UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Linux: Project Report

Submitted for partial fulfillment of requirement for the award of

Degree of

Master of Computer Applications

Chandigarh University, Mohali

Submitted By:

Name: Dimpal Tyagi
U.ID: 24MCA20021

Submitted To:

Ms. Geetanjali Sharma

Linux Web Server Setup Project Report

Introduction

In this project, we will set up a basic web server on a Linux system using the Apache HTTP server (httpd). Apache is one of the most widely used web servers and serves webpages to users by hosting HTML, CSS, and other web files. The goal is to install the httpd package, start and configure the Apache service, and create a basic webpage that can be accessed through a browser by navigating to localhost.

This report will detail the step-by-step process of installing and configuring the web server, creating and serving web files, and verifying the server's functionality.

Objectives

The primary objectives of this project are:

1. **Install the Apache HTTP Server (httpd)** on the system.
2. **Start and enable the Apache service** to ensure the web server runs and automatically starts on system boot.
3. **Serve web files** such as HTML and CSS through the Apache web server.
4. **Test the web server** by accessing the hosted webpage via localhost.

Methodology

The process for setting up the web server is divided into the following steps:

1. **Install the httpd package** using the DNF package manager.
2. **Start the httpd service** to ensure the web server is running.
3. **Enable the httpd service** to start automatically upon system boot.
4. **Navigate to the web directory** where Apache serves the files.
5. **Create and edit basic HTML and CSS files** in the web directory.
6. **Verify the setup** by accessing the webpage on localhost.

1. Installing the Apache HTTP Server (httpd)

Apache HTTP Server is packaged as `httpd` in most Linux distributions. The following command installs the package:

sudo dnf install
httpd

```
sudo dnf install httpd
```

- **Explanation**:
  - dnf install httpd: Downloads and installs the httpd package and its dependencies.
  - sudo: Grants superuser privileges necessary for installation.

Outcome:

Once the installation completes, the Apache server is installed but not yet running.

2. Starting the Apache HTTP Service

To run the Apache server, the `httpd` service must be started. This is done with the following command:

sudo systemctl start httpd.service

```
sudo systemctl start httpd.service
```

• **Explanation**:

- systemctl: The command to interact with system services.
- start: Starts the specified service.
- httpd.service: Refers to the Apache HTTP server service.

Outcome:

The Apache server is now running and ready to serve web pages.

3. Enabling Apache to Start on Boot

To ensure that the Apache server starts automatically every time the system reboots, the following command is used to enable the service:

sudo systemctl enable httpd.service

```
sudo systemctl enable httpd.service
```

- **Explanation**:
  - enable: Configures the service to start automatically at system boot.

Outcome:

Apache is now configured to start automatically whenever the system is restarted, ensuring continuous availability of the web server.

4. Navigating to the Web Directory

Apache serves webpages from the /var/www/html directory by default. To place files in this directory, we navigate to it using the following command:

cd /var/www/html

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

- **Explanation**:

- cd: Change directory to /var/www/html, the default web server document root.

Outcome:

We are now inside the web directory where HTML, CSS, and other web files will be stored and served by Apache.

5. Creating HTML and CSS Files

Once inside the /var/www/html directory, the next step is to create the files that will be served by the web server. This project will include a simple HTML file and a CSS file.

Step 5.1: Creating an HTML File

We create an HTML file named index.html using the touch command:

touch index.html

```
touch index.html
```

Step 5.2: Creating a CSS File

Similarly, we create a CSS file named one.css:

touch one.css

```
touch one.css
```

Outcome:

Both the index.html and one.css files are created and ready to be edited.

6. Editing the HTML and CSS Files

Step 6.1: Editing the index.html File

The index.html file will be edited to add basic HTML content. Use the following command to open the file in the vi editor:

sudo vi index.html

```
sudo vi index.html
```

Inside the file, add the following basic HTML structure:

html
Copy
code

```
<!DOCTYPE html>
<html>
<head>
   <link rel="stylesheet" type="text/css" href="one.css">
   <title>My Apache Webpage</title>
</head>
<body>
   <h1>Welcome to My Apache Webpage</h1>
   <p>This is a simple webpage hosted on Apache HTTP Server.</p>
</body>
</html>
```

Save and exit the file by pressing ESC, typing :wq, and hitting Enter.

Step 6.2: Editing the one.css File

Next, edit the one.css file to add some basic styling:

bash Copy
code sudo vi
one.css

Inside the file, add the following CSS code:

css
Copy code

```
body {
background-color: lightblue;
font-family: Arial, sans-serif;
}

h1 {
   color: darkblue;
}
```

```
p {
   color: darkgreen;
}
```

Save and exit the file in the same manner as the index.html file.

Outcome:

The HTML and CSS files are now edited and ready to be served by Apache. The HTML file will display a basic webpage, and the CSS file will style the webpage.

7. Listing the Files in the Directory

To ensure that the files were created correctly, we list the contents of the /var/www/html directory:

bash
Copy
code
ls

Outcome:

The files index.html and one.css should be visible in the directory. This confirms that the files are in the correct location to be served by the Apache web server.

8. Testing the Web Server

Finally, to verify that the Apache server is correctly serving the files, open a web browser and navigate to http://localhost.

- **localhost**: Refers to the local machine. Apache will serve the files from /var/www/html to this address.

Outcome:

The browser should display the webpage with the content from index.html, styled by one.css. If everything is set up correctly, the webpage will show a welcoming message with the text "Welcome to My Apache Webpage" and a background color of light blue.

Summary of Commands

1. **Install the Apache HTTP server**:

   sudo dnf install httpd

```
sudo dnf install httpd
```

2. **Start the Apache service**:

   sudo systemctl start httpd.service

```
sudo systemctl start httpd.service
```

3. **Enable Apache to start on boot** (optional):

   sudo systemctl enable httpd.service

```
sudo systemctl enable httpd.service
```

4. **Navigate to the web directory**:

   cd /var/www/html

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

5. **Create an HTML file**:

   touch index.html

```
touch index.html
```

6. **Create a CSS file**:

   touch one.css

```
touch one.css
```

7. **Edit the HTML file**:

   sudo vi index.html

```
sudo vi index.html
```

8. **Edit the CSS file**:

   sudo vi one.css

```
sudo vi index.html
```

9. **List the files in the directory**:

   ls

```
ls
```

Conclusion

This project successfully demonstrates the installation and configuration of an Apache HTTP server, creating a basic web server environment. By following the outlined steps, we were able to install the httpd package, start and enable the service, create a basic webpage with HTML and CSS, and access it through a web browser using localhost. This setup is foundational for hosting more complex websites and serves as an excellent introduction to

web server management on Linux systems. Further enhancements could include enabling firewall rules for external access, configuring virtual hosts for serving multiple sites, and deploying dynamic web content like PHP or database-driven applications.