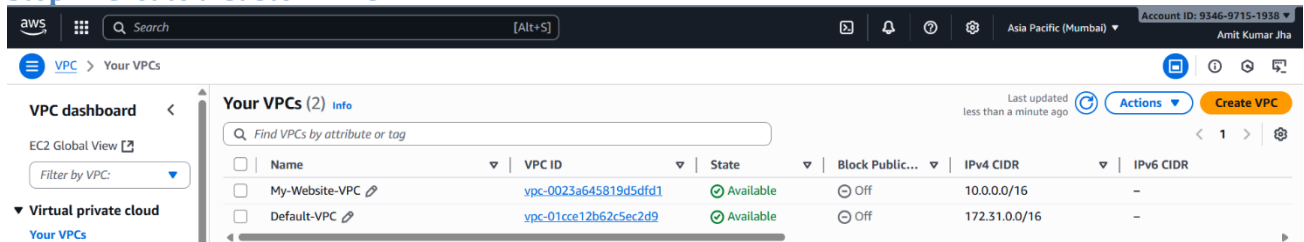


Website Deployment on AWS using Custom VPC and Monitoring

Project Overview

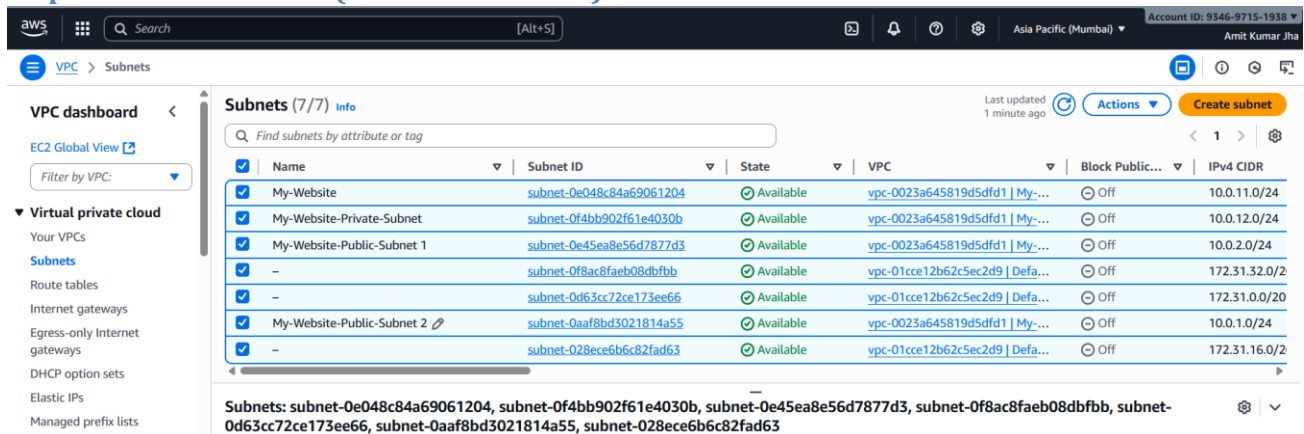
This project demonstrates website hosting on AWS by creating a custom VPC with Auto Scaling, Application Load Balancer, and monitoring using CloudWatch. It also integrates Simple Notification Service (SNS) for alerts and notifications, ensuring continuous availability, scalability, and operational visibility.

Step 1: Create a Custom VPC



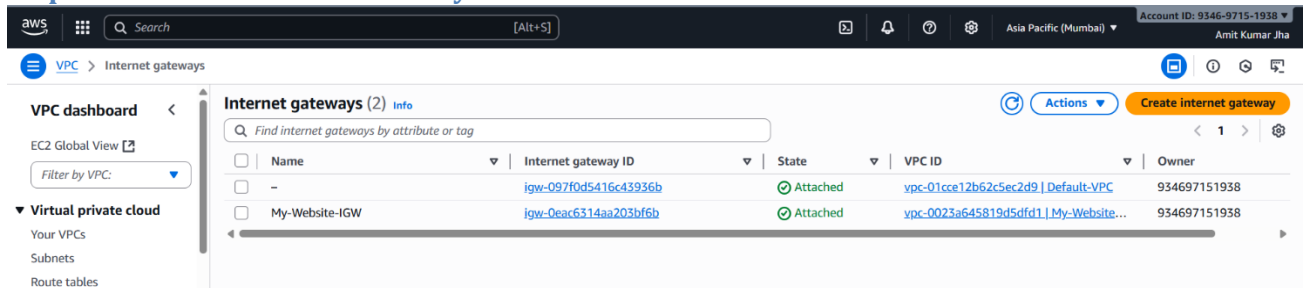
Screenshot 1: Custom VPC Creation

Step 2: Create 2 Subnets (Public and Private)



Screenshot 2: Subnet Configuration

Step 3: Create an Internet Gateway



Screenshot 3: Internet Gateway Setup

By- Amit Kumar Jha

Step 4: Create a NAT Gateway

The screenshot shows the AWS Management Console for the 'VPC' service, specifically the 'NAT gateways' page. The left sidebar shows the 'VPC dashboard' with options for 'EC2 Global View' and 'Filter by VPC'. The main content area shows a table of NAT gateways. One gateway, 'my-website-nat', is listed with ID 'nat-0013f467d83fbafdd', connectivity type 'Public', state 'Available', and primary public and private IP addresses '13.202.109.51' and '10.0.1.35' respectively. A 'Create NAT gateway' button is visible in the top right.

Name	NAT gateway ID	Connectivity...	State	State message	Primary public I...	Primary private I...
my-website-nat	nat-0013f467d83fbafdd	Public	Available	-	13.202.109.51	10.0.1.35

Screenshot 4: NAT Gateway Setup

Step 5: Create Route Tables (Public & Private)

The screenshot shows the AWS Management Console for the 'VPC' service, specifically the 'Route tables' page. The left sidebar shows the 'VPC dashboard' with options for 'EC2 Global View' and 'Filter by VPC'. The main content area shows a table of route tables. Four route tables are listed: 'my-website-rt-public', 'my-website-rt-private', and two unnamed tables. Each table has a 'Route table ID', 'Explicit subnet associ...', 'Edge associations', 'Main' status, and 'VPC' ID. The 'my-website-rt-public' and 'my-website-rt-private' tables are associated with VPC 'vpc-0023a645819d5dfd1'. The unnamed tables are associated with VPC 'vpc-01cce12b62c5ec2d9' and 'vpc-0023a645819d5dfd1' respectively.

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
my-website-rt-public	rtb-00944e23c02cfb5e1	2 subnets	-	No	vpc-0023a645819d5dfd1 My...
my-website-rt-private	rtb-0142c5ca21a43b17a	2 subnets	-	No	vpc-0023a645819d5dfd1 My...
-	rtb-02a9d85efd87cf8de	-	-	Yes	vpc-01cce12b62c5ec2d9 Defa...
-	rtb-0e156f314b479968d	-	-	Yes	vpc-0023a645819d5dfd1 My...

Screenshot 5: Route Tables Setup

Step 6: Create Security Group (Ports 22, 80)

The screenshot shows the AWS Management Console for the 'Security Groups' service, specifically the 'sg-0931f63c9ad0d89f2 - my-website-ec2-sg' page. The left sidebar shows the 'VPC' service with options for 'gateways', 'DHCP option sets', 'Elastic IPs', 'Managed prefix lists', 'NAT gateways', 'Peering connections', 'Security', 'PrivateLink and Lattice', 'Getting started', 'Endpoints', 'Endpoint services', 'Service networks', 'Lattice services', 'Resource configurations', and 'Resource gateways'. The main content area shows the details of the security group, including its name, ID, description, owner, inbound rules count, and outbound rules count. The 'Inbound rules' tab is selected, showing a table of inbound rules. Two rules are listed: one for HTTP (port 80) and one for SSH (port 22).

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-02f6161d24ce09194	-	HTTP	TCP	80
-	sgr-0f34913694eae964	IPv4	SSH	TCP	22

Screenshot 6: Security Group Rules

Step 7: Create an SNS Topic

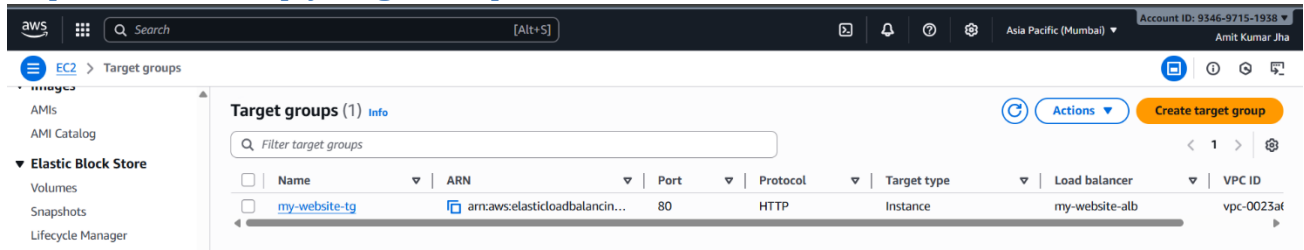
The screenshot shows the AWS Management Console for the 'Amazon SNS' service, specifically the 'Topics' page. The left sidebar shows the 'Amazon SNS' service with options for 'Dashboard', 'Topics', 'Subscriptions', 'Mobile', 'Push notifications', and 'Text messaging (SMS)'. The main content area shows the details of a new topic, 'my-website-alerts', with subscription '9a444778-77aa-4f7a-b46e-bc20b0f140dc'. The 'Details' tab is selected, showing the topic's ARN, endpoint, topic name, and subscription principal. The 'Status' is 'Confirmed' and the 'Protocol' is 'EMAIL'.

ARN	Status	Protocol
arn:aws:sns:ap-south-1:934697151938:my-website-alerts:9a444778-77aa-4f7a-b46e-bc20b0f140dc	Confirmed	EMAIL

Screenshot 7: SNS Topic Creation and Subscription

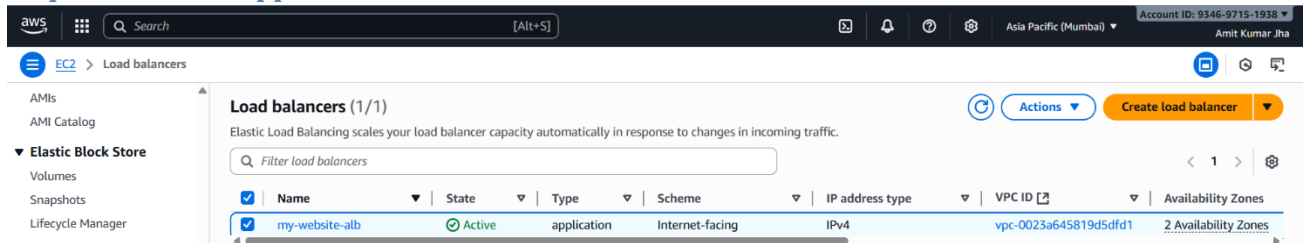
By- Amit Kumar Jha

Step 8: Create an Empty Target Group



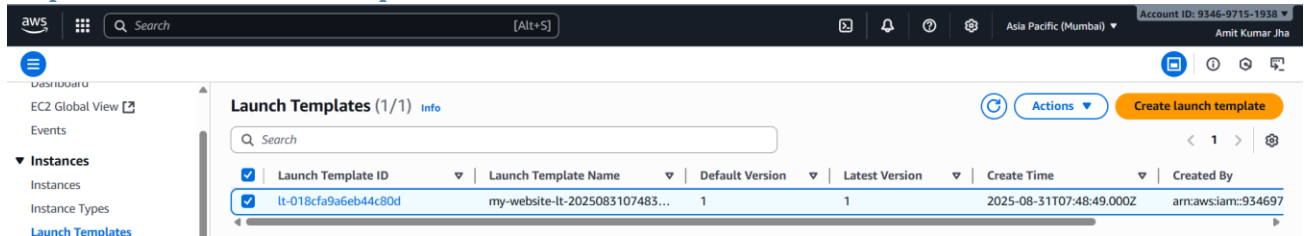
Screenshot 8: Target Group Setup

Step 9: Create an Application Load Balancer



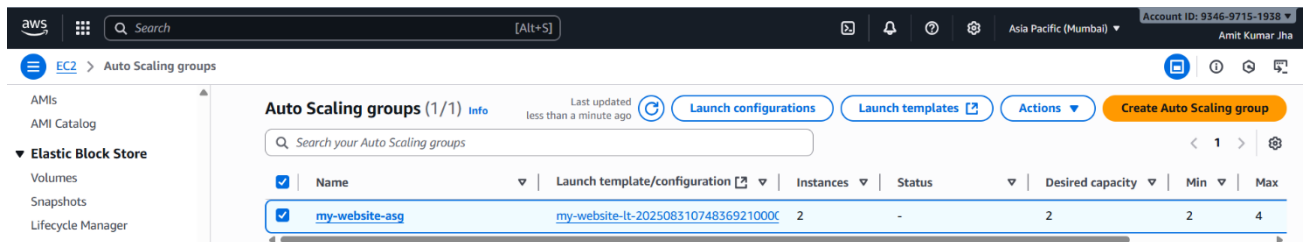
Screenshot 9: ALB Creation

Step 10: Create Launch Template



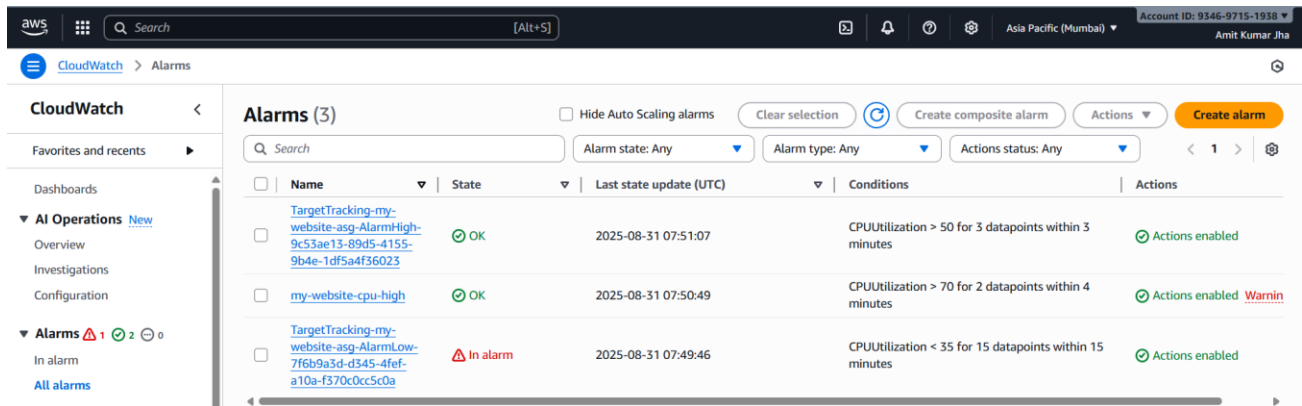
Screenshot 10: Launch Template Setup

Step 11: Create an Auto Scaling Group



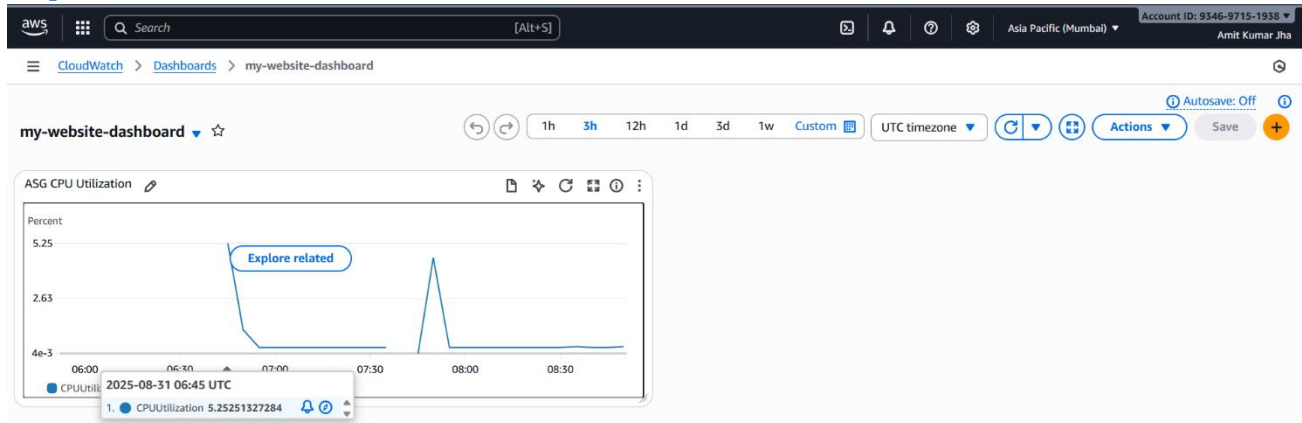
Screenshot 11: Auto Scaling Group

Step 12: Create 2 CloudWatch Alarms



Screenshot 12: CloudWatch Alarm

Step 13: Create a CloudWatch Dashboard



Screenshot 13: Monitoring Dashboard

Step 14: Attach Domain to Load Balancer

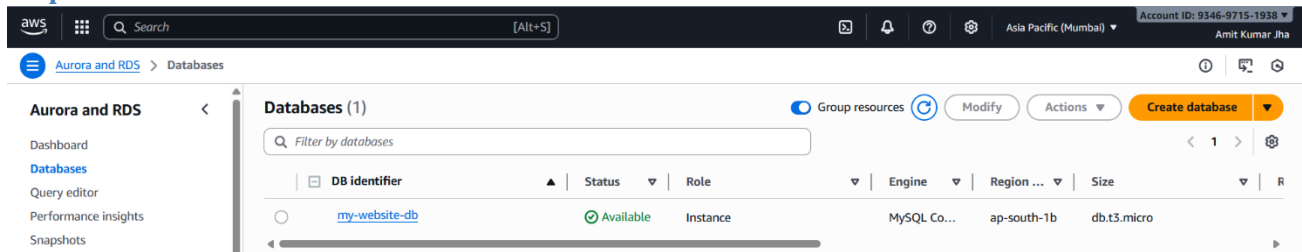
The screenshot shows the dccgodaddy.com DNS management interface. The left sidebar contains a navigation menu with options like 'World Universe', 'Dashboard', 'Domain', 'Website', 'Email', 'Store', 'Appointments', and 'Marketing'. The main area displays a table of DNS records:

Type	Name	Data	TTL	Delete	Edit
A	@	1.2.3.4	600 seconds		
NS	@	ns69.domaincontrol.com.	1 Hour	Can't delete	Can't edit
NS	@	ns70.domaincontrol.com.	1 Hour	Can't delete	Can't edit
CNAME	www	my-project-alb-1563156590.eu-west-3.elb.amazonaws.com.	1 Hour		

Screenshot 14: DNS Mapping

By- Amit Kumar Jha

Step 15: Launch Database in Private Subnet

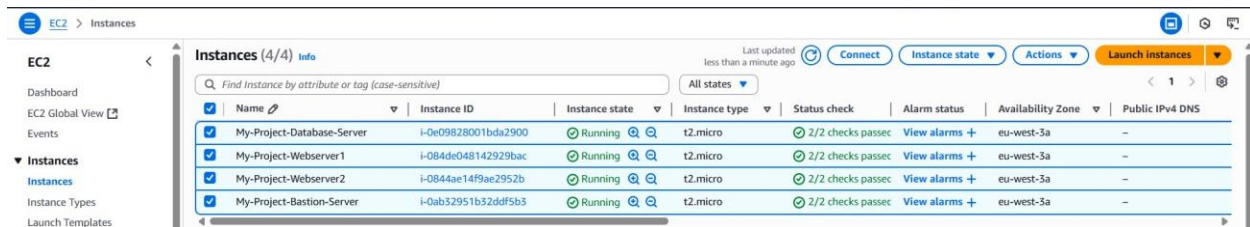


Screenshot 15: DB Instance Setup

Step 16: Outputs & Screenshots

This step showcases the visual output of the hosted website, including various pages such as Home, About, Contact, and Services. These screenshots demonstrate that the website is fully functional, accessible via the domain, and properly served through the load balancer.

Screenshot 16: Deployed Server Instances

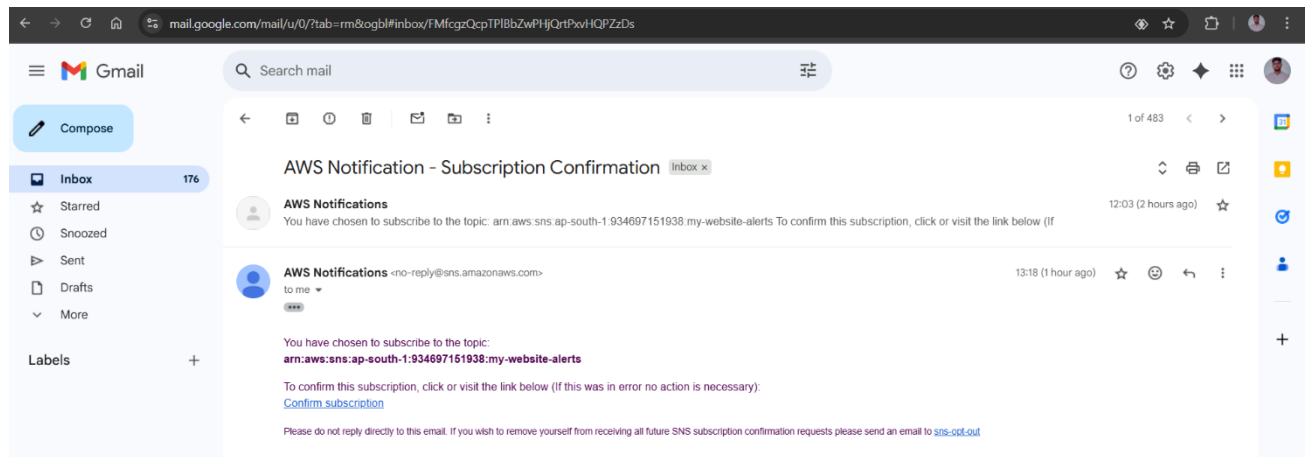


```
mysql> show tables;
+-----+
| Tables_in_JayshriProjectdb |
+-----+
| Employee                    |
+-----+
1 row in set (0.01 sec)

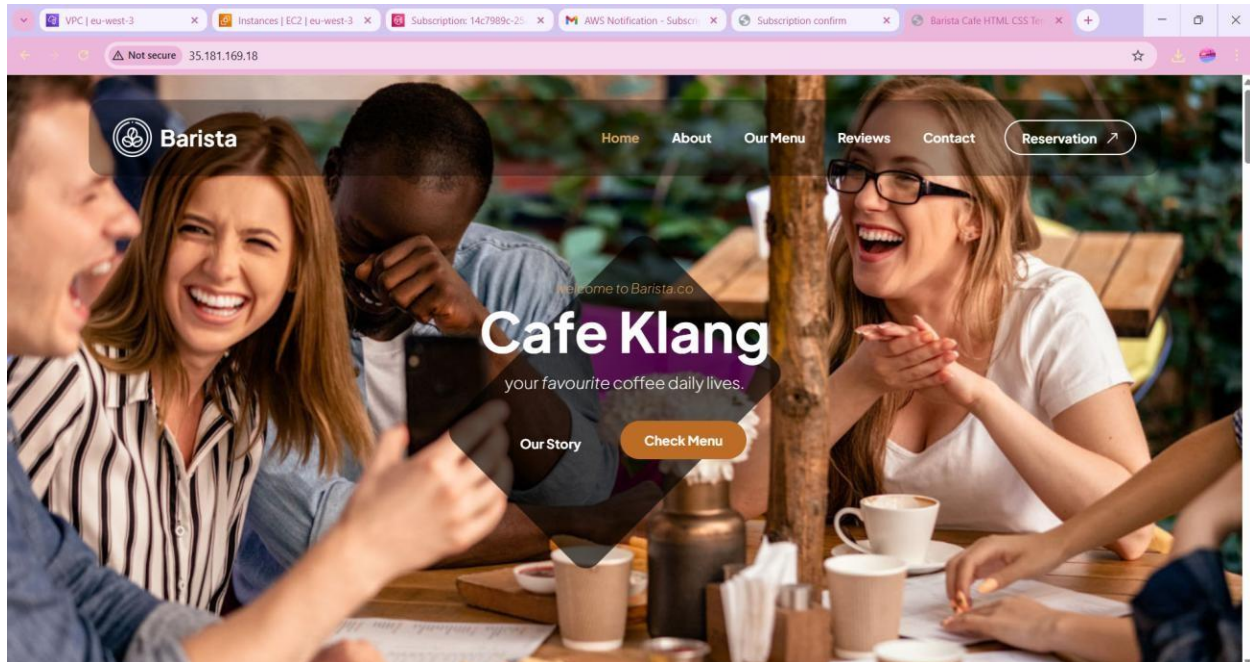
mysql> select * from Employee;
+-----+-----+-----+-----+-----+-----+-----+-----+
| employee_id | name   | designation | department | salary | email_id          | phone_no | city   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1           | Rohan  | Developer   | IT          | 60000  | rohan@gmail.com   | 2147483647 | Pune   |
| 2           | Neha   | HR          | HR          | 55000  | neha@gmail.com    | 2147483647 | Mumbai |
| 3           | Ravi   | Analyst     | Finance     | 58000  | ravi@gmail.com    | 2147483647 | Mumbai |
| 4           | Sneha  | Team Lead   | IT          | 80000  | sneha@gmail.com   | 2147483647 | Aurangabad |
| 5           | Sai    | Salesman    | Sales       | 45000  | sai@gmail.com     | 2147483647 | Nagpur  |
| 6           | Roshani | Accountant  | Finance     | 52000  | roshani@gmail.com | 2147483647 | Pune    |
| 7           | Rahul  | DevOps Engg | IT          | 67000  | rahul@gmail.com   | 2147483647 | Nagpur  |
| 8           | Mayur  | Intern      | IT          | 15000  | mayur@gmail.com   | 2147483647 | Pune    |
+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

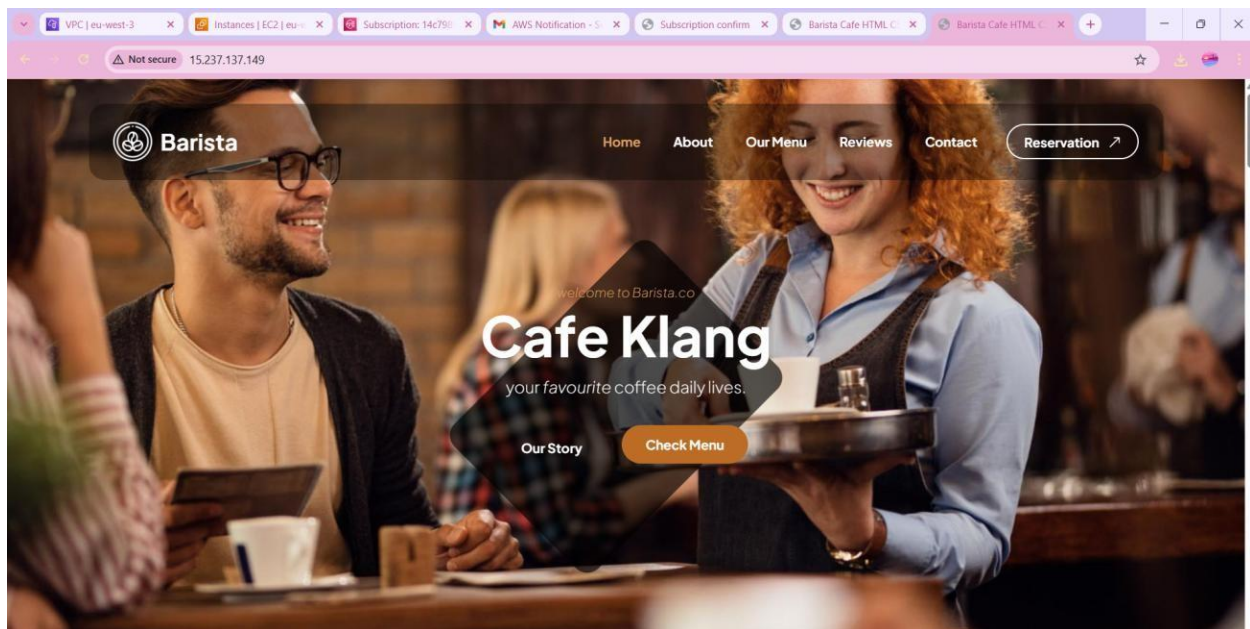
Screenshot 17: Database Table In Private Subnet



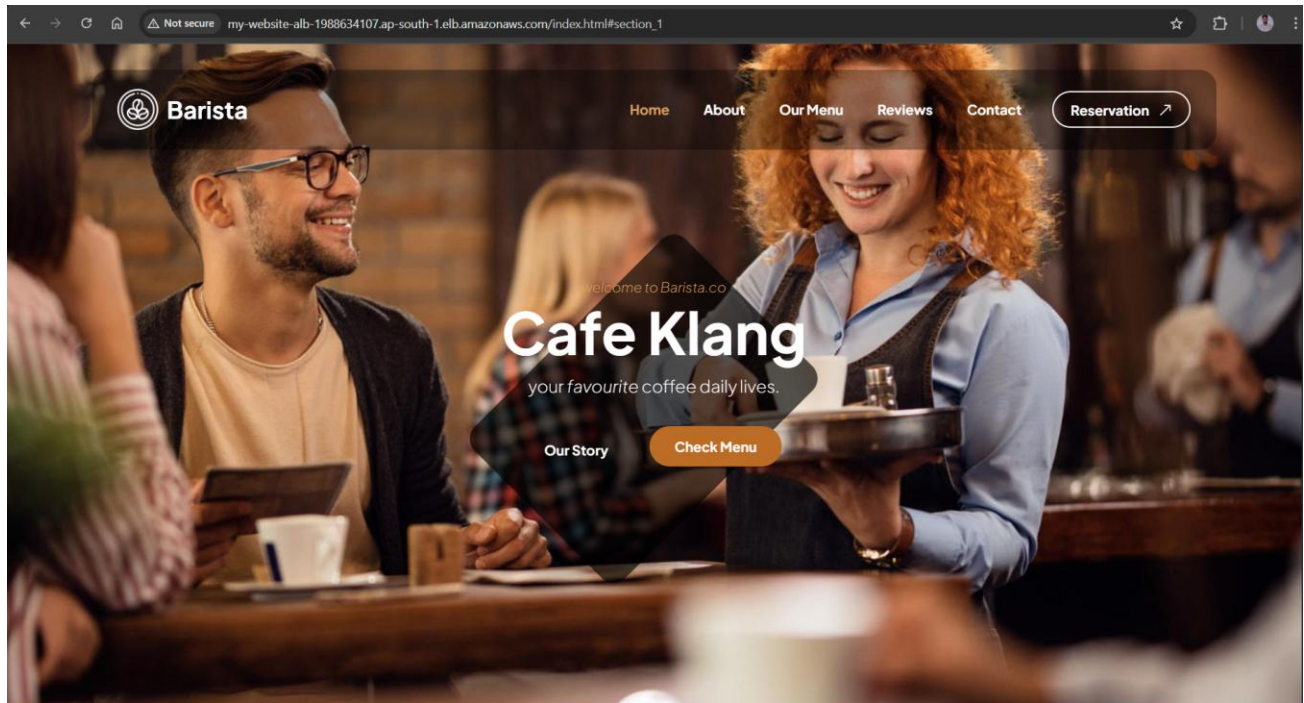
Screenshot 18: SNS Notification Received in Gmail



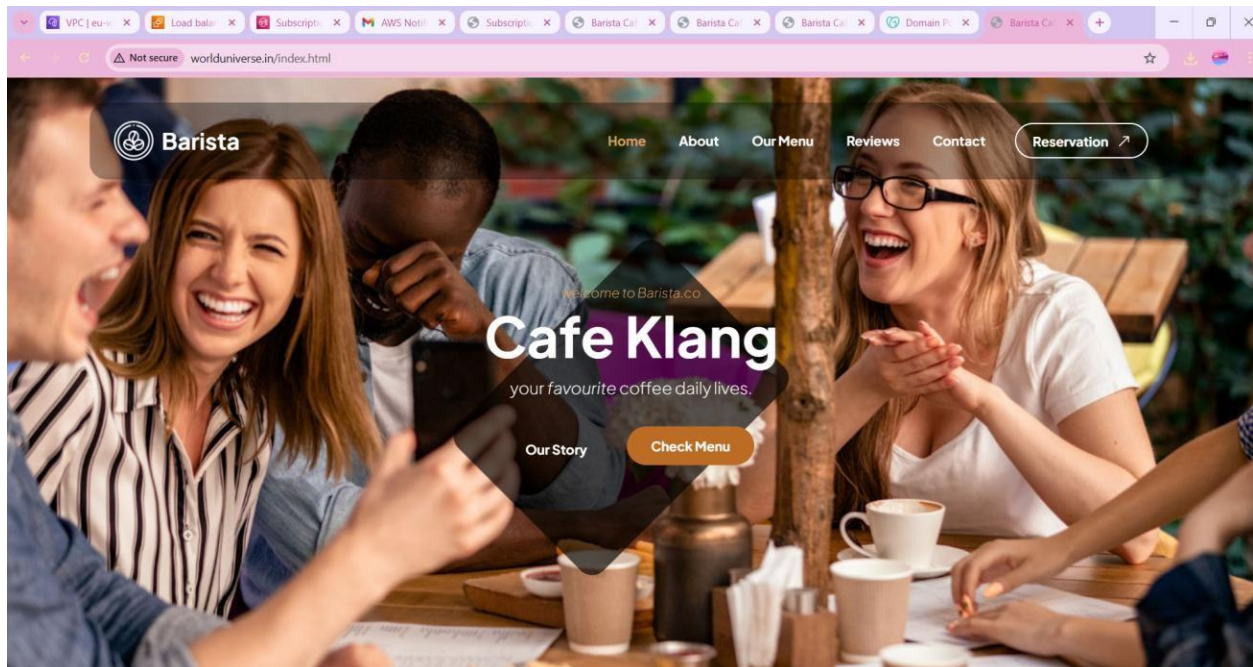
Screenshot 19: Output of My_Project_Webserver1 via Public IP



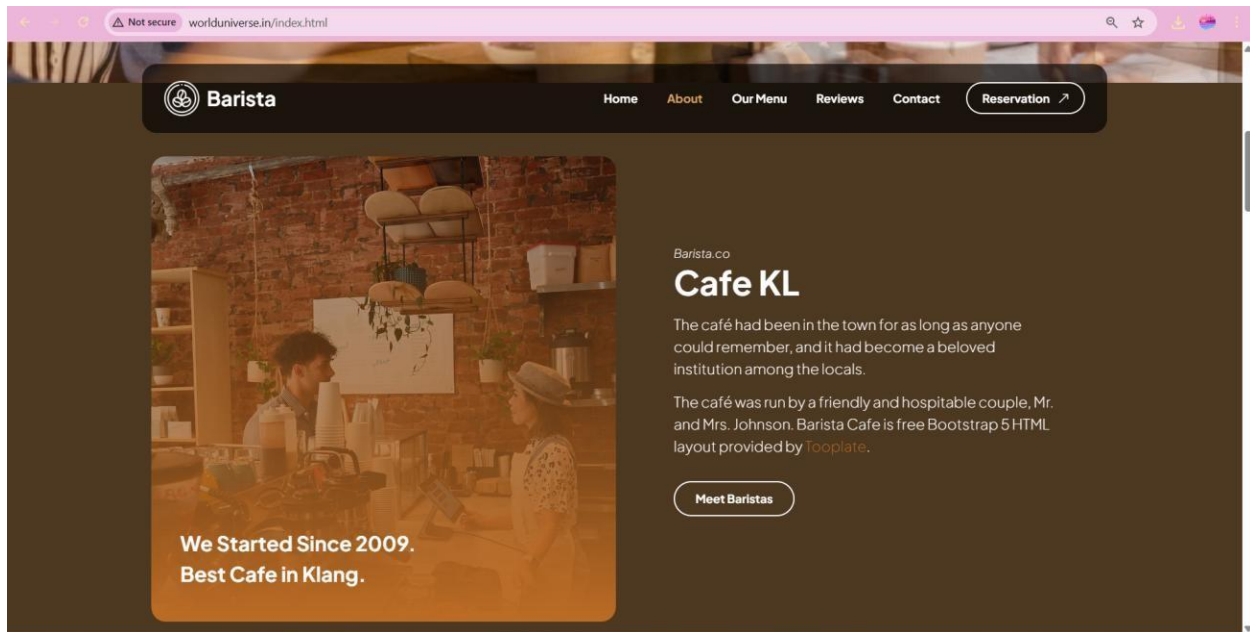
Screenshot 20: Output of My_Project_Webserver2 via Public IP



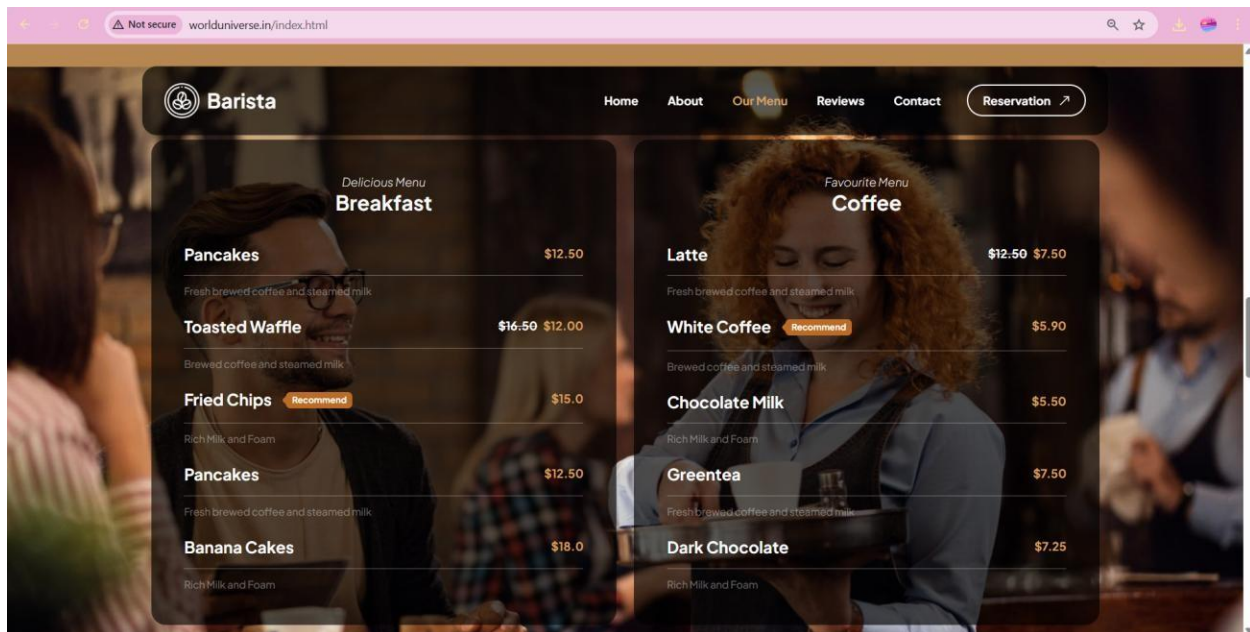
Screenshot 21: Application Output via Load Balancer DNS



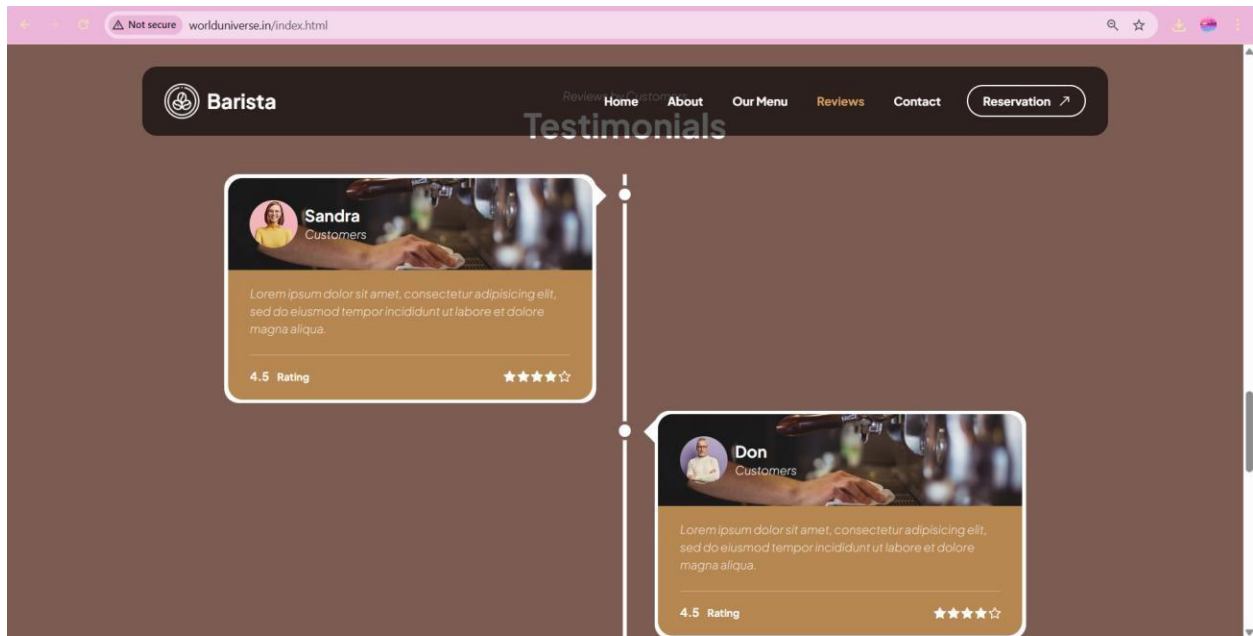
Screenshot 22: Final Deployed Website Home Page with Domain



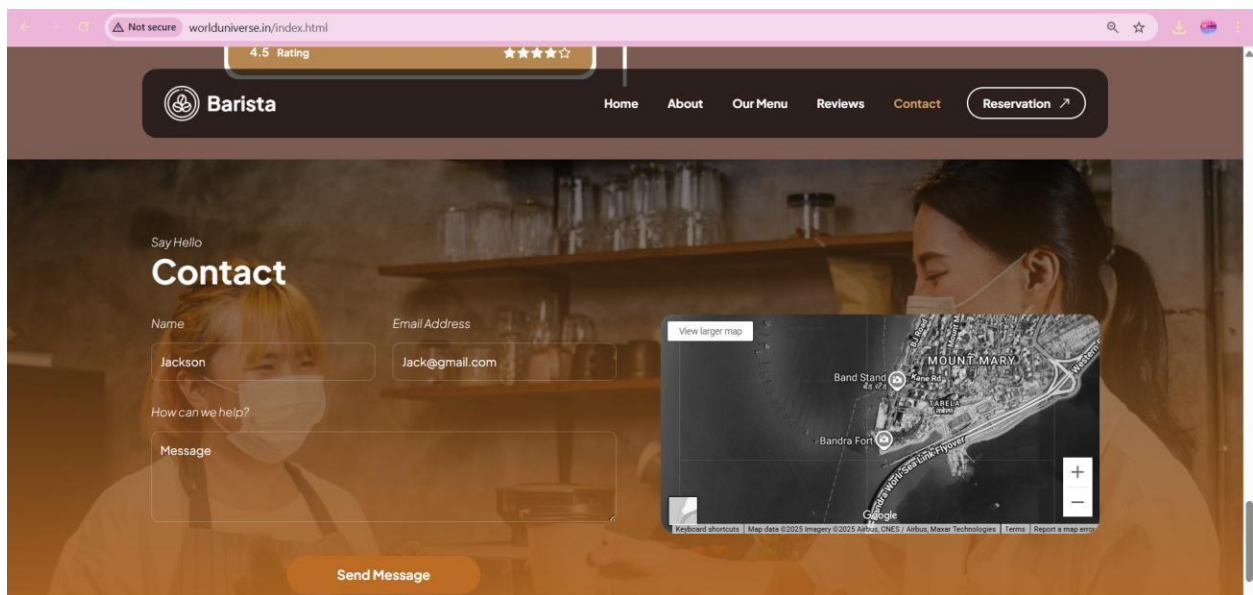
Screenshot 23: Deployed Website About Page Using Domain



Screenshot 24: Deployed Website Our Menu Section via Domain



Screenshot 25: Customer Reviews Section Using Domain

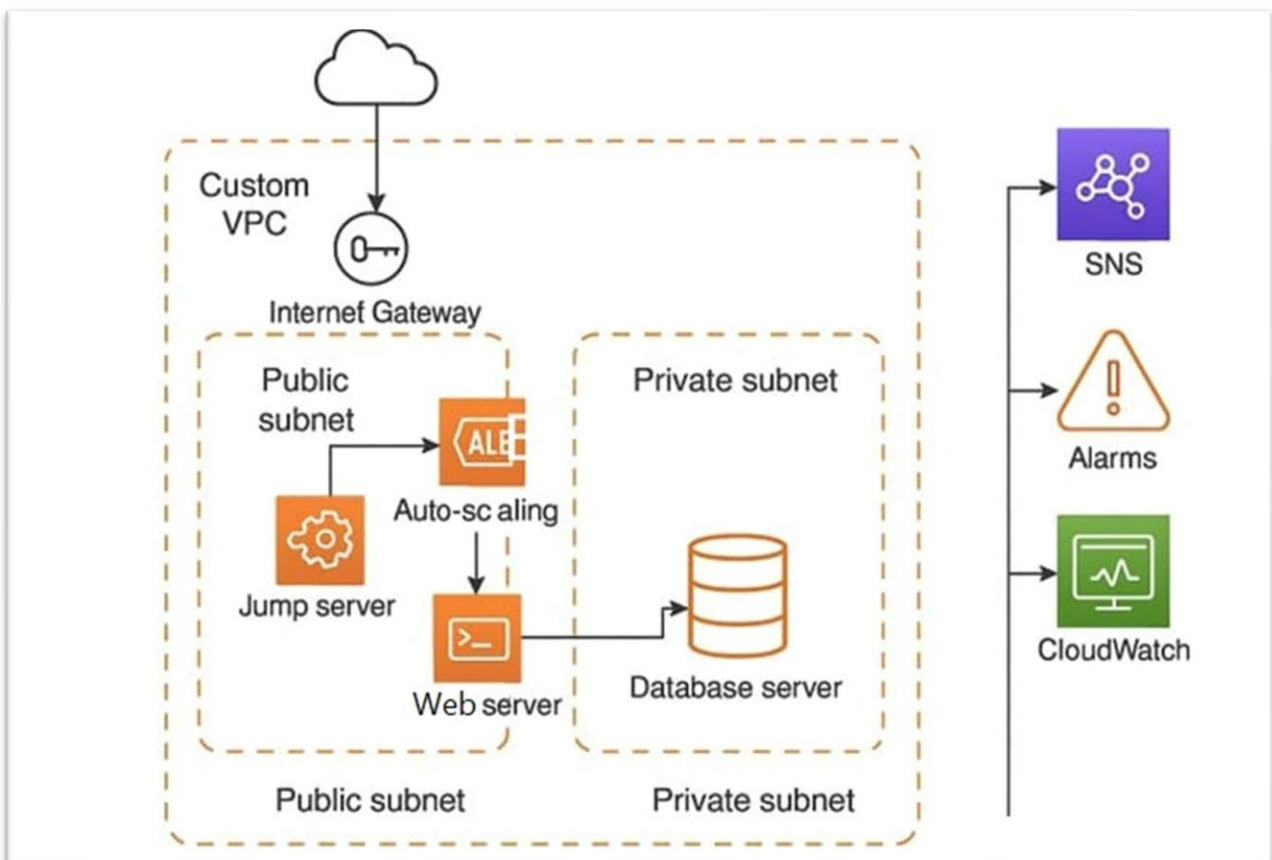


Screenshot 26: Contact Page of Final Deployed Website

Project Summary

This project focuses on deploying a reliable, scalable, and secure web hosting infrastructure using core AWS services. Below are the key highlights of the implementation:

- Hosted a scalable and highly available website on AWS using a custom Virtual Private Cloud (VPC).
- Implemented Auto Scaling and Application Load Balancer (ALB) for efficient traffic management and fault tolerance.
- Enabled monitoring using Amazon CloudWatch with configured alarms and dashboards for performance visibility.
- Integrated AWS SNS for real-time alert notifications to ensure quick response to system events.
- Deployed a secure architecture by isolating the database server in a private subnet.



Screenshot 27: AWS Project Architecture Diagram