Project Report: Domain Hosting & Routing with Route 53, ALB, and ACM

Project Title

Secure Domain Hosting with Host + Path-Based Routing Using Route 53, ALB & ACM

Objective

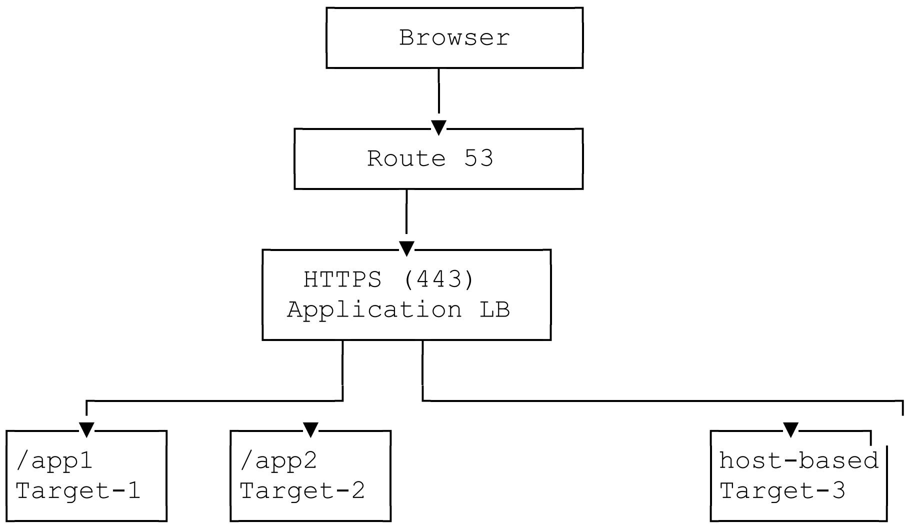
To host a custom domain using AWS Route 53 and configure an Application Load Balancer (ALB) to route traffic based on:

- **Host-based rules** (e.g., `app1.example.com`, `app2.example.com`)
- **Path-based rules** (e.g., `example.com/app1`, `example.com/app2`)
  Additionally, enforce HTTPS using a free **SSL certificate via ACM** and automatically redirect HTTP to HTTPS.

Key Concepts & Services Used

- **Route 53 (Domain Hosting & DNS)**
- **ACM (AWS Certificate Manager)**
- **Application Load Balancer (ALB)**
- **Target Groups**
- **Listener Rules (Host + Path Based)**
- **NGINX (on EC2 Instances)**
- **HTTP to HTTPS Redirection**
- **Security Groups**
- **Public Hosted Zone**

Architecture Overview



Steps Implemented

1. Domain Setup with Route 53

- Purchased domain (or imported external domain)
- Created **Hosted Zone** in Route 53
- Added **A record (Alias)** pointing to ALB DNS

2. SSL Certificate with ACM

- Requested public SSL cert for:
  - `example.com`
  - `*.example.com`
- Validated via **DNS method** (automatically updates in Route 53)

3. Application Load Balancer

- Internet-facing ALB in 2 public subnets
- Listener on **port 80 (HTTP)** and **443 (HTTPS)**
- Attached ACM SSL cert to HTTPS listener

4. Listener Rules

- **Host-based Routing**:
  - `app1.example.com` → Target Group 1
  - `app2.example.com` → Target Group 2
- **Path-based Routing**:
  - `/app1` → Target Group 1
  - `/app2` → Target Group 2
- **Default Action**: 404 or redirect to `/home`

5. HTTP to HTTPS Redirection

- HTTP Listener (Port 80):
    - Rule: Redirect to HTTPS (Status code: 301)
    - No backend traffic handled at HTTP layer

6. Target Groups

- Created two target groups with health checks
- Registered EC2 instances running NGINX for each app

7. EC2 Configuration (NGINX)

- App1: Custom content for app1, hosted on EC2
- App2: Custom content for app2, hosted on EC2

Testing

- Visit `http://example.com/app1` → Redirects to HTTPS → App1 content
- Visit `https://app2.example.com` → App2 content
- Visiting unconfigured routes → custom 404 or fallback