 

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

**Create a simple Backup Script**

Name: Monika K Department: ADS



**Introduction and Overview**

In this POC, we will learn how to host a static website locally using the Apache HTTP Server. This process involves setting up a local web server, configuring it correctly, and hosting a simple HTML page. By following these steps, you’ll get hands-on experience with configuring and running a local Apache server, which is a foundational skill for web hosting and server management.

**Objective**

**The goal of this project is to:**

1. Set up a local web server using Apache.
2. Configure the server to host static files.
3. Create and host a simple HTML page displaying your name.

**Importance of Local Hosting**

Local hosting is an essential skill for developers, as it allows them to test and experiment with web applications in a controlled environment. It offers several advantages, such as:

* Hands-On Learning: Gain practical experience with server setup and configuration.
* Testing Ground: Safely test and debug websites before deploying them to a live server.
* Offline Development: Work on web projects without requiring an active internet connection.

Step-by-Step Overview

**Step 1:**  
Search for "Apache Lounge" on Google and click the first link to access the official website.

**Step 2:**  
Click on the "Downloads" option located on the left-hand side of the Apache Lounge website.

**Step 3:**  
Click on the link "Apache 2.4.62-240904 Win64" (Windows version), download the file, and extract all its contents.

**Step 4:**  
Open Command Prompt as Administrator (Windows + R, type cmd, right-click and select 'Run as Administrator') and use the command:

cd C:\path\to\apache\bin

to set the path to the Apache bin folder.

**Step 5:**  
Run the installation command:

httpd.exe -k install

**Step 6:**  
Navigate to the Apache folder you downloaded, go to the conf folder, and right-click on the httpd.conf file; select 'Edit with Notepad' (Apache/conf/httpd.conf).

**Step 7:**  
Inside the httpd.conf file, replace the content with the provided configuration. Ensure you update the SRVROOT directive with your Apache installation path. This configuration defines the server’s root directory, listening port, modules, document root for serving web files, logging paths, and basic permissions, ensuring Apache serves content correctly from the specified htdocs directory.

**Step 8:**  
Open Command Prompt and type the command:

httpd.exe -t

to test the configuration file. If the configuration is correct, you should see 'Syntax OK'.

**Step 9:**  
Run the command:

httpd.exe -k start

to start the Apache server.

**Step 10:**  
Go to the Apache folder, navigate to the htdocs folder, and find the index.html file. Right-click on it and select 'Edit with Notepad'.

**Step 11:**  
Create a simple model to display your name in HTML (you may optionally add CSS for styling).

**Step 12:**  
Open the Chrome browser and type localhost/index.html in the address bar. You should be able to see the website hosted successfully.

git commit -m "Initial commit for feature-branch"

**Step 4:** Merge Changes to Main Branch

git checkout main

git merge feature-branch

**Step 5:** Push to Remote Repository (Optional)

git remote add origin <repository-url>

git push -u origin main

**Simple Backup Script**

Create a simple backup script to back up your project directory:

**backup.sh:**

#!/bin/bash

# Define backup source and destination

SOURCE\_DIR="/path/to/project"

BACKUP\_DIR="/path/to/backup"

TIMESTAMP=$(date +"%Y%m%d\_%H%M%S")

BACKUP\_FILE="$BACKUP\_DIR/backup\_$TIMESTAMP.tar.gz"

# Create backup directory if not exists

mkdir -p "$BACKUP\_DIR"

# Create a compressed backup

tar -czvf "$BACKUP\_FILE" "$SOURCE\_DIR"

# Print completion message

echo "Backup completed: $BACKUP\_FILE"

To use the script, replace /path/to/project and /path/to/backup with your actual directories, then run:

chmod +x backup.sh

./backup.sh