AWS Cloud Specialist

Project 1: Building a Highly Available, Scalable Web Application

Project Overview and Objectives

In this project, my objective is to build a cloud-native solution for hosting the university's student records web application. The existing on-premises system struggles during peak traffic, leading to complaints about performance and availability. To address this, I'll implement an architecture on AWS that supports thousands of concurrent users while maintaining high availability, scalability, and security. My approach will incorporate load balancing and auto-scaling to distribute traffic and handle demand spikes efficiently.

The project will proceed through several phases, starting with designing the architecture and estimating costs, followed by deploying and configuring AWS resources, and finally ensuring the application is resilient under load by using scaling strategies and security measures.

Key Deliverables and Milestones

1. **Architecture Design and Cost Estimation**
    - Develop an architecture diagram to illustrate how the web server (EC2), database (RDS), networking (VPC), and load balancing (ELB) interact.
    - Estimate the 12-month cost of running the application using the AWS Pricing Calculator, optimizing for cost-efficiency while meeting the performance and availability requirements.
2. **Deploying a Basic Web Application**
    - Set up a VPC for secure networking and launch an EC2 instance running the web application.
    - Configure an RDS MySQL instance for the database layer, decoupling the application to ensure fault tolerance and easier management.
    - Ensure the web application is fully functional by testing operations like viewing, adding, deleting, and modifying student records.
3. **Decoupling the Application Components**
    - Move the database to RDS while using Secrets Manager to manage database credentials securely.
    - Adjust the VPC to provide private subnets for the database, ensuring it is inaccessible directly from the internet.
    - Set up Cloud9 for development and testing, connecting the web application to the new database while keeping resources isolated and secure.

4. **Implementing High Availability and Scalability**
    - Deploy an Application Load Balancer (ALB) to distribute traffic across multiple EC2 instances, ensuring high availability.
    - Set up Auto Scaling based on traffic and performance metrics, allowing the infrastructure to scale in and out automatically as demand fluctuates.
    - Conduct performance and load tests to validate the solution's scalability and resilience under high traffic conditions.

Solution Requirements

- **Functional:** The web application must handle typical student records operations without noticeable delay, even during peak loads.
- **Scalable and Highly Available:** The infrastructure will be designed to dynamically scale and remain available in case of resource failures.
- **Secure:** Implement strict security measures for both the web application and database, protecting resources with IAM roles, security groups, and VPC configurations.
- **Cost-Optimized:** The solution will be built with a focus on cost-efficiency, ensuring the resources scale appropriately without unnecessary expenses.

Project 2: Securing and Monitoring Resources with AWS

Project Overview and Objectives:
The objective of this project is to enhance the security and monitoring capabilities of AnyCompany Financial Bank's AWS cloud infrastructure. With the increasing importance of data protection, network security, and compliance with regulatory standards, this initiative focuses on implementing best practices that align with the AWS Well-Architected Framework. By the end of the project, the bank will have a robust security posture, ensuring the protection of sensitive information and resources.

The project is structured into several phases, each targeting specific security and monitoring requirements. These phases include securing Amazon S3 data, implementing secure network access within VPCs, utilizing AWS Key Management Service (KMS) for data encryption, and establishing a comprehensive monitoring framework using Amazon CloudWatch and AWS Config.

Key Deliverables and Milestones

1. **Phase 1: Securing Data in Amazon S3**
    - **Bucket Creation and Policy Application:** Create an S3 bucket, apply a bucket policy, and test its effectiveness in restricting access.
    - **Versioning and Logging:** Enable versioning and object-level logging to maintain data integrity and support auditing.
    - **S3 Inventory Implementation:** Configure S3 Inventory to automate reporting on bucket contents and access.
    - **Logging Verification:** Confirm object-level logging by querying access logs with Amazon Athena for analysis.
    - **Inventory Report Review:** Analyze S3 Inventory reports using S3 Select for insights into data management.
2. **Phase 2: Securing VPCs**
    - **VPC Resource Review:** Assess LabVPC and its components to identify security gaps.
    - **Flow Log Creation:** Enable VPC flow logs to capture network traffic data for monitoring and troubleshooting.
    - **Access Review:** Test access to the WebServer instance and analyze VPC flow logs in CloudWatch.
    - **Network Security Configuration:** Adjust route tables and security groups to enhance network security.
    - **Network Firewall Implementation:** Create a network firewall to enforce security policies and filter traffic.

3. **Phase 3: Securing AWS Resources with AWS KMS**
    - **Key Creation and Rotation:** Create a customer-managed KMS key and implement key rotation policies.
    - **KMS Policy Update:** Update the KMS key policy and ensure alignment with IAM policies for compliance.
    - **Data Encryption:** Use AWS KMS to encrypt data in Amazon S3 and EBS volumes.
    - **Secrets Management:** Secure Secrets Manager secrets with AWS KMS encryption.
4. **Phase 4: Monitoring and Logging**
    - **CloudTrail Configuration:** Enable CloudTrail to log API calls for auditing and compliance.
    - **CloudWatch Logs Monitoring:** Set up CloudWatch Logs for continuous monitoring and analysis of logs.
    - **Alerting and Notifications:** Create CloudWatch alarms for detecting security incidents.
    - **Compliance Assessment:** Use AWS Config to assess compliance with security settings and remediate configuration issues.

Solution Requirements

- **Functional:** Implement security measures that allow for secure access and data integrity for AWS resources.
- **Scalable:** Ensure that the security solutions can handle increases in data volume and access requests.
- **Secure:** Utilize AWS best practices to protect sensitive information with IAM roles, security groups, and VPC configurations.
- **Cost-Optimized:** Optimize costs associated with security and monitoring services while meeting compliance requirements.

Conclusion

Upon completion of this project, AnyCompany Financial Bank will significantly improve its cloud security posture by implementing robust security measures for AWS resources, ensuring regulatory compliance, and establishing effective monitoring and incident response capabilities. This project leverages AWS services to protect sensitive data and enhance the overall security framework of the bank's cloud infrastructure.