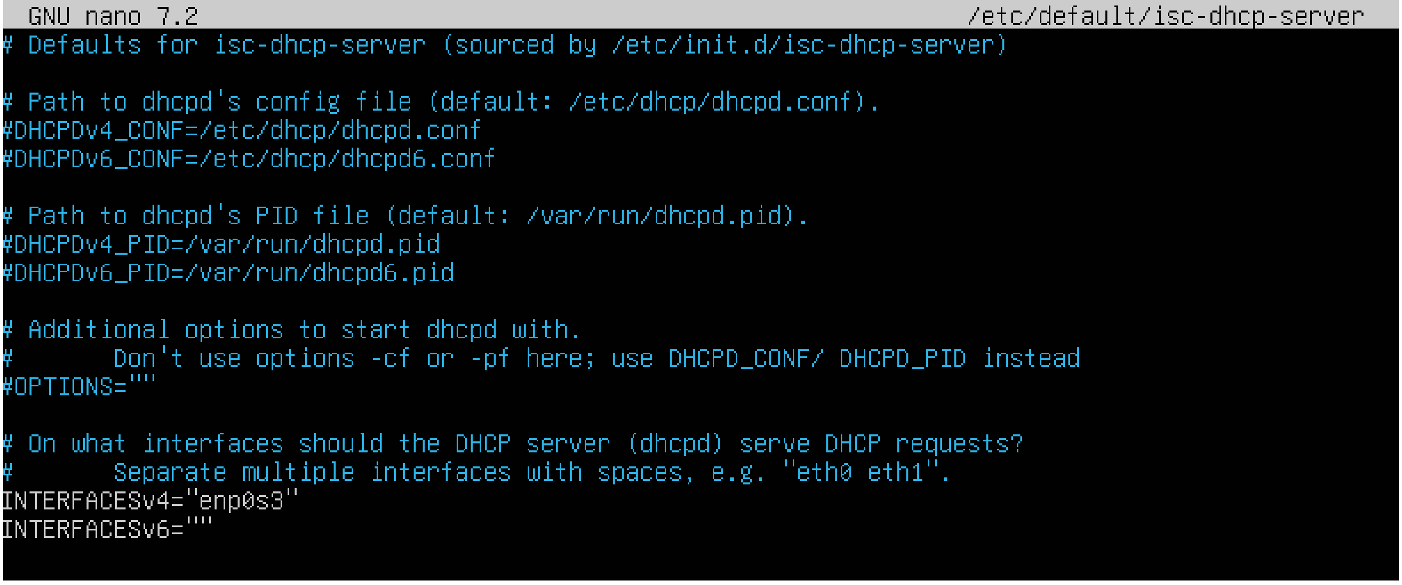
After I saved and exited from the dhcp.conf file I next went to configure the isc-dhcp-server file using:

sudo nano /etc/default/isc-dhcp-server



```
niel@vm3-server-daniel:~$ sudo nano /etc/default/isc-dhcp-server_
```

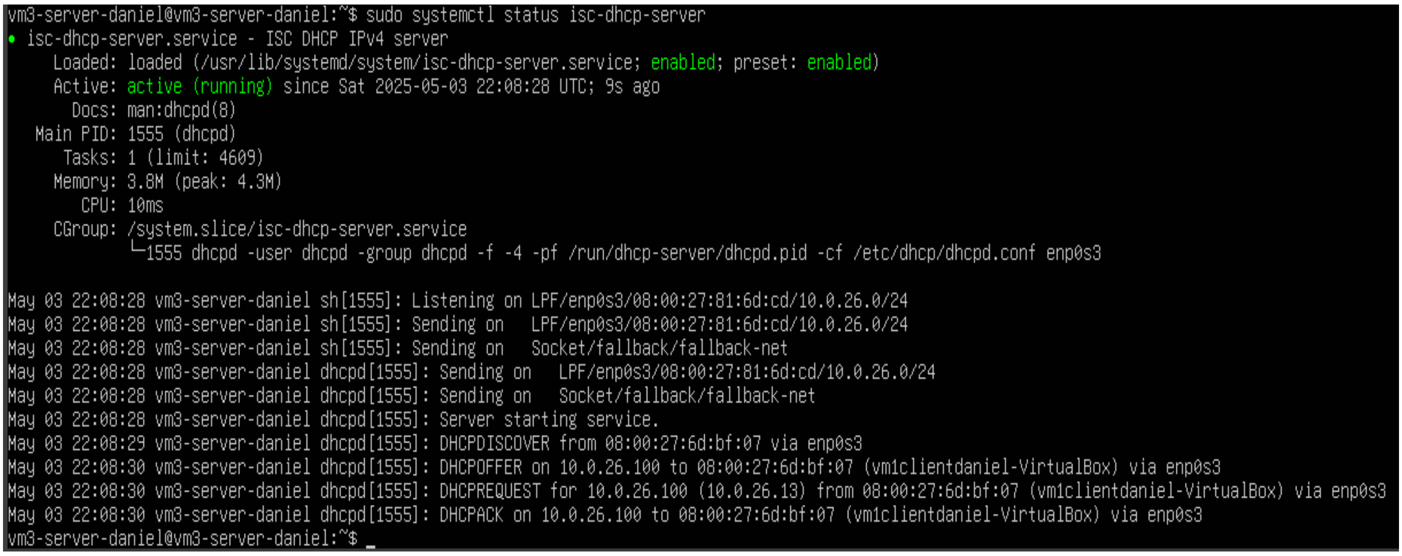
In this file I made sure the DHCP server was set up to use the correct interface for my internal network (enp0s3) on IPV4 and not use anything for IPV6 so it takes IPV4 addresses from the local range only.



```
  GNU nano 7.2                                                    /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s3"
INTERFACESv6=""
```

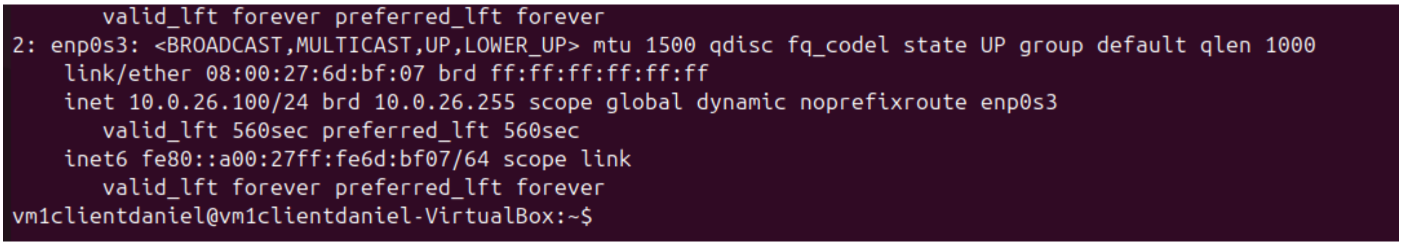
I saved and exited from the file once finished.

I then ran **sudo systemctl status isc-dhcp-server** to check if the service was up and running.



```
vm3-server-daniel@vm3-server-daniel:~$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/usr/lib/systemd/system/isc-dhcp-server.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-05-03 22:08:28 UTC; 9s ago
       Docs: man:dhcpd(8)
   Main PID: 1555 (dhcpd)
      Tasks: 1 (limit: 4609)
     Memory: 3.8M (peak: 4.3M)
        CPU: 10ms
     CGroup: /system.slice/isc-dhcp-server.service
             └─1555 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf enp0s3

May 03 22:08:28 vm3-server-daniel sh[1555]: Listening on LPF/enp0s3/08:00:27:81:6d:cd/10.0.26.0/24
May 03 22:08:28 vm3-server-daniel sh[1555]: Sending on   LPF/enp0s3/08:00:27:81:6d:cd/10.0.26.0/24
May 03 22:08:28 vm3-server-daniel sh[1555]: Sending on   Socket/fallback/fallback-net
May 03 22:08:28 vm3-server-daniel dhcpd[1555]: Sending on   LPF/enp0s3/08:00:27:81:6d:cd/10.0.26.0/24
May 03 22:08:28 vm3-server-daniel dhcpd[1555]: Sending on   Socket/fallback/fallback-net
May 03 22:08:28 vm3-server-daniel dhcpd[1555]: Server starting service.
May 03 22:08:29 vm3-server-daniel dhcpd[1555]: DHCPDISCOVER from 08:00:27:6d:bf:07 via enp0s3
May 03 22:08:30 vm3-server-daniel dhcpd[1555]: DHCPOFFER on 10.0.26.100 to 08:00:27:6d:bf:07 (vm1clientdaniel-VirtualBox) via enp0s3
May 03 22:08:30 vm3-server-daniel dhcpd[1555]: DHCPREQUEST for 10.0.26.100 (10.0.26.13) from 08:00:27:6d:bf:07 (vm1clientdaniel-VirtualBox) via enp0s3
May 03 22:08:30 vm3-server-daniel dhcpd[1555]: DHCPACK on 10.0.26.100 to 08:00:27:6d:bf:07 (vm1clientdaniel-VirtualBox) via enp0s3
vm3-server-daniel@vm3-server-daniel:~$ _
```

Lastly I switched to my Client on the internal network and ran: **ip a** . This gave me an IP from the set range on the correct interface telling me that DHCP was correctly set up and configured on my internal network.

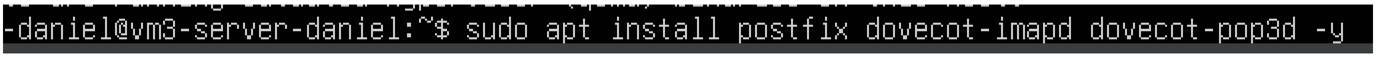


```
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6d:bf:07 brd ff:ff:ff:ff:ff:ff
    inet 10.0.26.100/24 brd 10.0.26.255 scope global dynamic noprefixroute enp0s3
       valid_lft 560sec preferred_lft 560sec
    inet6 fe80::a00:27ff:fe6d:bf07/64 scope link
       valid_lft forever preferred_lft forever
vm1clientdaniel@vm1clientdaniel-VirtualBox:~$
```

Email Server: Postfix and IMAP Server: Dovecot (VM2 Server):[xi]

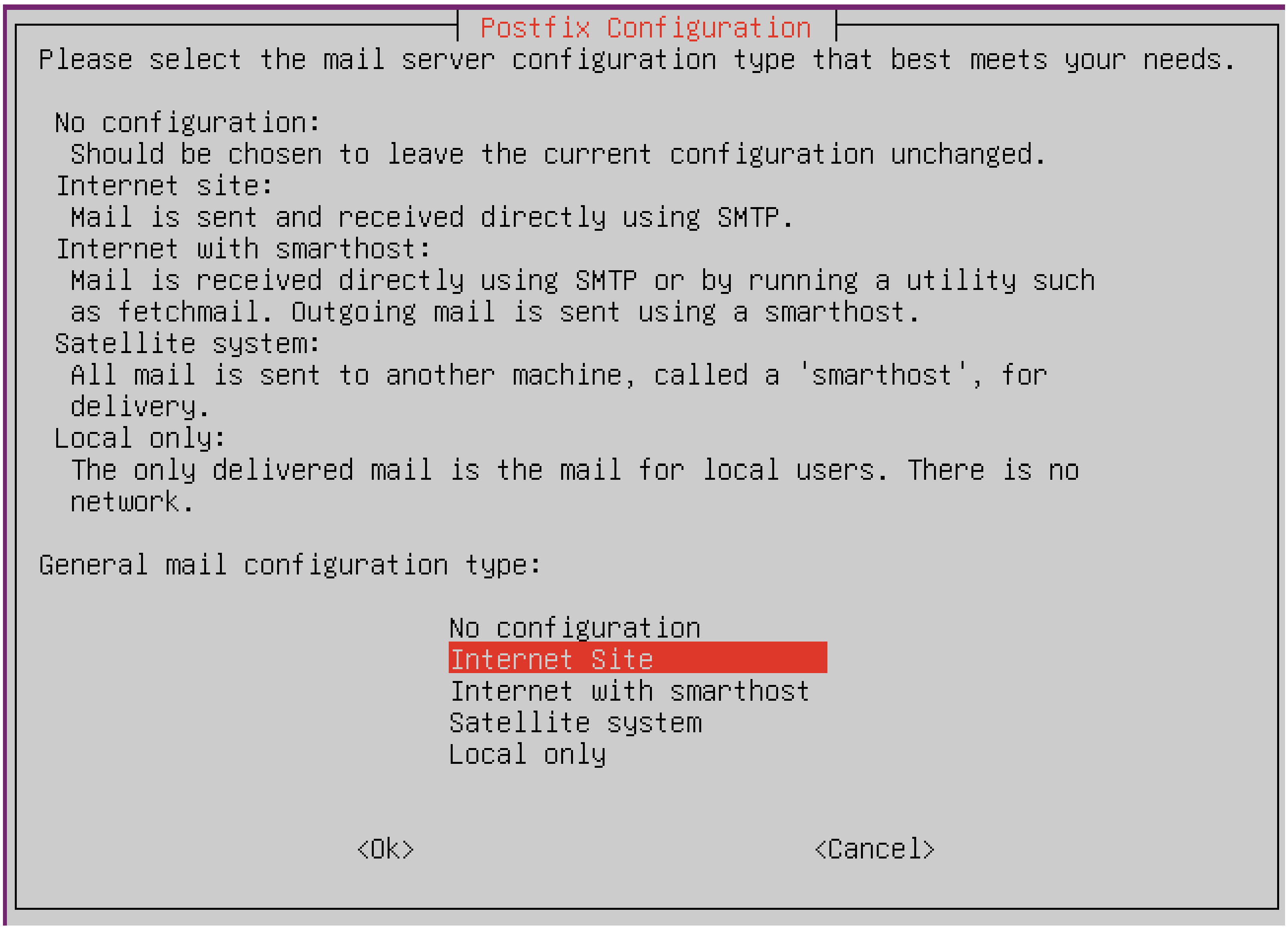
Next to set up my email server on my network, On VM2 I chose to go with PostFix for sending mail and dovecot for receiving mail using IMAP and POP3.

To install this I ran the command: **sudo apt install postfix dovecot-imapd dovecot-pop3d -y**



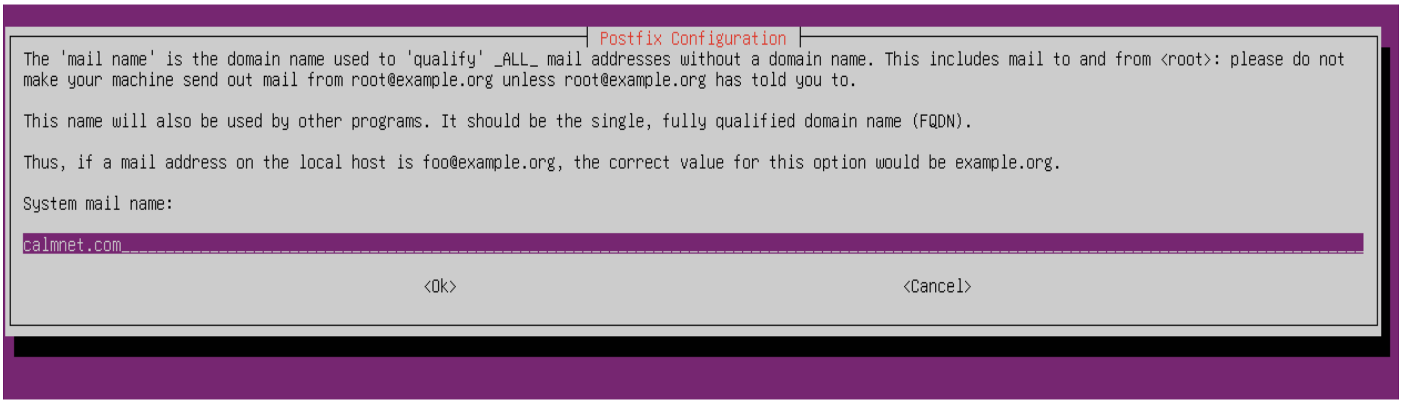
```
-daniel@vm3-server-daniel:~$ sudo apt install postfix dovecot-imapd dovecot-pop3d -y
```

This took me to a postfix configuration screen where I selected Internet Site as this seemed the most straightforward to configure.



```
                          Postfix Configuration
 Please select the mail server configuration type that best meets your needs.

  No configuration:
   Should be chosen to leave the current configuration unchanged.
  Internet site:
   Mail is sent and received directly using SMTP.
  Internet with smarthost:
   Mail is received directly using SMTP or by running a utility such
   as fetchmail. Outgoing mail is sent using a smarthost.
  Satellite system:
   All mail is sent to another machine, called a 'smarthost', for
   delivery.
  Local only:
   The only delivered mail is the mail for local users. There is no
   network.

 General mail configuration type:

                          No configuration
                          Internet Site
                          Internet with smarthost
                          Satellite system
                          Local only


                     <Ok>                         <Cancel>
```

Here I was prompted to configure the system mail name, I set this to calmnet.com to keep the domain names consistent with all of the other services on my internal network avoiding any issues.

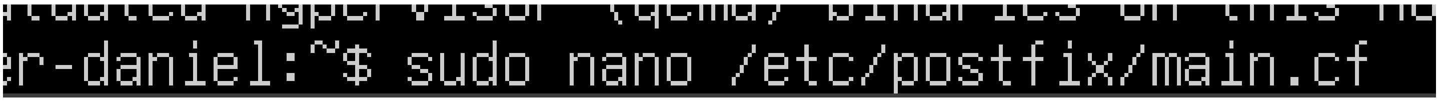


```
                                         Postfix Configuration
 The 'mail name' is the domain name used to 'qualify' _ALL_ mail addresses without a domain name. This includes mail to and from <root>: please do not
 make your machine send out mail from root@example.org unless root@example.org has told you to.

 This name will also be used by other programs. It should be the single, fully qualified domain name (FQDN).

 Thus, if a mail address on the local host is foo@example.org, the correct value for this option would be example.org.

 System mail name:

 calmnet.com_____________________________________________________________________________________________________________________________________

                               <Ok>                                                    <Cancel>
```

After setting the system’s mail name I navigated to Postfix’s main configuration file using:

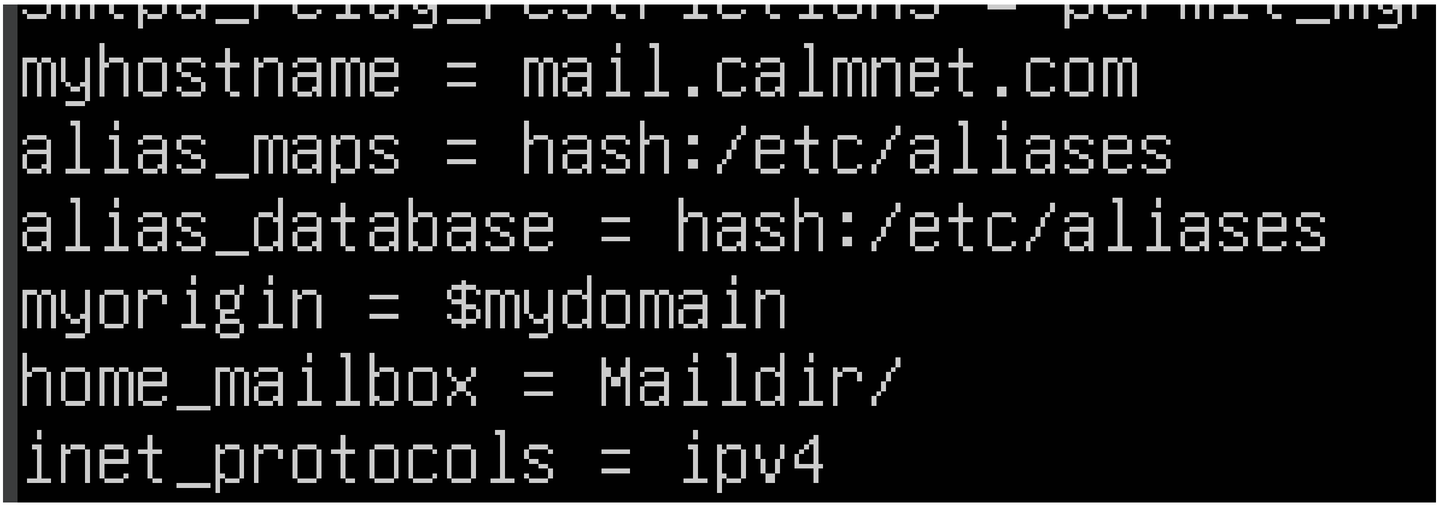
sudo nano /etc/postfix/main.cf

In this file I could tune how mail is handled on my internal network as well as set hostnames and domains



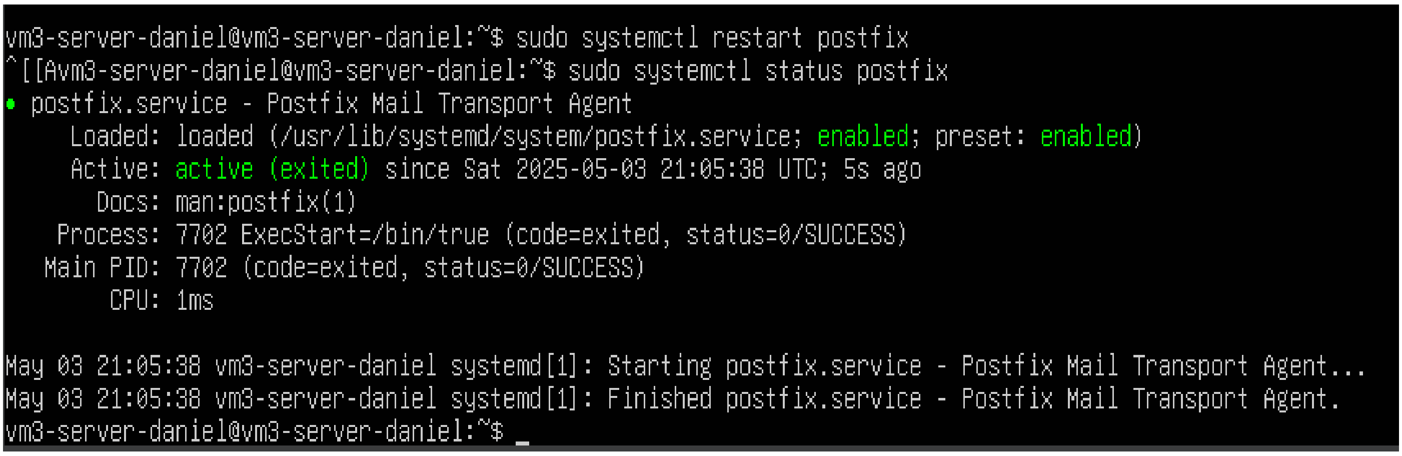
```
rtualized hypervisor (qemu) binaries on this ho
er-daniel:~$ sudo nano /etc/postfix/main.cf
```

In the main.cf file I made it so PostFix matches my network’s domain (calmnet.com) , I also set my hostname to mail.calmnet.com so it’s the same as the mail server’s. I then set the home mailbox to Maildir/ so incoming mail is easy to find and works with dovecot properly.



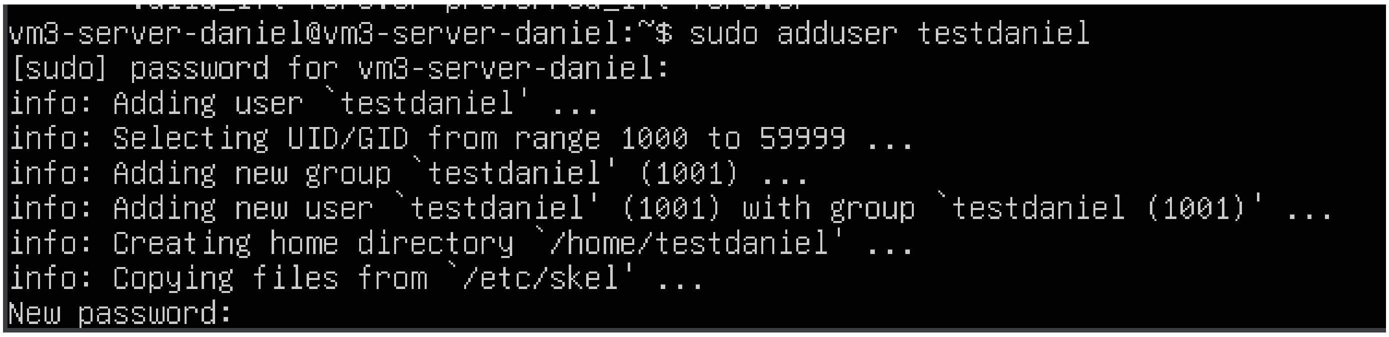
```
smtpd_relay_restrictions = permit_myn
myhostname = mail.calmnet.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = $mydomain
home_mailbox = Maildir/
inet_protocols = ipv4
```

Next I saved the configs I made and restarted postfix as well as checking its status where I could see that it was up and running.



```
vm3-server-daniel@vm3-server-daniel:~$ sudo systemctl restart postfix
^[[Avm3-server-daniel@vm3-server-daniel:~$ sudo systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
     Loaded: loaded (/usr/lib/systemd/system/postfix.service; enabled; preset: enabled)
     Active: active (exited) since Sat 2025-05-03 21:05:38 UTC; 5s ago
       Docs: man:postfix(1)
    Process: 7702 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 7702 (code=exited, status=0/SUCCESS)
        CPU: 1ms

May 03 21:05:38 vm3-server-daniel systemd[1]: Starting postfix.service - Postfix Mail Transport Agent...
May 03 21:05:38 vm3-server-daniel systemd[1]: Finished postfix.service - Postfix Mail Transport Agent.
vm3-server-daniel@vm3-server-daniel:~$ _
```

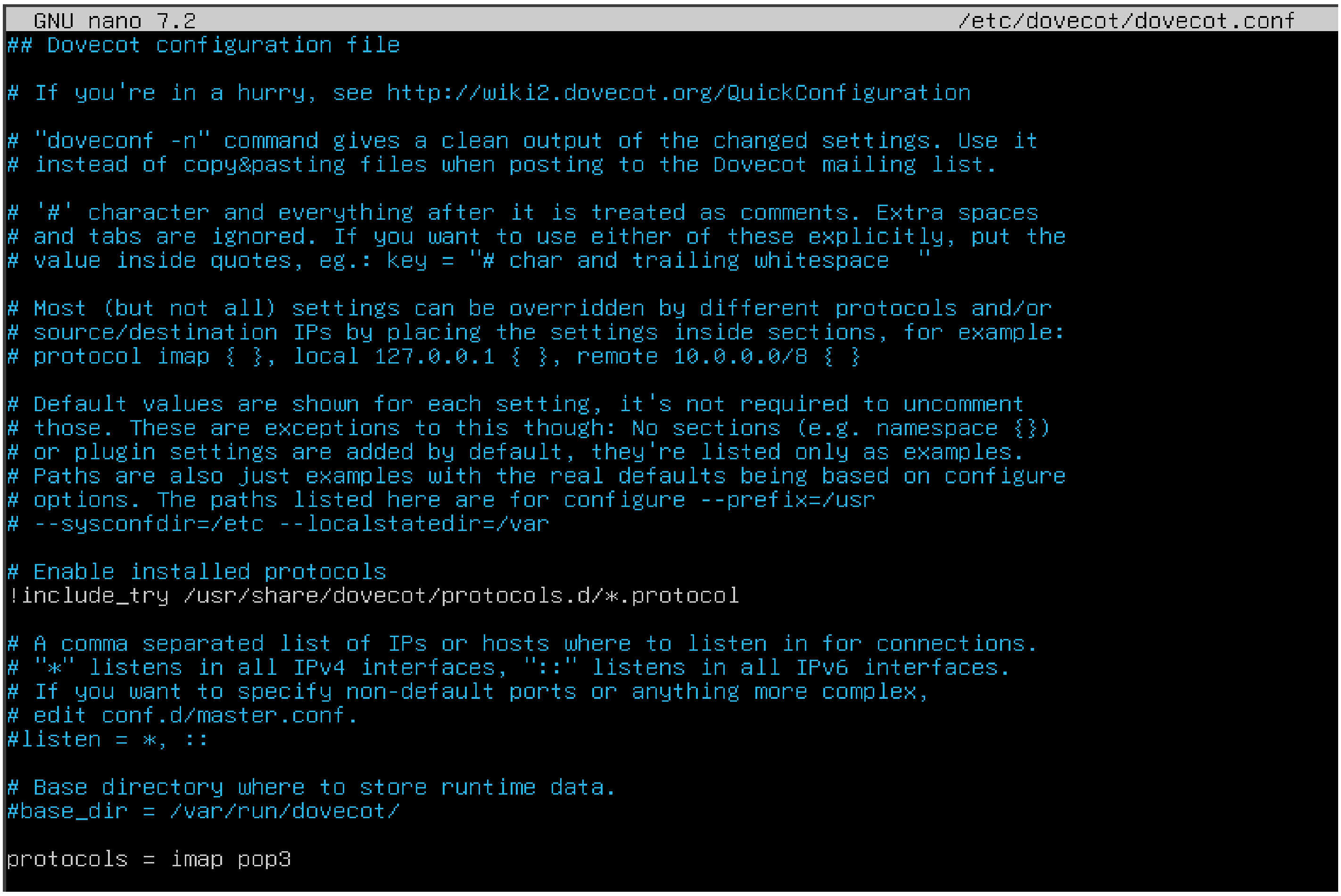
I also made a new test user named “testdaniel” using **sudo adduser testdaniel**



```
       valid_lft forever preferred_lft forever
vm3-server-daniel@vm3-server-daniel:~$ sudo adduser testdaniel
[sudo] password for vm3-server-daniel:
info: Adding user `testdaniel' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `testdaniel' (1001) ...
info: Adding new user `testdaniel' (1001) with group `testdaniel (1001)' ...
info: Creating home directory `/home/testdaniel' ...
info: Copying files from `/etc/skel' ...
New password:
```

Next I configured dovecot using **sudo nano /etc/dovecot/dovecot.conf**

To enable mail access I edited the dovecot config file and set the supported protocols to imap and pop3 so it works with most email clients that connect to IMAP and POP3 when getting mail, This works perfectly with postfix

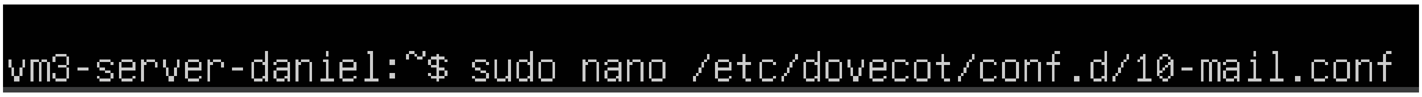


```
  GNU nano 7.2                                                      /etc/dovecot/dovecot.conf
## Dovecot configuration file

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

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

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

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

# Default values are shown for each setting, it's not required to uncomment
# those. These are exceptions to this though: No sections (e.g. namespace {})
# or plugin settings are added by default, they're listed only as examples.
# Paths are also just examples with the real defaults being based on configure
# options. The paths listed here are for configure --prefix=/usr
# --sysconfdir=/etc --localstatedir=/var

# Enable installed protocols
!include_try /usr/share/dovecot/protocols.d/*.protocol

# A comma separated list of IPs or hosts where to listen in for connections.
# "*" listens in all IPv4 interfaces, "::" listens in all IPv6 interfaces.
# If you want to specify non-default ports or anything more complex,
# edit conf.d/master.conf.
#listen = *, ::

# Base directory where to store runtime data.
#base_dir = /var/run/dovecot/

protocols = imap pop3
```

I saved and exited this and navigated to my next config file using:

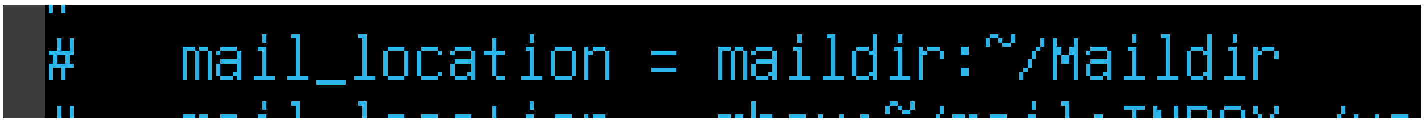
Sudo nano /etc/dovecot/conf.d/10-mail.conf

This was done to make sure that dovecot knows where to store and retrieve every users' mail properly



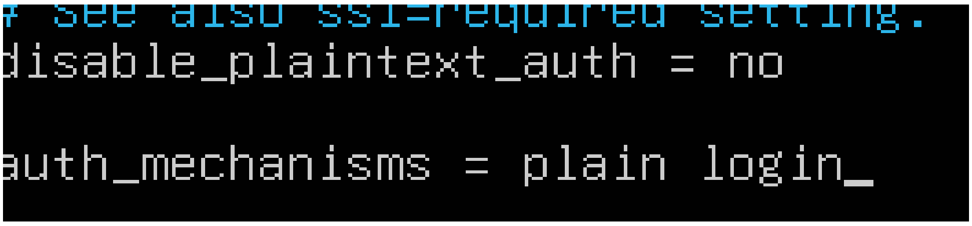
```
vm3-server-daniel:~$ sudo nano /etc/dovecot/conf.d/10-mail.conf
```

I set the mail location to the same format as Postfix's delivery settings so everything would work with no issues.



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

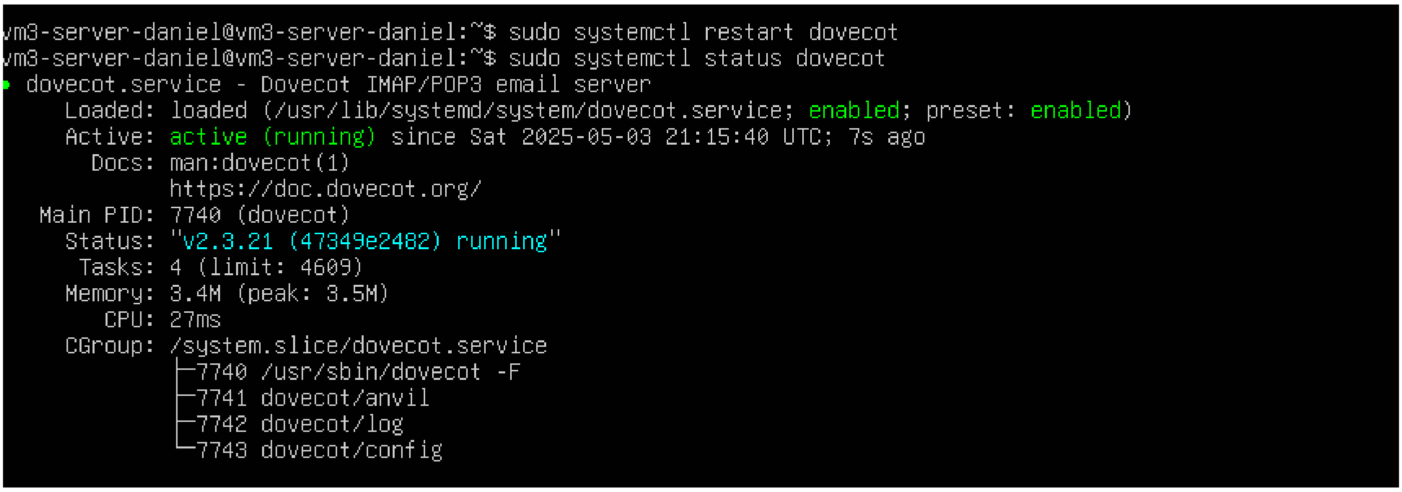
Also in the config file I chose to allow basic login authentication without SSL since this is a controlled internal network and makes it more convenient for users on the network. Auth_mechanisms = plain login also allows dovecot to accept plain login so its compatible with most basic email clients.



```
# See also ssl=required setting.
disable_plaintext_auth = no

auth_mechanisms = plain login
```

I saved and exited from the config file and next I restarted dovecot with: **sudo systemctl restart dovecot**

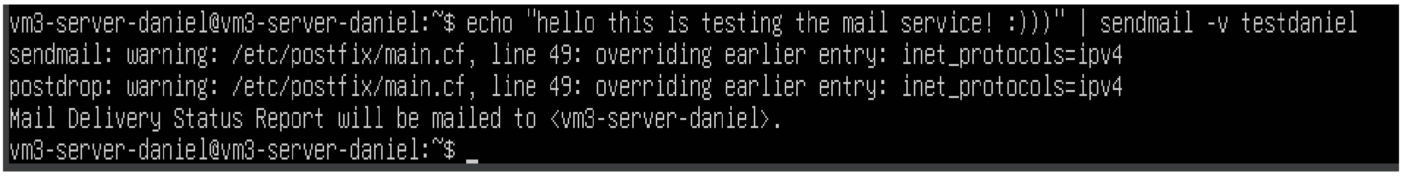
I then checked if it was up and running successfully with: **sudo systemctl status dovecot**



```
vm3-server-daniel@vm3-server-daniel:~$ sudo systemctl restart dovecot
vm3-server-daniel@vm3-server-daniel:~$ sudo systemctl status dovecot
● dovecot.service - Dovecot IMAP/POP3 email server
     Loaded: loaded (/usr/lib/systemd/system/dovecot.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-05-03 21:15:40 UTC; 7s ago
       Docs: man:dovecot(1)
             https://doc.dovecot.org/
   Main PID: 7740 (dovecot)
     Status: "v2.3.21 (47349e2482) running"
      Tasks: 4 (limit: 4609)
     Memory: 3.4M (peak: 3.5M)
        CPU: 27ms
     CGroup: /system.slice/dovecot.service
             ├─7740 /usr/sbin/dovecot -F
             ├─7741 dovecot/anvil
             ├─7742 dovecot/log
             └─7743 dovecot/config
```

Lastly to verify that my mail server was working I used the **sendmail** command to send a sample message to my "testdaniel" user that I set up earlier.

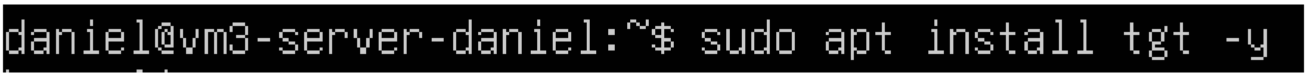
I was then met with a message that the Mail Delivery Status report will be mailed to my admin user this told me that my mail server is now working correctly.



```
vm3-server-daniel@vm3-server-daniel:~$ echo "hello this is testing the mail service! :)))" | sendmail -v testdaniel
sendmail: warning: /etc/postfix/main.cf, line 49: overriding earlier entry: inet_protocols=ipv4
postdrop: warning: /etc/postfix/main.cf, line 49: overriding earlier entry: inet_protocols=ipv4
Mail Delivery Status Report will be mailed to <vm3-server-daniel>.
vm3-server-daniel@vm3-server-daniel:~$
```

iSCSI Target Tool (tgt):[xii]

Lastly to set up shared storage over my network I decided to use tgt as my iSCSI target tool. I installed this package using: **sudo apt install tgt -y**

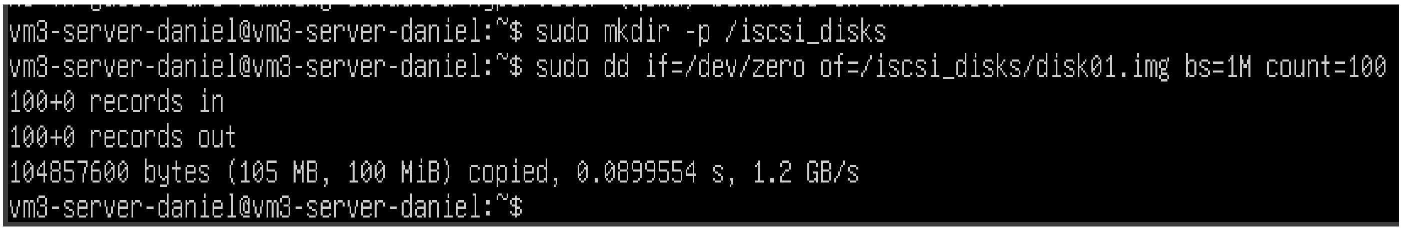


```
daniel@vm3-server-daniel:~$ sudo apt install tgt -y
```

After this was installed I next had to prepare a virtual disk that could be shared using iSCSI , I created a new dedicated directory for it using: **sudo mkdir -p /iscsi_disks**

In this directory I then created a blank 100MB disk image file using:

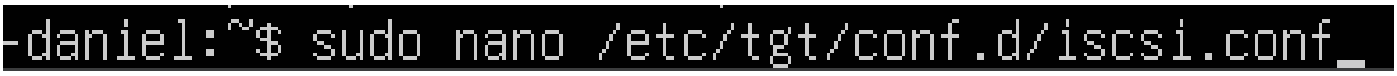
sudo dd if=/dev/zero of=/iscsi_disks/disk01.img bs=1M count =100



```
vm3-server-daniel@vm3-server-daniel:~$ sudo mkdir -p /iscsi_disks
vm3-server-daniel@vm3-server-daniel:~$ sudo dd if=/dev/zero of=/iscsi_disks/disk01.img bs=1M count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 0.0899554 s, 1.2 GB/s
vm3-server-daniel@vm3-server-daniel:~$
```

I saved and exited the config file , Next navigating to the iscsi.conf file using:

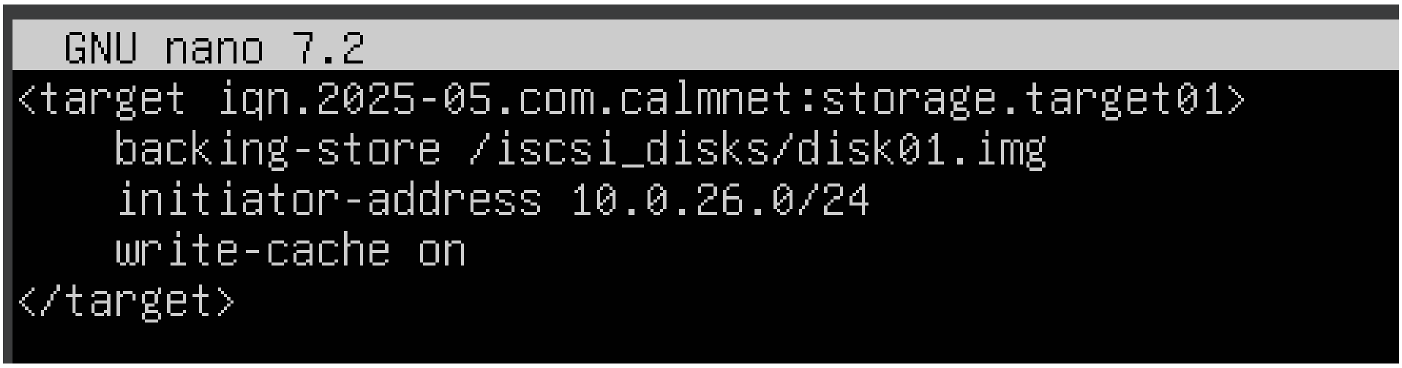
Sudo nano /etc/tgt/conf.d/iscsi.conf



```
-daniel:~$ sudo nano /etc/tgt/conf.d/iscsi.conf_
```

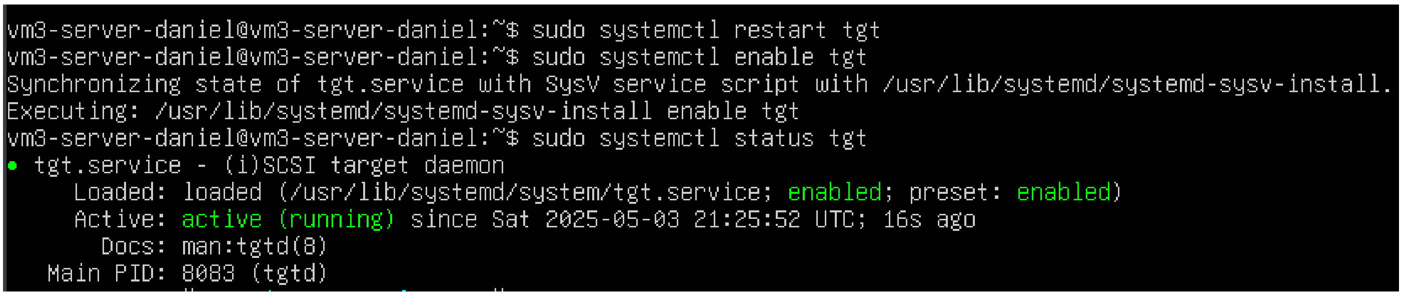
In the iscsi configuration file I created a target named: **iqn.2025-05.com.calmnet:storage.target01**

This points to the disk image and allows the 10.0.26.0 subnet on the network. After finishing this I saved and exited



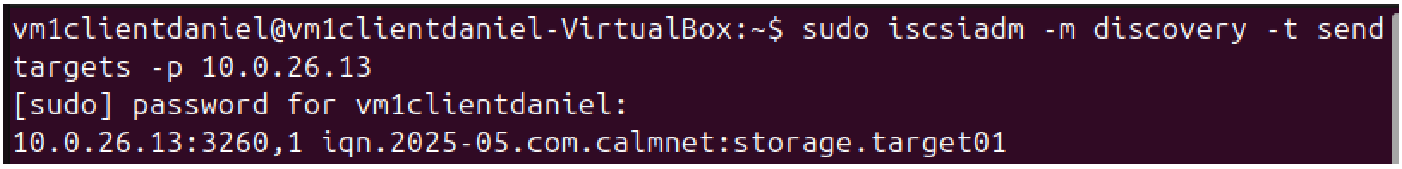
```
  GNU nano 7.2
<target iqn.2025-05.com.calmnet:storage.target01>
    backing-store /iscsi_disks/disk01.img
    initiator-address 10.0.26.0/24
    write-cache on
</target>
```

Lastly I restarted tgt , enabled it again just to be sure and checked it's status. Once I saw that it was up and running I was ready to test.



```
vm3-server-daniel@vm3-server-daniel:~$ sudo systemctl restart tgt
vm3-server-daniel@vm3-server-daniel:~$ sudo systemctl enable tgt
Synchronizing state of tgt.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable tgt
vm3-server-daniel@vm3-server-daniel:~$ sudo systemctl status tgt
● tgt.service - (i)SCSI target daemon
     Loaded: loaded (/usr/lib/systemd/system/tgt.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-05-03 21:25:52 UTC; 16s ago
       Docs: man:tgtd(8)
   Main PID: 8083 (tgtd)
```

On my DHCP client (VM1), I ensured I switched my adapter back to internal network and reapplied my netplan. After this was done I used isciadm to discover the target (VM3) 10.0.26.13, I could then see that the target was successfully discovered after entering my password, telling me iSCSI was working properly



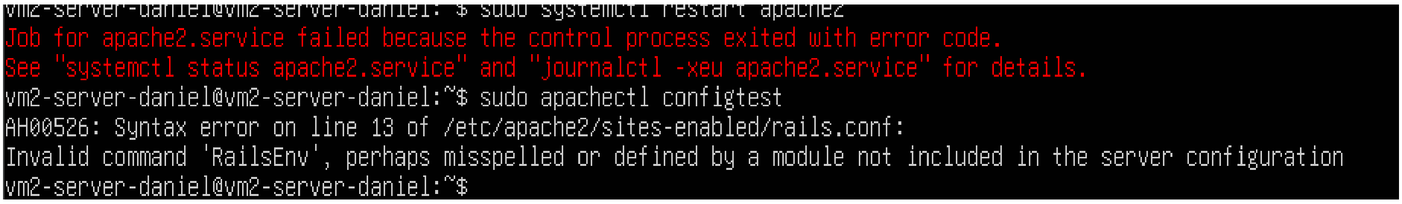
```
vm1clientdaniel@vm1clientdaniel-VirtualBox:~$ sudo iscsiadm -m discovery -t send
targets -p 10.0.26.13
[sudo] password for vm1clientdaniel:
10.0.26.13:3260,1 iqn.2025-05.com.calmnet:storage.target01
```

Final Conclusion/ Learning Outcome:

Throughout the course of doing this assignment I feel that I've greatly improved my skills in managing and navigating different Linux environments. I had previous experience installing and setting up features such as DHCP and SSH servers before however many of the services that I had to configure and install in this assignment were new to me. A major example of this was the difficulty that I had when setting up Ruby on Rails. This took me days to figure out and I had to start the entire process again a couple of times before understanding how important the passenger module is in making it so that Apache can host my Rails app successfully. I also felt extremely happy when I finally got my DHCP server working for VM1's client to get it's dynamic IP address from. I also feel like the experience of setting up a mail server could prove useful going forward if I was ever tasked with setting up or configuring one in a future workplace. That part in particular took me a while to understand how it worked fully but once it was complete and everything was working I was delighted with myself.

The main issues I experienced were:

I kept getting issues when trying to get Apache working, I kept receiving an error message when trying to run the job for my rails app

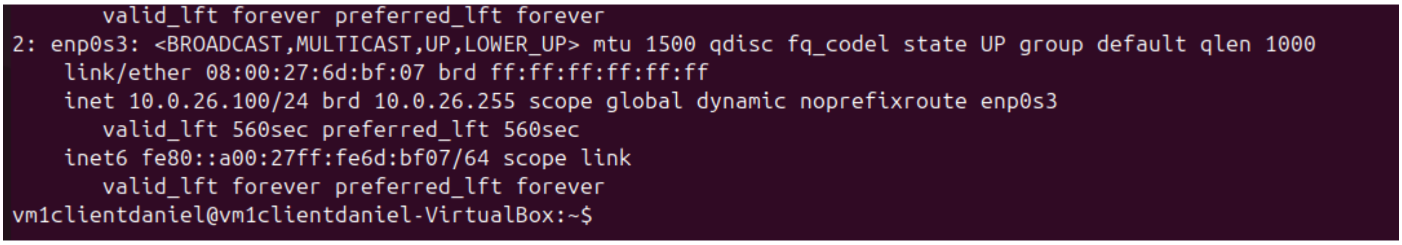


```
vm2-server-daniel@vm2-server-daniel:~$ sudo systemctl restart apache2
Job for apache2.service failed because the control process exited with error code.
See "systemctl status apache2.service" and "journalctl -xeu apache2.service" for details.
vm2-server-daniel@vm2-server-daniel:~$ sudo apachectl configtest
AH00526: Syntax error on line 13 of /etc/apache2/sites-enabled/rails.conf:
Invalid command 'RailsEnv', perhaps misspelled or defined by a module not included in the server configuration
vm2-server-daniel@vm2-server-daniel:~$
```

After days of trying to figure it out I found that using what's known as a "passenger module" fixes this error and is needed to serve Rails apps, I was missing this and so that was my issue, once I installed this and set it up everything worked perfectly.

Another issue that I had was when I was trying to make my DHCP server work my VM1 client kept getting an IP address in the prefix of 192.168.1.x/24 which is not correct as my DHCP server was set to follow the structure of 10.0.26.x/24 (26 being my student number)

I feel that I checked every possible scenario and even double checked I was on the internal network interface for VM1 which I was. I found in the end my issue was that in one of my config files for DHCP the interface was set to =ens33 which was wrong as this was for NAT, I changed it to match the interface of my internal network and my Client VM finally received an IP address in the correct range and prefix:



```
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6d:bf:07 brd ff:ff:ff:ff:ff:ff
    inet 10.0.26.100/24 brd 10.0.26.255 scope global dynamic noprefixroute enp0s3
       valid_lft 560sec preferred_lft 560sec
    inet6 fe80::a00:27ff:fe6d:bf07/64 scope link
       valid_lft forever preferred_lft forever
vm1clientdaniel@vm1clientdaniel-VirtualBox:~$
```

What I enjoyed

One major thing I enjoyed about doing this assignment was seeing each service come together and finally work after configuring everything as carefully as I could. I found a lot of frustration troubleshooting the issues that I already mentioned but I am proud of myself that I didn't give up and kept troubleshooting until I found a fix. It was worth the feeling when I finally got the whole thing working.

Another thing I found enjoyable about this assignment was messing around with cockpit management system on my managing server, I found it interesting to be able to monitor things like failed services, system health , logs and server statuses through a browser's interface which is a sort of convenience you don't get much in a Linux based setup.

Finally setting up the mail server was another thing that stood out to me, I found it confusing to set up at first but once it was all working it was interesting seeing how mail services work, being able to send and receive messages on a service that I configured by myself made me feel really proud of myself.

Overall this assignment has been a good learning experience for me, and I feel that I have a better understanding of how to set up and use all of the services I have covered.

Bibliography

[i] Linode. (2018, March 12). Install Ruby on Rails with Apache on Debian 9. *Linode Guides & Tutorials*. https://www.linode.com/docs/guides/ruby-on-rails-apache-debian/#configure-apache-to-work-with-passenger

[ii] Rbenv. (n.d.). *GitHub - rbenv/rbenv: Manage your app's Ruby environment*. GitHub. https://github.com/rbenv/rbenv

[iii] *31. Install BIND*. (2024, November 10). https://www.doublebastion.com/install-bind/

[iv] *Primary vs secondary DNS | Cloudflare*. (n.d.). https://www.cloudflare.com/en-gb/learning/dns/glossary/primary-secondary-dns/

[v] Configuring Open SSH (2024, April 29). *Configuring OpenSSH Server*. Oracle Help Center. https://docs.oracle.com/en/operating-systems/oracle-linux/openssh/openssh-ConfiguringOpenSSHServer.html

[vi] *How can I setup SSH so that it is restricted to my local network?* (n.d.). Ask Ubuntu. https://askubuntu.com/questions/115940/how-can-i-setup-ssh-so-that-it-is-restricted-to-my-local-network

[vii] *Install and configure a CUPS print server*. (n.d.). Ubuntu Server. https://documentation.ubuntu.com/server/how-to/networking/cups-print-server/index.html

[viii] Kenlon, S. (2021, August 21). *How to set up your printer on Linux*. Opensource.com. https://opensource.com/article/21/8/add-printer-linux

[ix] Formy-Duval, A. (2020, November 11). *How I use Cockpit for my home's Linux server management*. Opensource.com. https://opensource.com/article/20/11/cockpit-server-management

[x] Asif, S. (2024, November 14). Setting up a DHCP server on Ubuntu | Medium. *Medium*. https://medium.com/@sydasif78/setting-up-a-dhcp-server-on-ubuntu-a-guide-for-network-engineer-d620c5d7afb2

[xi] *Install and configure Dovecot*. (n.d.). Ubuntu Server. https://documentation.ubuntu.com/server/how-to/mail-services/install-dovecot/index.html

[xii] Sooraj, & Sooraj. (2024, June 8). *Install and configure ISCSI storage Server on Ubuntu 22.04 | VPSie tutorials*. Public & Private Data Sovereign Cloud Automation. https://vpsie.com/knowledge-base/install-and-configure-iscsi-storage-server-on-ubuntu-22-04/