**Summary of the project:**

**Project Name**: "Automated Deployment of a Static Website on AWS"

**Objective**: Set up a basic DevOps pipeline to automatically deploy a static website on AWS using popular DevOps tools. This project will introduce you to key AWS services, CI/CD concepts, and basic website deployment practices. It's a great starting point for beginners to gain hands-on experience with DevOps in the AWS cloud environment. As you become more comfortable with these tools and practices, you can explore more advanced DevOps projects and expand your AWS knowledge.

**Tools and Technologies to Use**:

Git for version control

GitHub (or GitLab, Bitbucket, etc.) for code hosting

AWS services: S3 (Simple Storage Service) for hosting the website, and Code Pipeline/Code Build for CI/CD.

AWS CLI (Command Line Interface) for interacting with AWS services.

**Project Steps:**

**Create a Simple Static Website:**

Create a basic static website using HTML, CSS, and possibly JavaScript.

Add your website code to a Git repository.

**Set Up AWS Account:**

Create an IAM (Identity and Access Management) user with programmatic access and give it appropriate permissions for the required AWS services.

**Configure AWS CLI:**

Install and configure the AWS CLI on your local machine.

Configure it with the credentials from the IAM user created in step 2.

**Create an S3 Bucket for Website Hosting:**

Use the AWS CLI to create an S3 bucket that will host your website files.

**Configure Static Website Hosting:**

Set up the S3 bucket to host a static website.

Configure the bucket to allow public access to the website files.

**Create a CI/CD Pipeline:**

Set up a Code Pipeline in AWS.

Create a Code Build project to build and package your website code.

Configure the pipeline to trigger when changes are pushed to your Git repository.

**Automate Deployment:**

Configure Code Pipeline to deploy the website to the S3 bucket when the build is successful.

**Test and Monitor:**

Test the pipeline by making changes to your website code and pushing them to the repository.

Implement basic monitoring and logging using AWS services like CloudWatch.

**Documentation and Reporting:**

Create documentation that explains how to set up and use the CI/CD pipeline.

Consider creating a report summarizing your project and lessons learned.