

AWS 3-Tier Web Application Architecture

A production-style implementation of a secure, scalable, and highly available AWS 3-Tier Web Application Architecture. This project follows AWS best practices by separating the application into Web, Application, and Database layers and deploying them across multiple Availability Zones.

Architecture Overview

The architecture is divided into three logical tiers:

- Web Layer (Public Subnets) – Handles client requests and serves frontend content
- Application Layer (Private Subnets) – Processes business logic using a Flask backend
- Database Layer (Private Subnets) – Stores application data using Amazon RDS (MySQL)

All components are deployed inside a custom VPC with proper network isolation, security controls, and high availability.

Project Implementation Steps

1. Created a custom VPC with DNS hostnames and DNS resolution enabled.
2. Created six subnets across multiple Availability Zones (Web, App, Database).
3. Configured public and private route tables with Internet Gateway and NAT Gateways.
4. Implemented five security groups following least privilege access.
5. Created Route 53 hosted zone and mapped domain records.
6. Validated ACM certificate using DNS validation.
7. Created Amazon RDS (MySQL) in private subnets.
8. Launched Web Server EC2 instances with Apache installed.
9. Launched Application Server EC2 instances with Flask backend.
10. Configured secure SSH access to private instances.
11. Set up database schema and sample data.
12. Deployed backend application services.
13. Configured frontend web server files.
14. Created backend and frontend Application Load Balancers.
15. Configured Route 53 alias record to ALB.
16. Attached ACM certificate to enable HTTPS.

Tech Stack

Cloud: AWS

Compute: EC2, Auto Scaling

Networking: VPC, Subnets, Route Tables, Internet Gateway, NAT Gateway

Load Balancing: Application Load Balancer

Database: Amazon RDS (MySQL)

Security: IAM, Security Groups, ACM

DNS: Route 53

Monitoring: CloudWatch

Web Server: Apache (httpd)

Backend: Python, Flask

Operating System: Amazon Linux

Final Outcome

This project delivers a production-ready AWS 3-Tier Architecture with strong emphasis on security, scalability, high availability, and AWS best practices, making it suitable for real-world cloud and DevOps environments.