



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Experiment - 2

**Student Name:** Tanay Manish Nesari

**UID:** 23BCS13761

**Branch:** BE-CSE

**Section/Group:** KRG\_2B

**Semester:** 6<sup>th</sup>

**Date of Performance:** 14/1/26

**Subject Name:** System Design

**Subject Code:** 23CSH-314

### **Aim -**

To design a secure, scalable, and highly available system architecture using cloud computing principles and AWS services.

### **Objectives -**

- To study cloud-based system design architecture
- To understand the use of VPC, subnets, and network isolation
- To design a highly available system using load balancing
- To implement scalability using auto scaling mechanisms
- To apply security, monitoring, and storage best practices in system design

### **Procedure -**

1. Analyzed system requirements and identified architectural components.
2. Created a Virtual Private Cloud (VPC) for network isolation.
3. Configured public and private subnets within the VPC.
4. Deployed an Application Load Balancer in the public subnet.
5. Launched EC2 instances in private subnets for application hosting.
6. Enabled Auto Scaling to handle variable traffic loads.
7. Deployed a relational database in private subnets for secure data storage.
8. Configured object storage for static content and backups.
9. Enabled monitoring and logging services for system performance tracking.
10. Documented the complete architecture using a system design diagram.

### **Functional Requirements -**

- The system shall accept incoming user requests through the internet
- The system shall distribute traffic using a load balancer
- The system shall process requests using application servers
- The system shall store application data in a database
- The system shall support automatic scaling of resources
- The system shall monitor application and infrastructure performance

## **Non-Functional Requirements -**

- **Scalability:** The system must scale automatically based on demand
  - **Availability:** The system must remain available during component failure
  - **Security:** Sensitive resources must be isolated in private subnets
  - **Performance:** Load balancing should ensure efficient request handling
  - **Reliability:** Data must be stored persistently with backup support
  - **Maintainability:** System components should be modular and manageable

## **Outcome / Result -**

- A secure and scalable cloud-based system architecture was successfully designed
  - High availability was achieved through load balancing and auto scaling
  - Network isolation was implemented using public and private subnets
  - Sensitive resources were protected by deploying them in private subnets
  - Monitoring and logging enabled effective system performance tracking
  - The design follows standard cloud and system design best practices.

## Required System Design –

