 

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

**Sheel Script To Monitor Logs**

Name: Monika K Department: ADS



**Introduction and Overview**

Deploying a **static website** manually can be time-consuming and repetitive. Automating this process improves efficiency, reduces errors, and ensures that updates are applied consistently.

In this **Proof of Concept (PoC)**, we will automate the **deployment of a static website** using a **local Apache HTTP Server** and a **Shell Script**. This automation ensures that any changes made to the website are quickly deployed without manual intervention.

This guide covers:

* **Setting up an Apache server** for hosting a static website.
* **Writing a deployment script** to automate website updates.
* **Using Git for version control** to track website changes.
* **Scheduling the deployment process** for regular updates.

By following these steps, you will gain **practical experience in automating website deployment** using **Apache, Shell Scripting, and Git**.

**Objectives**

The goal of this project is to:  
✅ Automate the deployment of a **static website**.  
✅ Use **Apache** as a local web server.  
✅ Implement a **Shell Script** to copy files automatically.  
✅ Integrate **Git** to track changes and deploy the latest version.  
✅ Schedule the deployment process using **Cron Jobs** or **Task Scheduler**.

**Importance of Automating Website Deployment**

📌 **Reduces Manual Effort** – No need to manually copy files after every update.  
📌 **Ensures Consistency** – Avoids deployment errors caused by human mistakes.  
📌 **Speeds Up Updates** – Automatically reflects changes in real time.  
📌 **Improves Workflow** – Makes it easier to test and deploy changes.  
📌 **Useful for Developers** – Helps in local development before deploying online.

**Step-by-Step Overview**

**Step 1: Install Apache HTTP Server**

1. **For Windows:**
   * Download **Apache** from [Apache Lounge](https://www.apachelounge.com/).
   * Extract the files and place them in C:\Apache24.
   * Open Command Prompt as Administrator and run:

CopyEdit

**cd C:\Apache24\bin**

**httpd.exe -k install**

* + Start Apache:

CopyEdit

**httpd.exe -k start**

1. **For Linux/Mac:**
   * Open Terminal and run:

CopyEdit

**sudo apt update && sudo apt install apache2 -y # For Ubuntu/Debian**

**sudo systemctl start apache2**

# Start Apache

* + For Mac, use:

CopyEdit

**sudo apachectl start**

**Step 2: Set Up the Static Website**

1. Create a directory for your website inside Apache's root folder:
   * **Windows:** C:\Apache24\htdocs\mywebsite
   * **Linux/Mac:** /var/www/html/mywebsite
2. Create an index.html file inside the folder with the following content:

html

CopyEdit

**<!DOCTYPE html>**

**<html>**

**<head><title>My Automated Website</title></head>**

**<body>**

**<h1>Welcome to My Automated Static Website</h1>**

**<p>Deployment is now automated!</p>**

**</body>**

**</html>**

**Step 3: Create a Deployment Script**

To automate deployment, we will use a **Shell Script** that:  
✅ **Pulls the latest changes from Git**  
✅ **Copies website files to the Apache directory**  
✅ **Restarts the Apache server**

1. Open Terminal and create a new script file:

CopyEdit

**nano deploy\_website.sh**

1. Add the following script:

CopyEdit

**#!/bin/bash**

# Define paths

**GIT\_REPO="/home/user/mywebsite"**

# Change this to your Git repository path

**DEPLOY\_PATH="/var/www/html/mywebsite**"

# Pull latest changes from Git

**cd "$GIT\_REPO"**

**git pull origin main**

# Copy updated files to Apache directory

**cp -r "$GIT\_REPO"/\* "$DEPLOY\_PATH"**

# Restart Apache server to apply changes

**sudo systemctl restart apache2**

# For Linux

# httpd.exe -k restart

# For Windows

echo "Website successfully deployed on $(date)" >> deployment\_log.txt

1. Save and exit (CTRL + X, then Y and Enter).
2. Make the script executable:

CopyEdit

**chmod +x deploy\_website.sh**

**Step 4: Automate Deployment with Cron Jobs (Linux/Mac) or Task Scheduler (Windows)**

**For Linux/Mac (Cron Job)**

1. Open the crontab file:

CopyEdit

**crontab -e**

1. Add the following line to **run the deployment script every 5 minutes**:

CopyEdit

**\*/5 \* \* \* \* /path/to/deploy\_website.sh**

1. Save and exit.

**For Windows (Task Scheduler)**

1. Press **Win + R**, type taskschd.msc, and hit **Enter**.
2. Click **Create Basic Task** → Name it **"Automate Website Deployment"**.
3. Choose **Daily** or another schedule.
4. Select **"Start a Program"** → Browse to deploy\_website.bat (Batch script to execute deploy\_website.sh).
5. Click **Finish** and run the task.