Group Assignment

(CT097-3-3 - CSVC)

Cloud Infrastructure and Services

APD3F2411IT(ISS), APU3F2411CS(CYB), APD3F2411CS(CYB)

Group 50

Lecturer:	**:**	Nur Azyyati Binti Ahmad
Hand in date:	**:**	6th April 2025
Weightage:	**:**	50%

Group 50	**Name:**	**TP. No:**
Student 1 (Leader)	Abdulrahman Gamil Mohammed Ahmed	TP071012
Student 2	Abdurahman Gamil Murshed Humadi	TP070609
Student 3	Yazen Abobakr Ahmed Al-mehdhar	TP069210
Student 4	Haziq Irfan Radzali	TP072306

Table of Contents

Table of Figures

Task 1: Designing and deployment of AWS Cloud

Section A

1.0 Cloud Architecture Solution

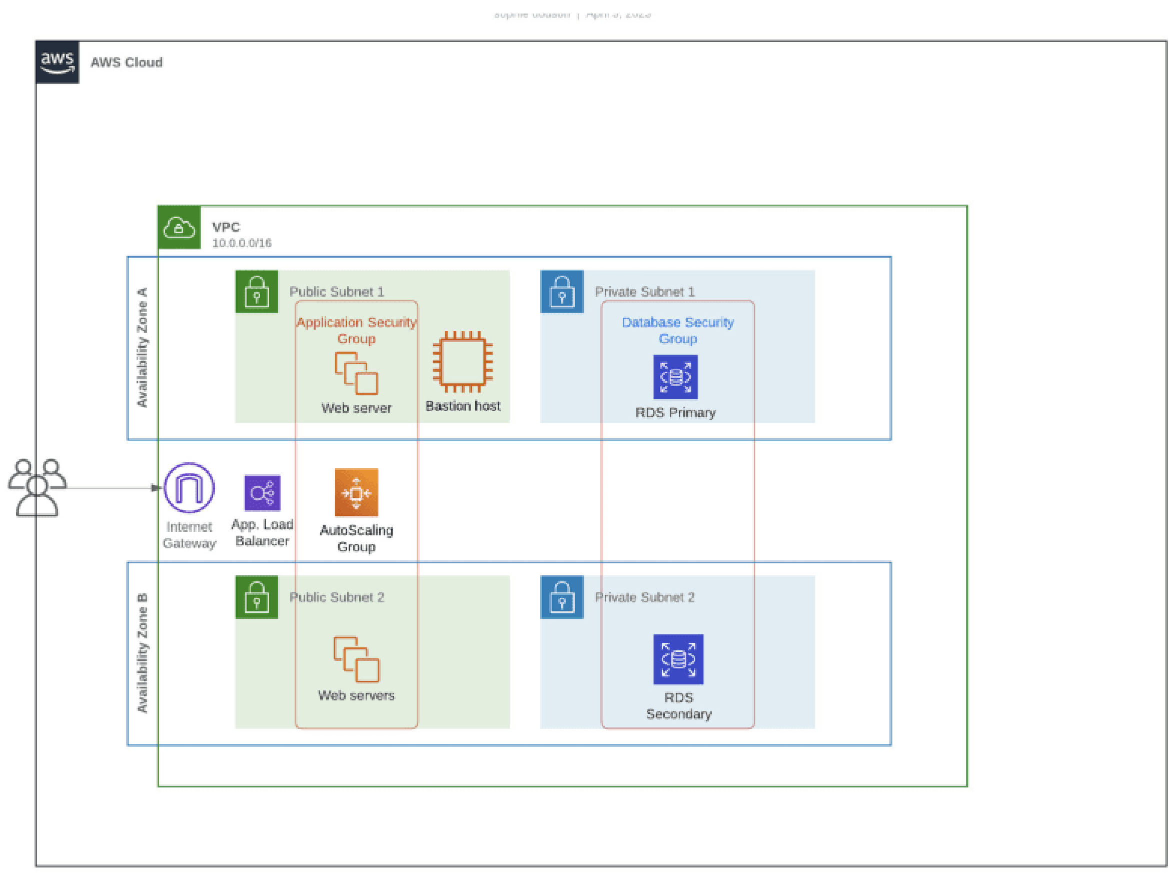
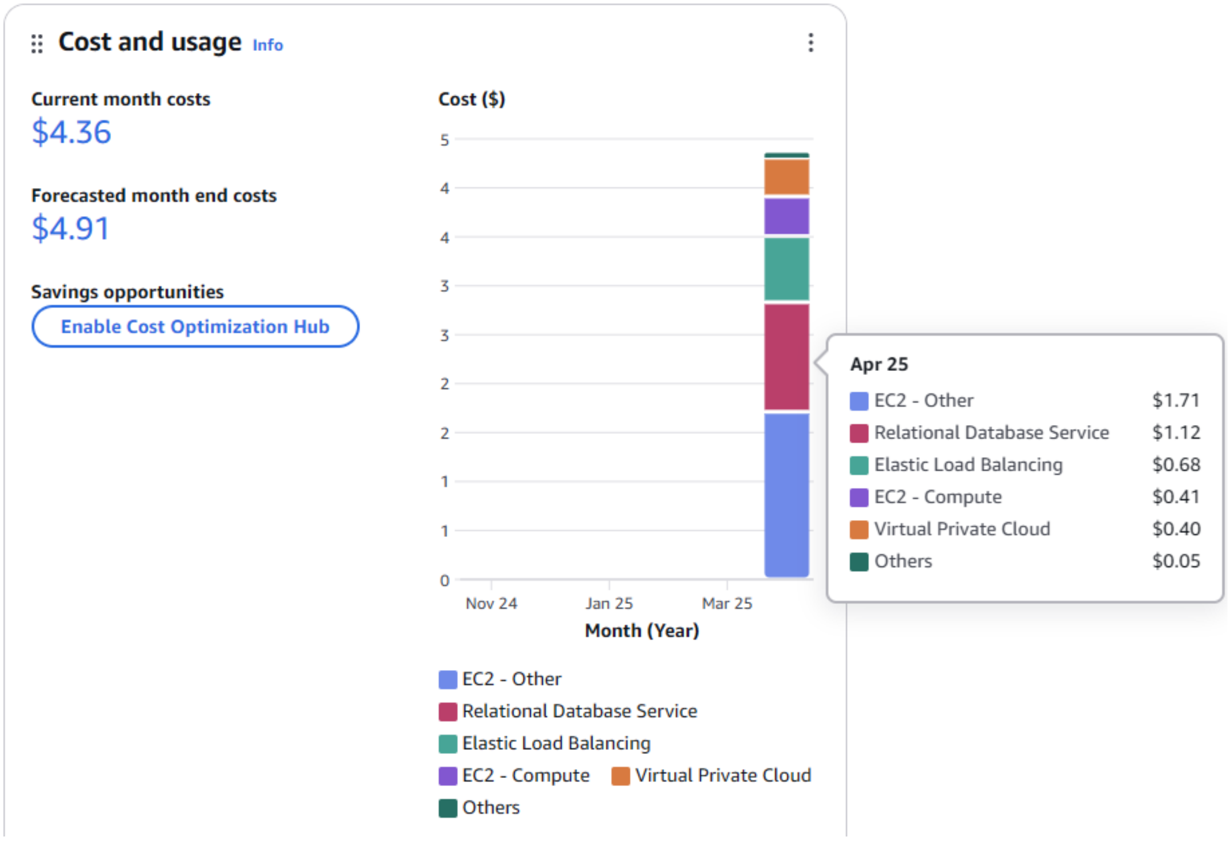


Figure 1: Cloud Architecture Solution

The AWS architecture diagram shows a high-level design of the Capstone Project infrastructure in the AWS Cloud, laid out logically in two availability zones Zone A and Zone B in a Virtual Private Cloud (VPC) to achieve high availability and fault tolerance. In Availability Zone A, Public Subnet 1 contains web servers, protected by an Application Security Group, and Private Subnet 1 has the primary Relational Database Service (RDS) instance. Availability Zone B duplicates this pattern, with Public Subnet 2 containing additional web servers and Private Subnet 2 offering a second RDS instance for redundancy. An Internet Gateway offers external connectivity, while an Application Load Balancer and AutoScaling Group work together to offer efficient traffic distribution and dynamic scaling, resulting in a secure, resilient, and highly optimized system design.

2.0 Pricing Cost



Current month costs
$4.36

Forecasted month end costs
$4.91

Apr 25

🟦 EC2 - Other	$1.71
🟥 Relational Database Service	$1.12
🟩 Elastic Load Balancing	$0.68
🟪 EC2 - Compute	$0.41
🟧 Virtual Private Cloud	$0.40
⬛ Others	$0.05

Figure 1: Cost and Usage

The graph above provides a detailed graphical **breakdown of the cost and usage** trends of the project, showing a current monthly expenditure of **$4.36** and an end-of-month estimated expenditure of **$4.91**, with data spanning from commencement to a peak in **April 2025** at **$4.36**, along with a detailed breakdown indicating EC2 – Other as the primary contributor at **$1.71**, followed by Relational Database Service at **$1.12**, Elastic Load Balancing at **$0.68**, EC2 Compute at **$0.41**, Virtual Private Cloud at **$0.40**, and Others at **$0.05**, along with an "Enable Cost Optimization Hub" option to explore potential cost-saving avenues.

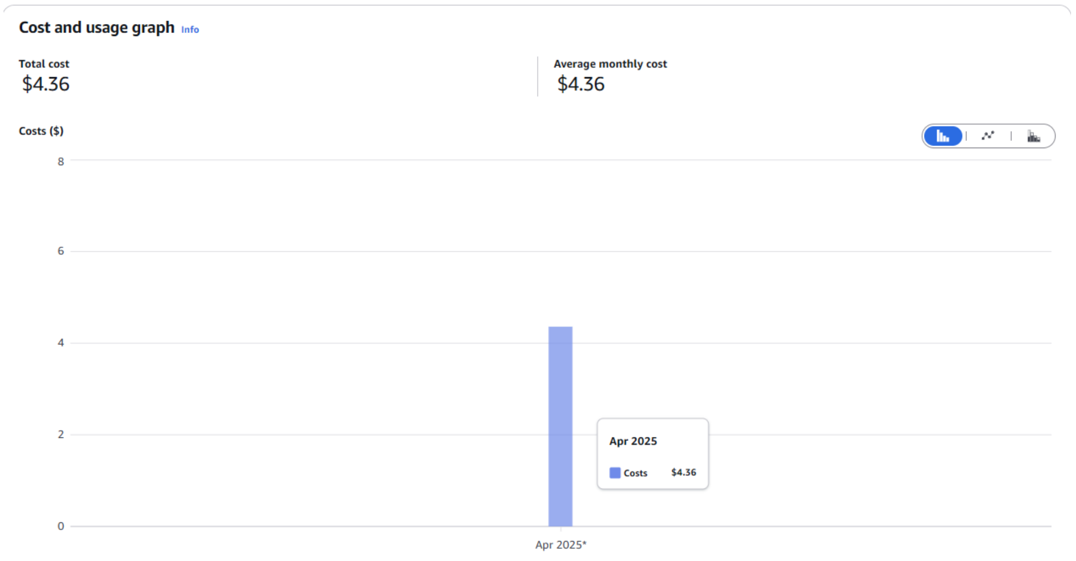


Figure 2: Cost and Usage Graph 1

The chart presents a quick overview of the cost and expenditure of the project with a spent amount of **$4.36** and an average spend per month of **$4.36**, and a stark surge is apparent in **April 2025** wherein the expense measures up to **$4.36** after a minimum of spend during the previous months, very noticeably indicative of an unmistakable sharp spike in spend activity through a tidy and readable graphical representation of the spend activity during the span.

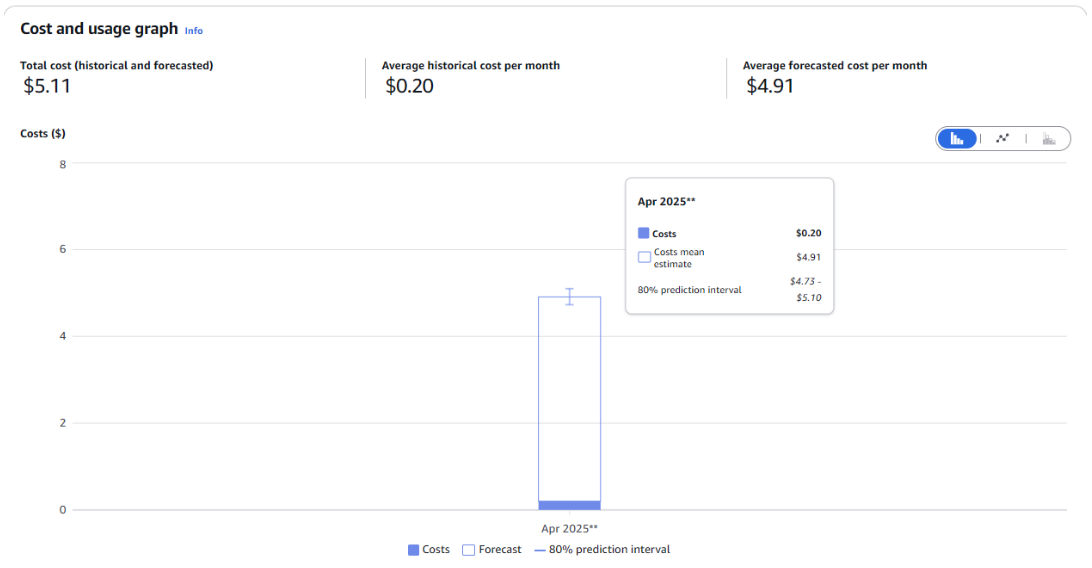
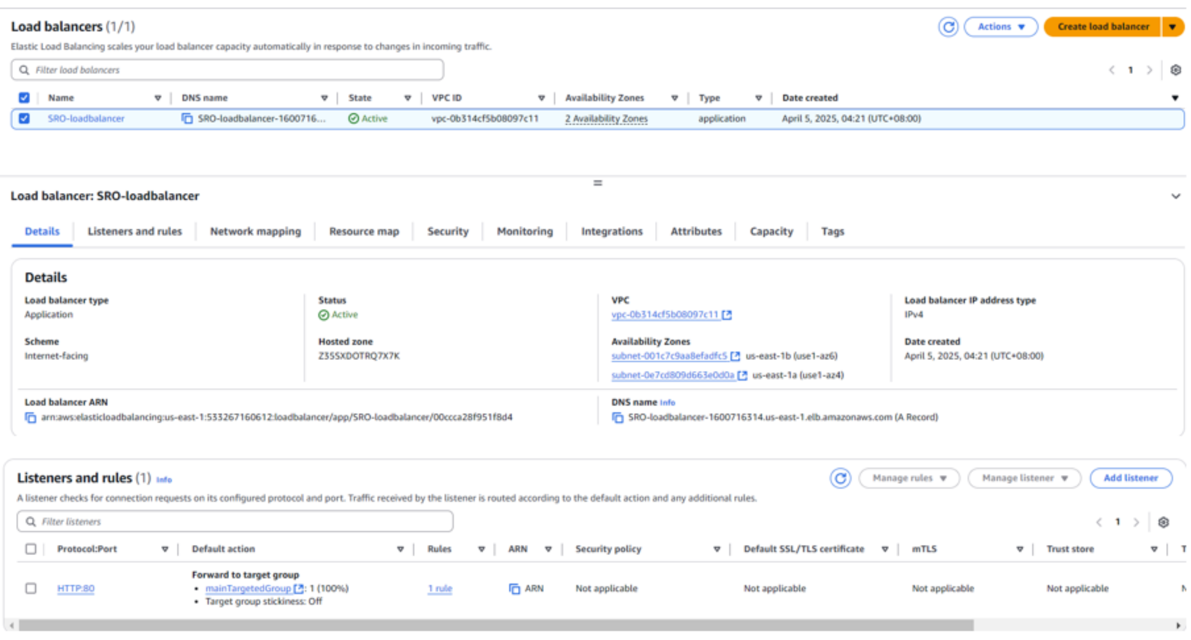


Figure 3: Cost and Usage Graph 2

The graph above indicates the trend of cost and usage of the project, having a total historical spend of $5.11 and average historical monthly cost of **$0.20**, as against an estimated average monthly cost of **$4.91** for **April 2025**, with an **80%** prediction interval ranging from **$4.75** to **$5.10,** illustrating a sudden increase in costs through a sudden spike in **April 2025** indicating a shift from insignificant past expenditures to increased future financial needs.

3.0 Cloud Environment

3.1 Capstone Project / AWS Learner Lab



Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
SRO-loadbalancer	SRO-loadbalancer-1600716...	Active	vpc-0b314cf5b08097c11	2 Availability Zones	application	April 5, 2025, 04:21 (UTC+08:00)

Load balancer: SRO-loadbalancer

Details | Listeners and rules | Network mapping | Resource map | Security | Monitoring | Integrations | Attributes | Capacity | Tags

Details

Load balancer type: Application
Scheme: Internet-facing
Status: Active
Hosted zone: Z35SXDOTRQ7X7K
VPC: vpc-0b314cf5b08097c11
Availability Zones: subnet-001c7c9aa8efadfc5 us-east-1b (use1-az6); subnet-0e7cd809d663e0d0a us-east-1a (use1-az4)
Load balancer IP address type: IPv4
Date created: April 5, 2025, 04:21 (UTC+08:00)
Load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:533267160612:loadbalancer/app/SRO-loadbalancer/00ccca28f951f8d4
DNS name Info: SRO-loadbalancer-1600716314.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust store
HTTP:80	Forward to target group • mainTargetedGroup: 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not applicable	Not applicable

Figure 4: Load Balancer State

The diagram above clearly illustrates the project operating at its peak level, with an ”active” status clearly labeled in the Capstone interface. This informative webpage accurately explains a huge amount of valuable information, such as the type of load balancer, scheme, running status, hosted zone, virtual private cloud (VPC), specified availability zones, and fully qualified DNS name, offering a complete and accurate description of the system’s smooth and efficient operation.

3.2 How the Team Worked Together to Complete the Project

As regards collaboration among the group members, the group utilized Microsoft Teams effectively to hold formal group sessions complemented with face-to-face meetings to advance the project. This combination approach greatly succeeded, encouraging frequent communication and maintaining momentum irrespective of the varying work schedule among group members. Microsoft Teams served as an effective tool for virtual meetings, providing screen-sharing capabilities for detailed technical presentations and enabling orderly conversations with well-coordinated task allocation. This virtual infrastructure provided a robust platform for seamless remote collaboration.

In contrast, the face-to-face sessions offered a different dynamic, reserved primarily for intense planning sessions. These sessions allowed the team to communicate face-to-face, generate solutions through brainstorming, resolve problems in the moment, and track recorded progress with hands-on examination. These sessions created an environment conducive to collaborative substantive problem-solving and strategic intent.

Through the combination of these two approaches, the team guaranteed complete involvement of all members, with any arising problems immediately identified and addressed. This two-framework not only allowed for individual working styles whether virtual or physical but also improved general group coordination and responsibility. Therefore, the project followed set schedules, producing outcomes with accuracy and effectiveness.

3.3 Web Services Deployed

3.3.1 Elastic Load Balancing (ELB)

Amazon Elastic Load Balancing (ELB) is a managed service that automatically routes incoming application traffic to multiple Availability Zones to one or more targets such as EC2 instances, containers, and IP addresses. ELB improves application performance, fault tolerance, and smooth user experience under varying levels of traffic.

One of the strongest aspects of ELB is the **Target Group**, which is an ordered collection of resources that may be caused to receive traffic on the basis of routing rules specified. Target groups offer high-granularity control over traffic routing and ensure that only healthy targets checked by ongoing health checks are requested.

ELB is also integrated with **Auto Scaling Groups**, whereby when application traffic increases or decreases, the instances are automatically scaled. This integration enables resources to be optimized in real-time while maintaining cost savings with uniform performance.

Additional features of ELB include:

- **Cross-Zone Load Balancing**, distributed traffic evenly over all targets in multiple Availability Zones for increased fault tolerance.
- **SSL/TLS Termination**, which delegates encryption and decryption operations to the load balancer to minimize the computational load on backend servers.
- Integration with **AWS IAM** and **Security Groups**, allowing for particular control over access and traffic flow.

Together, these features make ELB a cornerstone for building **highly available, scalable, and secure** cloud-native applications.

3.3.2 Amazon EC2

Amazon Elastic Compute Cloud (EC2) is among the fundamental AWS services that provide scalable cloud virtual servers. EC2 enables customers to host various applications with the provision of instances containing vectors of computer capacity, memory, storage, and networking capacity.

To maintain it efficient and accessible, EC2 fully integrates with **Auto Scaling Groups (ASGs)**. Auto Scaling Groups monitor traffic and app demand in real time and add or delete EC2 instances as needed based on policies defined. This always maintains applications perfectly manned with the right number of resources, even during traffic spikes or changes in load.

Each **Auto Scaling Group** can be linked to one or more Target Groups, which are used along with ELB in routing traffic to the appropriate set of EC2 instances. The design supports fault tolerance and high availability, along with health-status and config-aware smart routing.

EC2 offers several instance types for specific workloads from general-purpose to compute-optimized, memory-optimized, and more so that teams can choose the best fit for their application needs. In addition, EC2 is also supported by flexible pricing options, which include:

- **On-Demand Instances** for short-term, flexible workloads,
- **Reserved Instances** for long-term, cost-effective use,
- **Spot Instances** for available capacity at discounted prices,
- **Dedicated Hosts** for workloads need isolated hardware.

For storage, EC2 supports:

- **Amazon Elastic Block Store (EBS)** for persistent block-level storage,
- **Instance Store** for temporary storage tied to the lifecycle of the instance,
- **Amazon S3** for highly durable object storage.

Security capabilities include **AWS IAM** for access management, **Security Groups** for instance-level firewall rules, and **Network ACLs** for subnet-level control. In addition, **Elastic IP addresses** supply persistent public IP addressing even in case of instance restart or failure.

With global presence in multiple **AWS Regions and Availability Zones**, EC2 offers versatility, robustness, and elasticity required to run everything from small web applications to complex enterprise systems as well as computationally intensive applications like machine learning, analytics, and dev environments.

3.3.3 Amazon RDS

Amazon Relational Database Service (RDS) is an AWS managed cloud database service. It simplifies setting up, operating, and scaling relational databases by taking away administrative work such as provisioning, backup, patching, and monitoring. It results in less administrative burden, allowing teams to focus more time on application logic and development.

For this project, we will use MySQL as our relational database engine in Amazon RDS. **MySQL** is an open-source, popular database that is well known for its reliability, efficiency, and rich ecosystem. Running MySQL on RDS allows us to maintain all the benefit of the MySQL engine while taking advantage of AWS's automation, scalability, and security capabilities.

RDS also handles **automatic backups**, which are retained as per a configurable retention window, providing in-built data recovery in the case of unexpected failures. We can even **manually take snapshots** at will for long-term backup or versioned states of the database. Additionally, RDS automatically applies security patches and software updates within scheduled **maintenance** windows, ensuring our database environment remains up to date without direct intervention.

This is a fully managed service that supports high availability, scalability, and security, thus making Amazon RDS with MySQL a reliable and production-grade database solution in the cloud.

3.4 Project Challenges and Limitations

Throughout this project, our team encountered numerous challenges and limitations that required meticulous planning, coordination, and team problem-solving to overcome. Perhaps one of the greatest challenges was working with the complexities of cloud architecture. While AWS has a very large set of services, combining those services to offer optimal security, scalability, and performance required a great deal of technical work and team coordination.

Due to the nature of the project as a team effort, it was necessary that every member of the team was not just acquainted with AWS but also clearly knew the role and services they had been tasked with. This necessitated ongoing knowledge transfer, personal study, and an open mindset to learn in order to play our parts effectively and assist each other in resolving any architectural or configuration-based problems.

Time management was a further key challenge. Balancing the project workload on top of normal academic responsibilities consistently creates scheduling conflicts and delays in task commencement. To mitigate this, we used a formalized workflow in **Microsoft Teams** for everyday coordination, supported by **in-person meetings** for more comprehensive discussions. The hybrid communication strategy promoted consistency within the team and allowed us to maintain momentum despite the time constraints.

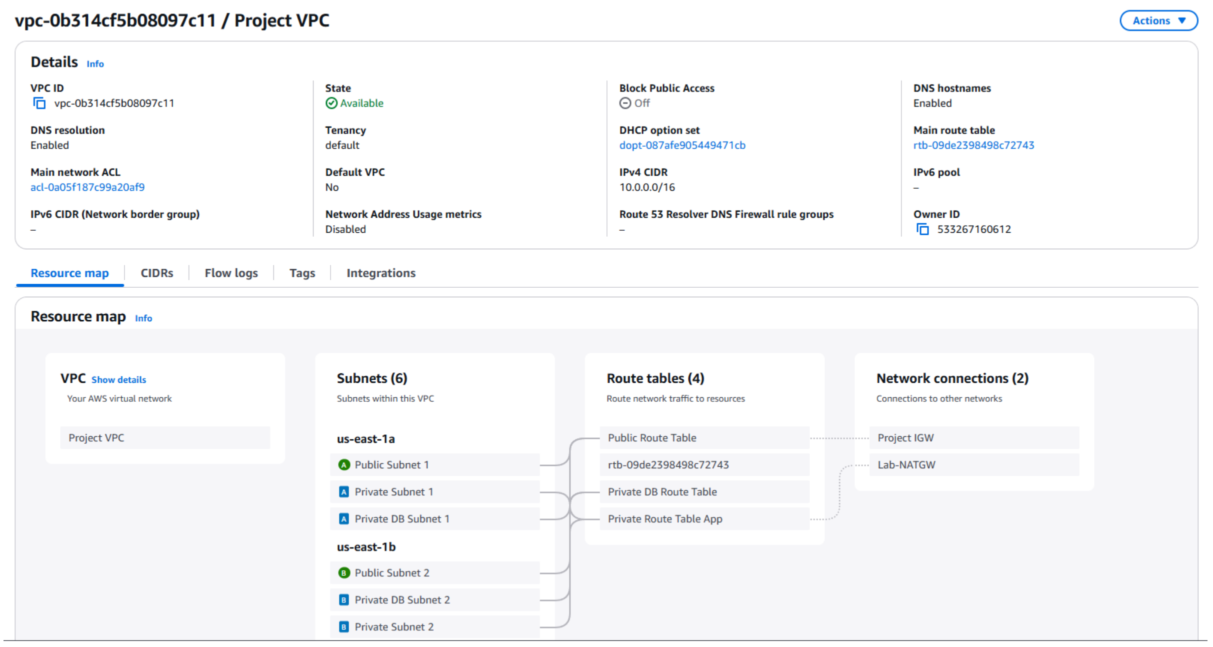
Cost was also a high consideration during implementation. Although AWS provides scalable solutions, it was difficult to keep costs affordable while meeting the project’s growing resource demands. We were forced to study **AWS pricing models** diligently, selecting resource configurations such as EC2 instance types and RDS installations that met both performance and cost considerations. Constant monitoring and fine-tuning were instrumental in keeping a lean infrastructure without compromising functionality.

Aside from these, there were also technical challenges due to a lack of experience in the past in using some of the AWS services such as **Elastic Load Balancing** and **NAT Gateway**. These demanded a learning process for those new to their configuration and integration. However, through the utilization of official AWS documentation and weekly team knowledge-sharing meetings, we could troubleshoot effectively and enhance collective knowledge over the duration.

4.0 Cloud Infrastructure

4.1. Virtual Private Cloud (VPC)

A Virtual Private Cloud (VPC) is an essential AWS service that allows users to provision a secure, isolated virtual network within the AWS Cloud. This virtual network closely resembles a traditional network that would be operate in a personal own data center (Amazon Web Services, 2025). It provides advanced features such as subnets for network segmentation, security groups, and Network Access Control Lists (NACLs) for traffic control, internet gateways for external communication, and elastic IP addresses for consistent addressing (Amazon Web Services, 2025).



vpc-0b314cf5b08097c11 / Project VPC

Details

VPC ID vpc-0b314cf5b08097c11	**State** Available	**Block Public Access** Off	**DNS hostnames** Enabled
DNS resolution Enabled	**Tenancy** default	**DHCP option set** dopt-087afe905449471cb	**Main route table** rtb-09de2398498c72743
Main network ACL acl-0a05f187c99a20af9	**Default VPC** No	**IPv4 CIDR** 10.0.0.0/16	**IPv6 pool** –
IPv6 CIDR (Network border group) –	**Network Address Usage metrics** Disabled	**Route 53 Resolver DNS Firewall rule groups** –	**Owner ID** 533267160612

Resource map

VPC — Your AWS virtual network
- Project VPC

Subnets (6) — Subnets within this VPC
- us-east-1a
  - Public Subnet 1
  - Private Subnet 1
  - Private DB Subnet 1
- us-east-1b
  - Public Subnet 2
  - Private DB Subnet 2
  - Private Subnet 2

Route tables (4) — Route network traffic to resources
- Public Route Table
- rtb-09de2398498c72743
- Private DB Route Table
- Private Route Table App

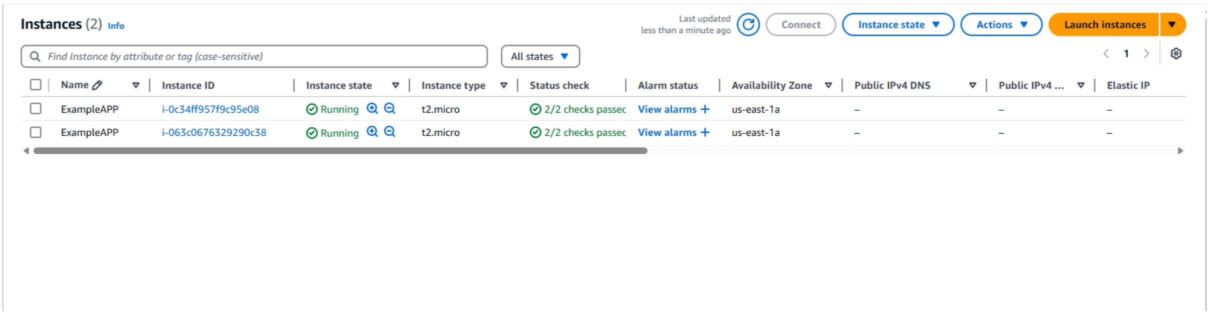
Network connections (2) — Connections to other networks
- Project IGW
- Lab-NATGW

Figure 5: AWS Virtual Private Cloud (VPC) Resource Map

The VPC for the project, designated "Project VPC," has been established with a CIDR block of 10.0.0.0/16, containing six subnets distributed across two Availability Zones (us-east-1a and us-east-1b). This covers public subnets for internet-facing resources, private subnets for internal applications, and specialised database subnets. Traffic management is executed via customised route tables, an Internet Gateway (IGW), and a NAT Gateway to safeguard outbound communication from private subnets.

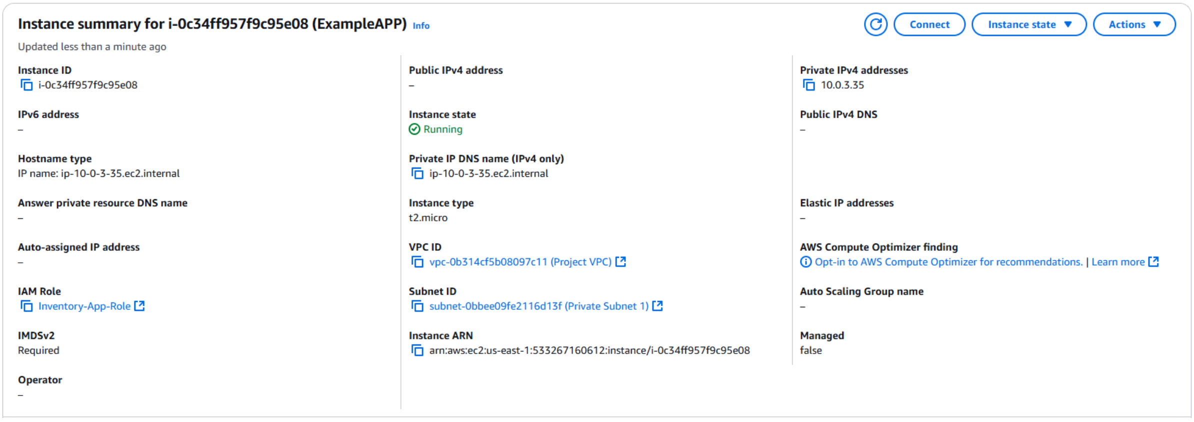
4.2. EC2

Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing resources within the Amazon Web Services (AWS) Cloud, allowing developers and organisations to set up and manage applications with flexibility and cost-effectiveness. Utilising EC2 allows organisations to reduce hardware costs while accelerating the development and deployment of their applications. The service enables customers to deploy any amount of virtual servers, known as EC2 instances, customising configurations for security, networking, and storage to fulfil particular requirements. Every EC2 instance functions as a virtual server, with its hardware specifications determined by the selected instance type, which differs in computational capacity, memory, networking, and storage provisions (Amazon Web Services, 2025).


Instances (2) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
ExampleAPP	i-0c34ff957f9c95e08	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	–	–	–
ExampleAPP	i-063c0676329290c38	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	–	–	–

Figure 6: AWS EC2 Instances Overview

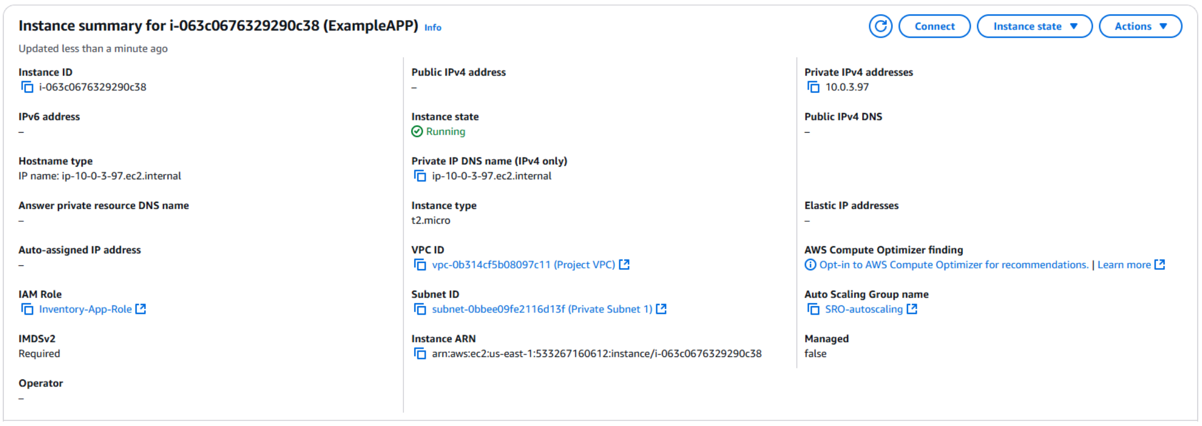


Instance summary for i-0c34ff957f9c95e08 (ExampleAPP) Info

Updated less than a minute ago

Instance ID i-0c34ff957f9c95e08	**Public IPv4 address** –	**Private IPv4 addresses** 10.0.3.35
IPv6 address –	**Instance state** Running	**Public IPv4 DNS** –
Hostname type IP name: ip-10-0-3-35.ec2.internal	**Private IP DNS name (IPv4 only)** ip-10-0-3-35.ec2.internal	
Answer private resource DNS name –	**Instance type** t2.micro	**Elastic IP addresses** –
Auto-assigned IP address –	**VPC ID** vpc-0b314cf5b08097c11 (Project VPC)	**AWS Compute Optimizer finding** Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role Inventory-App-Role	**Subnet ID** subnet-0bbee09fe2116d13f (Private Subnet 1)	**Auto Scaling Group name** –
IMDSv2 Required	**Instance ARN** arn:aws:ec2:us-east-1:533267160612:instance/i-0c34ff957f9c95e08	**Managed** false
Operator –		

Figure 7: Detailed Summary of AWS EC2 Instance 1



Instance summary for i-063c0676329290c38 (ExampleAPP)

Updated less than a minute ago

Instance ID
i-063c0676329290c38

IPv6 address
–

Hostname type
IP name: ip-10-0-3-97.ec2.internal

Answer private resource DNS name
–

Auto-assigned IP address
–

IAM Role
Inventory-App-Role

IMDSv2
Required

Operator
–

Public IPv4 address
–

Instance state
Running

Private IP DNS name (IPv4 only)
ip-10-0-3-97.ec2.internal

Instance type
t2.micro

VPC ID
vpc-0b314cf5b08097c11 (Project VPC)

Subnet ID
subnet-0bbee09fe2116d13f (Private Subnet 1)

Instance ARN
arn:aws:ec2:us-east-1:533267160612:instance/i-063c0676329290c38

Private IPv4 addresses
10.0.3.97

Public IPv4 DNS
–

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name
SRO-autoscaling

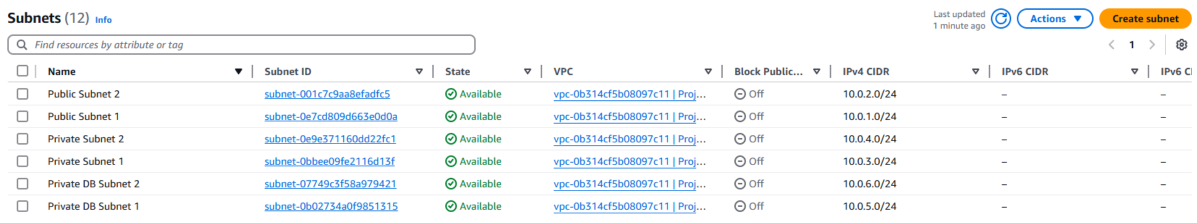
Managed
false

Figure 8: Detailed Summary of AWS EC2 Instance 2

Within the Project VPC deployment, two EC2 instances have been created named "ExampleApp". Both instances utilise the Amazon Linux 2023 AMI and are powered by the t2.micro instance type, which is appropriate for general-purpose workloads. Security measures include SSH key pair authentication and customised security groups to efficiently manage incoming and outgoing traffic. Each instance is provisioned with 8GB of General Purpose SSD (gp2) Elastic Block Store (EBS) volumes, providing an optimal combination of performance and storage efficiency. To guarantee high availability, the instances are carefully distributed across different Availability Zones which are us-east-1a and us-east-1b thereby boosting resilience and preserving operational continuity during an Availability Zone outage.

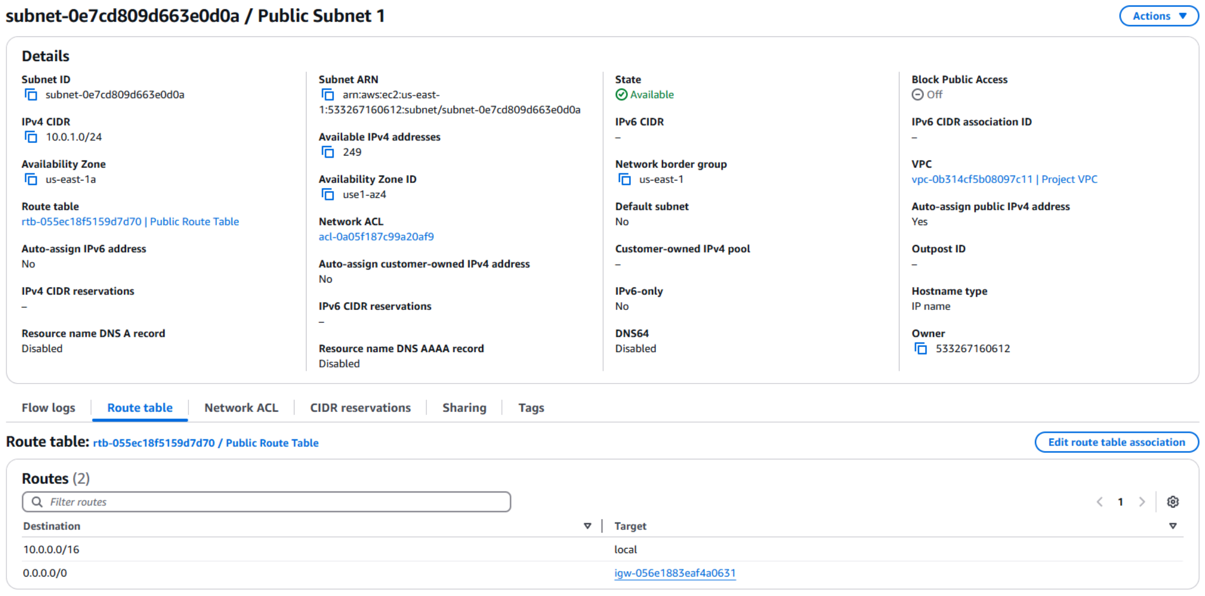
4.3. Subnets

A subnet, or subnetwork, denotes a segmented section of a larger network, specifically a logical partition of an Internet Protocol (IP) network into smaller, more manageable pieces. These subnets are allocated specific ranges of IP addresses and are typically utilised to categorise clients based on logical grouping or physical location, therefore enhancing network traffic efficiency (Wright et al., 2024). In a Virtual Private Cloud (VPC), subnets are crucial for network segmentation, dividing the IP address space into smaller blocks to improve manageability and efficiency. In the "Project VPC" deployment, six subnets have been established, each with a carefully created Classless Inter-Domain Routing (CIDR) block designed to fulfil specific tasks and functions inside the AWS infrastructure, hence assuring efficient organisation and operation of the network.



Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 Cl
Public Subnet 2	subnet-001c7c9aa8efadfc5	Available	vpc-0b314cf5b08097c11 | Proj...	Off	10.0.2.0/24	–	–
Public Subnet 1	subnet-0e7cd809d663e0d0a	Available	vpc-0b314cf5b08097c11 | Proj...	Off	10.0.1.0/24	–	–
Private Subnet 2	subnet-0e9e371160dd22fc1	Available	vpc-0b314cf5b08097c11 | Proj...	Off	10.0.4.0/24	–	–
Private Subnet 1	subnet-0bbee09fe2116d13f	Available	vpc-0b314cf5b08097c11 | Proj...	Off	10.0.3.0/24	–	–
Private DB Subnet 2	subnet-07749c3f58a979421	Available	vpc-0b314cf5b08097c11 | Proj...	Off	10.0.6.0/24	–	–
Private DB Subnet 1	subnet-0b02734a0f9851315	Available	vpc-0b314cf5b08097c11 | Proj...	Off	10.0.5.0/24	–	–

Figure 9: Overview of AWS VPC subnet configurations



subnet-0e7cd809d663e0d0a / Public Subnet 1

Details

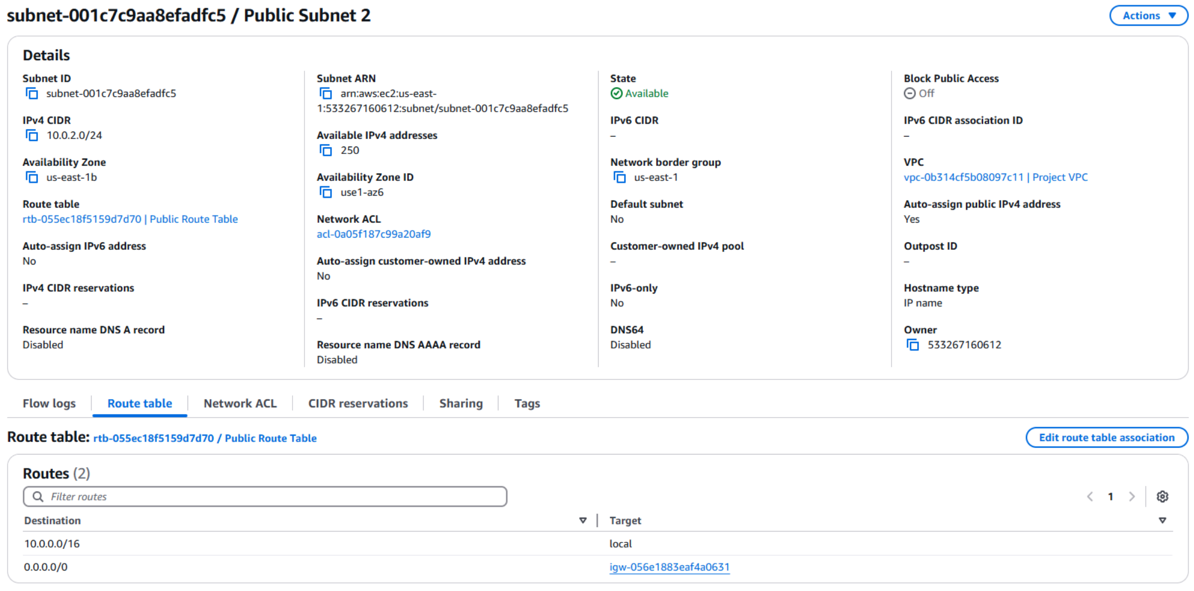
Subnet ID subnet-0e7cd809d663e0d0a	**Subnet ARN** arn:aws:ec2:us-east-1:533267160612:subnet/subnet-0e7cd809d663e0d0a	**State** Available	**Block Public Access** Off
IPv4 CIDR 10.0.1.0/24	**Available IPv4 addresses** 249	**IPv6 CIDR** –	**IPv6 CIDR association ID** –
Availability Zone us-east-1a	**Availability Zone ID** use1-az4	**Network border group** us-east-1	**VPC** vpc-0b314cf5b08097c11 | Project VPC
Route table rtb-055ec18f5159d7d70 | Public Route Table	**Network ACL** acl-0a05f187c99a20af9	**Default subnet** No	**Auto-assign public IPv4 address** Yes
Auto-assign IPv6 address No	**Auto-assign customer-owned IPv4 address** No	**Customer-owned IPv4 pool** –	**Outpost ID** –
IPv4 CIDR reservations –	**IPv6 CIDR reservations** –	**IPv6-only** No	**Hostname type** IP name
Resource name DNS A record Disabled	**Resource name DNS AAAA record** Disabled	**DNS64** Disabled	**Owner** 533267160612

Route table: rtb-055ec18f5159d7d70 / Public Route Table

Routes (2)

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-056e1883eaf4a0631

Figure 10: Detailed AWS VPC Public Subnet 1 Configurations



subnet-001c7c9aa8efadfc5 / Public Subnet 2

Details

Subnet ID subnet-001c7c9aa8efadfc5	**Subnet ARN** arn:aws:ec2:us-east-1:533267160612:subnet/subnet-001c7c9aa8efadfc5	**State** Available	**Block Public Access** Off
IPv4 CIDR 10.0.2.0/24	**Available IPv4 addresses** 250	**IPv6 CIDR** –	**IPv6 CIDR association ID** –
Availability Zone us-east-1b	**Availability Zone ID** use1-az6	**Network border group** us-east-1	**VPC** vpc-0b314cf5b08097c11 | Project VPC
Route table rtb-055ec18f5159d7d70 | Public Route Table	**Network ACL** acl-0a05f187c99a20af9	**Default subnet** No	**Auto-assign public IPv4 address** Yes
Auto-assign IPv6 address No	**Auto-assign customer-owned IPv4 address** No	**Customer-owned IPv4 pool** –	**Outpost ID** –
IPv4 CIDR reservations –	**IPv6 CIDR reservations** –	**IPv6-only** No	**Hostname type** IP name
Resource name DNS A record Disabled	**Resource name DNS AAAA record** Disabled	**DNS64** Disabled	**Owner** 533267160612

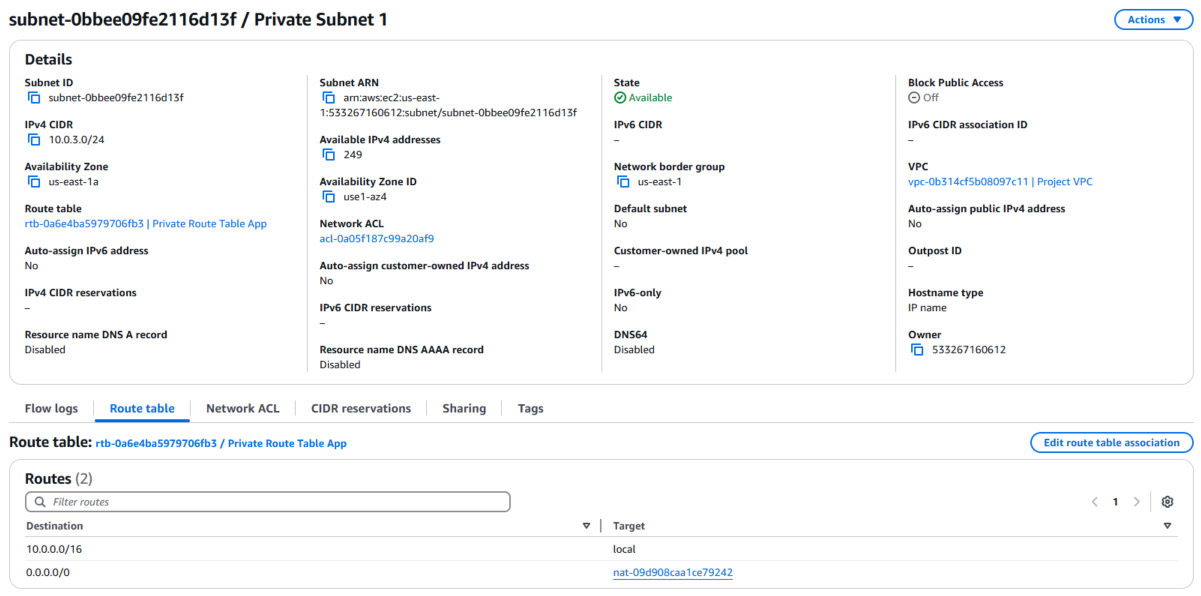
Route table: rtb-055ec18f5159d7d70 / Public Route Table

Routes (2)

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-056e1883eaf4a0631

Figure 11: Detailed AWS VPC Public Subnet 2 Configurations

The public subnets (Public Subnet 1: 10.0.1.0/24 in us-east-1a, Public Subnet 2: 10.0.2.0/24 in us-east-1b) are configured to automatically assign public IPv4 addresses, enabling resources in these subnets to communicate with the internet through an Internet Gateway.



subnet-0bbee09fe2116d13f / Private Subnet 1

Details

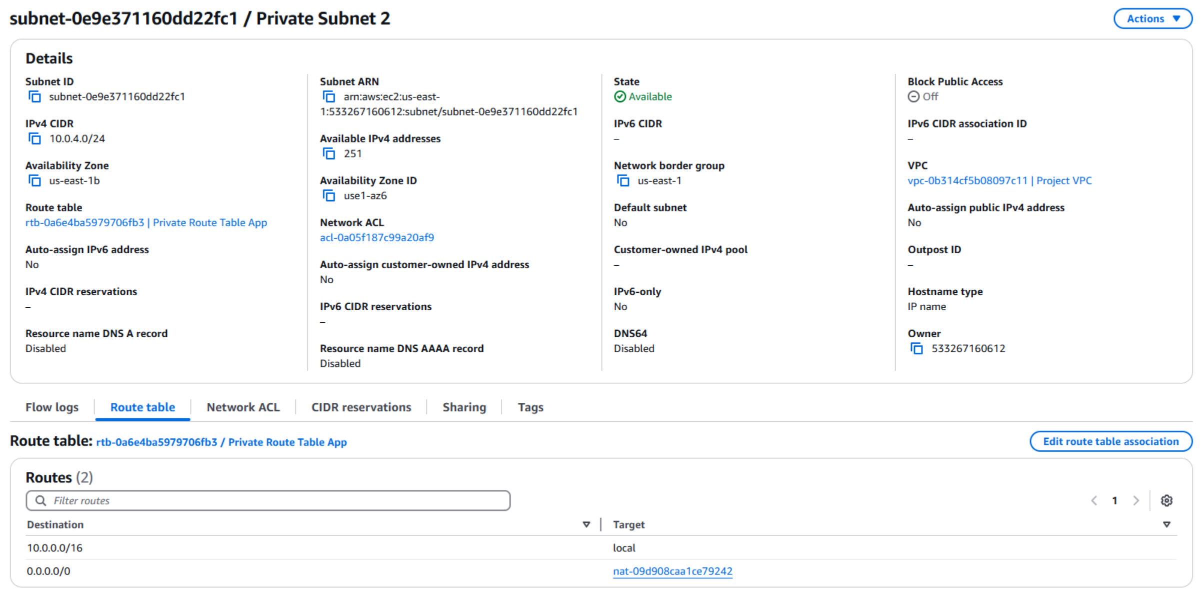
Subnet ID subnet-0bbee09fe2116d13f	**Subnet ARN** arn:aws:ec2:us-east-1:533267160612:subnet/subnet-0bbee09fe2116d13f	**State** Available	**Block Public Access** Off
IPv4 CIDR 10.0.3.0/24	**Available IPv4 addresses** 249	**IPv6 CIDR** –	**IPv6 CIDR association ID** –
Availability Zone us-east-1a	**Availability Zone ID** use1-az4	**Network border group** us-east-1	**VPC** vpc-0b314cf5b08097c11 | Project VPC
Route table rtb-0a6e4ba5979706fb3 | Private Route Table App	**Network ACL** acl-0a05f187c99a20af9	**Default subnet** No	**Auto-assign public IPv4 address** No
Auto-assign IPv6 address No	**Auto-assign customer-owned IPv4 address** No	**Customer-owned IPv4 pool** –	**Outpost ID** –
IPv4 CIDR reservations –	**IPv6 CIDR reservations** –	**IPv6-only** No	**Hostname type** IP name
Resource name DNS A record Disabled	**Resource name DNS AAAA record** Disabled	**DNS64** Disabled	**Owner** 533267160612

Route table: rtb-0a6e4ba5979706fb3 / Private Route Table App

Routes (2)

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-09d908caa1ce79242

Figure 12: Detailed AWS VPC Private Subnet 1 Configurations



subnet-0e9e371160dd22fc1 / Private Subnet 2

Details

Subnet ID subnet-0e9e371160dd22fc1	**Subnet ARN** arn:aws:ec2:us-east-1:533267160612:subnet/subnet-0e9e371160dd22fc1	**State** Available	**Block Public Access** Off
IPv4 CIDR 10.0.4.0/24	**Available IPv4 addresses** 251	**IPv6 CIDR** –	**IPv6 CIDR association ID** –
Availability Zone us-east-1b	**Availability Zone ID** use1-az6	**Network border group** us-east-1	**VPC** vpc-0b314cf5b08097c11 | Project VPC
Route table rtb-0a6e4ba5979706fb3 | Private Route Table App	**Network ACL** acl-0a05f187c99a20af9	**Default subnet** No	**Auto-assign public IPv4 address** No
Auto-assign IPv6 address No	**Auto-assign customer-owned IPv4 address** No	**Customer-owned IPv4 pool** –	**Outpost ID** –
IPv4 CIDR reservations –	**IPv6 CIDR reservations** –	**IPv6-only** No	**Hostname type** IP name
Resource name DNS A record Disabled	**Resource name DNS AAAA record** Disabled	**DNS64** Disabled	**Owner** 533267160612

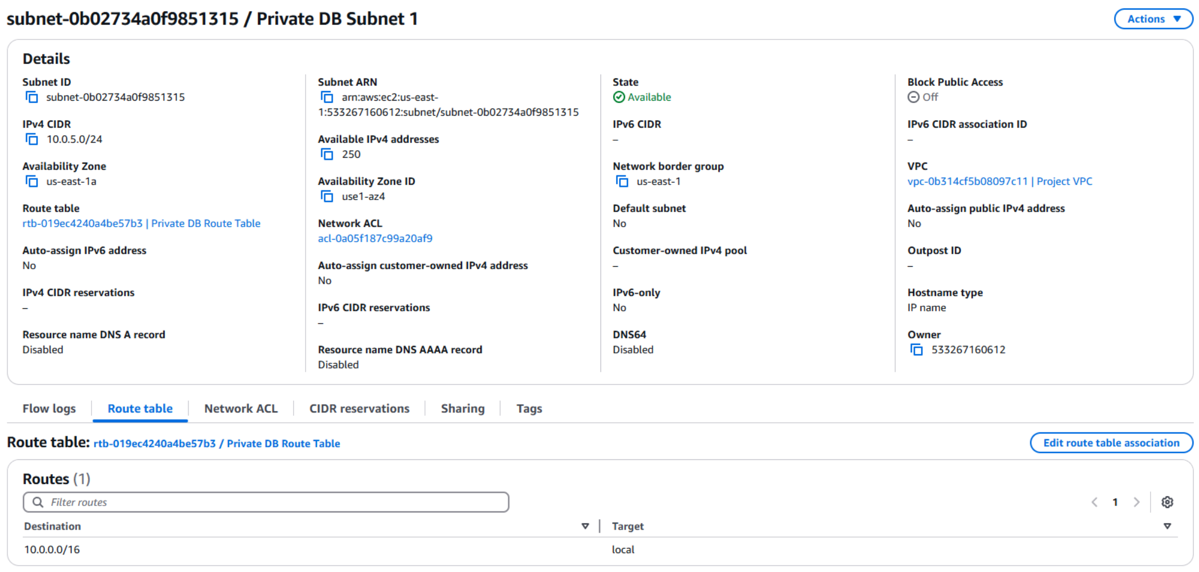
Route table: rtb-0a6e4ba5979706fb3 / Private Route Table App

Routes (2)

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-09d908caa1ce79242

Figure 13: Detailed AWS VPC Private Subnet 2 Configurations

The private subnets (Private Subnet 1: 10.0.3.0/24 in us-east-1a, Private Subnet 2: 10.0.4.0/24 in us-east-1b) host internal resources that do not require direct internet access and route their outbound internet traffic through a NAT Gateway for enhanced security.


subnet-0b02734a0f9851315 / Private DB Subnet 1

Actions

Details

Subnet ID
subnet-0b02734a0f9851315

IPv4 CIDR
10.0.5.0/24

Availability Zone
us-east-1a

Route table
rtb-019ec4240a4be57b3 | Private DB Route Table

Auto-assign IPv6 address
No

IPv4 CIDR reservations
–

Resource name DNS A record
Disabled

Subnet ARN
arn:aws:ec2:us-east-1:533267160612:subnet/subnet-0b02734a0f9851315

Available IPv4 addresses
250

Availability Zone ID
use1-az4

Network ACL
acl-0a05f187c99a20af9

Auto-assign customer-owned IPv4 address
No

IPv6 CIDR reservations
–

Resource name DNS AAAA record
Disabled

State
Available

IPv6 CIDR
–

Network border group
us-east-1

Default subnet
No

Customer-owned IPv4 pool
–

IPv6-only
No

DNS64
Disabled

Block Public Access
Off

IPv6 CIDR association ID
–

VPC
vpc-0b314cf5b08097c11 | Project VPC

Auto-assign public IPv4 address
No

Outpost ID
–

Hostname type
IP name

Owner
533267160612

Flow logs | Route table | Network ACL | CIDR reservations | Sharing | Tags

Route table: rtb-019ec4240a4be57b3 / Private DB Route Table

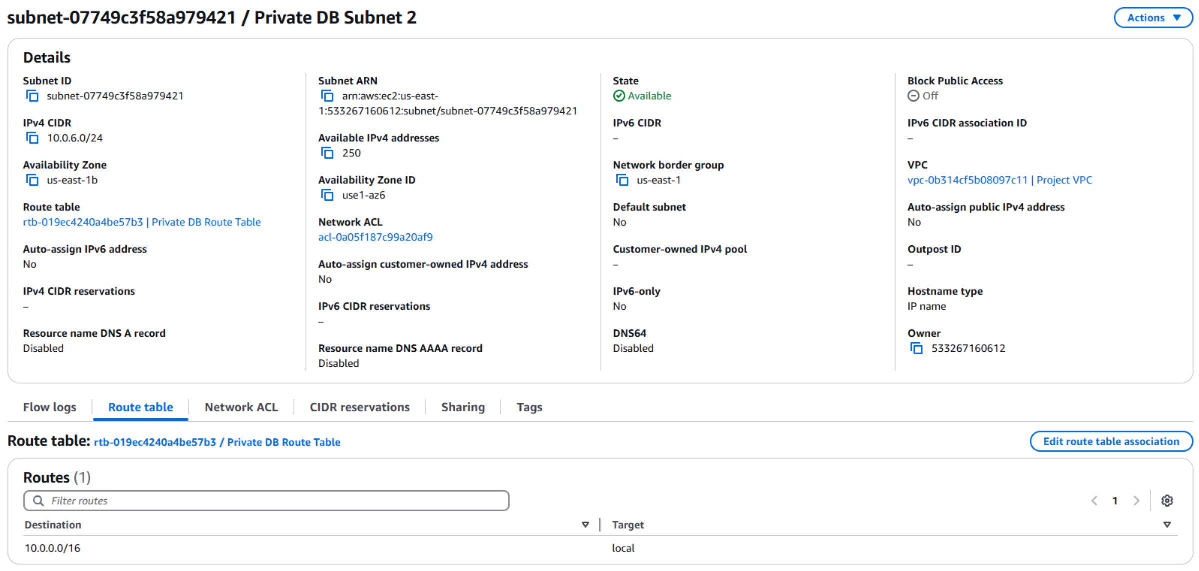
Edit route table association

Routes (1)

Filter routes

Destination	Target
10.0.0.0/16	local

Figure 14: Detailed AWS VPC Private DB Subnet 1 Configurations


subnet-07749c3f58a979421 / Private DB Subnet 2

Actions

Details

Subnet ID
subnet-07749c3f58a979421

IPv4 CIDR
10.0.6.0/24

Availability Zone
us-east-1b

Route table
rtb-019ec4240a4be57b3 | Private DB Route Table

Auto-assign IPv6 address
No

IPv4 CIDR reservations
–

Resource name DNS A record
Disabled

Subnet ARN
arn:aws:ec2:us-east-1:533267160612:subnet/subnet-07749c3f58a979421

Available IPv4 addresses
250

Availability Zone ID
use1-az6

Network ACL
acl-0a05f187c99a20af9

Auto-assign customer-owned IPv4 address
No

IPv6 CIDR reservations
–

Resource name DNS AAAA record
Disabled

State
Available

IPv6 CIDR
–

Network border group
us-east-1

Default subnet
No

Customer-owned IPv4 pool
–

IPv6-only
No

DNS64
Disabled

Block Public Access
Off

IPv6 CIDR association ID
–

VPC
vpc-0b314cf5b08097c11 | Project VPC

Auto-assign public IPv4 address
No

Outpost ID
–

Hostname type
IP name

Owner
533267160612

Flow logs | Route table | Network ACL | CIDR reservations | Sharing | Tags

Route table: rtb-019ec4240a4be57b3 / Private DB Route Table

Edit route table association

Routes (1)

Filter routes

Destination	Target
10.0.0.0/16	local

Figure 15: Detailed AWS VPC Private DB Subnet 2 Configurations

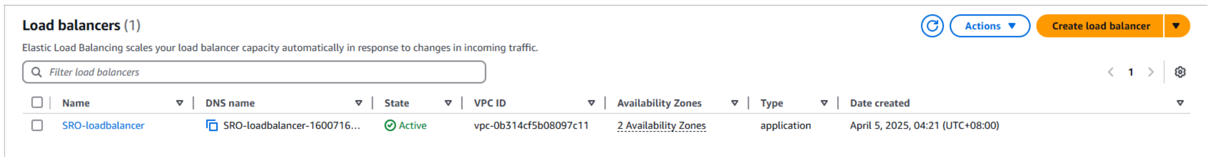
The dedicated database subnets (Private DB Subnet 1: 10.0.5.0/24 in us-east-1a, Private DB Subnet 2: 10.0.6.0/24 in us-east-1b) isolate sensitive database systems, further restricting access and improving data security.

4.4. Load Balancers

AWS Elastic Load Balancing (ELB) allocates incoming application traffic among multiple targets including EC2 instances, containers, and IP addresses across different Availability Zones, thereby improving fault tolerance, availability, and scalability by directing traffic solely to healthy instances, as assessed by health checks (Amazon Web Services, 2025). It provides four versions:

i) Classic Load Balancer (CLB) for fundamental Layer 4/7 balancing in EC2-Classic networks.
ii) Application Load Balancer (ALB) for Layer 7 HTTP/HTTPS traffic, featuring advanced capabilities such as path-based routing, WebSocket, and HTTP/2 support.
iii) Network Load Balancer (NLB) for Layer 4 TCP/UDP traffic, offering high throughput, static IPs, and TLS termination.
iv) Gateway Load Balancer (GLB) for Layer 3 management of third-party appliances (AWS, "Elastic Load Balancing").

Functioning as a singular client contact, ELB employs listeners to oversee connection requests and autonomously adjusts to variable traffic, incorporating Auto Scaling alongside functionalities such as SSL/TLS decryption, AWS WAF, and cross-zone balancing to enhance performance, security, and resource flexibility. However, it encounters limitations, including latency due to misconfiguration or backend complications and constraints related to target group size (Philogène, 2021).

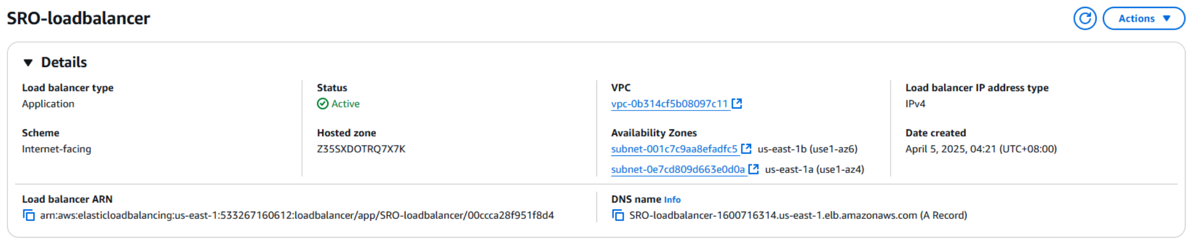


Load balancers (1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
SRO-loadbalancer	SRO-loadbalancer-1600716...	Active	vpc-0b314cf5b08097c11	2 Availability Zones	application	April 5, 2025, 04:21 (UTC+08:00)

Figure 16: Overview of the Application Load Balancer (SRO-loadbalancer)



SRO-loadbalancer

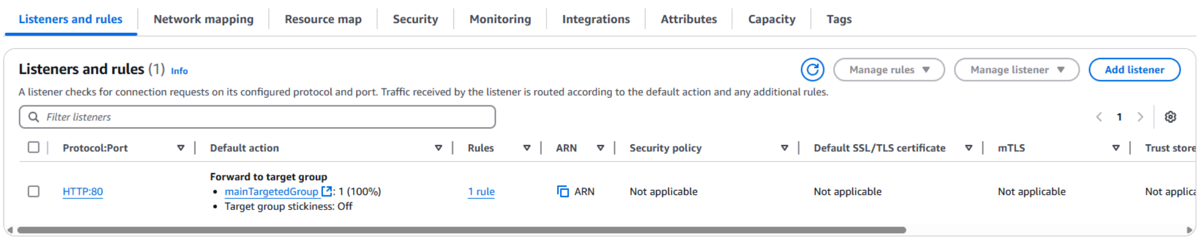
Details

Load balancer type Application	**Status** Active	**VPC** vpc-0b314cf5b08097c11	**Load balancer IP address type** IPv4
Scheme Internet-facing	**Hosted zone** Z35SXDOTRQ7X7K	**Availability Zones** subnet-001c7c9aa8efadfc5 us-east-1b (use1-az6) subnet-0e7cd809d663e0d0a us-east-1a (use1-az4)	**Date created** April 5, 2025, 04:21 (UTC+08:00)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:533267160612:loadbalancer/app/SRO-loadbalancer/00ccca28f951f8d4

DNS name Info
SRO-loadbalancer-1600716314.us-east-1.elb.amazonaws.com (A Record)

Figure 17: Load Balancer Details for SRO-loadbalancer

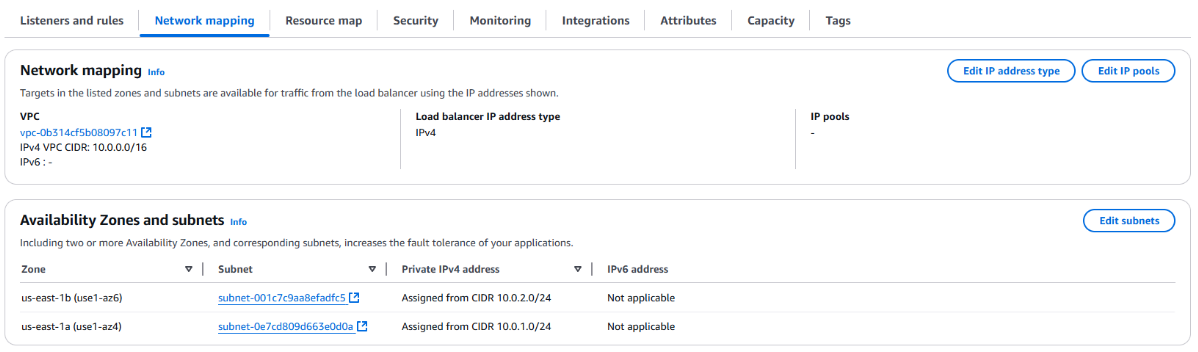


Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust store
HTTP:80	**Forward to target group** • mainTargetedGroup: 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not applicable	Not applica

Figure 18: Listener and Rule Configuration for the Application Load Balancer



Network mapping Info

Targets in the listed zones and subnets are available for traffic from the load balancer using the IP addresses shown.

VPC
vpc-0b314cf5b08097c11
IPv4 VPC CIDR: 10.0.0.0/16
IPv6 : -

Load balancer IP address type
IPv4

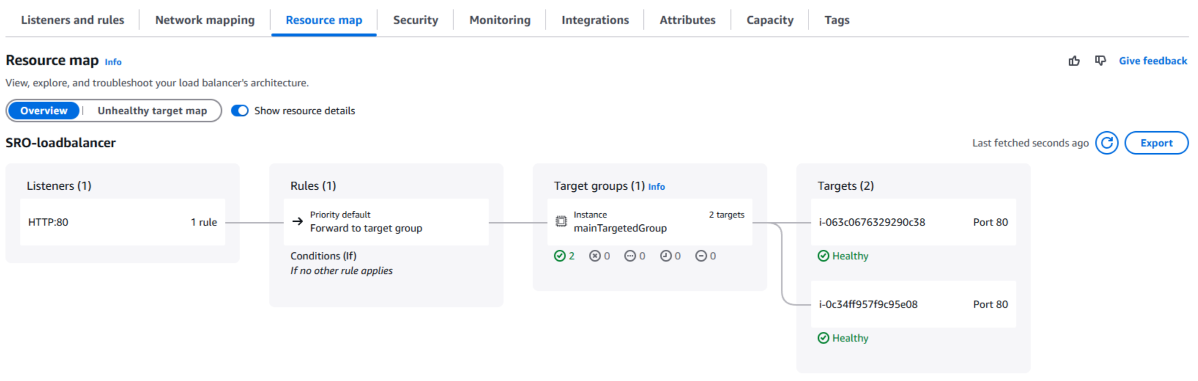
IP pools
-

Availability Zones and subnets Info

Including two or more Availability Zones, and corresponding subnets, increases the fault tolerance of your applications.

Zone	Subnet	Private IPv4 address	IPv6 address
us-east-1b (use1-az6)	subnet-001c7c9aa8efadfc5	Assigned from CIDR 10.0.2.0/24	Not applicable
us-east-1a (use1-az4)	subnet-0e7cd809d663e0d0a	Assigned from CIDR 10.0.1.0/24	Not applicable

Figure 19: Network Mapping Configuration for the Application Load Balancer



Resource map Info

View, explore, and troubleshoot your load balancer's architecture.

SRO-loadbalancer

Last fetched seconds ago

Listeners (1)
HTTP:80 1 rule

Rules (1)
Priority default
Forward to target group
Conditions (If)
If no other rule applies

Target groups (1) Info
Instance
mainTargetedGroup 2 targets
✓ 2 ⊗ 0 ⊖ 0 ◷ 0 ⊖ 0

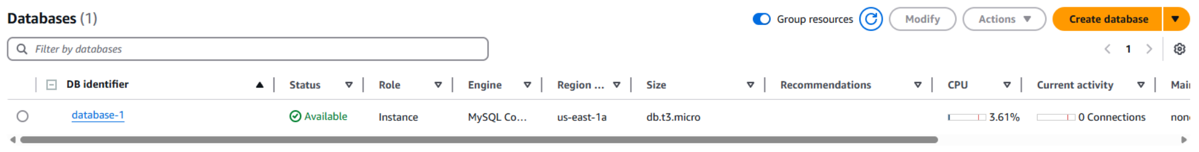
Targets (2)
i-063c0676329290c38 Port 80
✓ Healthy
i-0c34ff957f9c95e08 Port 80
✓ Healthy

Figure 20: Resource Map Overview for the Application Load Balancer

An Application Load Balancer (ALB) named "SRO-loadbalancer" has been deployed to manage and distribute incoming HTTP traffic in the project's architecture. This ALB is set up to operate across two separate Availability Zones which are us-east-1a and us-east-1b within the Project Virtual Private Cloud (VPC), guaranteeing strong operational continuity and resilience against localised failures. The "SRO-loadbalancer," is designed to manage HTTP traffic on port 80, rendering it accessible to external clients via the public internet. It effectively directs this traffic to a designated target group named "mainTargetedGroup," which includes two EC2 instances strategically located in the specified Availability Zones. The ALB performs regular health assessments to evaluate the operational integrity of EC2 instances, ensuring traffic is directed just to those identified as healthy and operational. This careful method significantly improves fault tolerance by reducing the effects of instance or zone-specific failures, enhances scalability by enabling the effortless addition of resources as demand varies, and increases overall application performance through optimised traffic distribution and resource use.

4.5. Relational Database

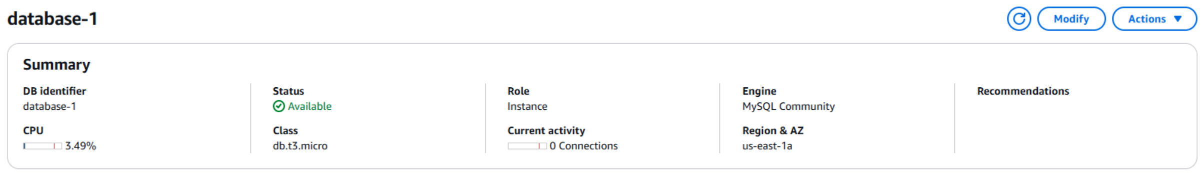
Amazon Web Services (AWS) Relational Database Service (RDS) is a managed service that facilitates the deployment, operation, and scaling of relational databases in the cloud that supports such as MySQL, PostgreSQL, MariaDB, Oracle, SQL Server, and Amazon Aurora, which organise structured data in tables with established relationships to ensure integrity and optimise querying (W3Schools, n.d.). RDS automates administrative functions including hardware provisioning, database configuration, patching, backups, and replication, enabling customers to concentrate on application development instead of infrastructure maintenance (Amazon Web Services, 2025). It provides functionalities such as Multi-AZ deployments for improved availability, wherein data is replicated across Availability Zones to augment fault tolerance, and Read Replicas for enhanced read performance and scalability. Security is enhanced with encryption at rest utilising AWS Key Management Service (KMS) and in transit via SSL/TLS, with credential management administered by AWS Secrets Manager (Amazon Web Services, 2025). RDS accommodates several instance types and storage configurations offering versatility for workloads from minor apps to large-scale business systems.



Databases (1)

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations	CPU	Current activity	Mai
database-1	Available	Instance	MySQL Co...	us-east-1a	db.t3.micro		3.61%	0 Connections	non

Figure 21: Amazon RDS Instance Overview

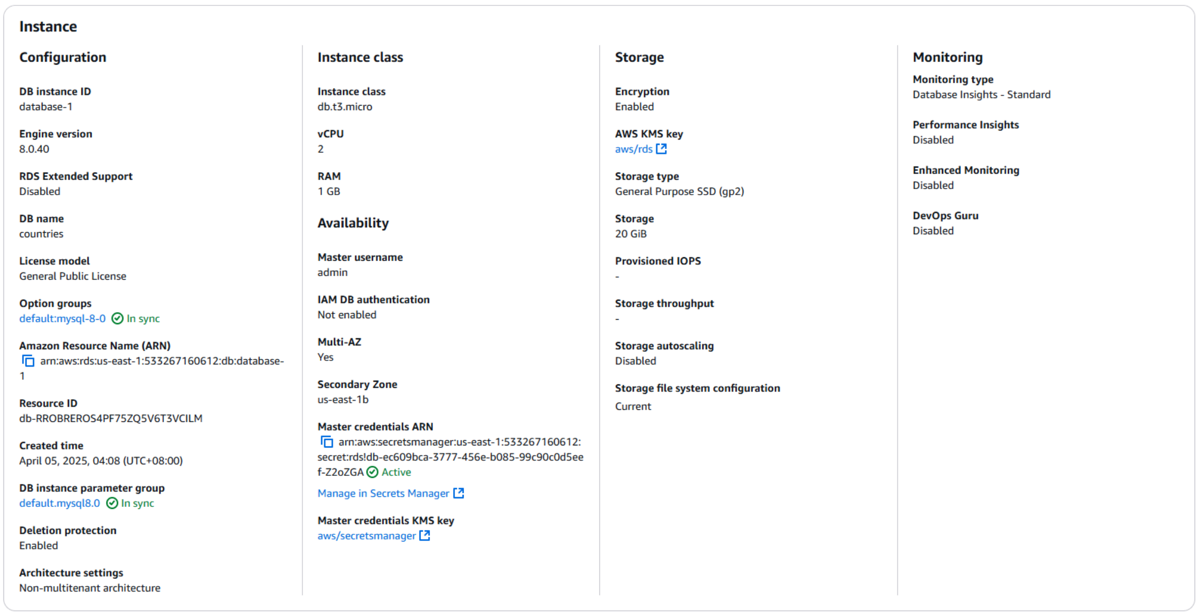


database-1

Summary

DB identifier database-1	Status Available	Role Instance	Engine MySQL Community	Recommendations
CPU 3.49%	Class db.t3.micro	Current activity 0 Connections	Region & AZ us-east-1a	

Figure 22: Database-1 Summary



Instance

Configuration

DB instance ID	database-1
Engine version	8.0.40
RDS Extended Support	Disabled
DB name	countries
License model	General Public License
Option groups	default:mysql-8-0 ✓ In sync
Amazon Resource Name (ARN)	arn:aws:rds:us-east-1:533267160612:db:database-1
Resource ID	db-RROBREROS4PF75ZQ5V6T3VCILM
Created time	April 05, 2025, 04:08 (UTC+08:00)
DB instance parameter group	default.mysql8.0 ✓ In sync
Deletion protection	Enabled
Architecture settings	Non-multitenant architecture

Instance class

Instance class	db.t3.micro
vCPU	2
RAM	1 GB

Availability

Master username	admin
IAM DB authentication	Not enabled
Multi-AZ	Yes
Secondary Zone	us-east-1b
Master credentials ARN	arn:aws:secretsmanager:us-east-1:533267160612:secret:rds!db-ec609bca-3777-456e-b085-99c90c0d5eef-Z2oZGA ✓ Active Manage in Secrets Manager
Master credentials KMS key	aws/secretsmanager

Storage

Encryption	Enabled
AWS KMS key	aws/rds
Storage type	General Purpose SSD (gp2)
Storage	20 GiB
Provisioned IOPS	-
Storage throughput	-
Storage autoscaling	Disabled
Storage file system configuration	Current

Monitoring

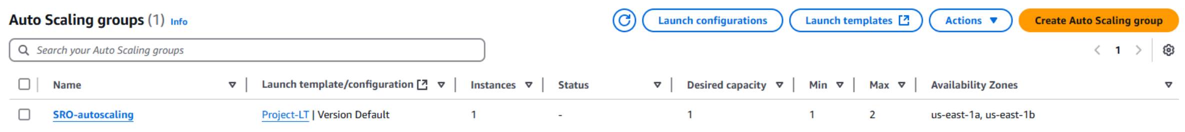
Monitoring type	Database Insights - Standard
Performance Insights	Disabled
Enhanced Monitoring	Disabled
DevOps Guru	Disabled

Figure 23: Database – 1 Detailed Configuration

An Amazon RDS instance titled "database-1" has been established within the project's parameters, utilising the MySQL Community Edition engine (version 8.0.40). This instance utilises a db.t3.micro class, including 2 virtual CPUs (vCPUs) and 1 GB of RAM, supported by a 20 GB General Purpose SSD (gp2) storage volume with encryption enabled for enhanced security. Located in the us-east-1a Availability Zone, the database utilises Multi-AZ deployment, providing redundancy through data synchronisation to a standby instance in the us-east-1b Availability Zone for improved fault tolerance. AWS Secrets Manager is utilised to securely manage database credentials, automating the handling of sensitive information and enhancing security measures. This configuration provides reliable performance, high availability, and strong protection customised for the project's database needs.

4.6. Auto Scaling

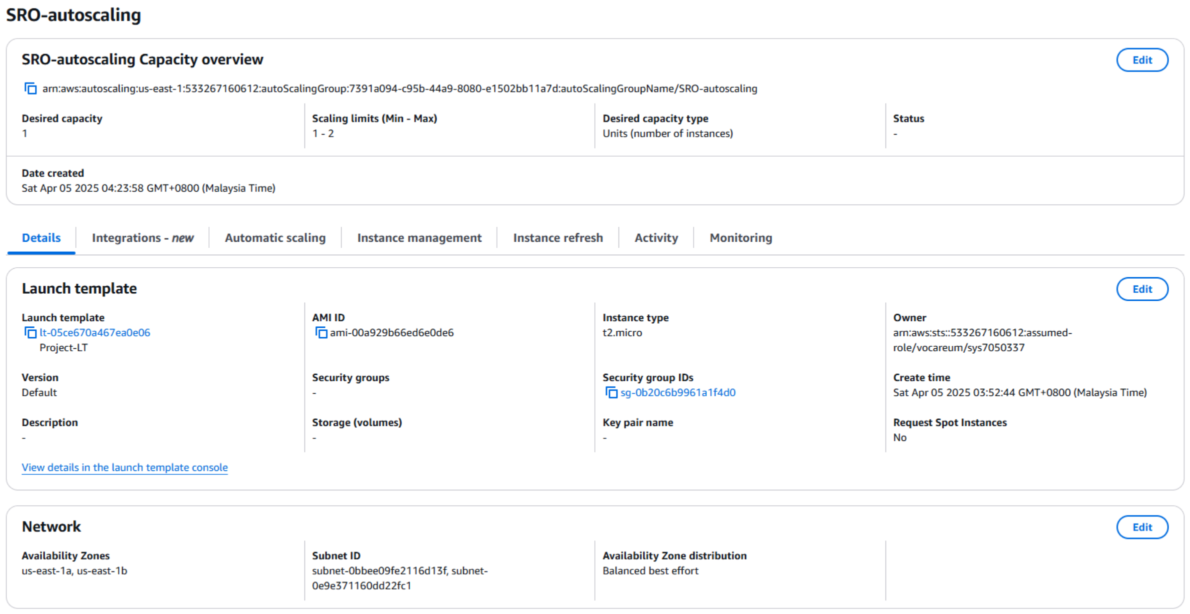
AWS Auto Scaling, a feature provided by Amazon Web Services (AWS), allows organisations to dynamically modify the distribution of AWS resources in response to varying application requirements. It dynamically enhances processing power or storage capacity such as for Amazon Elastic Compute Cloud (EC2), EC2 Spot Fleet requests, Elastic Container Service (ECS), DynamoDB, and Amazon Aurora during peak workloads and reduces it when demand declines, thereby optimising resource utilisation (flexera, 2025). The AWS Auto Scaling Console functions as a centralised interface, enabling users to handle auto scaling capabilities across many AWS services efficiently. This service enables users to implement and manage scalability via predefined scaling strategies that determine resource optimization focusing on availability, cost efficiency, or a balanced approach while also providing the option to create custom strategies suited to particular requirements (W3Schools, n.d.). Moreover, scaling strategies can be executed, utilising rules that incorporate dynamic scaling for immediate modifications or predictive scaling to forecast resource needs based on consumption trends, hence improving operational efficiency and flexibility (Amazon Web Services, 2025).



Auto Scaling groups (1) Info

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
SRO-autoscaling	Project-LT | Version Default	1	-	1	1	2	us-east-1a, us-east-1b

Figure 24: Auto Scaling Groups Overview



SRO-autoscaling

SRO-autoscaling Capacity overview

arn:aws:autoscaling:us-east-1:533267160612:autoScalingGroup:7391a094-c95b-44a9-8080-e1502bb11a7d:autoScalingGroupName/SRO-autoscaling

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
1	1 - 2	Units (number of instances)	-

Date created
Sat Apr 05 2025 04:23:58 GMT+0800 (Malaysia Time)

Launch template

Launch template lt-05ce670a467ea0e06 Project-LT	AMI ID ami-00a929b66ed6e0de6	Instance type t2.micro	Owner arn:aws:sts::533267160612:assumed-role/vocareum/sys7050337
Version Default	Security groups -	Security group IDs sg-0b20c6b9961a1f4d0	Create time Sat Apr 05 2025 03:52:44 GMT+0800 (Malaysia Time)
Description -	Storage (volumes) -	Key pair name -	Request Spot Instances No

Network

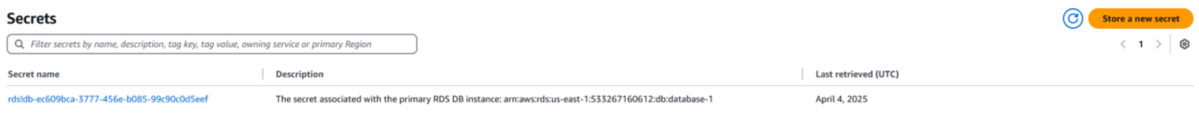
Availability Zones	Subnet ID	Availability Zone distribution
us-east-1a, us-east-1b	subnet-0bbee09fe2116d13f, subnet-0e9e371160dd22fc1	Balanced best effort

Figure 25: Auto Scaling Group Configuration

The project’s structure includes an Auto Scaling group designated as “SRO-autoscaling,” which utilises a launch template named “Project-LT,” configured with the Amazon Linux 2023 AMI and t2.micro instance types. This Auto Scaling group sustains a desired capacity of one instance, with the capability to scale from a minimum of one to a maximum of two instances, regulated by established scaling policies that respond to workload requirements. The setup, covering two Availability Zones—us-east-1a and us-east-1b—improves redundancy and fault tolerance, guaranteeing operational continuity across geographically distinct sites. A chosen security group (sg-0b20c6b9961a1f4d0) enforces security by methodically regulating inbound and outbound traffic to protect the environment. This configuration ensures reliable performance and operational efficiency for the apps included in the project.

4.7. AWS Secrets Manager

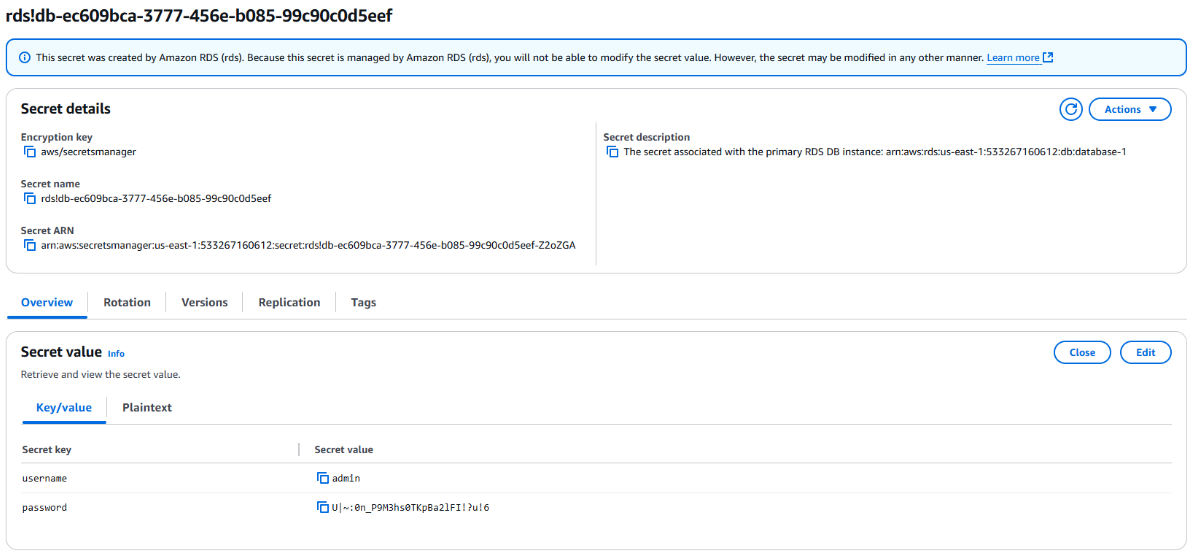
AWS Secrets Manager is a service offered by Amazon Web Services (AWS) that improves an organization's security by managing, retrieving, and rotating sensitive information such as database credentials, application credentials, OAuth tokens, API keys, and other secrets throughout their lifecycles. Thereby, mitigating the risks linked to hard-coded credentials in application source code, CI/CD workflows, or Kubernetes YAML manifests (Guo, 2023). A secret manager substitutes static, hard-coded credentials with dynamic runtime calls to get or synchronise secrets as required. Therefore diminishing the risk of compromise by someone examining program components or configurations. AWS Secrets Manager, integrated with services like Amazon RDS, Redshift, and Lambda, provides centralised storage, automated secret rotation through schedules or Lambda functions, and encryption via AWS Key Management Service (KMS) for data protection at rest, while AWS Identity and Access Management (IAM) policies ensure fine-grained access control (Amazon Web Services, 2024). It facilitates auditing via AWS CloudTrail to monitor secret utilisation and offers APIs/SDKs for effortless programmatic access, thereby streamlining secret management, enhancing compliance, and reducing the risks of credential leakage relative to conventional methods (Amazon Web Services, 2024).



Secrets
Store a new secret
Filter secrets by name, description, tag key, tag value, owning service or primary Region
1

Secret name	Description	Last retrieved (UTC)
rds!db-ec609bca-3777-456e-b085-99c90c0d5eef	The secret associated with the primary RDS DB instance: arn:aws:rds:us-east-1:533267160612:db:database-1	April 4, 2025

Figure 26: Secrets Manager Overview


rds!db-ec609bca-3777-456e-b085-99c90c0d5eef

This secret was created by Amazon RDS (rds). Because this secret is managed by Amazon RDS (rds), you will not be able to modify the secret value. However, the secret may be modified in any other manner. Learn more

Secret details

Encryption key: aws/secretsmanager

Secret name: rds!db-ec609bca-3777-456e-b085-99c90c0d5eef

Secret ARN: arn:aws:secretsmanager:us-east-1:533267160612:secret:rds!db-ec609bca-3777-456e-b085-99c90c0d5eef-Z2oZGA

Secret description: The secret associated with the primary RDS DB instance: arn:aws:rds:us-east-1:533267160612:db:database-1

Retrieve and view the secret value.

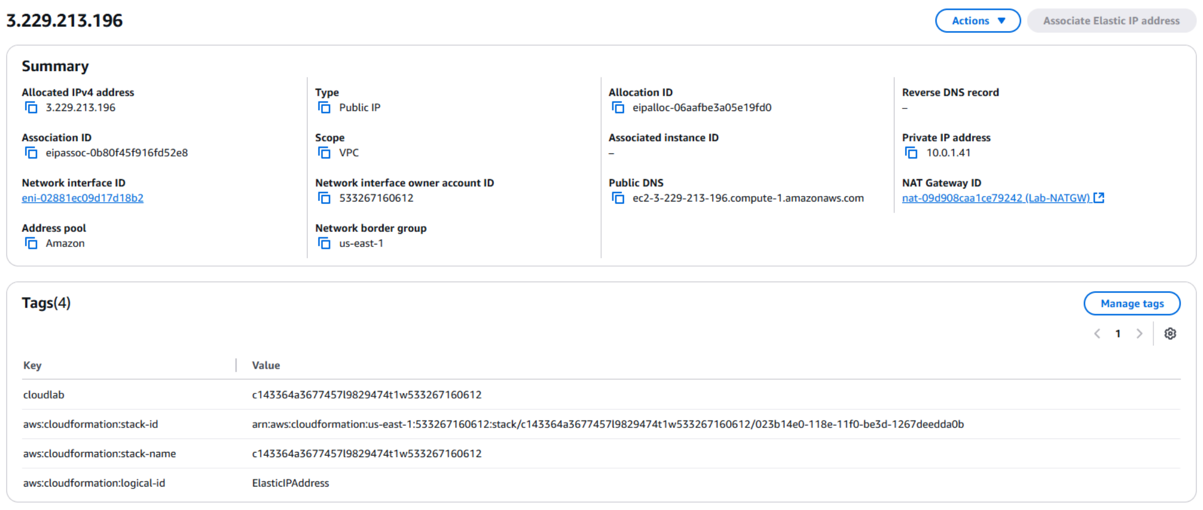
Secret key	Secret value
username	admin
password	U|~:0n_P9M3hs0TKpBa2lFI!?u!6

Figure 27: AWS Secrets Manager Configuration for the RDS Database

The project's architecture employs AWS Secrets Manager to securely manage the credentials for the RDS database instance named "database-1." The secret, designated as "rds!db-ec609bca-3777-456e-b085-99c90c0d5eef," contains the username and password required for MySQL database access which has been created during the database configuration. This secret is protected by encryption via an AWS Key Management Service (KMS) key, providing strong defence against unauthorised access. Only through the retrieval of the Secret Value can the username and password be obtained The credentials are subject to routine automated rotation, enhancing security by mitigating the risk of exposure to prospective breaches or leaks. This configuration enables secure and efficient credential management throughout the project's AWS infrastructure.

4.8. Elastic IP Address

An AWS Elastic IP (EIP) address is a static, public IPv4 address intended for dynamic cloud computing, assigned to an AWS account and maintained until explicitly relinquished, offering a stable endpoint for resources such as Amazon EC2 instances, Network Load Balancers, or network interfaces within a Virtual Private Cloud (VPC) (Amazon Web Services, 2025). This feature enables organisations to conceal instance or software failures by promptly remapping the EIP to an alternative instance, thereby minimising downtime and improving fault tolerance, or by assigning it to a DNS record for consistent domain routing. Thereby streamlining external service configurations such as firewall rules (Amazon Web Services, n.d.). In contrast to conventional public IPs that alter upon the cessation or termination of an instance, an EIP remains static and can be programmatically linked or unlinked, facilitating adaptable traffic rerouting between instances in various Availability Zones to accommodate changing requirements or facilitate server migration (Sharma, n.d.). In AWS EC2 environments, EIPs provide advantages including enhanced resilience, scalability, and manageability by supplying a persistent public address accessible from the internet, which is vital for instances without a public IPv4 address to facilitate external communication (Rai, 2024). Although free when associated with an active instance, unutilised EIPs incur a nominal hourly fee, promoting the effective utilisation of this essential element for dynamic and dependable cloud infrastructure.



3.229.213.196

Summary

Allocated IPv4 address 3.229.213.196	**Type** Public IP	**Allocation ID** eipalloc-06aafbe3a05e19fd0	**Reverse DNS record** –
Association ID eipassoc-0b80f45f916fd52e8	**Scope** VPC	**Associated instance ID** –	**Private IP address** 10.0.1.41
Network interface ID eni-02881ec09d17d18b2	**Network interface owner account ID** 533267160612	**Public DNS** ec2-3-229-213-196.compute-1.amazonaws.com	**NAT Gateway ID** nat-09d908caa1ce79242 (Lab-NATGW)
Address pool Amazon	**Network border group** us-east-1		

Tags(4)

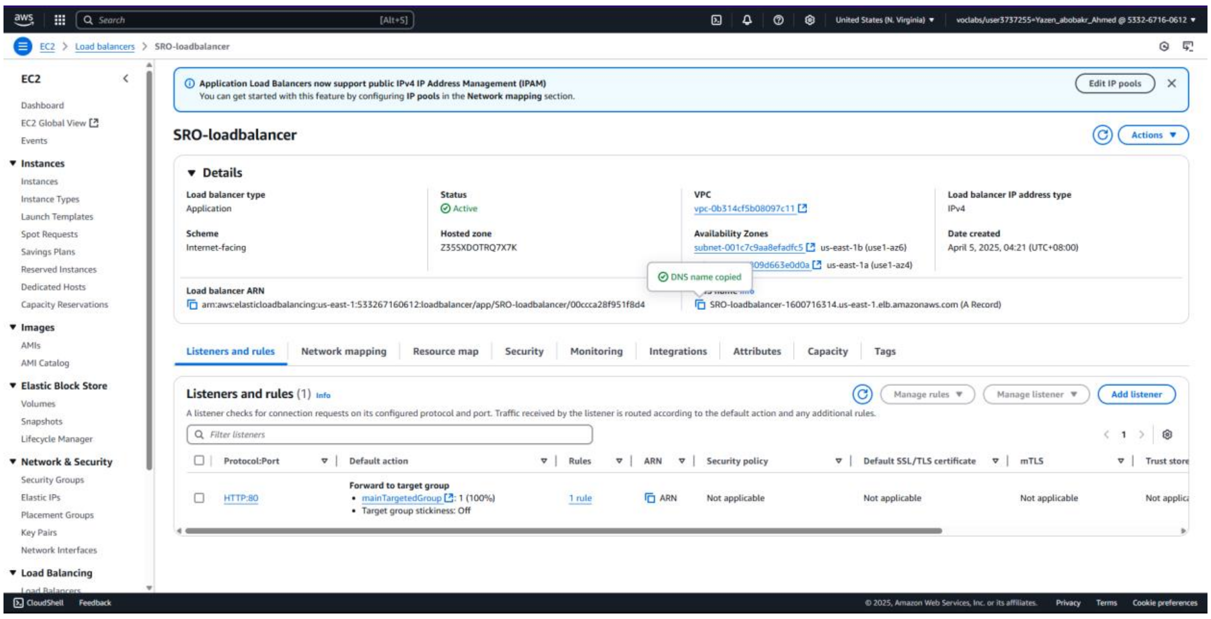
Key	Value
cloudlab	c143364a3677457l9829474t1w533267160612
aws:cloudformation:stack-id	arn:aws:cloudformation:us-east-1:533267160612:stack/c143364a3677457l9829474t1w533267160612/023b14e0-118e-11f0-be3d-1267deedda0b
aws:cloudformation:stack-name	c143364a3677457l9829474t1w533267160612
aws:cloudformation:logical-id	ElasticIPAddress

Figure 28: Elastic IP Address Configuration

An Elastic IP address (3.229.213.196) has been allocated within the project's infrastructure and associated with the Virtual Private Cloud (VPC), especially linked to a network interface to ensure stable and persistent public connectivity for applications operating on EC2 instances. This Elastic IP is associated with a NAT Gateway (nat-09d908caa1ce79242), which manages outbound internet traffic from private subnets, thereby augmenting the security of network connections. This configuration guarantees a reliable, robust, and consistently accessible connection for the project's applications and services, facilitating uninterrupted operational performance.

5.0 Final Outcome

Upon finalising the setup and configuration of all AWS cloud infrastructure components including the VPC, EC2 instances, subnets, Application Load Balancer, RDS database, Auto Scaling group, AWS Secrets Manager, and Elastic IP address the concluding validation step entailed accessing the deployed application via the internet.



Application Load Balancers now support public IPv4 IP Address Management (IPAM)
You can get started with this feature by configuring **IP pools** in the **Network mapping** section.

SRO-loadbalancer

Details

Load balancer type Application	**Status** Active	**VPC** vpc-0b314cf5b08097c11	**Load balancer IP address type** IPv4
Scheme Internet-facing	**Hosted zone** Z35SXDOTRQ7X7K	**Availability Zones** subnet-001c7c9aa8efadfc5 us-east-1b (use1-az6); 09d663e0d0a us-east-1a (use1-az4)	**Date created** April 5, 2025, 04:21 (UTC+08:00)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:533267160612:loadbalancer/app/SRO-loadbalancer/00ccca28f951f8d4

DNS name copied

SRO-loadbalancer-1600716314.us-east-1.elb.amazonaws.com (A Record)

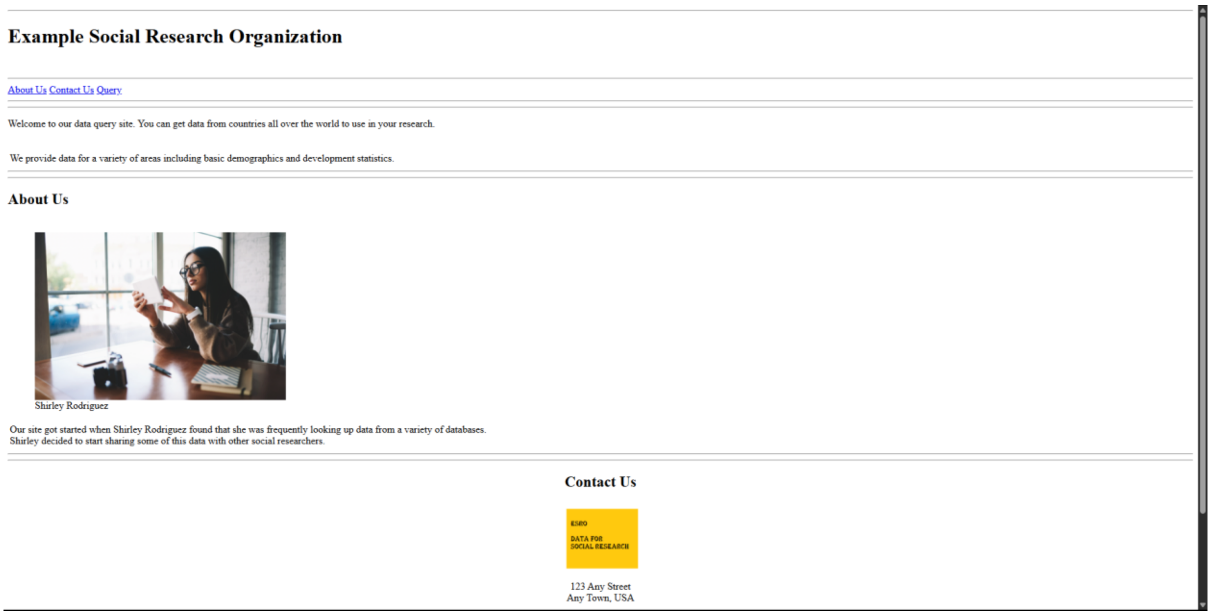
Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust store
HTTP:80	**Forward to target group** • mainTargetedGroup: 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not applicable	Not applic

Figure 29: DNS Name Obtained from SRO-loadbalancer

The DNS name of the Application Load Balancer ("SRO-loadbalancer") was obtained from the AWS Management Console to verify successful deployment. The DNS name was subsequently input into a web browser.



Example Social Research Organization

About Us Contact Us Query

Welcome to our data query site. You can get data from countries all over the world to use in your research.

We provide data for a variety of areas including basic demographics and development statistics.

About Us

Shirley Rodriguez

Our site got started when Shirley Rodriguez found that she was frequently looking up data from a variety of databases.
Shirley decided to start sharing some of this data with other social researchers.

Contact Us

123 Any Street
Any Town, USA

Figure 30: Example Social Research Organization Website

The web application properly rendered upon loading the address, demonstrating that all components, including the front-end EC2 instances and the backend RDS database, were effectively integrated and functioning. This final result confirms that the system is very available, scalable, and securely accessible online, achieving the objectives of the project architecture.

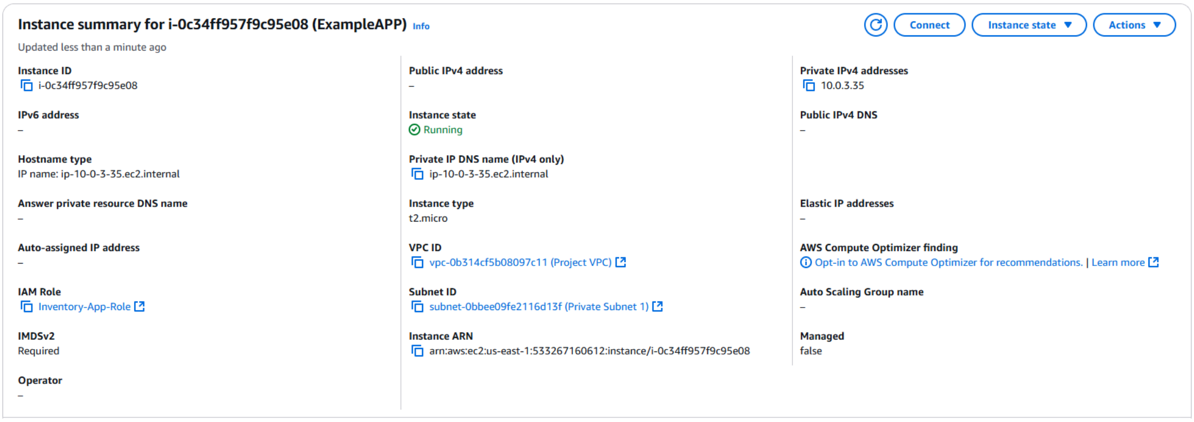
Section B (Individual Work)

6.0 Individual Parts

6.1 Deployment of Cloud Infrastructure & Services (Haziq Irfan Radzali TP072306)

The primary responsibility for this part of the project involved the implementation and precise configuration of diverse AWS cloud infrastructure services to support the operation of a web application. This function focused on developing the fundamental computing and database layers, while also incorporating vital services necessary for maintaining the application's operational effectiveness and secure access. The assignment required a thorough strategy for managing AWS resources, ensuring they conformed to the project's technical specifications while following cloud deployment best practices.

The deployment procedure began with the provisioning of EC2 instances in the designated public and private subnets of the project's Virtual Private Cloud (VPC), employing the Amazon Linux 2023 Amazon Machine Image (AMI) and the t2.micro instance type, appropriate for general-purpose applications. Each instance was precisely designed with sufficient storage volumes specifically 8GB General Purpose SSD (gp2) and fortified with security groups to impose stringent control over inbound and outbound traffic, ensuring secure and regulated access to the resources.



Instance summary for i-0c34ff957f9c95e08 (ExampleAPP)

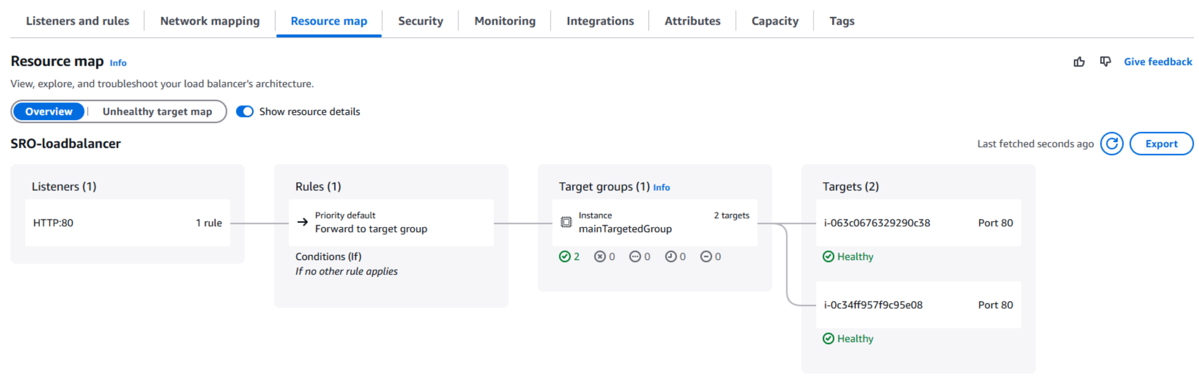
Updated less than a minute ago

Instance ID i-0c34ff957f9c95e08	**Public IPv4 address** –	**Private IPv4 addresses** 10.0.3.35
IPv6 address –	**Instance state** Running	**Public IPv4 DNS** –
Hostname type IP name: ip-10-0-3-35.ec2.internal	**Private IP DNS name (IPv4 only)** ip-10-0-3-35.ec2.internal	
Answer private resource DNS name –	**Instance type** t2.micro	**Elastic IP addresses** –
Auto-assigned IP address –	**VPC ID** vpc-0b314cf5b08097c11 (Project VPC)	**AWS Compute Optimizer finding** Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role Inventory-App-Role	**Subnet ID** subnet-0bbee09fe2116d13f (Private Subnet 1)	**Auto Scaling Group name** –
IMDSv2 Required	**Instance ARN** arn:aws:ec2:us-east-1:533267160612:instance/i-0c34ff957f9c95e08	**Managed** false
Operator –		

Figure 31: AWS EC2 Instance 1 Detailed Summary (Adapted from Figure 7)

This figure presents the specific configuration of an EC2 instance utilised in the project. It encompasses details such the instance ID, private IPv4 address, instance type (t2.micro), VPC and subnet affiliations, IAM role attachment, and instance status. The instance functions within the Project VPC and is located in Private Subnet 1, emphasising its role in internal applications without public internet access.

Subsequent to the creation of the EC2 instances, an Application Load Balancer (ALB) designated "SRO-loadbalancer" was created and configured to route HTTP traffic on port 80 to the instances through a specified target group named "mainTargetedGroup." This configuration allowed external users to access the web application via a stable and dependable DNS endpoint supplied by the ALB, promoting uninterrupted engagement with the deployed services online.



Resource map

View, explore, and troubleshoot your load balancer's architecture.

Show resource details

SRO-loadbalancer

Last fetched seconds ago

Listeners (1)
HTTP:80 1 rule

Rules (1)
Priority default
Forward to target group
Conditions (If)
If no other rule applies

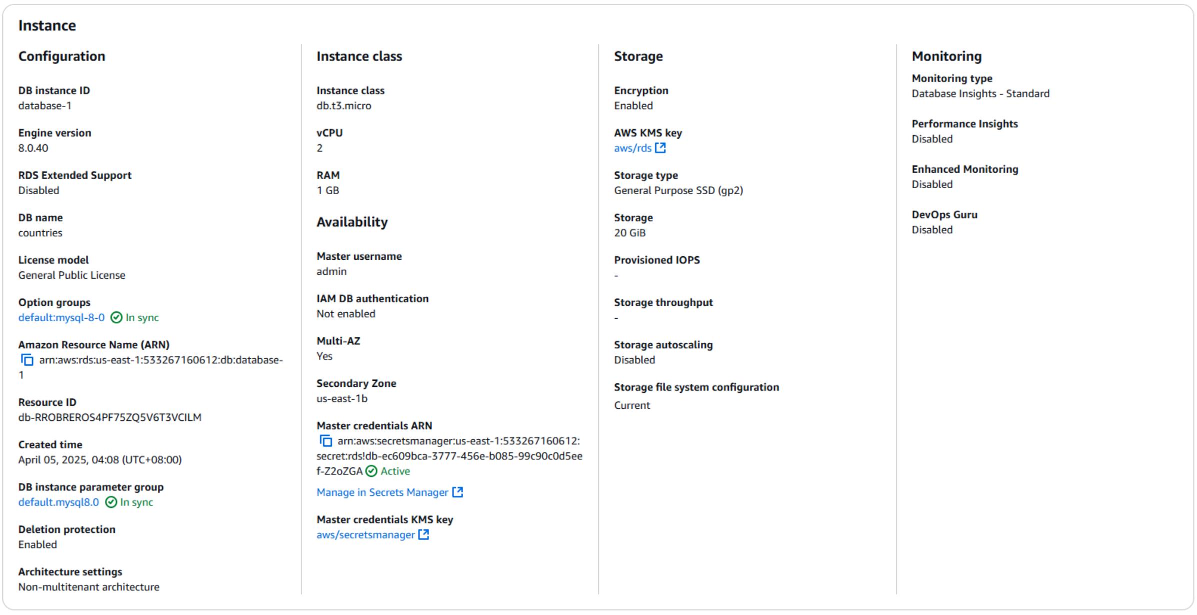
Target groups (1)
Instance
mainTargetedGroup 2 targets
✓ 2 ⊗ 0 ⊖ 0 ◷ 0 ⊖ 0

Targets (2)
i-063c0676329290c38 Port 80
✓ Healthy
i-0c34ff957f9c95e08 Port 80
✓ Healthy

Figure 32: Application Load Balancer Resource Map Overview (Adapted from Figure 20)

This figure illustrates the resource map of the SRO-loadbalancer, demonstrating the routing of inbound HTTP traffic on port 80 through a listener to a target group designated as mainTargetedGroup. The ensemble comprises two robust EC2 instances, guaranteeing high availability. This configuration illustrates the way the Application Load Balancer allocates traffic to various targets, hence improving application reliability and performance.

A MySQL database was implemented via Amazon Relational Database Service (RDS), utilising MySQL Community Edition (version 8.0.40) on a db.t3.micro instance class for the backend infrastructure. The instance featured 20 GB of General Purpose SSD (gp2) storage, with encryption activated to safeguard data at rest. The administration of database credentials was assigned to AWS Secrets Manager, which securely managed the secret "rds!db-ec609bca-3777-456e-b085-99c90c0d5eef" containing the login and password, and executed automated rotation to improve security. This method guaranteed the protection of sensitive information, mitigating risks associated with hardcoding credentials into apps or disclosing them to unauthorised parties.



Instance

Configuration

DB instance ID
database-1

Engine version
8.0.40

RDS Extended Support
Disabled

DB name
countries

License model
General Public License

Option groups
default:mysql-8-0 In sync

Amazon Resource Name (ARN)
arn:aws:rds:us-east-1:533267160612:db:database-1

Resource ID
db-RROBREROS4PF75ZQ5V6T3VCILM

Created time
April 05, 2025, 04:08 (UTC+08:00)

DB instance parameter group
default.mysql8.0 In sync

Deletion protection
Enabled

Architecture settings
Non-multitenant architecture

Instance class

Instance class
db.t3.micro

vCPU
2

RAM
1 GB

Availability

Master username
admin

IAM DB authentication
Not enabled

Multi-AZ
Yes

Secondary Zone
us-east-1b

Master credentials ARN
arn:aws:secretsmanager:us-east-1:533267160612:secret:rds!db-ec609bca-3777-456e-b085-99c90c0d5eef-Z2oZGA Active

Manage in Secrets Manager

Master credentials KMS key
aws/secretsmanager

Storage

Encryption
Enabled

AWS KMS key
aws/rds

Storage type
General Purpose SSD (gp2)

Storage
20 GiB

Provisioned IOPS
-

Storage throughput
-

Storage autoscaling
Disabled

Storage file system configuration
Current

Monitoring

Monitoring type
Database Insights - Standard

Performance Insights
Disabled

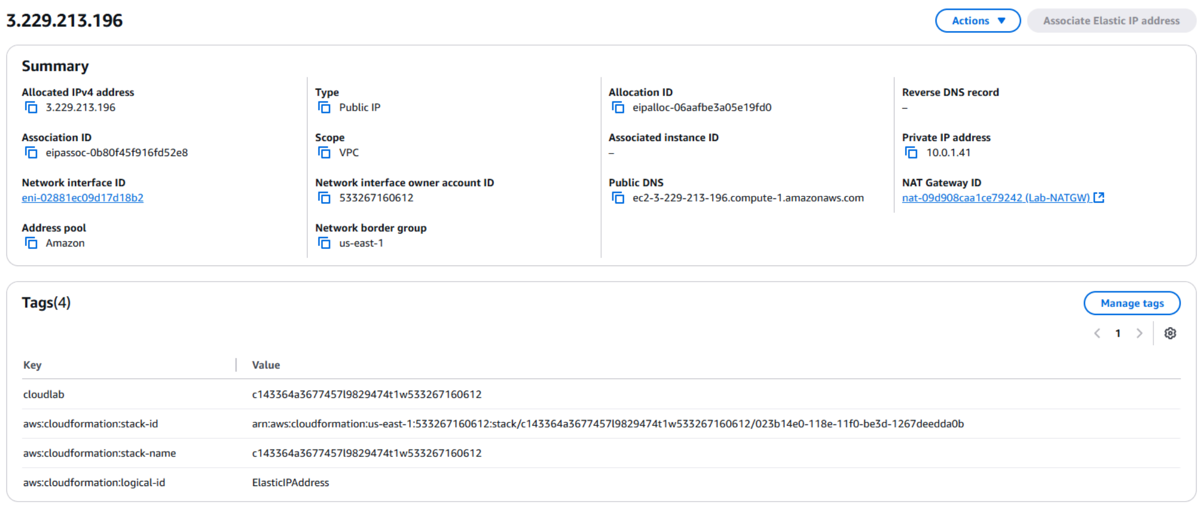
Enhanced Monitoring
Disabled

DevOps Guru
Disabled

Figure 33: Detailed Configuration for Database – 1(Adapted from Figure 23)

This figure displays the complete configuration of the Amazon RDS instance "database-1." It emphasises essential characteristics including the instance class (db.t3.micro), storage type (20 GiB General Purpose SSD), and Multi-AZ deployment for enhanced availability. The instance employs AWS Secrets Manager for safe credential management, utilising encryption via the AWS KMS key. Monitoring settings such as Database Insights are accessible, despite the deactivation of advanced monitoring in this setup.

An Elastic IP address (3.229.213.196) was allocated and associated with the project's Virtual Private Cloud (VPC) to guarantee dependable and secure outbound internet access for internal resources, specifically linked to a NAT Gateway designated as "nat-09d908caa1ce79242." This configuration allowed instances in private subnets to access the internet for necessary updates, software patches, or external services while remaining protected from unsolicited inbound traffic, thus preserving a secure network boundary and ensuring the operational integrity of the infrastructure.



3.229.213.196

Summary

Allocated IPv4 address 3.229.213.196	**Type** Public IP	**Allocation ID** eipalloc-06aafbe3a05e19fd0	**Reverse DNS record** –
Association ID eipassoc-0b80f45f916fd52e8	**Scope** VPC	**Associated instance ID** –	**Private IP address** 10.0.1.41
Network interface ID eni-02881ec09d17d18b2	**Network interface owner account ID** 533267160612	**Public DNS** ec2-3-229-213-196.compute-1.amazonaws.com	**NAT Gateway ID** nat-09d908caa1ce79242 (Lab-NATGW)
Address pool Amazon	**Network border group** us-east-1		

Tags(4)

Key	Value
cloudlab	c143364a3677457l9829474t1w533267160612
aws:cloudformation:stack-id	arn:aws:cloudformation:us-east-1:533267160612:stack/c143364a3677457l9829474t1w533267160612/023b14e0-118e-11f0-be3d-1267deedda0b
aws:cloudformation:stack-name	c143364a3677457l9829474t1w533267160612
aws:cloudformation:logical-id	ElasticIPAddress

Figure 34: Configuration for Elastic IP Address (Adapted from Figure 28)

The figure represents the distribution and configuration specifics of the Elastic IP address utilised in the project. The IP address (3.229.213.196) is linked to the NAT Gateway, facilitating safe outbound internet access for instances located in private subnets. Essential characteristics including the scope (VPC), allocation ID, and public DNS are presented, alongside metadata tags utilised for resource management and CloudFormation integration.

6.2 Migration (Yazen Abobakr Ahmed Al-mehdhar TP069210)

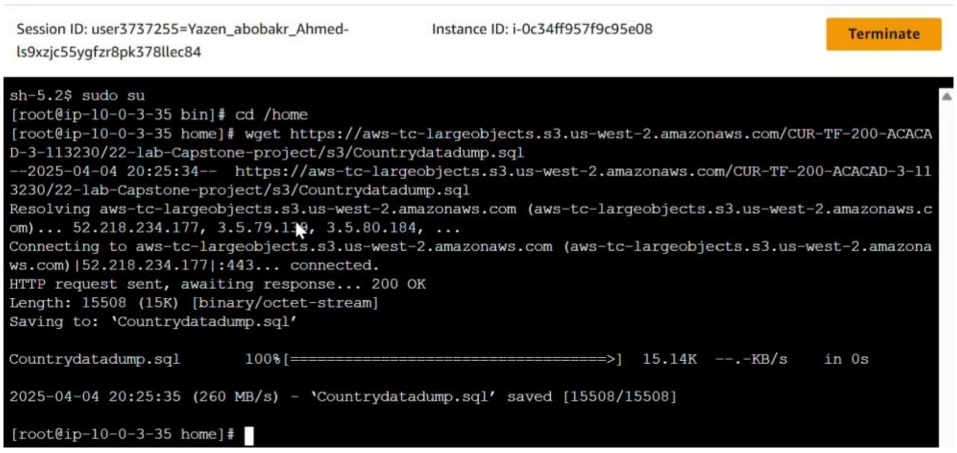
mySQL Data Migration

This migration of the database was done through amazon's RDS using MySQL because of its managed service benefits: scalability, high availability, and security features. The RDS instance was configured using a Multi-AZ deployment pattern that creates a primary database instance in one Availability Zone and a synchronous standby replica in another Availability Zone. The failover capability is built to ensure no downtime failover even if infrastructure fails and maintain service level agreement (SLA) of more than 99.95% uptime.

In terms of networking security, the database instances were purposely provisioned inside private subnet of the Virtual Private Cloud (VPC). This architectural decision has been aimed at achieving critical security controls: the database resource is not accessible from the Internet directly but can be accessed only by authorized connections from an application-tier EC2 instance through precise security group rules. VPC flow logging is enabled for complete monitoring and auditing of all network traffic related to access to the database.

The migration of an existing database into a SQL dump file (countriesdatadump.sql) containing schema definition and complete data records was the starting part of the migration process. The process followed the key technical steps given below:

1. **Data Retrieval**: Super User Privilege was attained, the SQL dump file (Countrydatadump.sql) was secured download from AWS S3 through wget, with the command executed as:



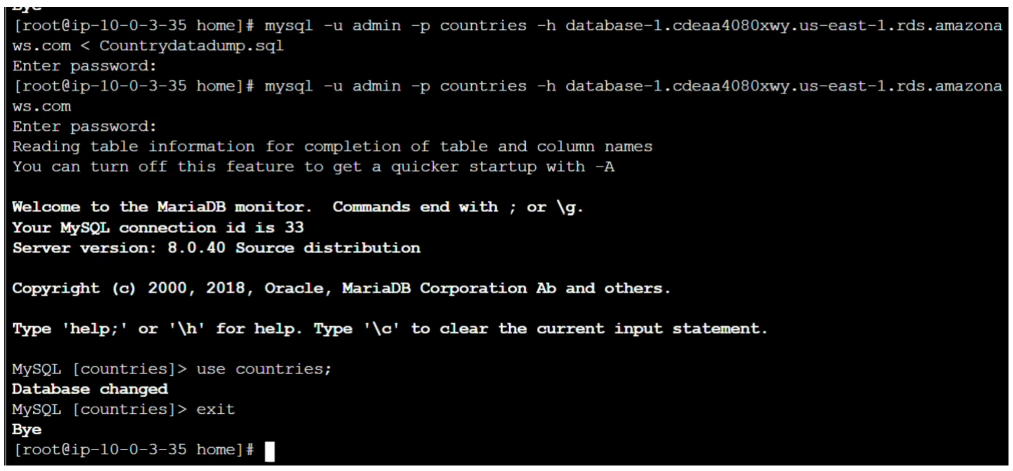
Session ID: user3737255=Yazen_abobakr_Ahmed-ls9xzjc55ygfzr8pk378llec84
Instance ID: i-0c34ff957f9c95e08

```
sh-5.2$ sudo su
[root@ip-10-0-3-35 bin]# cd /home
[root@ip-10-0-3-35 home]# wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACACA
D-3-113230/22-lab-Capstone-project/s3/Countrydatadump.sql
--2025-04-04 20:25:34--  https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACACAD-3-11
3230/22-lab-Capstone-project/s3/Countrydatadump.sql
Resolving aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.c
om)... 52.218.234.177, 3.5.79.130, 3.5.80.184, ...
Connecting to aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazona
ws.com)|52.218.234.177|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15508 (15K) [binary/octet-stream]
Saving to: ‘Countrydatadump.sql’

Countrydatadump.sql       100%[===================================>]  15.14K  --.-KB/s    in 0s

2025-04-04 20:25:35 (260 MB/s) - ‘Countrydatadump.sql’ saved [15508/15508]

[root@ip-10-0-3-35 home]# █
```

Figure 35: Data Retrieval

2. **Secure Database Import**:

   The migration took place in MySQL client with this command, afterwards the database was accessed again to ensure changes were done, “use countries;” Is used to access the new database :



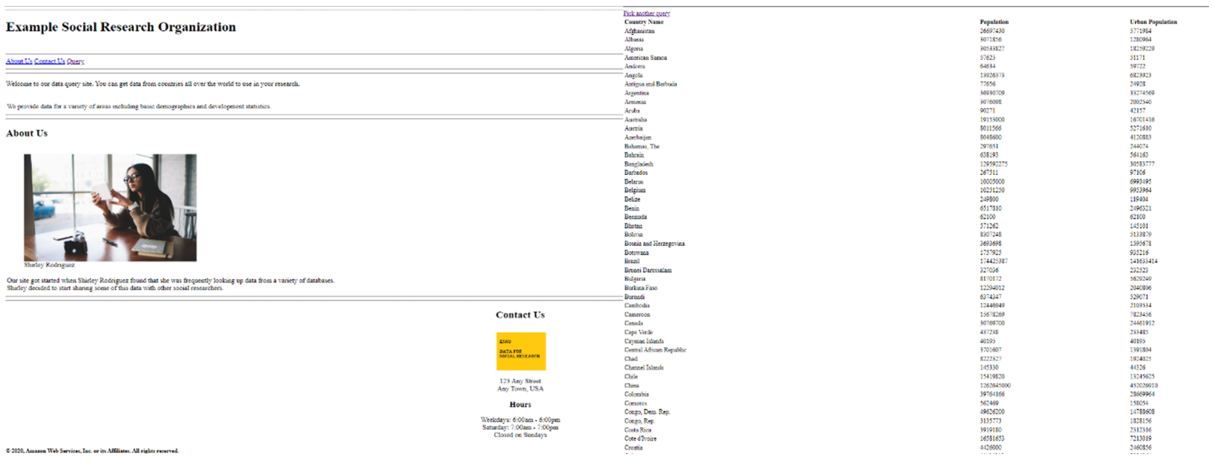
```
[root@ip-10-0-3-35 home]# mysql -u admin -p countries -h database-1.cdeaa4080xwy.us-east-1.rds.amazona
ws.com < Countrydatadump.sql
Enter password:
[root@ip-10-0-3-35 home]# mysql -u admin -p countries -h database-1.cdeaa4080xwy.us-east-1.rds.amazona
ws.com
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 33
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [countries]> use countries;
Database changed
MySQL [countries]> exit
Bye
[root@ip-10-0-3-35 home]# 
```

Figure 36: Secure Database Import

3. **Verification**:

   Migration was confirmed successful by:

- Creation of new MySQL session to the RDS instance
- Checking out the 'countries' database was accessible
- Confirming table information completion was working fine via mySQL and the web App.



Example Social Research Organization

About Us Contact Us Query

Welcome to our data query site. You can get data from countries all over the world to use in your research.

We provide data for a variety of areas including basic demographics and development statistics.

About Us

Shirley Rodriguez

Our site got started when Shirley Rodriguez found that she was frequently looking up data from a variety of databases. Shirley decided to start sharing some of this data with other social researchers.

Contact Us

123 Any Street
Any Town, USA

Hours

Weekdays: 6:00am - 6:00pm
Saturday: 7:00am - 7:00pm
Closed on Sundays

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Pick another query

Country Name	Population	Urban Population
Afghanistan	26697430	5771984
Albania	3071856	1280964
Algeria	30533827	18259229
American Samoa	57625	51171
Andorra	64634	59722
Angola	13926373	6823923
Antigua and Barbuda	77656	24928
Argentina	36930709	33274569
Armenia	3076098	2002540
Aruba	90271	42157
Australia	19153000	16701416
Austria	8011566	5271610
Azerbaijan	8048600	4120883
Bahamas, The	297651	244074
Bahrain	638193	564163
Bangladesh	129592275	30583777
Barbados	267511	97106
Belarus	10005000	6993495
Belgium	10251250	9953964
Belize	249800	119404
Benin	6517810	2496321
Bermuda	62100	62100
Bhutan	571262	145101
Bolivia	8307248	5133879
Bosnia and Herzegovina	3693698	1595678
Botswana	1757925	935216
Brazil	174425387	141633414
Brunei Darussalam	327036	232523
Bulgaria	8170172	5629249
Burkina Faso	12294012	2040806
Burundi	6374347	529071
Cambodia	12446949	2103534
Cameroon	15678269	7823456
Canada	30769700	24461912
Cape Verde	437238	233485
Cayman Islands	40195	40195
Central African Republic	3701607	1391804
Chad	8222327	1924025
Channel Islands	145330	44326
Chile	15419820	13245625
China	1262645000	452026910
Colombia	39764166	28669964
Comoros	562469	158054
Congo, Dem. Rep.	49626200	14788608
Congo, Rep.	3135773	1828156
Costa Rica	3919180	2312316
Cote d'Ivoire	16581653	7213019
Croatia	4426000	2460856

Figure 37: Verification

Web Application Migration

The deployed web application was done using a predefined launch template (Example-LT) such that standardized EC2 instance configuration. Such associates using an Amazon Linux 2023 AMI with Apache HTTP Server and PHP runtime pre-installed, application's files well put up under the web root directory (/var/www/html), and file directory permissions. The template also enrolled IAM roles cited to establish least privilege with regard to access to any AWS service.

In the case of manual installations, an all-encompassing procedure would apply. It consists of downloading the application package through secure transfer, extracting the contents to the web server directory, assigning proper ownership and permissions, and restarting the web services. All application servers resided in private subnets with inbound access having tight restrictions set only to allow HTTP traffic through the Application Load Balancer while SSH access was limited to specific management hosts.

Other Critical Considerations

- Scalability & High Availability

both must be in mind upon designing an architecture for the application. these considerations were catered to using Auto Scaling Groups capable of making on-demand adjustments to EC2 capacity and are distributed across multiple AZs for fault tolerance. RDS Multi-AZ deployment keeps a standby replica for automatic failover (usually completed within 2 minutes) while the Application Load Balancer intelligently routes requests to healthy instances. High availability with automatic failover brings together many layers of scalability and continuous availability.

- Security & Compliance

They aim to provide a multi-layered protection scheme for all components:

- At rest (KMS) and in transit (TLS 1.2+) encryption.
- Credentials management via Secrets Manager and automatic rotation.
- Network isolation via private subnets and security groups.
- Monitoring via CloudWatch and CloudTrail.
- Automated backups, with a 7-day retention period for data durability.

6.3 Security (Abdulrahman Gamil Mohammed Ahmed TP071012)

Virtual Private Cloud (VPC)

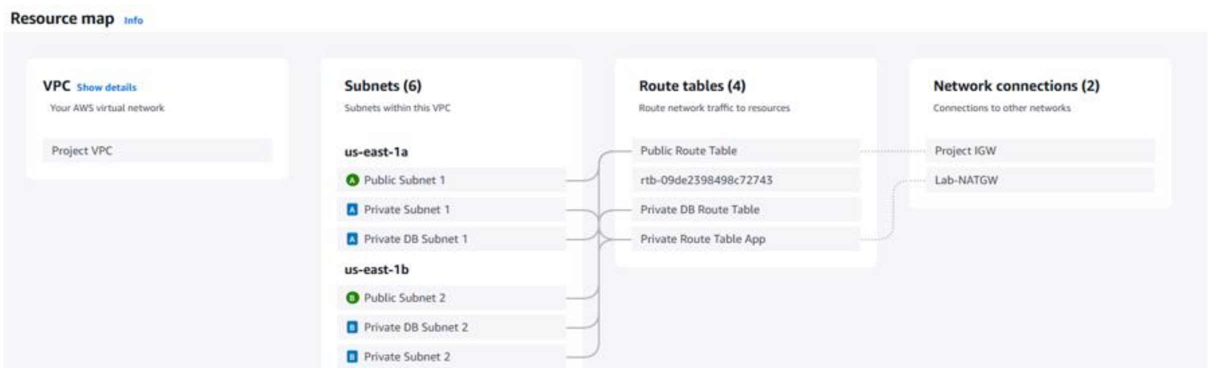


Figure 38 VPC

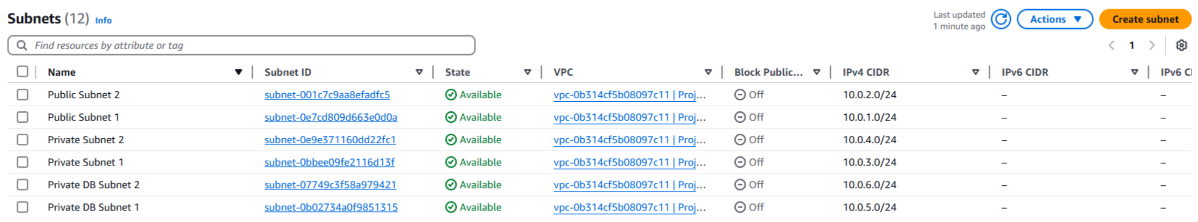
Virtual Private Cloud (VPC) is the foundation layer of networking and security in cloud architecture. It plays a pivotal role in determining the structure of the application environment and serves as the cloud-based equivalent of a physical on-premises network but with cloud-specific flexibility and scalability.

In our configuration, the VPC is to provide a secure and isolated network environment where cloud resources are logically divided into different subnets based on their roles and levels of access needed. The segregation helps in enforcing a clear boundary between public-facing services (e.g., web servers) and private resources (e.g., databases or internal layers of applications), hence enhancing security and control of performance.

The VPC allows for granular traffic management through features like route tables, network ACLs, and security groups, whereby we can highly specifically control which traffic can move in and out of each subnet. Doing so ensures that only authorized access is granted to sensitive components of the infrastructure.

Furthermore, the VPC also supports hybrid network setups that support safe connections to other AWS services, availability zones, or even on-premises settings via VPN or AWS Direct Connect. This means that scalable and distributed systems can be implemented without compromising data security or integrity.

In this project, the VPC has been established to follow cloud networking best practice, thus ensuring that all aspects of our solution operate inside an explicitly outlined, secure, and sizable environment. This deliberate structuring not only enables robust communication between different resource groups, but it also ensures high-level security, isolation, and operational performance across the entirety of the infrastructure.


Subnets (12) Info

Last updated 1 minute ago

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 C
Public Subnet 2	subnet-001c7c9aa8efadfc5	Available	vpc-0b314cf5b08097c11 | Proj...	Off	10.0.2.0/24	–	–
Public Subnet 1	subnet-0e7cd809d663e0d0a	Available	vpc-0b314cf5b08097c11 | Proj...	Off	10.0.1.0/24	–	–
Private Subnet 2	subnet-0e9e371160dd22fc1	Available	vpc-0b314cf5b08097c11 | Proj...	Off	10.0.4.0/24	–	–
Private Subnet 1	subnet-0bbee09fe2116d13f	Available	vpc-0b314cf5b08097c11 | Proj...	Off	10.0.3.0/24	–	–
Private DB Subnet 2	subnet-07749c3f58a979421	Available	vpc-0b314cf5b08097c11 | Proj...	Off	10.0.6.0/24	–	–
Private DB Subnet 1	subnet-0b02734a0f9851315	Available	vpc-0b314cf5b08097c11 | Proj...	Off	10.0.5.0/24	–	–

Figure 39 VPC's subnets

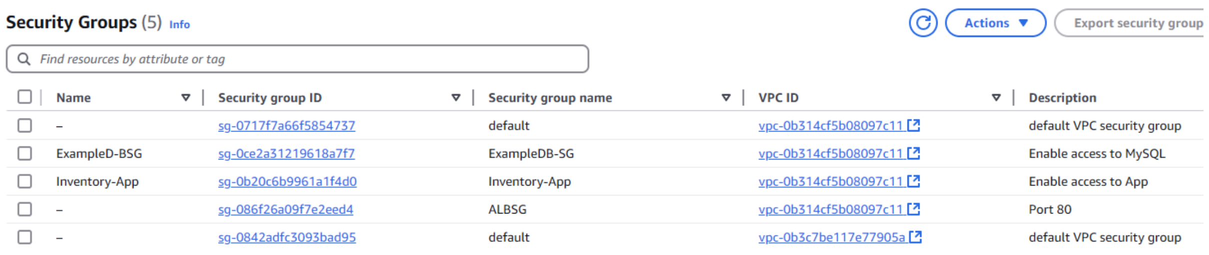
The VPC architecture setup in total has six subnets divided across two Availability Zones (us-east-1a and us-east-1b) for fault and high availability. Each AZ has:

- One internet-facing public subnet for items like the **Elastic Load Balancer**.
- One **application instance's** private subnet.
- One private database subnet for **hosting RDS instances**.

In us-east-1a, we have Public Subnet 1, Private Subnet 1, and Private DB Subnet 1. These are repeated in us-east-1b with identical Subnet 2 configurations. This will make sure that services will still be accessible even in case of an AZ failure.

Public subnets allow all outside access, primarily to send traffic to the load balancer. Private subnets, on the other hand, are used to manage internal resources such as application servers and databases, which are not directly exposed to the internet, thus providing higher security.

It is also possible to split database and application layers into separate private subnets. That provides even more traffic granularity and policy control. Routing tables are utilized in order to govern communication among subnets with proper isolation and secure data sharing within the VPC.


Security Groups (5) Info

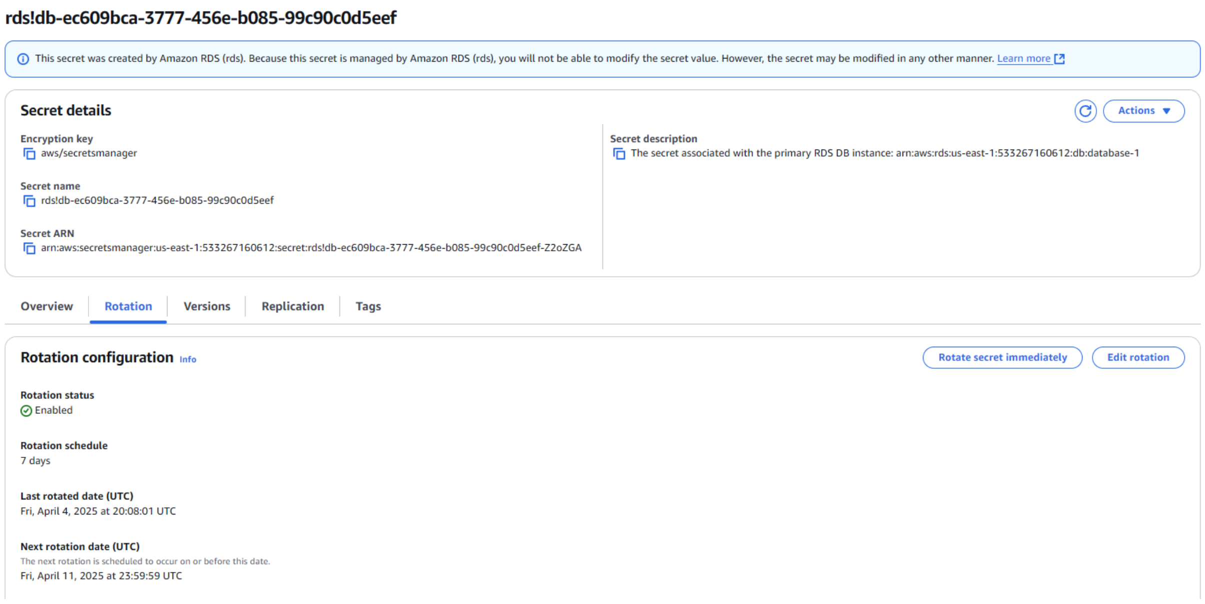
Name	Security group ID	Security group name	VPC ID	Description
–	sg-0717f7a66f5854737	default	vpc-0b314cf5b08097c11	default VPC security group
ExampleD-BSG	sg-0ce2a31219618a7f7	ExampleDB-SG	vpc-0b314cf5b08097c11	Enable access to MySQL
Inventory-App	sg-0b20c6b9961a1f4d0	Inventory-App	vpc-0b314cf5b08097c11	Enable access to App
–	sg-086f26a09f7e2eed4	ALBSG	vpc-0b314cf5b08097c11	Port 80
–	sg-0842adfc3093bad95	default	vpc-0b3c7be117e77905a	default VPC security group

Figure 40: Security Groups

Security groups enhance security overall by acting as virtual firewalls, controlling traffic into and out of a single instance. Each security group contains rules that allow or deny types of traffic based on what the instance needs. For example, the Inventory-App security group allows instances of the application to communicate with each other while denying all other unwanted traffic for the application to run securely. Similarly, the Example-BSG security group facilitates allowed communication to the MySQL database but denies unauthorized access, thus preserving the data integrity.

Additional security groups are created for other components, such as the load balancer and default VPC security groups, so each section of the infrastructure is properly protected against any possible attacks. The layered security model in VPC that handles both subnet and instance access has multiple layers of protection, so even if one layer is breached, there are others that remain to secure the application and data.

AWS Secret Manager



rds!db-ec609bca-3777-456e-b085-99c90c0d5eef

This secret was created by Amazon RDS (rds). Because this secret is managed by Amazon RDS (rds), you will not be able to modify the secret value. However, the secret may be modified in any other manner. Learn more

Secret details

Encryption key
aws/secretsmanager

Secret name
rds!db-ec609bca-3777-456e-b085-99c90c0d5eef

Secret ARN
arn:aws:secretsmanager:us-east-1:533267160612:secret:rds!db-ec609bca-3777-456e-b085-99c90c0d5eef-Z2oZGA

Secret description
The secret associated with the primary RDS DB instance: arn:aws:rds:us-east-1:533267160612:db:database-1

Rotation configuration Info

Rotation status
Enabled

Rotation schedule
7 days

Last rotated date (UTC)
Fri, April 4, 2025 at 20:08:01 UTC

Next rotation date (UTC)
The next rotation is scheduled to occur on or before this date.
Fri, April 11, 2025 at 23:59:59 UTC

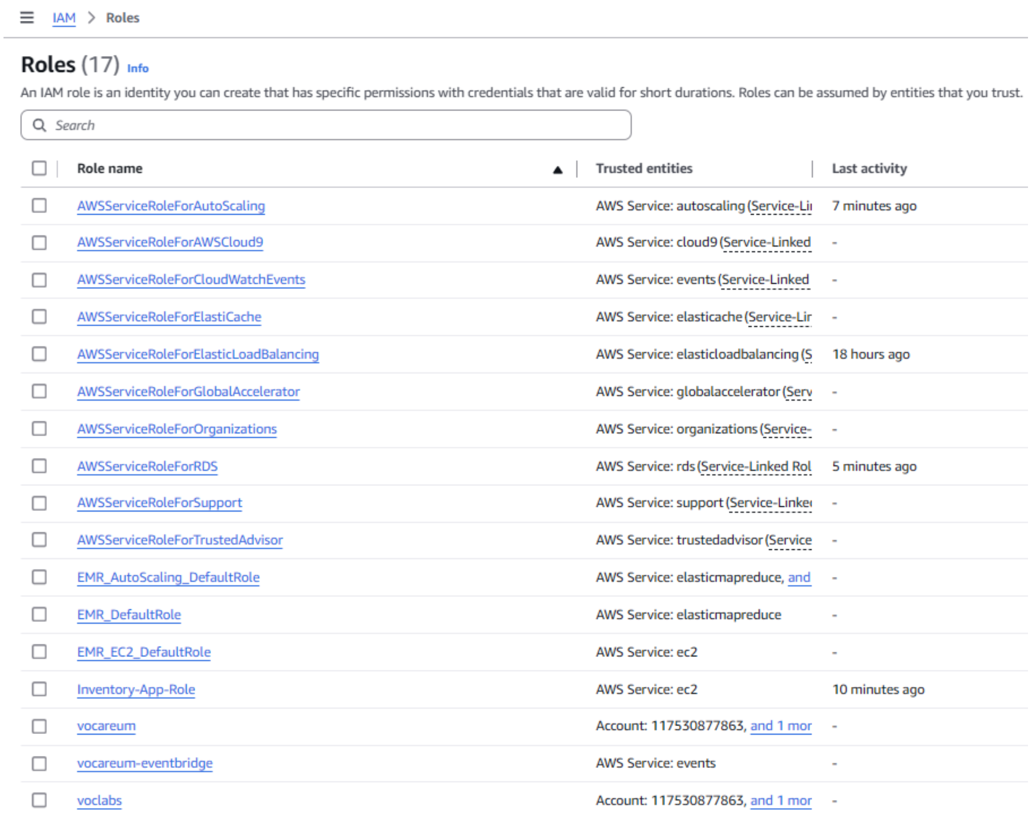
Figure 41: Configuration of AWS Secret Manager

AWS Secrets Manager also plays a key role in securing sensitive information, including the credentials to the Amazon Aurora MySQL database. It stores these credentials such as usernames and passwords securely and rotates them automatically every seven days. Rotation is crucial in minimizing credential compromise risk because it updates credentials at regular intervals without the involvement of humans, thus promoting security and preventing unauthorized usage.

The rotation is set and was last done on **April 4th, 2025**, and is to be done on **April 11th, 2025**. The preventive measure ensures the database credentials remain updated, thereby limiting the threat of exposure to potential attackers who seek static credentials.

Use of the secrets stored in AWS Secrets Manager is securely controlled by **IAM roles**, and privileged access is granted to only certain administrators. The roles adhere to the least privilege principle, i.e., only authorized users or services are permitted to see or modify sensitive credentials. Such close coupling of **IAM** roles with Secrets Manager adheres to best practices in security and compliance and provides a robust means of managing credentials in the cloud platform.

Identity and Access Management (IAM)


IAM > Roles

Roles (17) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

Role name	Trusted entities	Last activity
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Li	7 minutes ago
AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked	-
AWSServiceRoleForCloudWatchEvents	AWS Service: events (Service-Linked	-
AWSServiceRoleForElastiCache	AWS Service: elasticache (Service-Lir	-
AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (S	18 hours ago
AWSServiceRoleForGlobalAccelerator	AWS Service: globalaccelerator (Serv	-
AWSServiceRoleForOrganizations	AWS Service: organizations (Service-	-
AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Rol	5 minutes ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
EMR_AutoScaling_DefaultRole	AWS Service: elasticmapreduce, and	-
EMR_DefaultRole	AWS Service: elasticmapreduce	-
EMR_EC2_DefaultRole	AWS Service: ec2	-
Inventory-App-Role	AWS Service: ec2	10 minutes ago
vocareum	Account: 117530877863, and 1 mor	-
vocareum-eventbridge	AWS Service: events	-
voclabs	Account: 117530877863, and 1 mor	-

Figure 42: Roles of IAM

In the AWS ecosystem, IAM roles have been configured to enact the **principle of least privilege**. That is, each resource and service only has the permissions necessary to carry out its specific task, enhancing security in general.

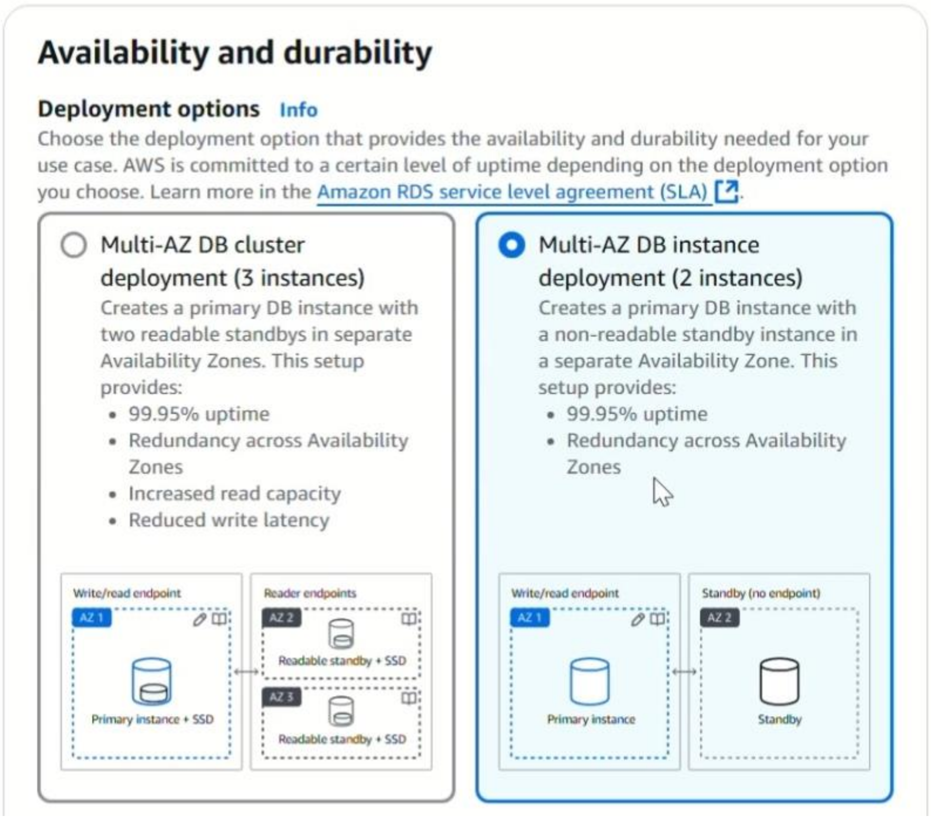
Various IAM roles are tailored for specific AWS services, such as **Auto Scaling**, **CloudWatch**, **Elastic Load Balancing**, and **RDS (Relational Database Service)**. For example:

- **AWSServiceRoleForAutoScaling** allows the service to dynamically adjust the number of EC2 instances based on traffic patterns.
- **AWSServiceRoleForElasticLoadBalancing** handles the distribution of incoming traffic across EC2 instances.
- **AWSServiceRoleForRDS** manages secure access to the Aurora MySQL database.

In addition, functionality like **AWS Trusted Advisor** helps monitor and enhance security and performance. Special roles like **Inventory-App-Role and voclabs** are created for each application to restrict them to only accessing the necessary resources. With careful control over these roles, we limit access to key entities, reducing the risk of illegal access and creating a secure cloud environment.

6.4 High Availability & Scalability (Abdulrahman Humadi TP070609)

Multi-AZ Deployment



Availability and durability

Deployment options Info

Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the Amazon RDS service level agreement (SLA).

Multi-AZ DB cluster deployment (3 instances)
Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:
- 99.95% uptime
- Redundancy across Availability Zones
- Increased read capacity
- Reduced write latency

[✓ Multi-AZ DB instance deployment (2 instances)]
Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:
- 99.95% uptime
- Redundancy across Availability Zones

Figure 43 RDS Multi-AZ Instance Deployment (2 Instances)

The application’s backend infrastructure received added durability and fault tolerance by selecting a Multi-AZ DB instance deployment during database creation on Amazon RDS MySQL database. View the first illustration showing how this deployment method uses two instances for its setup.

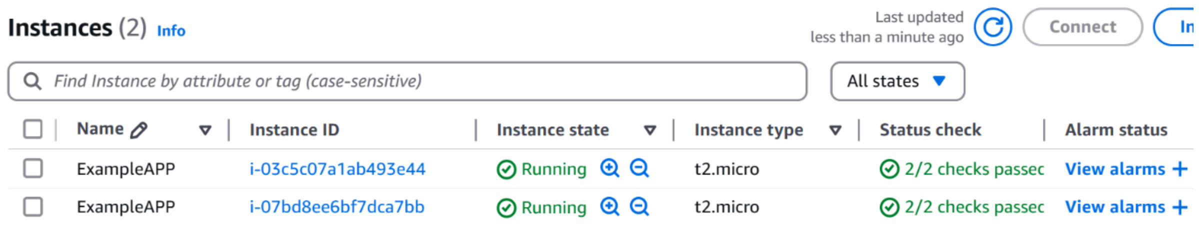
The application runs its primary database from us-east-1a Availability Zone.

Through the Multi-AZ DB instance deployment we selected two instances which include a primary database instance located in one Availability Zone (e.g., us-east-1a) together with a standby replica located in a separate Availability Zone (e.g., us-east-1b).

Synchronous replication functions between the two database instances as a result of this implementation. Every primary database write instruction instantly sends copies to the standby database that preserves real-time consistency with data protection across instances. If a failure occurs in the primary AZ AWS RDS enables an automatic failover of the standby instance that delivers applications through a brief disruption.

The chosen DB instance deployment comprises two instances while excluding readable replicas because its main purpose is redundancy combined with failover rather than read

scaling performance. The non-readable standby instance serves an essential function in achieving 99.95% uptime because of its role in the AWS SLA.
The deployment enables straightforward maintenance activities since AWS takes care of patching and backup procedures which apply to the standby instance before promotion thereby reducing disruptions to production operations.


Instances (2) Info

Last updated less than a minute ago | Connect | In

Find Instance by attribute or tag (case-sensitive) | All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
ExampleAPP	i-03c5c07a1ab493e44	Running	t2.micro	2/2 checks passec	View alarms +
ExampleAPP	i-07bd8ee6bf7dca7bb	Running	t2.micro	2/2 checks passec	View alarms +

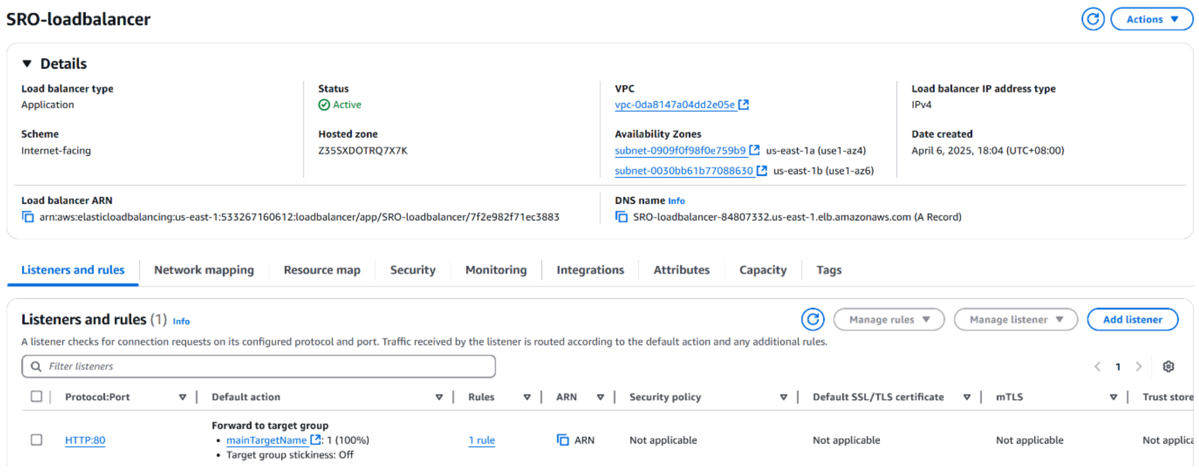
Figure 44 EC2 Instances Distributed Across Availability Zones

The application server deployment strategy uses Amazon EC2 instances spread across different Availability Zones to achieve its high availability design from the database layer. The two running instances with the name ExampleAPP use t2.micro instance type per the second screenshot. A status check ensured the operational readiness of both instances since they passed the AWS 2/2 verification process. The EC2 instances function as a part of an Auto Scaling Group (ASG) which spreads instances throughout at least two AZs. The image lacks any specific reference to AZs but redundancy remains essential for the complete HA strategy. This setup ensures that:

- If one AZ or one instance fails, the Application Load Balancer (ALB) automatically redirects traffic to healthy instances in the remaining AZ.
- The web application remains accessible even if a single point of failure occurs in one zone.
- Load is balanced across instances to improve performance and responsiveness under varying traffic conditions.

A deployment combining EC2 frontend with RDS backend in a Multi-AZ structure fulfills AWS's requirements for designing resilient systems described in its Well-Architected Framework. Users enjoy uninterrupted access to the application because its robust cloud-native design enables survival during outages.

Elastic Application Load Balancer (ALB)



SRO-loadbalancer

Actions

Details

Load balancer type Application	**Status** Active	**VPC** vpc-0da8147a04dd2e05e	**Load balancer IP address type** IPv4
Scheme Internet-facing	**Hosted zone** Z35SXDOTRQ7X7K	**Availability Zones** subnet-0909f0f98f0e759b9 us-east-1a (use1-az4); subnet-0030bb61b77088630 us-east-1b (use1-az6)	**Date created** April 6, 2025, 18:04 (UTC+08:00)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:533267160612:loadbalancer/app/SRO-loadbalancer/7f2e982f71ec3883

DNS name Info
SRO-loadbalancer-84807332.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules | Network mapping | Resource map | Security | Monitoring | Integrations | Attributes | Capacity | Tags

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust store
HTTP:80	Forward to target group • mainTargetName: 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not applicable	Not applica

Figure 45 ALB Configuration and Listener Rule

The deployment of Elastic Application Load Balancer (ALB) named SRO-loadbalancer maintains web traffic management and ensures web application availability through consistent operation. This Internet-facing Elastic Application Load Balancer acts as the main entry point through which users access the application from the public internet. SRO-loadbalancer functions as a web entrance to handle HTTP traffic while sending requests to EC2 instances located in private subnets which guarantees application tier performance alongside security boundaries.

The deployment area for the load balancer intersects both the us-east-1a and us-east-1b Availability Zones through separate public subnets. A multi-AZ configuration for the ALB stands vital for high availability purposes since it ensures continued traffic routing to instances hosted in operational Availability Zones during any zone downtime incidents. The application infrastructure maintains higher resilience through distributed failure points which decreases chances of complete failure.

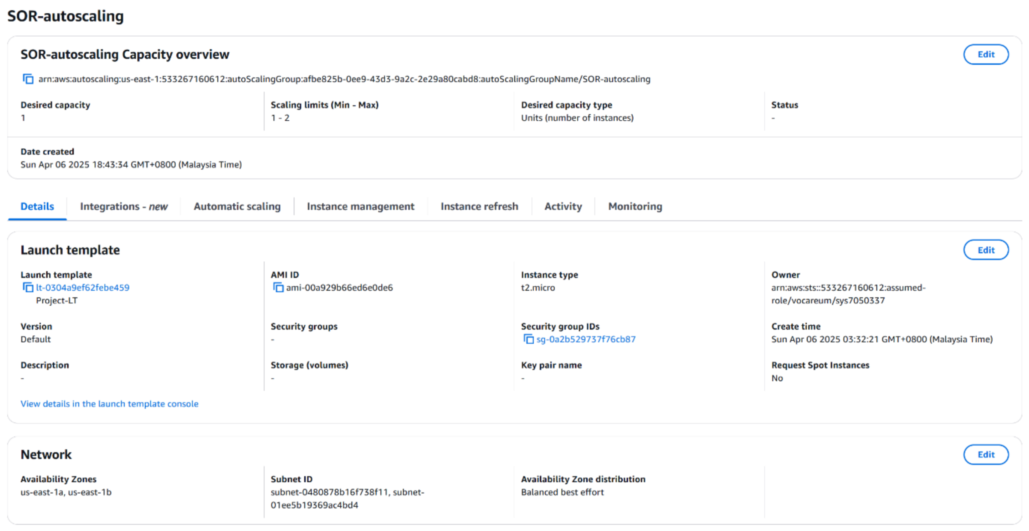
Through port 80 (HTTP) the ALB acts as a listener which maintains steady observance of incoming requests from the network. The listener directs every incoming request toward the target group named mainTargetName for forwarding. миниTargroup holds PHP application-running EC2 instances whom the ALB keeps regularly inspecting for health reliability. The health checking system prevents unhealthy instances from receiving application traffic. The pool will automatically reroute traffic to functioning instances when one is found nonhealthy or fails to meet health requirements. Failover happens automatically through this process which guarantees application responsiveness while the operating conditions change.

In this configuration:

- The entire traffic volume distributes equally (100%) among each healthy target in the group.
- Disable of sticky sessions enables the ALB to allocate traffic dynamically while preventing user bindings to particular instances.
- Application-level decision making for routing allows future upgrades such as path-based or host-based routing to be implemented.

The described Elastic Load Balancer configuration leads to better web application availability together with superior performance. The Elastic Load Balancer helps manage traffic flow during busy periods while creating protected zones and allows scaling of infrastructure through integration with Auto Scaling. Failure conditions alongside high traffic spikes do not affect application availability because the Multi-AZ EC2 deployment and RDS failover features work together with the ALB.

Auto Scaling


SOR-autoscaling

SOR-autoscaling Capacity overview

arn:aws:autoscaling:us-east-1:533267160612:autoScalingGroup:afbe825b-0ee9-43d3-9a2c-2e29a80cabd8:autoScalingGroupName/SOR-autoscaling

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
1	1 - 2	Units (number of instances)	-

Date created
Sun Apr 06 2025 18:43:34 GMT+0800 (Malaysia Time)

Launch template

Launch template lt-0304a9ef62febe459 Project-LT	**AMI ID** ami-00a929b66ed6e0de6	**Instance type** t2.micro	**Owner** arn:aws:sts::533267160612:assumed-role/vocareum/sys7050337
Version Default	**Security groups** -	**Security group IDs** sg-0a2b529737f76cb87	**Create time** Sun Apr 06 2025 03:32:21 GMT+0800 (Malaysia Time)
Description -	**Storage (volumes)** -	**Key pair name** -	**Request Spot Instances** No

View details in the launch template console

Network

Availability Zones	Subnet ID	Availability Zone distribution
us-east-1a, us-east-1b	subnet-0480878b16f738f11, subnet-01ee5b19369ac4bd4	Balanced best effort

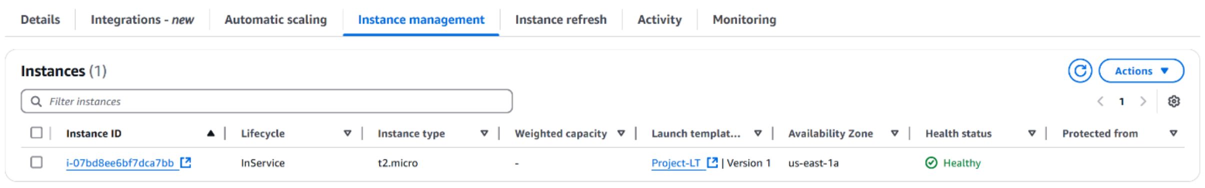
Figure 46 SRO-autoscaling Configuration

The core infrastructure foundation for application stability and performance during traffic fluctuations embraced an Auto Scaling Group named SOR-autoscaling. Amazon EC2 instance numbers automatically modify through Auto Scaling to achieve high availability and cost reduction capabilities.

The **Figure X** displays ASG configuration which sets its desired capacity at 1 alongside a scaling range between 1 and 2 instances. One instance needs to stay active at all times and the group enables automatic instance expansion to the second instance when traffic rises. This particular design absolutely excels at preserving both system quickness under traffic peaks and preventing resource waste during periods of inactivity. The EC2 instances scale in whole units because the group operates with unit count rules.

Figure X shows both the Launch Template Project-LT that the ASG employs. The predefined configuration of the group applies identically to each instance the group launches. The entire deployment of required application environment and security rules uses this template which contains the AMI ID and instance type (t2.micro) alongside security group IDs and key pair information. By using this methodology organizations achieve deployment consistency along with minimizing configuration drift which serves as a key necessity for cloud-native infrastructure deployment.

The Auto Scaling Group implements a deployment across two Availability Zones by using distribution policy us-east-1a and us-east-1b. The distribution of EC2 instances across different zones through this method protects against failures within single zones. The ASG has the capability to provision instances across available AZs as an automatic response when an outage occurs in one zone to maintain application availability.


Details | Integrations - *new* | Automatic scaling | Instance management | Instance refresh | Activity | Monitoring

Instances (1)

Instance ID	Lifecycle	Instance type	Weighted capacity	Launch templat...	Availability Zone	Health status	Protected from
i-07bd8ee6bf7dca7bb	InService	t2.micro	-	Project-LT | Version 1	us-east-1a	Healthy	

Figure 47 Healthy Instance in Instance Management

The InService lifecycle state currently applies to the instance i-07bd8ee6bf7dca7bb as shown in **Figure X+1**. The instance operates from the us-east-1a Availability Zone where it demonstrates a Healthy condition after finishing all related status and health checks. The instance status demonstrates that it operates fully to handle traffic in the load-balanced web infrastructure. The instance began through direct execution from the Project-LT launch template to uphold the designated environment specifications.

The infrastructure achieves resilience and automatic self-healing and elastic characteristics when Auto Scaling works alongside the Application Load Balancer along with Multi-AZ deployment methods. The system launches new instances to meet traffic growth automatically and eliminates unneeded ones during decreased demand requirements so there is no need for manual involvement. The system provides optimized costs with no interruption to users by maintaining operational efficiency.

Task 2: Research and justify if serverless architecture and automation will help to improvise the solution architecture in Task 1.

Case Study 1

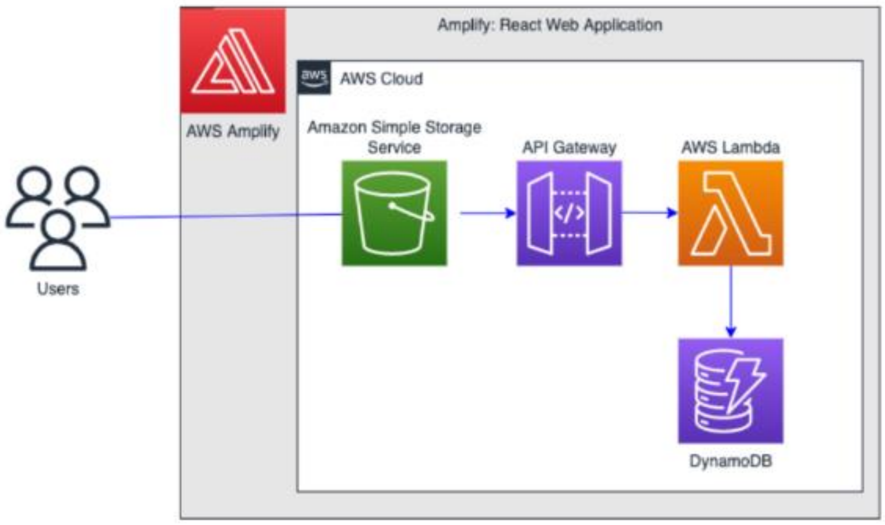


Figure 48 Full Stack Serverless Architecture

1. Introduction

Large companies have chosen cloud computing as their solution for managing increasing online requirements with greater efficiency since recent years. Serverless computing has risen into a compelling approach because it provides automatic scalability together with reduced operational overhead and cost-efficient operations (Baldini et al., 2017). Task 1 shows how the café operates with Amazon EC2 as its hosting provider but requires regular server maintenance that undermines business speed during times of fast expansion. A serverless architecture transition solves the mentioned drawbacks through automated resource management and it enhances availability during busy times (AWS, 2023).

A Serverless architecture built on Amazon Web Services (AWS) can implement its entire set of services including AWS Amplify and Amazon S3 together with AWS Lambda and API Gateway and DynamoDB. The complete services set allows businesses to react swiftly to market shifts and achieve automatic scalability based on user need variations (Roberts, Chapin, & Roberts, 2022). This research examines how a serverless system can enhance the web infrastructure of the café through a complete architecture review while identifying both obstacles and ways to maximize cloud benefits.

2. Proposed Architecture Overview

A full-stack serverless architecture creates substantial development changes because it eliminates operational complications linked to infrastructure provisioning and maintenance. A serverless approach for the café’s online business operations brings better scaling abilities combined with simplified management along with lower operational expenses. The proposed serverless architecture incorporates various AWS services which operate as system components:

- **AWS Amplify**: The framework makes frontend application deployment easier for JavaScript frameworks React together with Angular as well as other popular frameworks. The combination of Amplify and its hosting features together with continuous integration services allows the café to quickly implement changes across their customer web application through AWS (2024).
- **Amazon S3 (Simple Storage Service)**: Amazon S3 serves as a storage solution for business assets consisting of static website files like HTML and CSS and JavaScript files as well as images. AWS S3 functions as an object storage platform delivering durable and protected scalability whereas it integrates harmoniously with AWS Amplify frontends for deployment (Elger, Shanaghy, & Gopalakrishnan, 2023).
- **API Gateway**API Gateway operates as the central communication hub that receives HTTP/HTTPS front-end requests to direct them to AWS Lambda functions. Through its functionality API Gateway manages authorization tasks while controlling throttle speeds and implementing caching and performs request-response transformations thus streamlining backend operations (Eivy, 2019).
- **AWS Lambda**: Lambda functions as the serverless computing aspect which executes backend operations while eliminating the need for server provisioning. Lambda adjusts its resources automatically according to rising request levels while running code explicitly during required periods thus making it economically effective for variable workload patterns (AWS, 2024).
- **Amazon DynamoDB**: DynamoDB functions as a NoSQL database service which provides both increased speed and automated scalability and performance reliability. Application data consisting of user orders as well as profiles and transaction details has a secure management system within DynamoDB. Database queries in DynamoDB become more responsive when Lambda functions integrate seamlessly to the system (Farag & Sakr, 2023).

Architectural Interaction Flow:

Users reach the café website through a frontend running on AWS Amplify which S3 serves to them. The system activates HTTP requests through API Gateway whenever users interact with the frontend. AWS Lambda functions receive these requests to execute the backend operations that handle order processing in combination with user authentication and data retrieval. The API Gateway triggers DynamoDB database requests from Lambda functions which return responses back to API Gateway for purposes of frontend-user interaction.

The solution's architecture design allows independent scaling for its components and supports rapid dynamic responses which both prevent system failure points and decrease user wait times especially during busy times like promotional events or peak dining periods.

3. Benefits of the Architecture

An entire serverless stack deployment proves superior to conventional server-to-host solutions when focused on online business growth such as the café based enterprise explored in Task 1. The serverless architecture delivers multiple advantages such as advanced scalability features alongside improved efficiency at low cost and decreased operational need and heightened agility together with better reliability.

Scalability:

The proposed architecture demonstrates its ability to scale by nature. The architecture composed of AWS Lambda functions and API Gateway together with DynamoDB expands their resources to match customer application demand automatically. Lambda adds resources automatically upon detecting increased requests which keeps performance high during every traffic surge (Eivy, 2019). During peak café online order hours the instant throughput scaling of Amazon DynamoDB allows the system to manage thousands of concurrent transactions without experiencing performance issues (Farag & Sakr, 2023).

Cost Efficiency:

Servers without infrastructure provide cost efficiency due to usage-based payment models. Payment for traditional EC2 instances continues without restraint but AWS Lambda and API Gateway demand payment only for actual usage patterns through processed requests and calculation durations. The DynamoDB pricing system bases costs on the actual amount of read/write activities and storage taken up from customers so it provides financial benefits for businesses dealing with changing workload requirements (AWS, 2024). The café benefits from reduced operational expenses across off-peak periods together with managed peak costs through this approach.

Operational Simplicity:

Serverless technology eliminates operating difficulties because it handles infrastructure management duties behind the scenes. Programmers handle application deployment and management tasks independently of server maintenance responsibilities and security patches installation. AWS Amplify provides an easy way to deploy frontends by enabling automated CI/CD operations from version control systems (Elger et al., 2023). The reduced complexity allows IT personnel to enhance user experience and launch new features because they no longer need to spend time managing servers.

Agility and Speed of Deployment:

When AWS manages infrastructure responsibilities businesses obtain faster delivery times for their new features as well as system enhancements. Rapid iteration and continuous deployment become possible because serverless applications use modular design elements that make update and modification simple according to Roberts et al. (2022). The café adopts a rapid response capability regarding market trends and customer feedback together with seasonal promotions through its serverless framework.

Reliability and Availability:

The built-in redundancy features and fault tolerance mechanisms in serverless architectures work to boost application reliability. The data replication capabilities of AWS services DynamoDB and S3 across multiple availability zones protect the café from data loss and service interruptions (AWS, 2024). Data redundancy in the café’s online system offers continuous service availability which builds trust with customers and enhances their satisfaction.

The café's transition to serverless architecture would deliver essential features of effortless scalability and cost control along with simplified management and rapid innovation and reliability to establish competitive advantages in current fast-paced online business establishments.

4. Potential Limitations

Organizations that relocate from Amazon EC2 to full-stack serverless architecture must handle several inherent obstacles alongside their numerous benefits.

Cold Start Latency:

Computer servers operating on demand face an extensive performance slowdown challenge commonly known as "cold start" latency according to AWS Lambda users. The time delay which Lambda functions experience stems from idle periods during which they need to create a new runtime environment (Shafiei, Khonsari, & Mousavi, 2022). User experience suffers because of AWS improvements and cold start latency creates problems when applications need unvarying quick responses.

Monitoring and Debugging Complexity:

Standard monitoring and debugging approaches become complicated because serverless applications lack servers and operate in distributed platforms. Developers encounter challenges when trying to follow application logic between AWS services such as API Gateway and Lambda and DynamoDB which complicates the process of identifying system problems (Farag & Sakr, 2023). Companies can leverage AWS CloudWatch alongside X-Ray yet operational expertise and resourceful setup constitutes a challenge for proper implementation.

Vendor Lock-In:

The full implementation of serverless solutions depends on specific AWS provider services which establishes substantial dependence upon AWS infrastructure. The lock-in effect reduces migration flexibility between cloud providers which in turn may lead to higher future costs together with complex adjustments to strategic technological plans (Eivy, 2019).

Database Constraints:

The NoSQL design of DynamoDB enables high scalability along with performance excellence yet its limitations include challenging complex relational queries and transactions that become easier with traditional relational database management systems. Applications that depend on traditional relational database structures will face obstacles with their queries since DynamoDB provides limited join-operation capabilities and limited transactional features (Roberts, Chapin, & Roberts, 2022).

The restaurant can maximize serverless computing benefits and reduce these disadvantages by using best practices, and additional complementary services.

5. Enhancement Recommendations

Various strategic improvements need implementation to optimize the performance alongside security and manageability features of the full-stack serverless architecture.

AWS Cognito for Authentication and Authorization:

AWS Cognito brings advanced authentication security by enabling secure multi-region Sign-Up, Sign-In and access management capabilities. The combination of Cognito with multi-factor authentication (MFA) along with social identity providers enables secure features for dealing with customer applications through API Gateway (AWS, 2024).

AWS CloudFront for Latency Reduction:

AWS CloudFront enables enhanced delivery performance together with minimized latency for static assets that exist on Amazon S3. CloudFront utilizes its worldwide network to store website elements in cache which delivers swift responses along with uniform user experience to users worldwide. The integration reduces Lambda function invocations by providing data caching services for popular data requests which eliminates cold-start problems to enhance application performance (Goyal & Singh, 2022).

Enhanced Security with AWS WAF and AWS Shield:

The AWS Web Application Firewall (WAF) must function together with API Gateway to protect backend resources against SQL injection attacks along with preventing cross-site scripting (XSS). AWS Shield ensures strong defense against Distributed Denial of Service (DDoS) attacks that helps maintain application uptime during both traffic spikes and malicious attempts (AWS, 2023).

Advanced Monitoring with CloudWatch and X-Ray:

AWS CloudWatch and X-Ray should be properly configured for complete monitoring of a distributed serverless environment so debugging and performance tracking becomes simpler. The combination of CloudWatch real-time logging with X-Ray tracing across AWS services allows the system to monitor application health and to detect issues swiftly (Roberts, Chapin, & Roberts, 2022).

CI/CD Automation with AWS CodePipeline:

Automated testing along with deployment procedures became possible through the implementation of AWS CodePipeline alongside AWS CodeBuild for continuous integration and deployment (CI/CD). The approach minimizes human mistakes and speeds up development timelines while quickly deploying new features which improves the café's ability to meet customer requirements and adapt to market trends (Elger, Shanaghy, & Gopalakrishnan, 2023).

Through these improvements the café becomes able to handle serverless architecture built-in constraints which results in increased performance across reliability and security and UX features.

6. Conclusion

Serverless full-stack deployment architecture creates significant desired business advantages which support the expansion of the café's internet operations. AWS Amplify and Lambda along with API Gateway and DynamoDB combined with S3 allow the café to switch from EC2-based deployments and gain better scalability and cost-effectiveness with simplified operations and quickness and dependable services. Cloud-native services enable the café to handle rising customer demands effortlessly while boosting performance and lowering infrastructure costs using operating expenses-based payment.

It is vital to understand the disadvantages of serverless computing through an awareness of cold-start delay times as well as complicated monitoring procedures and dependence on specific vendors and restricted databases. The recommended AWS Cognito authentication combined with CloudFront delivery optimization and WAF and Shield protection and CloudWatch monitoring and X-Ray analysis as well as CI/CD pipelines enable the café to handle its vulnerabilities effectively.

The carefully designed serverless architecture upgrade positions the café to efficiently handle its present needs alongside strong future business expansion with superior customer service quality.

Case Study 2

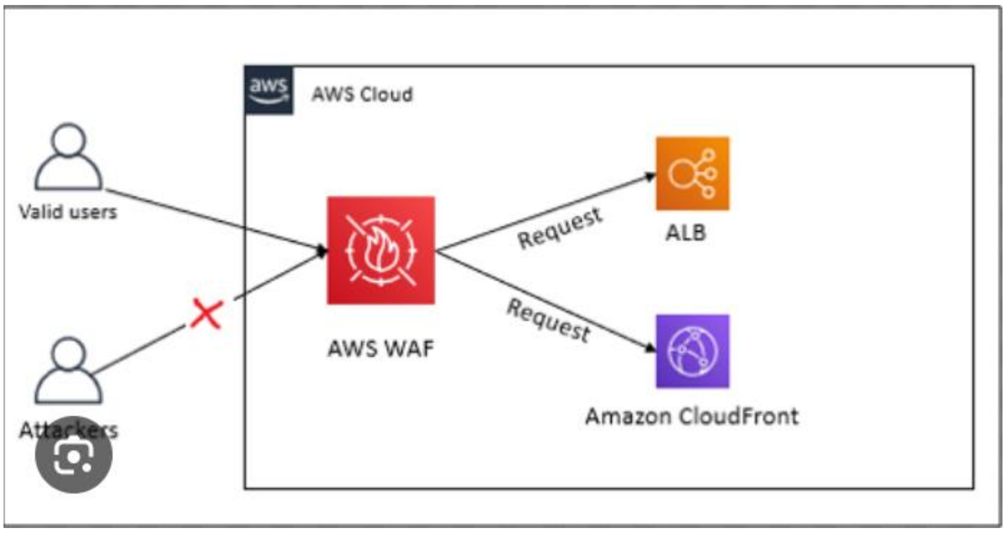


Figure 49 Cloud Security Serverless Architecture

1. Introduction

The rising number of businesses that use cloud-based operations has led to substantial changes in security threats that are now highly complex and widespread. Organizations encounter four major security threats which include Distributed Denial of Service (DDoS) attacks, injection vulnerabilities, unauthorized access, and data breaches (AWS, 2023) and need enhanced proactive security solutions. Optimal security solutions must be implemented by the café business in Task 1 for three essential purposes: customer data protection and operational uptime maintenance while building online market trust.

AWS distributes Amazon Web Services (AWS) with its complete collection of sophisticated security services that significantly improves current cloud implementations. Firms benefit from defending their web applications with AWS Web Application Firewall (WAF) and protecting against DDoS attacks with AWS Shield and implementing AWS Cognito for secure user authentication along with the auditing benefits of AWS CloudTrail (Yu Guo, & Shen, 2022). This analytical study examines how deploying AWS security services as part of an advanced security framework will strengthen the café's cloud system while enhancing its defense against cyber dangers and meeting contemporary industry security regulations.

2. Proposed Architecture Overview

The proposed advanced security platform implements various integrated security solutions from Amazon Web Services (AWS) to protect the existing cloud deployment of the café. The security architecture depends on multiple Amazon Web Service components to shield different infrastructure elements including network perimeter protection and user verification and detailed operational log tracking.

AWS Web Application Firewall (WAF):

AWS WAF operates at web application interfaces to defend against the major web attack vectors which include SQL injection and XSS and bot-driven assaults (AWS, 2023). Through WAF the café gains the ability to deploy custom security rules while accessing managed rule sets from AWS which automatically respond to new security threats. When WAF operates with the Application Load Balancer or Amazon CloudFront it blocks malicious requests prior to their delivery to backend servers thus minimizing potential server compromises.

AWS Shield:

Using AWS Shield protects the cloud infrastructure of the café by safeguarding it against Distributed Denial of Service (DDoS) attacks which disrupts online business operations. Without manual action AWS Shield automatically protects against the basic DDoS attacks that target the customers. AWS Shield Advanced provides enhanced detection capabilities along with response functionalities and visibility into sophisticated High-volume DDoS attacks which preserve service availability (AWS, 2024).

Amazon Cognito:

The Cognito platform gives users a scalable authentication system to implement secure access systems both for customer interactions and administrator access management. The service enables three main authentication approaches including MFA along with social logins as well as identity federation. Cognito integration manages user credentials together with access tokens securely which eliminates vulnerabilities that stem from faulty authentication mechanisms as reported by (Patel and Shaikh 2022).

AWS CloudTrail:

The CloudTrail solution allows complete monitoring and auditing of every AWS account activity. The system collects an extensive log of history from events to show what users did and what API calls and resource modifications and administrative changes occurred. CloudTrail combined with Amazon CloudWatch enables users to create real-time alarms that enhance incident response through improved incident response and regulatory compliance (AWS, 2023).

Architecture Interaction Flow:

Providing security for incoming user requests occurs via CloudFront and AWS WAF before malicious traffic is filtered. AWS Shield operates to defend against DDoS attacks that hit the frontend section automatically. Customer authentication processes are controlled by Amazon Cognito which ensures safe user verification for system entry. AWS CloudTrail records every activity pertaining to AWS resources which encompasses API calls plus security events together with administrative activities. CloudWatch enables the real-time monitoring of security alerts by processing log information from these logs. The café implements multiple security layers which keeps digital assets together with customer information and online business operations always protected in a way that maintains trust and fulfills regulatory requirements.

3. Benefits of the Architecture

The implementation of an advanced security architecture brings multiple crucial advantages that both build trust with stakeholders and comply with regulations for the fast-growing online services of the café. AWS's built-in security solutions provide customers with multilayer defense capabilities and real-time monitoring together with improved user protection.

Comprehensive Threat Protection:

AWS Web Application Firewall (WAF) offers top-level defense against SQL injections and XSS attacks and protects against malicious bots (AWS, 2023). The combination of AWS Shield with WAF enables the café to stay operational during severe cyber events through proactive DDoS attack defense (Yu, Guo, & Shen, 2022). The integration of these security measures provides rapid defense capabilities to stop threats thus preventing operational disruptions and financial losses in the business.

Improved User Security and Access Control:

The café can construct safe and flexible authentication and authorization systems by adopting Amazon Cognito. Cognito enables administrators to implement multi-factor authentication (MFA) and user pools in combination with social identity providers to minimize the risk from weak or stolen credentials as documented by Patel and Shaikh (2022). The combination of strong authentication mechanisms delivers improved security elements as well as greater customer trust in internet-based services.

Enhanced Monitoring and Incident Response:

AWS CloudTrail breaks down the entire cloud environment by recording every user interaction and resource modification through its complete logging system which enables transparency and identification monitoring. Real-time suspicious activity monitoring occurs through the combination of CloudTrail and Amazon CloudWatch services (AWS, 2023). The enabled features speed up incident response times since they help organizations achieve swift containment measures leading to minimized security event and breach impacts.

Regulatory Compliance and Auditing:

The security platform includes state-of-the-art features which help organizations fulfill their regulatory obligations and security recommendations. AWS CloudTrail uses auditable records to help the café fulfill compliance reporting requirements which enables security framework adherence to GDPR, HIPAA, and PCI DSS standards. The management of regulatory requirements becomes simpler because of this capability which also decreases operational costs for the café to effectively comply with regulatory standards (Alenezi, Hussain, & Jhanjhi, 2022).

Cost-Effective Security Management:

The bundled security solutions offered by AWS generate superior value compared to separate security products on the market. Both AWS Shield and WAF deliver their services by adopting price models which adjust automatically to traffic fluctuations thus making themselves affordable for organizations of every market range. Through its automated security processes the café saves valuable IT resources that are free to work on essential strategic projects instead of managing basic tasks. By implementing this advanced security architecture the café obtains robust infrastructure protection that secures business growth along with customer satisfaction.

4. Potential Limitations

The proposed security architecture enables numerous advantages yet executives should recognize potential implementation challenges together with limitations to implement it effectively.

Complexity in Configuration and Management:

Specialized knowledge and continuous management become essential when businesses deploy security solutions that include AWS WAF along with Shield and CloudTrail plus Cognito. Security rules that are improperly set or contain errors will create potential threats to the system which can result in both accidental system vulnerabilities and incorrect algorithm reactions while affecting genuine users (Alenezi, Hussain, & Jhanjhi, 2022). Continuous protection of this architecture requires both constant attention and specialized expertise.

Cost Implications of Advanced Security Services:

Advanced security services from AWS present various cost-related challenges to businesses. AWS Shield (particularly its Shield Advanced version) along with comprehensive AWS WAF implementation leads to elevated operational expenses. The café and other smaller enterprises and quickly expanding businesses need to manage their security requirements by considering their budgetary limits (AWS, 2024).

Vendor Lock-In Risks:

Strong security strategies built on AWS platforms produce higher dependency on solutions that operate exclusively within AWS architecture. Organizations that become bound to a single vendor through vendor lock-in experience restricted flexibility because migrating between cloud providers becomes difficult and expensive according to Yu et al. (2022).

Performance and Latency Impact:

Security features with WAF rule evaluations and CloudTrail logging together with real-time alerting have an effect on request-response cycles that stays within measurable but minimal levels. These brief additional delays combine into substantial impacts for user experience when they affect latency-sensitive applications (Patel & Shaikh, 2022).

Alert Fatigue and Monitoring Overhead:

Extended data collection done by CloudTrail and CloudWatch accumulates large data volumes which end up swamping security teams either leading them to overlook important threats or causing operational slowdown from information saturation (Alenezi et al., 2022).

The café needs to address its limitations by using carefully configured security measures with automated processes and regular training opportunities under strategic planning for achieving maximum security strength alongside cost-efficiency and operational ease.

5. Enhancement Recommendations

Multiple specific enhancements provide an effective solution to resolve the known weaknesses within the advanced security platform. Enhanced measures for management, cost-efficiency and system performance optimization work to deliver the maximum security strength to the café.

Automated Security Management with AWS Firewall Manager:

Through AWS Firewall Manager users gain automated management control that centrally enforces security policies to multiple accounts alongside numerous resources. Firewall Manager makes it easy for organizations to manage AWS WAF and AWS Shield policies which protects against both configuration mishaps and increases management complexity in large and expanding cloud deployments (AWS, 2023).

Intelligent Alert Management Using Amazon GuardDuty:

AWS accounts and network traffic continually receive security analysis from Amazon GuardDuty to find malicious activities. The machine learning algorithms used by GuardDuty perform automatic false positive detection which supports the creation of security alerts with high-confidence levels. With GuardDuty integration to CloudWatch finding genuine security threats becomes easier for teams to respond quickly while managing fewer erroneous security alerts (AWS, 2024).

Cost Optimization through AWS Budgets and Cost Explorer:

Through use of AWS Budgets and AWS Cost Explorer the café gains visibility to handle all expenses related to advanced security services. These tools deliver detailed cost transparency which leads to effective budget management systems and enhanced forecasting abilities and optimal expenditure control for security costs. The tool helps organizations maintain secure budgets at targeted levels through combined financial control and operational protection (AWS, 2024).

Performance Optimization with AWS CloudFront:

Through combined operation of AWS CloudFront and AWS WAF system latency caused by security features becomes minimized by delivering content from nearby caching servers to end-users. AWS CloudFront successfully integrates with request processing to deliver both improved user experience and complete security features (Goyal & Singh, 2022).

Training and Continuous Skill Development:

It is essential to provide technical professionals with continuous training as well as schedule scheduled assessment meetings about AWS security standards and recent improvements. AWS security services get optimal use through training programs that maintain security team skills in complex configuration management while minimizing misconfigurations (Alenezi, Hussain, & Jhanjhi, 2022).

The implemented enhancements deliver an effective advanced security framework which combines streamlined operations with cost management and strengthens the café's defense against emerging cyber threats.

6. Conclusion

The café achieves improved cloud security posture through its adoption of AWS WAF, AWS Shield, Amazon Cognito and AWS CloudTrail in its advanced security architecture implementation. These secure solutions provide the café with efficient web threat protection along with better regulatory conformity and clearer monitoring of cloud operations.

Although these advanced security frameworks come with notable limitations which include increased complexity, vendor lock-in risks, possible performance impacts and elevated costs the café should implement appropriate strategic solutions to these constraints. The proposed amendments of AWS Firewall Manager and GuardDuty threat detection alongside AWS CloudFront performance and continuous cost management and training procedures effectively address these issues.

The advanced cloud security architecture investment done by the café makes both the infrastructure and customer trust stronger and adds business confidence for market development and regulatory compliance.

References

1. Alenezi, M., Hussain, M., & Jhanjhi, N. Z. (2022). Cloud Computing Security: A Systematic Review. *Computers, 11*(5), 75. https://doi.org/10.3390/computers11050075
2. Amazon Web Services. (2024, September 10). *What is AWS Secrets manager? - AWS secrets manager*. AWS. https://docs.aws.amazon.com/secretsmanager/latest/userguide/intro.html
3. Amazon Web Services. (2025). *What is Amazon EC2? - amazon elastic compute cloud*. AWS. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html
4. Amazon Web Services. (2025, April 4). *Elastic IP addresses - amazon elastic compute cloud*. AWS. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html
5. Amazon Web Services. (2025, February 1). *What is elastic load balancing? - elastic load balancing*. AWS. https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html
6. Amazon Web Services. (2025, February 26). *How elastic load balancing works - elastic load balancing*. AWS. https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/how-elastic-load-balancing-works.html
7. Amazon Web Services. (2025, February 26). *What is Amazon Relational Database Service (amazon RDS)? - amazon relational database service*. AWS. https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html
8. Amazon Web Services. (2025, January 23). *What is application auto scaling? - application auto scaling*. AWS. https://docs.aws.amazon.com/autoscaling/application/userguide/what-is-application-auto-scaling.html
9. Amazon Web Services. (2025, March 24). *How amazon VPC Works - Amazon Virtual Private Cloud*. aws. https://docs.aws.amazon.com/vpc/latest/userguide/how-it-works.html
10. Amazon Web Services. (2025, March 7). *What is Amazon VPC? - amazon virtual private cloud*. aws. https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html

11. Amazon Web Services. (n.d.). *Associate elastic IP addresses with resources in your VPC - Amazon Virtual Private Cloud*. AWS. https://docs.aws.amazon.com/vpc/latest/userguide/vpc-eips.html
12. AWS. (2023). *AWS Firewall Manager*. Retrieved from https://aws.amazon.com/firewall-manager/
13. AWS. (2023). *AWS Security Best Practices*. Amazon Web Services. Retrieved from https://docs.aws.amazon.com/whitepapers/latest/aws-overview/security-best-practices.html
14. AWS. (2023). *AWS Shield: Managed DDoS Protection*. Retrieved from https://aws.amazon.com/shield/
15. AWS. (2024). *Amazon Cognito Developer Guide*. Retrieved from https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html
16. AWS. (2024). *Amazon GuardDuty: Intelligent Threat Detection.* Retrieved from https://aws.amazon.com/guardduty/
17. AWS. (2024). *AWS Budgets and Cost Management*. Retrieved from https://aws.amazon.com/aws-cost-management/aws-budgets/
18. AWS. (2024). *AWS Serverless Architectures: Best Practices.* Amazon Web Services. Retrieved from https://docs.aws.amazon.com/whitepapers/latest/serverless-architectures-best-practices/welcome.html
19. AWS. (2024). *AWS Shield: Managed DDoS Protection.* Retrieved from https://aws.amazon.com/shield/
20. Eivy, A. (2019). Be wary of the economics of "Serverless" Cloud Computing. *IEEE Cloud Computing, 6*(3), 6–12. https://doi.org/10.1109/MCC.2019.2898920
21. Elger, P., Shanaghy, E., & Gopalakrishnan, V. (2023). *Serverless Architectures on AWS: With Examples Using AWS Lambda* (2nd ed.). Manning Publications.
22. Farag, A., & Sakr, S. (2023). Serverless Computing: Concepts, Architectures, and Frameworks. In *Advances in Computers* (Vol. 132, pp. 77-111). Elsevier. https://doi.org/10.1016/bs.adcom.2023.01.004
23. flexera. (2025, January 21). *AWS Auto Scaling: Scaling EC2, ECS, RDS, and more*. Spot.io. https://spot.io/resources/aws-autoscaling/scaling-ec2-ecs-rds-and-more/
24. Goyal, S., & Singh, M. (2022). Comparative analysis of AWS CloudFront and Akamai for efficient cloud-based content delivery. *Journal of Cloud Computing, 11*(1), 35. https://doi.org/10.1186/s13677-022-00318-w

25. Guo, T. (2023, September 8). *Handling secrets with AWS Secrets manager*. GitGuardian Blog - Take Control of Your Secrets Security. https://blog.gitguardian.com/handling-secrets-with-aws-secrets-manager/
26. https://docs.aws.amazon.com/whitepapers/latest/aws-overview/security-best-practices.html
27. Patel, S., & Shaikh, F. (2022). Secure User Authentication and Authorization using AWS Cognito. *International Journal of Advanced Computer Science and Applications (IJACSA), 13*(6), 218–224. https://doi.org/10.14569/IJACSA.2022.0130626
28. Philogène, R. (2021, October 26). *Guide to AWS load balancers*. Qovery RSS. https://www.qovery.com/blog/the-complete-guide-to-aws-load-balancers/
29. Rai, A. (2024, October 1). *AWS EC2 elastic IP address pricing and cost monitoring*. Economize Cloud – Cloud Cost Optimization Software. https://www.economize.cloud/blog/ec2-elastic-ip-address-pricing/
30. Roberts, M., Chapin, J., & Roberts, S. (2022). *AWS for Solutions Architects: Design your cloud infrastructure by implementing DevOps, containers, and Amazon Web Services* (2nd ed.). Packt Publishing.
31. Shafiei, H., Khonsari, A., & Mousavi, M. R. (2022). Mitigating cold-start latency in serverless computing: A systematic literature review. *Journal of Systems and Software, 190*, 111318. https://doi.org/10.1016/j.jss.2022.111318
32. Sharma, G. (n.d.). *AWS - elastic IP - Learning-Ocean*. ocean.com. https://learning-ocean.com/tutorials/aws/aws-elastic-ip/
33. W3Schools. (n.d.). *W3schools.com*. What is Amazon AWS Auto Scaling. https://www.w3schools.com/whatis/whatis_aws_auto_scaling.asp
34. W3Schools. (n.d.). *What is AWS RDS?*. W3Schools Online Web Tutorials. https://www.w3schools.com/whatis/whatis_aws_rds.asp
35. Wright, G., Ferguson, K., & Slattery, T. (2024, June 24). *What is a subnet (subnetwork)?: Definition from TechTarget*. Search Networking. https://www.techtarget.com/searchnetworking/definition/subnet
36. Yu, X., Guo, Y., & Shen, J. (2022). A comprehensive review on cloud security challenges and solutions. *Journal of Network and Computer Applications, 205*, 103429. https://doi.org/10.1016/j.jnca.2022.103429

Appendix

Workload Matrix

Task/name	Haziq Irfan Radzali TP072306	Yazen Abobakr Ahmed Al-mehdhar TP069210	Abdulrahman Gamil Mohammed Ahmed TP071012	Abdulrahman Humadi TP070609
Task 1	25%	25%	25%	25%
Deployment of Cloud Infrastructure & Services	70%	10%	10%	10%
Data Migration	10%	70%	10%	10%
Security	10%	10%	70%	10%
High Availability & Scalability	10%	10%	10%	70%
Task 2	25%	25%	25%	25%
Documentation	25%	25%	25%	25%