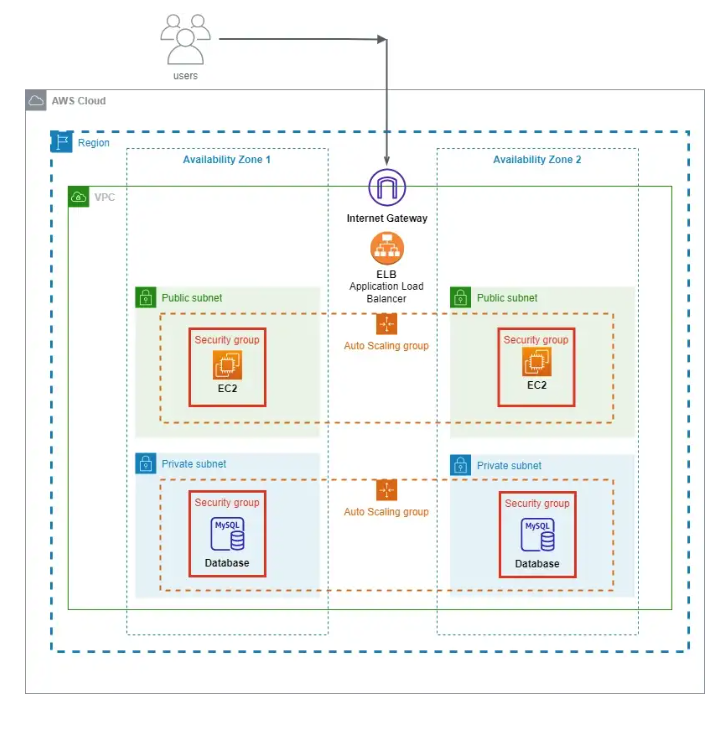
**AWS Cloud Technical Essentials  
 CAPSTONE PROJECT**

**Scenario:**

**You have a web application that accepts requests from the internet. Clients can send requests to query for data. When a request comes in, the web application queries a MySQL database and returns the data to the client.**



The Internet Gateway connected to the VPC allows clients to access the web application. The ELB receives the requests and then sends traffic to EC2 instances located in various Availability Zones. The Public Subnet contains the EC2 instances that are web-facing.

Scaling is controlled by Auto Scaling Groups tied to the instances based on changes in internet traffic volume. The Security Group associated with the Public Subnet has authority over its security. The only traffic allowed by this security group comes from the Internet Gateway.

Through the same channel, EC2 instances converse with MySQL databases located in several Availability Zones and exchange the results with the client.

The MySQL databases are protected from direct internet access through a Private Subnet.

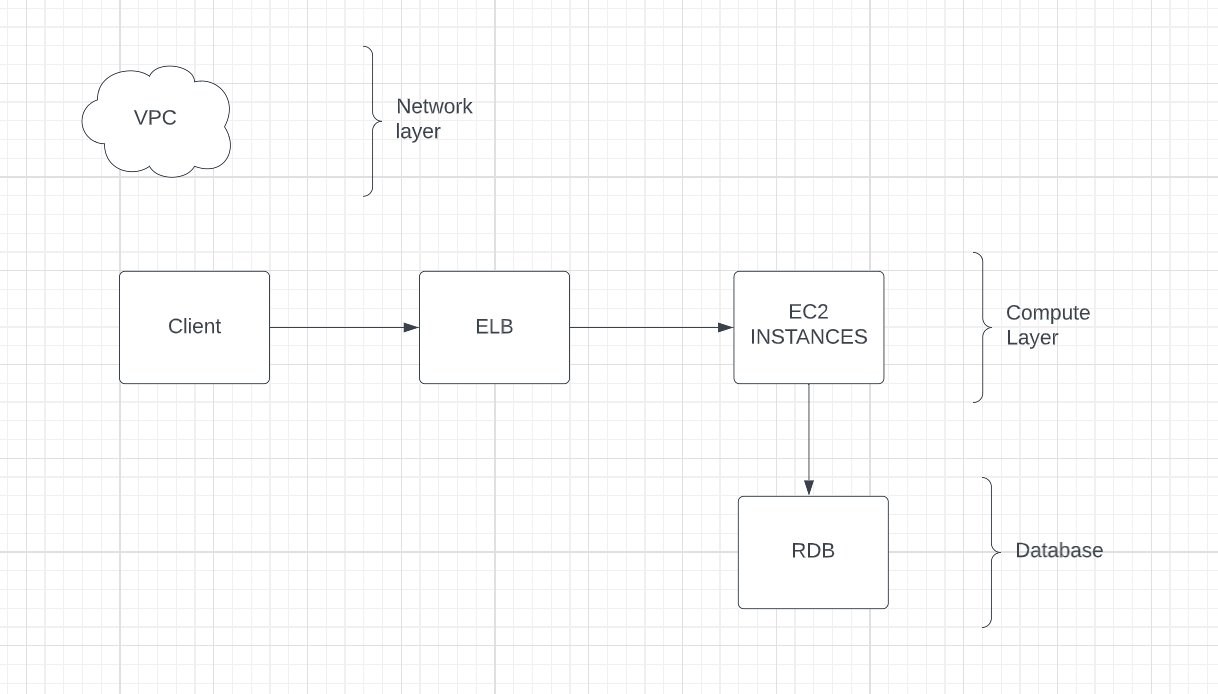
The scaling is controlled by the Auto Scaling Groups associated to the database based on changes in EC2 traffic.

The Security Group associated with the Private Subnet has authority over its security. Only traffic from the security group of the EC2 instances is permitted by this security group.

The VPS is under a single Region, and the instances are placed under two Availability Zones that are set up for high availability within the VPC.

By design, the Internet Gateway and Application Load Balancer is highly scalable. As a result, they can scale automatically based on the web traffic entering the system.

**Design a three-tier architecture that follows AWS best practices by using services such as Amazon Virtual Private Cloud (Amazon VPC), Amazon Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS) with high availability, and Elastic Load Balancing (ELB). Create an architecture diagram that lays out your design, including the networking layer, compute layer, database layer, and anything else that’s needed to accurately depict the architecture.**

****

In this design, clients communicate with the application tier through an Elastic Load Balancer (ELB), which distributes incoming traffic across multiple EC2 instances in the compute layer. This allows for high availability and scalability of the application tier, as the ELB will automatically route traffic to available instances, and new EC2 instances can be added or removed to adjust capacity as needed.

The compute layer consists of EC2 instances running the application, which in turn communicate with the database tier through Amazon Relational Database Service (RDS). RDS provides a managed database service, taking care of routine database management tasks such as backups and software patching. RDS can also be configured for high availability, with multiple instances in different Availability Zones to ensure that the database is available even in the event of a failure.

Amazon Virtual Private Cloud (VPC) is used to define the network topology for this architecture, allowing for control over the IP address range, subnets, and routing. This provides isolation and security for the resources within the VPC, as well as the ability to connect to other AWS services or existing on-premise resources.

Traffic flows from the client through the ELB to the EC2 instances in the compute layer, where the application processes the request and queries the RDS database as needed. The response is then sent back to the client through the ELB. By using these AWS services, we are able to provide a highly available and scalable architecture with robust security and performance.

**Write a few paragraphs that explain why you chose the AWS services that you used and how they would support the solution for the given scenario. Your explanation must describe how traffic flows through the different AWS components—from the client to the backend database, and back to the client.**

The AWS services chosen for this architecture were selected to provide a highly available, scalable, and secure solution for the given scenario.

Elastic Load Balancing (ELB) is used to distribute incoming traffic from clients to the application tier. This provides high availability, as the ELB can automatically route traffic to available EC2 instances, and scalability, as new EC2 instances can be added or removed to adjust capacity as needed. Traffic flows from the client to the ELB, and then to the EC2 instances.

The compute layer consists of EC2 instances running the application, which communicate with the database tier through Amazon Relational Database Service (RDS). RDS provides a managed database service, taking care of routine database management tasks such as backups and software patching. RDS can also be configured for high availability, with multiple instances in different Availability Zones, to ensure that the database is available even in the event of a failure. Traffic flows from the EC2 instances to RDS to retrieve or store data as needed.

Amazon Virtual Private Cloud (VPC) is used to define the network topology for the architecture, providing control over the IP address range, subnets, and routing. This provides isolation and security for the resources within the VPC, as well as the ability to connect to other AWS services or existing on-premise resources.

In summary, the AWS services used in this architecture work together to provide a highly available, scalable, and secure solution for the given scenario. Traffic flows from the client to the ELB, then to the EC2 instances in the compute layer, and finally to the RDS database in the database layer, with the VPC providing the network configuration and security.