

A CAPSTONE PROJECT ON  
  
HIGHLY AVAILABLE WEB APPLICATION  
  
UNDERTAKEN AT  
QWIKSKILLS  
By  
Ms. Malvika Grover

SUBMITTED TO



**QWIKSKILLS**

As Partial Fulfilment of the Requirements  
for the Program  
**AWS ESSENTIALS**  
**JUNE - 2023**

GUIDED By  
**Mr. SURAJ VERMA**

Reference No.

Date: 04/12/2023

## DECLARATION

I Malvika Grover a student of the AWS Essentials Program declared that the project report submitted by me is a bonafide work for the partial fulfilment of the requirement of the AWS Essentials project work. I have incorporated all the suggestions provided by my guide from time to time.

I further declare that to the best of my knowledge this dissertation contains my original work and does not contain any part of any work which has been submitted for any work entitlement either in this organisation or in any other university/Deemed university/institute organisation etc. without proper citation and I shall be fully responsible for any plagiarism found at any stage.



Name: Ms. Malvika Grover

Reference No.

Date: 04/12/2023

## Capstone Project Approval Certificate

This is to certify that Ms. Malvika Grover, a student of “AWS Essentials” at QwikSkills, has successfully completed the project work entitled “ Highly Available Web Application ” as per guidance. I have thoroughly assessed the progress of the work and reviewed the end result. The student has incorporated all the suggestions provided by instructors in this dissertation. This dissertation is a bona fide piece of work of the standard of capstone project work carried out by the student under instructor’s supervision. Internal examination has been successfully completed.

Name: Mr. Suraj Verma  
(AWS Cloud Engineer)

S.NO.	INDEX	PAGE NO.
1	<b>Introduction</b>	4
2	<b>Background</b>	5
3	<b>Requirements</b>	6
4	<b>Overview</b>	7
5	<b>Descriptions</b>	8
6	<b>Data Flow Diagrams (DFD)</b>	10
7	<b>Flowchart (Modules &amp; Sub-Modules)</b>	11
8	<b>Data Description (E-R diagrams)</b>	12
9	<b>Sample Code</b>	13
10	<b>Screenshots</b>	16
11	<b>Conclusion</b>	36
12	<b>Salient Features</b>	37
13	<b>Future Scope</b>	38
14	<b>Bibliography/ References</b>	39

# INTRODUCTION

## *Harnessing the Power of AWS to Craft Resilient and Scalable Web Applications*

In today's ever-evolving digital landscape, the demand for robust and scalable web applications has never been greater. Users expect seamless experiences and uninterrupted services, and businesses must deliver. The "Highly Available Web Application" project takes on this challenge directly, using the advanced features of Amazon Web Services (AWS) to create a web structure that goes beyond the usual limits.

In a digital ecosystem where user satisfaction hinges on seamless experiences and uninterrupted services, the focus of this project is to engineer a web application that goes beyond the traditional boundaries of availability, scalability, and fault tolerance.

The core objective is to harness the capabilities of AWS services, including Amazon EC2, Amazon RDS, Auto Scaling, Elastic Load Balancing, CloudTrail, S3, CloudWatch, and SNS, to construct an architecture that stands resilient in the face of challenges.

As I start this journey, my goal is clear: to create a web application that works well with different amounts of work and handles a lot of users without stopping. We'll use the latest AWS tools to automatically adjust how much we need, share the work well, and quickly fix any problems. This project isn't just a guide; it's like a plan for making a web app that goes beyond the usual way of doing things. It's about making a website that stays smooth and reliable even when things get tough. Come along as we explore how to build a web app that can handle a lot, a true example of the future of strong and flexible websites.

The vision of this project is to design and deploy a resilient and scalable web application architecture capable of handling high traffic and maintaining availability. By utilizing AWS services such as EC2 instances, RDS databases, Auto Scaling groups, and Load Balancers, the architecture ensures automatic scalability and even traffic distribution. This approach enables the web application to seamlessly handle increased demand and maintain optimal performance even during peak loads or unexpected instance failures. The project aligns with the modern paradigm of cloud-native development, focusing on scalability, reliability, and efficiency.

# BACKGROUND

In the contemporary digital landscape, the seamless functioning of web applications with high availability and scalability has become paramount to delivering an optimal user experience. Recognizing this, Amazon Web Services (AWS), a prominent cloud services provider, offers an array of tools and services empowering developers to construct resilient and scalable architectures.

The focal point of this project is to harness the capabilities of AWS services in crafting a web application infrastructure capable of dynamically adapting to diverse workloads, mitigating failures, and sustaining peak performance levels.

Web applications operate within a dynamic environment characterized by interactive user experiences, variable traffic patterns, and the constant potential for unforeseen system challenges. Developers and organizations alike now prioritize achieving high availability and scalability to meet the evolving expectations of modern users. AWS, distinguished in the realm of cloud computing, provides a versatile and dependable platform, allowing developers to architect solutions that adeptly respond to the unpredictable nature of the contemporary digital landscape.

Through the strategic utilization of AWS services, this project seeks to establish a robust foundation for constructing a web application that not only meets user expectations but surpasses them. The goal is to ensure a dependable, scalable, and responsive user experience, addressing the ongoing evolution of digital demands.

Leveraging AWS services facilitates the creation of a contemporary, adaptable infrastructure capable of navigating the intricacies of present-day digital challenges, thereby providing a sturdy and efficient basis for web application development.

# REQUIREMENTS

The AWS architecture for a highly available web application typically consists of the following components:

- Load balancer: A load balancer distributes traffic across multiple web servers, ensuring that no single server is overloaded.
- Web servers: The web servers host the web application and serve it to users.
- Database: The database stores the data used by the web application.
- Auto scaling group: An auto scaling group automatically scales the number of web servers up or down based on demand, ensuring that the web application can handle changes in traffic.
- CloudTrail: CloudTrail audits all activity in the AWS account, providing a record of who did what when.
- S3: S3 stores log files for the web application, providing a durable and secure repository for data analysis and troubleshooting.
- SNS: SNS provides a highly scalable and reliable pub/sub messaging service that can be used to send notifications to subscribers when certain events occur.
- CloudWatch: Implementation of AWS CloudWatch for real-time monitoring, logging, and automatic scaling, ensuring proactive management of the web application's performance and resource utilization as a key project requirement

## **Technical Requirements:**

- Proficiency in AWS services, including EC2, RDS, Auto Scaling, Elastic Load Balancing, CloudTrail, Route 53, CloudWatch, and SNS.
- Understanding of networking concepts, security configurations, and database management.
- Familiarity with Infrastructure as Code (IaC) tools for efficient resource provisioning (optional).

## **Infrastructure Requirements:**

- Access to an AWS account with the necessary permissions to create and manage resources.
- Reliable internet connectivity for accessing AWS services and documentation.

# OVERVIEW

The project's objective is to establish a highly available web application using Amazon EC2, Amazon RDS, Auto Scaling, and Elastic Load Balancing, ensuring robust architecture with high availability, scalability, and fault tolerance. Utilizing CloudTrail, S3, CloudWatch, and SNS enables proactive measures for EC2 instances, Load Balancer, Auto Scaling group, Databases that is documented in a comprehensive step-by-step guide. The architecture incorporates load balancers for traffic distribution, auto scaling groups for dynamic server adjustments, CloudTrail for auditing, S3 for log storage, and SNS for event notifications. This integrated approach provides businesses with a solution that excels in high availability, scalability, security, and cost-effectiveness.

This project revolves around the development and deployment of a highly available web application using Amazon Web Services (AWS). The objective is to create an architecture that ensures optimal performance, fault tolerance, and scalability by integrating key AWS services. The chosen components, including Amazon EC2, Amazon RDS, Auto Scaling, Elastic Load Balancing, CloudTrail, Route 53, CloudWatch, and SNS, collectively contribute to the project's success by providing a robust and proactive environment.

# DESCRIPTION

## 1. EC2 Instance Creation:

- Involves launching and configuring EC2 instances for hosting the web application.
- Selection of AMI, instance type, storage options, security settings, and networking configurations.
- The aim is to distribute workload and ensure availability in case of instance failures.

## 2. RDS Database Setup:

- Focuses on setting up Amazon RDS to provide a managed database service.
- Selection of the database engine, instance details, connectivity options, and security configurations.
- Configuration of backup and maintenance options for data durability and availability.

## 3. Auto Scaling Group Configuration:

- Involves setting up an Auto Scaling group to dynamically adjust EC2 instances based on demand.
- Configuration of scaling policies based on metrics, specifying instance capacities, and defining cooldown periods.
- Ensures the application can handle increased demand without manual intervention.

## 4. Load Balancer Setup:

- Focuses on setting up Elastic Load Balancing to evenly distribute incoming traffic.
- Selection of load balancer type, configuration of settings including listener ports, and SSL certificates.
- Enhances availability and fault tolerance by distributing traffic evenly.

## 5. CloudTrail Creation:

- Enables monitoring and logging of AWS API calls for enhanced traceability.
- Configuration to store logs in Amazon S3 for durable and scalable storage.
- Essential for maintaining visibility and tracking changes in the infrastructure.

7. CloudWatch:

- Collects and tracks metrics, logs, and events from various AWS resources.
- Configured to monitor CloudTrail logs and detect stopped EC2 instances.
- Allows proactive measures to ensure high availability of the web application.

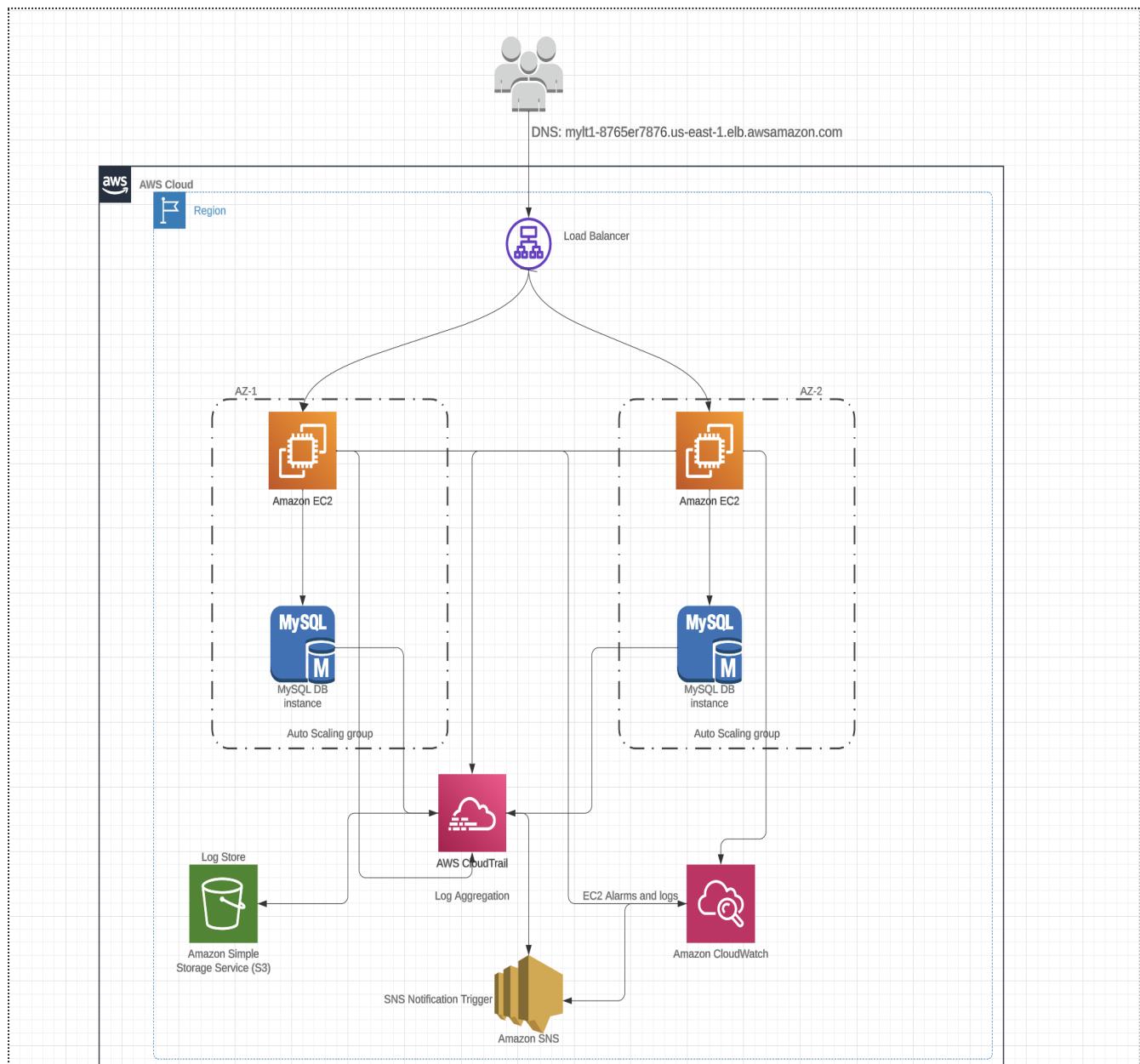
8. SNS (Simple Notification Service):

- Sends notifications to subscribed endpoints or clients.
- Used to send email notifications when CloudWatch detects a stopped EC2 instance.
- Facilitates quick alerting and actions to address potential issues, ensuring reliability.

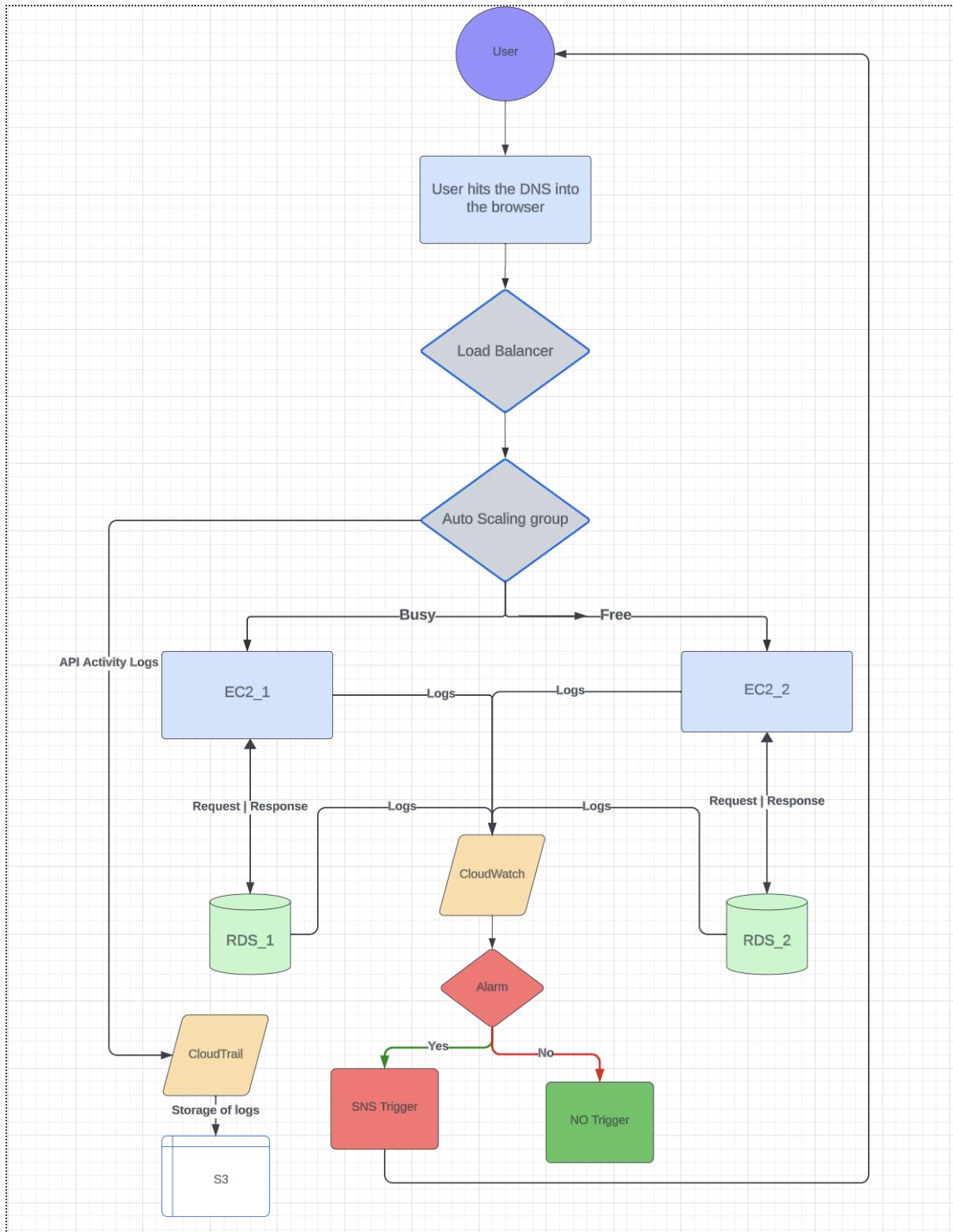
9. S3 (Simple Storage Service):

- Configuring Amazon Simple Storage Service (S3) to facilitate robust storage and retrieval of log files generated by the web application
- Bucket Creation: Establishing an S3 bucket to serve as a secure and scalable repository for storing log files and other essential data.
- Access Controls: Defining precise access controls and permissions for the S3 bucket, ensuring secure data storage and compliance with security best practices.

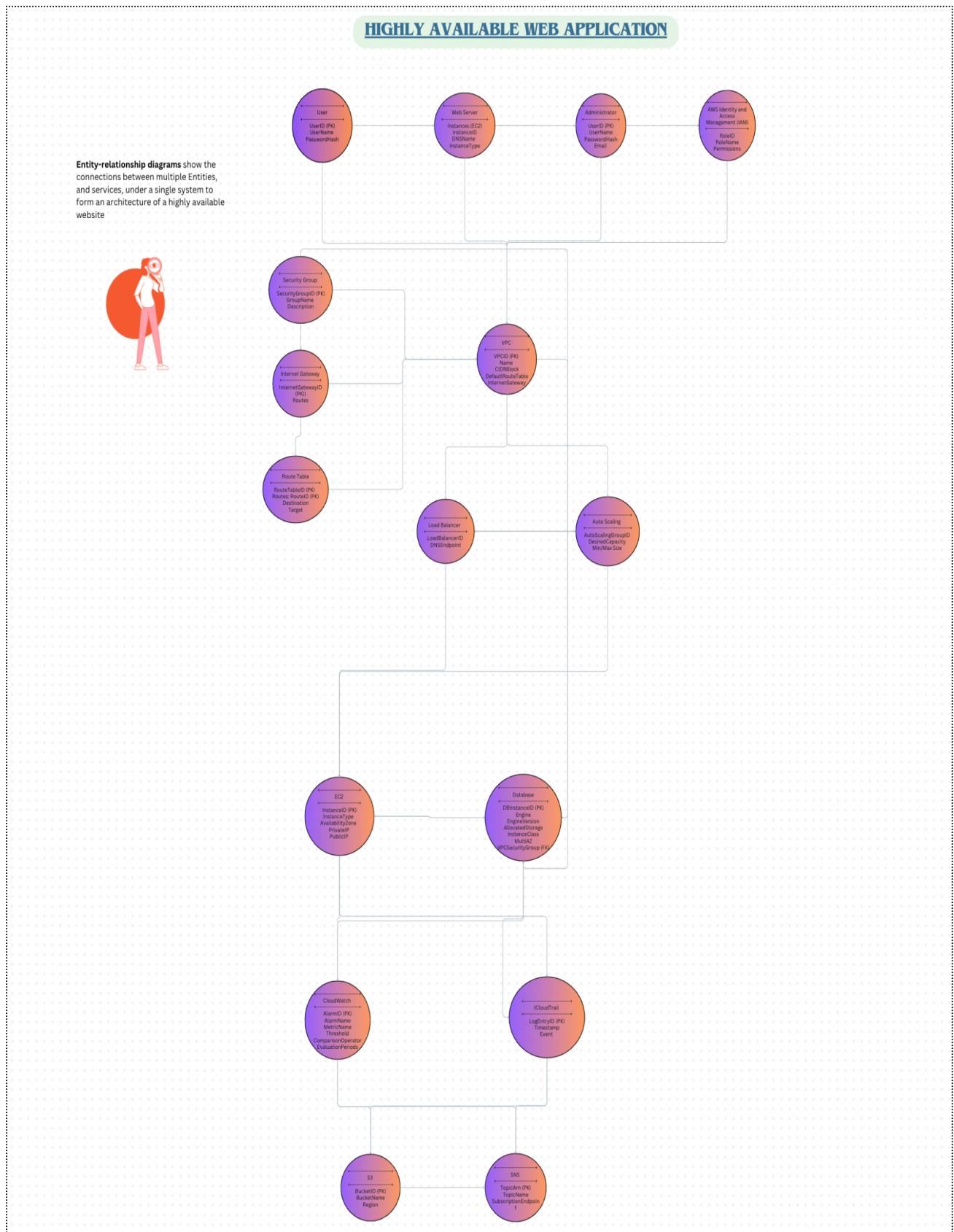
# DATA FLOW DIAGRAMS (DFD)



# FLOWCHART (MODULES & SUB-MODULES)



# DATA DESCRIPTION (E-R DIAGRAMS)



# SAMPLE CODE

## Script 1: RDS connection from ec2

```
# Database configuration
host="mysqladb.cyjffj58cggo4.us-east-1.rds.amazonaws.com"
username="admin"
password="admin123"
database="mydatabase"
table="table01"

# Install required packages
sudo apt update
sudo apt install -y apache2 php libapache2-mod-php php-mysql

# To connect to the database
$ mysql -h mysqladb.cyjffj58cggo4.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
admin123
```

## Script 2: RDS dummy data insertion

```
# Creating DB
mysql> CREATE DATABASE mydatabase;
Query OK, 1 row affected (0.00 sec)

# List DB
mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mydatabase    |
| mysql          |
| mysqlDB        |
| performance_schema |
| sys            |
+-----+
6 rows in set (0.00 sec)
```

```

# Pointing the desired DB to use

mysql> USE mydatabase;
Database changed


# Creating Table using the desired Schema

mysql> Create table table01 (id INT PRIMARY KEY, name VARCHAR(50), age INT);
Query OK, 0 rows affected (0.04 sec)

# Result

mysql> SHOW TABLES;
+-----+
| Tables_in_mydatabase |
+-----+
| table01              |
+-----+
1 row in set (0.00 sec)

# Interesting 1st Data into table just created

mysql> INSERT INTO table01(id,name,age) VALUES (01,'Ross',35);
Query OK, 1 row affected (0.00 sec)

# Interesting 2nd Data into table just created

mysql> INSERT INTO table01(id,name,age) VALUES (02,'Chandler',32);
Query OK, 1 row affected (0.00 sec)

# Interesting 3rd Data into table just created

mysql> INSERT INTO table01(id,name,age) VALUES (03,'Rachel',32);
Query OK, 1 row affected (0.01 sec)

# Interesting 4th Data into table just created

mysql> INSERT INTO table01(id,name,age) VALUES (04,'Phebe',30);
Query OK, 1 row affected (0.00 sec)

# Interesting 5th Data into table just created

mysql> INSERT INTO table01(id,name,age) VALUES (05,'Joe',30);
Query OK, 1 row affected (0.00 sec)

```

```
# Final Result
mysql> SELECT * FROM table01;
+----+-----+-----+
| id | name   | age   |
+----+-----+-----+
| 1  | Ross    | 35    |
| 2  | Chandler | 32    |
| 3  | Rachel   | 32    |
| 4  | Phoebe  | 30    |
| 5  | Joe      | 30    |
+----+-----+-----+
5 rows in set (0.00 sec)
```

## Script 3: Apache server fetches data from RDS

```
sudo tee /var/www/html/index.php >/dev/null <<EOF
<?php
\$conn = new mysqli("$host", "$username", "$password", "$database");
if (\$conn->connect_error) {
    die("Connection failed: " . \$conn->connect_error);
}
\$sql = "SELECT * FROM \$table";
\$result = \$conn->query(\$sql);
if (\$result->num_rows > 0) {
    echo "<table><tr><th>ID</th><th>Name</th></tr>";
    while (\$row = \$result->fetch_assoc()) {
        echo "<tr><td>" . \$row["id"] . "</td><td>" . \$row["name"] . "</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}
\$conn->close();
?>
EOF

# Configure Apache to serve PHP files
sudo sed -i "s/index.html/index.php/g" /etc/apache2/mods-enabled/dir.conf

# Restart Apache
sudo systemctl restart apache2

# Display public IP
echo "Web application is now accessible at: http://$(curl -s
http://checkip.amazonaws.com) /"
```

# SCREENSHOTS

## VPC

**VPC Details**

**Details**

- VPC ID: vpc-0c903feef5a8edd5b
- State: Available
- Tenancy: Default
- Default VPC: No
- Network Address Usage metrics: Disabled
- DNS hostnames: Enabled
- Main route table: rtb-06c6144241d3a0da
- IPv4 CIDR: 175.1.0.0/16
- Route 53 Resolver DNS Firewall rule groups: -
- IPv6 pool: -
- Owner ID: 005238736607
- DNS resolution: Enabled
- Main network ACL: acl-01d9cb9280401f590
- IPv6 CIDR (Network border group): -

**Resource map**

- VPC:** myVPC-vpc
- Subnets (2):** us-east-1a (public-subnet-1), us-east-1b (public-subnet-2)
- Route tables (2):** myVPC-rtb-public (rtb-06c6144241d3a0da)
- Network connections (1):** myVPC-igw

**Create VPC**

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

**VPC settings**

**Resources to create**

- VPC only
- VPC and more

**Name tag auto-generation**

Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

Auto-generate

**IPv4 CIDR block**

Determine the starting IP and the size of your VPC using CIDR notation.

175.1.0.0/16      65,536 IPs

**IPv6 CIDR block**

No IPv6 CIDR block

Amazon-provided IPv6 CIDR block

**Tenancy**

Default

**Number of Availability Zones (AZs)**

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1	2	3
---	---	---

**Customize AZs**

**Preview**

- VPC:** myVPC-vpc (175.1.0.0/16, No IPv6)
- Subnets (2):** us-east-1a (myVPC-subnet-public1-us-east-1a), us-east-1b (myVPC-subnet-public2-us-east-1b)
- Route tables (1):** myVPC-rtb-public (0.0.0.0/0 routes to myVPC-igw)
- Network connections (1):** myVPC-igw (Internet routes to 2 public subnets route 1)

Screenshot of the AWS VPC Wizard - Step 2: Configure VPC Settings.

**Customize AZs**

- Number of public subnets:** Info (2)
- Number of private subnets:** Info (0)
- Customize subnets CIDR blocks:**
  - Public subnet CIDR block in us-east-1a:** 175.1.0.0/20 (4,096 IPs)
  - Public subnet CIDR block in us-east-1b:** 175.1.16.0/20 (4,096 IPs)
- NAT gateways (\$):** Info (None)
- VPC endpoints:** Info (None)
- DNS options:** Info (Enable DNS hostnames, Enable DNS resolution)

**Preview**

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS VPC Wizard - Step 3: Review and Create.

**Create VPC workflow**

**Success**

- Create VPC: vpc-0c903feef5a8eddd5b
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: vpc-0c903feef5a8eddd5b
- Create subnet: subnet-04e9918f8265d6d62
- Create subnet: subnet-05b58370e08fe4bc4
- Create internet gateway: igw-000dd1db9a06e666f4
- Attach internet gateway to the VPC
- Create route table: rtb-00132a9163f5da558
- Create route
- Associate route table
- Associate route table
- Verifying route table creation

**View VPC**

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Subnets

**Subnets (2/8) Info**

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPV4 address
-	subnet-088f1d83d936f1a9b	Available	vpc-022d154fc5b8aa5a	172.31.80.0/20	-	4091
-	subnet-006b680b9dcf3e76	Available	vpc-022d154fc5b8aa5a	172.31.64.0/20	-	4091
-	subnet-0765f4a114c44bd3d	Available	vpc-022d154fc5b8aa5a	172.31.48.0/20	-	4091
<input checked="" type="checkbox"/> public-subnet-2	subnet-05b58370e08fe48c4	Available	vpc-0c903feef5a8edd5b1myV...	175.1.16.0/20	-	4091
-	subnet-06894cb31e45f75e	Available	vpc-022d154fc5b8aa5a	172.31.16.0/20	-	4091
-	subnet-07576d684a062ae99	Available	vpc-022d154fc5b8aa5a	172.31.0.0/20	-	4091
<input checked="" type="checkbox"/> public-subnet-1	subnet-04c9918f8265dd682	Available	vpc-0c903feef5a8edd5b1myV...	175.1.0.0/20	-	4091
-	subnet-079f6aaaf8edcf92	Available	vpc-022d154fc5b8aa5a	172.31.32.0/20	-	4091

Subnets: subnet-04c9918f8265dd682, subnet-05b58370e08fe48c4

## Route tables

**rtb-00132a9163f5da558 / myVPC-rtb-public**

**Details** **Info**

Route table ID rtb-00132a9163f5da558	Main No	Explicit subnet associations 2 subnets	Edge associations -
VPC vpc-0c903feef5a8edd5b1myVPC-vpc	Owner ID 005238736607		

**Routes** **Subnet associations** **Edge associations** **Route propagation** **Tags**

**Explicit subnet associations (2)**

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
public-subnet-2	subnet-05b58370e08fe48c4	175.1.16.0/20	-
public-subnet-1	subnet-04c9918f8265dd682	175.1.0.0/20	-

**Subnets without explicit associations (0)**

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
------	-----------	-----------	-----------

No subnets without explicit associations  
All your subnets are associated with a route table.

# Security group

The screenshot shows the 'Create security group' page in the AWS VPC console. In the 'Basic details' section, the security group name is 'mySG' and the description is 'Allowing all incoming and outgoing traffic from anywhere'. The VPC is set to 'vpc-0c903feef5a8edd5b (myVPC-vpc)'. Under 'Inbound rules', there is one rule allowing all traffic from anywhere. A warning message at the bottom states: '⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' The top navigation bar includes services like AWS, EFS, and EC2.

The screenshot shows the 'Create security group' page in the AWS VPC console. It includes both 'Inbound rules' and 'Outbound rules' sections. Both sections show a single rule allowing all traffic from anywhere. A warning message at the bottom of each section states: '⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' The top navigation bar includes services like AWS, EFS, and EC2.

The screenshot shows the AWS VPC console with a success message: "Security group (sg-03bd37b0aa7aa4768 | mySG) was created successfully". The main panel displays the details for the security group "sg-03bd37b0aa7aa4768 - mySG".

Security group name	Security group ID	Description	VPC ID
mySG	sg-03bd37b0aa7aa4768	Allowing all incoming and outgoing traffic from anywhere	vpc-0c903feef5a8edd5b

**Inbound rules (1)**

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-071ce82e74ddada28	IPv4	All traffic	All	All	0.0.0.0/0	-

## Internet gateway

The screenshot shows the AWS VPC console with the "Internet gateways (1)" page. It lists a single internet gateway named "myVPC-igw" which is attached to the VPC.

Name	Internet gateway ID	State	VPC ID	Owner
myVPC-igw	igw-00dd1db9a06e666f4	Attached	vpc-0c903feef5a8edd5b   myVPC-vpc	005238736607

Select an internet gateway above

# EC2 Instances

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation links: EC2 Dashboard, Services, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, and Load Balancers. The main content area displays a table titled "Instances (1/2) Info" with two rows. Row 1 contains "Instance\_1" (ID: i-03a99d93c696f3e4c, State: Running, Type: t2.micro, Status checks: 2/2 checks passed, Availability Zone: us-east-1a). Row 2 contains "Instance\_2" (ID: i-08a1f9bd962608199, State: Running, Type: t2.micro, Status checks: 2/2 checks passed, Availability Zone: us-east-1b). Below the table, there's a detailed view for "Instance: i-03a99d93c696f3e4c (Instance\_1)". This view includes tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. Under the Details tab, it shows Instance ID (i-03a99d93c696f3e4c), Public IPv4 address (-), Private IPv4 addresses (175.1.7.39), IPv6 address (-), Instance state (Running), Public IP DNS (-), Hostname type (IP name: ip-175-1-7-39.ec2.internal), Private IP DNS (ip-175-1-7-39.ec2.internal), Answer private resource DNS name (-), Instance type (t2.micro), Elastic IP addresses (-), Auto-assigned IP address (-), VPC ID (vpc-0c903feef5a8ed5b (myVPC-vpc)), and AWS Compute Optimizer finding (Opt-in to AWS Compute Optimizer for recommendations). At the bottom of the page, there are links for CloudShell and Feedback, along with a standard browser toolbar.

# RDS instances

```
ubuntu@ip-175-1-7-39:~$ host="mysqlb.cyjffj58cgg04.us-east-1.rds.amazonaws.com"
username="admin"
password="admin123"
database="mydatabase"
table="table01"
ubuntu@ip-175-1-7-39:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:3 http://security.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 packages upgraded or downgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-175-1-7-39:~$ sudo apt install -y apache2 php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  apache2-bin apache2-data apache2-utils bz2ip2 libapache2-mod-php8.1 libapr1 libaprutil1 libaprutil1-db-sqlite3
  libaprutil1-ldap liblubus3.0 mailcap mime-support php-common php8.1 php8.1-clnt php8.1-common php8.1-mysql php8.1-opcache
php8.1-common php8.1-mysql php8.1-opcache php8.1-readline ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser bzip2-doc php-pear
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bz2ip2 libapache2-mod-php libapache2-mod-php8.1 libapr1 libaprutil1
  libaprutil1-ldap liblubus3.0 mailcap mime-support php-common php8.1 php8.1-clnt php8.1-common php8.1-mysql php8.1-opcache
php8.1-common php8.1-mysql php8.1-opcache php8.1-readline ssl-cert
0 upgraded, 24 newly installed, 0 to remove and 62 not upgraded.
Need to get 7483 kB of archives.
After this operation, 20.4 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-0ubuntu0.22.04.1 [100 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-Subuntu4.22.04.2 [92.8 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-Subuntu4.22.04.2 [11.3 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-amd64 1.6.1-Subuntu4.22.04.2 [9178 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-mysql5.5 amd64 1.6.1-Subuntu4.22.04.2 [140 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-Subuntu4.7 [1346 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-Subuntu4.7 [165 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-Subuntu4.7 [88.8 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-db-openssl amd64 1.6.1-Subuntu4.7 [36.8 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mime-support all 3.66 [3696 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 2.4.52-Subuntu4.7 [97.8 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bz2ip2 amd64 1.0.8-Subuntu1 [34.8 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libapr1-0.92ubuntu0.22.04.1 [100 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-0.92ubuntu0.22.04.2 [1127 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libphp8.1-opcache amd64 8.1.2-Subuntu2.14 [365 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libphp8.1-readline amd64 8.1.2-Subuntu2.14 [13.6 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libphp8.1-mysql amd64 8.1.2-Subuntu2.14 [1834 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libphp8.1-opcache amd64 8.1.2-Subuntu2.14 [1766 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libapache2-mod-php all 2.8.1-92ubuntu1 [2898 B]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1 all 8.1.2-Subuntu2.14 [9158 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libapr1-1.7.0-Subuntu0.22.04.1.amd64.deb [88.8 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1-0.92ubuntu0.22.04.1.amd64.deb [130 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1-dbd-sqlite3 amd64 8.1.2-Subuntu2.14 [1834 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php-mysql all 8.1.2 [17.4 kB]
Fetched 7483 kB in 6s (30.8 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libapr1:amd64.
(Reading database ... 65328 files and directories currently installed.)
Preparing to unpack .../00-libapr1_1.7.0-Subuntu0.22.04.1_amd64.deb ...
Unpacking libapr1:amd64 (1.7.0-Subuntu0.22.04.1) ...
Selecting previously unselected package libaprutil1:amd64.
Preparing to unpack .../01-libaprutil1_1.6.1-Subuntu4.22.04.2_amd64.deb ...
Unpacking libaprutil1:amd64 (1.6.1-Subuntu4.22.04.2) ...
Selecting previously unselected package libaprutil1-dbd-sqlite3:amd64.
Preparing to unpack .../02-libaprutil1-dbd-sqlite3_amd64 ...

```

```

...ject_Qwikskills -- ubuntu@ip-175-1-7-39: ~ ssh -i ubuntuKeyPair.pem ubuntu@ec2-54-237-112-77.compute-1.amazonaws.com
Preparing to unpack .../01-libaprutil1_1.6.1-Subuntu4.22.04.2_amd64.deb ...
Unpacking libaprutil1:amd64 (1.6.1-Subuntu4.22.04.2) ...
Selecting previously unselected package libaprutil1-openssl:amd64.
Preparing to unpack .../02-libaprutil1-dbd-sqlite3:amd64 ...
Unpacking libaprutil1-dbd-sqlite3:amd64 (1.6.1-Subuntu4.22.04.2) ...
Selecting previously unselected package libaprutil1-ldap:amd64.
Preparing to unpack .../03-libaprutil1-ldap_1.6.1-Subuntu4.22.04.2_amd64.deb ...
Unpacking libaprutil1-ldap:amd64 (1.6.1-Subuntu4.22.04.2) ...
Selecting previously unselected package libaprutil1-openssl3:amd64.
Preparing to unpack .../04-liblubus5.3-0.5_3.6-Subuntu4.22.04.2_amd64.deb ...
Unpacking liblubus5.3-0.5_amd64 (5.3.6-Subuntu4.22.04.2) ...
Selecting previously unselected package apache2-bin.
Preparing to unpack .../05-apache2-bin_2.4.52-Subuntu4.7_amd64.deb ...
Unpacking apache2-bin (2.4.52-Subuntu4.7) ...
Selecting previously unselected package apache2-data.
Preparing to unpack .../06-apache2-data_2.4.52-Subuntu4.7_all.deb ...
Unpacking apache2-data (2.4.52-Subuntu4.7) ...
Selecting previously unselected package apache2-utils.
Preparing to unpack .../07-apache2-utils_2.4.52-Subuntu4.7_amd64.deb ...
Unpacking apache2-utils (2.4.52-Subuntu4.7) ...
Selecting previously unselected package mailcap.
Preparing to unpack .../08-mailcap_3.70+multiubuntu1 ...
Unpacking mailcap (3.70+multiubuntu1) ...
Selecting previously unselected package mime-support.
Preparing to unpack .../09-mime-support_3.66_all.deb ...
Unpacking mime-support (3.66) ...
Selecting previously unselected package apache2.
Preparing to unpack .../10-apache2_2.4.52-Subuntu4.7_amd64.deb ...
Unpacking apache2 (2.4.52-Subuntu4.7) ...
Selecting previously unselected package bzip2.
Preparing to unpack .../11-bzip2_1.0.6-Subuntu4.6-Sbuild1_amd64.deb ...
Unpacking bzip2 (1.0.6-Subuntu4.6-Sbuild1) ...
Selecting previously unselected package php-common.
Preparing to unpack .../12-php-common_2.83a912Subuntu1_all.deb ...
Unpacking php-common (2.83a912Subuntu1) ...
Selecting previously unselected package php8.1-common.
Preparing to unpack .../13-php8.1-common_8.1.2-Subuntu2.14_amd64.deb ...
Unpacking php8.1-common (8.1.2-Subuntu2.14) ...
Selecting previously unselected package libapache2-mod-php8.1.
Preparing to unpack .../14-php8.1-ocache_8.1.2-Subuntu2.14_amd64.deb ...
Unpacking php8.1-ocache (8.1.2-Subuntu2.14) ...
Selecting previously unselected package php8.1-readline.
Preparing to unpack .../15-php8.1-readline_8.1.2-Subuntu2.14_amd64.deb ...
Unpacking php8.1-readline (8.1.2-Subuntu2.14) ...
Selecting previously unselected package php8.1-clnt.
Preparing to unpack .../16-php8.1-clnt_8.1.2-Subuntu2.14_amd64.deb ...
Unpacking php8.1-clnt (8.1.2-Subuntu2.14) ...
Selecting previously unselected package libapache2-mod-php8.1.
Preparing to unpack .../17-libapache2-mod-php8.1_8.1.2-Subuntu2.14_amd64.deb ...
Unpacking libapache2-mod-php8.1 (8.1.2-Subuntu2.14) ...
Selecting previously unselected package libapache2-mod-php.
Preparing to unpack .../18-libapache2-mod-php_2.83a8.1+92Subuntu1_all.deb ...
Unpacking libapache2-mod-php (2.83a8.1+92Subuntu1) ...
Selecting previously unselected package php8.1.
Preparing to unpack .../19-php8.1_8.1.2-Subuntu2.14_all.deb ...
Unpacking php8.1 (8.1.2-Subuntu2.14) ...
Selecting previously unselected package php.
Preparing to unpack .../20-php_2.83a8.1+92Subuntu1_all.deb ...
Unpacking php (2.8.1+92Subuntu1) ...
Selecting previously unselected package php8.1-mysql.
Preparing to unpack .../21-php8.1-mysql_8.1.2-Subuntu2.14_amd64.deb ...
Unpacking php8.1-mysql (8.1.2-Subuntu2.14) ...
Selecting previously unselected package php8.1-mySQL.
Preparing to unpack .../22-php8.1-mySQL_8.1.2-Subuntu2.14_amd64.deb ...
Unpacking php8.1-mySQL (8.1.2-Subuntu2.14) ...
Selecting previously unselected package ssl-cert.
Preparing to unpack .../23-ssl-cert_1.1.2_all.deb ...

```

```

...ject_Qwikskills -- ubuntu@ip-175-1-7-39: ~ ssh -i ubuntuKeyPair.pem ubuntu@ec2-54-237-112-77.compute-1.amazonaws.com
Setting up php8.1-common (8.1.2-Subuntu2.14) ...
Creating config file /etc/php/8.1/mods-available/calendar.ini with new version
Creating config file /etc/php/8.1/mods-available/ctype.ini with new version
Creating config file /etc/php/8.1/mods-available/exif.ini with new version
Creating config file /etc/php/8.1/mods-available/fileinfo.ini with new version
Creating config file /etc/php/8.1/mods-available/ffi.ini with new version
Creating config file /etc/php/8.1/mods-available/ftp.ini with new version
Creating config file /etc/php/8.1/mods-available/gettext.ini with new version
Creating config file /etc/php/8.1/mods-available/iconv.ini with new version
Creating config file /etc/php/8.1/mods-available/pdo.ini with new version
Creating config file /etc/php/8.1/mods-available/phar.ini with new version
Creating config file /etc/php/8.1/mods-available/posix.ini with new version
Creating config file /etc/php/8.1/mods-available/shmop.ini with new version
Creating config file /etc/php/8.1/mods-available/sockets.ini with new version
Creating config file /etc/php/8.1/mods-available/sysmsg.ini with new version
Creating config file /etc/php/8.1/mods-available/sysvsem.ini with new version
Creating config file /etc/php/8.1/mods-available/sysvshm.ini with new version
Creating config file /etc/php/8.1/mods-available/tokenizer.ini with new version
Setting up libapr1:amd64 (1.6.1-Subuntu4.22.04.3) ...
Setting up bzip2 (1.0.6-Subuntu4.6-Sbuild1) ...
Setting up php8.1-mysql (8.1.2-Subuntu2.14) ...
Creating config file /etc/php/8.1/mods-available/mysqlnd.ini with new version
Creating config file /etc/php/8.1/mods-available/mysqli.ini with new version
Creating config file /etc/php/8.1/mods-available/pdo_mysql.ini with new version
Setting up ssl-cert (1.1.2) ...
Setting up liblubus5.3-0.5_amd64 (5.3.6-Subuntu4.22.04.2) ...
Setting up php8.1-readline (8.1.2-Subuntu2.14) ...
Creating config file /etc/php/8.1/mods-available/readline.ini with new version
Setting up apache2-data (2.4.52-Subuntu4.7) ...
Setting up mailcap (3.70+multiubuntu1) ...
Setting up php8.1-ocache (8.1.2-Subuntu2.14) ...
Creating config file /etc/php/8.1/mods-available/opcache.ini with new version
Setting up libaprutil1:amd64 (1.6.1-Subuntu4.22.04.2) ...
Setting up php-mysql (2.83a8.1+92Subuntu1) ...
Setting up liblubus5.3-0.5_amd64 (5.3.6-Subuntu4.22.04.2) ...
Setting up libaprutil1-ldap:amd64 (1.6.1-Subuntu4.22.04.2) ...
Setting up libaprutil1-dbd-sqlite3:amd64 (1.6.1-Subuntu4.22.04.2) ...
Setting up php8.1-clnt (8.1.2-Subuntu2.14) ...
update-alternatives: using /usr/bin/php8.1 to provide /usr/bin/php (php) in auto mode
update-alternatives: using /usr/bin/phar8.1 to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar8.1 to provide /usr/bin/phar.phar (phar.phar) in auto mode
Creating config file /etc/php/8.1/cgi/php.ini with new version
Setting up apache2-utils (2.4.52-Subuntu4.7) ...
Setting up apache2-bin (2.4.52-Subuntu4.7) ...

```

```

...ject_Qwikskills - ubuntu@ip-175-1-7-39: ~ ssh -i ubuntuKeyPair.pem ubuntu@ec2-54-237-112-77.compute-1.amazonaws.com
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module timeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serveralias-bin.
Enabling site _000-default.
Info: switch to rpm prefork for package libapache2-mod-php8.1
Module mod_rpm_premod loaded.
Enabling module app-prework.
Info: Executing deferred module 'a2enmod php8.1' for package libapache2-mod-php8.1
Enabling module php8.1.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Setting up libapache2-mod-php (2:8.1+92ubuntu1) ...
Setting up libapache2-mod-php (2:8.1+92ubuntu1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.3) ...
Processing triggers for php8.1-cll (8.1.2-1ubuntu2.14) ...
Processing triggers for libapache2-mod-php8.1 (8.1.2-1ubuntu2.14) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-175-1-7-39:~$ sudo tee /var/www/html/index.php >/dev/null <>EOF
<?php
$conn = new mysqli($host, "$username", "$password", "$database");
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT * FROM stable";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    echo "<table><thead><tr><th>Name</th></tr></thead><tbody>";
    while($row = $result->fetch_assoc()) {
        echo "<tr><td>" . $row["id"] . "</td><td>" . $row["name"] . "</td></tr>";
    }
    echo "</tbody></table>";
} else {
    echo "#0 results";
}
$conn->close();
?
EOF
ubuntu@ip-175-1-7-39:~$ sudo sed -i "s/index.html/index.php/g" /etc/apache2/mods-enabled/dir.conf
ubuntu@ip-175-1-7-39:~$ echo "Web application is now accessible at: http://$(curl -s http://checkip.amazonaws.com)/"
Web application is now accessible at: http://142.237.111.77/
ubuntu@ip-175-1-7-39:~$ 

```

us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#logsV2:log-groups/log-group/RDSOSMetrics/log-events/db-NEOAYY4I7SF6SEWNARJUJSIFDM

**Log events**

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Timestamp	Message
2023-11-24T17:00:42.000+05:30	<pre>{   "engine": "MySQL",   "instanceID": "mysqlDb",   "instanceResourceID": "db-NEOAYY4I7SF6SEWNARJUJSIFDM",   "version": "8.0.32",   "version": "8.0.32",   "uptime": "02:15:25",   "runCPU": 2,   "cpuUtilization": {     "guest": 0,     "avg": 0,     "sys": 1.2,     "net": 0.2,     "idle": 96.4,     "user": 2.1,     "total": 2.6,     "steal": 0,     "nice": 0   },   "loadAverage5Minute": {     "one": 0.31,     "five": 0.18,     "fifteen": 0.06   },   "memory": {     "writeback": 0,     "hugePagesUsed": 0,     "hugePagesTotal": 0,     "hugePagesSurp": 0,     "cached": 804648,     "hugePageSize": 2048,     "free": 89280,     "hugePagesTotal": 0,     "inactive": 1872244,     "pageTables": 7412,     "dirty": 948   } }</pre>

Actions ▾ Start tailing Create metric filter

Clear 1m 30m 1h 12h Custom Local timezone Display

Copy

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Launch Template

The screenshot shows the AWS EC2 Load Balancers console. On the left, a navigation sidebar lists various services like EC2 Dashboard, Instances, and Load Balancing. The main area displays a table titled "Load balancers (1/1)" with one entry: "myLT-1". The table includes columns for Name, DNS name, State, VPC ID, Availability Zones, Type, and Date created. The "myLT-1" row is selected, showing its details in a modal window. The modal window has tabs for Details, Listeners and rules, Network mapping, Security, Monitoring, Integrations, Attributes, and Tags. The "Details" tab is selected, showing information such as Load balancer type (Application), Status (Active), VPC (vpc-0c903feef5a8edd5b), IP address type (IPv4), Scheme (Internet-facing), Hosted zone (Z35SXDOTRQ7X7K), and Availability Zones (us-east-1a, us-east-1b). The "DNS name" field contains "myLT-1-1320066603.us-east-1.amazonaws.com". The bottom of the screen shows the macOS dock with various application icons.

# Auto Scaling Group

The screenshot shows the AWS Auto Scaling Groups console. On the left, a navigation sidebar lists various services like EC2 Dashboard, Instances, and Auto Scaling. The main area displays a table titled "Auto Scaling groups (1/1)" with one entry: "myLT". The table includes columns for Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Availability Zones. The "myLT" row is selected, showing its details in a modal window. The modal window has tabs for Details, Activity, Automatic scaling, Instance management, Monitoring, and Instance refresh. The "Details" tab is selected, showing information such as Auto Scaling group name (myLT), Desired capacity (2), Desired capacity type (Units (number of instances)), Amazon Resource Name (ARN) (arn:aws:autoscaling:us-east-1:005238736607:autoScalingGroup:2286e3e2-dee5-45a8-9f1-39a3b182ea9:autoScalingGroupName/myLT), Date created (Fri Nov 24 2023 14:55:25 GMT+0530 (India Standard Time)), Minimum capacity (2), Maximum capacity (2), and Status (-). The bottom of the screen shows the macOS dock with various application icons.

# Elastic IPs

The screenshot shows the AWS Management Console for the EC2 service. The left sidebar navigation includes: AWS Dashboard, Services (EFS, EC2), Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, **Elastic IPs** (selected), Placement Groups, Key Pairs, Network Interfaces, Load Balancing, and Load Balancers.

The main content area displays a table titled "Elastic IP addresses (2)". The table columns are: Name, Allocated IPv4 address, Type, Allocation ID, Reverse DNS record, Associated instance ID, and Private IP address. The data shows:

Name	Allocated IPv4 address	Type	Allocation ID	Reverse DNS record	Associated instance ID	Private IP address
-	44.194.240.82	Public IP	eipalloc-0f0ee438daaa84f822	-	i-08a1f9bd962608199	175.1.26.3
-	54.237.112.77	Public IP	eipalloc-0a20d1146a1412f89	-	i-03a99d93c696f3e4c	175.1.7.39

At the bottom of the main content area, there is a callout box with the text: "View IP address usage and recommendations to release unused IPs with Public IP insights".

The bottom navigation bar includes CloudShell, Feedback, and various AWS service icons. The footer contains the text: "© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

# Load Balancer

The screenshot shows the AWS Management Console for the EC2 service. The left sidebar navigation includes: AWS Dashboard, Services (EFS, EC2), Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, **Elastic IPs**, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, and **Load Balancers** (selected).

The main content area displays a table titled "Load balancers (1/1)". The table columns are: Name, DNS name, State, VPC ID, Availability Zones, Type, and Date created. The data shows:

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
myLT-1	myLT-1-1320066603.us-east-1.amazonaws.com	Active	vpc-0c903feef5a8edd5b	2 Availability Zones	application	November 24, 2023, 14:55 (UTC+05:30)

The bottom navigation bar includes CloudShell, Feedback, and various AWS service icons. The footer contains the text: "© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

## DNS result, and result from both EC2 Instances with their public IP



The image consists of three vertically stacked screenshots of a Mac OS X desktop environment. Each screenshot shows a single browser window open to a web page. The browser's address bar indicates the URL is "Not Secure | mylt-1-1320066603.us-east-1.elb.amazonaws.com" for the top two screenshots, and "Not Secure | 54.237.112.77" for the bottom one. The browser interface includes standard controls like back, forward, and search, along with a toolbar and a menu bar at the top. Below the browser is the Mac OS X Dock, which contains icons for various applications such as Finder, Mail, Safari, and others. The desktop background is a solid light color.

**ID Name**

- 1 Ross
- 2 Chandler
- 3 Rachel
- 4 Phoebe
- 5 Joe

# CloudTrail

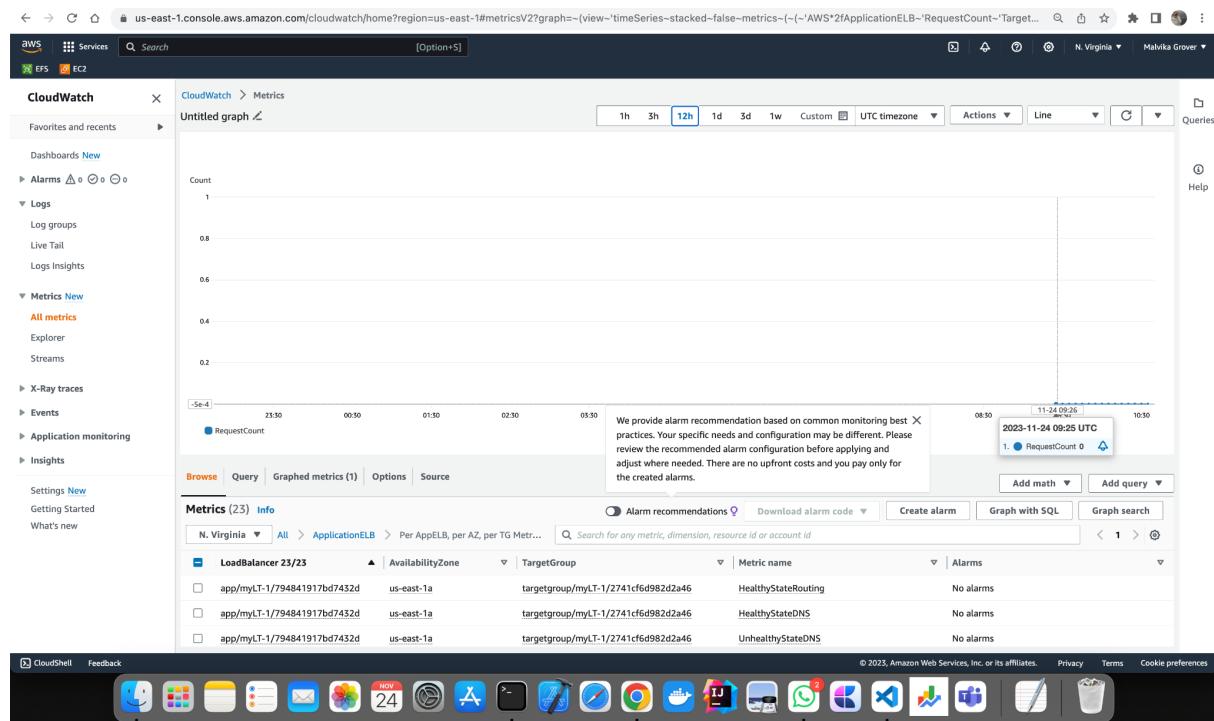
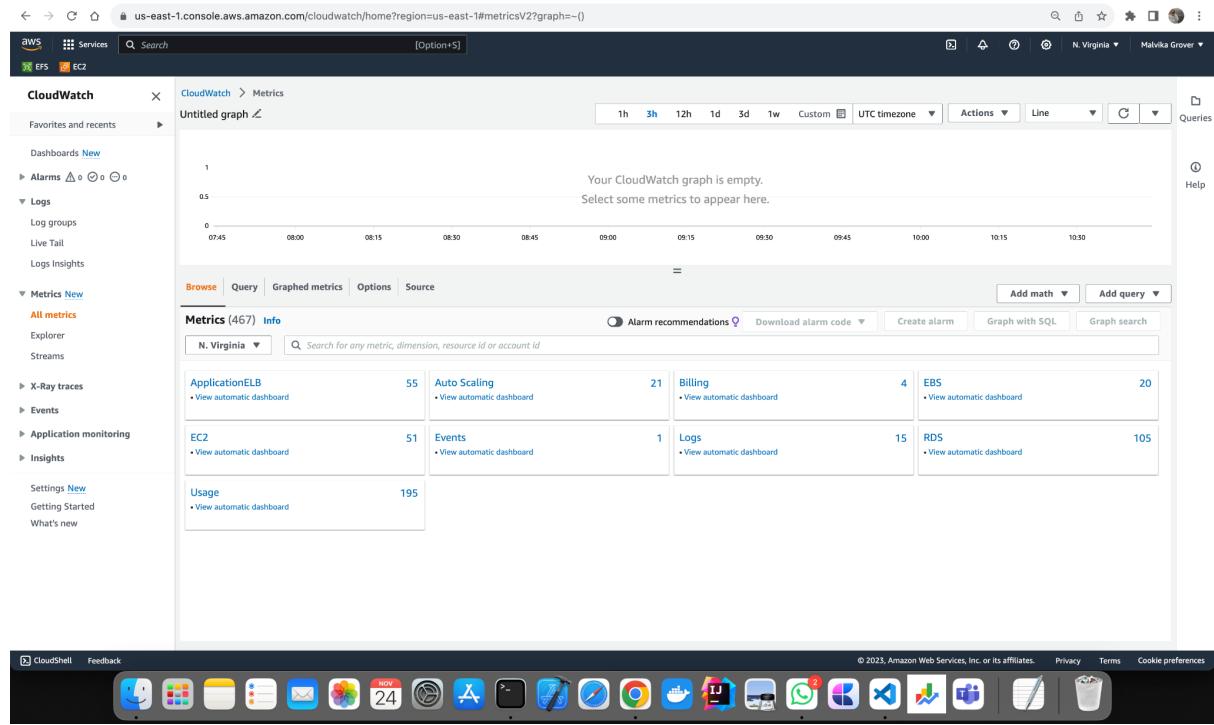
The screenshot shows the AWS CloudTrail Event history page. The left sidebar includes options like Dashboard, Event history (selected), Insights, Lake, Settings, Pricing, Documentation, Forums, and FAQs. The main content area displays a table of event history with columns: Event name, Event time, User name, Event source, Resource type, and Resource name. The table lists various CloudTrail actions such as TerminateInstances, CreateLogStream, UpdateTrail, and PutRule, along with their respective timestamps and resource details. A search bar and filter button are at the top of the table.

The screenshot shows the AWS CloudTrail Trails page. The left sidebar includes options like Dashboard, Event history, Insights, Lake (selected), Settings, Pricing, Documentation, Forums, and FAQs. The main content area shows the configuration for a trail named 'CatchLogsTrail001'. It includes sections for General details (with fields like Trail log location, Log file validation, and SNS notification delivery), CloudWatch Logs (with a log group and IAM role), and Tags (with a 'Manage tags' button). The browser toolbar at the bottom includes CloudShell, Feedback, and several application icons.

The screenshot shows the AWS CloudTrail console interface. On the left, there's a sidebar with navigation links like Dashboard, Event history, Insights, Lake (Dashboard, Query, Event data stores, Integrations), Trails (selected), Settings, Pricing, Documentation, Forums, and FAQs. The main content area has tabs for Log group (aws-cloudtrail-logs-005238736607-3ea44893) and IAM Role (arn:aws:iam::005238736607:role/service-role/CloudTrailRole). Below these are sections for Tags, Management events (not configured), Data events (CloudTrail log selector template: Log all events, Selector name: --, All events), and Insights events (disabled). A note at the bottom says "You can only enable CloudTrail Insights on trails that log management events." At the bottom of the page, there's a standard AWS footer with links for CloudShell, Feedback, and various AWS services.

## CloudWatch

The screenshot shows the AWS CloudWatch console interface. The left sidebar includes Favorites and recent items, Dashboards (New), Alarms, Logs (Log groups selected), Metrics (New), X-Ray traces, Events, Application monitoring, and Insights. Under Logs, the Log groups section is active, showing RDSMetrics and db-NEOAYY417SF6SEWNARJIUSIFDM. The main content area displays the Log events page for db-NEOAYY417SF6SEWNARJIUSIFDM. It features a filter bar with a search input, timestamp range (Clear, 1m, 30m, 1h, 12h, Custom, Local timezone), and display options (Display, Create metric filter). Below the filter bar is a table with columns for Timestamp and Message. The table lists numerous log entries from November 2023, mostly from MySQL instances (engine: "MySQL", instanceID: "mysql..."). Each entry includes details like timestamp, engine, instance ID, and uptime. The table ends with a note: "There are older events to load. Load more." At the bottom of the page, there's a standard AWS footer with links for CloudShell, Feedback, and various AWS services.



# CloudWatch Alert

The screenshot shows the AWS CloudWatch Alarms page. A success message at the top says "Successfully created alarm EC2INALARM." The main table lists one alarm:

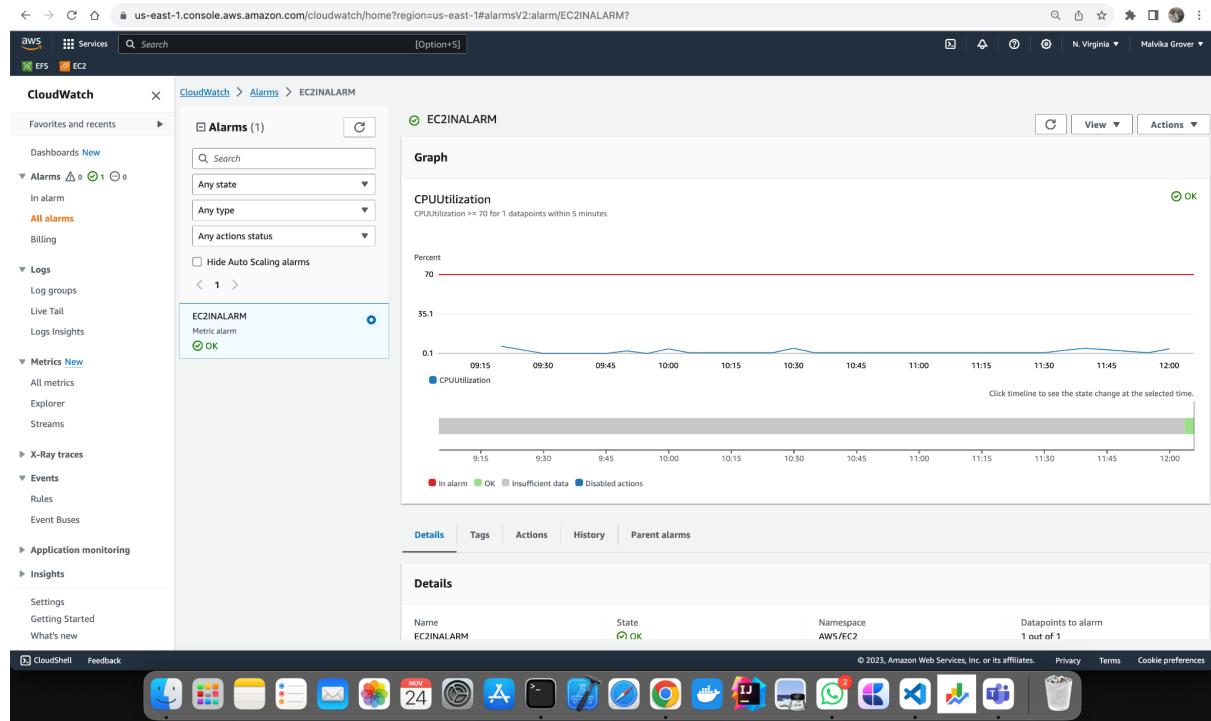
Name	State	Last state update	Conditions	Actions
EC2INALARM	Insufficient data	2023-11-24 12:02:52	CPUUtilization >= 70 for 1 datapoints within 5 minutes	Actions enabled

The left sidebar includes sections for Dashboards, Alarms (with a red alert icon), Logs, Metrics, X-Ray traces, Events, Application monitoring, and Insights.

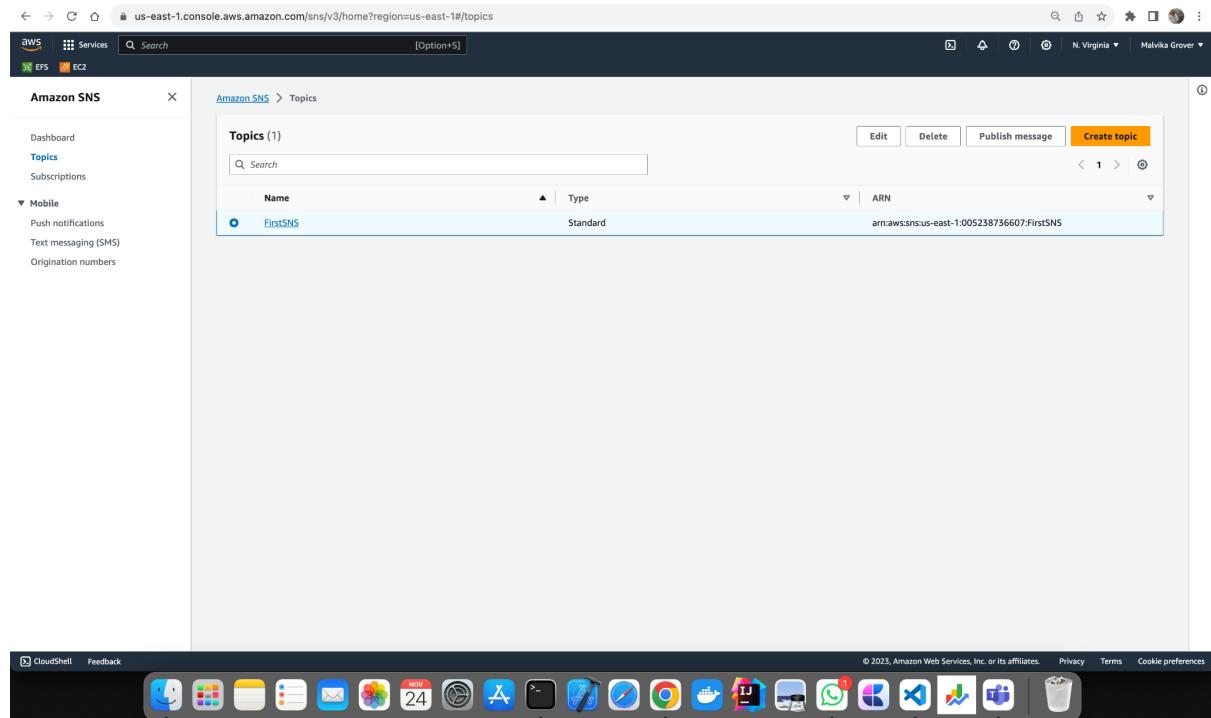
The screenshot shows the AWS CloudWatch Alarms page again. The success message is still present. The main table now shows the alarm in the "OK" state:

Name	State	Last state update	Conditions	Actions
EC2INALARM	OK	2023-11-24 12:03:41	CPUUtilization >= 70 for 1 datapoints within 5 minutes	Actions enabled

The left sidebar remains the same as the first screenshot.



## Email for SNS subscription



The screenshot shows the AWS SNS console with a topic named "FirstSNS". The topic details include:

- Name: FirstSNS
- Display name: FirstSNS
- ARN: arn:aws:sns:us-east-1:005238736607:FirstSNS
- Type: Standard

The "Subscriptions" tab is selected, showing one subscription:

ID	Endpoint	Status	Protocol
a03d2805-0895-4d1a-a963-0a550f66cc3	malvikagrover22@gmail.com	Confirmed	EMAIL

The screenshot shows an email inbox with an incoming message titled "AWS Notification - Subscription Confirmation". The message content is:

You have chosen to subscribe to the topic:  
arn:aws:sns:us-east-1:005238736607:FirstSNS

To confirm this subscription, click or visit the link below (if this was in error no action is necessary):  
[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

## SNS Notification on Email

The screenshot shows an email inbox with four notifications from 'FirstSNS'. Each notification is a JSON message:

- First message (4:52 PM):

```
{"version": "0", "id": "3c6b23a8-e689-70d5-4687-ab0c45cd8f0e", "detail-type": "EC2 Instance State-change Notification", "source": "aws.ec2", "account": "005238736607", "time": "2023-11-24T11:22:42Z", "region": "us-east-1", "resources": ["arn:aws:ec2:us-east-1:005238736607:instance/i-062b1edb93cdca643"], "detail": {"instance-id": "i-062b1edb93cdca643", "state": "shutting-down"}}
```
- Second message (4:53 PM):

```
{"version": "0", "id": "7dd53379-8e0e-9281-80a5-5e7687da5753", "detail-type": "EBS Volume Notification", "source": "aws.ec2", "account": "005238736607", "time": "2023-11-24T11:23:12Z", "region": "us-east-1", "resources": ["arn:aws:ec2:us-east-1:005238736607:volume/vol-0505364066e16e9c"], "detail": {"result": "deleted", "cause": "", "event": "deleteVolume", "request-id": "7488abc2-3372-4bf9-a7d7-337b01731fa0"}}
```
- Third message (4:53 PM):

```
{"version": "0", "id": "4ed2eb59-b952-bad8-7995-012c11626a69", "detail-type": "EC2 Instance State-change Notification", "source": "aws.ec2", "account": "005238736607", "time": "2023-11-24T11:23:12Z", "region": "us-east-1", "resources": ["arn:aws:ec2:us-east-1:005238736607:instance/i-062b1edb93cdca643"], "detail": {"instance-id": "i-062b1edb93cdca643", "state": "terminated"}}
```
- Fourth message (4:53 PM):

```
{"version": "0", "id": "2eaecc08-5814-043a-aa57-546bbf1335e9", "detail-type": "EC2 Instance State-change Notification", "source": "aws.ec2", "account": "005238736607", "time": "2023-11-24T11:23:31Z", "region": "us-east-1", "resources": ["arn:aws:ec2:us-east-1:005238736607:instance/i-0854d2fe84f7af9df"], "detail": {"instance-id": "i-0854d2fe84f7af9df", "state": "pending"}}
```

# S3 Bucket

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with navigation links like 'Buckets', 'Storage Lens', 'Feature spotlight', and 'AWS Marketplace for S3'. The main area is titled 'Objects (17)' and displays a table of objects. The table has columns for Name, Type, Last modified, Size, and Storage class. Most entries are folders named after AWS regions (ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-southeast-1, ap-southeast-2, ca-central-1, eu-central-1, eu-north-1, eu-west-1, eu-west-2, eu-west-3, sa-east-1, us-east-1, us-east-2, us-west-1, us-west-2). One object, '005238736607\_CloudTrail-Digest\_us-east-1\_CatchLogsTrail001\_us-east-1\_20231124T111043Z.json.gz', is listed as a file type (gz) from November 24, 2023, at 17:12:15 UTC+05:30, with a size of 349.0 B and Standard storage class. At the bottom, there's a toolbar with buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload.

This screenshot shows a detailed view of a single object within a bucket. The path in the top navigation bar is: Amazon S3 > Buckets > aws-cloudtrail-logs-005238736607-ebc605cc > AWSLogs/ > 005238736607/ > CloudTrail-Digest/ > us-east-1/ > 2023/ > 11/ > 24/. The object name is '24/'. Below the object name, there are tabs for 'Objects' and 'Properties'. The 'Objects' tab is selected, showing a table with one item. The table columns are Name, Type, Last modified, Size, and Storage class. The item is '005238736607\_CloudTrail-Digest\_us-east-1\_CatchLogsTrail001\_us-east-1\_20231124T111043Z.json.gz', which is a gz file from November 24, 2023, at 17:12:15 UTC+05:30, with a size of 349.0 B and Standard storage class. A 'Copy S3 URI' button is located at the top right of this section. The bottom of the screen shows a Mac OS X dock with various application icons.

Screenshot of the AWS S3 console showing the 'Buckets' page. The left sidebar shows navigation options like 'Access Points', 'Storage Lens', and 'AWS Marketplace'. The main area displays an 'Account snapshot' with storage usage trends and a table of buckets. The table includes columns for Name, AWS Region, Access, and Creation date.

Name	AWS Region	Access	Creation date
aws-cloudtrail-logs-005238736607-ebc605cc	US East (N. Virginia) us-east-1	Bucket and objects not public	November 24, 2023, 16:35:18 (UTC+05:30)
elasticbeanstalk-ap-south-1-005238736607	Asia Pacific (Mumbai) ap-south-1	Objects can be public	April 11, 2023, 00:57:48 (UTC+05:30)
elasticbeanstalk-us-east-1-005238736607	US East (N. Virginia) us-east-1	Objects can be public	February 23, 2023, 18:23:57 (UTC+05:30)
qwikskillscloudtraillogs	US East (N. Virginia) us-east-1	Bucket and objects not public	November 24, 2023, 16:29:01 (UTC+05:30)

Screenshot of the AWS S3 object details page for the file '005238736607\_CloudTrail-Digest\_us-east-1\_CatchLogsTrail001\_us-east-1\_20231124T111043Z.json.gz'. The left sidebar is identical to the previous screenshot. The main area shows the object's properties, including its URI, ARN, Etag, and URL.

**Object overview**

- Owner: grover.malvika1
- AWS Region: US East (N. Virginia) us-east-1
- Last modified: November 24, 2023, 17:12:15 (UTC+05:30)
- Size: 349.0 B
- Type: gz
- Key: AWSLogs/005238736607/CloudTrail-Digest/us-east-1/2023/11/24/005238736607\_CloudTrail-Digest\_us-east-1\_CatchLogsTrail001\_us-east-1\_20231124T111043Z.json.gz

**Properties**

- S3 URI: s3://aws-cloudtrail-logs-005238736607-ebc605cc/AWSLogs/005238736607/CloudTrail-Digest/us-east-1/2023/11/24/005238736607\_CloudTrail-Digest\_us-east-1\_CatchLogsTrail001\_us-east-1\_20231124T111043Z.json.gz
- Amazon Resource Name (ARN): arn:aws:s3:::aws-cloudtrail-logs-005238736607-ebc605cc/AWSLogs/005238736607/CloudTrail-Digest/us-east-1/2023/11/24/005238736607\_CloudTrail-Digest\_us-east-1\_CatchLogsTrail001\_us-east-1\_20231124T111043Z.json.gz
- Entity tag (Etag): bd9577be7b5aac0b8d2c8cd278942a3d
- Object URL: https://aws-cloudtrail-logs-005238736607-ebc605cc.s3.amazonaws.com/AWSLogs/005238736607/CloudTrail-Digest/us-east-1/2023/11/24/005238736607\_CloudTrail-Digest\_us-east-1\_CatchLogsTrail001\_us-east-1\_20231124T111043Z.json.gz

# CONCLUSION

In conclusion, the development and deployment of the highly available web application on AWS represent a successful integration of advanced cloud technologies to meet the demands of today's dynamic digital landscape. By leveraging key AWS services, including Amazon EC2, Amazon RDS, Auto Scaling, Elastic Load Balancing, CloudTrail, Route 53, CloudWatch, and SNS, the project achieves a resilient and scalable architecture. The comprehensive step-by-step guide provided in the documentation ensures that the implementation is accessible and reproducible.

The implemented architecture not only addresses the immediate requirements of high availability, fault tolerance, and scalability but also sets the foundation for continued growth and adaptability. Through effective use of Auto Scaling and Load Balancing, the web application is equipped to seamlessly handle varying workloads, guaranteeing an optimal user experience. The incorporation of CloudTrail, Route 53, CloudWatch, and SNS adds proactive monitoring, logging, and notification capabilities, enhancing the overall reliability and security of the application.

# SALIENT FEATURES

## Scalability:

- Auto Scaling and Load Balancing ensure the web application can dynamically adjust to varying traffic demands, providing scalability for future growth.

## Fault Tolerance:

- Multi-AZ deployment of RDS and distribution of traffic through Load Balancing enhance fault tolerance, minimizing the impact of potential failures.

## Proactive Monitoring and Logging:

- CloudTrail and CloudWatch work together to provide detailed logs and metrics, enabling proactive measures and quick responses to potential issues.

## Reliability:

- The combination of reliable storage with Amazon S3, automated backups, and efficient notification through SNS contributes to the overall reliability of the web application.

## Ease of Reproduction:

- The step-by-step documentation provides a clear and accessible guide, making it easy for developers to reproduce the architecture and adapt it to their specific needs.

# FUTURE SCOPE

*The project lays the groundwork for future enhancements and optimizations.*

Potential avenues for future development include:

- Advanced Scaling Policies: Exploring and implementing more sophisticated scaling policies based on custom metrics, allowing for even more granular control over resource provisioning.
- Security Enhancements: Introducing additional security measures, such as AWS WAF (Web Application Firewall) or AWS Shield, to further safeguard the web application against potential threats.
- Containerization and Serverless Computing: Exploring the adoption of containerization services (e.g., Amazon ECS or EKS) or serverless computing with AWS Lambda for more efficient resource utilization and cost optimization.
- Global Distribution: Implementing a multi-region architecture to enhance global availability and reduce latency for users in different geographical locations.
- Continuous Integration and Deployment (CI/CD): Integrating CI/CD pipelines to automate the deployment process, ensuring faster and more reliable application updates.

# BIBLIOGRAPHY/REFERENCES

The successful implementation of this project draws upon a range of AWS documentation, online resources, and best practices. Key references include:

## AWS Documentation:

- <https://docs.aws.amazon.com/ec2/>
- <https://docs.aws.amazon.com/rds/>
- <https://docs.aws.amazon.com/autoscaling/>
- <https://docs.aws.amazon.com/elasticloadbalancing/>
- <https://docs.aws.amazon.com/cloudtrail/>
- <https://docs.aws.amazon.com/cloudwatch/>
- <https://docs.aws.amazon.com/sns/>

## Online Tutorials and Guides:

Various online tutorials, forums, and community discussions that provided insights and practical tips for implementing specific AWS services.

By referencing these sources, the project ensures alignment with AWS best practices and recommendations, contributing to the creation of a robust and reliable web application architecture.

