

Scenario 1:

You're a database administrator for a growing e-commerce company. The company uses Amazon RDS MySQL to store customer data, product information, and order details. Your task is to set up, secure, and monitor the database to ensure optimal performance and reliability.

Requirements:

1. Set up a highly available RDS MySQL instance to store critical e-commerce data.
2. Implement strong security measures to protect sensitive customer information.
3. Establish a monitoring system to track database performance and health.
4. Prepare for potential data recovery scenarios.
5. Optimize database performance for handling increased traffic during sales events. Explain the process

Questions:

1. How would you set up a highly available RDS MySQL instance?
2. Describe how you would implement IAM database authentication for RDS MySQL. What are the steps involved, and what are the benefits of using this method? Ensure the database is accessible only to the application servers and authorized personnel.
3. Design a monitoring strategy using Amazon CloudWatch. What key metrics would you track, and how would you set up alerts for potential issues? Your team wants to be immediately notified of any critical RDS events. Outline the process to set up event notifications for specific RDS events using Amazon Event Bridge (CloudWatch Events) and SNS
4. You need to create a disaster recovery solution for your RDS instance using automated snapshots. Also Copy Snapshots in any one of secondary region.
5. How would you optimize the database to handle increased load during sales events? Discuss any performance tuning techniques or scaling options you would consider. Explain the process theoretically.
6. Handle Upgrades & Patches on Weekends
7. (Optional) Utilize AWS Secrets Manager to store your DB credentials.

Scenario 2:

Your company is launching a new web application and requires a highly available and scalable infrastructure on AWS. The goal is to automatically handle traffic spikes by scaling EC2 instances up or down, distribute incoming traffic evenly across instances, and monitor the system's performance. Additionally, the system should notify the operations team via email when instances are added or removed based on CPU utilization.

Requirements:

1. Set up an EC2 instance running a web application that serves a "Hello World" page.
2. Set up an Auto Scaling Group (ASG) with a minimum of 1 and a maximum of 3 instances to manage EC2 instances for web traffic.
3. Create an Elastic Load Balancer (ELB) to evenly distribute incoming traffic.
4. Configure CloudWatch Alarms to monitor CPU utilization for scaling events.
 - a. Scale out when CPU utilization exceeds 70% for 5 minutes.
 - b. Scale in when CPU utilization drops below 30% for 5 minutes.
5. Ensure the infrastructure automatically scales to handle varying traffic loads.

Scenario 3:

Your company is developing a custom web application that requires hosting on AWS using EC2. To ensure effective monitoring and troubleshooting, the application needs to push its logs to CloudWatch. Additionally, you want to capture and monitor custom metrics from the application to gain insights into its performance.

Your task is to create an EC2 instance running this custom application, configure it to push its logs to CloudWatch, and set up custom metrics for monitoring.

Requirements:

1. Set up an EC2 instance running a custom web application.
2. Install and configure the CloudWatch Agent to push application logs to CloudWatch.
3. Modify the application to send custom metrics (e.g., Memory Utilization and errors) to CloudWatch.
4. Set up CloudWatch Alarms to monitor custom metrics and trigger alerts for any issues or if it exceeds threshold.
5. Optional (Implement SNS notifications to alert the team when alarms are triggered)
6. Validate that logs and metrics are correctly displayed in CloudWatch.
7. Make sure CloudWatch logs are encrypted & set to auto expire after 7 Days.