**Deployment of a Three-Tier Architecture with Auto-Scaling and Secure Access**

**Objective**

The goal of this project was to design and deploy a **highly available, scalable, and secure** three-tier architecture on AWS. This architecture consists of a **frontend, backend, and database layer**, ensuring optimized traffic management, efficient resource utilization, and secure access control.

**Step 1: Setting Up the Network Infrastructure**

To establish a well-structured environment, the following AWS networking components were configured:

* **Created a Virtual Private Cloud (VPC)** to encapsulate all resources.
* **Provisioned Subnets:**
  + **2 Public Subnets**: Hosting frontend servers.
  + **1 Private Subnet**: Hosting backend and database servers for enhanced security.
* **Configured a NAT Gateway** to allow private subnet resources to access the internet while keeping them secure from external exposure.

**Step 2: Deploying Application and Database**

* **Frontend Layer:**
  + Deployed **two frontend servers** in public subnets.
  + Hosted the application UI, allowing users to submit application forms.
  + Configured an **Application Load Balancer (ALB)** to distribute incoming traffic between frontend servers.
* **Backend Layer:**
  + Deployed backend servers in a **private subnet**.
  + Ensured seamless integration between frontend and backend components.
* **Database Layer:**
  + Set up an **MSSQL database server** in the private subnet.
  + Configured database connectivity for storing and retrieving form submissions securely.

**Step 3: Implementing Auto-Scaling**

To ensure high availability and scalability, **AWS Auto Scaling** was implemented:

* **Created an EC2 Launch Template** with predefined configurations.
* **Configured Auto Scaling Group with the following parameters:**
  + **Minimum Instance Count:** 2
  + **Maximum Instance Count:** 5
  + **Scaling Policy:**
    - If **CPU utilization exceeds 60%**, new instances are launched automatically (**Scale-Up Event**).
    - If **CPU utilization drops below 40%**, instances are terminated (**Scale-Down Event**).
* **Integrated CloudWatch Alarms** to monitor CPU usage and trigger scaling actions.

**Step 4: Securing Backend Access with VPN**

To provide secure access to backend servers, **AWS VPN Access** was configured:

* Eliminated the need for SSH keys and public exposure of backend servers.
* Enabled employees to securely log in and manage backend services.

**Step 5: Domain Mapping and SSL Certificate**

To enhance accessibility and security:

* **Registered a domain in Route 53** and mapped it to the load balancer.
* **Issued an SSL/TLS Certificate** using AWS Certificate Manager (ACM) to ensure encrypted HTTPS communication.

**Conclusion**

This project successfully implemented a **highly available, scalable, and secure three-tier architecture** on AWS. By incorporating **load balancing, auto-scaling, secure backend access, and traffic encryption**, the architecture ensures **optimal performance, reliability, and security** for the deployed application.