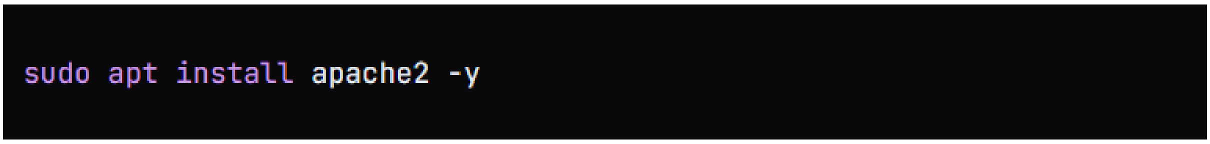
5-Aug-2024

Internship Day - 15 Report:

Step-by-Step Guide to Upload a Website Online Using Apache Server on Ubuntu

1. Install Apache2

- Open the terminal on your Ubuntu server and run the following command to install Apache2:



```
sudo apt install apache2 -y
```

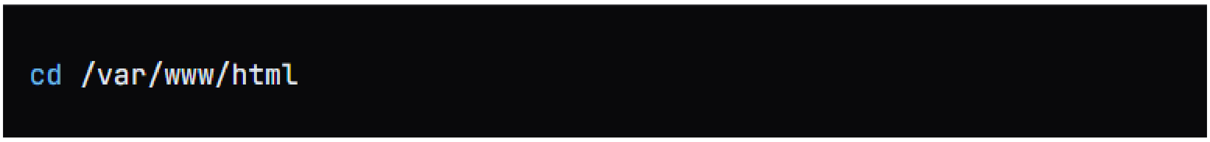
- This command installs the Apache web server. By default, Apache starts automatically after installation.

2. Access the Apache Server

- Copy the public IP address of your instance and paste it into your web browser. You should see the Apache2 default welcome page, indicating that the server is running correctly.

3. Navigate to the Web Directory

- In the terminal, change the directory to the default web root for Apache:

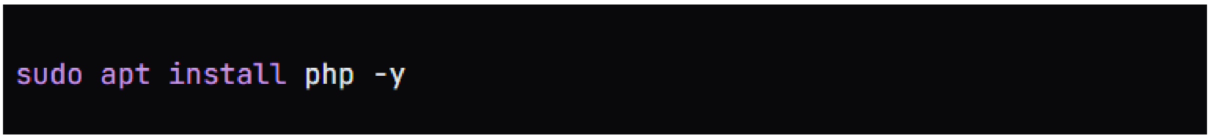


```
cd /var/www/html
```

- This is where you will upload your website files.

4. Install PHP

- To enable PHP support for your web applications, install PHP by running:

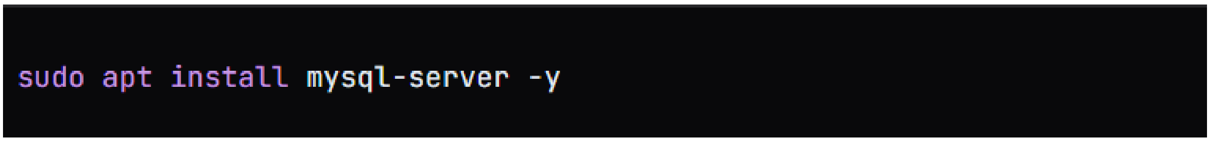


```
sudo apt install php -y
```

- This command installs PHP and its necessary modules.

5. Install MySQL

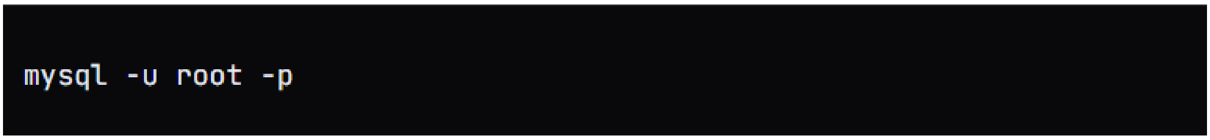
- Next, install the MySQL server to manage your databases:



```
sudo apt install mysql-server -y
```

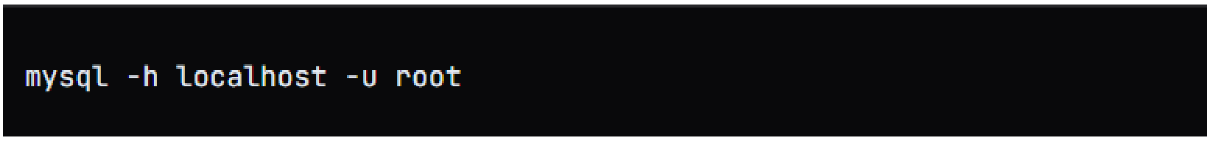
6. Access MySQL

- To access the MySQL prompt, use the following command:



```
mysql -u root -p
```

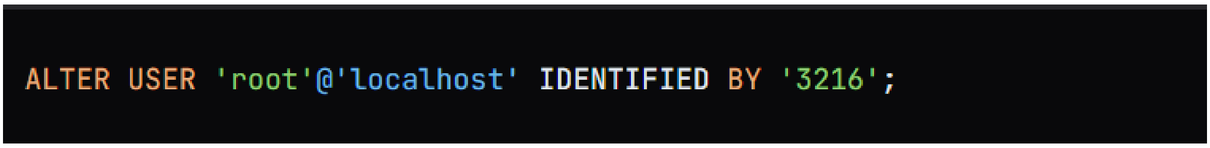
- Alternatively, you can use:



```
mysql -h localhost -u root
```

7. Set MySQL Root Password

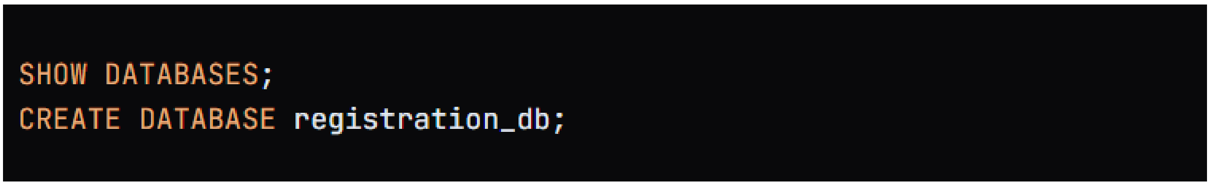
- Once logged in to MySQL, set a password for the root user by executing:



```
ALTER USER 'root'@'localhost' IDENTIFIED BY '3216';
```

8. Create a Database

- Show existing databases and create a new database for your application:

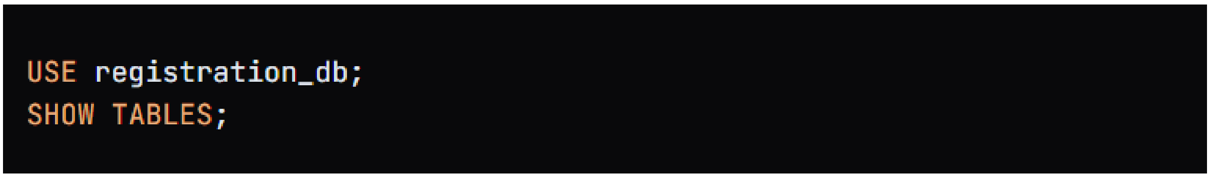


```
SHOW DATABASES;
CREATE DATABASE registration_db;
```

- Replace registration_db with your desired database name.

9. Use the Database

- Select the database you just created:



```
USE registration_db;
SHOW TABLES;
```

- You can now create tables or import data into this database.

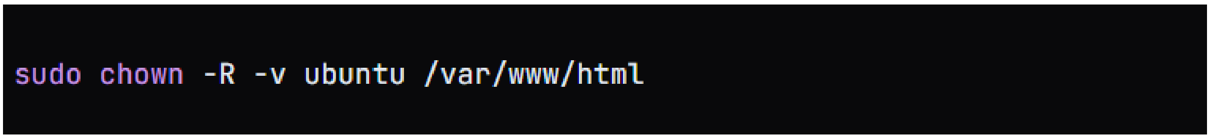
10. Upload Files Using FileZilla

- Open FileZilla and connect to your server using your .pem key file:
  1. **Host**: Your server's IP address
  2. **Username**: ubuntu
  3. **Authentication**: Use your .pem file

- Upload your SQL file to the /var/www/html directory through FileZilla.

11. Change Ownership of the Web Directory

- After uploading your files, change the ownership of the /var/www/html directory to ensure that the ubuntu user has the necessary permissions:



```
sudo chown -R -v ubuntu /var/www/html
```

- This command recursively changes the ownership of the directory.

12. Import the SQL File

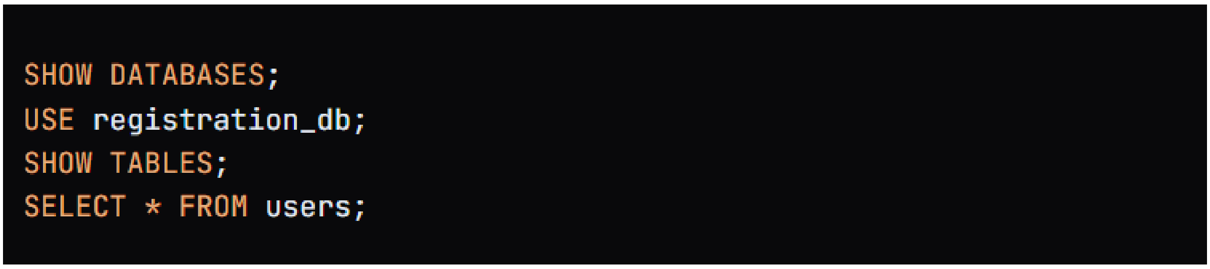
- To import the SQL file into your database, run the following command in the terminal:

  mysql -h localhost -u root -p registration_db < /var/www/html/registration_db.sql

- This command imports the SQL file into the registration_db database.

13. Verify Database Content

- To check the contents of your database, log in to MySQL again and run:



```
SHOW DATABASES;
USE registration_db;
SHOW TABLES;
SELECT * FROM users;
```

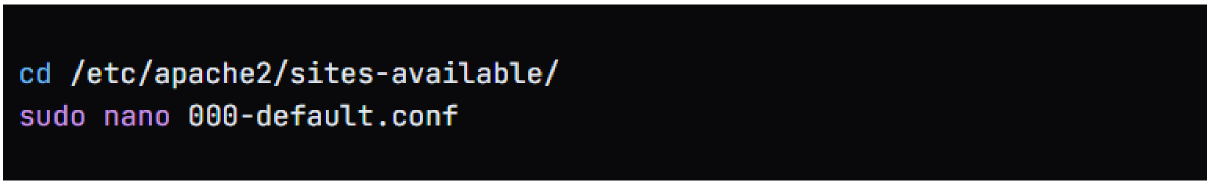
- This will display the tables and data within the users table.

14. Upload PHP Files

- Using FileZilla, upload your PHP files to the /var/www/html directory. Ensure that your PHP files are correctly placed and configured to connect to your database.

15. Configure Apache Virtual Host

- Finally, change the configuration for your website by editing the 000-default.conf file:



```
cd /etc/apache2/sites-available/
sudo nano 000-default.conf
```

- Modify this file to set the DocumentRoot or any other configurations necessary for your site.

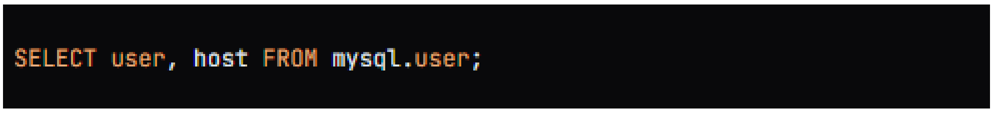
6-Aug-2024

Internship Day - 16 Report:

Step-by-Step Guide to Manage MySQL Users and Configure PHP on Apache Server

1. Check User Information in MySQL

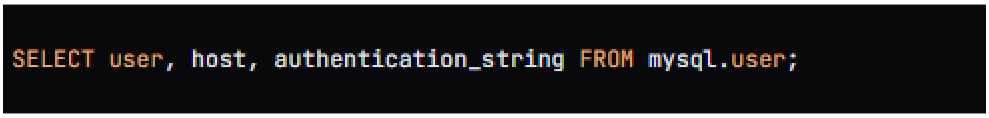
- To view the current users and their hosts, run the following command in the MySQL prompt:



```
SELECT user, host FROM mysql.user;
```

2. View User Authentication Details

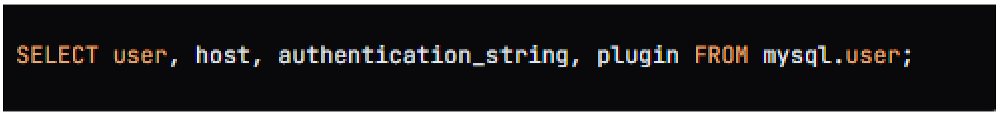
- To see more detailed information about users, including their authentication strings, execute:



```
SELECT user, host, authentication_string FROM mysql.user;
```

3. Check User Authentication Plugin

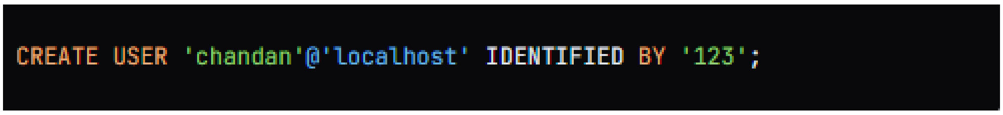
- To view the authentication plugin used by each user, run:



```
SELECT user, host, authentication_string, plugin FROM mysql.user;
```

4. Create a New MySQL User

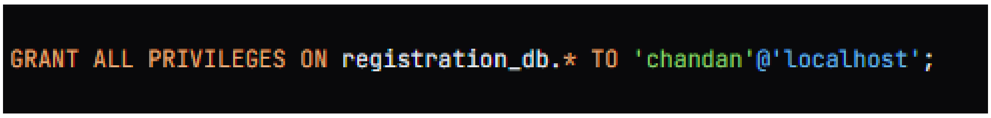
- To create a new user named chandan with a password, use the following command:



```
CREATE USER 'chandan'@'localhost' IDENTIFIED BY '123';
```

5. Grant Privileges to the New User

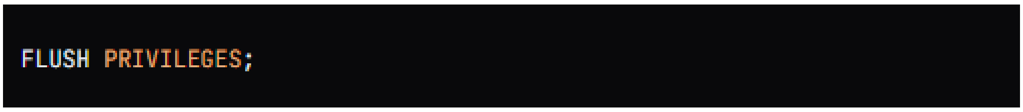
- To grant all privileges on the registration_db database to the new user, execute:



```
GRANT ALL PRIVILEGES ON registration_db.* TO 'chandan'@'localhost';
```

6. Flush Privileges

- To ensure that MySQL recognizes the changes made to user privileges, run:

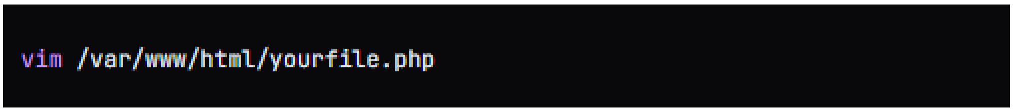


```
FLUSH PRIVILEGES;
```

- This command clears any cached privileges and reloads the user permissions.

7. Change Database Credentials in PHP File

- Open your PHP file using vim or any text editor to update the database username and password for the connection. For example:

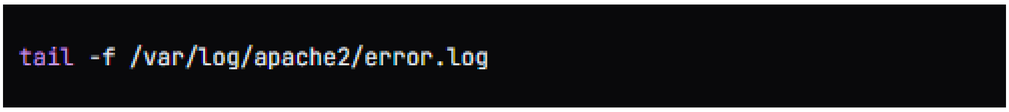


```
vim /var/www/html/yourfile.php
```

- Change the credentials in the connection string to reflect the new user chandan and the password '123'.

8. Check Apache Error Logs

- To monitor the Apache error logs in real-time for any issues, use the following command:



```
tail -f /var/log/apache2/error.log
```

This will display the latest error messages, helping you troubleshoot any problems with your web application.

7-Aug-2024

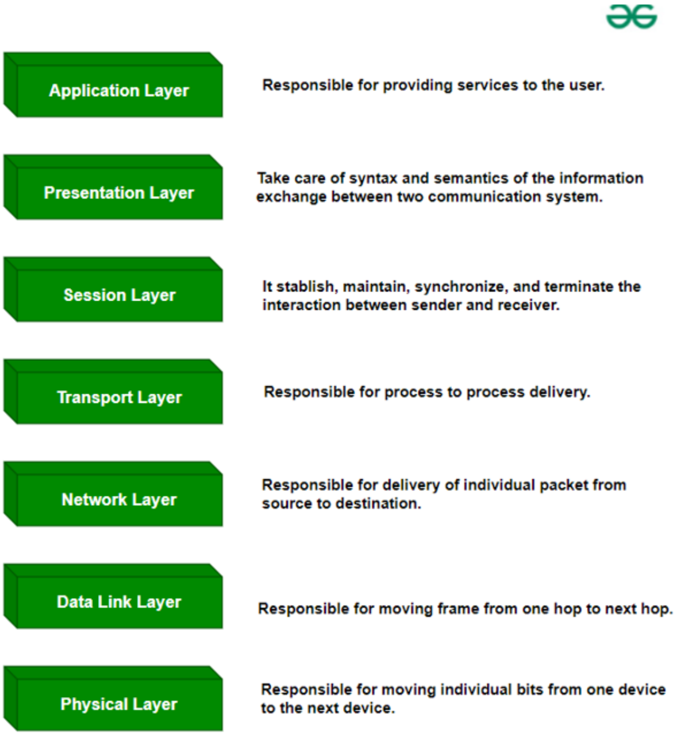
Internship Day - 17 Report:

Define Firewall

- **Firewall**: Protects a network by controlling incoming and outgoing traffic based on predetermined security rules. Requires knowledge of port numbers to control access.

OSI (Open System Interconnection) Model

- **Application Layer**: Handles human-computer interaction, where applications access network services.
- **Presentation Layer**: Ensures data is in a usable format and handles encryption.
- **Session Layer**: Maintains and controls sessions and ports.
- **Transport Layer**: Transmits data using transmission protocols, TCP and UDP.
- **Network Layer**: Determines the path for data transmission.
- **Data Link Layer**: Defines the data format for transmission over the network.
- **Physical Layer**: Deals with hardware (processor, RAM, HDD).


Application Layer
Responsible for providing services to the user.
Presentation Layer
Take care of syntax and semantics of the information exchange between two communication system.
Session Layer
It stablish, maintain, synchronize, and terminate the interaction between sender and receiver.
Transport Layer
Responsible for process to process delivery.
Network Layer
Responsible for delivery of individual packet from source to destination.
Data Link Layer
Responsible for moving frame from one hop to next hop.
Physical Layer
Responsible for moving individual bits from one device to the next device.

Protocols

- **Protocols**: A set of rules and regulations governing data exchange.
  - **Application, Presentation, Session Layers**: Associated with port numbers.
  - **Transport Layer**: Involves TCP/IP and UDP protocols.
  - **UDP**: Used for small queries (e.g., DNS).
  - **TCP**: Used for larger work/queries or requests.

DNS in AWS (Route 53)

- **AWS Route 53**: Manages DNS, mapping domain names to IP addresses.
- **Domain Name Website**: Paste DNS information to link a domain name to AWS.

Ports and Their Functions

- **SSH**: Port 22
- **MySQL**: Port 3306
- **HTTPS**: Port 443
- **SMTP**: Port 25
- **HTTP**: Port 80
- **POP3**: Port 995
- **RDP**: Port 3389
- **FTP**: Port 21 (data) and 20 (command)
- **POP2**: Port 109
- **DNS**: Port 53

Role of Port Numbers in Firewalls

- Port numbers help block or allow traffic through the firewall by controlling access based on protocol and destination.

Network Layer and Data Link Layer

- **Network Layer**: Uses IP addresses to route data.
- **Data Link and Physical Layer**: Handles network connections via WiFi, LAN, or WAN.

Client-Server Communication

- Example: **Client** (IP and port 192.168.1.1:3306) sends a request to **Server**.

Ping of Death Attack

- A **DOS (Denial of Service)** attack where excessive ping requests overload the network, particularly through UDP protocol due to its small request size.

8-Aug-2024

Internship Day - 18 Report:

Topics in Firewall

a. What is a Firewall?

b. Firewalld service in Linux

c. Enable/disable Firewall

d. How to see the existing firewall rules

e. Adding & deleting Firewall Rules

f. Adding/Removing Ports

g. Block incoming/outgoing Traffic

h. Block ICMP

Types of Firewall:

1. **Software Firewall:** A software firewall is installed on individual devices, such as computers or servers, and monitors incoming and outgoing traffic at the software level. It's more flexible and often suitable for individual or small network protection.

2. **Hardware Firewall:** A hardware firewall, on the other hand, is a physical device placed between a network and external traffic sources, providing a strong barrier for larger networks. It offers robust protection for enterprise environments, managing traffic before it reaches internal devices.

Make a Linux instance in AWS and access through xfreedp in linux:

1. Basic Setup and Initial Configuration

Installing Apache with Firewall Protection:

```
# Install Apache
sudo apt install apache2 -y

# Install firewalld
sudo apt install firewalld -y

# Start and enable firewalld
sudo systemctl start firewalld
sudo systemctl enable firewalld

# Verify installations
apache2 -v
firewall-cmd --version
```

2. Common Firewall Scenarios

Scenario 1: Setting Up a Web Server

```
# Allow HTTP and HTTPS
sudo firewall-cmd --zone=public --add-service=http --permanent
sudo firewall-cmd --zone=public --add-service=https --permanent
sudo firewall-cmd --reload

# Verify configuration
curl localhost
```

Scenario 2: Database Server Setup

```
# Allow MySQL
sudo firewall-cmd --zone=public --add-port=3306/tcp --permanent
sudo firewall-cmd --reload

# Test MySQL connection
telnet localhost 3306
```

3. Advanced Service Management

Working with Custom Services

```
# Create new service
sudo firewall-cmd --new-service=myapp --permanent

# Configure service
sudo firewall-cmd --service=myapp --add-port=8080/tcp --permanent
sudo firewall-cmd --zone=public --add-service=myapp --permanent
```

Multiple Port Configuration

```
# Add port range
sudo firewall-cmd --zone=public --add-port=4000-4100/tcp --permanent

# Add multiple individual ports
sudo firewall-cmd --zone=public --add-port=8080/tcp --add-port=8443/tcp --permanent
```

4. Security Hardening Examples

Blocking Suspicious IP Addresses

```
# Block single IP
sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="10.10.10
.10" reject'

# Block IP range
sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="192.168.
1.0/24" reject'
```

Rate Limiting

```
# Limit SSH connections
sudo firewall-cmd --permanent --add-rich-rule='rule service name="ssh" accept limit value="
3/m"'
```

5. Real-World Applications

Setting Up a LAMP Stack

```
# Allow required services
sudo firewall-cmd --zone=public --add-service=http --permanent
sudo firewall-cmd --zone=public --add-service=https --permanent
sudo firewall-cmd --zone=public --add-port=3306/tcp --permanent
sudo firewall-cmd --reload

# Verify configuration
sudo firewall-cmd --list-all
```

Securing Remote Access

```
# Configure SSH access
sudo firewall-cmd --zone=public --add-service=ssh --permanent

# Allow specific IP for SSH
sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="trusted-i
p" service name="ssh" accept'
```

6. Monitoring and Troubleshooting

Real-time Monitoring

```
# Watch active connections
watch -n1 "sudo firewall-cmd --list-connections"
```

```
# Monitor rejected packets
sudo tail -f /var/log/messages | grep REJECT
```

Common Issues and Solutions

Problem 1: Service Not Accessible

```
# Check if service is allowed
sudo firewall-cmd --list-services

# Verify port is open
sudo firewall-cmd --list-ports

# Test port locally
nc -zv localhost <port>
```

Problem 2: Website Blocking Not Working

```
# Clear DNS cache
sudo systemd-resolve --flush-caches

# Verify rich rules
sudo firewall-cmd --list-rich-rules

# Test blocking
ping blocked-website.com
```

7. Advanced Configurations

Custom Zone Configuration

```
# Create custom zone
sudo firewall-cmd --permanent --new-zone=customzone

# Configure zone
sudo firewall-cmd --zone=customzone --add-service=http --permanent
sudo firewall-cmd --zone=customzone --add-source=192.168.1.0/24 --permanent
```

Time-Based Rules

```
# Allow service during business hours
sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" service name="http" acce
pt time start="9:00" end="17:00"'
```

8. Backup and Recovery

Backup Configuration

```
# Export configuration
sudo firewall-cmd --runtime-to-permanent
sudo cp -r /etc/firewalld /etc/firewalld.bak

# Restore configuration
sudo cp -r /etc/firewalld.bak/* /etc/firewalld/
sudo systemctl restart firewalld
```

9. Regular Maintenance Tasks

Weekly Maintenance Checklist

```
# 1. Check firewall status
sudo firewall-cmd --state

# 2. Review active rules
sudo firewall-cmd --list-all

# 3. Check logs for suspicious activity
sudo journalctl -u firewalld --since "24 hours ago"

# 4. Verify services
sudo firewall-cmd --list-services

# 5. Update firewall
sudo apt update
sudo apt upgrade firewalld
```

10. Performance Optimization

Optimize Rule Processing

```
# Remove redundant rules
sudo firewall-cmd --remove-rich-rules='rule family="ipv4" source address="0.0.0.0/0" accept
'

# Organize rules by frequency
# Most used rules should be at the top
sudo firewall-cmd --permanent --add-rich-rule='rule priority=1 service name="http" accept'
```

9-Aug-2024

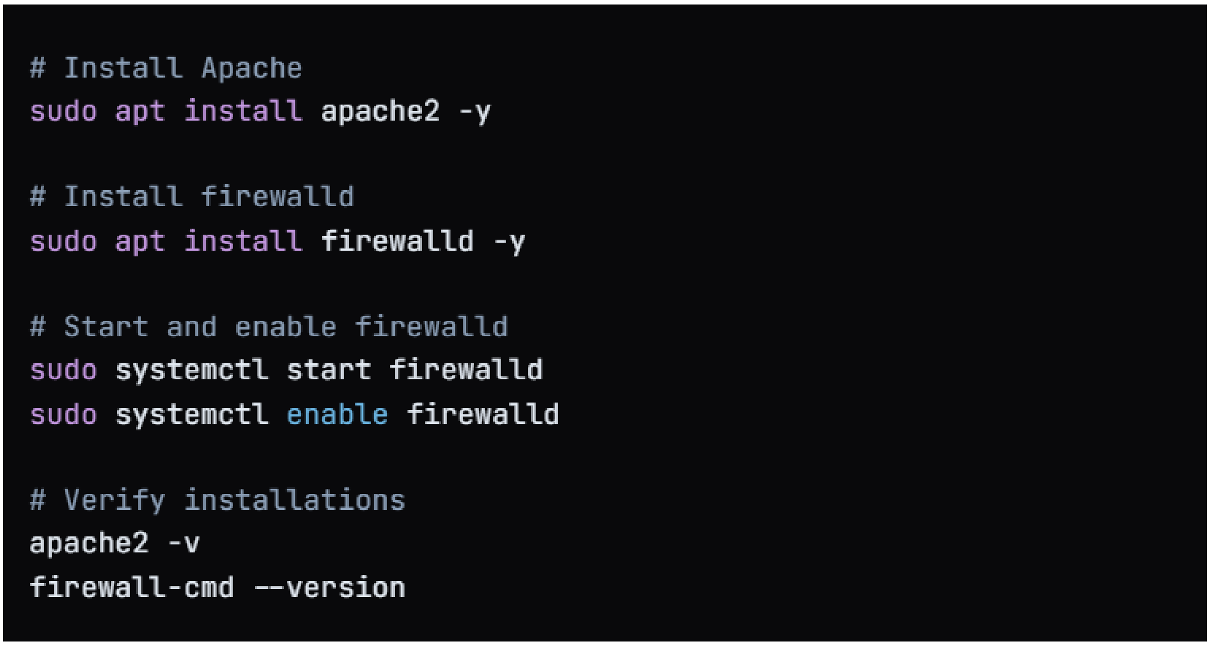
Internship Day - 19 Report:

Comprehensive Firewall Management Guide Continue

apt install apache2

service apache2 status

apt install firewalld –y



```
# Install Apache
sudo apt install apache2 -y

# Install firewalld
sudo apt install firewalld -y

# Start and enable firewalld
sudo systemctl start firewalld
sudo systemctl enable firewalld

# Verify installations
apache2 -v
firewall-cmd --version
```

ping gndec.ac.in

For Block a website temporary :

firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -d 175.176.187.102 -j DROP

To check list which website has been blocked :

iptables -t filter -L -v -n

For UnBlock a website temporary:

firewall-cmd --direct --remove-rule ipv4 filter OUTPUT 0 -d 175.176.187.102 -j DROP

OR

firewall-cmd –reload

For Block a website Permanent :

```
firewall-cmd --direct --remove-rule ipv4 filter OUTPUT 0 -d 175.176.187.102 -j DROP
```