Assignment No. 1

Title: The DNS Configuration Assignment report file.

Aim:

- 1 DNS server in Ubuntu system
- 1 http host in ubuntu system
- 1 client, accessing the host using domain name
- report should include snap shots of the configuration files, terminal outputs, browserwindows-confirming the successful configuration of the service.

Submitted by:

1. Harshawardhan Prakash Parit C70
2. Yash Rajesh Kurne C64
3. Prathamesh Balaso Thambekar C83
4. Aditya Sadaguru Gavali C46
5. Gunjan Mangesh Karekar C79

DATE :

Three-Machine Setup Configuration

This report summarizes the configuration steps for a three-machine setup on a private network (192.168.1.0/24) involving a Web Server, a DNS Server, and a Client, with a dedicated external proxy for internet access. The configurations are tailored for Ubuntu using the terminal.

Machine 1: Web Server (IP: 192.168.1.10)

This machine is configured to host a website and use an external proxy (172.25.1.32) for internet connectivity.

1. Network Configuration

- **Static IP Assignment:** A static IP address of 192.168.1.10/24 is assigned to the machine's network interface (eth0).
- **Gateway:** The default gateway is set to 192.168.1.1.
- **DNS Servers:** The primary DNS server is configured to be the internal DNS server (192.168.1.20), with 8.8.8.8 as a public fallback.
- **Tool:** The configuration is implemented using Netplan by editing the /etc/netplan/01-netcfg.yaml file and applying the changes with sudo netplan apply.

2. Proxy Configuration

- **System-Wide Proxy:** The proxy settings are configured system-wide to ensure all users and applications (including apt for package management) use the proxy server http://172.25.1.32:8080.
    - **Acquire Configuration:** The apt package manager is configured via /etc/apt/apt.conf.d/proxy.conf to use the proxy for both HTTP and HTTPS requests.
    - **Environment Variables:** Environment variables (http_proxy, https_proxy, ftp_proxy) are set in /etc/environment to ensure applications like wget and others that respect these variables use the proxy. The no_proxy variable is used to exclude local network traffic from being routed through the proxy.

3. Web Server Installation

- **Software:** Either Apache2 or Nginx is installed as the web server software. Both are standard, widely used, and available in the Ubuntu repositories.
- **Service Management:** The web server service is enabled to start automatically on boot (sudo systemctl enable [service]) and is started immediately (sudo systemctl start [service]).

4. Simple Web Page Creation

- **Content:** A basic index.html file is created in the standard web root directory (/var/www/html/). The page contains simple HTML to confirm the web server is operational and to display its IP address (192.168.1.10).

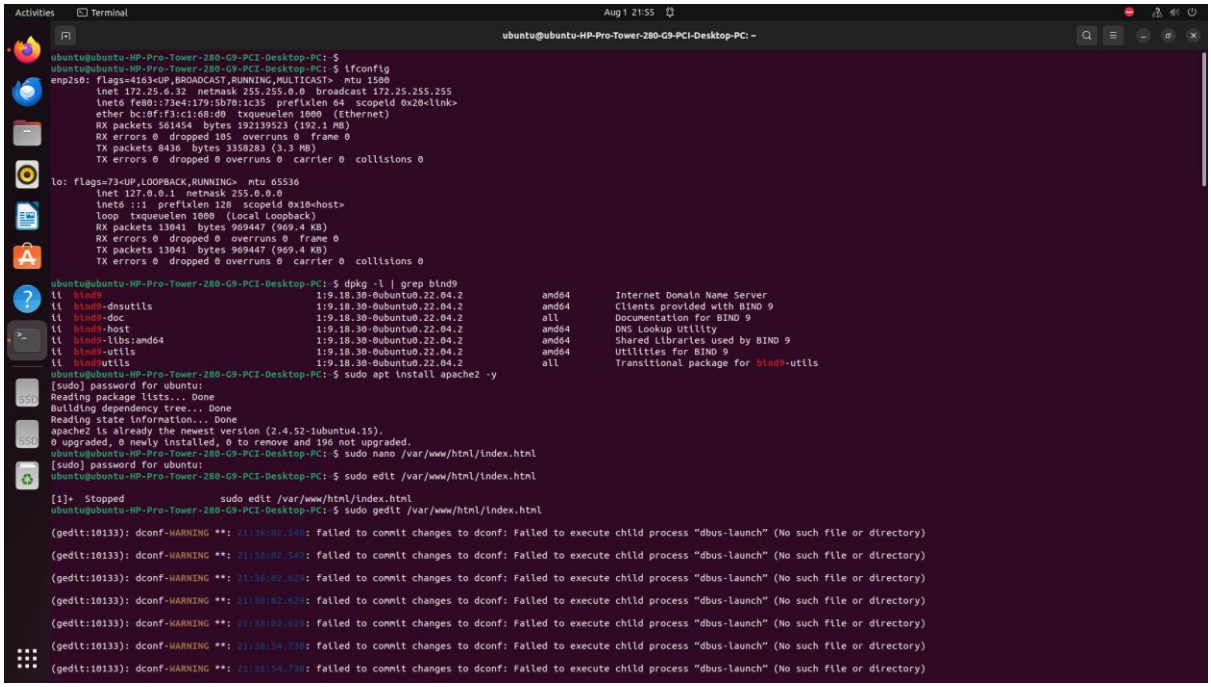
5. Firewall Configuration

- **Tool:** The Uncomplicated Firewall (UFW) is used to manage incoming traffic.

- **Rules:** Rules are added to allow incoming HTTP (port 80) and HTTPS (port 443) traffic, typically using predefined UFW profiles like 'Apache Full' or 'Nginx Full'. The firewall is enabled to enforce these rules.

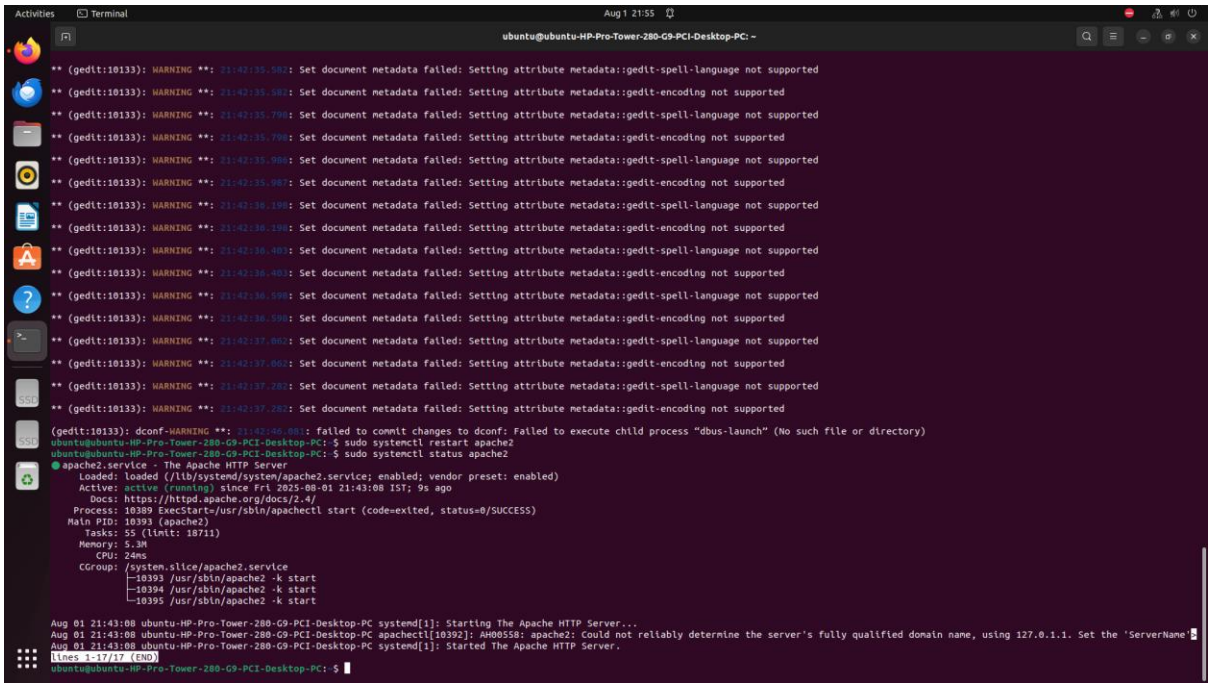
Installation of the web server Apache2:

Command: sudo apt install apache2 -y



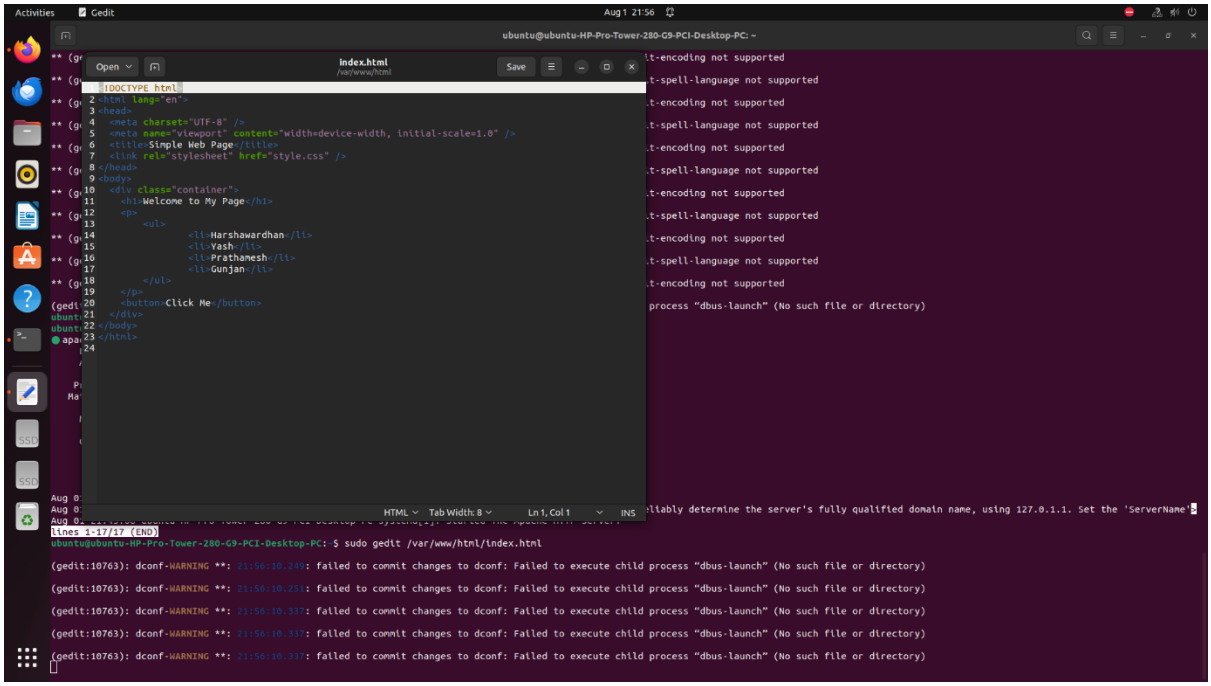
Status checking of Apache2 server

Command: sudo systemctl status apache2



Creating html page for access in the client machine using domain name:(index.html)

Command : sudo gedit /var/www/html.index.html



Machine 2: DNS Server (IP: 192.168.1.20)

This machine is configured to be the central authority for internal name resolution, ensuring that all devices can find each other by hostname.

1. Network Configuration

- **Static IP Assignment:** The DNS server is assigned a static IP of 192.168.1.20/24 on the same subnet as the web server.
- **Tool:** This configuration is performed via Netplan by editing the /etc/netplan/01-netcfg.yaml file. The nameservers are configured to point to an external, public DNS server (e.g., 8.8.8.8), as this machine's primary role is to resolve names for other devices and forward external queries.

YAML

network:

version: 2

renderer: networkd

ethernets:

eth0: # Replace with your actual network interface name

dhcp4: no

addresses: [192.168.1.20/24]

routes:

- to: default

via: 192.168.1.1 # Replace with your gateway IP

nameservers:

addresses: [8.8.8.8, 8.8.4.4] # Public DNS for forwarding external queries

- **Apply the changes:**

sudo netplan apply

2. DNS Server Installation and Configuration

- **Software:** The DNS server software, Bind9, is the standard choice for this role on Ubuntu. It's a robust, production-grade DNS daemon.
- **Installation:** Bind9 and its related utilities are installed from the Ubuntu repositories.

Bash

sudo apt update

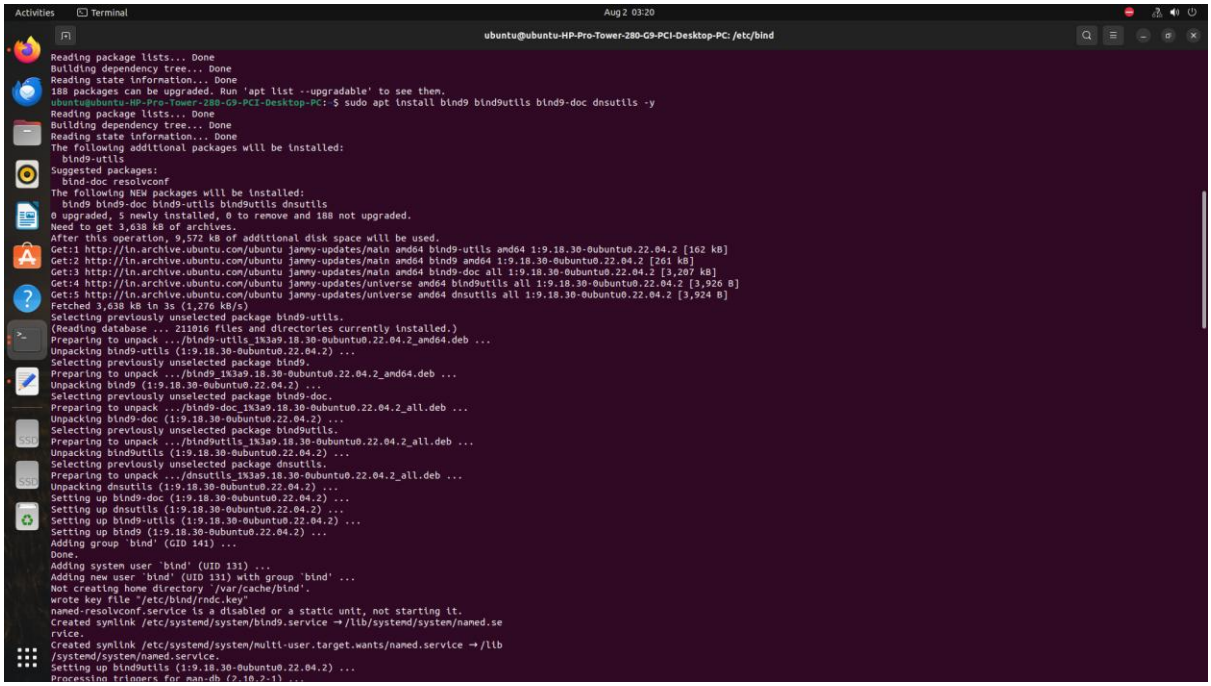
sudo apt install bind9 bind9-utils bind9-doc

- **Configuration Files:** The Bind9 configuration involves editing several files:
    - **/etc/bind/named.conf.local:** This file defines the zones that the DNS server will be authoritative for (e.g., a forward zone for yourdomain.com and a reverse zone for the 192.168.1.0/24 subnet).
    - **Forward Zone File (db.yourdomain.com):** This file contains A records that map hostnames to IP addresses (e.g., www.yourdomain.com to 192.168.1.10). It also defines the nameserver (ns1.yourdomain.com).
    - **Reverse Zone File (db.192.168.1):** This file contains PTR records for reverse lookups, mapping IP addresses back to hostnames.
- **Service Management:** The Bind9 service is started and enabled to run on boot, just like the web server.

DNS server Setup:

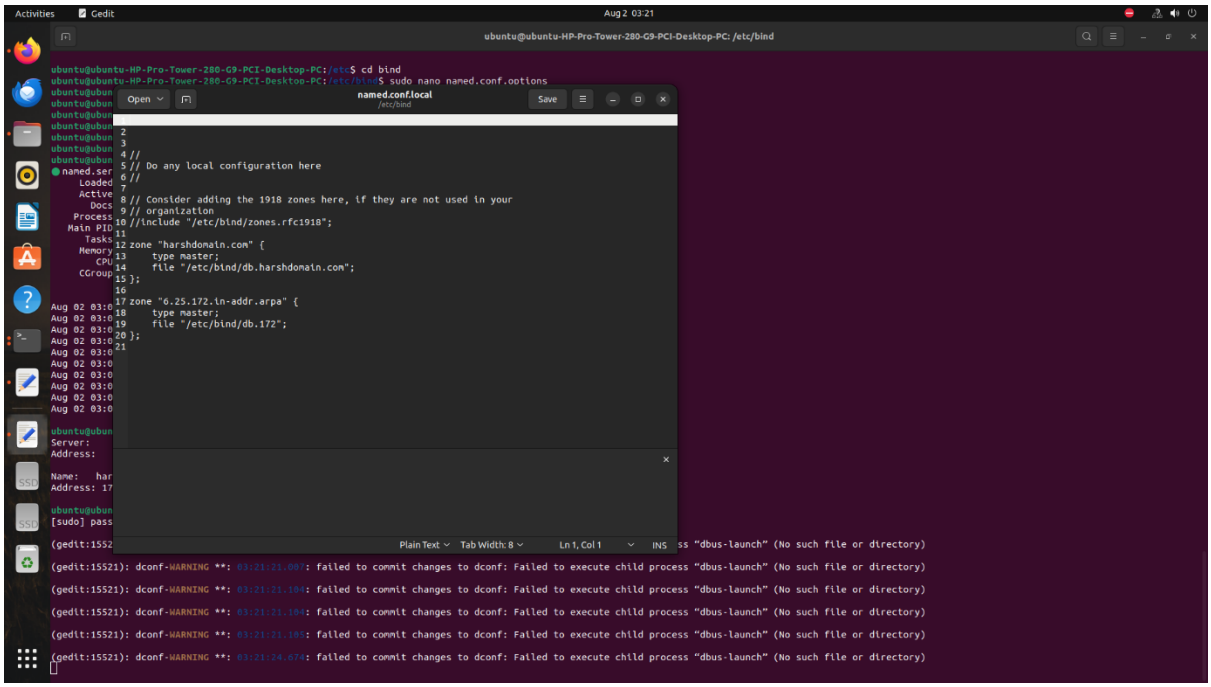
Bind9 Installation for the DNS :

Command: Sudo apt install bind9 bind9utils bind9-doc dnsutils -y



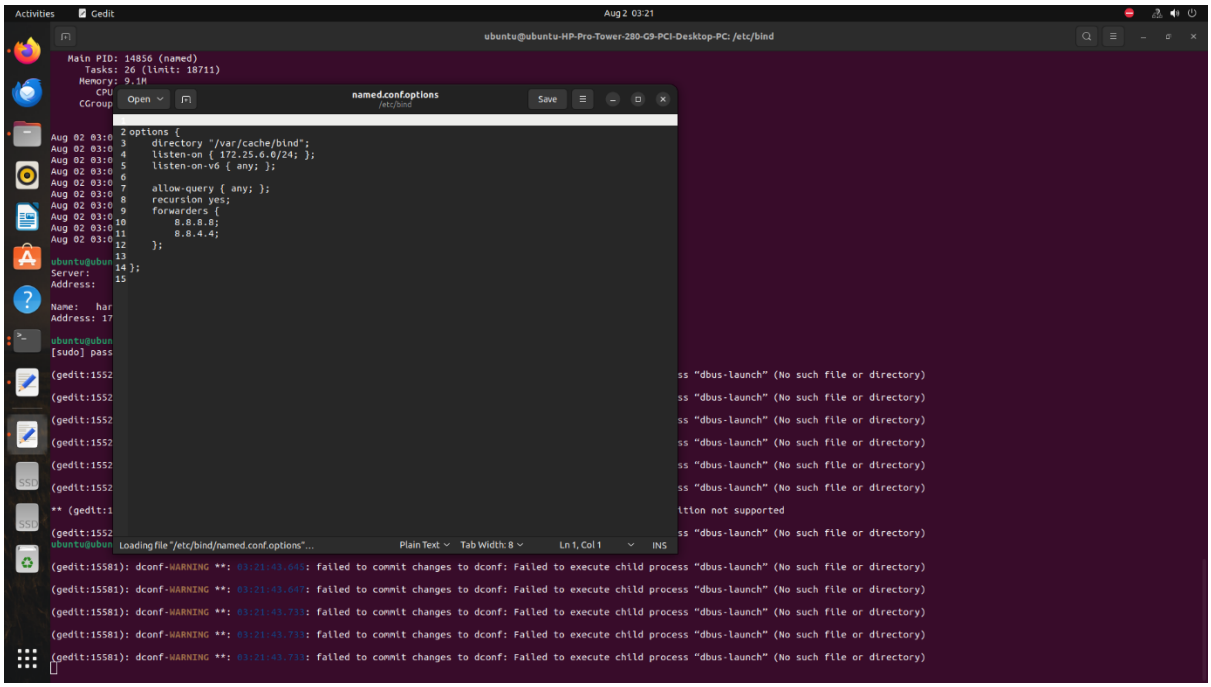
Edit Named.conf.local File:

Command : sudo nano /etc/named.conf.local



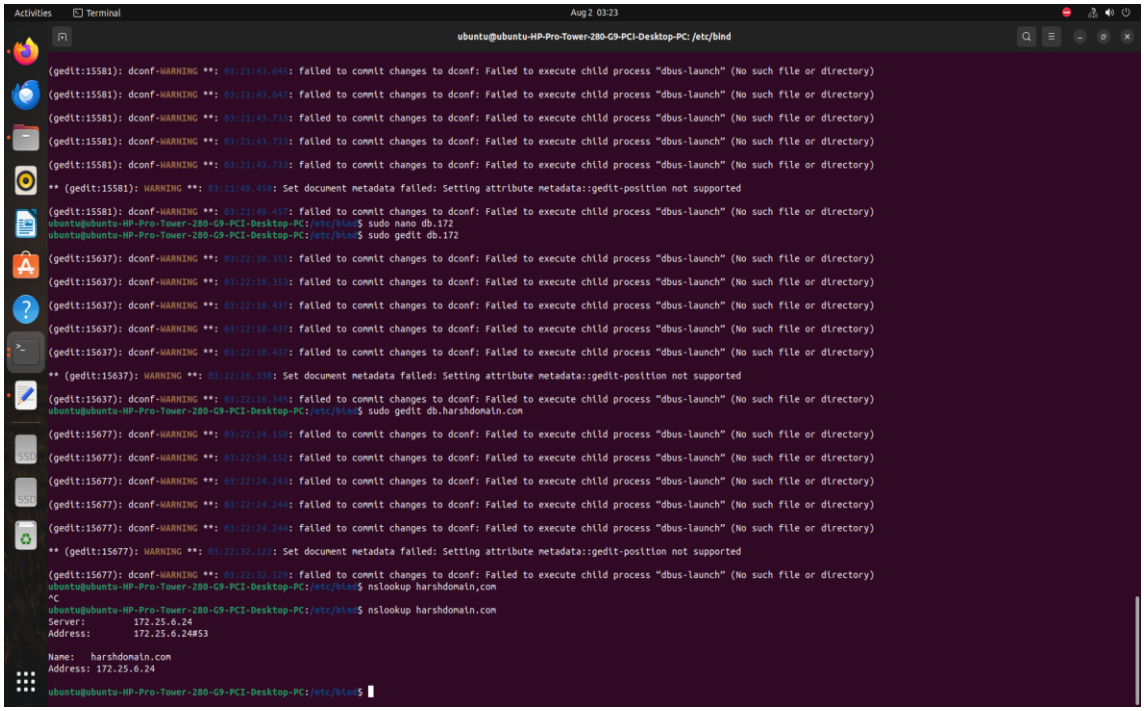
Edit named.option file:

Command: sudo nano /etc/bind/named.conf.option



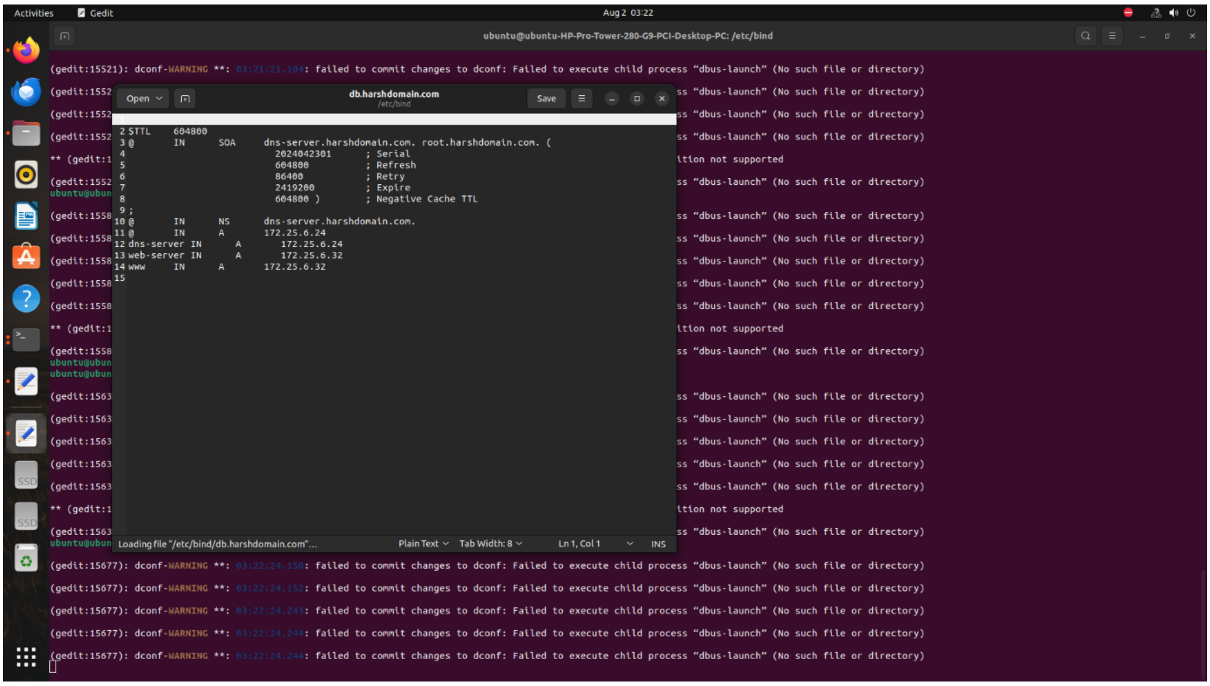
Edit db.172 file

Command: sudo nano /etc/bind/db.172



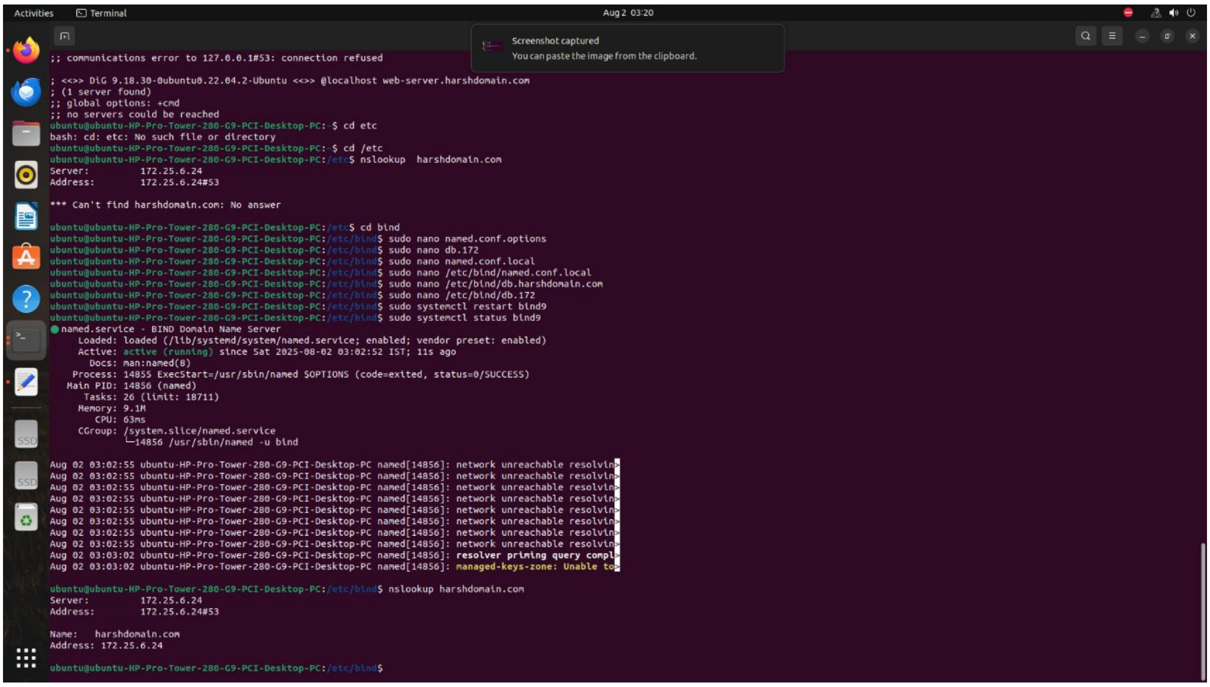
Edit harshdomain.com file

Command: sudo nano /etc/bind/harshdomain.com



Bind9 service status checking

Command : sudo systemctl status bind9



Machine 3: Client

This machine is configured to use the internal DNS server (192.168.1.20) to resolve hostnames and access the web server.

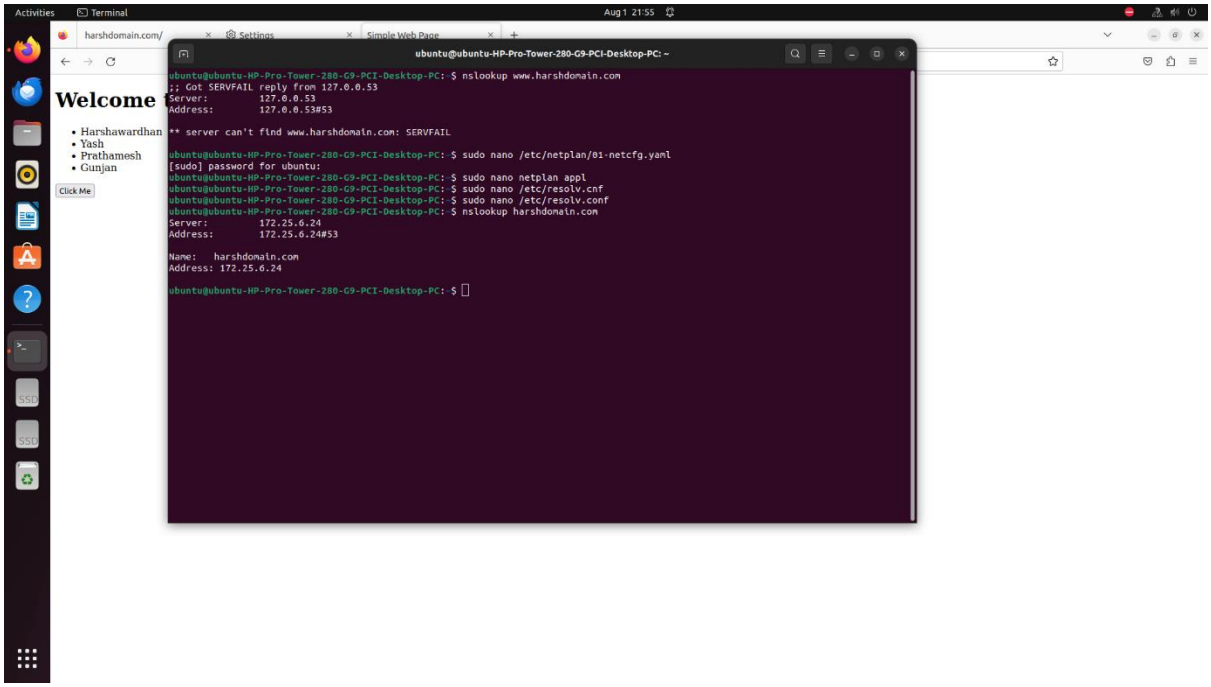
1. Network Configuration

- **Static IP or DHCP:** The client is assigned a static IP (e.g., 192.168.1.30) or obtains one from a DHCP server.
- **DNS Servers:** The most critical step is to configure this machine's network settings to use 192.168.1.20 as the primary DNS server. This is done via Netplan (for server) or NetworkManager (for desktop) settings, similar to the web server's configuration.

2. Verification

- **Connectivity:** The client should be able to ping the web server (192.168.1.10), the DNS server (192.168.1.20), and external websites.
- **DNS Resolution:** The dig command is used to test DNS resolution.
    - dig yourdomain.com: This should resolve to the web server's internal IP (192.168.1.10).
    - dig google.com: This should resolve correctly via the local DNS server, which forwards the request to the public internet.
- **Web Access:** The client should be able to access the web page by its domain name (e.g., http://yourdomain.com or http://www.yourdomain.com) in a web browser, confirming end-to-end functionality.

Client Setup Result :



Website Access Using the domain name (www.harshdomain.com)