Project Report: Hosting a Website on AWS EC2

Author: K. Mayuk Reddy

Date: July 2025

1. Objective

Introducing the foundational concepts of cloud deployment, Linux server management, and web hosting by deploying a basic static website using Amazon EC2 and Apache2. By the end, the user should understand:

- Provisioning and configuring an EC2 instance.
- Using SSH for secure cloud server access.
- Installing and managing Apache2 web server.
- Serving custom HTML content from a Linux-based cloud VM.

2. Tools & Technologies Used

Tool / Service	Description
Amazon EC2	Elastic Compute Cloud - virtual machine hosting platform.
Ubuntu 22.04 LTS	Lightweight and secure Linux OS for servers.
Apache2	Widely-used open-source web server.
Git Bash	Terminal emulator for Windows with Unix command support.
SSH	Secure communication protocol for remote access.
Web Browser	Used to access and verify the hosted web page.

3. Implementation Workflow

Step 1: Launching an EC2 Instance

Choosing Ubuntu Server 22.04 LTS as the base image, t2.micro instance type within AWS Free Tier, enabling Auto-assign Public IP, and creating a key pair named ec2-key.pem for secure SSH login. Restricting key file permissions to read-only using chmod 400.

Step 2: Security Group Configuration

Configuring firewall rules:

Protocol	Port	Source	Purpose
SSH	22	0.0.0.0/0	Remote terminal access
HTTP	80	0.0.0.0/0	Web browser access to website

Note: 0.0.0.0/0 is suitable for testing but not for production.

Step 3: SSH into EC2 Instance

Connecting using Git Bash:

```
chmod 400 ec2-key.pem
ssh -i ec2-key.pem ubuntu@<your-ec2-public-ip>
```

Step 4: Installing Apache2

Updating and installing Apache:

```
sudo apt update
sudo apt install apache2 -y
```

Verifying with:

```
systemctl status apache2
```

Step 5: Deploying Custom HTML Content

Replacing default Apache page:

```
echo '<h1>Hello from Siva Sai'''''s EC2 Website!</h1>' | sudo tee /var/www/html/index
```

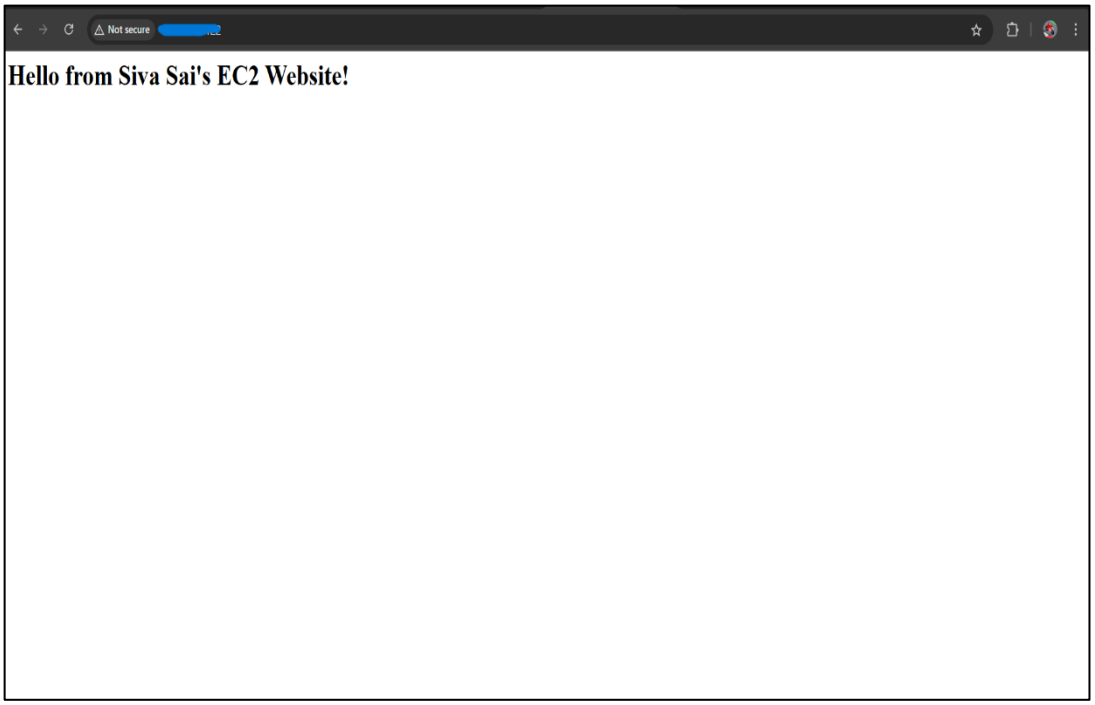
Step 6: Accessing the Website

Accessing via browser at http://<your-ec2-public-ip>.

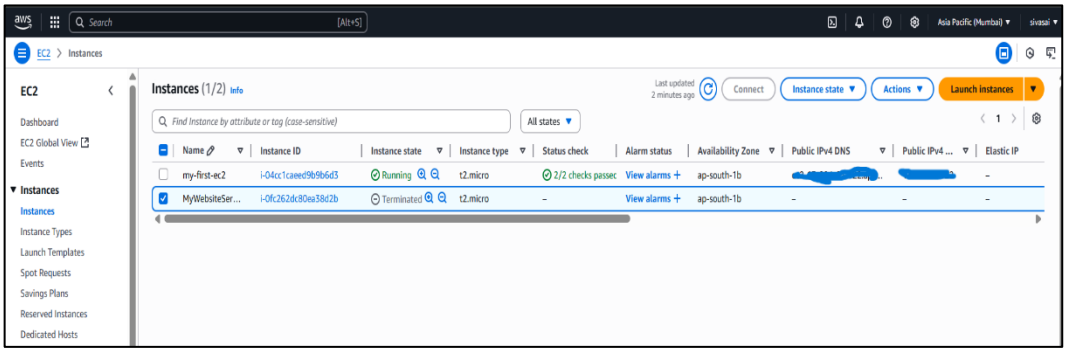
4. Screenshots to Include

Including visual support:

- Web Browser Screenshot – Showing the hosted site.



- AWS EC2 Dashboard – Instance running with public IP.



- Security Group Rules – Inbound rules for SSH and HTTP.

Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
-	sgr-0541ddd0791288ab0	22	TCP	0.0.0.0/0	launch-wizard-2	-
-	sgr-088b8c514e402823e	80	TCP	0.0.0.0/0	launch-wizard-2	-

5. Insights and Learning Outcomes

- Learned AWS Management Console usage.
- Understood networking concepts like IP addresses and ports.
- Gained Linux terminal and software installation experience.
- Deployed a live website using free-tier resources.

6. Challenges Faced

- Public IP changes on instance stop/start (fix: use Elastic IP).
- Local firewall blocking SSH/HTTP.
- Apache conflicts (check /var/log/apache2/error.log).

7. Future Improvements

Improvement	Description
HTTPS with SSL	Add Let's Encrypt certificate.
Custom Domain	Link own domain using Route53.
Auto Deployment	Use scripts or Ansible.
Use S3 or LightSail	For lower cost and higher availability.
CI/CD Integration	Connect to GitHub for auto-updates.

8. Conclusion

This project demonstrates deploying a functional website on AWS EC2 with minimal resources, forming the groundwork for full-stack deployments, scalable architectures, and DevOps integration.