

## Set 1

### Scenario 1

#### Context:

You are an AWS cloud engineer for a growing e-commerce platform. The platform hosts its web application on EC2 instances, and the underlying data is stored on EBS volumes. Ensuring high availability, data durability, and fast recovery from failures is critical to the business.

#### Requirements:

1. The company wants to ensure that the EC2 instances are available across multiple Availability Zones (AZs) for high availability.
2. Data stored on EBS volumes must be backed up regularly to recover from failures or data corruption.
3. The team wants to ensure there are clear cost-control mechanisms in place to avoid excessive spending on EC2 and EBS services.

#### Questions:

1. How would you set up EC2 instances across multiple AZs to ensure high availability?
2. What steps would you take to ensure regular and automated backups of EBS volumes?
3. How would you design a cost-control mechanism to avoid over-provisioning EC2 instances or EBS volumes?

### Scenario 2

#### Context:

You are tasked with deploying a highly available and scalable web application using EC2 instances. The company anticipates variable traffic, especially during peak seasons. To handle the load and ensure a seamless user experience, they want to implement a load balancing solution.

#### Requirements:

1. The web traffic should be distributed evenly among the EC2 instances in multiple Availability Zones.
2. The load balancer should automatically redirect HTTP requests to HTTPS for secure communication.
3. You need to monitor the health of the EC2 instances and automatically stop routing traffic to any unhealthy instance.

#### Questions:

1. How would you set up the Application Load Balancer (ALB) to distribute traffic across multiple EC2 instances?
2. What steps would you take to configure automatic HTTP to HTTPS redirection for the application?
3. How would you implement health checks to ensure traffic is only sent to healthy EC2 instances?

### **Scenario 3**

#### **Context:**

Your company is migrating from a monolithic application architecture to a microservices- based architecture. Each microservice runs on a separate EC2 instance, and the load varies depending on the service. The services need to scale independently and be distributed across multiple EC2 instances.

#### **Requirements:**

1. Implement load balancing for each microservice using an Elastic Load Balancer (ELB).

#### **Questions:**

1. How would you configure an Elastic Load Balancer for each microservice, and how would you ensure that the traffic is distributed effectively?