AUGUST 16, 2025

STATIC PORTFOLIO WEBSITE WITH SSL CERTS

A STEP-BY-STEP GUIDE TO HOSTING A SECURE STATIC WEBSITE ON AWS

TEBOGO MATSEDING
tmatseding@outlook.com

Table of Contents

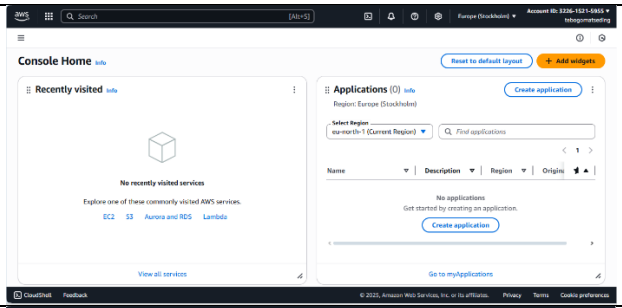
START

Introduction

This project walks through the process of setting up a static portfolio website using Amazon S3, Route 53, ACM (Amazon Certificate Manager), and CloudFront. The goal is to host a secure website with SSL certificates, custom domain integration, and proper routing through AWS services.

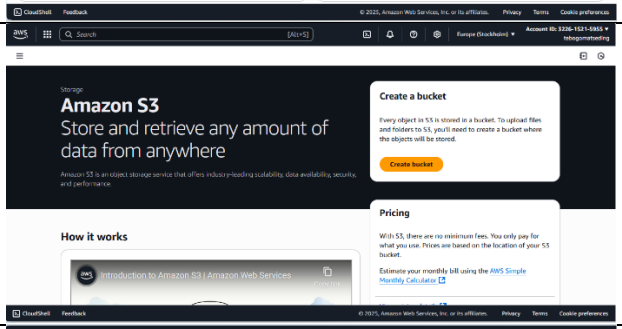
Console Page

We start by signing in to the AWS Management Console. This is the main dashboard where all AWS services are accessed. Every step of the project begins here.



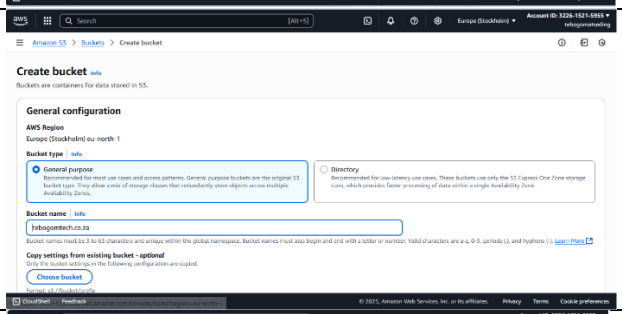
Create S3 Bucket

In S3, we create a bucket that will store our website files. An S3 bucket acts like a cloud folder for hosting static content.



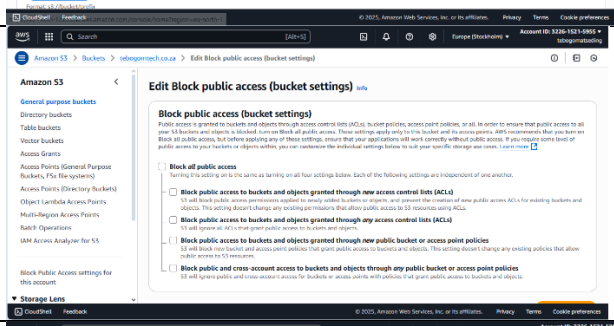
Name S3 Bucket

The bucket name must exactly match your domain name (e.g., tebogomtech.co.za). This makes integration with Route 53 and CloudFront easier later.



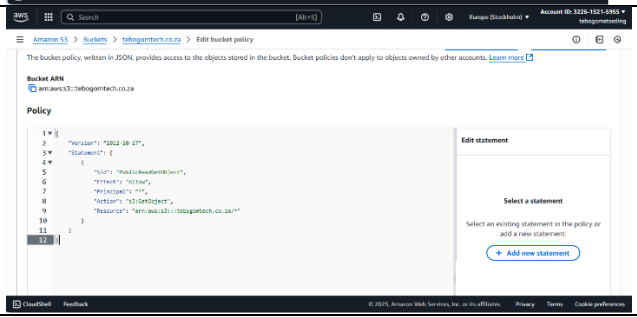
Disable Block Public Access

By default, S3 blocks public access. Since our site needs to be publicly accessible, we disable this setting for the bucket.



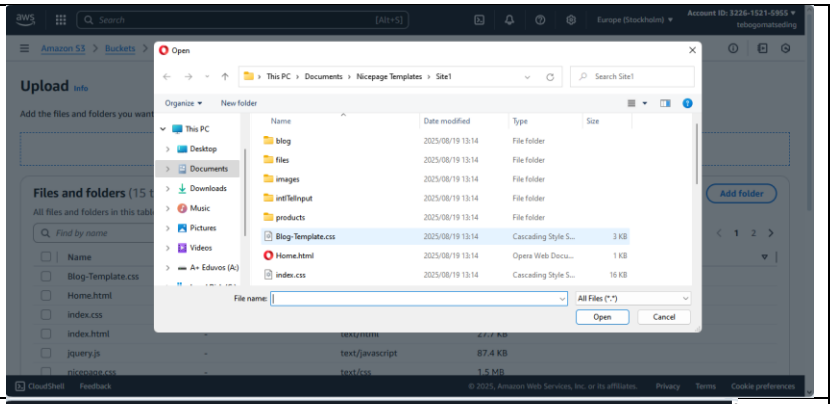
S3 Bucket Policy

We apply a bucket policy that explicitly allows public read access to the files. This ensures that visitors can load images, HTML, CSS, and JavaScript.



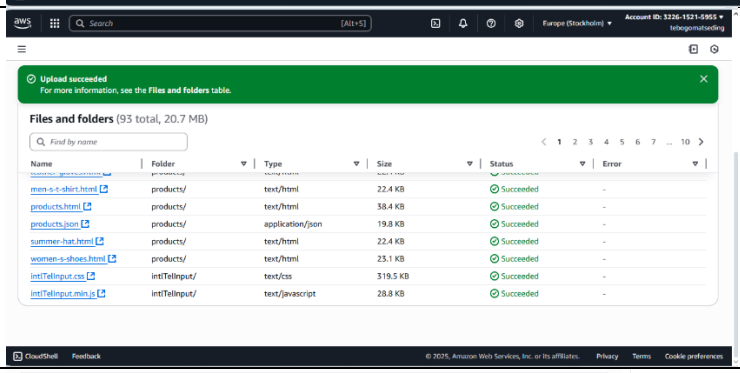
Upload Website Files

Next, we upload the actual website files (HTML, CSS, JS, images) into the bucket. These files form the structure and design of the portfolio.



Upload Complete

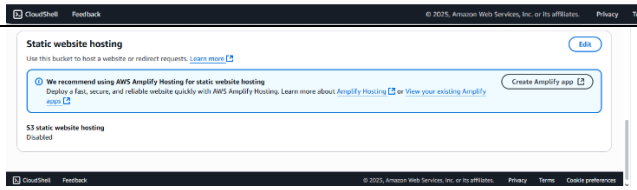
Once the upload is done, we verify that all required files are present in the bucket. At this stage, the files exist in AWS but aren't yet accessible via the internet.



Enable Static Web Hosting

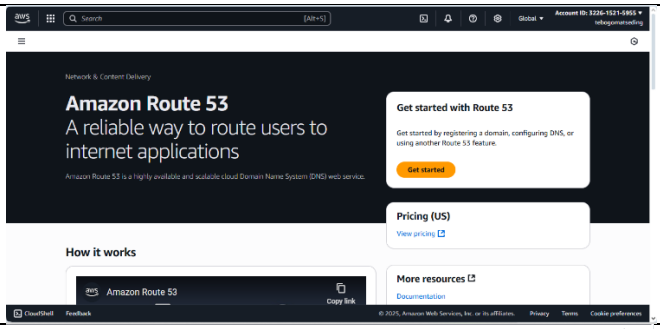
(On S3 Bucket)

We enable static website hosting in S3 and specify the index.html as the entry point. This allows the bucket to serve files like a website.



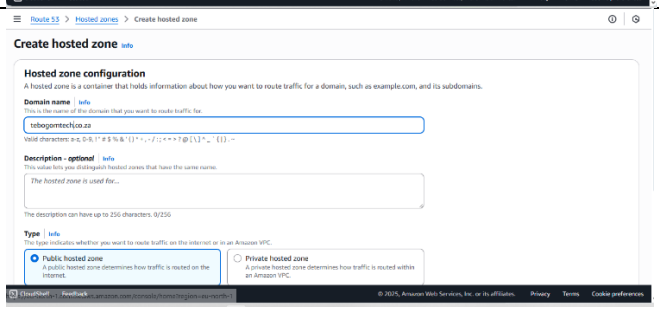
Navigate to Route 53

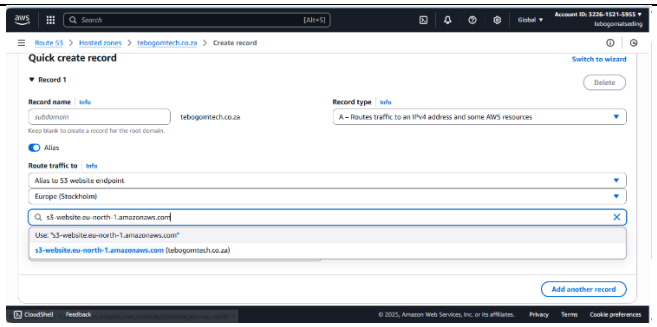
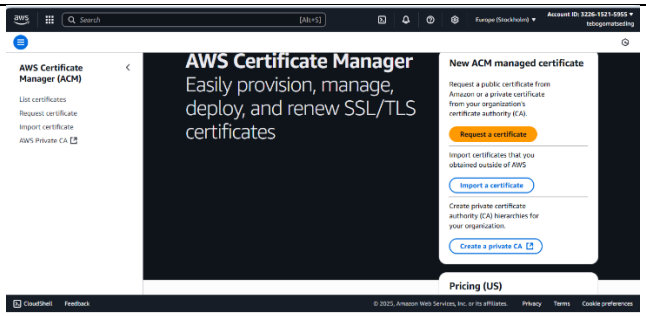
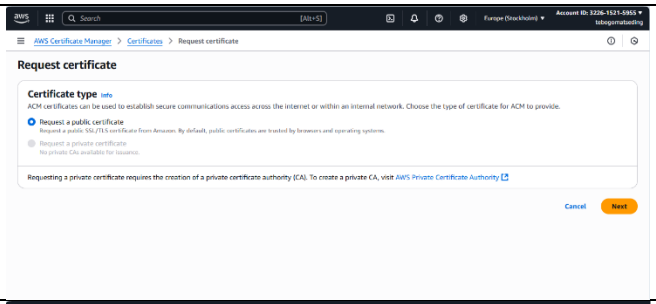
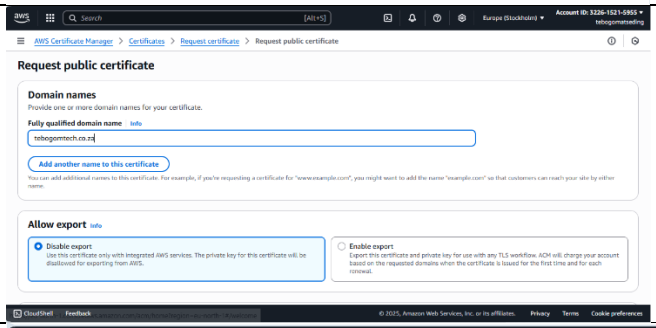
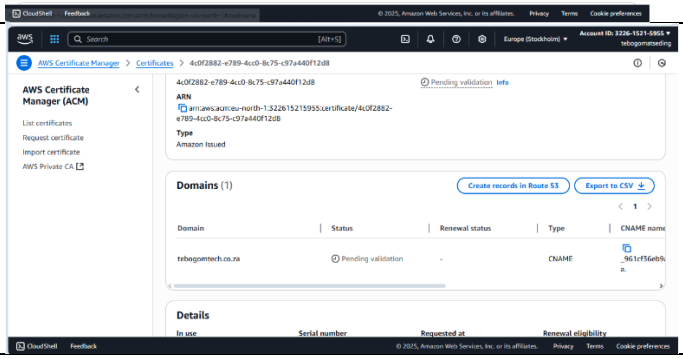
To connect the domain to our S3 bucket, we move to Route 53, AWS's DNS management service.



Create new hosted zone

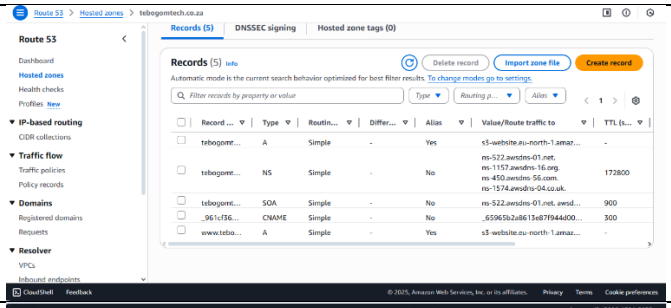
We create a hosted zone for our domain (e.g., tebogomtech.co.za). This zone will store DNS records that route traffic to AWS resources.



Create a record for your Alias (where an endpoint directs traffic to the S3 Bucket) We add an Alias Record pointing the domain name to the S3 bucket endpoint. This ensures that typing your domain name directs traffic to your hosted site.	
Navigate to Amazon Certificate Manager (ACM) To secure the site with HTTPS, we need an SSL certificate. We open ACM, which manages SSL/TLS certificates in AWS.	
Request a public certificate We request a new certificate and select “public” since this website is accessible on the internet.	
Fully qualified domain name We enter the full domain name (e.g., tebogomtech.co.za) and any subdomains (like www.tebogomtech.co.za) if required.	
Validate Certificate ACM requires domain ownership verification. AWS provides a CNAME record that must be added to Route 53.	

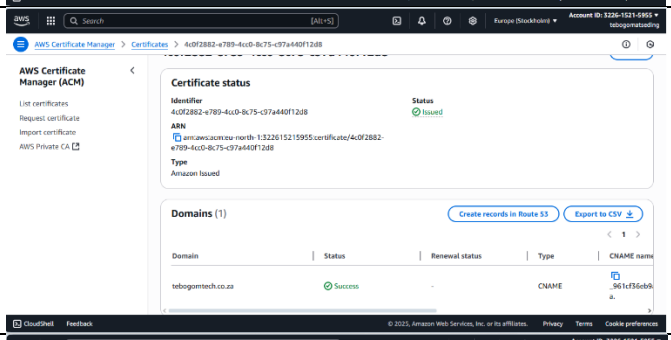
Verify CNAME was created in Route 53

We return to Route 53 and confirm that the CNAME record has been added. This proves ownership of the domain to ACM.



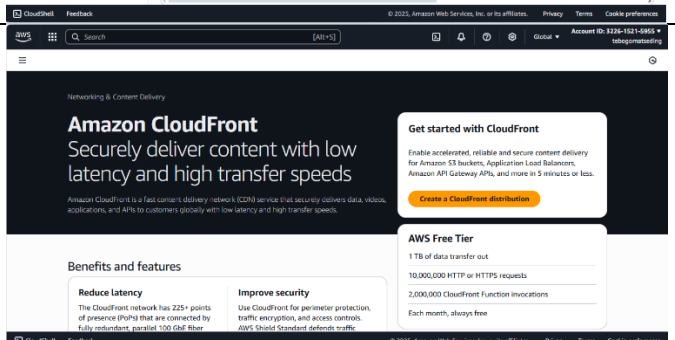
SSL Issued

After validation, ACM issues the SSL certificate. This is what allows the website to be served over HTTPS.



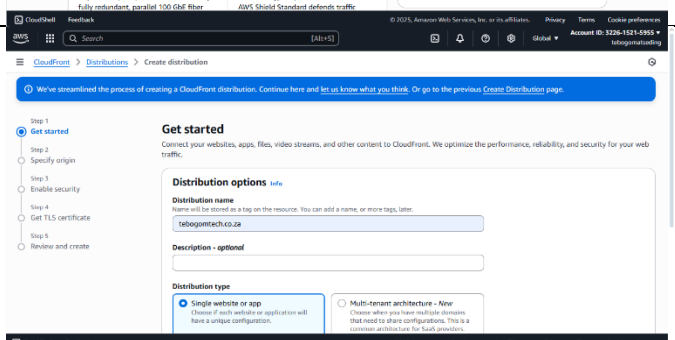
Navigate to CloudFront

Next, we move to CloudFront, AWS's CDN (Content Delivery Network), to distribute the website globally with low latency.



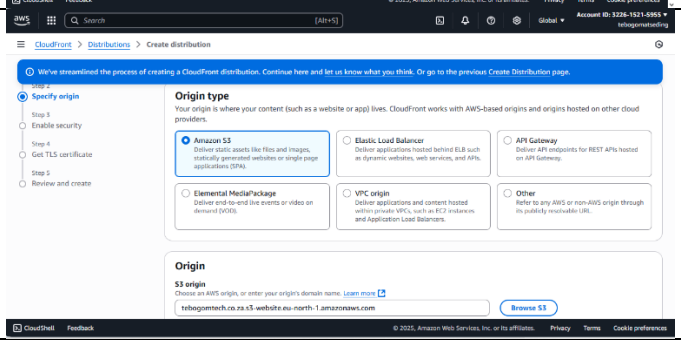
Create CloudFront Distribution

We create a new distribution, which acts as a content delivery layer in front of our S3 bucket.



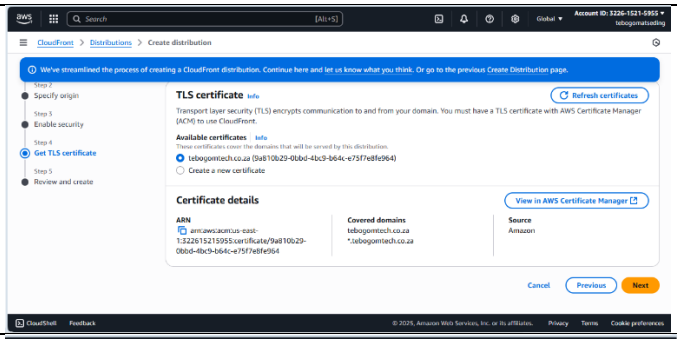
Specify Origin

Here, we specify the S3 bucket as the origin source for the CloudFront distribution. This ensures CloudFront knows where to pull website files from.



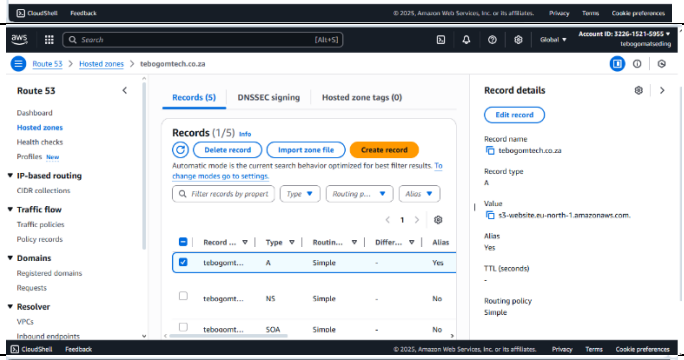
Get TLS certificate

We attach the SSL certificate from ACM to the CloudFront distribution. This enables HTTPS for secure traffic.



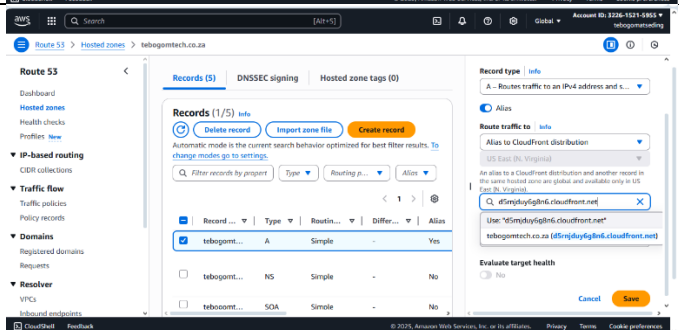
Edit Record in Route 53

We go back to Route 53 and edit the DNS records to point the domain to the CloudFront distribution instead of the S3 endpoint.



Route Traffic to CloudFront Distribution

Once updated, all traffic to the domain will be routed through CloudFront, ensuring faster loading speeds and encrypted connections.



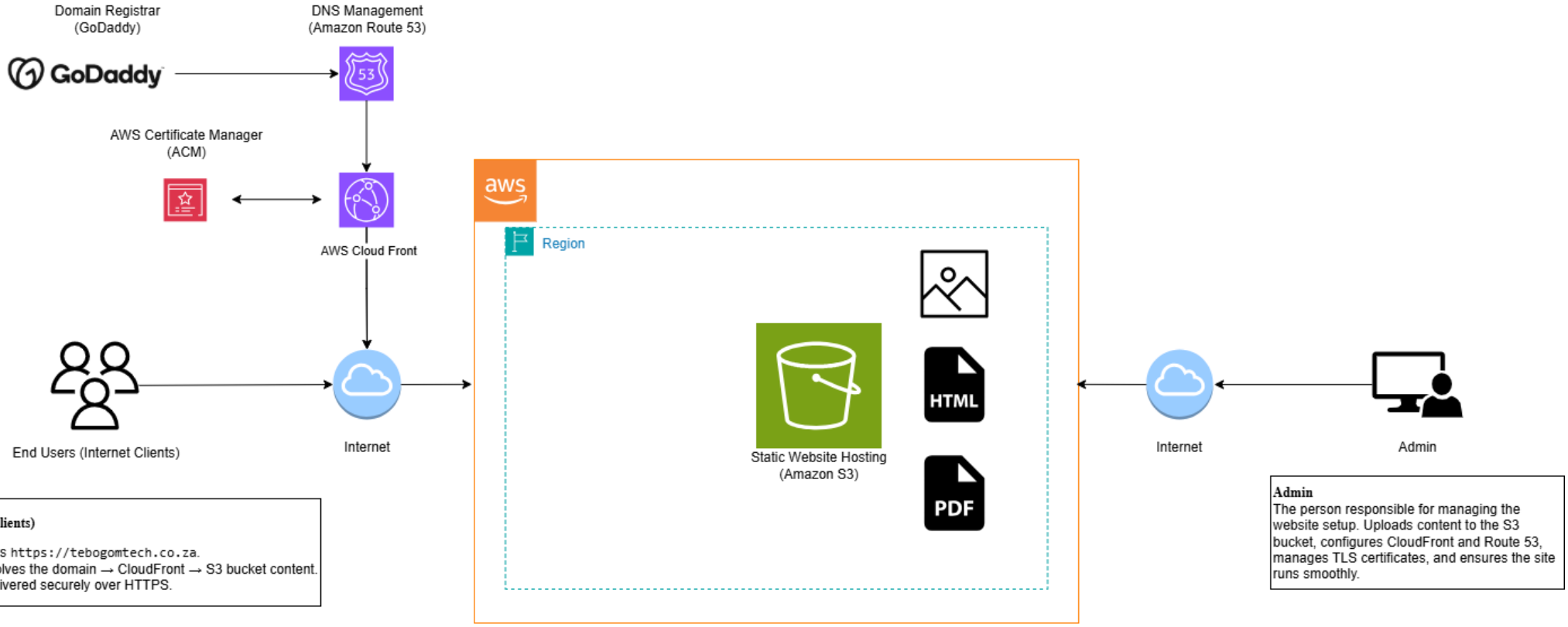
Website is up & Secure

At this stage, the website is fully functional, served through CloudFront, and secured with SSL certificates. Visitors can now access it using https://yourdomain.com.

Link to my portfolio: tebogomtech.co.za

Project Illustration

A simple diagram is created to show how all the components (S3 → Route 53 → ACM → CloudFront → User) connect together.



End Users (Internet Clients)

- Visitors access `https://tebogomtech.co.za`.
- Route 53 resolves the domain → CloudFront → S3 bucket content.
- Content is delivered securely over HTTPS.

Admin
The person responsible for managing the website setup. Uploads content to the S3 bucket, configures CloudFront and Route 53, manages TLS certificates, and ensures the site runs smoothly.

Domain Registrar (GoDaddy)

- Domain `tebogomtech.co.za` is registered here.
- Nameservers updated to use Route 53.

DNS Management (Amazon Route 53)

- Hosted zone created for `tebogomtech.co.za`.
- DNS records (A/AAAA alias) point the domain to the CloudFront distribution.

Content Delivery & Security (Amazon CloudFront)

- CloudFront distribution sits in front of the S3 bucket.
- Provides caching, global performance, and HTTPS delivery.
- Integrated with **AWS Certificate Manager (ACM)** for TLS/SSL certificate.

Static Website Hosting (Amazon S3)

- Primary bucket: `tebogomtech.co.za`.
- Stores and serves static website files (HTML, CSS, JS, media).
- Configured as the **origin** for CloudFront.

Next Steps & Enhancements

Now that the static website is live and secured, there are several ways to refine and strengthen the setup:

Domain Redirection

Create an additional S3 bucket to handle domain redirects (e.g., redirect www.domain.com → domain.com). This ensures visitors always land on the correct version of your site.

Monitoring & Logging

Turn on access logs in CloudFront and S3 to track visitor activity and troubleshoot errors. Integrate with AWS CloudWatch for monitoring and alerting.

Security Hardening

Add AWS WAF (Web Application Firewall) to the CloudFront distribution to filter malicious requests. Enable AWS Shield (standard is free) for protection against DDoS attacks.

Custom Error Pages

Add user-friendly error pages (404, 500, etc.) in S3 for a polished experience when something goes wrong.

Scalability & Expansion

Extend the setup to host multiple sites or subdomains under the same CloudFront distribution. This can be useful if you plan to add a blog, documentation site, or client projects.