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.

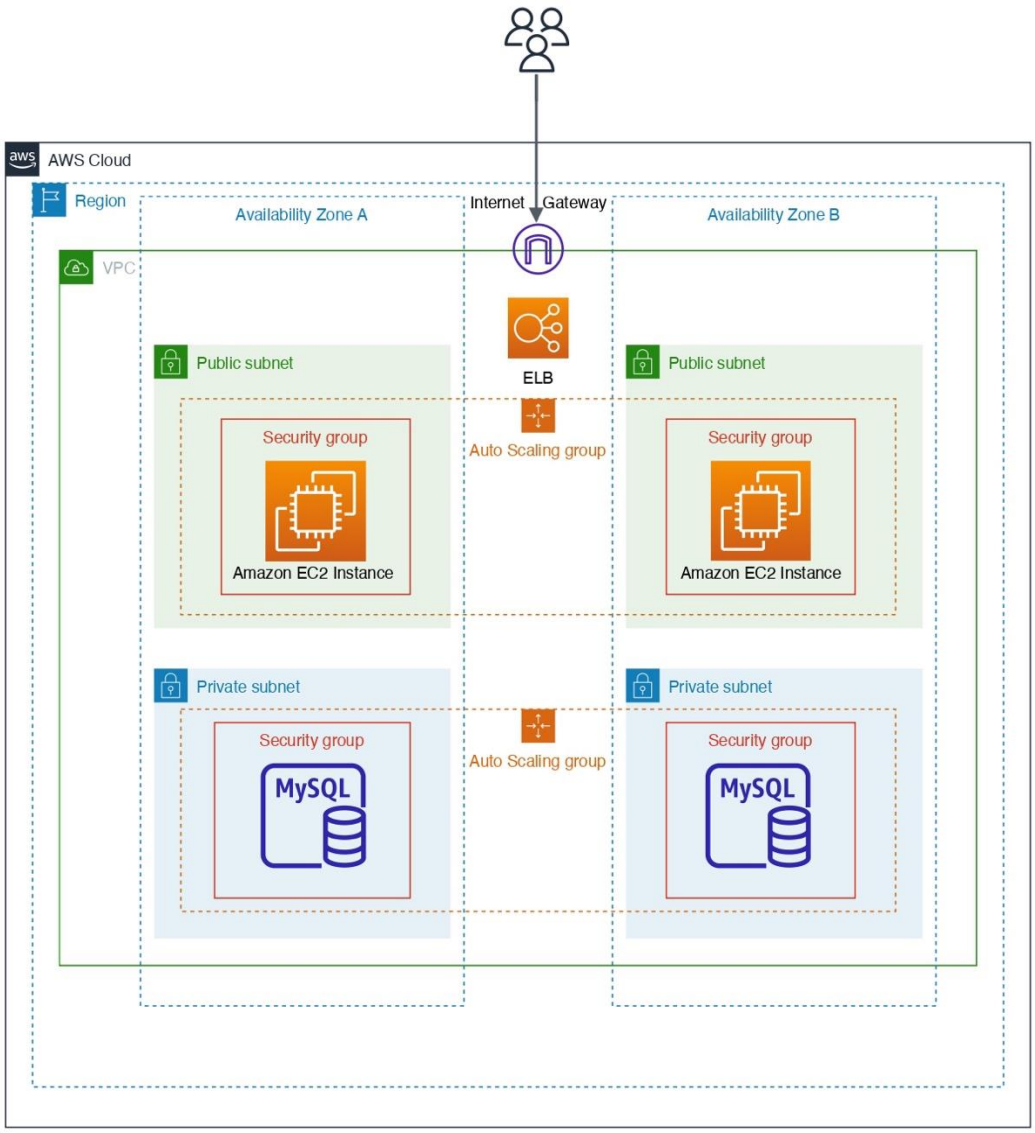
Solution:
EC2 instances are resources that host our application. They are run on distinguishable Availability Zones for refining performance and increasing an availability. Then, it is necessary to establish the Virtual Private Network (VPC). The idea of the VPC resembles to walls around a data center, it creates a boundary between our application and any internal movement. Nothing is able to come into the VPC, nothing is able to come out of the VPC without provision of a permission.

After creating a VPC, I had to build subnets inside of this network. Subnets are smaller networks inside our base network. They are deployed for high availability and ensuring diverse connectivity options. In order to maintain redundancy and fault tolerance, I create two subnets configured in two different Availability Zones. The goal of these subnets is to provide more granular controls over access to the resources. In our case, we have public resources like a web application for users, consequently it should be accessed over the internet, I can EC2 resources inside a subnet with internet connectivity. Besides that, we have more private resources like a MySQL database, that is why another private subnet was created and to keep those resources private. The access is restricted by using Security Group.

The clients can connect to the web application through the Internet Gateway (IGN) attached to the Virtual Private Cloud (VPC). Internet Gateway has a similarity with a modem, it connects our isolated Virtual Private Cloud to the internet. IGN is highly available and scalable. Also, I have used a Security Group service with customized configuration. Security Group is a layer of security that controls the traffic that is allowed to reach and leave the resources that it is associated with it, in our case with EC2 instances. It will control the inbound and outbound traffic for the instances.

Elastic Load balancing (ELB) helps to distribute the requests across all the servers hosting the application. A typical request for the application would start from the browser of the client. It will be sent to a load balancer. Then, it is sent to one of the EC2 instances that hosts the application. The return traffic would go back through the load balancer and back to the client browser. ELB can load balance to IP addresses and more importantly it automatically scales to meet the demand of the incoming traffic. It handles the incoming traffic and sends it to my backend applications.

Diagram


AWS Cloud
Region
Availability Zone A
Internet Gateway
Availability Zone B
VPC
ELB
Public subnet
Security group
Amazon EC2 Instance
Auto Scaling group
Public subnet
Security group
Amazon EC2 Instance
Private subnet
Security group
MySQL
Auto Scaling group
Private subnet
Security group
MySQL