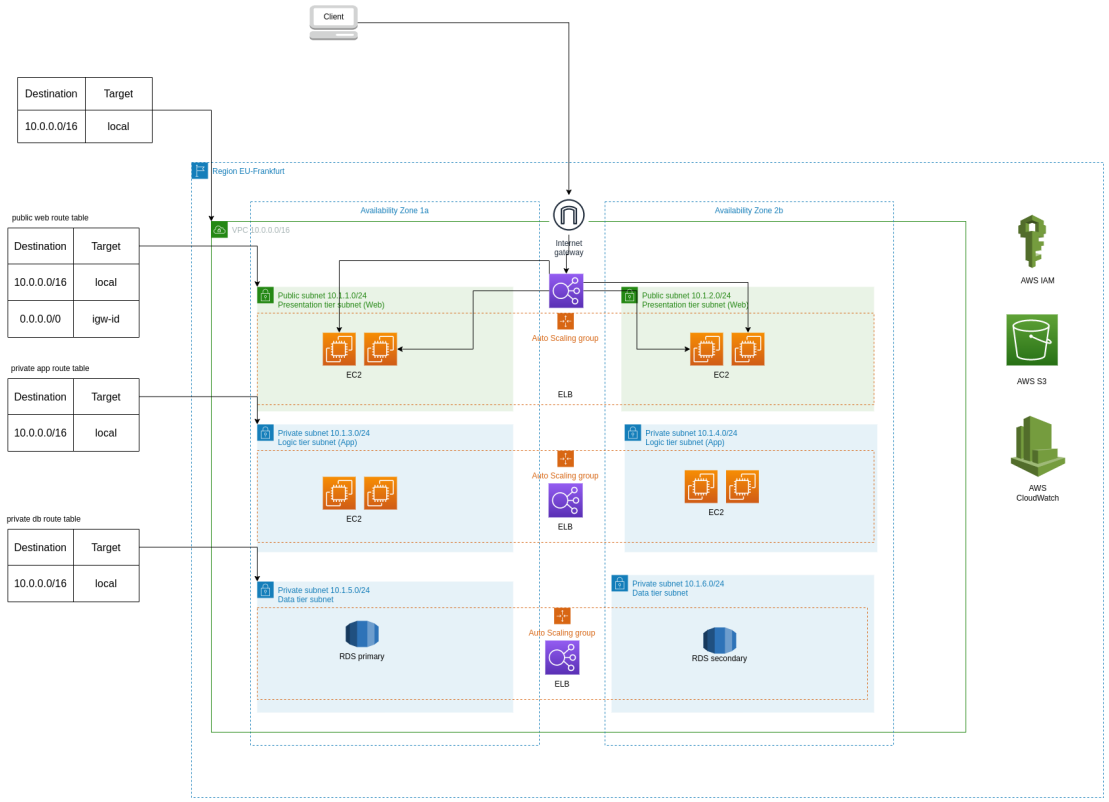
Design a three-tier architecture that follows AWS best practices by using services such as Amazon Virtual Private Cloud (Amazon VPC), Amazon Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS) with high availability, and Elastic Load Balancing (ELB). Create an architecture 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. 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.

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.

Diagram



Solution:

3 tier application:

Presentation layer:

- User-interface: HTML, CSS, JS

Application layer:

- Business/application logic

Data layer:

- Database

AWS EC2 service - is a web service that provides secure, resizable compute capacity.
In our use case we need a proven computing environment, when requests come in the application queries MySQL database should return data feedback to the client as fast as possible. When our app will be more in use or not EC2 allows us to quickly scale capacity up and down.

AWS ELB service - automatically distributes incoming application traffic across multiple targets and virtual appliances in one more AZs.

- this service simplify complex configuration and API gateways for our app
- highly available
- regional service

AWS EC2 Auto Scaling service - for this solution we define our scaling policy and based on this lets automatically add or remove EC2 instances. It is a great choice for our app because we skip the part where someone must solve this issue manually.

AWS IAM service - via this service you can specify who or what can access services and resources in AWS, centrally manage fine-grained permission.
We use this service for our app to create a policy in which users can get access for some queries about data.

AWS S3 - object storage built to retrieve any amount of data from anywhere. Object storage is assets like video, pictures and similar unstructured data, we will be use this for storage of static file: photos.
AWS CloudWatch - this service collects and visualizes real-time logs, metrics and event data in automated dashboards. This service we will use to visualize performance data and to set alarm when our resources break baseline demanding to before crash our maintenance upload some new solution hence our app will be not error-prone.

Explanation: How traffic flows through the different AWS components - from client to the backend database, and back to client, analyze access to refine permission.

In public subnets we have multiple EC2 instances hosting our app, using ELB service we are able to distribute our request across servers. How it is work? Typically requests come from the browser of the client side and are then sent to the load balancer. When load balancer gets a request, then load balancer determines which EC2 instances to send requests to. After it is sends the request, the traffic would go back through load balancer and back to the client browser

Summary:
The application is currently being hosted across multiple Amazon EC2 instances inside VPC and private subnets.
EC2 instances are part of EC2 Auto Scaling group and traffic is distributed across them using Elastic Load Balancer. Database is being hosted on Amazon RDS.