 

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

Automate static website Deployment Locally

Create a script that updates your server whenever changes are pushed

Name: S Akshaya Department:ADS



**Introduction**

In modern web development, ensuring that your website is automatically updated whenever changes are made is crucial for efficient workflows. Static websites, in particular, benefit greatly from automation, as you can easily deploy changes with minimal manual intervention. This document will guide you through automating static website deployment locally using Git hooks and scripting. The goal is to ensure that whenever code changes are pushed to a Git repository, the corresponding server will be updated automatically.

**Objective**

The objective of this guide is to teach you how to:

1. **Set up Git hooks** to automate the deployment process.
2. **Write a deployment script** that pulls the latest changes from a repository and restarts the web server.
3. **Configure web servers (Apache/Nginx)** to serve your static website, ensuring changes are immediately visible once deployed.

**Basics of Git Hooks**

Git hooks are scripts that Git executes before or after certain Git commands. These hooks allow you to trigger specific actions in response to events like pushing changes or committing code. The most commonly used hook for deployment is the **post-receive** hook, which gets triggered when new commits are received in the repository.

**Step 1: Setting Up Git Hooks for Deployment**

1. **Navigate to the hooks directory** inside your local or remote Git repository:

cd /path/to/your/repository/.git/hooks

1. **Create a post-receive hook** to automate deployment when changes are pushed. This script will run after each successful push to the repository.

echo "Starting deployment..."

# Pull the latest code

git pull origin main

# Run your deployment script (e.g., restart Apache or Nginx)

bash /path/to/your/deploy-script.sh

1. **Make the script executable**:

chmod +x post-receive

Now, whenever you push to the repository, Git will automatically trigger the post-receive hook, pulling the latest code and running the deployment script.

**Step 2: Writing a Deployment Script**

The deployment script can perform tasks such as pulling the latest code from the repository, restarting the web server (Apache or Nginx), or clearing the cache.

# Navigate to your website's directory

cd /path/to/your/static/website

# Pull the latest changes from the Git repository

git pull origin main

# Restart Apache

sudo systemctl restart apache2

**Step 3: Restarting Services (Apache)**

When deploying updates, you’ll need to restart the web server to ensure that the latest files are served to visitors.

* **For Apache**:

sudo systemctl restart apache2

**Conclusion**

Automating static website deployment using Git hooks and scripts is a powerful way to streamline your development workflow. With just a few steps, you can ensure that changes are automatically deployed and that your web server always serves the latest version of your site. Whether you're working with Apache or Nginx, integrating Git hooks and scripts makes deployment fast and efficient.