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: Kevin Christopher Ruban Department: AI-ML

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY
AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

Introduction and Overview

This document outlines the step-by-step procedure for automating the deployment of a static website from a Git repository to a local server using a deployment script (deploy.bat). This automation ensures that any changes made to the HTML files in the repository are immediately replicated in the deployment directory, facilitating seamless updates and reducing manual intervention.

Objectives

- Automate the deployment of changes made to a static website in a Git repository.
- Use a PowerShell script (deploy.bat) to pull changes from the repository and update the deployment folder.
- Demonstrate how to set up a post-receive Git hook to trigger the deployment process.
- Ensure changes to the repository reflect automatically in the deployment directory.

Importance

- **Efficiency**: Automates the process of deploying website updates, reducing time and effort.
- **Consistency**: Ensures the deployment folder always reflects the latest repository changes.
- **Reduced Errors**: Minimizes the chances of human error by automating the entire deployment flow.
- **Version Control**: By using Git, it provides versioning and history of changes made to the website files.

STEPS:

STEP 1: Adding the HTML Website to the Repository

1. Navigate to the folder where your static website files are stored.
2. Open Git Bash or your terminal.
3. Initialize the repository (if not already done) by running:

   git init

4. Add your HTML files (or any other website files) to the repository:

   git add .

5. Commit the files to the repository:

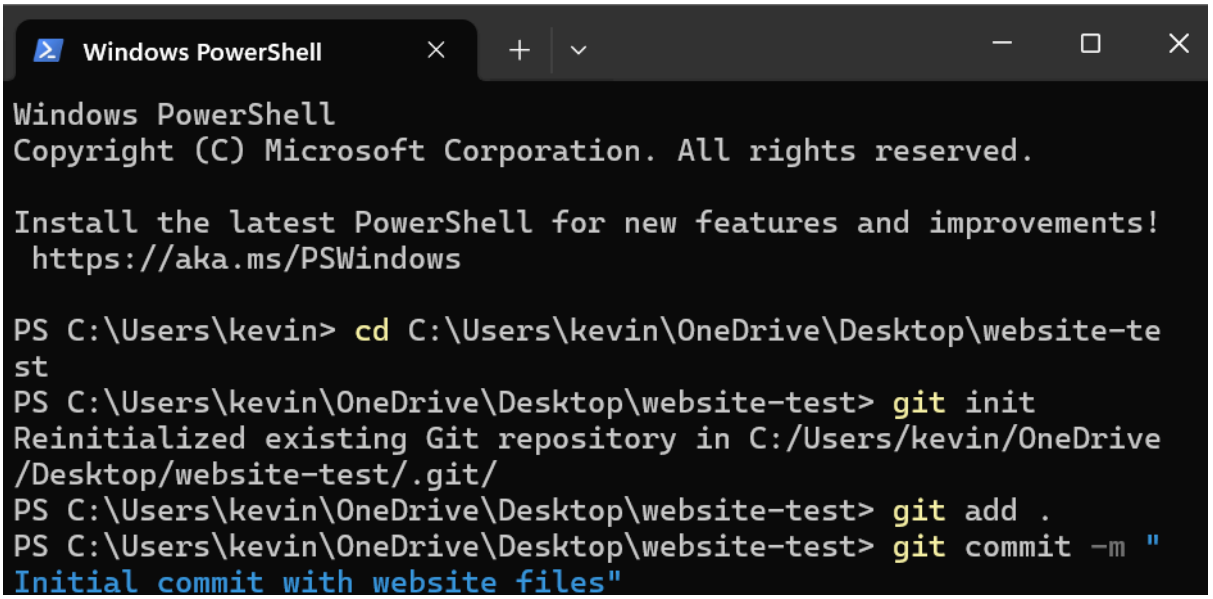
   git commit -m "Initial commit with website files"

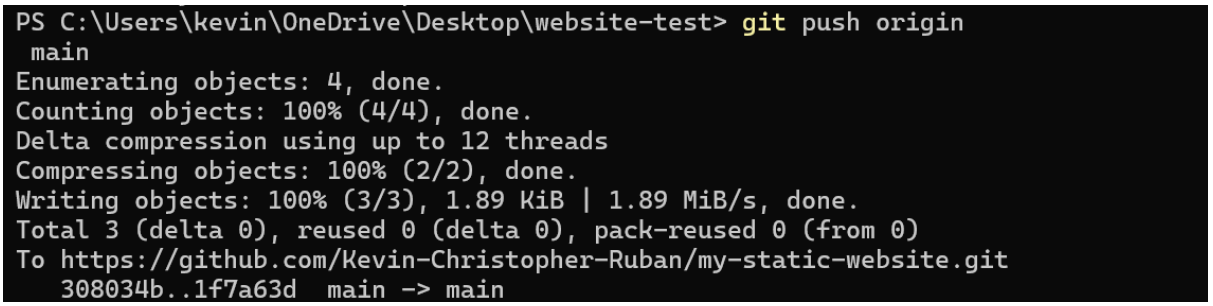
6. Link the repository to your remote GitHub repository (replace the URL with your actual repository URL):

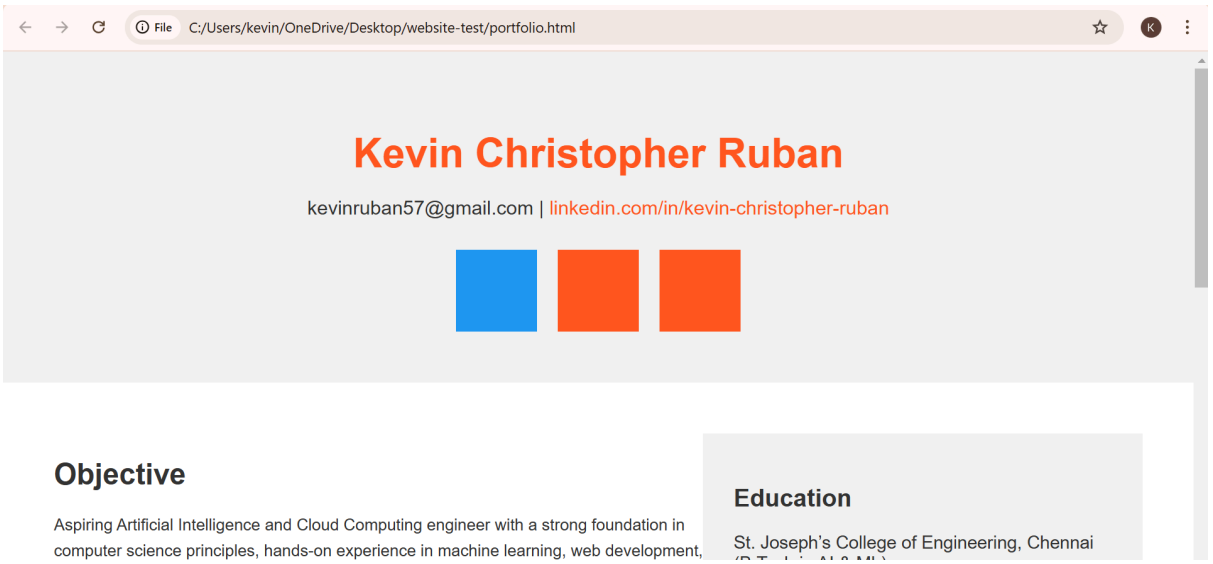
   git remote add origin https://github.com/yourusername/yourrepository.git

7. Push the changes to the remote repository:

   Git push -u origin main







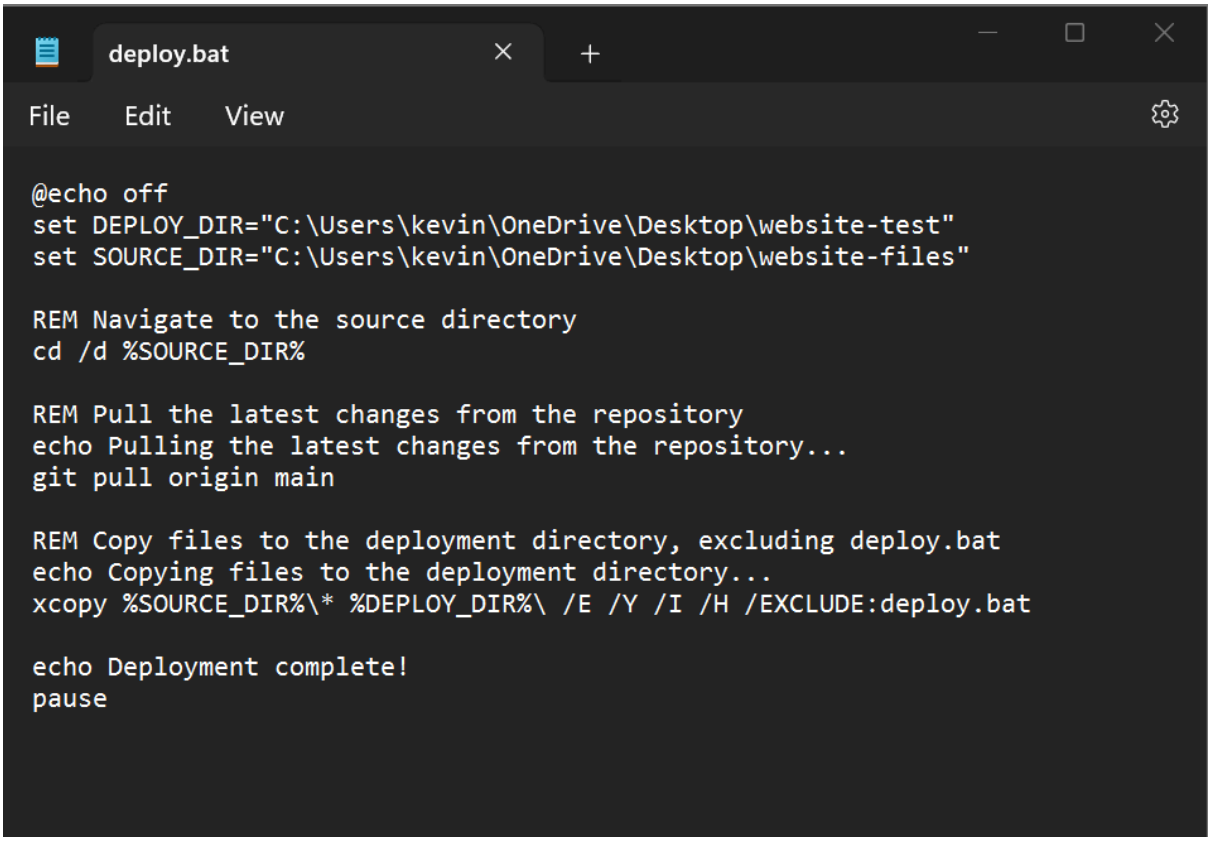
Step 2: Creating the deploy.bat File

1. In your project folder, create a new text file and rename it to deploy.bat.

2. Open the file in a text editor (e.g., Notepad++).
3. Paste the following script in the file:

```
@echo off
set DEPLOY_DIR="C:\Users\kevin\OneDrive\Desktop\website-test"
set SOURCE_DIR="C:\Users\kevin\OneDrive\Desktop\website-files"

REM Navigate to the source directory
cd /d %SOURCE_DIR%

REM Pull the latest changes from the repository
echo Pulling the latest changes from the repository...
git pull origin main

REM Copy files to the deployment directory, excluding deploy.bat
echo Copying files to the deployment directory...
xcopy %SOURCE_DIR%\* %DEPLOY_DIR%\ /E /Y /I /H
/EXCLUDE:deploy.bat

echo Deployment complete!
pause
```

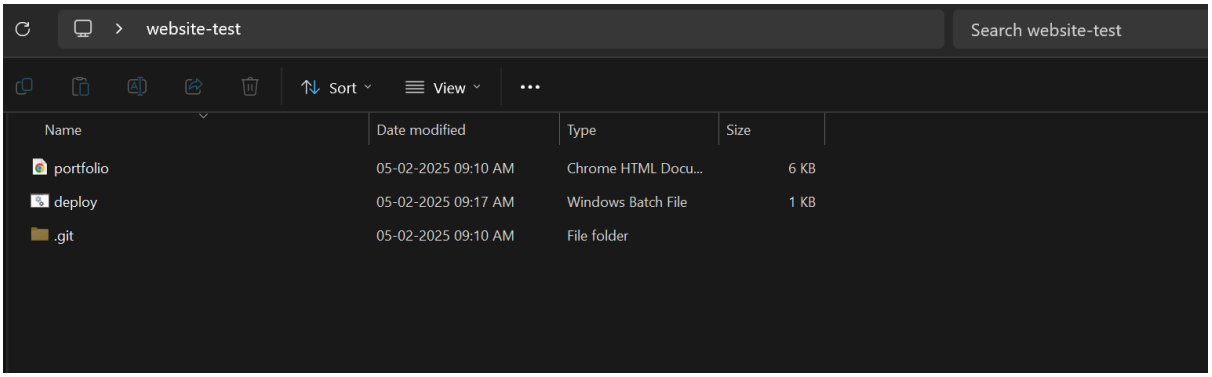
4. Save the file.



Step 3: Revealing the .git Folder

By default, Git hides the .git folder. If you need to make it visible for some reason, follow these steps:

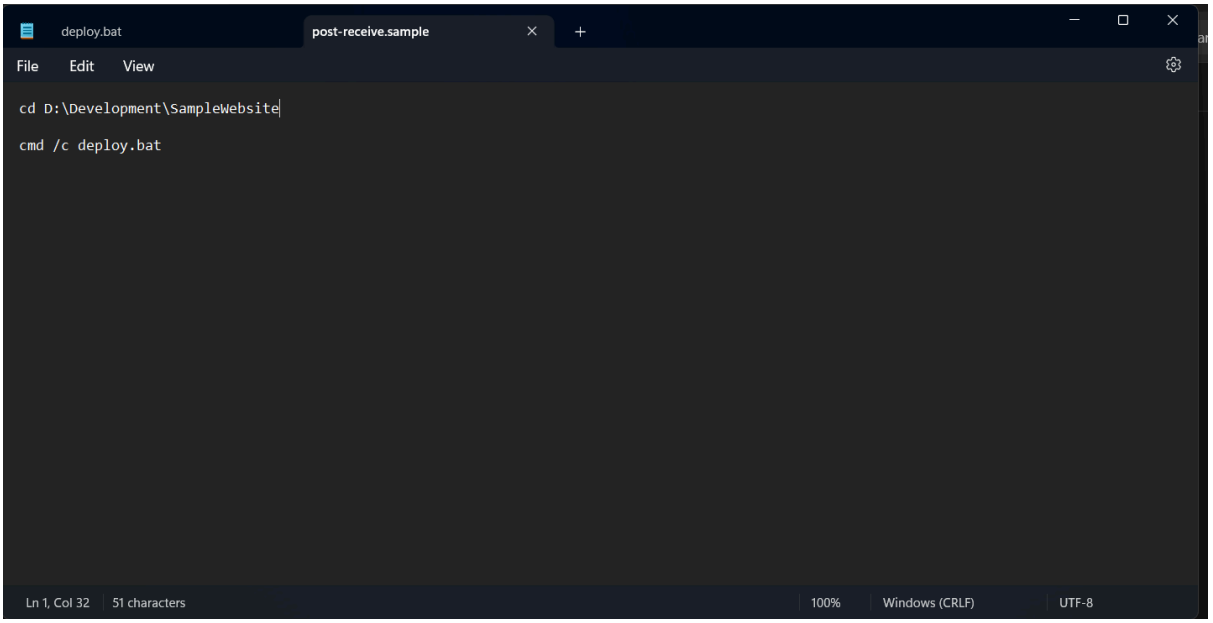
1. Open File Explorer.
2. Navigate to your project directory.
3. In the File Explorer menu, click on **View → Show → Hidden items**.
4. The .git folder should now be visible. If it is still hidden, check your folder options and ensure "Hide protected operating system files" is unchecked.



Step 4: Creating the post-receive.sample File

1. Navigate to your .git\hooks folder.
2. Create a new file named post-receive (with the .sample extension).
3. Open the file and paste the following script:

```
#!/bin/bash

# Navigate to the project directory

cd /path/to/your/project/directory


 # Execute the deployment script

 cmd /c deploy.bat
```



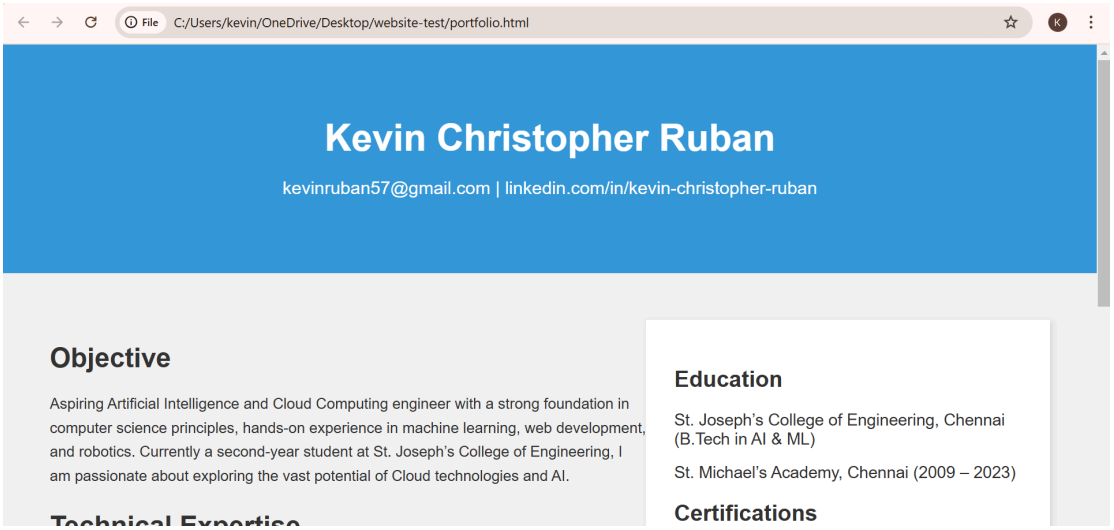
4. Replace /path/to/your/project/directory with the actual path to your repository directory.
5. Make the post-receive hook executable by running:
    - **cd C:\Users\kevin\OneDrive\Desktop\website-test\.git\hooks**
    - **Unblock-File -Path .\post-receive.sample**

Step 5: Setting Up the Source and Deployment Directories

1. Keep your deploy.bat file in the source repository directory.
2. Create a new directory where the autonomous deployment will occur. This will be the directory where the latest changes will be copied after they are pulled from the repository.
3. Ensure both the source and deployment directories are correctly defined in the deploy.bat file.

Step 6: Testing the Deployment Process

1. Make a change to your HTML file (e.g., index.html) in the repository.



2. Stage and commit the change:

   git add index.html

   git commit -m "Updated index.html"

3. Push the changes to the remote repository:

   git push origin main

4. Open **PowerShell** as an administrator.

5. Navigate to the directory where the deploy.bat file is located:

   cd C:\Users\kevin\OneDrive\Desktop\website-files

6. Run the deploy.bat script:

   .\deploy.bat

7. Check if the files were copied successfully to the deployment directory (C:\Users\kevin\OneDrive\Desktop\website-files).

8. Verify that the changes made in the HTML file are reflected in the deployment directory.