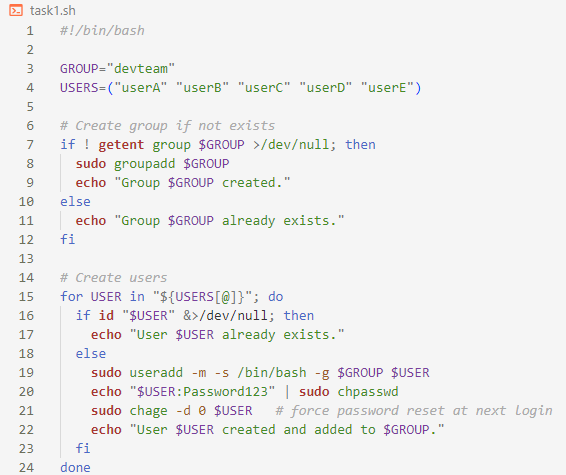
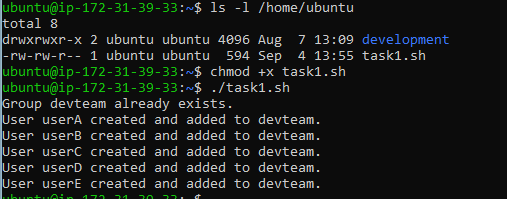
Ahmad Dangana  
010240383

**Task 1**

* Create a script file task1.sh: 
* Upload the file to the server then reconnect 
* Confirm the existence of the file then make it executable using chmod +x task1.sh and run

Task 2:

* Create the shared directory with sudo mkdir /shared\_data

****

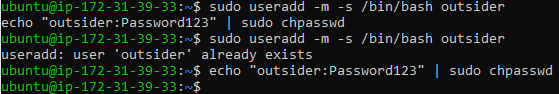
* Set group ownership to devteam



* Set group permissions (read + write, deleting other users files not allowed)



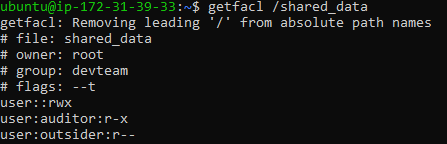
* Create an extra user outside the group



* Grant outsider read-only access via ACL

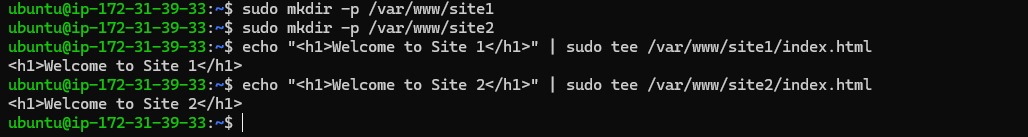


* Confirm it worked



Task 3:

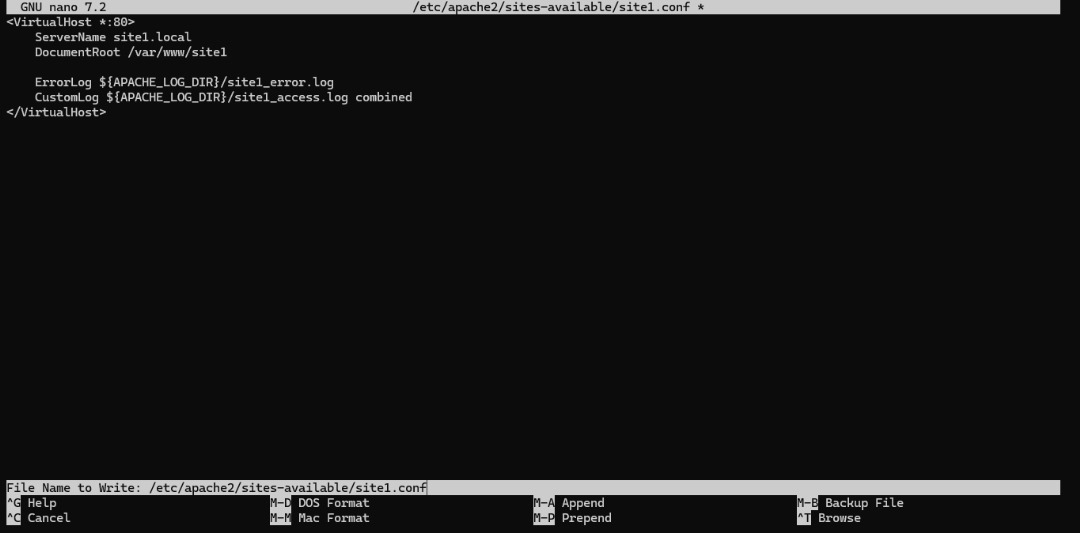
* Create document roots and add sample index



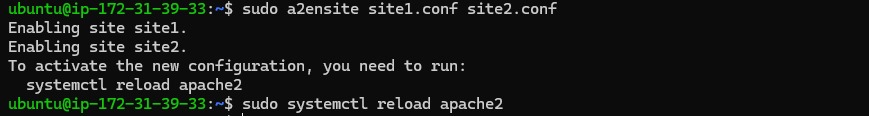
* Configure virtual hosts for both sites one after the other



Add this for each

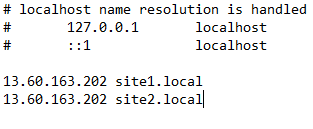


* Enable the sites



* Update local host files in the PC



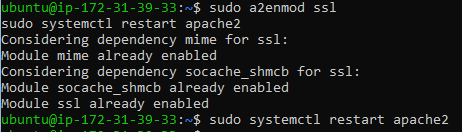


* Test in browser



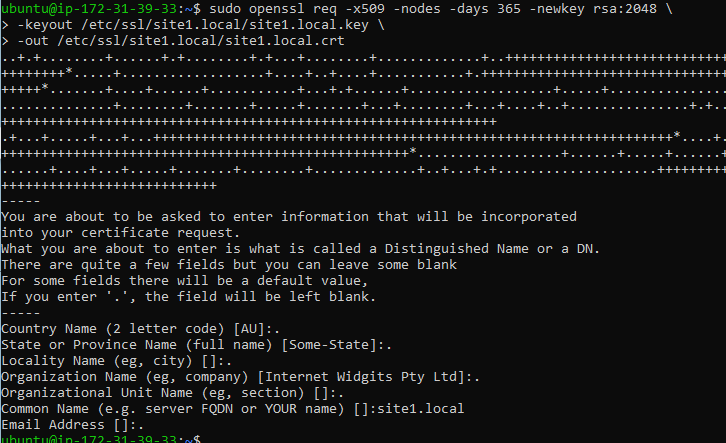


Task 4:

* Enable SSL module in Apache
* Create directories for certificate

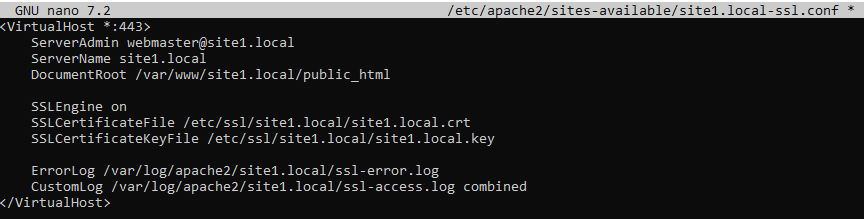


* Generate a self-signed SSL certificate

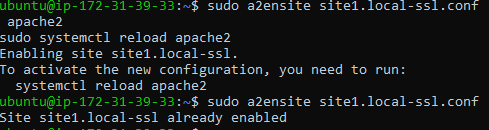


* Create an HTTPS Virtual Host for site1



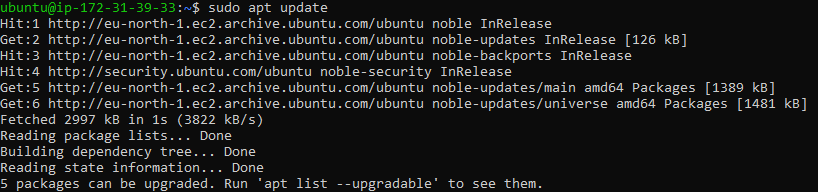


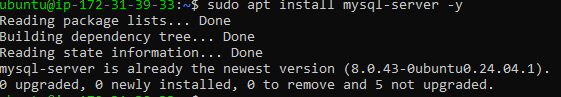
* Enable the SSL site



Task 5:

* Install MySQL Server





* Configure MySQL to Listen on All Interfaces





* Restart MySQL



* Login to MySQL



* Create DATABASE projectdb



* Create a Least-Privilege User





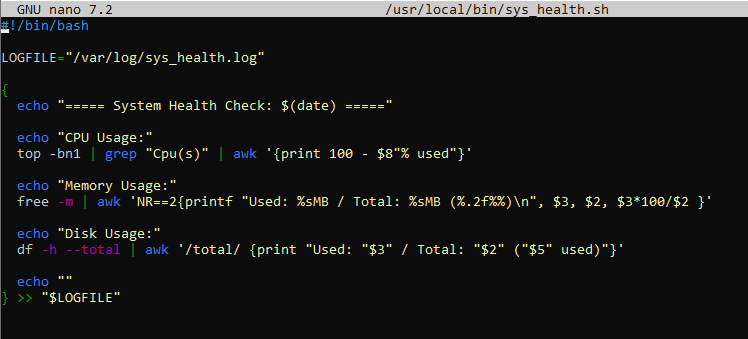


Task 6:

Task 7:

* Open the script file in an editor and add the script content





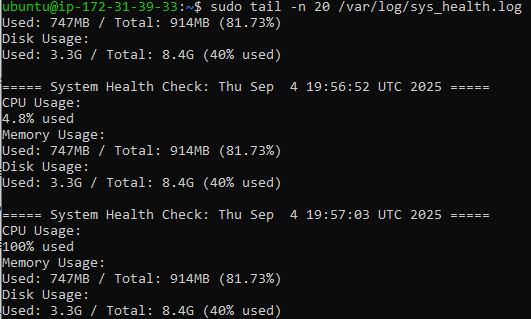
* Make the script executable



* Test run the script

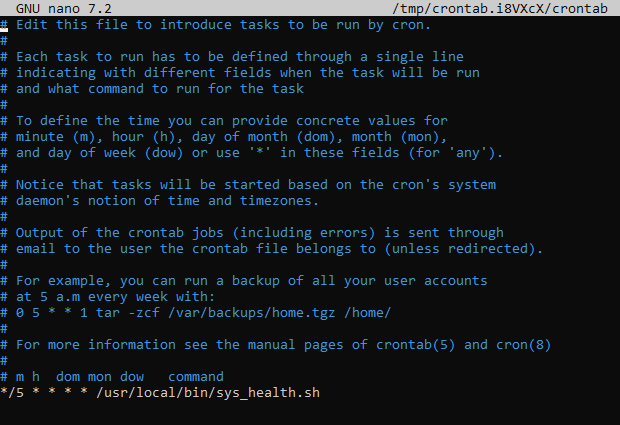


Check if the log file was created



* Set it as a cron job (runs every 5 mins)

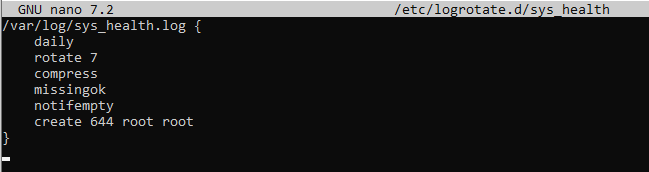




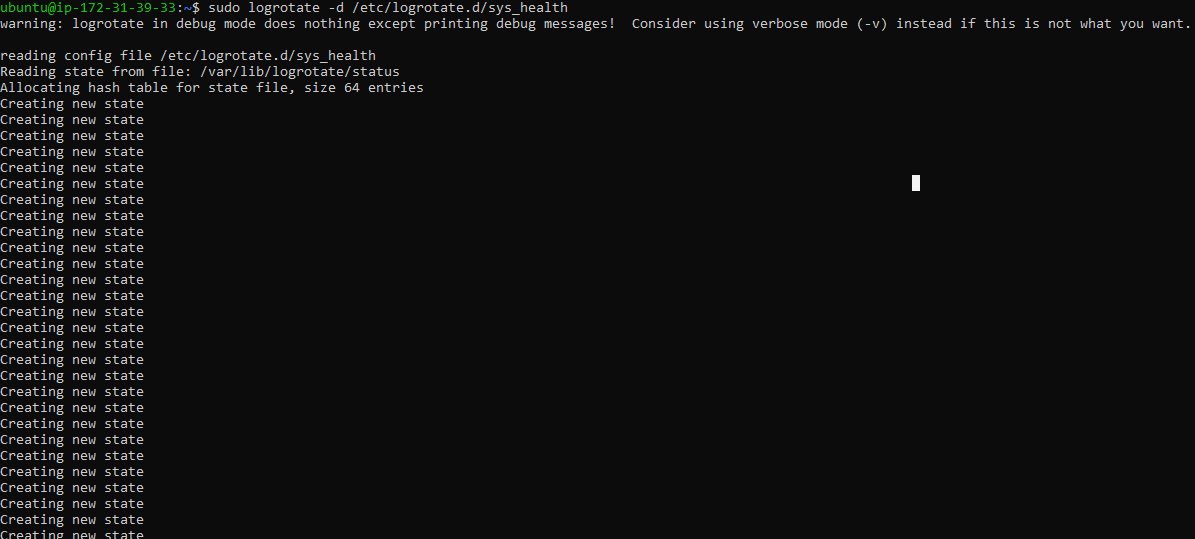
Task 8:

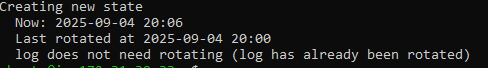
* Create a Logrotate Config File and add script content





* Test the Logrotate Configuration





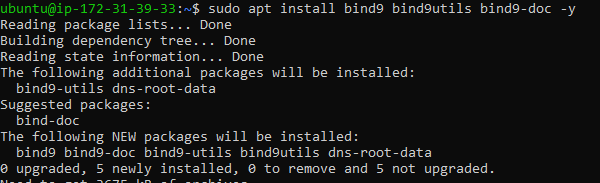
* Verify Rotation

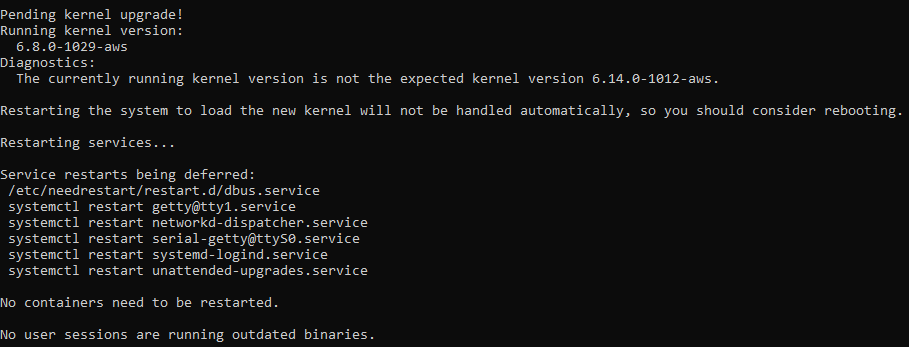


Task 9:

* Install Bind9





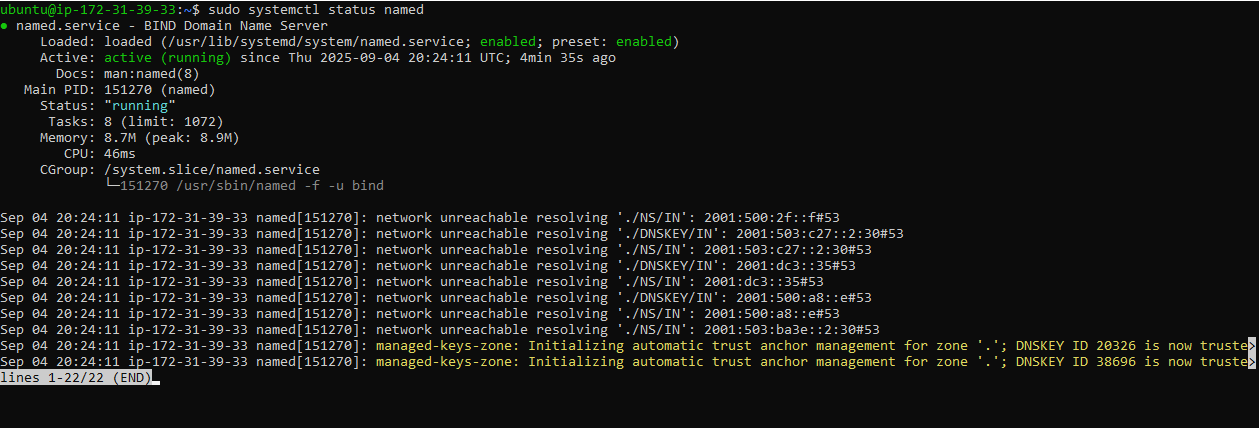


* Enable and start BIND9





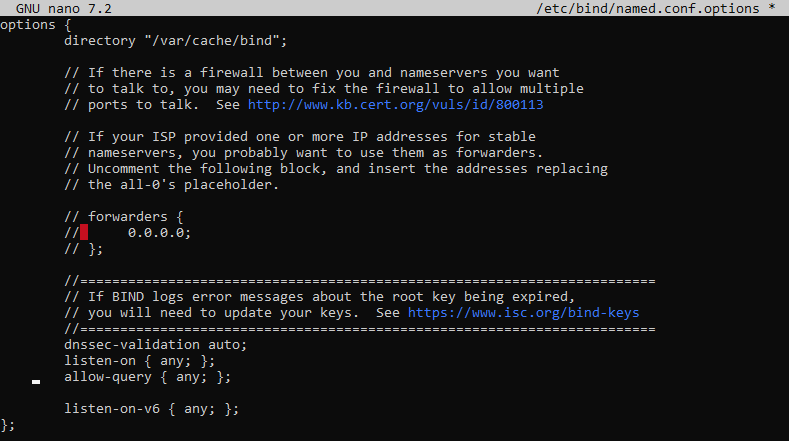
* Check the status after starting



* Configure as a Caching DNS Server by editing the options with listen-on { any; }; and

allow-query { any; };





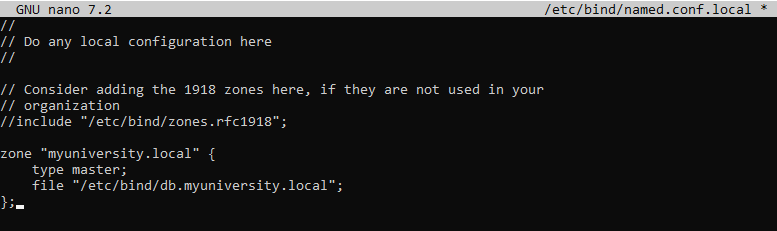
* Restart BIND9



* Define a New Zone for myuniversity.local

Edit zone configuration

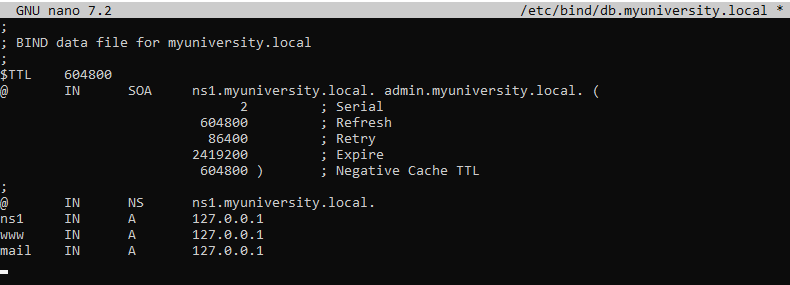




* Create the Zone File



* Edit the new file



* Check Config & Restart BIND9





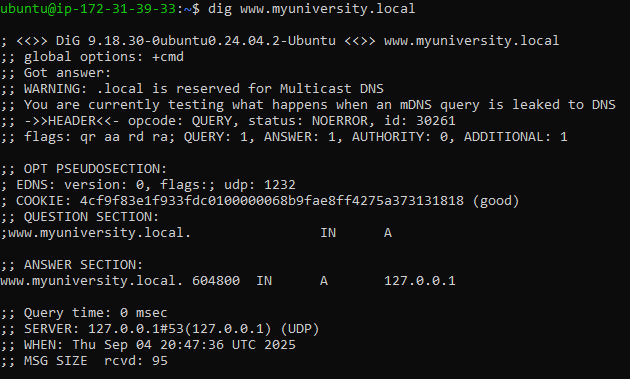


* Test DNS Resolution

Edit nameserver in /etc/resolv.conf:



Test Resolution:

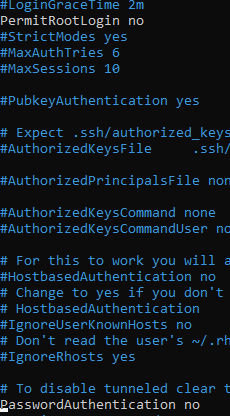


Task 10:

* Open the SSH server config file:



* Edit SSH Configuration



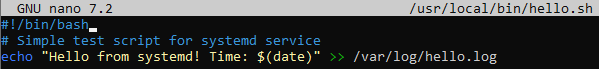
* Restart SSH



Task 11:

* Create a Simple Script



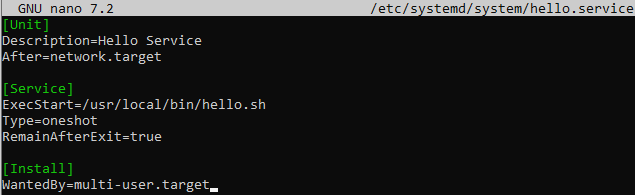


* Make it executable

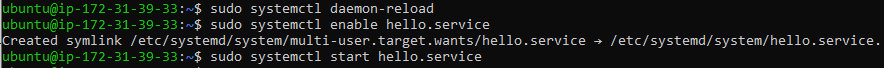


* Create a Systemd Service File

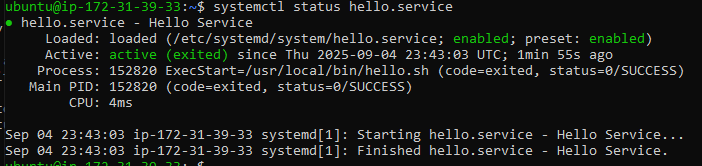




* Reload Systemd & Enable Service



* Verify Service

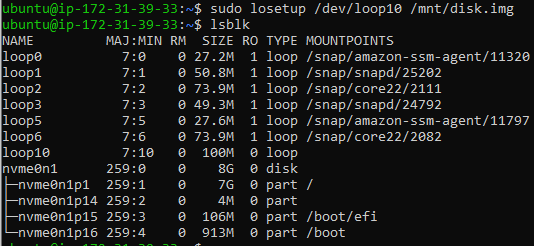


Task 12:

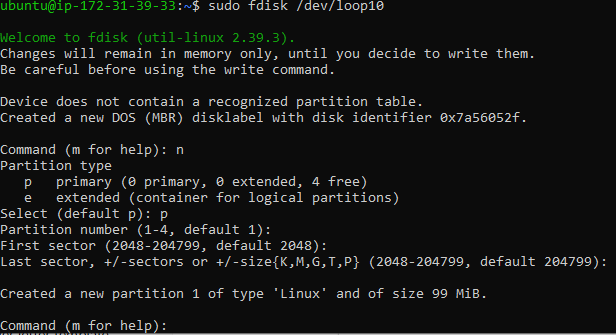
* An AWS EC2 instance, your root disk is usually small (8 GB) so I create a Fake Disk (Loopback File)



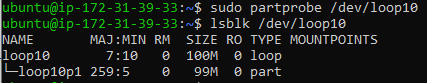
* Turn It Into a Block Device and confirm



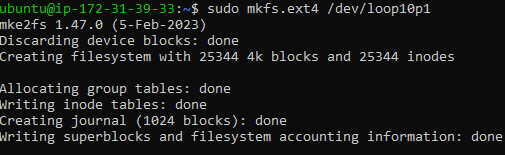
* Partition the Disk



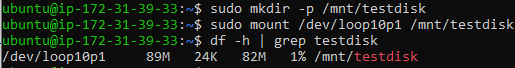
* Refresh Partition Table and check for new partition



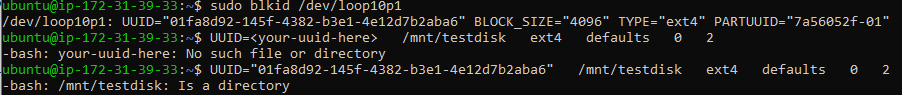
* Format Partition



* Mount Partition

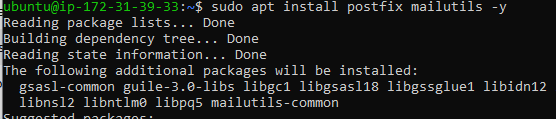


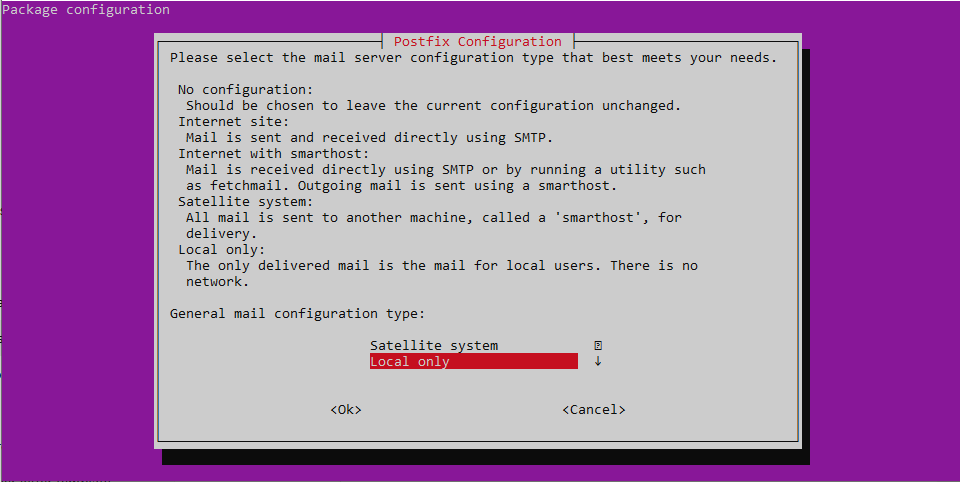
* Get UUID and add to /etc/fstab for persistence

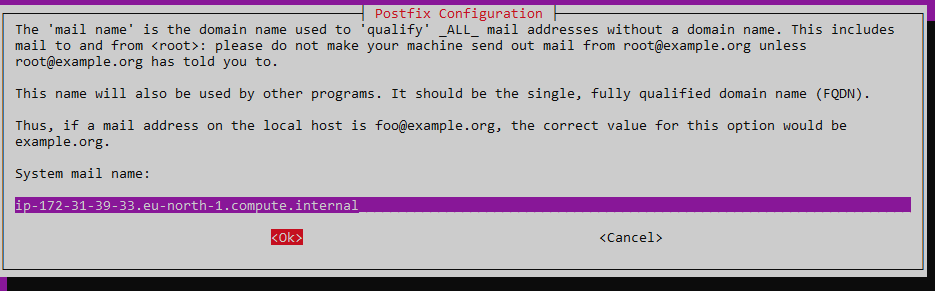


Task 13:

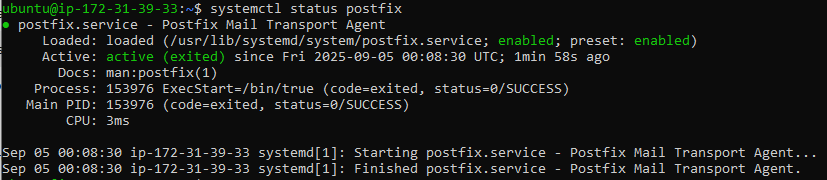
* Install Postfix and configure during installation



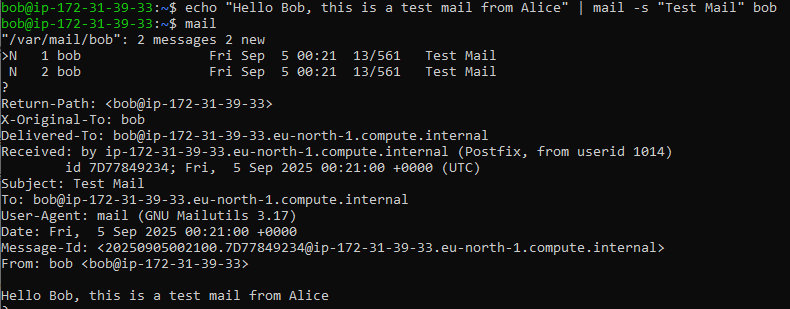




* Verify Postfix is Running



* Send a Test Mail and check user inbox



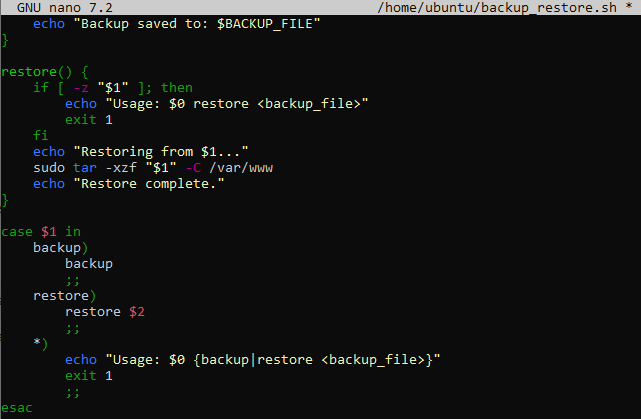
Task 14:

* Create Backup Directory



* Create the Script

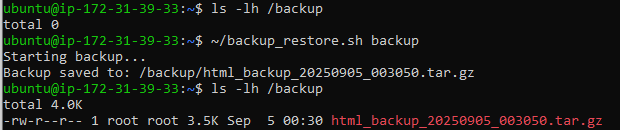




* Make It Executable



* Run script and check backups



* Simulate data loss

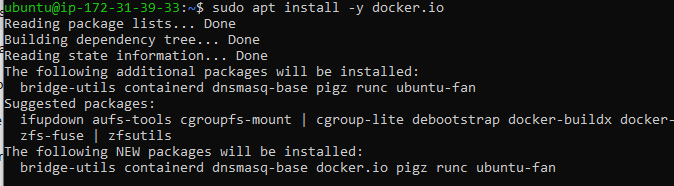


* Restore from backup

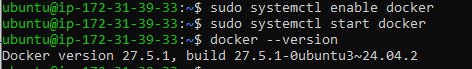


Task 15:

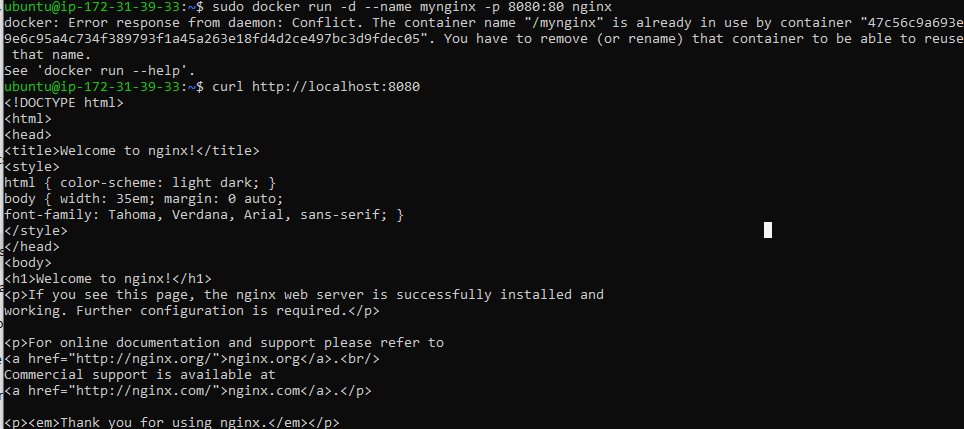
* Install Docker



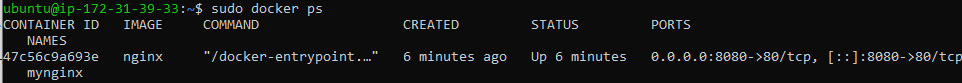
* Enable, start Docker and check version



* Run Nginx Container on Port 8080



* Verify Container is Running



* Test Nginx

