CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

A PROJECT REPORT

ON LINUX SERVER CONFIGURATION

(LAMP STACK AND FTP)

Program Name: 23BCA

Linux Administration

23CAP-305

SUBMITED BY

Name: - Mumtaj Ali

Uid: - 23BCA10662

Group/Section: -23BCA (3 "B")

SUBMITED BY

Name: - Rajat Kapoor

Designation: Assistant Professor

Sign: _____

BONAFIDE CERTIFICATE

Certified that this project report "**LINUX SERVER CONFIGURATION (LAMP STACK AND FTP)**" is the Bonafide work of "**Mumtaj Ali"** who carried out the project work under my/our supervision.

.....

SIGNATURE

UIC DEPARTMENT

ASSISTANT PROFESSOR

Mr. RAJAT KAPOOR

Table Of Content

1. Aim of the Project

The aim of this project is to successfully transform a base Linux installation into a fully functional and secure server by configuring three core network services: the Apache web server, the MySQL database server, and the VSFTPD file transfer protocol server. This setup demonstrates proficiency in managing essential components of a typical web hosting environment (often called a LAMP stack, minus PHP).

2. Introduction

Modern web services rely on robust and interconnected servers to handle traffic, process data, and manage file transfers. The Linux operating system, particularly distributions like Ubuntu or CentOS, is the foundation for the majority of these services due to its stability and security features. This project focuses on building this foundation by configuring key daemons (Apache, MySQL, and vsftpd) that enable the system to serve content, store structured data, and facilitate remote file operations.

3. Objectives

The specific objectives achieved through this project include:

1. To successfully install and start the Apache HTTP Server.
2. To configure a basic web page and ensure it is accessible via the network.
3. To install and secure the MySQL database server.
4. To create a test database and a dedicated user with appropriate permissions.
5. To set up and configure the VSFTPD daemon for secure local user file transfers.
6. To manage the Linux firewall (ufw) to selectively allow traffic only on required service ports (21, 22, 80, 3306).
7. To demonstrate proper service management using the systemctl utility.

4. System Requirements

Hardware Requirements:

- CPU: 2 Cores (Minimum)
- RAM: 2 GB (Minimum for smooth VM operation)
- Storage: 20 GB of free disk space
- Network: Stable internet connection for package downloads.

Software Requirements:

- Host OS: Windows, macOS, or Linux
- Virtualization Software: VirtualBox or VMware Workstation
- Guest OS: Ubuntu Server 22.04 LTS (Recommended) or CentOS Stream

5. Tools and Technologies Used

Primary Tools:

- **Apache2:** The open-source web server used to serve HTTP content (Port 80).
- **MySQL Server:** The relational database management system used for structured data storage (Port 3306).
- **VSFTPD (Very Secure FTP Daemon):** The lightweight, secure FTP server used for remote file management (Port 21).
- **UFW (Uncomplicated Firewall):** Used for managing network access and hardening the system security.

Core Commands and Utilities:

Category	Key Command/Tool	Purpose
Package Mgmt.	apt (or dnf/yum)	Installing and managing software packages.

Service Control	systemctl	Starting, stopping, enabling, and checking the status of services (daemons).
Networking	ip a (or ifconfig)	Checking the server's network interfaces and IP address.
Configuration	nano or vi	Text editors used to modify configuration files (e.g., vsftpd.conf, Apache sites).
Security	ufw	Managing firewall rules to control incoming and outgoing network traffic.
Database	mysql	Command-line client for interacting with the MySQL server.

Additional Utilities:

- **ss (Socket Statistics) / netstat:** Used to verify which ports are listening for connections (e.g., ss -tuln).
- **wget / curl:** Used for internal testing to confirm Apache is serving content correctly from the command line.
- **mysql_secure_installation:** The script used to perform initial security hardening of the MySQL server.
- **a2ensite / a2dissite:** Apache utilities to enable and disable site configuration files.

6. Project Methodology (Implementation Steps)

The project followed a modular and iterative configuration approach to ensure each service was operational and secure before proceeding to the next.

STEPS INCLUDE:

The steps below summarize the practical configuration process executed on the Linux VM:

S1) System Setup and Firewall Configuration

1. **System Update:** sudo apt update && sudo apt upgrade.
2. **Firewall Installation/Enablement:** sudo apt install ufw (if not present) followed by sudo ufw enable.
3. **Essential Rules:** sudo ufw allow ssh and sudo ufw allow http.

S2) Apache Web Server (HTTP) Configuration

1. **Installation:** sudo apt install apache2 -y.
2. **Configuration:** Modify the default document root page (/var/www/html/index.html) to display a custom project message.
3. **Verification:** Check status (sudo systemctl status apache2) and confirm web access via the server's IP in a browser.

S3) MySQL Database Server Configuration

1. **Installation:** sudo apt install mysql-server -y.
2. **Security:** Run sudo mysql_secure_installation to set the root password, disable remote root login, and remove anonymous users.
3. **User/DB Setup:** Log into the MySQL prompt and execute SQL commands to create the database (project_db) and a local user (project_user).
4. **Firewall:** sudo ufw allow 3306 for local and potentially remote database access.

S4) VSFTPD (File Transfer) Configuration

1. **Installation:** sudo apt install vsftpd -y.
2. **Configuration:** Edit /etc/vsftpd.conf to disable anonymous access (anonymous_enable=NO), enable local user login (local_enable=YES), and set up a secure chroot jail.
3. **Firewall:** sudo ufw allow ftp (Port 21).
4. **Restart:** sudo systemctl restart vsftpd.

S5) Final Testing and Service Verification

1. Final check of all open ports: sudo ufw status.
2. Test connectivity to all configured services (Web, Database, FTP) from the host machine.

Project Methodology:

The project was executed through five distinct phases:

Phase 1: Planning and Analysis

- **Goal:** Determine the necessary packages, security requirements (ports), and the execution sequence.
- **Action:** Decided to use Apache, MySQL, VSFTPD, and UFW on an Ubuntu VM. Identified required ports: 22, 80, 3306, 21.

Phase 2: Script and Installation Development

- **Goal:** Install and configure the core packages.
- **Action:** Ran sudo apt install for all services. Edited configuration files (/etc/vsftpd.conf, /var/www/html/index.html) using the nano editor.

Phase 3: Automation Setup (Service Enablement)

- **Goal:** Ensure all services start automatically on boot and are actively running.
- **Action:** Used sudo systemctl enable [service_name] for Apache, MySQL, and VSFTPD. Also used sudo ufw enable to start the firewall persistently.

Phase 4: Testing and Validation

- **Goal:** Verify that each service functions correctly and securely.
- **Action:** Tested Apache via HTTP request, MySQL via command-line login with the new user, and VSFTPD via a separate FTP client connection. Confirmed correct firewall state (sudo ufw status).

Phase 5: Documentation and Deployment

- **Goal:** Record all steps, commands, and outputs.
- **Action:** Compiled this report, documenting the methodology, code snippets, and results of the verification tests.

Algorithm / Logic / Flow chart:

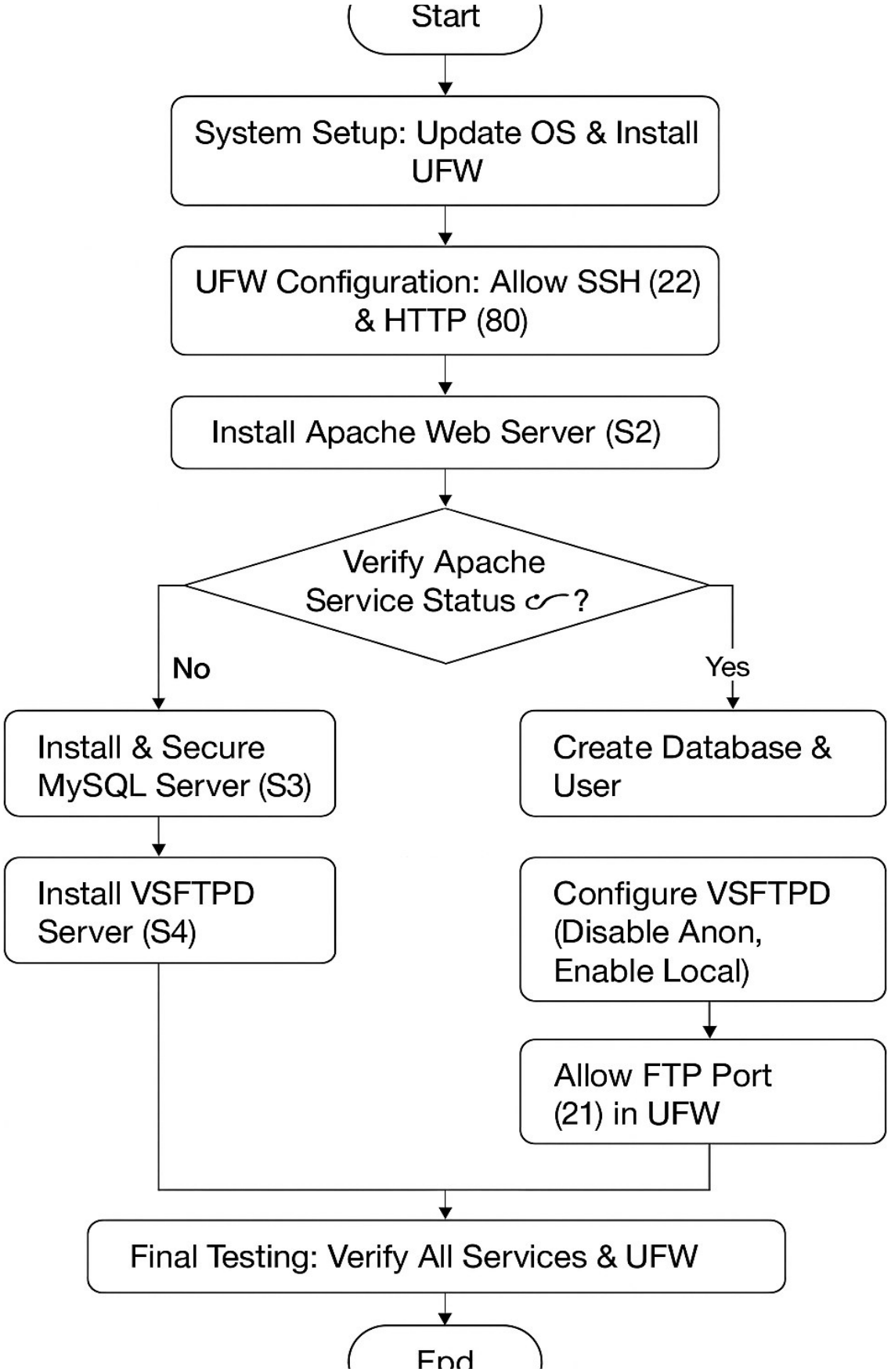
Logic Explanation:

The logic for the project is sequential and dependency-based. First, the base system and its protective layer (Firewall) are configured. Only after the core network paths are secured (SSH, HTTP) can the services be installed. Each service (Apache, MySQL, VSFTPD) is then installed, configured for security, and verified, ensuring the integrity of the server is maintained throughout the process.

Flowchart:

The process starts, configures the base OS and Firewall, then branches sequentially to install and verify Apache, MySQL, and VSFTPD before reaching the final verification step and termination.

Flowchart Diagram:



Code Overview:

Code Overview

The following snippets highlight the critical configuration and setup commands:

1. Firewall Configuration

```
# Enable firewall and allow essential services
sudo ufw enable
sudo ufw allow ssh
sudo ufw allow http
sudo ufw allow 3306/tcp  # MySQL
sudo ufw allow ftp
```

2. MySQL Database Setup

```
# SQL commands executed in the MySQL shell
CREATE DATABASE project_db;
CREATE USER 'project_user'@'localhost' IDENTIFIED BY 'secure_password';
GRANT ALL PRIVILEGES ON project_db.* TO 'project_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

3. VSFTPD Configuration (Key lines in /etc/vsftpd.conf)

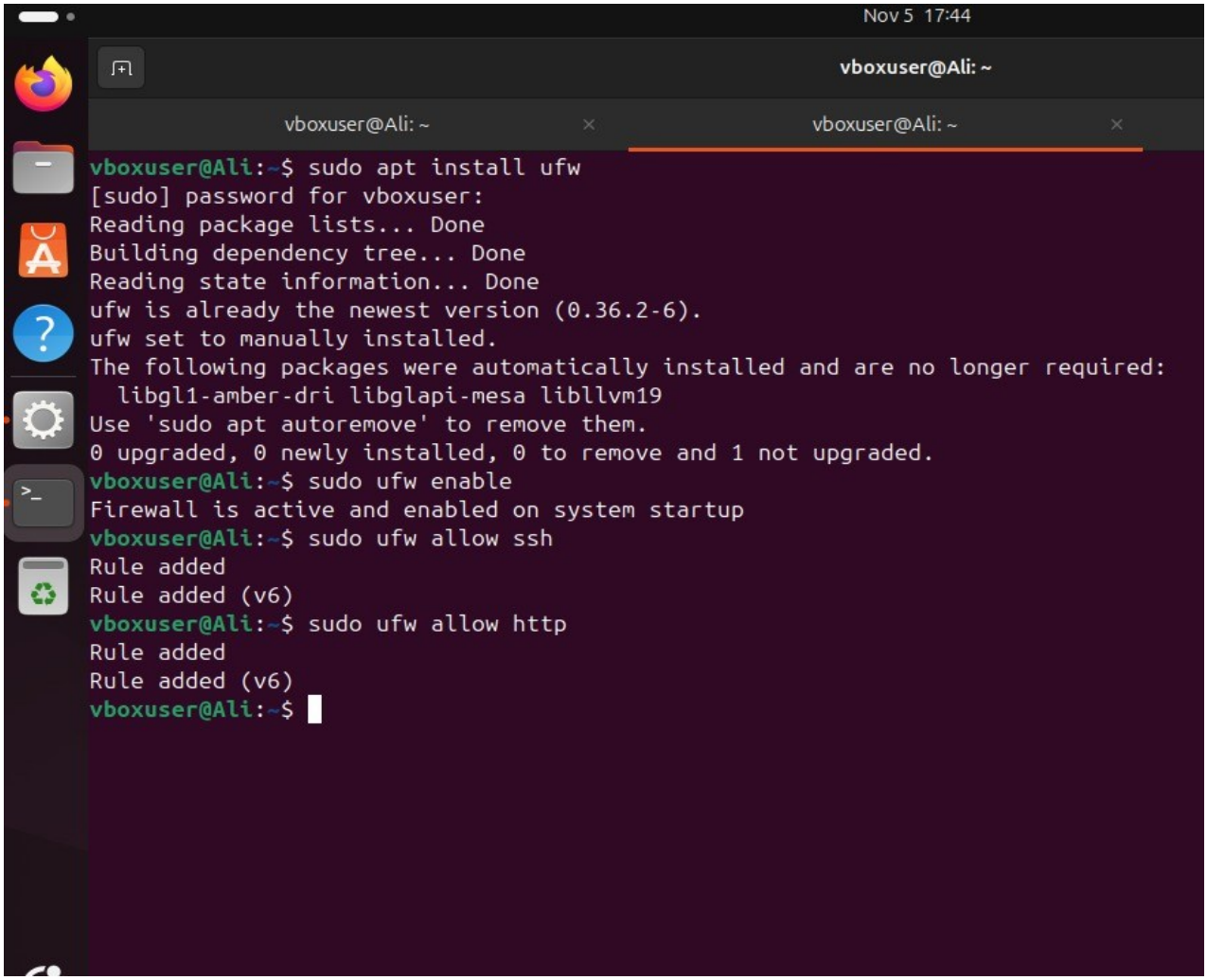
```
anonymous_enable=NO
local_enable=YES
write_enable=YES
chroot_local_user=YES
```

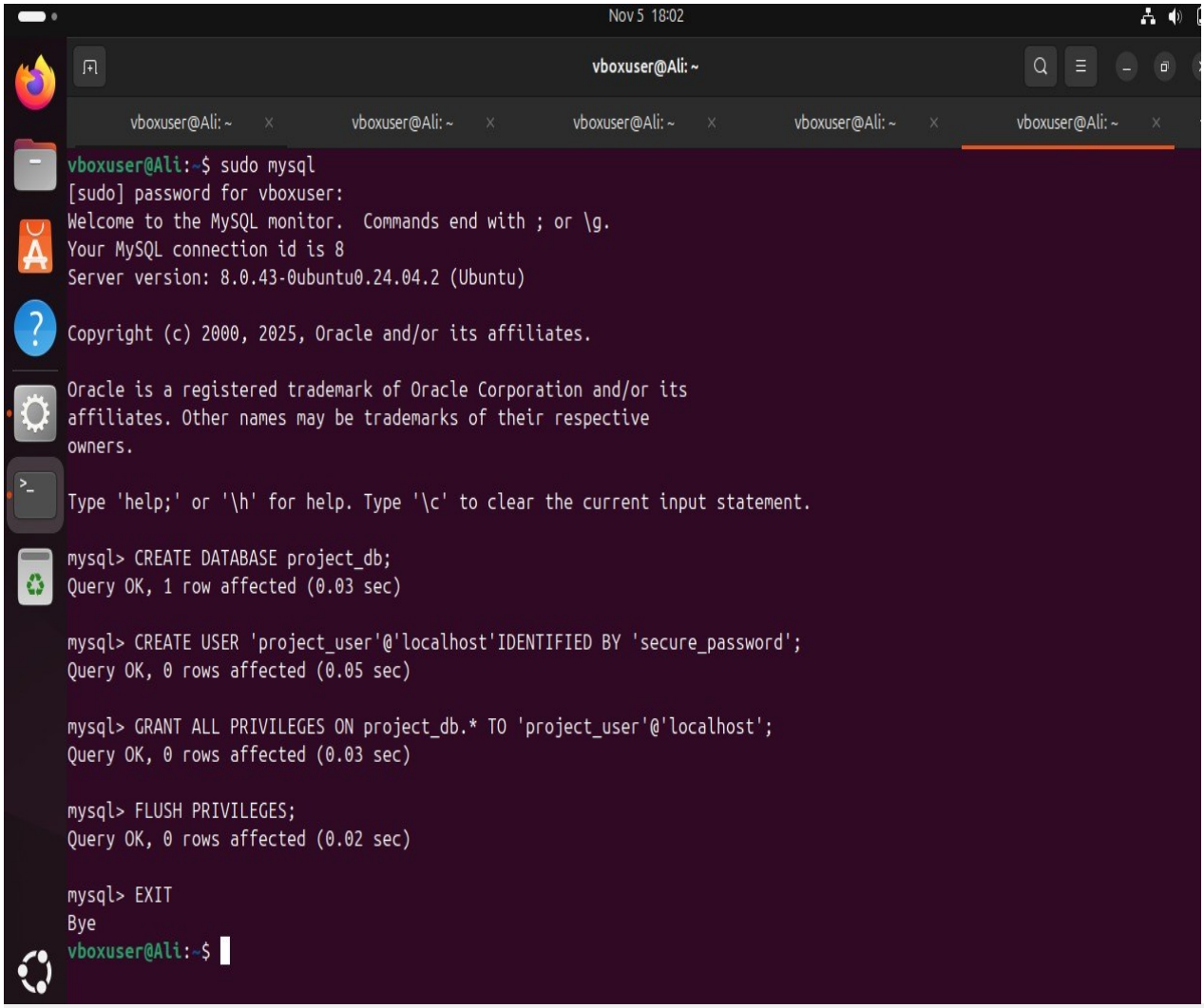
7. OUTPUT:

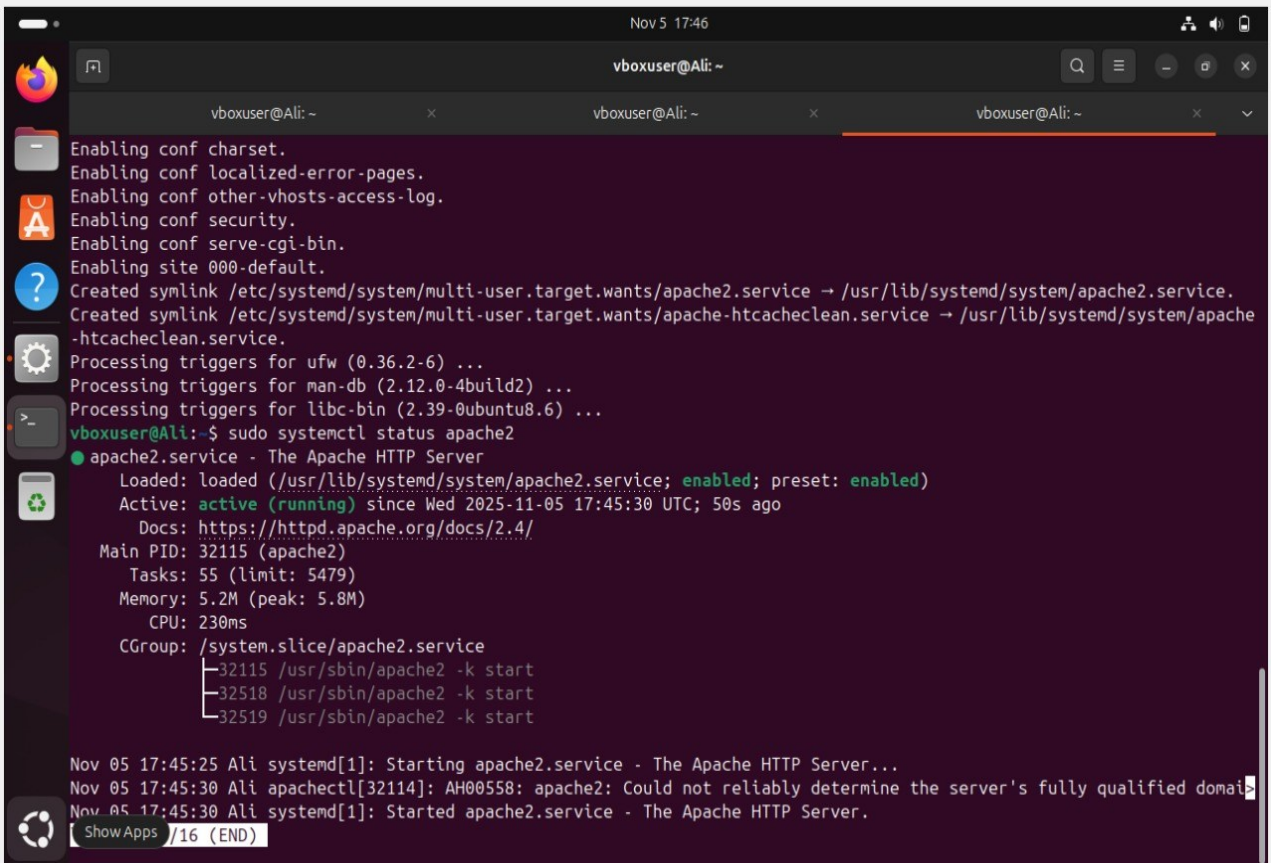
The output was verified by running connectivity tests:

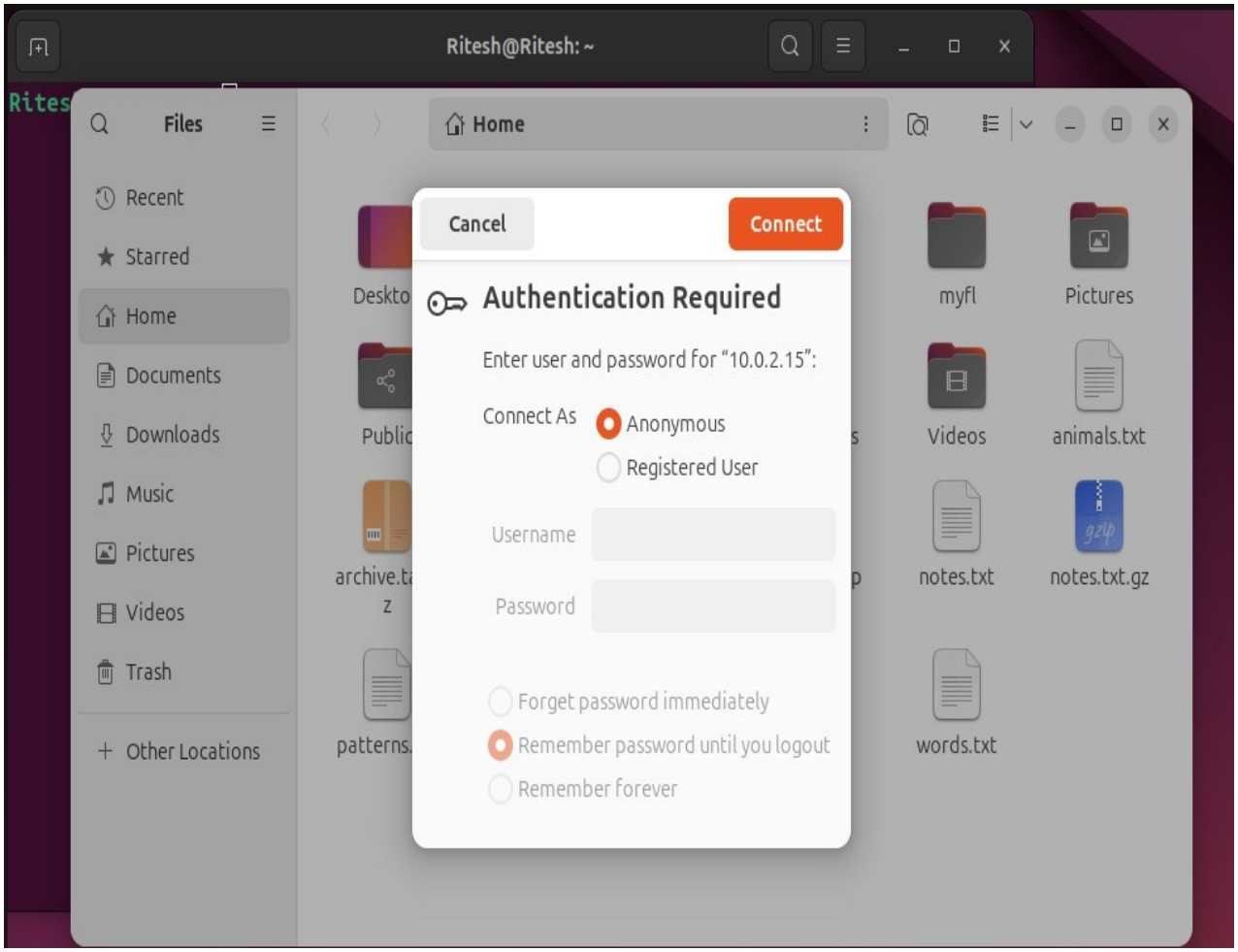
1. **Apache Verification:** Accessing http://[Server IP] from the host machine successfully rendered the custom index.html file.
2. **Firewall Status:** The command sudo ufw status showed the **active** status with rules for **22/tcp (SSH)**, **80/tcp (HTTP)**, **21/tcp (FTP)**, and **3306/tcp (MySQL)** listed as **ALLOW** from anywhere.
3. **MySQL Verification:** The command mysql -u project_user -p successfully connected to the database server using the new credentials.
4. **VSFTPD Verification:** Connecting using a local user via an external FTP client was successful, allowing file upload/download, while an anonymous login attempt was correctly blocked.

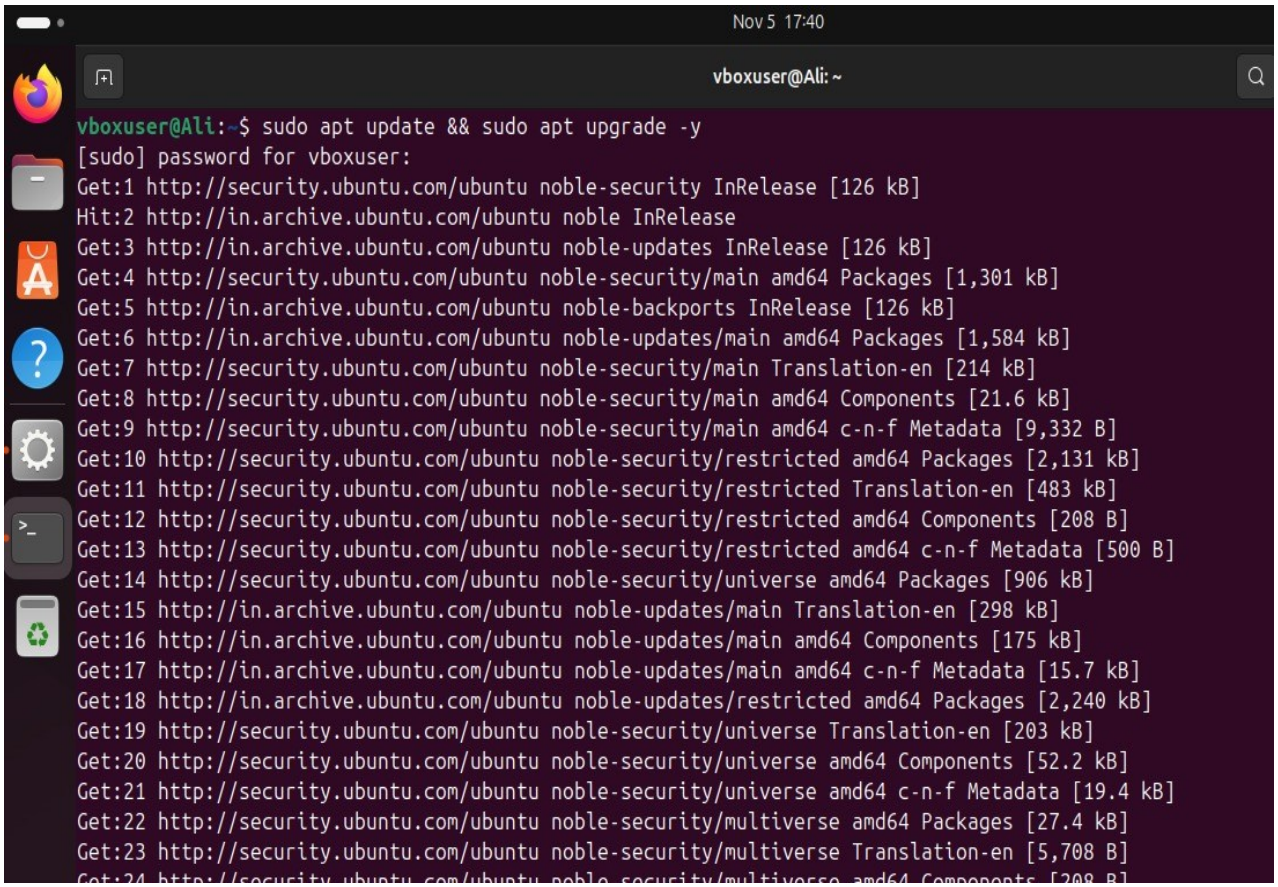
Screenshot:

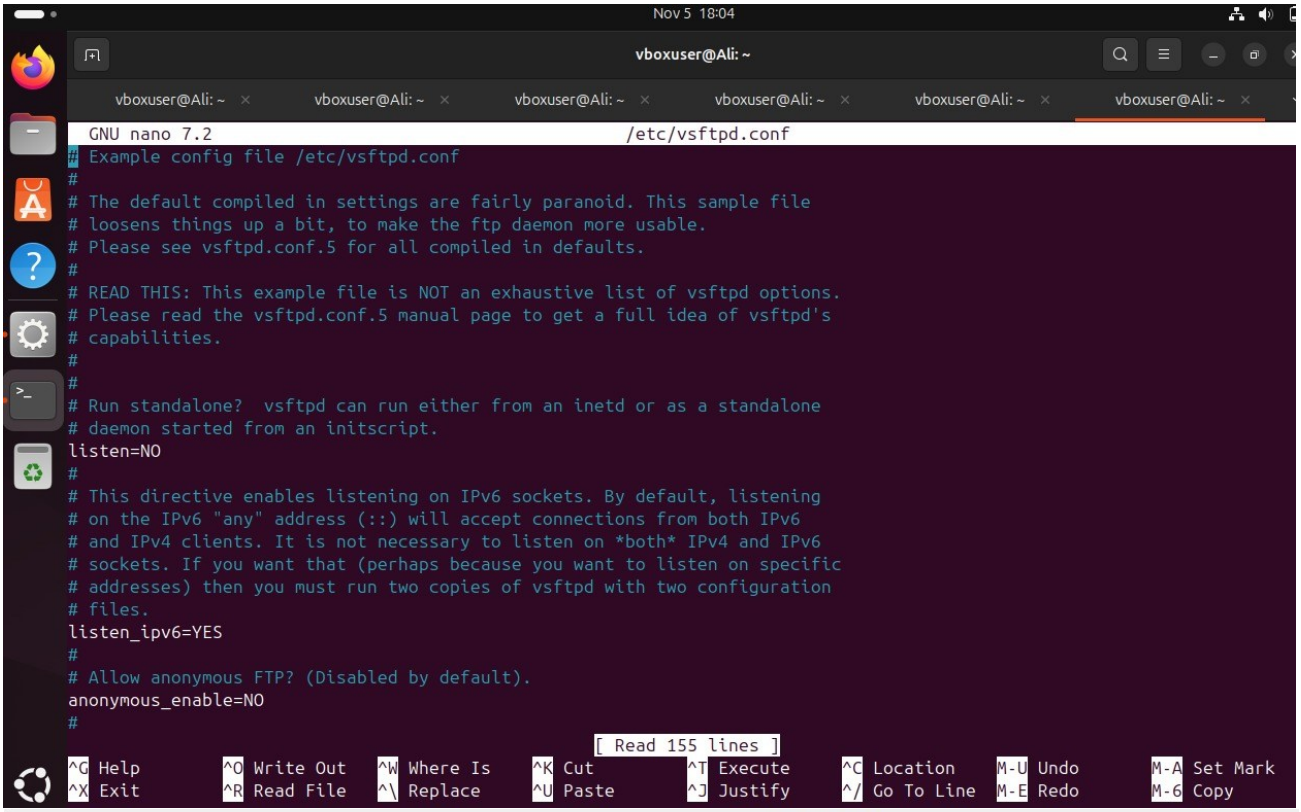


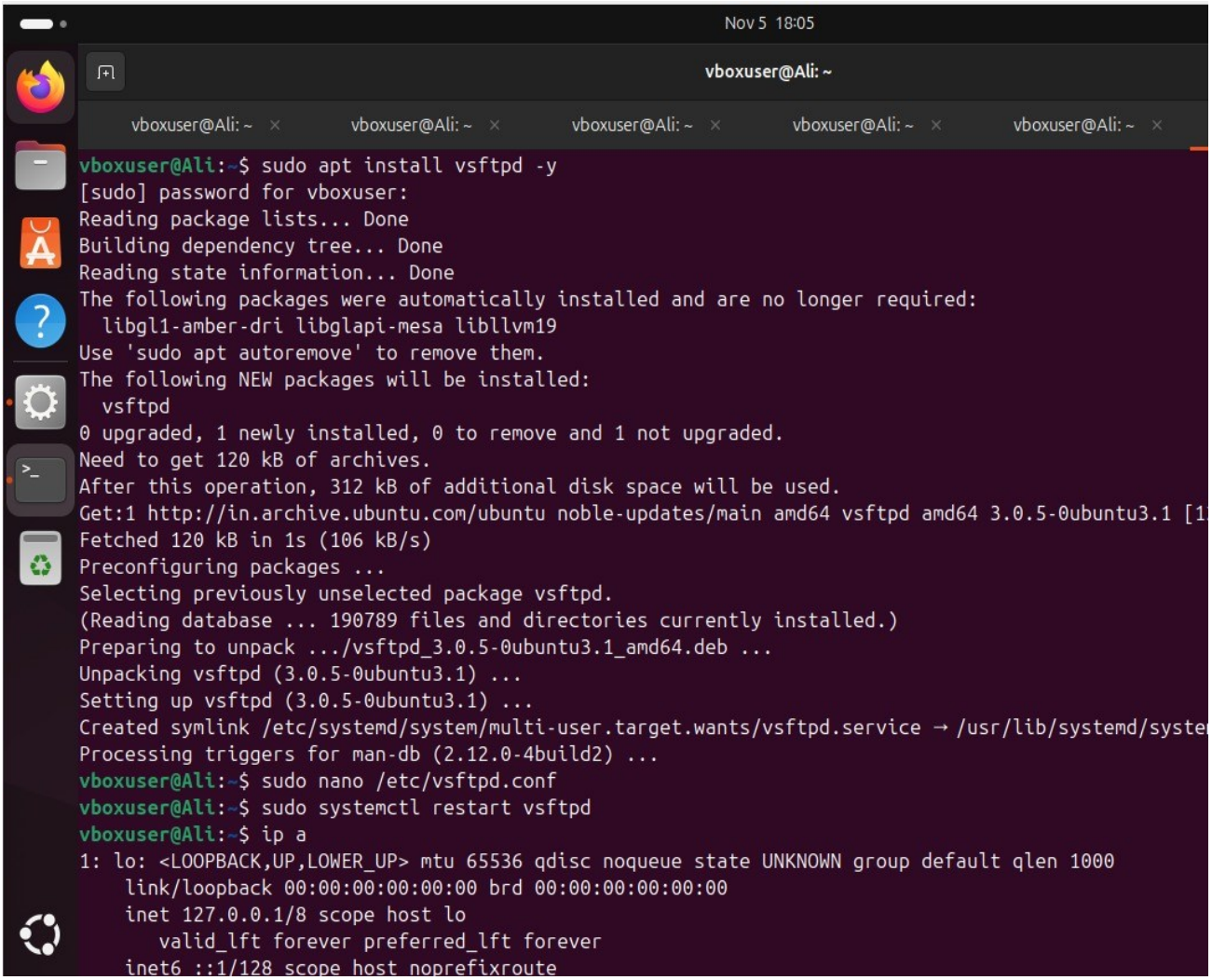

Nov 5 18:02
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali:~$ sudo mysql
[sudo] password for vboxuser:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)
Copyright (c) 2000, 2025, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> CREATE DATABASE project_db;
Query OK, 1 row affected (0.03 sec)
mysql> CREATE USER 'project_user'@'localhost'IDENTIFIED BY 'secure_password';
Query OK, 0 rows affected (0.05 sec)
mysql> GRANT ALL PRIVILEGES ON project_db.* TO 'project_user'@'localhost';
Query OK, 0 rows affected (0.03 sec)
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)
mysql> EXIT
Bye
vboxuser@Ali:~$

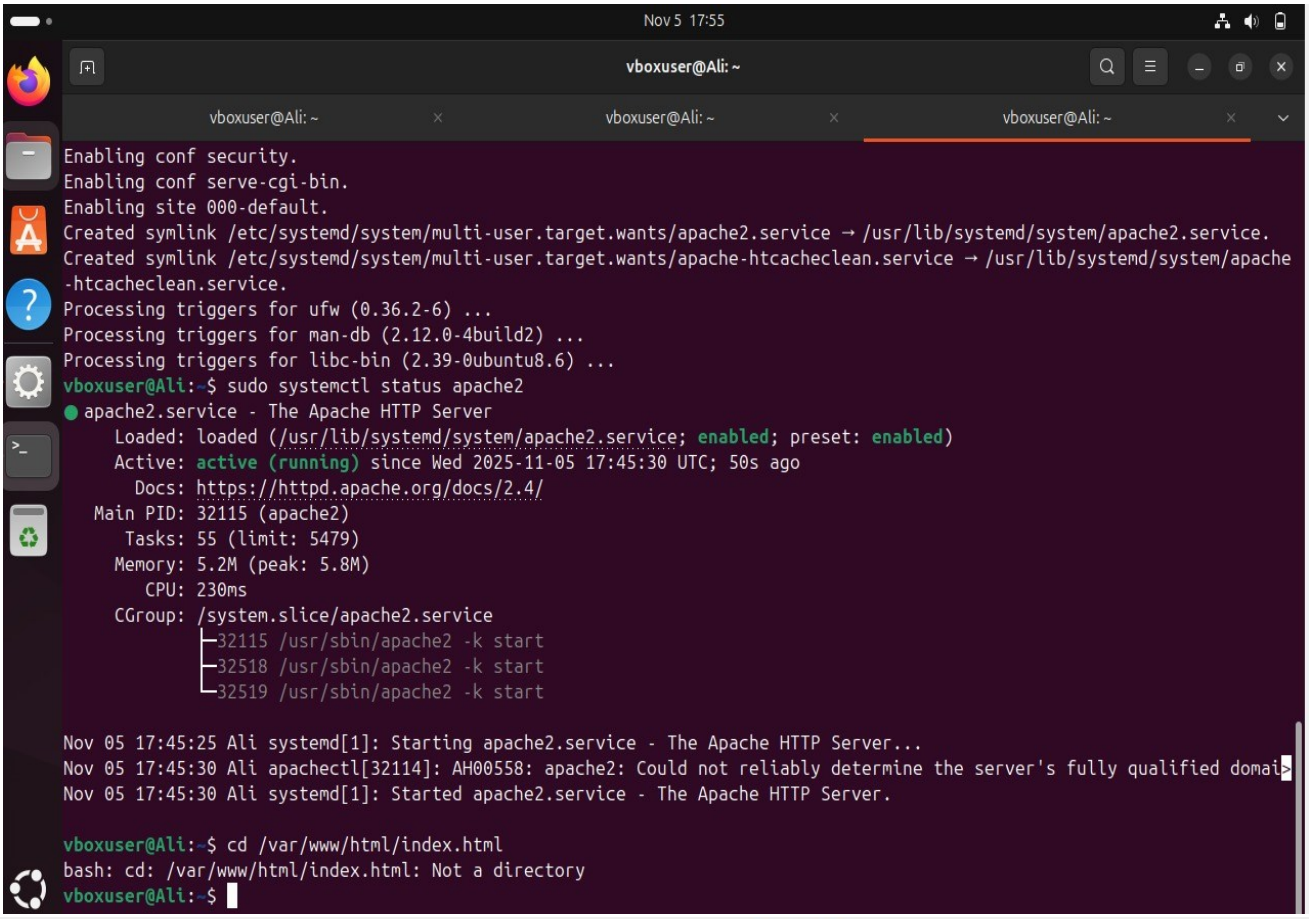

Nov 5 17:46
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache
-htcacheclean.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
vboxuser@Ali:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
Active: active (running) since Wed 2025-11-05 17:45:30 UTC; 50s ago
Docs: https://httpd.apache.org/docs/2.4/
Main PID: 32115 (apache2)
Tasks: 55 (limit: 5479)
Memory: 5.2M (peak: 5.8M)
CPU: 230ms
CGroup: /system.slice/apache2.service
├─32115 /usr/sbin/apache2 -k start
├─32518 /usr/sbin/apache2 -k start
└─32519 /usr/sbin/apache2 -k start
Nov 05 17:45:25 Ali systemd[1]: Starting apache2.service - The Apache HTTP Server...
Nov 05 17:45:30 Ali apachectl[32114]: AH00558: apache2: Could not reliably determine the server's fully qualified domai
Nov 05 17:45:30 Ali systemd[1]: Started apache2.service - The Apache HTTP Server.
Show Apps
/16 (END)

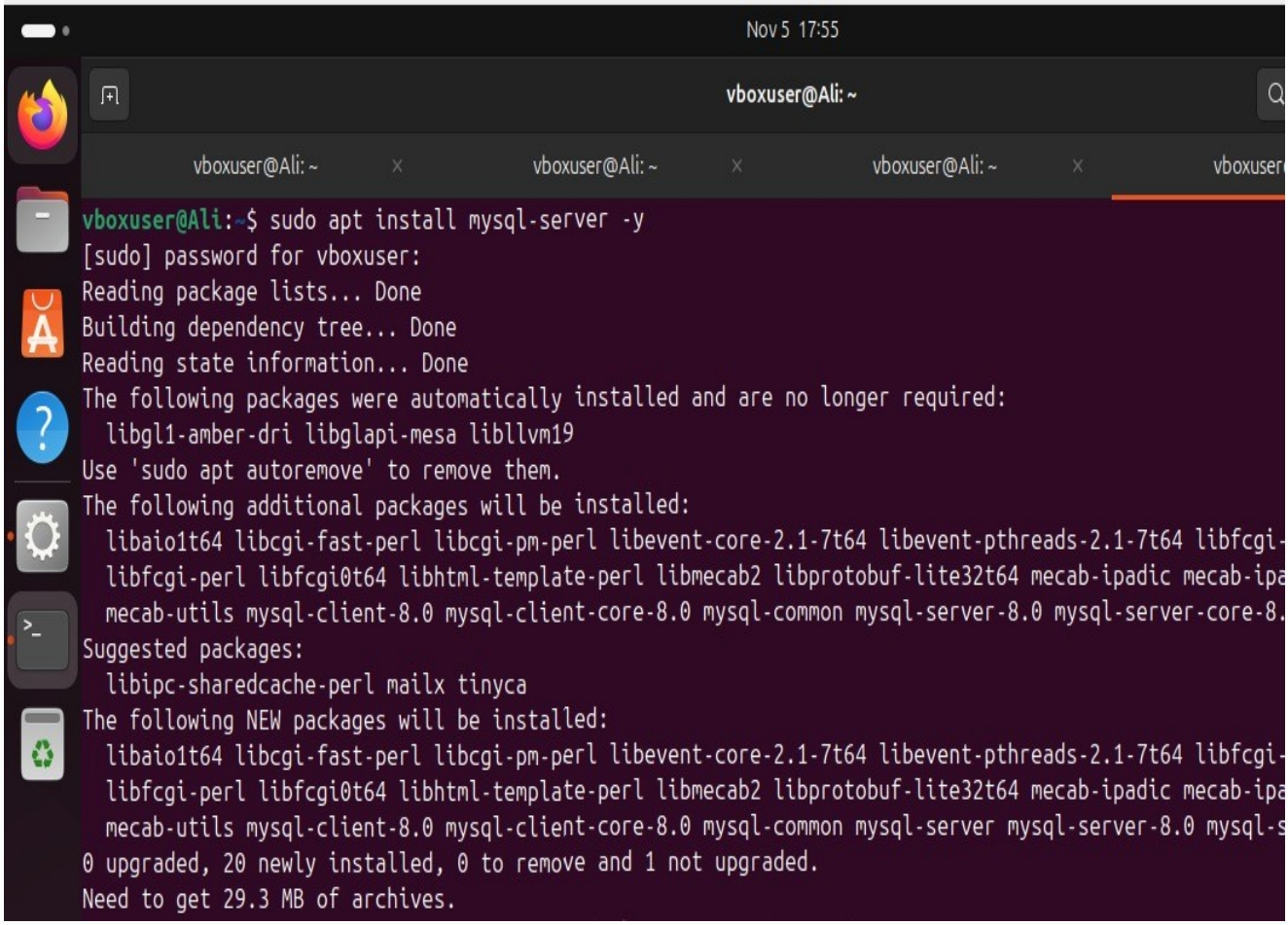

Ritesh@Ritesh: ~
Files
Home
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
Cancel
Connect
Authentication Required
Enter user and password for "10.0.2.15":
Connect As
Anonymous
Registered User
Username
Password
Forget password immediately
Remember password until you logout
Remember forever
myfl
Pictures
Videos
animals.txt
notes.txt
notes.txt.gz
words.txt

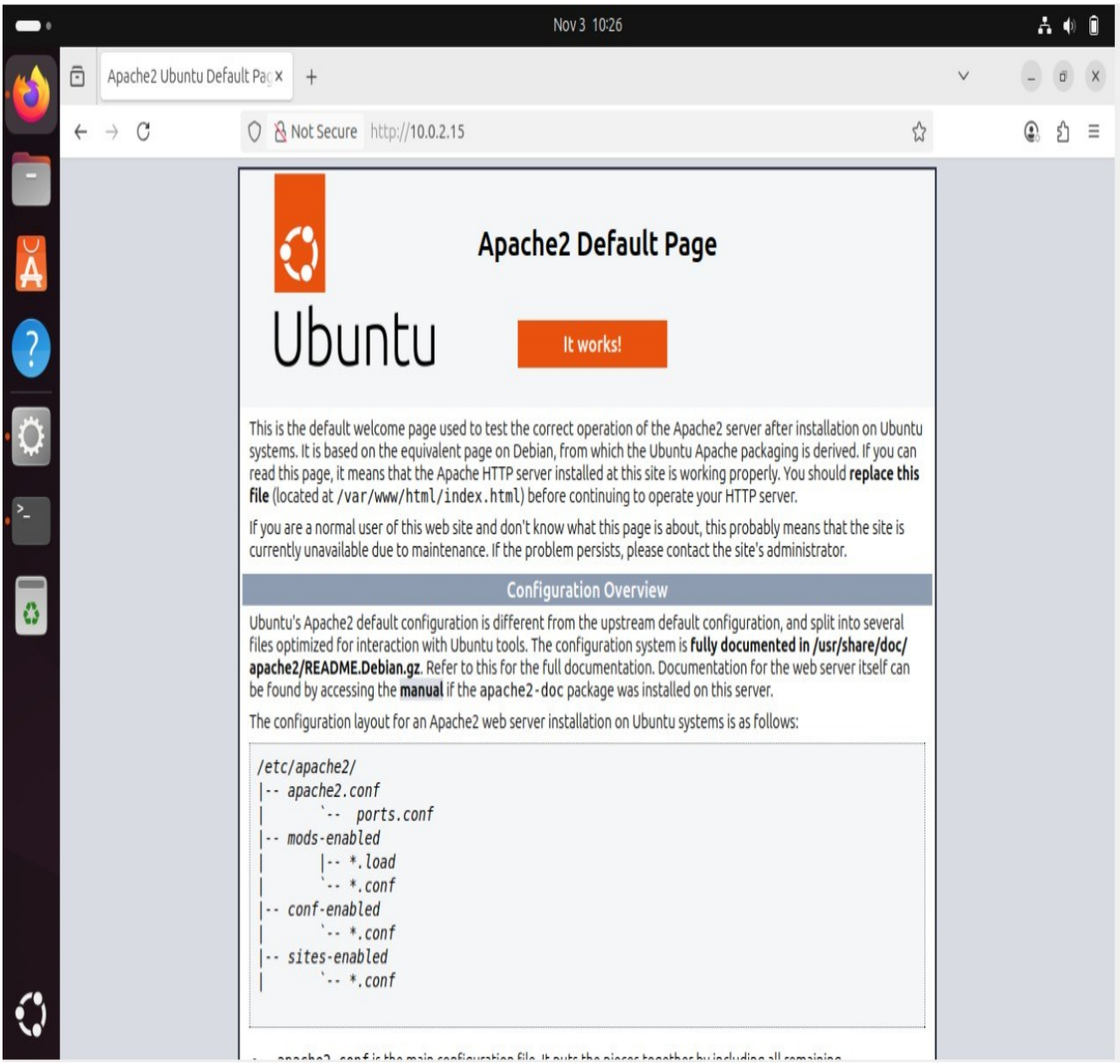

Nov 5 17:40
vboxuser@Ali: ~
vboxuser@Ali:~$ sudo apt update && sudo apt upgrade -y
[sudo] password for vboxuser:
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1,301 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,584 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [214 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [9,332 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2,131 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [483 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
Get:13 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [500 B]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [906 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [298 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.7 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2,240 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [203 kB]
Get:20 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [19.4 kB]
Get:22 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [27.4 kB]
Get:23 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [5,708 B]


Nov 5 18:04
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
GNU nano 7.2 /etc/vsftpd.conf
Example config file /etc/vsftpd.conf
#
The default compiled in settings are fairly paranoid. This sample file
loosens things up a bit, to make the ftp daemon more usable.
Please see vsftpd.conf.5 for all compiled in defaults.
#
READ THIS: This example file is NOT an exhaustive list of vsftpd options.
Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
capabilities.
#
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
[Read 155 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy


Nov 5 18:05
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali:~$ sudo apt install vsftpd -y
[sudo] password for vboxuser:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
libgl1-amber-dri libglapi-mesa libllvm19
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
vsftpd
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 120 kB of archives.
After this operation, 312 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 vsftpd amd64 3.0.5-0ubuntu3.1 [1
Fetched 120 kB in 1s (106 kB/s)
Preconfiguring packages ...
Selecting previously unselected package vsftpd.
(Reading database ... 190789 files and directories currently installed.)
Preparing to unpack .../vsftpd_3.0.5-0ubuntu3.1_amd64.deb ...
Unpacking vsftpd (3.0.5-0ubuntu3.1) ...
Setting up vsftpd (3.0.5-0ubuntu3.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/syste
Processing triggers for man-db (2.12.0-4build2) ...
vboxuser@Ali:~$ sudo nano /etc/vsftpd.conf
vboxuser@Ali:~$ sudo systemctl restart vsftpd
vboxuser@Ali:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute


Nov 5 17:55
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache
-htcacheclean.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
vboxuser@Ali:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
Active: active (running) since Wed 2025-11-05 17:45:30 UTC; 50s ago
Docs: https://httpd.apache.org/docs/2.4/
Main PID: 32115 (apache2)
Tasks: 55 (limit: 5479)
Memory: 5.2M (peak: 5.8M)
CPU: 230ms
CGroup: /system.slice/apache2.service
├─32115 /usr/sbin/apache2 -k start
├─32518 /usr/sbin/apache2 -k start
└─32519 /usr/sbin/apache2 -k start
Nov 05 17:45:25 Ali systemd[1]: Starting apache2.service - The Apache HTTP Server...
Nov 05 17:45:30 Ali apachectl[32114]: AH00558: apache2: Could not reliably determine the server's fully qualified domai
Nov 05 17:45:30 Ali systemd[1]: Started apache2.service - The Apache HTTP Server.
vboxuser@Ali:~$ cd /var/www/html/index.html
bash: cd: /var/www/html/index.html: Not a directory
vboxuser@Ali:~$


Nov 5 17:55
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali: ~
vboxuser@Ali:~$ sudo apt install mysql-server -y
[sudo] password for vboxuser:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
libgl1-amber-dri libglapi-mesa libllvm19
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
libaio1t64 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7t64 libevent-pthreads-2.1-7t64 libfcgi-
libfcgi-perl libfcgi0t64 libhtml-template-perl libmecab2 libprotobuf-lite32t64 mecab-ipadic mecab-ipa
mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.
Suggested packages:
libipc-sharedcache-perl mailx tinyca
The following NEW packages will be installed:
libaio1t64 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7t64 libevent-pthreads-2.1-7t64 libfcgi-
libfcgi-perl libfcgi0t64 libhtml-template-perl libmecab2 libprotobuf-lite32t64 mecab-ipadic mecab-ipa
mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server mysql-server-8.0 mysql-s
0 upgraded, 20 newly installed, 0 to remove and 1 not upgraded.
Need to get 29.3 MB of archives.



9. Conclusion

The "Linux Server Configuration" project successfully achieved the goal of establishing a comprehensive, multi-service server environment. By meticulously configuring Apache, MySQL, and VSFTPD, and securing them with UFW, a foundation for hosting dynamic web applications and managing data securely was built.

Key Achievements:

- Successful integration and configuration of three critical Linux services (Web, Database, File Transfer).

- Implementation of layered security via the UFW firewall.
- Mastery of the systemctl framework for daemon management.

Project Impact:

This project provides a deployable baseline for any developer needing a local or small-scale web stack (LAMP). It serves as a fundamental building block for future projects involving dynamic scripting and database interaction.

Technical Proficiency Demonstrated:

- Command-line proficiency in package management (apt), configuration file editing (nano), and troubleshooting (ip, ss).
- Database security best practices (mysql_secure_installation).
- Network access control and port management (ufw).

Real-World Applicability:

The skills gained are directly applicable to system administration roles, cloud computing platforms (AWS, Azure), and DevOps workflows where managing and provisioning backend services is a daily task.

10. Future Enhancements:

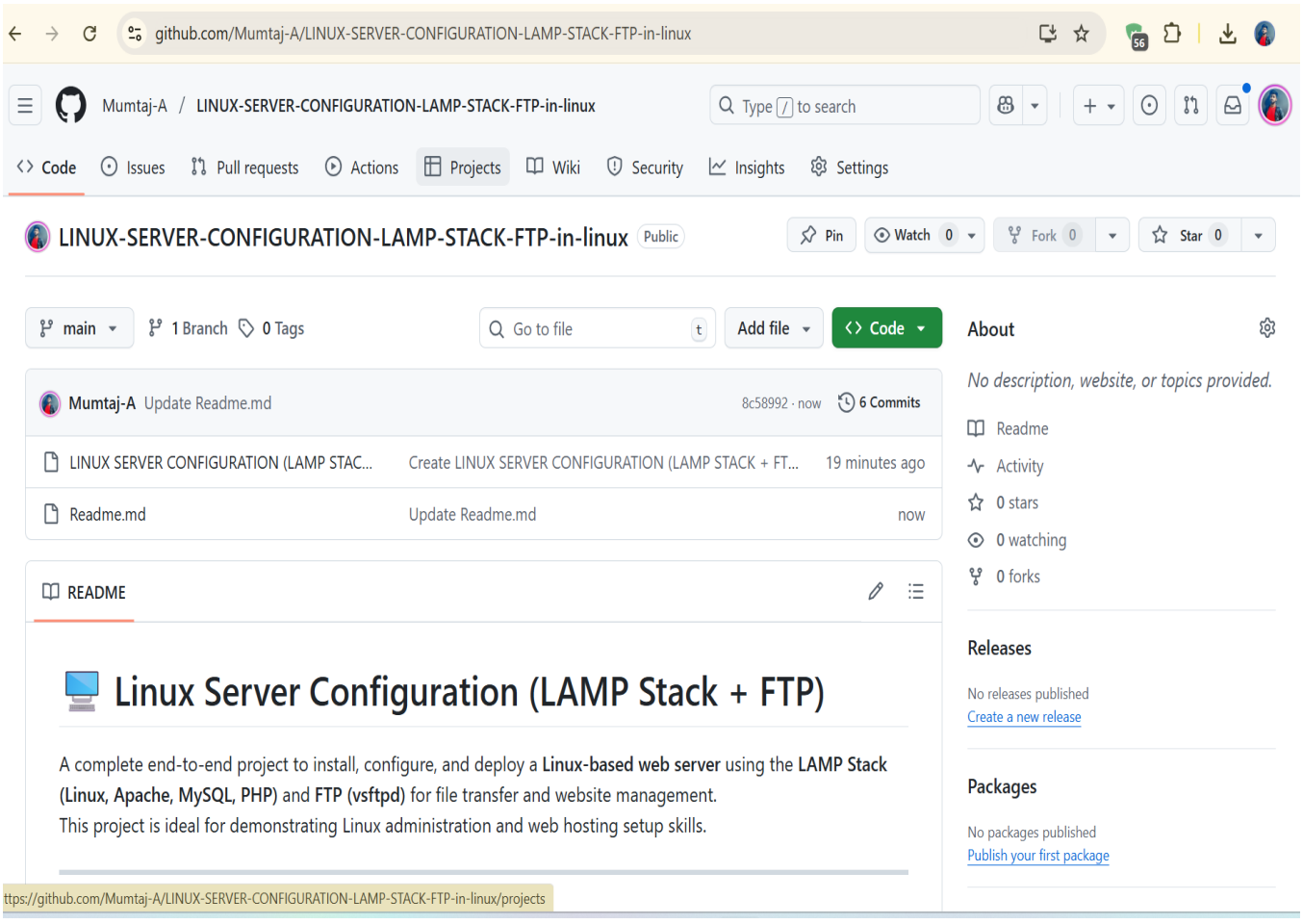
- **DNS Integration:** Adding a BIND9/DNS server to map the server's IP to a friendly domain name.
- **Security Hardening:** Implementing SSL/TLS (HTTPS) on Apache using Let's Encrypt.
- **Automation:** Developing a comprehensive **Shell Scripting Automation** script (Project 2) to deploy this entire stack with a single command.
- **Load Balancing:** Introducing NGINX as a reverse proxy for future load distribution.

11. LEARNING OUTCOMES:

- Mastered the lifecycle of services (start, stop, restart, enable) using systemctl.
- Developed skills in reading, modifying, and troubleshooting common service configuration files (.conf files).
- Gained practical experience in securing a server, specifically through mysql_secure_installation and firewall management.
- Understood the interdependencies between network ports and services.
- Learned to verify and troubleshoot network connectivity using tools like ip a.

GITHUB LINK:

https://github.com/Mumtaj-A/LINUX-SERVER-CONFIGURATION-LAMP-STACK-FTP-in-linux

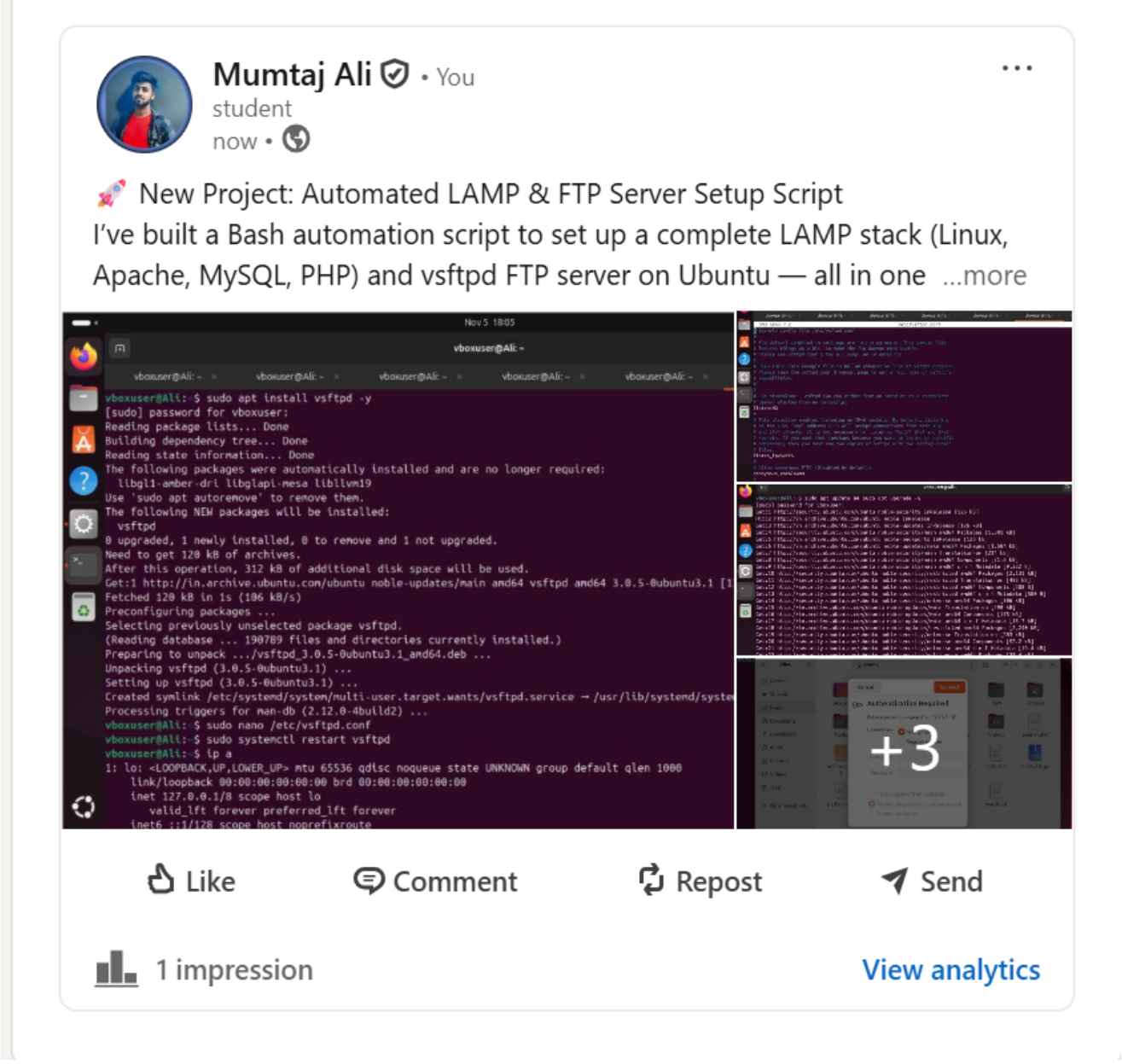


PORTFOLIO LINK:

https://alisoft7.netlify.app/

LINKED IN

Link: https://www.linkedin.com/posts/mumtaj-ali-03865328b_linux-devops-automation-activity-7391913128621465601-NYBx?utm_source=social_share_send&utm_medium=member_desktop_web&rcm=ACoAAEZloWwBdQLoIs041jWX_HP47LjVlV4gMls



14. Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Project Title		2 Marks
2.	Design & Implementation		5 Marks
3.	GitHub Link		1 Mark
4.	LinkedIn Blog Link		1 Mark
5.	Portfolio Link		1 Mark
	TOTAL		**10 Marks**
	AVG		**6 Marks**

Teacher Signature

…….