Linux Server Deployment & Configuration Within VirtualBox

Adriel Sanchez

June 2025

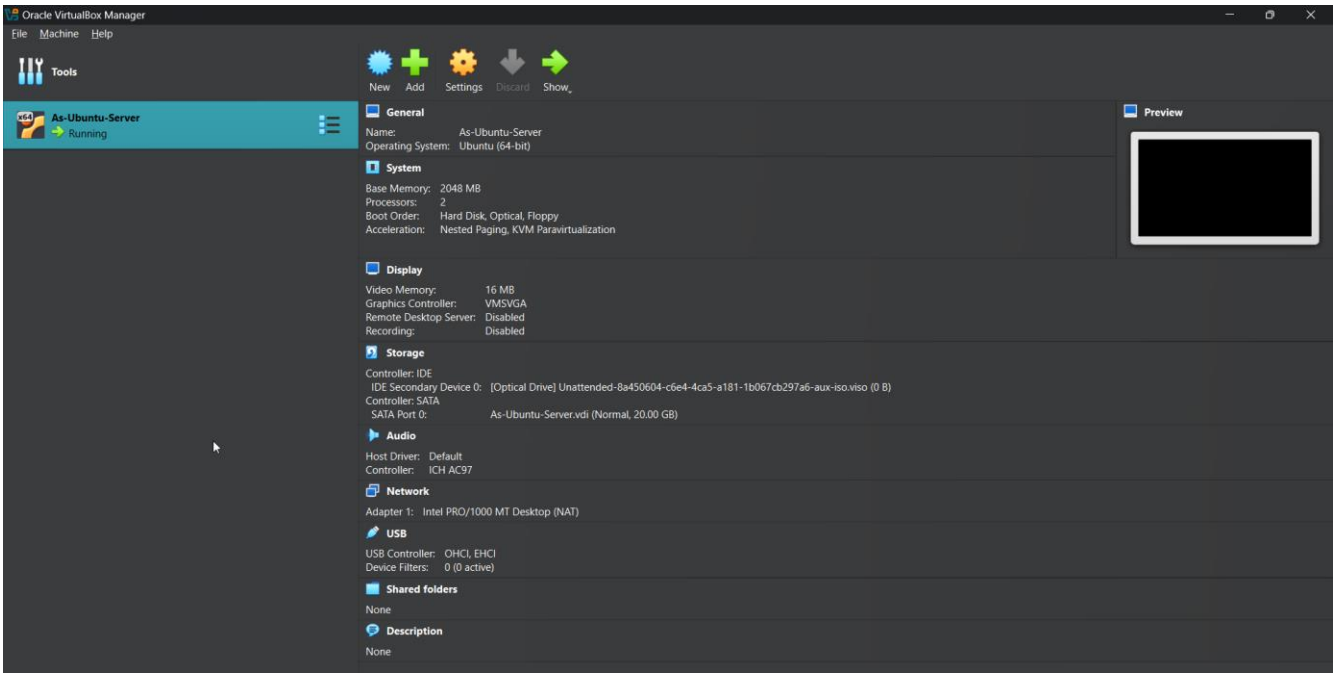
Summary:

This documentation covers the step-by-step process of setting up a Linux server in a VirtualBox environment. It includes downloading and installing Ubuntu Server, configuring basic packages and system upgrades, setting up essential tools, adding users, configuring a firewall, enabling SSH access, enhancing security with Fail2Ban, installing a GUI and Docker, and deploying Portainer and Netdata containers. Screenshots and commands are provided throughout to support replication.

Skills Demonstrated:

- Virtualization
- Linux server installation and configuration
- Package management with APT
- User and permission management
- Firewall configuration with UFW
- SSH setup and remote access using PuTTY
- Fail2Ban setup for intrusion prevention
- GUI installation and configuration
- Docker and container management
- Monitoring tools (Neofetch, htop, iptraf, Netdata)
- Networking concepts (port forwarding, local-only access)

1. First i Download VirtualBox & Ubuntu server iso to begin my project.



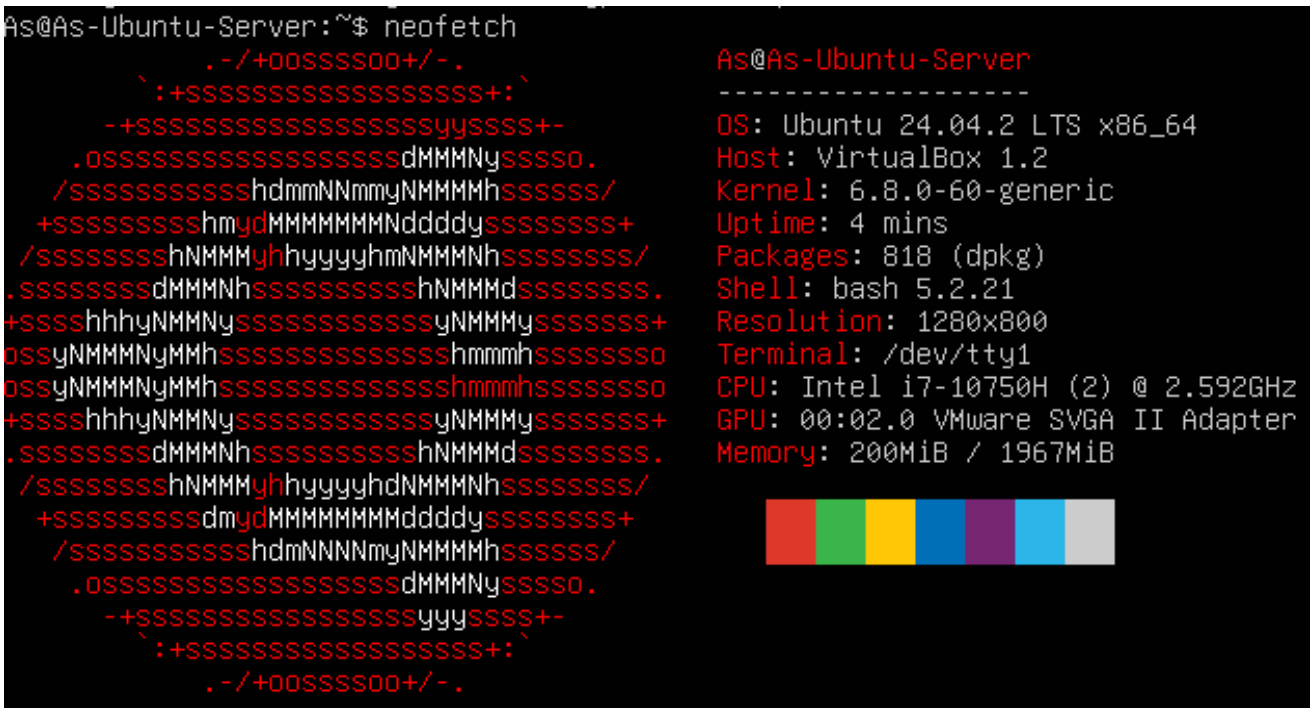
2. The next step was to Update the environment to begin working on it, i used the following comands:

- sudo apt update && sudo apt upgrade -y
- Sudo apt install unattended-upgrades
- Sudo dpkg-reconfigure –priority=low unattended-upgrades

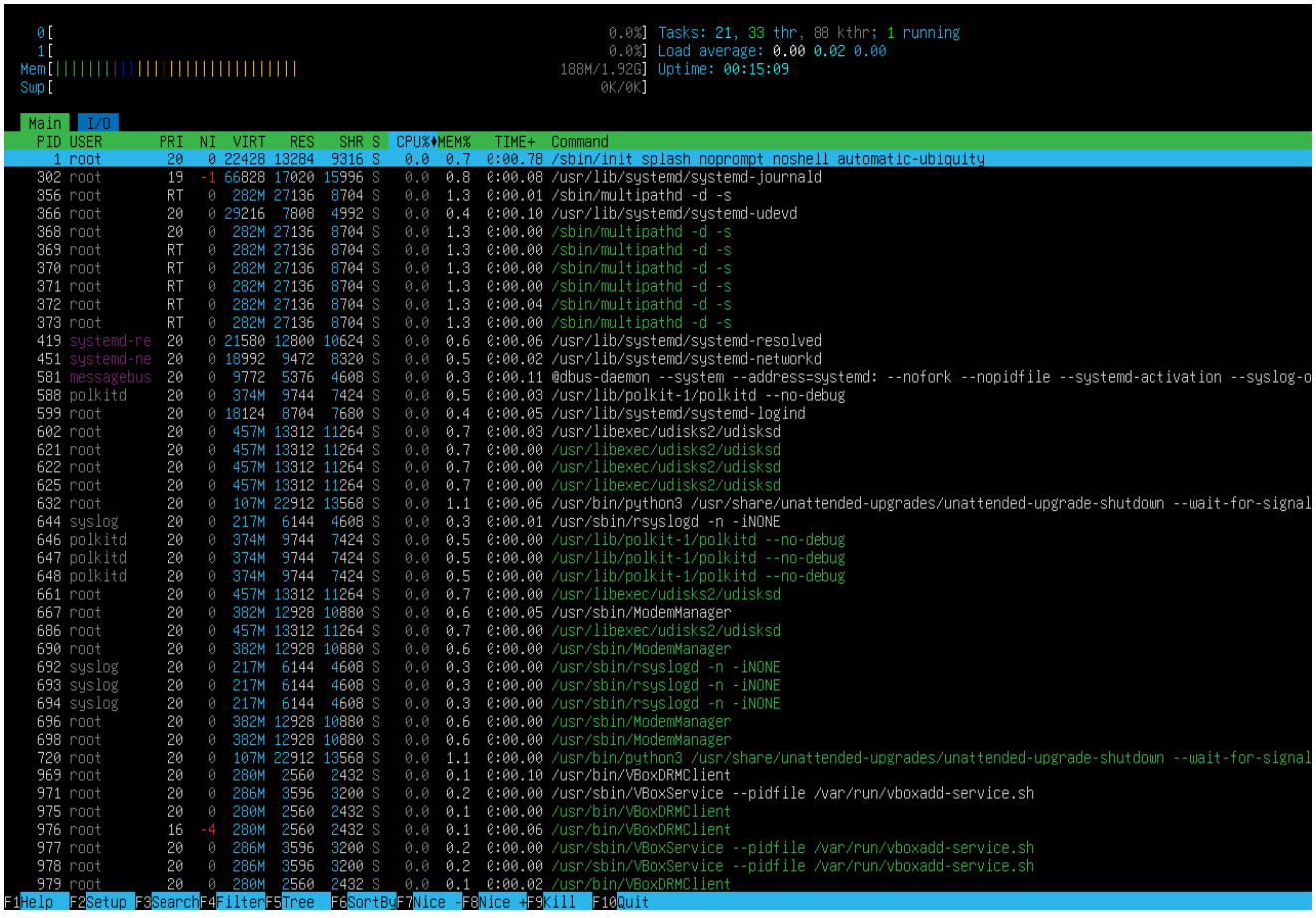
3. Then I installed several tools and repositories to enhance system monitoring, file transfers, graphical support, access to extra software and user conveinece.

-neofetch, htop, Curl, wget, xorg, tldr, dust, Mulitverse/Universe and restricted extras

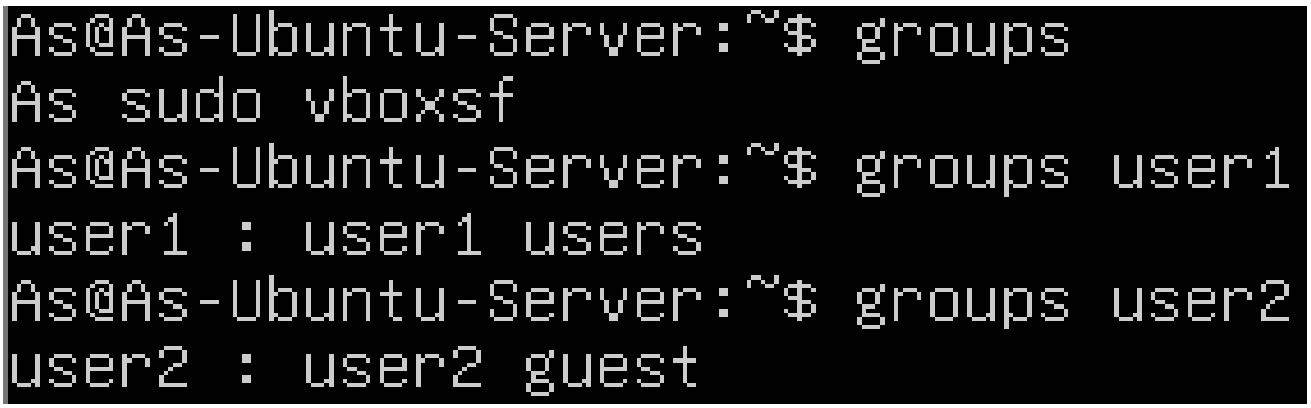
Neofetch



htop



4. Next created 2 other users and added them to groups to replicate the management of permisions and access control.



5. Then i proceeded to Attached Guest additions to enable better integration between my host and Vm using the following steps:

   -In virtualBox attached the guest additions iso

   -in the VM menu Display > insert Guest additions Image

   -sudo apt install build-essential dkms linux-headers-$(uname -r)

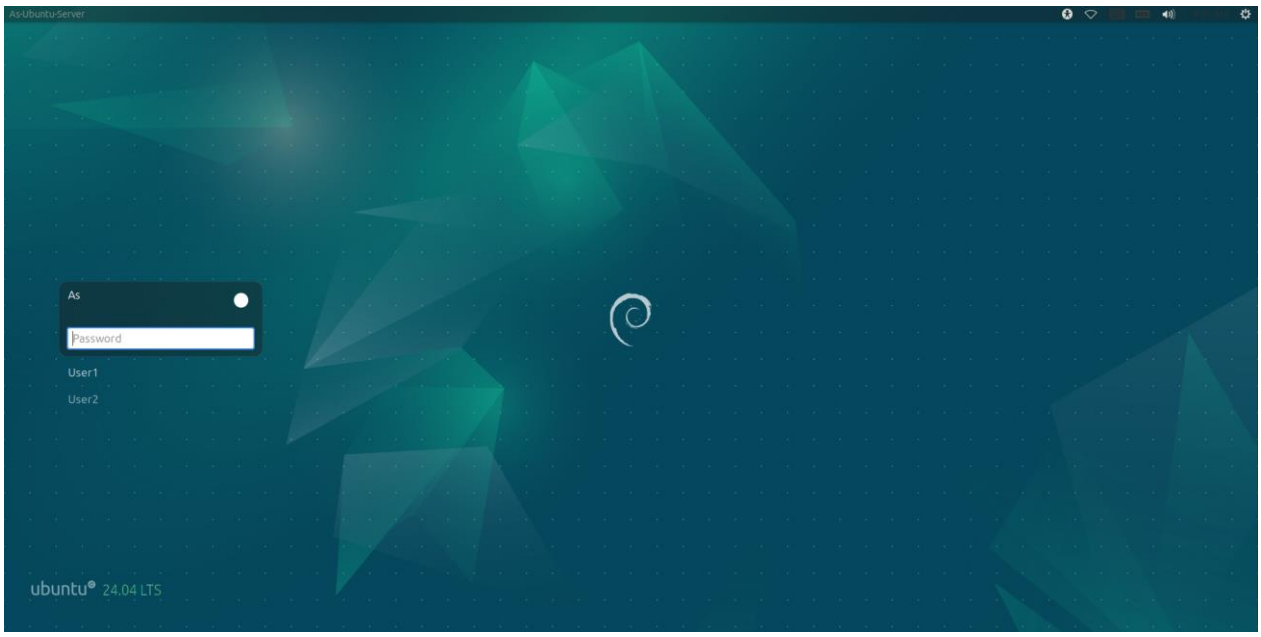
   -sudo mkdir -p /mnt/cdrom

   -sudo mount /dev/cdrom /mnt/cdrom

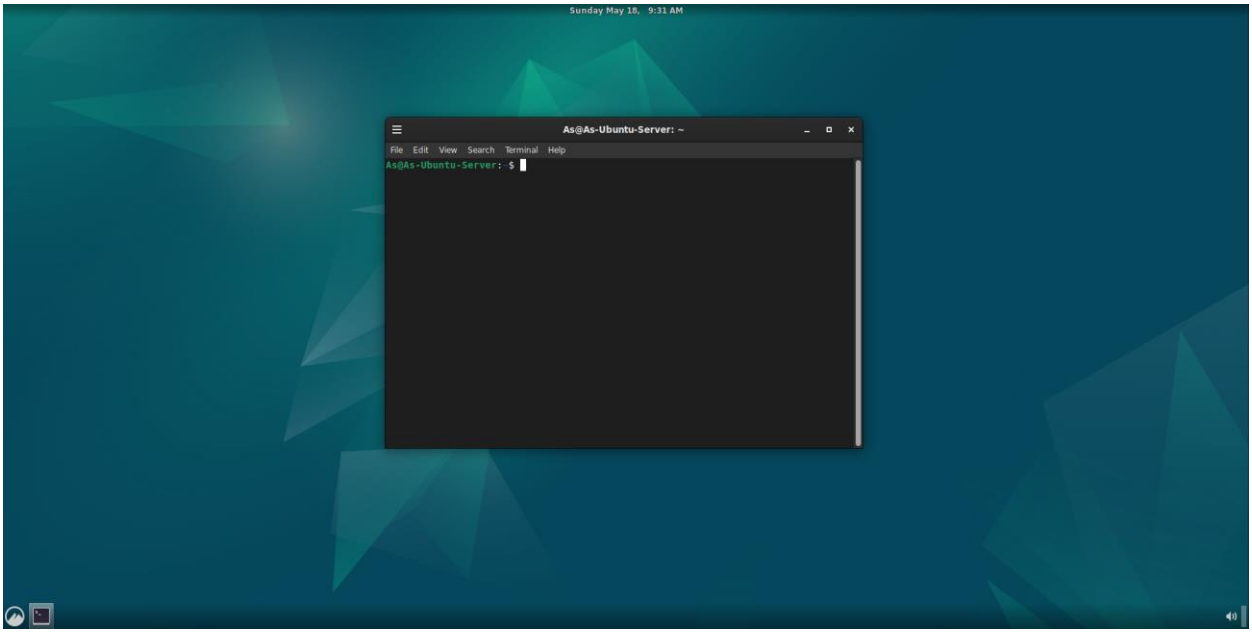
   -sudo sh /mnt/cdrom/VBoxLinuxAdditions.run

6. Next Installed a Gui (cinnamon core) and a display manager (lightdm) to make it user friendly.

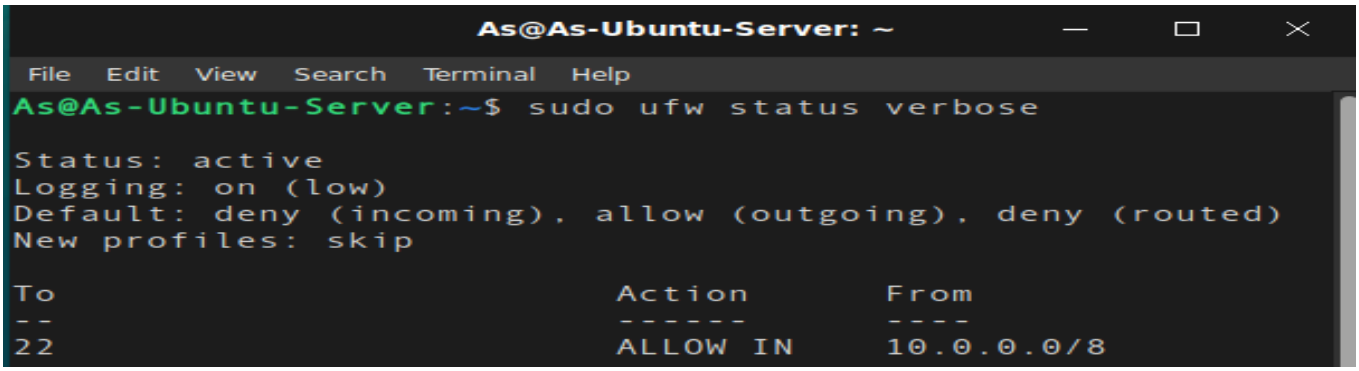
Lightdm



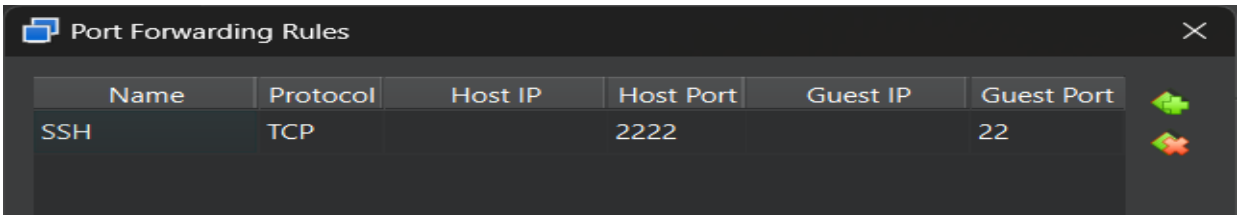
Cinnamon-core

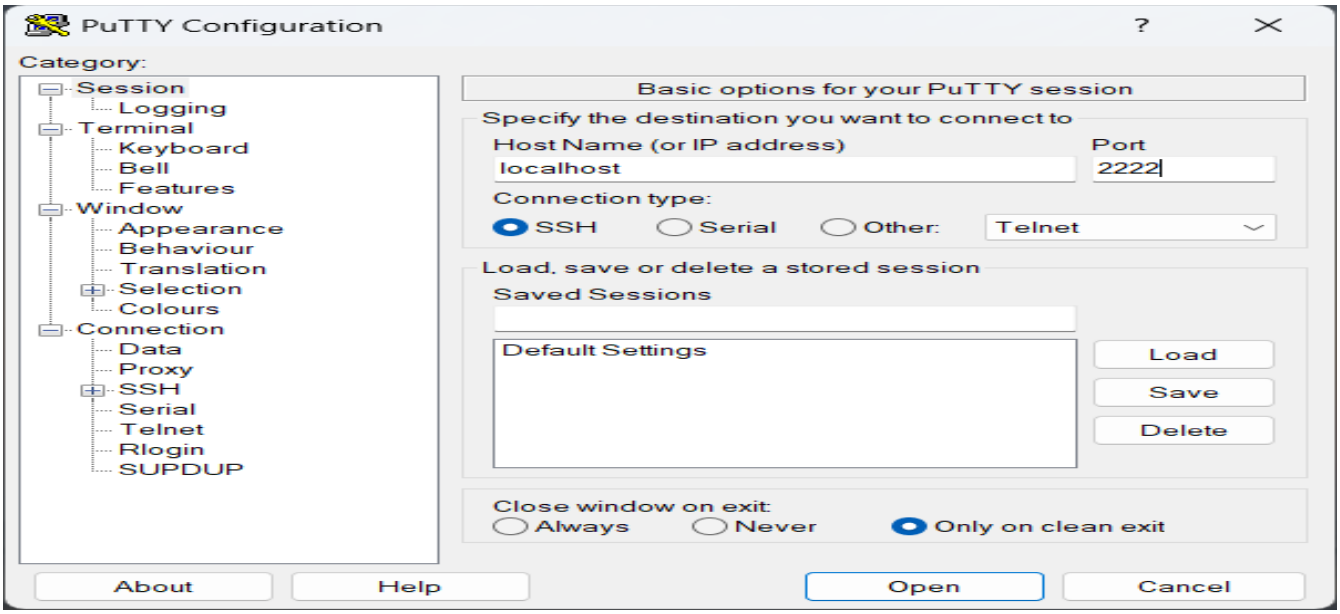


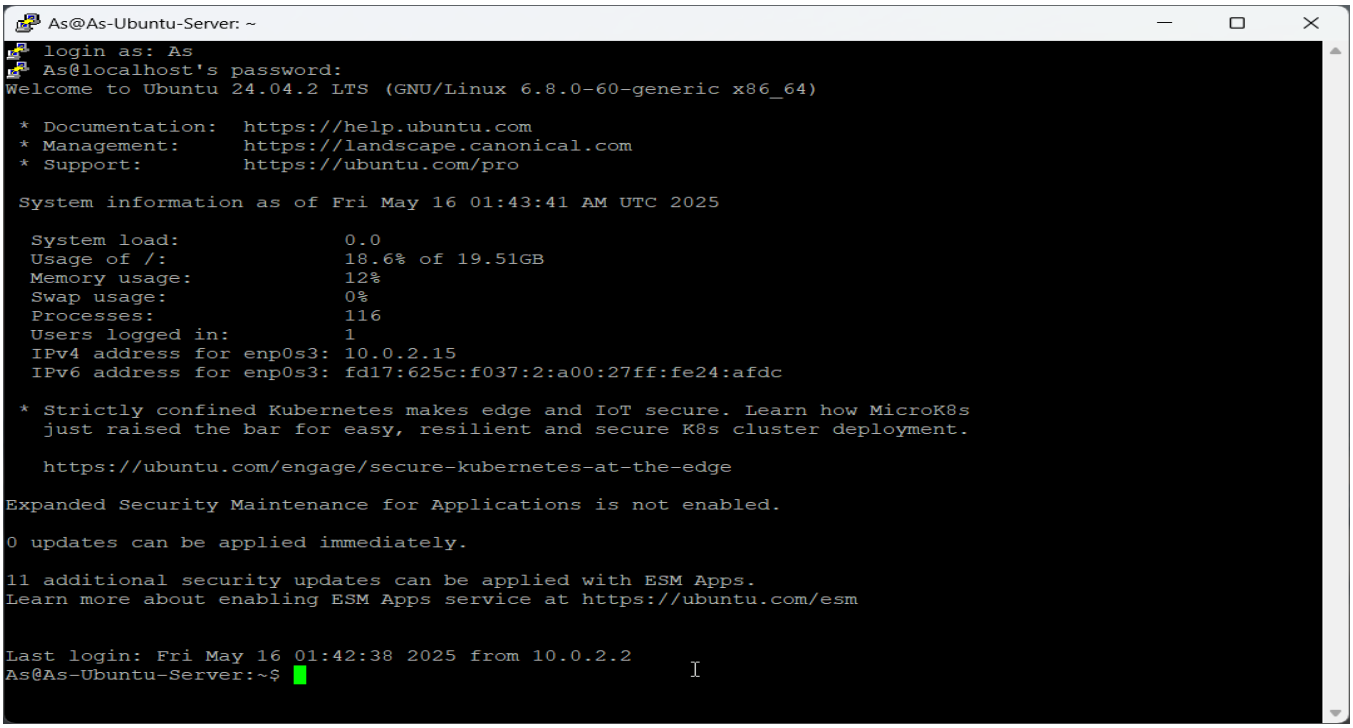
7. Next i Configured the firewall to Deny incoming, allow outgoing, allow ssh from local network, logging on to controll network traffic.



8. Then i proceeded to Install OpenSSH set up port forwarding in VirtualBox then shelledin from my host to the Vm through putty to securely manage the server remotley







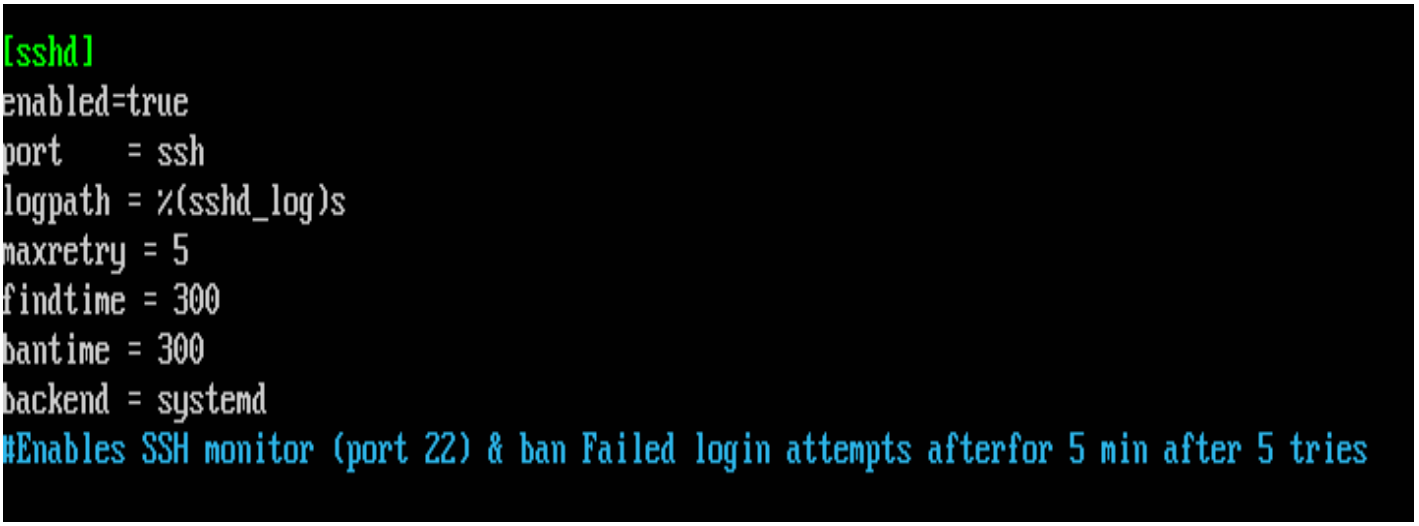
9. Then i installed and configured Fail2ban to guard against intrusion Attempts. using the following comands i installed, configured and tested it:

-Sudo apt install fail2ban -y

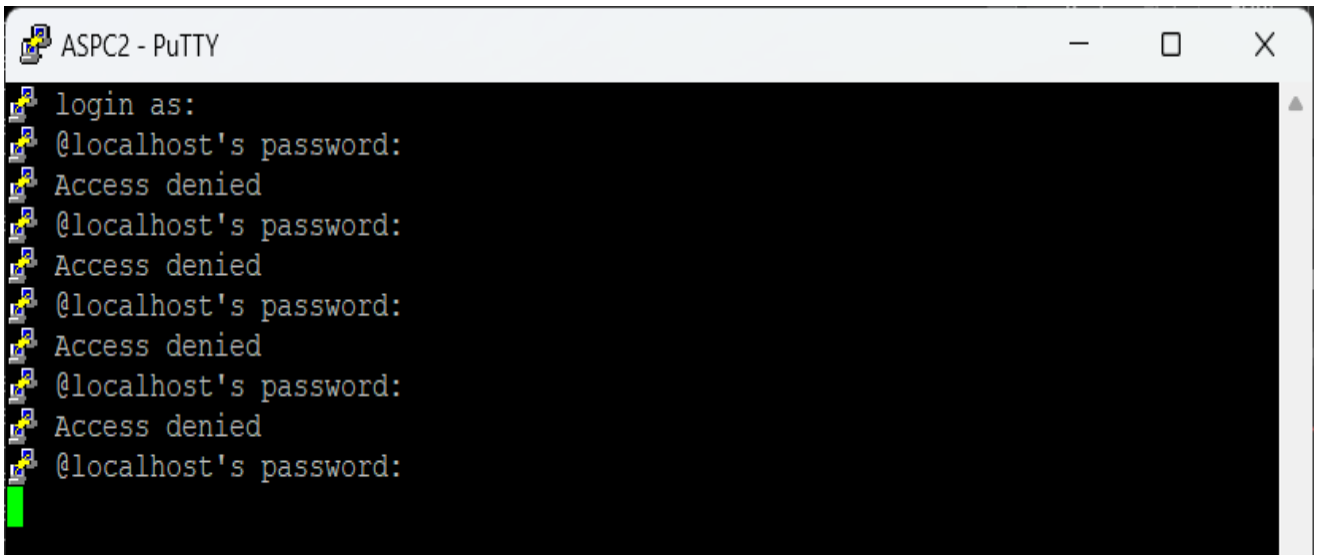
-sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local

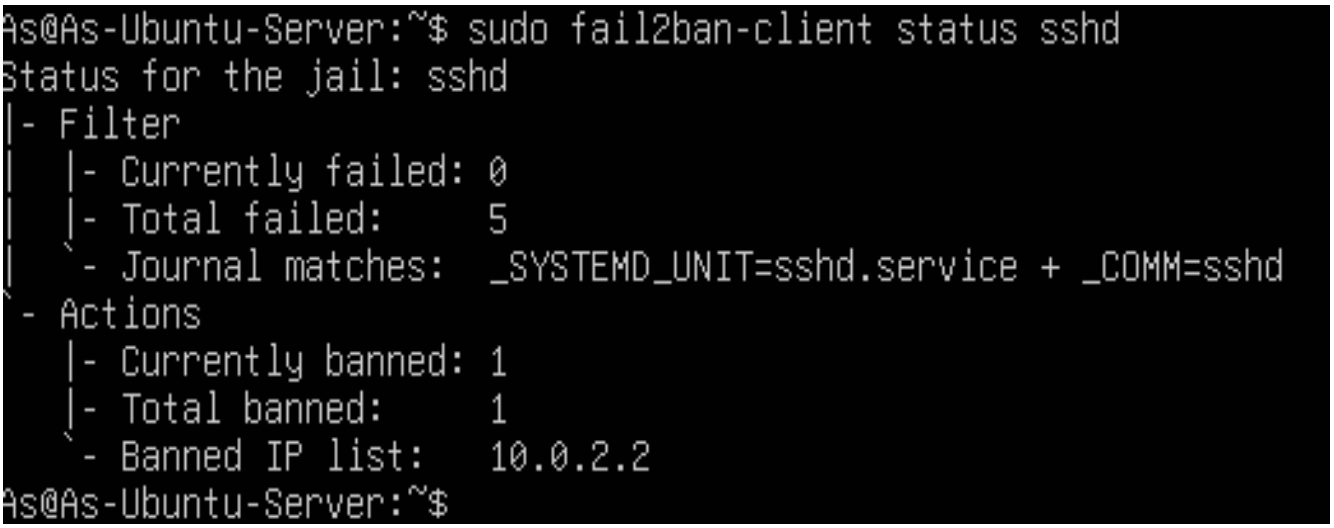
-sudo nano /etc/fail2ban/jail.local

I Made this edit in the jail.local file under sshd

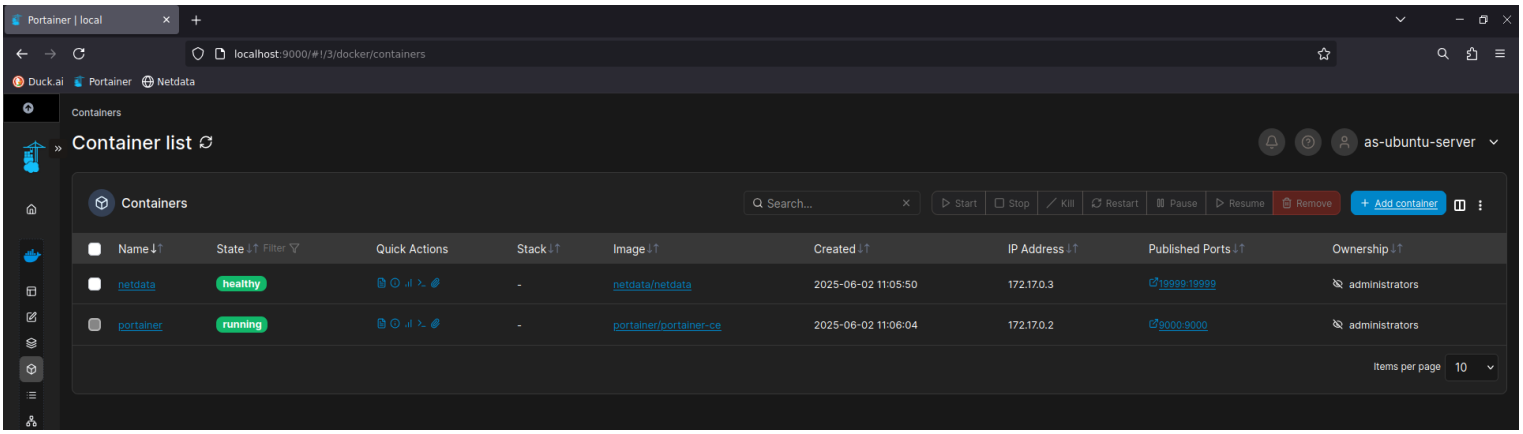


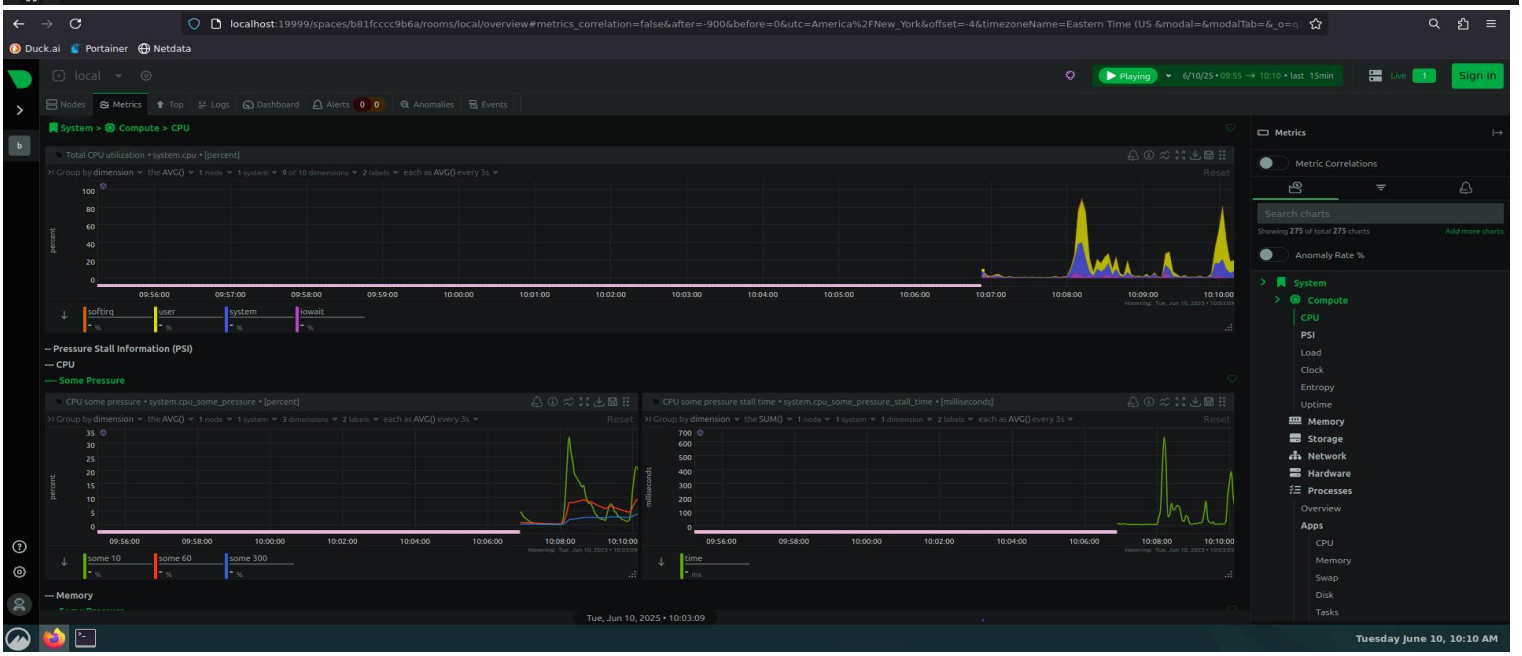
I Incorrectly input credentials then viewed the sshd status on Server to test



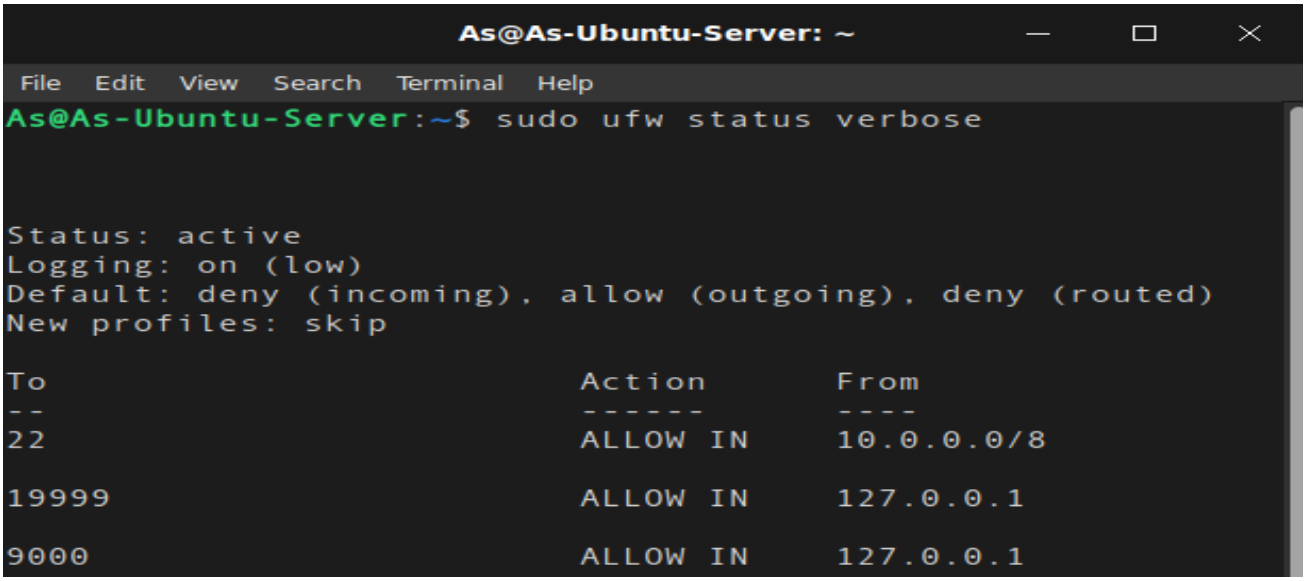


10. Next installed docker and set up a container with portainer then another for netdata. for a user friendly Gui for monitering systems and container performace in real time.

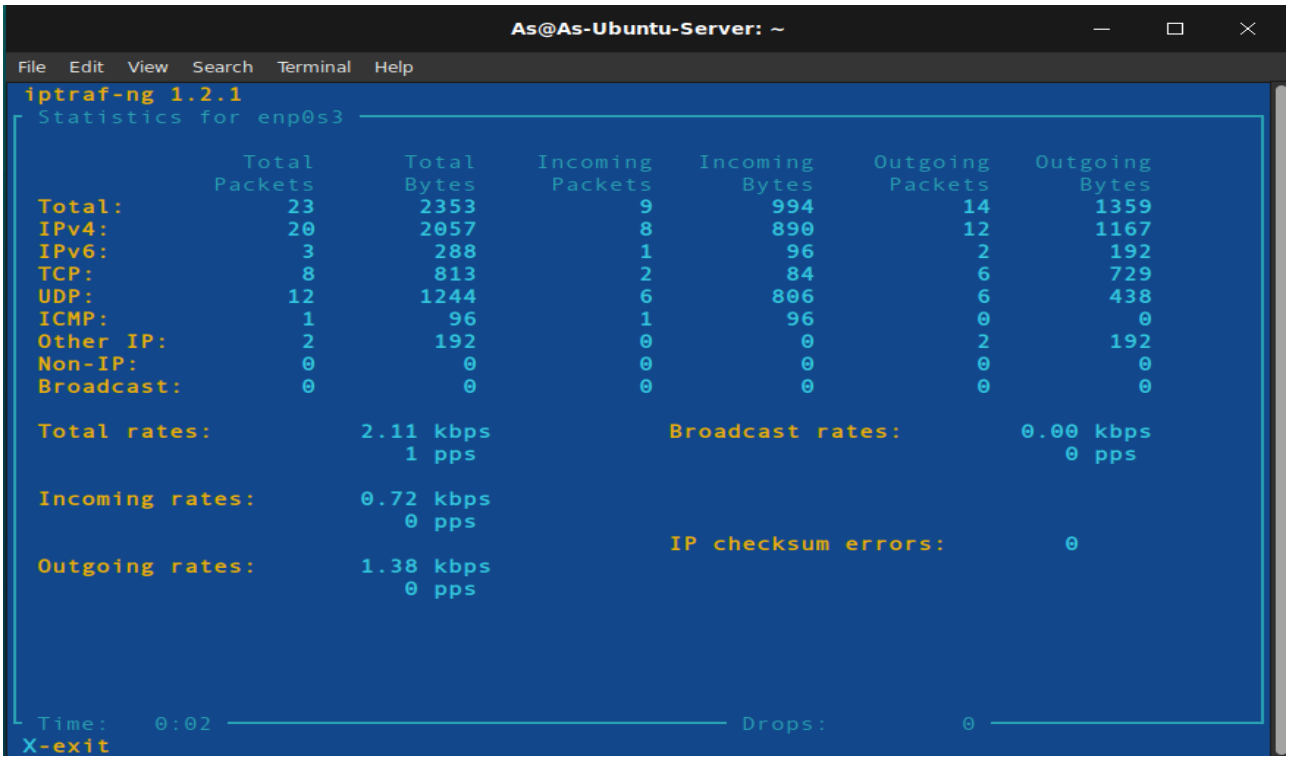




11. Then proceeded to updated firewall Rules to only allow local host to access the containers to restrict external connections and enhance security.



12. Finally Installed iptraf to monitor packets in real time.



After Action Summary:

I started up my project by installing VirtualBox and Ubuntu Server, I configured the VM, updated the system, and installed key tools and repositories for monitoring, file transfer, and user convenience. I created additional users for permission management, integrated Guest Additions, and installed a GUI with LightDM for easier use. Security was enhanced by configuring the firewall, enabling remote management via OpenSSH, and setting up Fail2ban. I deployed Docker containers for Portainer and Netdata, restricted container access to localhost, and concluded by adding Iptraf for real-time network monitoring.