CSC11006 - CLOUD IMPLEMENTATIONS AND OPERATIONS

PROJECT

Project 1: Deploying a Highly Available & Scalable E-commerce Platform on AWS

I. General information

ID:	PROJECT1
Period:	4-5 weeks
Deadline:	
Team	Group of 3 students

II. Outcome:

This lab will adapt to these following course's outcomes:

- G2.1, G2.2, G3.1, G3.2, G4.1, G5.1, G6.1

III. Describe:

Objective: This project aims to provide hands-on experience in designing, deploying, automating, and managing a robust e-commerce platform infrastructure on Amazon Web Services (AWS). Students will apply SysOps principles to ensure high availability, scalability, security, and cost-efficiency, incorporating serverless functions and caching strategies, with an emphasis on leveraging AWS Free Tier services where feasible.

Project Duration: (Specify as per course guidelines, e.g., 4-5 weeks)

Important Note on AWS Free Tier: This project encourages the use of the AWS Free Tier. However, be mindful that some services or configurations required for a robust setup (like NAT Gateways, certain Load Balancer types, RDS Proxy, Amazon Aurora, Amazon ElastiCache, or larger EC2/RDS instances if needed) may have costs beyond the Free Tier. Students should carefully track usage, set up billing alerts, and document any services that necessarily incur costs.

1. Project Overview

You will deploy a multi-tier e-commerce platform (e.g., WordPress with WooCommerce, Magento, OpenCart, or a sample Node.js/Python-based e-commerce application). The core platform will be supplemented by serverless functions (AWS Lambda) for specific backend tasks and a caching layer (Amazon ElastiCache) to optimize database performance. The

focus will be on the infrastructure setup, automation, monitoring, and operational best practices.

2. Core Requirements & AWS Services

Your deployed infrastructure should meet the following requirements, utilizing the specified AWS services, deployed across **multiple Availability Zones (AZs)** for high availability:

2.1. Network Infrastructure (Amazon VPC)

- **Custom VPC:** Design and implement a custom Virtual Private Cloud (VPC) with public and private subnets across at least two Availability Zones (AZs).
- **Network Routing:** Configure route tables for internet access (via Internet Gateway) for public subnets. For private subnets, configure a NAT Gateway for outbound internet access.
    - **Free Tier Note:** VPC is free. Internet Gateway is free. NAT Gateway has costs. Consider a NAT Instance on a Free Tier eligible EC2 instance (e.g., t2.micro/t3.micro) as a cost-saving alternative for learning, but be aware of its management overhead and lower HA compared to a NAT Gateway.
- **Security Groups & NACLs:** Implement granular security groups for each tier (web/app, database, proxy, cache, Lambda) and Network Access Control Lists (NACLs) for subnet-level security.

2.2. E-commerce Application Hosting (Amazon EC2, ELB, Auto Scaling)

- **E-commerce Application Tier:**
    - Deploy your chosen e-commerce platform's core components on Amazon EC2 instances within an Auto Scaling Group, distributed across multiple AZs.
    - Use an Application Load Balancer (ALB) to distribute traffic across instances.
    - Configure the Auto Scaling Group to automatically scale the number of EC2 instances based on demand (e.g., CPU utilization).
    - Utilize EC2 Launch Templates for consistent instance provisioning.
- **Operating System:** Choose a common Linux distribution (e.g., Amazon Linux 2, Ubuntu Server).
    - **Free Tier Note:** EC2 has a Free Tier (e.g., 750 hours of t2.micro or t3.micro per month). ALBs have costs. Auto Scaling itself is free; you pay for the EC2 instances.

2.3. Database Tier (Amazon RDS with Multi-AZ or Amazon Aurora, Read Replicas, & RDS Proxy)

- ***Option 1:*** **Managed Database (Amazon RDS with Multi-AZ)**
    - Deploy a relational database (e.g., MySQL, PostgreSQL) using Amazon RDS, configured for **Multi-AZ deployment** for high availability and automated failover.

    - **Read Replicas:** Implement one or more RDS Read Replicas (potentially across AZs) to offload read traffic from the primary database instance, improving read scalability.
    - **Free Tier Note:** RDS has a Free Tier for the primary instance (e.g., 750 hours of db.t2.micro/db.t3.micro/db.t4g.micro, some storage/backup). Read Replicas are charged as standard DB instances and may not fall under the Free Tier unless they are also micro instances and total usage remains within limits.
- ***Option 2:*** **Clustered Database (Amazon Aurora) - Recommended for Enhanced Backup/Recovery & Scalability**
    - Deploy an **Amazon Aurora cluster** (MySQL or PostgreSQL compatible). Aurora's architecture provides superior performance, availability, and durability.
    - **Aurora Replicas:** Implement one or more Aurora Replicas (up to 15) to offload read traffic. Aurora Replicas share the same underlying storage volume as the primary instance, providing very low replica lag.
    - **Backup & Recovery Benefits:**
        - **Continuous Backups:** Aurora continuously backs up your cluster volume to Amazon S3 in real-time, with no performance impact.
        - **Point-in-Time Recovery (PITR):** Allows recovery to any point in time within your backup retention period, down to the second.
        - **Fast Snapshot Creation:** Database snapshots are very fast and do not impact performance.
        - **Aurora Backtrack (MySQL compatible only):** Allows you to quickly "rewind" your database cluster to a previous point in time without needing to restore from a backup, ideal for quick recovery from logical errors.
    - **Free Tier Note:** Amazon Aurora does **not** have a persistent Free Tier like standard RDS. There might be occasional promotional credits or a limited trial. This option will generally incur costs for both primary and replica instances.
- **RDS Proxy (Applicable to both options):** Implement Amazon RDS Proxy to manage database connections efficiently, improve resiliency, and enhance security. Place the RDS Proxy in private subnets.
    - **Free Tier Note:** RDS Proxy is **not** typically part of the RDS Free Tier and will incur costs.
- **Private Subnet:** Ensure both RDS/Aurora instances/clusters, read replicas, and the RDS Proxy are in private subnets, accessible only from your application tier (and potentially bastion host for management).
- **Backups (Standard RDS):** For standard RDS Multi-AZ, configure automated backups and understand point-in-time recovery. (Aurora's backup is inherently more advanced).

2.4. Database Caching Layer (Amazon ElastiCache)

- **In-Memory Cache:** Deploy Amazon ElastiCache (using Memcached or Redis) to store frequently accessed data in memory, reducing latency and offloading requests from the database.

- **Cache Strategy:** Implement an appropriate caching strategy (e.g., cache-aside, read-through, write-through) for common queries, session data, or product catalog information.
- **Placement:** Deploy ElastiCache nodes in private subnets, accessible from your application tier. Consider multi-AZ deployment for Redis if high availability for the cache is critical.
    - **Free Tier Note:** Amazon ElastiCache offers a Free Tier for cache.t2.micro or cache.t3.micro nodes (750 hours per month for 12 months for new accounts).

2.5. Serverless Functions (AWS Lambda & API Gateway)

- **Backend Logic:** Implement specific backend functionalities for the e-commerce platform using AWS Lambda functions. Examples include:
    - Order processing (e.g., sending confirmation emails, updating inventory, invalidating cache).
    - Image resizing/processing for product images uploaded to S3.
    - User notification services.
    - Webhook integrations with third-party services.
- **API Gateway:** Expose Lambda functions as HTTP(S) endpoints using Amazon API Gateway.
    - **Free Tier Note:** Lambda has a generous Free Tier. API Gateway also has a Free Tier for the first 12 months.

2.6. Storage & Content Delivery Network (CDN - Amazon S3, Amazon CloudFront)

- **Static Content Storage (Amazon S3):** Host static assets on Amazon S3.
- **Content Delivery Network (Amazon CloudFront):** Configure Amazon CloudFront to serve static assets from S3 and potentially accelerate dynamic content.
- **Backup Storage:** Use S3 for application backups or logs. Aurora also uses S3 for its continuous backups.
    - **Free Tier Note:** S3 and CloudFront have generous Free Tiers.

2.7. Security & Identity (AWS IAM)

- **IAM Roles:** Utilize IAM roles for EC2, Lambda, RDS Proxy, ElastiCache, etc.
- **IAM Users & Groups:** Create IAM users with least privilege.
- **Password Policies:** Implement strong password policies.
    - **Free Tier Note:** IAM is free.

2.8. Monitoring & Logging (Amazon CloudWatch)

- **Metrics:** Monitor key metrics for EC2, ELB, Auto Scaling, RDS/Aurora, Read Replicas, RDS Proxy, **ElastiCache**, Lambda, API Gateway, and CloudFront.
- **Alarms:** Configure CloudWatch Alarms for critical events.
- **Dashboards:** Create a CloudWatch Dashboard.
- **Logging:** Centralize logs in CloudWatch Logs.
    - **Free Tier Note:** CloudWatch has a Free Tier.

2.9. DNS & Routing (Amazon Route 53 - Optional but Recommended)

- **Domain Name:** Configure Route 53 if you have a domain.
- **Health Checks:** Implement Route 53 health checks.
    - **Free Tier Note:** Route 53 hosted zones are not free. Some health checks have a free tier.

2.10. Infrastructure as Code (IaC - Highly Recommended)

- **Automation:** Use AWS CloudFormation or AWS Cloud Development Kit (CDK).
    - *Alternative:* Terraform.
    - **Free Tier Note:** CloudFormation/CDK are free; you pay for resources created.

3. Project Phases & Deliverables

Phase 1: Design & Planning (Week 1)

- **Architecture Diagram:** Detailed diagram including VPC, EC2, ASG, ELB, **RDS/Aurora cluster with Read Replicas, RDS Proxy, ElastiCache cluster**, Lambda, API Gateway, S3, CloudFront.
- **Service Selection Justification:** Rationale for database choice, read replicas, caching strategy, Lambda use cases.
- **Security Plan:** Include security for the cache layer.
- **Cost Estimation:** Estimate costs, highlighting Free Tier usage and potential overages (NAT Gateway, ELB, RDS Proxy, Aurora, **ElastiCache beyond free tier**). Set billing alerts.
- **Deliverable:** Design Document.

Phase 2: Implementation & Configuration (Weeks 2-3)

- VPC, IAM, EC2 & E-commerce Platform Deployment.
- ELB, Auto Scaling, **RDS/Aurora Setup (including Read Replicas)**, RDS Proxy Setup.
- **ElastiCache Cluster Deployment and Application Integration.**
- Lambda Function Development & API Gateway Configuration.
- S3 & CloudFront Configuration.
- (IaC): Develop/extend scripts.
- **Deliverable:** Working infrastructure. Progress report.

Phase 3: Monitoring, Optimization & Testing (Week 4)

- CloudWatch Setup (metrics, alarms, dashboards for all components including **ElastiCache and Read Replicas**).
- **Performance Testing:** Load testing to observe Auto Scaling, ELB, CloudFront, RDS Proxy, **database performance (primary vs. replicas), cache hit/miss ratios**, Lambda performance.

- **Failover Testing:**
    - Simulate EC2 instance/AZ failure.
    - Simulate RDS/Aurora failover (primary and replica behavior).
    - Test resilience of Lambda functions.
    - **Test ElastiCache node failure (if applicable/configured for HA).**
    - (If using Aurora) Test PITR and Backtrack.
- Security Audit.
- Cost Optimization Review.
- **Deliverable:** Monitoring setup. Test results.

Phase 4: Documentation & Presentation (Weeks 5)

- **Final Report:** Comprehensive document including:
    - Updated Architecture Diagram.
    - Implementation details (IaC templates, Lambda code, **database & cache configurations**).
    - Challenges, solutions.
    - Security, Monitoring strategy.
    - Test results (performance, failover, **cache effectiveness**).
    - Final cost analysis.
    - Lessons learned.
- **Presentation/Demo:** Live demo including e-commerce platform, infrastructure, monitoring, failover (including **database replica behavior and cache impact**).
- **Deliverable:** Final Project Report and Presentation.

4. Evaluation Criteria

- **Functionality (30%):** E-commerce platform, Lambda functions work. Infrastructure correctly configured. **Database and cache effectively integrated.**
- **High Availability & Scalability (25%):** Effective use of Multi-AZ, ELB, Auto Scaling, RDS Proxy/Aurora, **Read Replicas, ElastiCache**, CDN, Lambda scaling. Successful failover/scaling demos. **Demonstration of database recovery and cache benefits.**
- **Security (20%):** VPC, SGs/NACLs, IAM, RDS Proxy/Aurora security, **ElastiCache security**, CDN security, API Gateway security.
- **Monitoring & Operations (15%):** Comprehensive CloudWatch setup.
- **Documentation & Presentation (10%):** Clarity, completeness. Bonus for IaC, clear explanation of Free Tier, advanced database/caching features.

5. Tools & Technologies

- **Required:**
    - AWS Account (with Free Tier focus and billing alerts).
    - Chosen E-commerce Platform (and its caching integration capabilities).
    - Development tools for Lambda functions.
    - SSH Client, Web Browser.
- **Recommended:**

- AWS CLI, Version Control (Git).
- Diagramming tool, Load testing tool.
- AWS CloudFormation or AWS CDK.
- Serverless Framework.

Good Luck!