Manara SAA Final Project

Scalable Web Application with ALB and Auto Scaling

1. Project Overview

1.1 Introduction

This architecture represents a highly available and scalable web application hosted on AWS. It consists of multiple tiers to ensure better separation of concerns, performance optimization, and security. The architecture is designed to handle increased traffic efficiently by leveraging AWS services like EC2, Application Load Balancer (ALB), Auto Scaling Groups (ASG), and Amazon RDS.

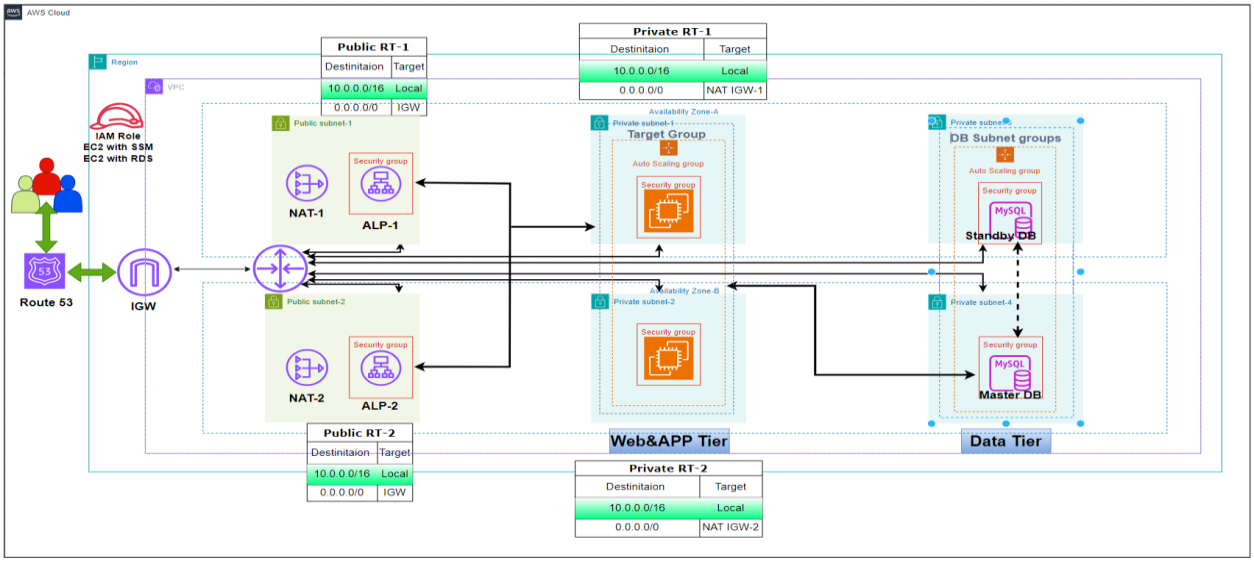
1.2 Objectives

- Implementing the infrastructure Network
- Implement an Application Load Balancer (ALB) for traffic distribution.
- Configure an Auto Scaling Group (ASG) to handle dynamic scaling based on demand.
- Deploy a secure and scalable web application on AWS.
- Optimize security and cost using IAM roles, security groups, and CloudWatch monitoring.

---

2. Solution Architecture

2.1 Architecture Diagram



2.2 Components & AWS Services Used

1. Virtual Private Cloud (VPC) – Provides network isolation and security.
2. Public & Private Subnets – Distributes workloads for security and scalability.
3. Internet Gateway & NAT Gateway – Facilitates secure communication.
4. EC2 Instances – Hosts the web application.
5. Application Load Balancer (ALB) – Distributes incoming traffic across instances.
6. Auto Scaling Group (ASG) – Ensures high availability and automatic scaling.
7. Amazon RDS (Optional) – Provides a managed relational database.
8. IAM Roles & Security Groups – Implements access control and security policies.
9. CloudWatch & SNS – Monitors performance and sends alerts.

---

3. Implementation Details

3.1 Configure VPC and Networking

Goal

- An Amazon VPC is a logically isolated virtual network you define, allowing you to launch AWS resources in a secure, isolated environment. We'll use the *VPC wizard* to quickly set up the entire virtual network for our web server, including subnets, routing, and other resources.
- Create a VPC with CIDR block 10.0.0.0/16.
- **Subnets:**
    - Public Subnets: Used for ALB and NAT Gateway.
    - Private Subnets: Used for EC2 instances and databases.
- **Route Tables:**
    - Public route table with Internet Gateway (IGW) for external access.
    - Private route table with NAT Gateway for private instances to access the internet.
- **Security Groups:**
    - ALB Security Group: Allow HTTP for inbound and Custom TCP (Target group) for outbound.
    - EC2 Security Group: Allow HTTP only for inbound and any for outbound.
    - RDS Security Group: Allows traffic only from EC2 instances.

3.2 Launch EC2 Instances

- Select an Amazon Machine Image (AMI).
- Install the web application on ec2.
- Configure security groups to allow HTTP (80).

3.3 Configure Auto Scaling Group (ASG)

- Create a Launch Template defining instance type, AMI, and security settings.
- Set up an Auto Scaling Group with min/desired/max instance capacity settings and attach it with target group.

3.4 Set Up Application Load Balancer (ALB)

- Create an ALB and configure HTTP listeners.
- Attach target group linked to Auto Scaling Group.
- Update security groups to allow HTTP for inbound and Custom TCP (Target group) for outbound traffic.

3.5 Configure IAM Roles and Security

- Assign IAM role to EC2 instances to control access to AmazonEC2RoleforSSM to access the ec2 from Systems Manager and IAM role to control access to RDS.
- Implement security groups and restrict access to necessary ports.

3.6 Implement Amazon RDS

- Create a Security Group for the RDS DB Instance: Allow *MySQL/Aurora* (Target group) for inbound and any for outbound.
- Create a DB Subnet Group
- Deploy a MySQL database with multi-AZ for high availability (Master and Standby).

3.7 Monitor & Optimize with CloudWatch

- **CloudWatch Alarms:**
    - CPU utilization, memory usage, and response time thresholds.
- **SNS Notifications:**
    - Alerts for scaling events and security threats.
- **AWS Trusted Advisor:**
    - Cost optimization, security compliance, and performance recommendations.

- **CloudWatch Logs:**
    - Log collection and analysis for troubleshooting.
- **AWS Cost Explorer & Savings Plans:**
    - Track usage and optimize EC2 costs.
- **Amazon Guard Duty & AWS Shield:**
    - Proactive threat detection and DDoS protection**.**
- **AWS Compute Optimizer:**
    - Recommendations for right-sizing instances.

4.Workflow of the Solution

1. A client sends a request to Amazon Route 53, which resolves the domain name and directs the request to the ALB.
2. The ALB distributes traffic among EC2 instances in the Web & App Tier.
3. The EC2 instance processes the request based on business logic and interacts with Amazon RDS.
4. The RDS retrieves or stores the necessary data and sends the response back to the application.
5. The response is forwarded to the client via ALB.

5.Why Were These AWS Services Chosen?

5.1 Scalability & High Availability

- EC2 Auto Scaling Groups: Ensure automatic scaling based on demand.
- Application Load Balancer (ALB): Provides redundancy and distributes traffic.
- Amazon RDS Multi-AZ: Ensures database availability and failover protection.
- Amazon Route 53: Efficiently directs client traffic to the ALB.

5.2 Security

- Private Subnets: Protect EC2 instances and databases from direct internet exposure.
- IAM Roles: Provide secure access control.
- Security Groups: Restrict unnecessary network traffic.

- NAT Gateway: Enables security to secure outbound internet access for private instances.

5.3 Monitoring & Alerts

- Amazon CloudWatch: Tracks performance metrics.
- Amazon SNS: Sends alerts when predefined thresholds are exceeded.

---

4. Conclusion

This project successfully demonstrates the deployment of a highly available, secure, and scalable web application using AWS services. The use of ALB and ASG ensures high availability, while IAM roles, security measures, and CloudWatch enhance monitoring and protection. Implementing cost optimization strategies further aligns with AWS's Well-Architected Framework, making this architecture robust and cost-efficient.
