AWS 3-Tier Web Application Deployment

Project Description

This project involves deploying a highly available and scalable 3-tier web application on AWS using EC2 instances. The architecture follows AWS best practices by leveraging Elastic Load Balancing (ALB), Auto Scaling Groups (ASG), and Amazon RDS for database management. The implementation ensures compute scalability, security, and cost optimization.

Architecture Overview

The application is structured into three tiers:

1. **Web Tier**: Public-facing layer with EC2 instances running web services, managed by an Auto Scaling Group behind an Application Load Balancer.
2. **App Tier**: Application logic layer with EC2 instances in a separate Auto Scaling Group.
3. **Data Tier**: Managed relational database using Amazon RDS with Multi-AZ deployment for high availability.

Key AWS Services Used

- **Amazon EC2**: Hosts the web and application tiers.
- **Application Load Balancer (ALB)**: Distributes incoming traffic among web instances.
- **Auto Scaling Groups (ASG)**: Ensures automatic scaling of EC2 instances based on demand.
- **Amazon RDS**: Provides a reliable, managed database service (MySQL/PostgreSQL) with Multi-AZ support.
- **IAM (Identity and Access Management)**: Manages role-based access to AWS resources.
- **CloudWatch & SNS**: Monitors system performance and sends alerts.

Network and Security Considerations

- **Virtual Private Cloud (VPC)**: Isolates application components within a controlled network environment.
- **Subnets**:
  - Public subnets for the web and application tiers.
  - Private subnets for the database tier to restrict external access.
- **Security Groups & NACLs**:
  - Web and application tiers allow traffic only from ALB.
  - Database tier accepts connections only from the application layer.
  - IAM roles restrict permissions based on the least privilege principle.

Deployment Workflow

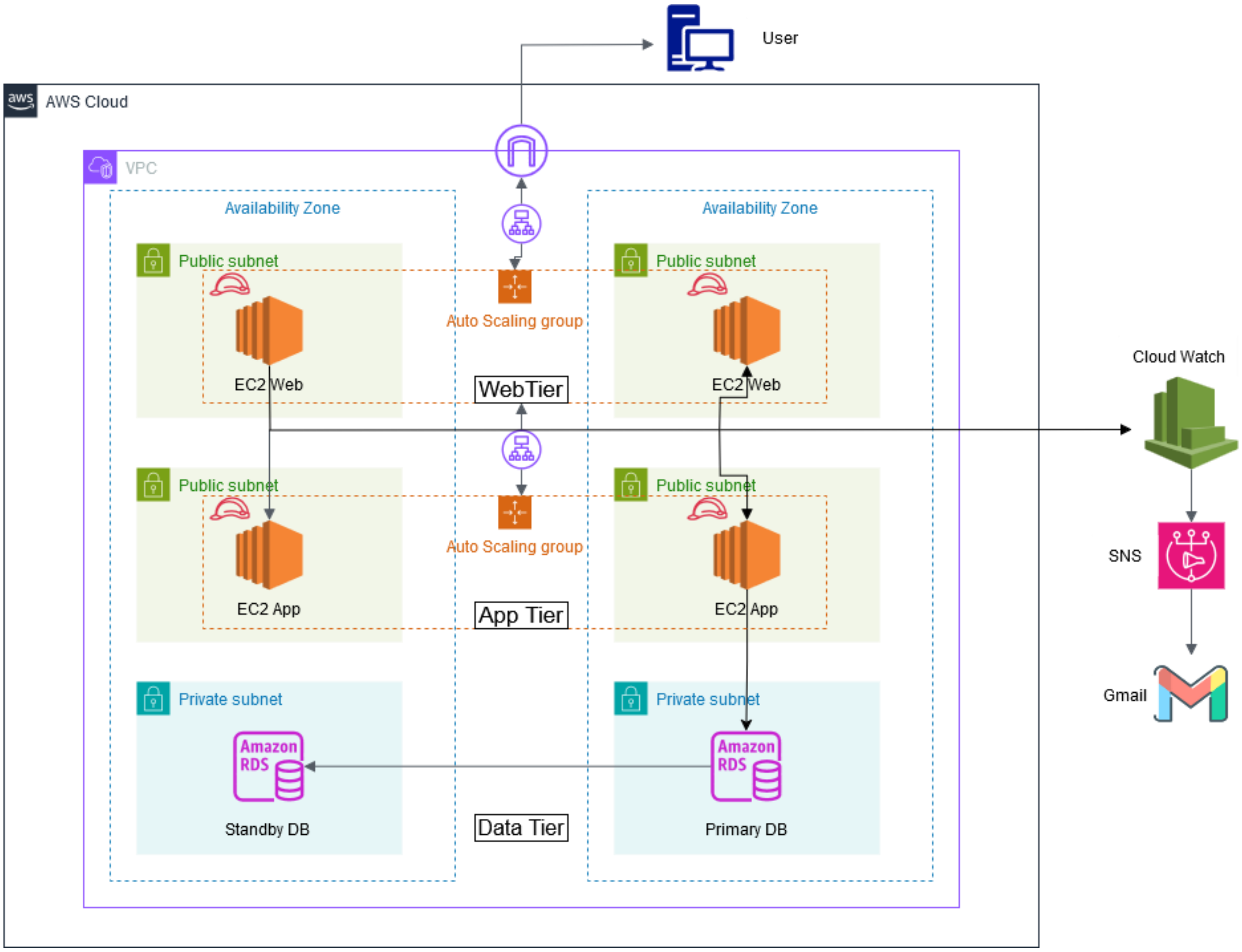
1. **Provision VPC and Subnets**: Set up private and public subnets across multiple availability zones.
2. **Launch EC2 Instances**:
   - Deploy web-tier EC2 instances behind ALB.
   - Deploy application-tier EC2 instances in a separate ASG.
3. **Configure Auto Scaling**:
   - Define policies to scale instances up or down based on CPU utilization or request traffic.
4. **Set Up Amazon RDS**:
   - Deploy a Multi-AZ database with automated failover.
   - Restrict access to the private subnet.
5. **Monitoring and Alerts**:
   - Configure CloudWatch metrics and set up SNS notifications for performance insights.

Major Benefits

- **High Availability**: ALB and ASG ensure minimal downtime.
- **Scalability**: Auto Scaling dynamically adjusts resources as needed.
- **Security**: IAM, Security Groups, and private networking enhance system protection.
- **Cost Optimization**: Resources scale up and down based on demand, reducing unnecessary expenses.

Conclusion

This AWS-based 3-tier architecture provides a reliable, secure, and cost-efficient way to deploy scalable web applications. By leveraging AWS managed services, the system achieves operational efficiency and ensures business continuity.


User
AWS Cloud
VPC
Availability Zone
Availability Zone
Public subnet
Public subnet
Auto Scaling group
EC2 Web
EC2 Web
WebTier
Cloud Watch
Public subnet
Public subnet
Auto Scaling group
EC2 App
EC2 App
App Tier
SNS
Gmail
Private subnet
Private subnet
Amazon RDS
Amazon RDS
Standby DB
Primary DB
Data Tier