Student Name: *Samyukhtha Haryaal*

Student ID: *35397098*

GitHub Repository Link: *https://github.com/Agathokakological713/cloud-server-project-ict171*

Live Server Link: *https://samportfolio.space/*

Video Explainer: *[Link to your explainer video]*

Public IP Address: *51.20.227.122*

# Project Overview

This repository documents the setup of a personal portfolio website, hosted on an AWS EC2 instance, configured with Apache, secured with SSL, and linked to a custom domain. The project involves the use of Infrastructure as a Service (IaaS), focusing on the configuration and management of web servers in the cloud, and the development of an automated setup script.

# Timeline

## Week 1: Initial Setup & Server Launch

* EC2 Instance Setup: I created an AWS EC2 instance using the Ubuntu image.
* Apache Installation: Installed Apache web server on the EC2 instance.
* Domain Purchase: Purchased a domain from Namecheap (samportfolio.space).

## Week 2: Apache Configuration & Security

* Apache Configuration: Configured Apache to serve the website on the domain.
* Firewall & UFW: Configured security settings to allow SSH and Apache traffic.
* SSL Setup: Installed SSL using Certbot to secure the site with HTTPS.

## Week 3: Domain Linking & Automation

* DNS Setup: Linked the domain to the EC2 instance via A record in Namecheap.
* Automation Script: Created a script to automate the setup process, including Apache, SSL, and firewall configuration.
* Testing: Ensured the server was fully operational by accessing the site via both HTTP and HTTPS.

## Week 4: Final Documentation & GitHub Submission

* Documentation: Wrote detailed documentation on the process and challenges faced.
* GitHub Upload: Uploaded all files (website code, automation script, and documentation) to GitHub.
* Video Explainer: Recorded a 1-2 minute video walkthrough explaining the process and demonstrating the live website.

# Challenges and Solutions

## Challenge 1: Apache Installation and Configuration

* Issue: Encountered issues with Apache not starting due to incorrect configuration.
* Solution: Reviewed Apache logs, identified the issue in the 000-default.conf file, and corrected the DocumentRoot and ServerName.

## Challenge 2: Domain Linking

* Issue: Initially faced issues with the DNS settings not propagating properly.
* Solution: Verified the A record in Namecheap and confirmed propagation using online tools like [whatsmydns.net](https://www.whatsmydns.net/).

## Challenge 3: SSL Setup (Certbot)

* Issue: Failed to install Certbot due to missing dependencies.
* Solution: Installed the necessary dependencies using sudo apt install software-properties-common and followed the Certbot instructions to resolve the issue.

## Challenge 4: Firewall Configuration (UFW)

* Issue: Accidentally blocked SSH access after enabling UFW.
* Solution: Allowed SSH connections explicitly by running sudo ufw allow OpenSSH before enabling the firewall.

# Process Overview

## Creating the EC2 Instance

* I started by launching an EC2 instance with Ubuntu as the operating system. This instance would host the Apache web server for my portfolio site.

## Installing Apache

* Apache was installed on the EC2 instance using sudo apt update followed by sudo apt install apache2. This allowed the server to serve my website.

## Configuring Apache

* I edited the Apache configuration file (000-default.conf) to point to my website's directory and set up the domain name (samportfolio.space).
* This involved setting the ServerName and DocumentRoot within Apache’s configuration.

## Securing the Server

* I enabled firewall rules to allow SSH and HTTP traffic using UFW. I also installed SSL certificates with Certbot for HTTPS.

## Linking Domain to Server

* After purchasing the domain from Namecheap, I updated the DNS settings to point to my EC2 instance’s public IP. I set up an A record to ensure the domain would resolve to the correct IP.

## Automation Script

* I created a shell script (setup.sh) that automates the process of setting up Apache, installing SSL certificates, and configuring the firewall. This script simplifies future deployments or re-installations.