Task - 1

Concepts:

Introduction

Getting Started

IAM & AWS CLI

AWS SDK, IAM Roles & Policies

IAM Advanced

EC2 Fundamentals

RDS + Aurora + ElastiCache

Task:

Understanding the Assessment Goals

Before we dive into the scenario, let's clarify the assessment's objectives:

- Expert Level: Candidates should demonstrate deep understanding, practical application, and troubleshooting skills.
- Independent Implementation: Candidates should be able to architect, code, and deploy a solution without significant guidance.
- Scenario-Based: The task should mimic real-world challenges.
- AWS Core Services: Focus on IAM, EC2, RDS, Aurora, and ElastiCache.

Assessment Scenario

Scenario Overview

You are a cloud architect for a rapidly growing e-commerce company. The company is experiencing increasing traffic and needs a scalable, high-performance, and secure infrastructure to handle its growing customer base. Your task is to design and implement a proof-of-concept (POC) solution to demonstrate the company's ability to handle increased load.

Task Breakdown

Task 1: Infrastructure Setup

- Create an IAM user with programmatic access and attach necessary permissions to manage EC2, RDS, ElastiCache, and IAM resources.
- Create a VPC with two public subnets and two private subnets.

- Launch an Auto Scaling group with two EC2 instances in the private subnets.
- Create a RDS MySQL database instance in a multi-AZ configuration.
- Create an ElastiCache Redis cluster with two nodes.

Task 2: Application Deployment

- Deploy a sample web application on the EC2 instances using a deployment tool of your choice (e.g., Code Deploy, Jenkins).
- Configure the application to connect to the RDS database and ElastiCache cluster.
- Implement basic load balancing using EC2 instances.

Task 3: Performance Testing and Optimization

- Simulate increased traffic using a load testing tool (e.g., Apache JMeter).
- Monitor the performance of the EC2 instances, RDS database, and ElastiCache cluster.
- Identify performance bottlenecks and implement optimizations.

Task 4: Security Best Practices

- Implement security best practices for EC2, RDS, and ElastiCache.
- Configure IAM roles and policies to restrict access to resources.
- Enable AWS security features (e.g., AWS WAF, Shield).
  - Create Ruleset to restrict traffic from specific country

Assessment Criteria

- Code Quality: Adherence to coding standards, efficiency, and readability.
- Infrastructure Design: Correctness of VPC, subnet, security group configuration.
- Performance Optimization: Ability to identify and address performance issues.
- Security: Implementation of security best practices.
- Documentation: Clear and concise documentation of the solution.

Additional Considerations

- Tooling: Candidates can use their preferred AWS CLI, SDK, or cloud formation templates.
- Complexity: The task can be adjusted in difficulty based on the candidate's experience level.

- Real-world Challenges: Introduce unexpected issues (e.g., database errors, network latency) to assess troubleshooting skills.