# Free Questions for SAA-C03

## Shared by Eaton on 04-10-2024

**For More Free Questions and Preparation Resources**

**Check the Links on Last Page**

# Question 1

**Question Type:** MultipleChoice

A company has applications that run in an organization in AWS Organizations. The company outsources operational support of the applications. The company needs to provide access for the external support engineers without compromising security.

The external support engineers need access to the AWS Management Console. The external support engineers also need operating system access to the company's fleet of Amazon EC2 instances that run Amazon Linux in private subnets.

Which solution will meet these requirements MOST securely?

## Options:

**A-** Confirm that AWS Systems Manager Agent (SSM Agent) is installed on all instances. Assign an instance profile with the necessary policy to connect to Systems Manager. Use AWS 1AM Identity Center to provide the external support engineers console access. Use Systems Manager Session Manager to assign the required permissions.

**B-** Confirm that AWS Systems Manager Agent {SSM Agent) is installed on all instances. Assign an instance profile with the necessary policy to connect to Systems Manager. Use Systems Manager Session Manager to provide local 1AM user credentials in each AWS account to the external support engineers for console access.

**C-** Confirm that all instances have a security group that allows SSH access only from the external support engineers source IP address ranges. Provide local 1AM user credentials in each AWS account to the external support engineers for console access. Provide each external support engineer an SSH key pair to log in to the application instances.

**D-** Create a bastion host in a public subnet. Set up the bastion host security group to allow access from only the external engineers' IP address ranges Ensure that all instances have a security group that allows SSH access from the bastion host. Provide each external support engineer an SSH key pair to log in to the application instances. Provide local account 1AM user credentials to the engineers for console access.

## Answer:

A

## Explanation:

This solution provides the most secure access for external support engineers with the least exposure to potential security risks.

AWS Systems Manager (SSM) and Session Manager: Systems Manager Session Manager allows secure and auditable access to EC2 instances without the need to open inbound SSH ports or manage SSH keys. This reduces the attack surface significantly. The SSM Agent must be installed and configured on all instances, and the instances must have an instance profile with the necessary IAM permissions to connect to Systems Manager.

IAM Identity Center: IAM Identity Center provides centralized management of access to the AWS Management Console for external support engineers. By using IAM Identity Center, you can control console access securely and ensure that external engineers have the appropriate permissions based on their roles.

Why Not Other Options?:

Option B (Local IAM user credentials): This approach is less secure because it involves managing local IAM user credentials and does not leverage the centralized management and security benefits of IAM Identity Center.

Option C (Security group with SSH access): Allowing SSH access opens up the infrastructure to potential security risks, even when restricted by IP addresses. It also requires managing SSH keys, which can be cumbersome and less secure.

Option D (Bastion host): While a bastion host can secure SSH access, it still requires managing SSH keys and opening ports. This approach is less secure and more operationally intensive compared to using Session Manager.

AWS Reference:

AWS Systems Manager Session Manager - Documentation on using Session Manager for secure instance access.

AWS IAM Identity Center - Overview of IAM Identity Center and its capabilities for managing user access.

# Question 2

**Question Type:** **MultipleChoice**

A company is building a cloud-based application on AWS that will handle sensitive customer dat

a. The application uses Amazon RDS for the database. Amazon S3 for object storage, and S3 Event Notifications that invoke AWS Lambda for serverless processing.

The company uses AWS 1AM Identity Center to manage user credentials. The development, testing, and operations teams need secure access to Amazon RDS and Amazon S3 while ensuring the confidentiality of sensitive customer data. The solution must comply with the principle of least privilege.

Which solution meets these requirements with the LEAST operational overhead?

## Options:

**A-** Use 1AM roles with least privilege to grant all the teams access. Assign 1AM roles to each team with customized 1AM policies defining specific permission for Amazon RDS and S3 object access based on team responsibilities.

**B-** Enable 1AM Identity Center with an Identity Center directory. Create and configure permission sets with granular access to Amazon RDS and Amazon S3. Assign all the teams to groups that have specific access with the permission sets.

**C-** Create individual 1AM users for each member in all the teams with role-based permissions. Assign the 1AM roles with predefined policies for RDS and S3 access to each user based on user needs. Implement 1AM Access Analyzer for periodic credential evaluation.

**D-** Use AWS Organizations to create separate accounts for each team. Implement cross-account 1AM roles with least privilege Grant specific permission for RDS and S3 access based on team roles and responsibilities.

## Answer:

B

## Explanation:

This solution allows for secure and least-privilege access with minimal operational overhead.

IAM Identity Center: AWS IAM Identity Center (formerly AWS SSO) enables you to centrally manage access to multiple AWS accounts and applications. By using IAM Identity Center, you can assign permission sets that define what users or groups can access, ensuring that only necessary permissions are granted.

Permission Sets: Permission sets in IAM Identity Center allow you to define granular access controls for specific services, such as Amazon RDS and S3. You can tailor these permissions to meet the needs of different teams, adhering to the principle of least privilege.

Group Management: By assigning users to groups and associating those groups with specific permission sets, you reduce the complexity and overhead of managing individual IAM roles and policies. This method also simplifies compliance and audit processes.

Why Not Other Options?:

Option A (IAM roles): While IAM roles can provide least-privilege access, managing multiple roles and policies across teams increases operational overhead compared to using IAM Identity Center.

Option C (Individual IAM users): Managing individual IAM users and roles can be cumbersome and does not scale well compared to group-based management in IAM Identity Center.

Option D (AWS Organizations with cross-account roles): Creating separate accounts and cross-account roles adds unnecessary complexity and overhead for this use case, where IAM Identity Center provides a more straightforward solution.

AWS Reference:

AWS IAM Identity Center - Overview and best practices for using IAM Identity Center.

Managing Access Permissions Using IAM Identity Center - Guide on creating and managing permission sets for secure access.

# Question 3

**Question Type:** MultipleChoice

A company is designing an application on AWS that processes sensitive dat

a. The application stores and processes financial data for multiple customers.

To meet compliance requirements, the data for each customer must be encrypted separately at rest by using a secure, centralized key management solution. The company wants to use AWS Key Management Service (AWS KMS) to implement encryption.

Which solution will meet these requirements with the LEAST operational overhead'?

## Options:

**A-** Generate a unique encryption key for each customer. Store the keys in an Amazon S3 bucket. Enable server-side encryption.

**B-** Deploy a hardware security appliance in the AWS environment that securely stores customer-provided encryption keys. Integrate the security appliance with AWS KMS to encrypt the sensitive data in the application.

**C-** Create a single AWS KMS key to encrypt all sensitive data across the application.

**D-** Create separate AWS KMS keys for each customer's data that have granular access control and logging enabled.

## Answer:

D

## Explanation:

This solution meets the requirement of encrypting each customer's data separately with the least operational overhead by leveraging AWS Key Management Service (KMS).

Separate AWS KMS Keys: By creating separate KMS keys for each customer, you can ensure that each customer's data is encrypted with a unique key. This approach satisfies the compliance requirement for separate encryption and provides fine-grained control over access to the keys.

Granular Access Control: AWS KMS allows you to define key policies and use IAM policies to grant specific permissions to the keys. This ensures that only authorized users or services can access the keys, thereby maintaining the principle of least privilege.

Logging and Monitoring: AWS KMS integrates with AWS CloudTrail, which logs all key usage and management activities. This provides an audit trail that is essential for meeting compliance requirements.

Why Not Other Options?:

Option A (Store keys in S3): Storing keys in S3 is not recommended because it does not provide the same level of security, access control, or integration with AWS services as KMS does.

Option B (Hardware security appliance): Deploying a hardware security appliance adds significant operational overhead and complexity, which is unnecessary given that KMS already provides a secure and centralized key management solution.

Option C (Single KMS key for all data): Using a single KMS key does not meet the requirement of encrypting each customer's data separately.

AWS Reference:

AWS Key Management Service (KMS) - Overview of KMS, its features, and best practices for key management.

Using AWS KMS for Multi-Tenant Applications - Guidance on how to design applications using KMS for multi-tenancy.

# Question 4

**Question Type:** MultipleChoice

A company currently runs an on-premises stock trading application by using Microsoft Windows Server. The company wants to migrate the application to the AWS Cloud. The company needs to design a highly available solution that provides low-latency access to block storage across multiple Availability Zones. Which solution will meet these requirements with the LEAST implementation effort?

## Options:

**A-** Configure a Windows Server cluster that spans two Availability Zones on Amazon EC2 instances. Install the application on both cluster nodes. Use Amazon FSx for Windows File Server as shared storage between the two cluster nodes.

**B-** Configure a Windows Server cluster that spans two Availability Zones on Amazon EC2 instances. Install the application on both cluster nodes Use Amazon Elastic Block Store (Amazon EBS) General Purpose SSD (gp3) volumes as storage attached to the EC2 instances. Set up application-level replication to sync data from one EBS volume in one Availability Zone to another EBS volume in the second Availability Zone.

**C-** Deploy the application on Amazon EC2 instances in two Availability Zones Configure one EC2 instance as active and the second EC2 instance in standby mode. Use an Amazon FSx for NetApp ONTAP Multi-AZ file system to access the data by using Internet Small Computer Systems Interface (iSCSI) protocol.

**D-** Deploy the application on Amazon EC2 instances in two Availability Zones. Configure one EC2 instance as active and the second EC2 instance in standby mode. Use Amazon Elastic Block Store (Amazon EBS) Provisioned IOPS SSD (io2) volumes as storage attached to the EC2 instances. Set up Amazon EBS level replication to sync data from one io2 volume in one Availability Zone to another io2 volume in the second Availability Zone.

## Answer:

A

## Explanation:

This solution is designed to provide high availability and low-latency access to block storage across multiple Availability Zones with minimal implementation effort.

Windows Server Cluster Across AZs: Configuring a Windows Server Failover Cluster (WSFC) that spans two Availability Zones ensures that the application can failover from one instance to another in case of a failure, meeting the high availability requirement.

Amazon FSx for Windows File Server: FSx for Windows File Server provides fully managed Windows file storage that is accessible via the SMB protocol, which is suitable for Windows-based applications. It offers high availability and can be used as shared storage between the cluster nodes, ensuring that both nodes have access to the same data with low latency.

Why Not Other Options?:

Option B (EBS with application-level replication): This requires complex configuration and management, as EBS volumes cannot be directly shared across AZs. Application-level replication is more complex and prone to errors.

Option C (FSx for NetApp ONTAP with iSCSI): While this is a viable option, it introduces additional complexity with iSCSI and requires more specialized knowledge for setup and management.

Option D (EBS with EBS-level replication): EBS-level replication is not natively supported across AZs, and setting up a custom replication solution would increase the implementation effort.

AWS Reference:

Amazon FSx for Windows File Server - Overview and benefits of using FSx for Windows File Server.

Windows Server Failover Clustering on AWS - Guide on setting up a Windows Server cluster on AWS.

# Question 5

**Question Type:** **MultipleChoice**

A company is running a media store across multiple Amazon EC2 instances distributed across multiple Availability Zones in a single VPC. The company wants a high-performing solution to share data between all the EC2 instances, and prefers to keep the data within the VPC only.

What should a solutions architect recommend?

## Options:

**A-** Create an Amazon S3 bucket and call the service APIs from each instance's application.

**B-** Create an Amazon S3 bucket and configure all instances to access it as a mounted volume.

**C-** Configure an Amazon Elastic Block Store (Amazon EBS) volume and mount it across all instances.

**D-** Configure an Amazon Elastic File System (Amazon EFS) file system and mount It across all instances.

## Answer:

D

## Explanation:

Amazon Elastic File System (EFS) is a managed file storage service that can be mounted across multiple EC2 instances. It provides a scalable and high-performing solution to share data among instances within a VPC.

High Performance: EFS provides scalable performance for workloads that require high throughput and IOPS. It is particularly well-suited for applications that need to share data across multiple instances.

Ease of Use: EFS can be easily mounted on multiple instances across different Availability Zones, providing a shared file system accessible to all the instances within the VPC.

Security: EFS can be configured to ensure that data remains within the VPC, and it supports encryption at rest and in transit.

Why Not Other Options?:

Option A (Amazon S3 bucket with APIs): While S3 is excellent for object storage, it is not a file system and does not provide the low-latency access required for shared data between instances.

Option B (S3 bucket as a mounted volume): S3 is not designed to be mounted as a file system, and this approach would introduce unnecessary complexity and latency.

Option C (EBS volume shared across instances): EBS volumes cannot be attached to multiple instances simultaneously. It is not designed to be shared across instances like EFS.

AWS Reference:

Amazon EFS - Overview of Amazon EFS and its features.

Best Practices for Amazon EFS - Recommendations for using EFS with multiple instances.

## Question 6

**Question Type:** MultipleChoice

A company is developing a new application that uses a relational database to store user data and application configurations. The company expects the application to have steady user growth. The company expects the database usage to be variable and read-heavy, with occasional writes.

The company wants to cost-optimize the database solution. The company wants to use an AWS managed database solution that will provide the necessary performance.

Which solution will meet these requirements MOST cost-effectively?

## Options:

**A-** Deploy the database on Amazon RDS. Use Provisioned IOPS SSD storage to ensure consistent performance for read and write operations.

**B-** Deploy the database on Amazon Aurora Serveriess to automatically scale the database capacity based on actual usage to accommodate the workload.

**C-** Deploy the database on Amazon DynamoDB. Use on-demand capacity mode to automatically scale throughput to accommodate the workload.

**D-** Deploy the database on Amazon RDS Use magnetic storage and use read replicas to accommodate the workload

## Answer:

B

## Explanation:

Amazon Aurora Serverless is a cost-effective, on-demand, autoscaling configuration for Amazon Aurora. It automatically adjusts the database's capacity based on the current demand, which is ideal for workloads with variable and unpredictable usage patterns. Since the application is expected to be read-heavy with occasional writes and steady growth, Aurora Serverless can provide the necessary performance without requiring the management of database instances.

Cost-Optimization: Aurora Serverless only charges for the database capacity you use, making it a more cost-effective solution compared to always running provisioned database instances, especially for workloads with fluctuating demand.

Scalability: It automatically scales database capacity up or down based on actual usage, ensuring that you always have the right amount of resources available.

Performance: Aurora Serverless is built on the same underlying storage as Amazon Aurora, providing high performance and availability.

Why Not Other Options?:

Option A (RDS with Provisioned IOPS SSD): While Provisioned IOPS SSD ensures consistent performance, it is generally more expensive and less flexible compared to the autoscaling nature of Aurora Serverless.

Option C (DynamoDB with On-Demand Capacity): DynamoDB is a NoSQL database and may not be the best fit for applications requiring relational database features.

Option D (RDS with Magnetic Storage and Read Replicas): Magnetic storage is outdated and generally slower. While read replicas help with read-heavy workloads, the overall performance might not be optimal, and magnetic storage doesn't provide the necessary

performance.

AWS Reference:

Amazon Aurora Serverless - Information on how Aurora Serverless works and its use cases.

Amazon Aurora Pricing - Details on the cost-effectiveness of Aurora Serverless.

# Question 7

**Question Type:** MultipleChoice

An ecommerce company runs Its application on AWS. The application uses an Amazon Aurora PostgreSQL cluster in Multi-AZ mode for the underlying database. During a recent promotional campaign, the application experienced heavy read load and write load. Users experienced timeout issues when they attempted to access the application.

A solutions architect needs to make the application architecture more scalable and highly available.

Which solution will meet these requirements with the LEAST downtime?

## Options:

**A-** Create an Amazon EventBndge rule that has the Aurora cluster as a source. Create an AWS Lambda function to log the state change events of the Aurora cluster. Add the Lambda function as a target for the EventBndge rule Add additional reader nodes to fail over to.

**B-** Modify the Aurora cluster and activate the zero-downtime restart (ZDR) feature. Use Database Activity Streams on the cluster to track the cluster status.

**C-** Add additional reader instances to the Aurora cluster Create an Amazon RDS Proxy target group for the Aurora cluster.

**D-** Create an Amazon ElastiCache for Redis cache. Replicate data from the Aurora cluster to Redis by using AWS Database Migration Service (AWS DMS) with a write-around approach.

## Answer:

C

## Explanation:

This solution directly addresses the scalability and high availability requirements with minimal downtime.

Additional Reader Instances: Adding more reader instances to the Aurora cluster will distribute the read load, improving the performance of the application under heavy read traffic. Aurora reader instances automatically replicate the data from the writer instance, enabling you to scale out read operations.

Amazon RDS Proxy: RDS Proxy improves database availability by managing database connections more efficiently and providing a connection pool. This reduces the overhead on the Aurora cluster during peak loads, further enhancing performance and availability without requiring changes to the application code.

Why Not Other Options?:

Option A (EventBridge and Lambda): This doesn't directly address the performance and availability issues. Logging state changes and adding reader nodes on failure events doesn't provide proactive scalability.

Option B (Zero-Downtime Restart and Activity Streams): Zero-Downtime Restart (ZDR) is useful for minimizing downtime during maintenance but doesn't directly improve scalability. Database Activity Streams are more for security monitoring than for performance enhancement.

Option D (ElastiCache for Redis): While adding a caching layer can help with read performance, it introduces complexity and may not be necessary if additional reader instances can handle the load.

AWS Reference:

Amazon Aurora Scaling - Information on scaling Aurora clusters with reader instances.

Amazon RDS Proxy - Details on how RDS Proxy can improve database performance and availability.

# Question 8

**Question Type:** **MultipleChoice**

An ecommerce company wants a disaster recovery solution for its Amazon RDS DB instances that run Microsoft SQL Server Enterprise Edition. The company's current recovery point objective (RPO) and recovery time objective (RTO) are 24 hours.

Which solution will meet these requirements MOST cost-effectively?

## Options:

**A-** Create a cross-Region read replica and promote the read replica to the primary instance

**B-** Use AWS Database Migration Service (AWS DMS) to create RDS cross-Region replication.

**C-** Use cross-Region replication every 24 hours to copy native backups to an Amazon S3 bucket

**D-** Copy automatic snapshots to another Region every 24 hours.

## Answer:

D

## Explanation:

This solution is the most cost-effective and meets the RPO and RTO requirements of 24 hours.

Automatic Snapshots: Amazon RDS automatically creates snapshots of your DB instance at regular intervals. By copying these snapshots to another AWS Region every 24 hours, you ensure that you have a backup available in a different geographic location, providing disaster recovery capability.

RPO and RTO: Since the company's RPO and RTO are both 24 hours, copying snapshots daily to another Region is sufficient. In the event of a disaster, you can restore the DB instance from the most recent snapshot in the target Region.

Why Not Other Options?:

Option A (Cross-Region Read Replica): This could provide a faster recovery time but is more costly due to the ongoing replication and resource usage in another Region.

Option B (DMS Cross-Region Replication): While effective for continuous replication, it introduces complexity and cost that isn't necessary given the 24-hour RPO/RTO.

Option C (Cross-Region Native Backup Copy): This involves more manual steps and doesn't offer as straightforward a solution as automated snapshot copying.

AWS Reference:

Amazon RDS Automated Backups and Snapshots - Details on automated backups and snapshots in RDS.

Copying an Amazon RDS DB Snapshot - How to copy DB snapshots to another Region.

# Question 9

**Question Type:** MultipleChoice

A company has an employee web portal. Employees log in to the portal to view payroll details. The company is developing a new system to give employees the ability to upload scanned documents for reimbursement. The company runs a program to extract text-based data from the documents and attach the extracted information to each employee's reimbursement IDs for processing.

The employee web portal requires 100% uptime. The document extract program runs infrequently throughout the day on an on-demand basis. The company wants to build a scalable and cost-effective new system that will require minimal changes to the existing web portal. The company does not want to make any code changes.

Which solution will meet these requirements with the LEAST implementation effort?

## Options:

**A-** Run Amazon EC2 On-Demand Instances in an Auto Scaling group for the web portal. Use an AWS Lambda function to run the document extract program. Invoke the Lambda function when an employee uploads a new reimbursement document.

**B-** Run Amazon EC2 Spot Instances in an Auto Scaling group for the web portal. Run the document extract program on EC2 Spot Instances Start document extract program instances when an employee uploads a new reimbursement document.

**C-** Purchase a Savings Plan to run the web portal and the document extract program. Run the web portal and the document extract program in an Auto Scaling group.

**D-** Create an Amazon S3 bucket to host the web portal. Use Amazon API Gateway and an AWS Lambda function for the existing functionalities. Use the Lambda function to run the document extract program. Invoke the Lambda function when the API that is associated with a new document upload is called.

## Answer:

A

## Explanation:

This solution offers the most scalable and cost-effective approach with minimal changes to the existing web portal and no code modifications.

Amazon EC2 On-Demand Instances in an Auto Scaling Group: Running the web portal on EC2 On-Demand instances ensures 100% uptime and scalability. The Auto Scaling group will maintain the desired number of instances, automatically scaling up or down as needed, ensuring high availability for the employee web portal.

AWS Lambda for Document Extraction: Lambda is a serverless compute service that allows you to run code in response to events without provisioning or managing servers. By using Lambda to run the document extraction program, you can trigger the function whenever an employee uploads a document. This approach is cost-effective since you only pay for the compute time used by the Lambda function.

No Code Changes Required: This solution integrates with the existing infrastructure with minimal implementation effort and does not require any modifications to the web portal's code.

Why Not Other Options?:

Option B (Spot Instances): Spot Instances are not suitable for workloads requiring 100% uptime, as they can be terminated by AWS with short notice.

Option C (Savings Plan): A Savings Plan could reduce costs but does not address the requirement for running the document extraction program efficiently or without code changes.

Option D (S3 with API Gateway and Lambda): This would require significant changes to the existing web portal setup, including moving the portal to S3 and reconfiguring its architecture, which contradicts the requirement of minimal implementation effort and no code changes.

AWS Reference:

Amazon EC2 Auto Scaling - Information on how to use Auto Scaling for EC2 instances.

AWS Lambda - Overview of AWS Lambda and its use cases.