St. JOSEPH'S
GROUP OF INSTITUTIONS
OMR, CHENNAI - 119

aws

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: Keerthana Sri G **Department : ECE**

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

Introduction

Efficient deployment processes are crucial for maintaining up-to-date static websites. Automating the deployment of static websites ensures that the latest changes are reflected on the server as soon as they are pushed to the repository. This document outlines how to create a script that updates the server with new changes and restarts the necessary services, using Git hooks and scripting techniques.

Objectives

- Understand the use of Git hooks for automating deployment.
- Learn to write scripts for pulling changes and managing server services.
- Automate the deployment process to minimize manual intervention.

Understanding Key Concepts

- Git Hooks: Scripts that Git executes automatically on specific events like commits or pushes.
- File Deployment: Updating the server files with the latest code changes.
- Server Management: Restarting services (e.g., Apache or Nginx) to apply changes.

Step by Step Overview

1. Set Up Your Local Development Environment

Ensure you have the necessary tools installed:

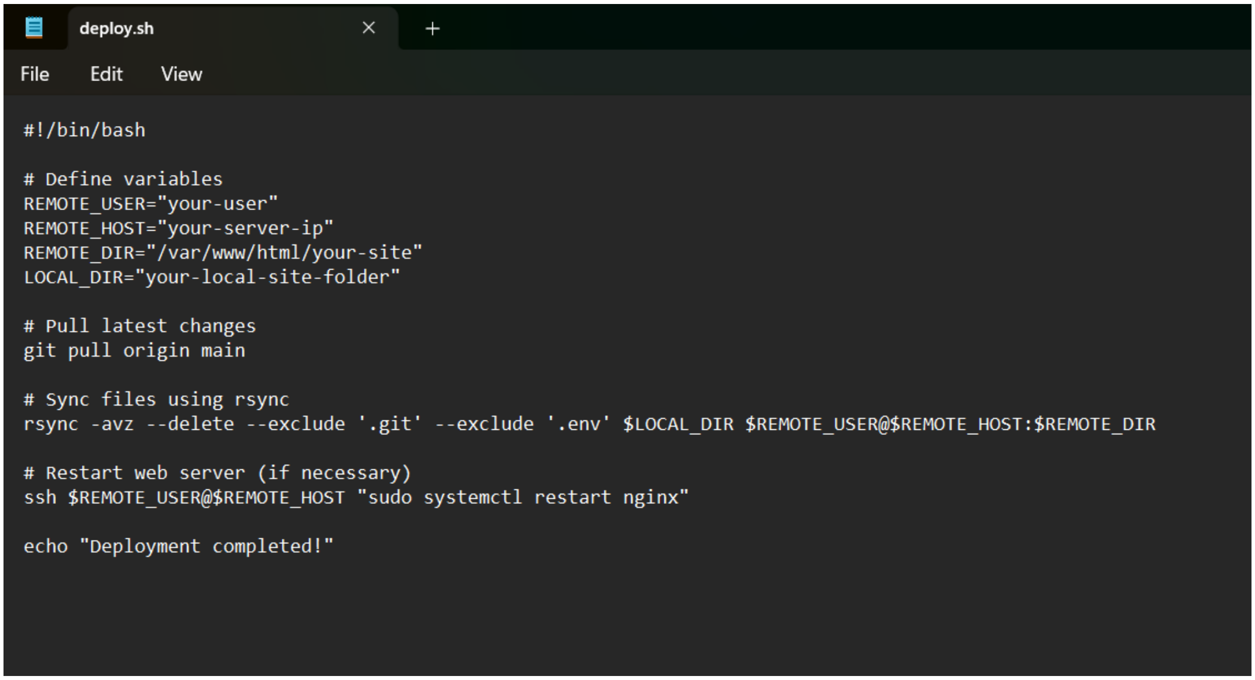
- Git
- SSH access to the server

2. Configure Your Remote Server

- Set up a web server (e.g., Apache)
- Ensure the deployment directory exists (e.g., /var/www/html/your-site)

3. Create a Deployment Script

Create a script (deploy.sh) in your project root:



```
#!/bin/bash

# Define variables
REMOTE_USER="your-user"
REMOTE_HOST="your-server-ip"
REMOTE_DIR="/var/www/html/your-site"
LOCAL_DIR="your-local-site-folder"

# Pull latest changes
git pull origin main

# Sync files using rsync
rsync -avz --delete --exclude '.git' --exclude '.env' $LOCAL_DIR $REMOTE_USER@$REMOTE_HOST:$REMOTE_DIR

# Restart web server (if necessary)
ssh $REMOTE_USER@$REMOTE_HOST "sudo systemctl restart nginx"

echo "Deployment completed!"
```

4. Automate Deployment on Git Push

Modify the post-receive hook on the remote server:

mkdir -p ~/deploy-hooks

nano ~/deploy-hooks/post-receive

then add, to the powershell

#!/bin/bash

GIT_WORK_TREE=/var/www/html/your-site git checkout -f

git remote add production user@your-server:/path/to/repo.git

git push production main

5. Add a Git Hook for Auto Deployment

In your local project, create a Git hook (.git/hooks/post-commit) and make it executable:

#!/bin/bash

./deploy.sh

chmod +x .git/hooks/post-commit

6. Test the Setup

Make a change to your local files, commit, and push:

sh

CopyEdit

```
git add .

git commit -m “Updated website”

git push
```

Check your server to confirm that the changes were deployed.

Outcome:

With this setup, your static website will update automatically whenever you push changes to your repository .