

Task 1: Host a simple webpage on AWS

Prepared by Fatima Ezzahrae Ezzouzi

Cloud Programming (DLBSEPCP01_E)

Field of Study- Bachelor of Science in Applied Artificial Intelligence

Matriculation Number- 9217425

Tutor's Name- Ahmet Nasri

February 27, 202

Table of Contents

1. Abstract	1
2. Introduction	2
2.1 Project Overview	
2.2 Objectives	
3. Project Idea and Concept	3
3.1 Project Idea	
3.2 Concept and Design Principles	
4. Architecture Design	4
4.1 System Architecture Overview	
4.2 Architecture Components	
4.3 Architecture Workflow	
5. Technologies Used	5
5.1 Amazon Web Services (AWS)	
5.2 Terraform (Infrastructure as Code)	
5.3 Supporting Tools	
6. Terraform Project Structure	6
6.1 File Structure Overview	
6.2 main.tf	
6.3 variables.tf	
6.4 terraform.tfvars	
6.5 outputs.tf	
7. Deployment Process	7
7.1 Terraform Initialization	
7.2 Planning and Validation	
7.3 Infrastructure Provisioning	
8. Improvements from Phase 1 and Phase 2	8
9. Challenges and Solutions	9
10. Results and Final Evaluation	10
11. Conclusion	10

1. Introduction

1.1 Project Overview

This project presents the design and deployment of a static website hosted on Amazon Web Services (AWS) using Infrastructure as Code (IaC). The objective is to demonstrate how cloud infrastructure can be automated, secured, and optimized using Terraform.

The system integrates AWS Identity and Access Management (IAM), Amazon S3, and Amazon CloudFront to build a scalable and globally accessible architecture.

1.2 Objectives

- Automate infrastructure deployment using Terraform
 - Host a static website using Amazon S3
 - Improve global availability using CloudFront
 - Apply security best practices using IAM
 - Optimize infrastructure after feedback from Phase 1 and Phase 2
-

2. Project Idea and Concept

2.1 Idea

The idea behind this project is to replace manual cloud configuration with an automated, repeatable deployment approach using Infrastructure as Code.

Instead of creating AWS resources manually via the AWS Console, Terraform scripts define the entire infrastructure declaratively.

2.2 Concept

The concept is based on three main principles:

- Automation

- Scalability
- Security

The website content is stored in S3, distributed globally through CloudFront, and accessed securely using IAM-managed credentials.

3. Architecture Design

3.1 Architecture Components

The system consists of:

1. IAM (Identity and Access Management)
 - Secure programmatic access using access keys
 - Controlled permissions
2. Amazon S3
 - Static website hosting
 - Storage of HTML/CSS files
3. Amazon CloudFront
 - Content Delivery Network (CDN)
 - Global distribution
 - Reduced latency

3.2 Architecture Flow

User → CloudFront → S3 Bucket → Website Files

 Insert Architecture Diagram Screenshot Here

4. Technologies Used

4.1 Amazon Web Services (AWS)

- IAM
- S3
- CloudFront

4.2 Terraform

Terraform is used as an Infrastructure as Code tool to:

- Define cloud resources
- Automate provisioning
- Manage infrastructure state
- Ensure reproducibility

4.3 Supporting Tools

- AWS CLI
 - Terraform CLI
 - Web browser for testing
-

5. Terraform Project Structure

5.1 File Structure

project-folder/

```
|  
|--- main.tf  
|--- variables.tf  
|--- outputs.tf  
|--- terraform.tfvars
```

5.2 main.tf

Contains:

- Provider configuration
- IAM configuration
- S3 bucket creation
- CloudFront distribution

5.3 variables.tf

Defines:

- AWS region
- Bucket name
- Project name
- Origin ID

5.4 terraform.tfvars

Stores actual values for variables.

5.5 outputs.tf

Displays:

- S3 bucket name
- CloudFront domain name

 Insert Screenshot of Terraform Files Here

6. Deployment Process

The infrastructure was deployed using the following steps:

1. terraform init
2. terraform plan
3. terraform apply

Terraform automatically created:

- IAM credentials
- S3 bucket
- CloudFront distribution

 Insert Screenshot of Successful terraform apply Here

7. Improvements from Phase 1 & Phase 2

After receiving feedback, the following improvements were made:

- Refactored Terraform into modular structure
- Introduced variables instead of hardcoding values
- Improved IAM permission configuration
- Optimized CloudFront configuration
- Cleaned up code formatting
- Added outputs for better monitoring

These changes improved maintainability, scalability, and security.

8. Challenges Faced

8.1 IAM Configuration Issues

Initially, access key configuration caused authentication errors.

Solution: Verified credentials and corrected IAM policies.

8.2 S3 Public Access Configuration

Public access settings required careful adjustment for static hosting.

8.3 CloudFront Propagation Delay

CloudFront distribution required time to deploy globally.

8.4 Terraform Debugging

Syntax errors and dependency issues required troubleshooting.

9. Results and Final Evaluation

The final result is a fully operational static website deployed using Infrastructure as Code principles.

Key achievements:

- Fully automated deployment
- Secure IAM configuration
- Globally distributed content
- Scalable architecture
- Clean and modular Terraform structure

The infrastructure can be easily reused, modified, or scaled for future projects.

This project demonstrates practical understanding of cloud computing, DevOps practices, and Infrastructure as Code methodologies.

What You Still Need To Add

- ✓ Architecture diagram
- ✓ Screenshots of AWS Console
- ✓ Screenshot of Terraform apply
- ✓ Screenshot of Website running via CloudFront
- ✓ Your Abstract (at the beginning after title page)