Anime Review Website – Cloud Deployment Report

Student Name: Roshan shrestha

Student ID: 35318397

Unit: ICT171 – Introduction to Server Environments and Architectures

Project: Anime Review Website

Public IP Address: 35.172.133.63

Domain Name: https://daifu.click

Video explainer: https://youtu.be/hosN_OOKvu8

Github link: UNKN0WN-deblog/anime-review-wesite

Year: 2025

1. Introduction

This document outlines the setup and deployment process for a cloud-hosted Anime Review Website using AWS EC2. The project demonstrates essential skills in Infrastructure as a Service (IaaS), web server configuration, secure file transfer, domain linking, and SSL certification. The website offers a static experience displaying spoiler-free anime reviews.

2. Objectives

- Set up a virtual server using EC2
- Install and configure Apache2 to host a static website
- Transfer files securely via SCP
- Point a custom domain to the server's IP
- Implement HTTPS using Certbot and Let's Encrypt
- Monitor server uptime with a Bash script
- Version control using GitHub

3. Technologies Used

- Amazon EC2 (Ubuntu 22.04 LTS)
- Apache2 Web Server
- SCP / SSH
- Route 53 DNS
- Certbot SSL
- HTML5, CSS3
- Git + GitHub

4. Website Creation Process

The website was built with HTML and CSS. Key steps included:

- Designing a responsive layout in `index.html`
- Styling with `styles.css`
- Placing logos and thumbnails in the `/images` folder
- Writing spoiler-free summaries of popular anime
- Testing locally in browser before deployment

5. EC2 and Web Server Setup

Launch and configure an EC2 instance:

```bash
sudo apt update
sudo apt install apache2 -y
sudo systemctl status apache2
```

6. Upload Website to Server

Use SCP to upload files to EC2:

```powershell
scp -i "Anime.pem" anime-review-site.zip ubuntu@35.172.133.63:~
```

Then SSH into EC2 and deploy:

```bash
ssh -i "C:\your path\keypair.pem" ubuntu@ip address

unzip anime-review-site.zip -d anime-site
sudo cp anime-site/* /var/www/html/
sudo chown -R www-data:www-data /var/www/html
```

sudo chmod -R 755 /var/www/html
```

## 7. Domain and DNS Configuration

Using Route 53, A records were added for `@` and `www` to point to the EC2 public IP (35.172.133.63). Propagation was verified using dnschecker.org.

## 8. SSL Setup with Certbot

Install SSL with the following commands:

```bash
sudo apt install certbot python3-certbot-apache -y
sudo certbot --apache -d daifu.click -d www.daifu.click
```

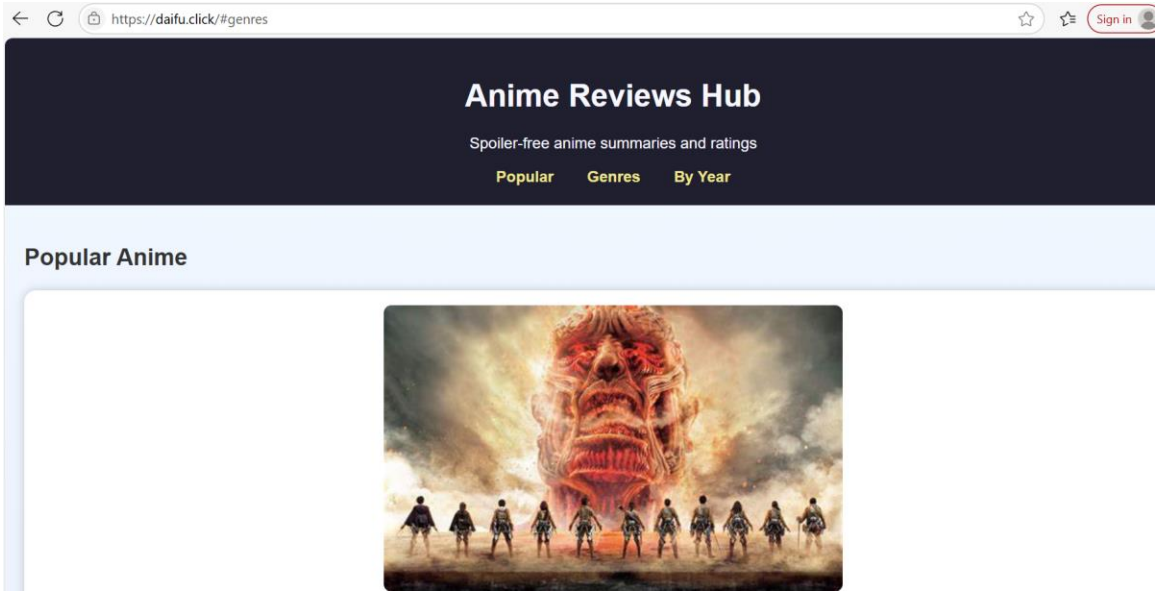
10. GitHub Integration

Project was tracked and uploaded to GitHub:

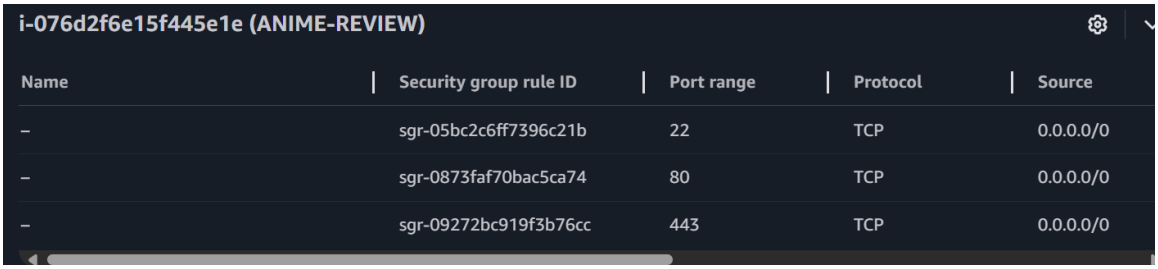
```bash
git init
git add .
git commit -m "Initial commit"
git remote add origin UNKN0WN-deblog/anime-review-wesite
git push -u origin main
```

11. Screenshots

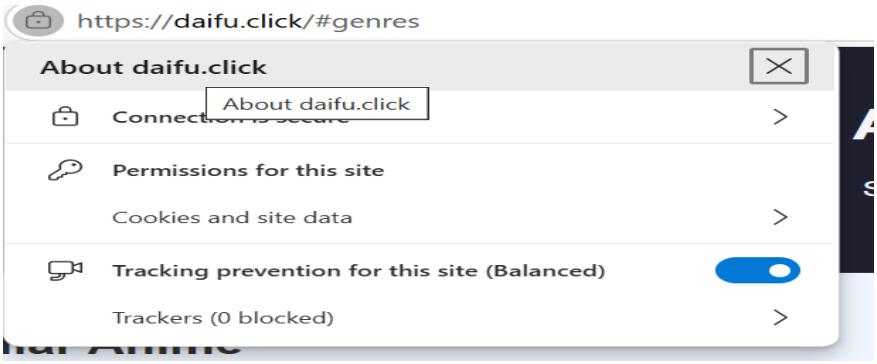
[Insert screenshots of EC2, Route 53, SCP, Apache status, browser output, etc.]
```



• Security group setting (ports 22, 80, 443)



• SSL Padlock visible on HTTPS



• Apache2 status

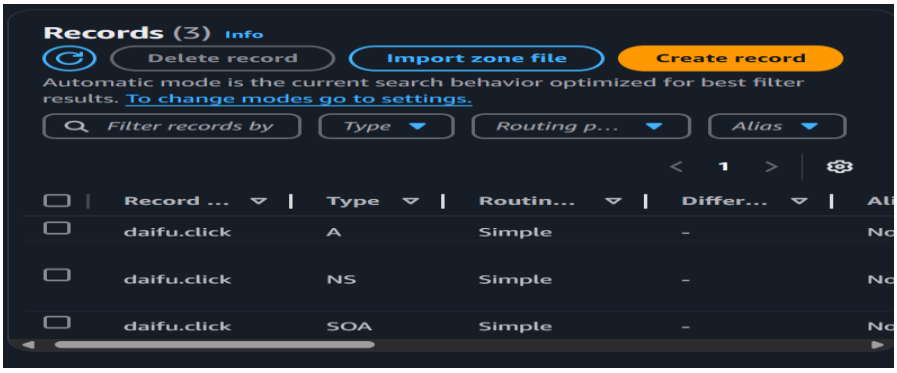
```
ubuntu@ip-172-31-95-141:~$ sudo systemctl status apache2
Warning: The unit file, source configuration file or drop-ins of apache2.service changed on disk. Run 'systemctl daemon-reload' to r
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-06-04 08:03:53 UTC; 15h ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2718 (apache2)
      Tasks: 55 (limit: 1129)
     Memory: 10.4M (peak: 10.8M)
        CPU: 3.311s
     CGroup: /system.slice/apache2.service
             ├─2718 /usr/sbin/apache2 -k start
             ├─5605 /usr/sbin/apache2 -k start
             └─5633 /usr/sbin/apache2 -k start

Jun 04 08:03:53 ip-172-31-95-141 systemd[1]: Starting apache2.service - The Apache HTTP Server...
```

• SCP terminal upload

```
PS C:\Users\My Computer> scp -i "C:\Users\My Computer\Downloads\Anime.pem" "C:\Users\My Computer\Downloads\AOT.jpg" ubuntu@35.172.133
.63:/var/www/html/images/
```

• DNS settings in Route 53



12. Conclusion

This project successfully demonstrates static website deployment in the cloud using AWS EC2. By combining essential server setup skills, DNS configuration, secure protocols, and basic scripting, it aligns with the learning outcomes of ICT171 and provides a strong foundation for future dynamic web development.

13. References

- Apache: https://httpd.apache.org/
- Certbot: https://certbot.eff.org/
- Creative Commons: https://creativecommons.org/licenses/
- AWS EC2: https://docs.aws.amazon.com/ec2/