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:

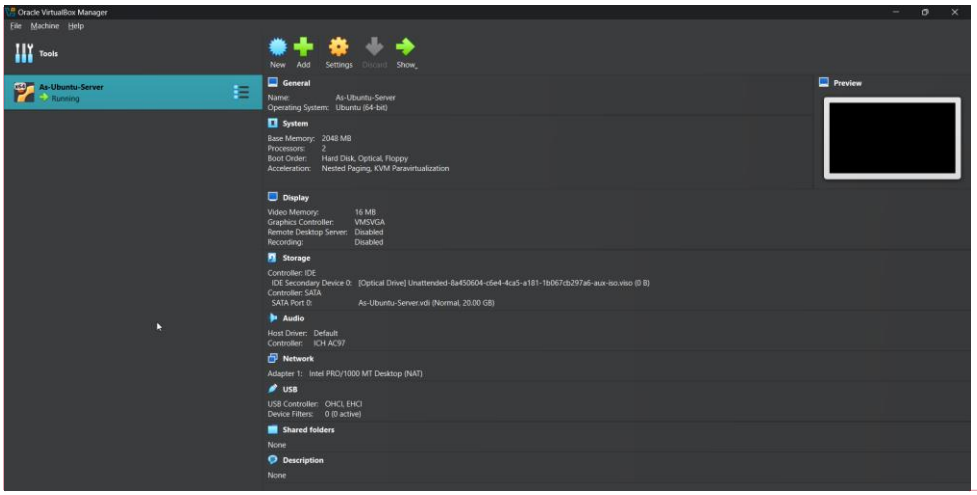
- Virtualization
- Linux server installation and configuration
- Package management with APT
- User and permission management
- Firewall configuration with UFW
- SSH setup and remote access using PuTTY
- Fail2Ban setup for intrusion prevention
- GUI installation and configuration
- Docker and container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

1. Download VirtualBox & Ubuntu server iso

Commented [AS1]: https://www.virtualbox.org/wiki/Downloads

https://ubuntu.com/download/server/thankyou?version=24.04.2&architecture=amd64&lts=true

Created a new VM with the iso file



Commented [AS2]: My virtual machine and the settings.

2. Updated the environment to begin working on it

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

Commented [AS3]: update the package index, upgrade installed packages, install the unattended-upgrades package for automatic updates, and configured it to manage updates.

3. Installed neofetch, htop and other extras and tools

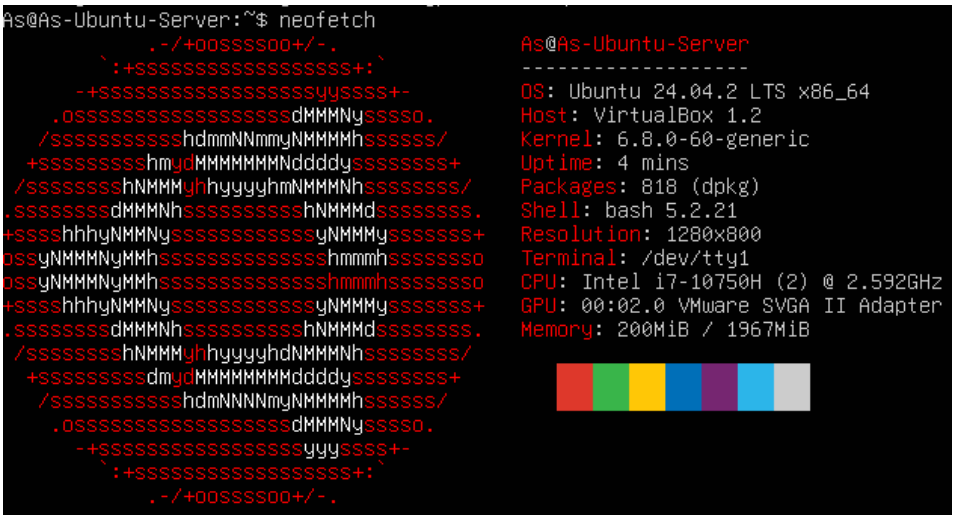
-add-apt-repository universe, add-apt-repository multiverse, ubuntu-restricted-extras

-Curl, wget, xorg, tldr, dust

Commented [AS4]: Enables access to optional software, Installs a package that includes various multimedia codecs and fonts necessary for playing restricted formats on Ubuntu.

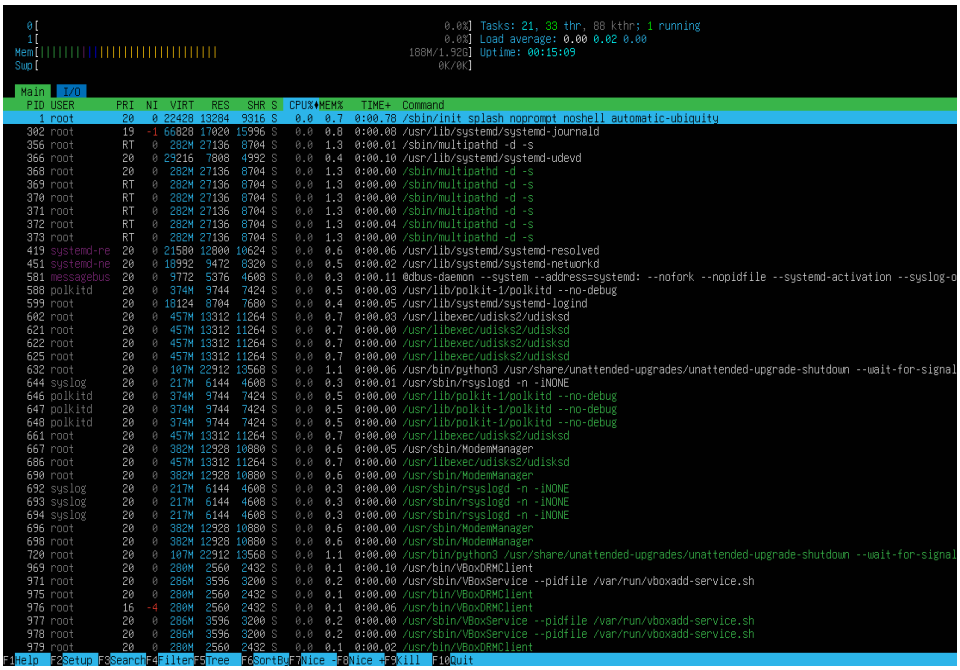
Neofetch

Commented [AS5]: neofetch displays system information in a visually appealing format in the terminal.



htop

Commented [AS6]: htop provides an interactive, real-time view of system processes and resource usage.



4. Created 2 other users and added them to groups

```
As@As-Ubuntu-Server:~$ groups
As sudo vboxsf
As@As-Ubuntu-Server:~$ groups user1
user1 : user1 users
As@As-Ubuntu-Server:~$ groups user2
user2 : user2 guest
```

Commented [AS7]: Users As, User1, User2 and their groups

5. Attached Guest additions

-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)

Commented [AS8]: Installs compiler tools, automatically builds and install kernel modules, installs headers for the current kernel version.

-sudo mkdir -p /mnt/cdrom

-sudo mount /dev/cdrom /mnt/cdrom

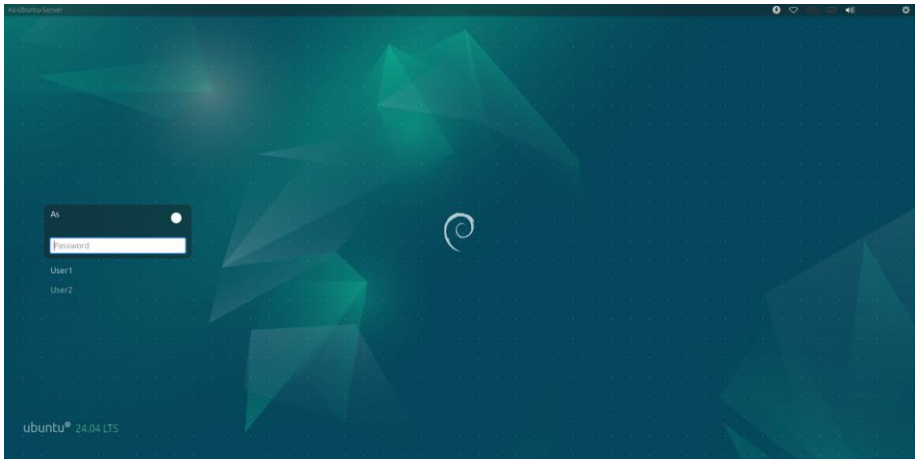
-sudo sh /mnt/cdrom/VBoxLinuxAdditions.run

6. Installed a Gui (cinnamon core) and a display manager (lightdm)

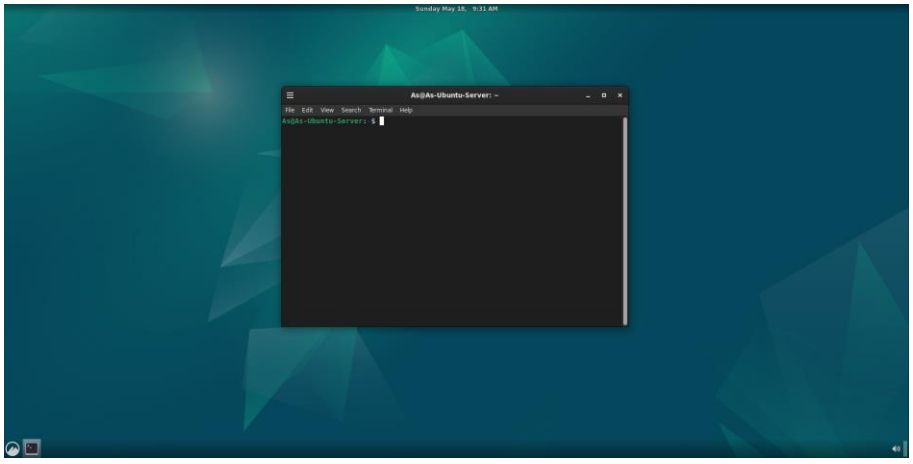
-sudo dpkg-reconfigure lightdm

Commented [AS9]: Sets lightdm as default display manager.

Lightdm

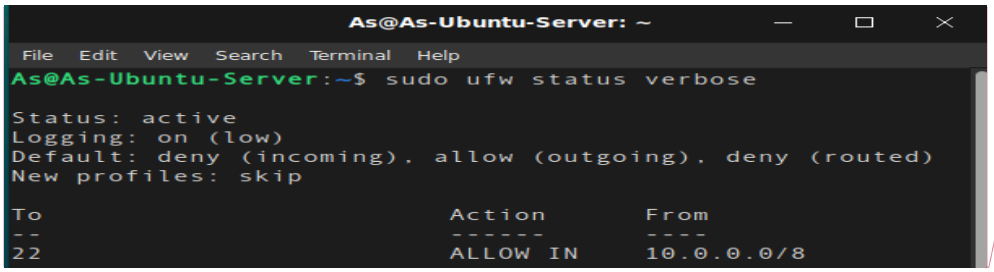


Cinnamon-core



7. Configured firewall

Deny incoming, allow outgoing, allow ssh from local network, logging on



Commented [AS10]: Allowed SSH in from local network.

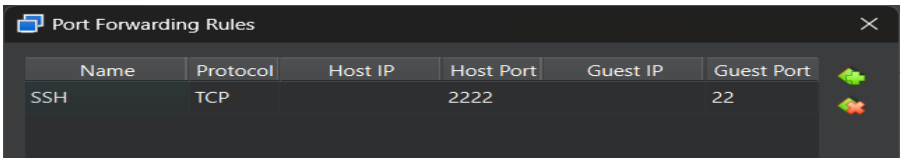
8. Installed OpenSSH & fail2ban

-Sudo apt install openssh-server

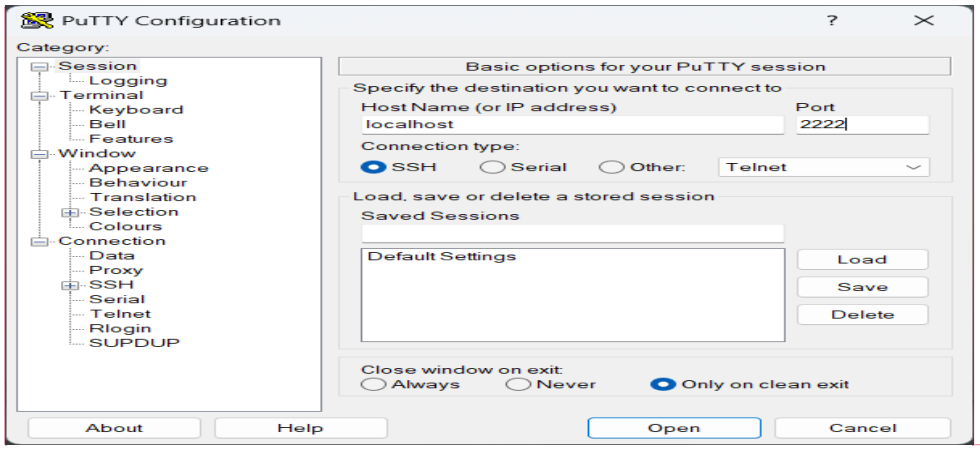
-sudo systemctl enable ssh, -sudo systemctl start ssh

Commented [AS11]: Enables and starts the service.

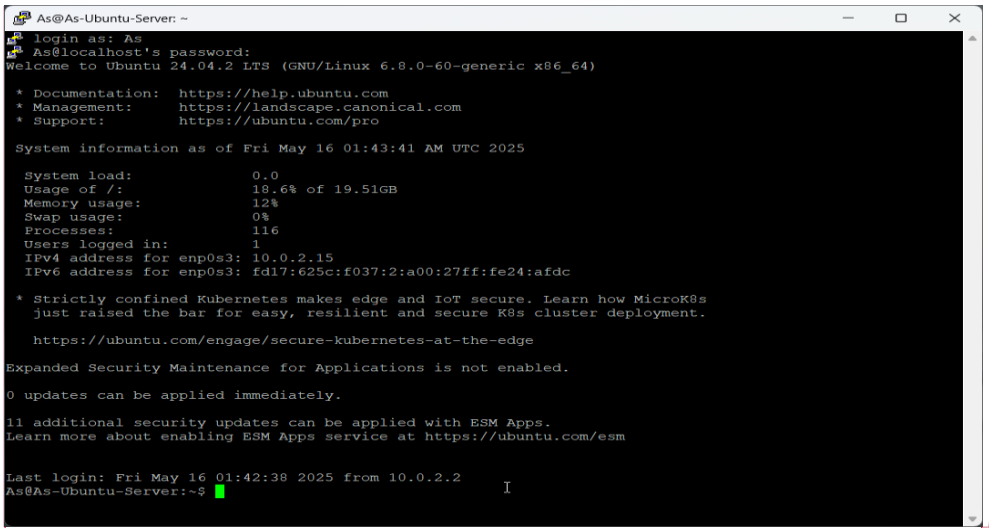
Set up port forwarding in VirtualBox



Utilized Putty on host machine and connected via SSH



Commented [AS12]: Can be SSH into by using "localhost" or the hosts IP address.



Commented [AS13]: Successful SSH login.

-Sudo apt install fail2ban -y

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

-sudo nano /etc/fail2ban/jail.local

Made this edit in the jail.local file under sshd

```
[sshd]
enabled=true
port    = ssh
logpath = %(sshd_log)s
maxretry = 5
findtime = 300
bantime = 300
backend = systemd
#Enables SSH monitor (port 22) & ban Failed login attempts afterfor 5 min after 5 tries
```

-sudo fail2ban-client status sshd

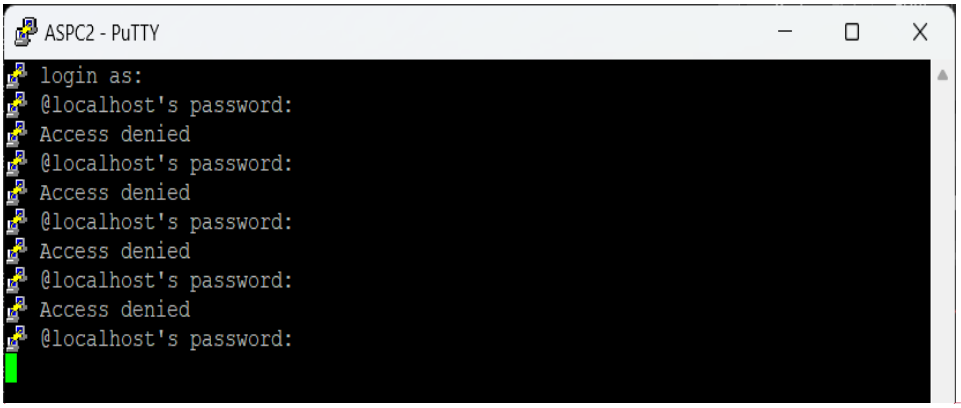
Commented [AS14]: Copies the default Fail2ban config file (jail.conf) to a local override file (jail.local) so you can safely customize settings without risking them being overwritten during updates.

Commented [AS15]: opens the editor for the jail.local file

Commented [AS16]: After 5 incorrect attempts to SSH into the server the Ip is banned for 5 Minutes.

Commented [AS17]: Shows # of IP banned, log file used, config in effect

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



Commented [AS18]: My failed login attempts through SSH.

```
As@As-Ubuntu-Server:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     5
|  `- Journal matches:  _SYSTEMD_UNIT=sshd.service + _COMM=sshd
`- Actions
   |- Currently banned: 1
   |- Total banned:     1
   `- Banned IP list:   10.0.2.2
As@As-Ubuntu-Server:~$
```

Commented [AS19]: My IP being temporarily banned.

9. Installed docker and set up a container with portainer

- sudo usermod -aG docker $USER

- sudo apt install apt-transport-https ca-certificates curl software-properties-common –y
- curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
- sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
- sudo apt install docker-ce -y
- sudo systemctl enable docker
- sudo systemctl start docker
- docker pull netdata/netdata

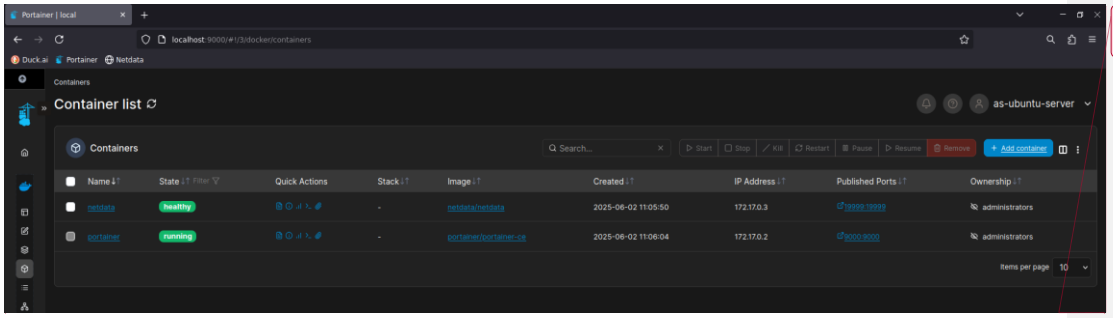
-Then created the 2 containers

Commented [AS20]: Adds the current user to the docker group, allowing them to run Docker commands without sudo.

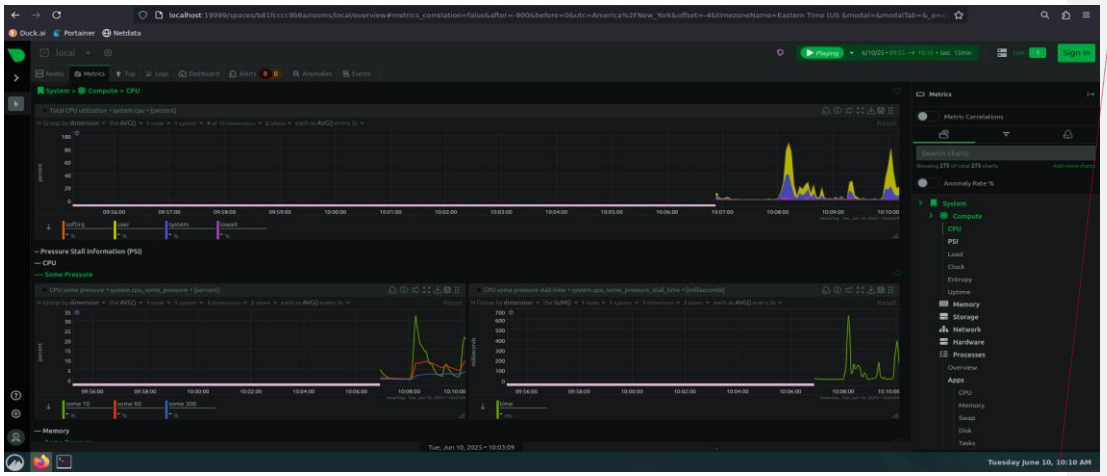
Commented [AS21]: installs required packages for secure package transfer, certificate handling, and software management, Downloads Docker's GPG key and adds it to your system for verifying Docker's packages, Adds the Docker repository to your system for package installation.

Commented [AS22]: Downloads the Netdata monitoring tool image from Docker Hub

Portainer and netdata containers running

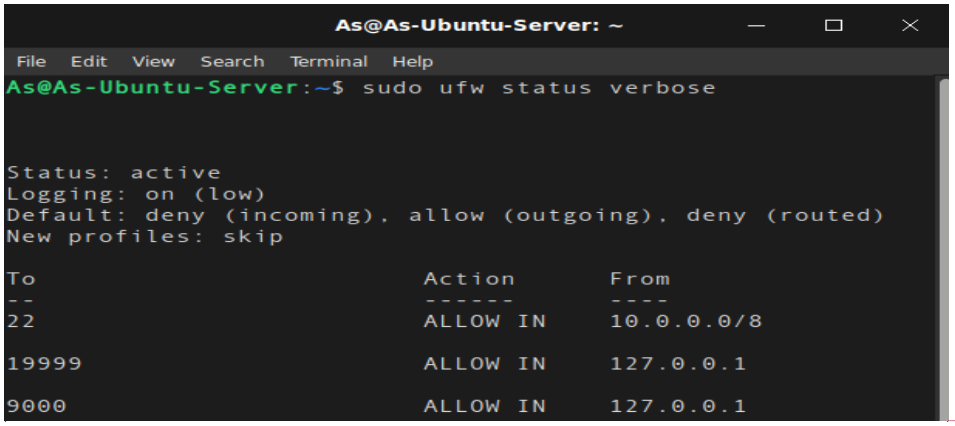


Commented [AS23]: Portainer web interface on port 9000 showing 2 containers active.



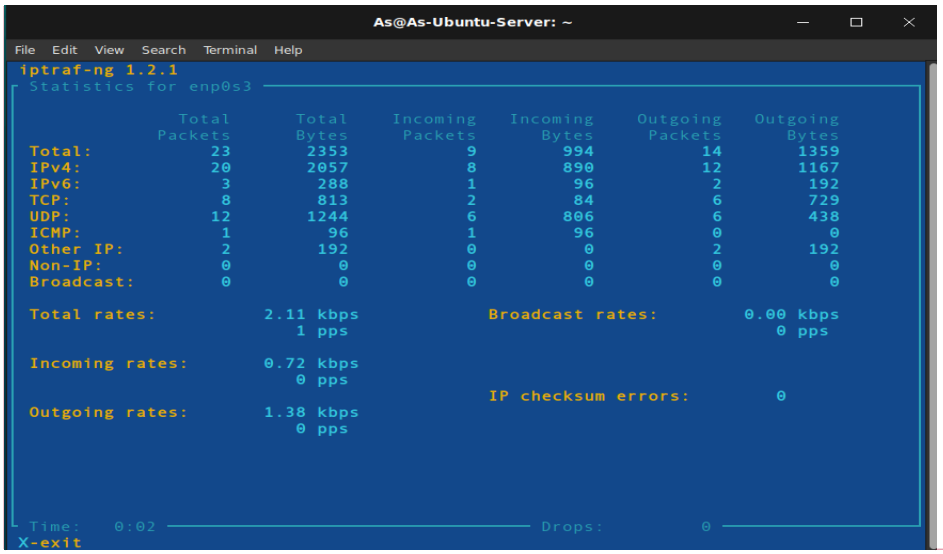
Commented [AS24]: netdata on port 1999 showing metrics for the server.

Updated firewall Rules to only allow local host to access the containers



Commented [AS25]: Containers live on ports 9000 & 19999 and the local host is 127.0.0.1

10. Installed iptraf to monitor packets



Commented [AS26]: Real time network monitoring tool that can monitor traffic by ports, protocols, packet and byte counts.