

Attempt 3

Question 1 Correct

A video analytics company runs data-intensive batch processing workloads that generate large log files and metadata daily. These files are currently stored in an on-premises NFS-based storage system located in the company's primary data center. However, the storage system is becoming increasingly difficult to scale and is unable to meet the company's growing storage demands. The IT team wants to migrate to a cloud-based storage solution that minimizes costs, retains NFS compatibility, and supports automated tiering of rarely accessed data to lower-cost storage. The team prefers to continue using existing NFS-based tools and protocols for compatibility with their current application stack.

Which solution will meet these requirements MOST cost-effectively?

Use Amazon FSx for Windows File Server to replace the NFS workload. Enable data deduplication and automatic backups. Use Amazon S3 Glacier to move snapshots to a cost-efficient storage tier. Reconfigure the analytics application to access files using SMB protocol

Deploy an AWS Storage Gateway Volume Gateway in cached mode. Attach it as a block device to an on-premises file server and mount NFS on top. Store snapshots in Amazon S3 Glacier Deep Archive, and use AWS Backup to manage recovery operations and tiering

Your answer is correct

Deploy an AWS Storage Gateway File Gateway on premises. Configure it to present an NFS-compatible file share to the workloads. Store the uploaded files in Amazon S3, and use S3 Lifecycle policies to automatically transition infrequently accessed objects to lower-cost storage classes

Provision an Amazon Elastic File System (Amazon EFS) file system with the One Zone–IA storage class. Use AWS DataSync to migrate the NFS data to EFS. Configure the application to mount the file system over NFS and activate lifecycle management to tier infrequently accessed files

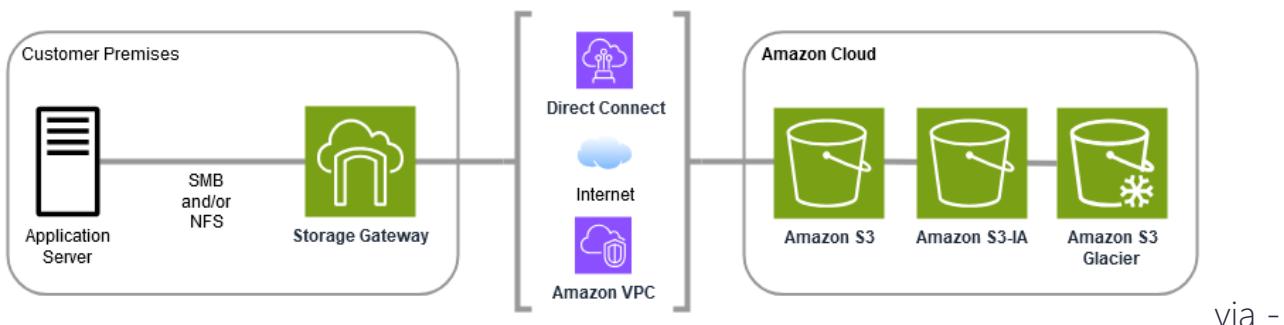
Overall explanation

Correct option:

Deploy an AWS Storage Gateway File Gateway on premises. Configure it to present an NFS-compatible file share to the workloads. Store the uploaded files in Amazon S3, and use S3 Lifecycle policies to automatically transition infrequently accessed objects to lower-cost storage classes

This is the most cost-effective and operationally seamless solution. File Gateway presents an NFS-compatible interface to on-premises applications, allowing the existing workloads to write to it as they would to a traditional NFS server. Behind the scenes, the data is stored in Amazon S3, allowing the company to leverage S3's virtually unlimited scalability. With S3 Lifecycle policies, files can be automatically transitioned to low-cost storage classes like S3 Standard-IA or Glacier, further optimizing costs over time. This approach requires minimal changes to applications and eliminates the scalability bottleneck.

How File Gateway Works:



via -

<https://docs.aws.amazon.com/filegateway/latest/files3/file-gateway-concepts.html>

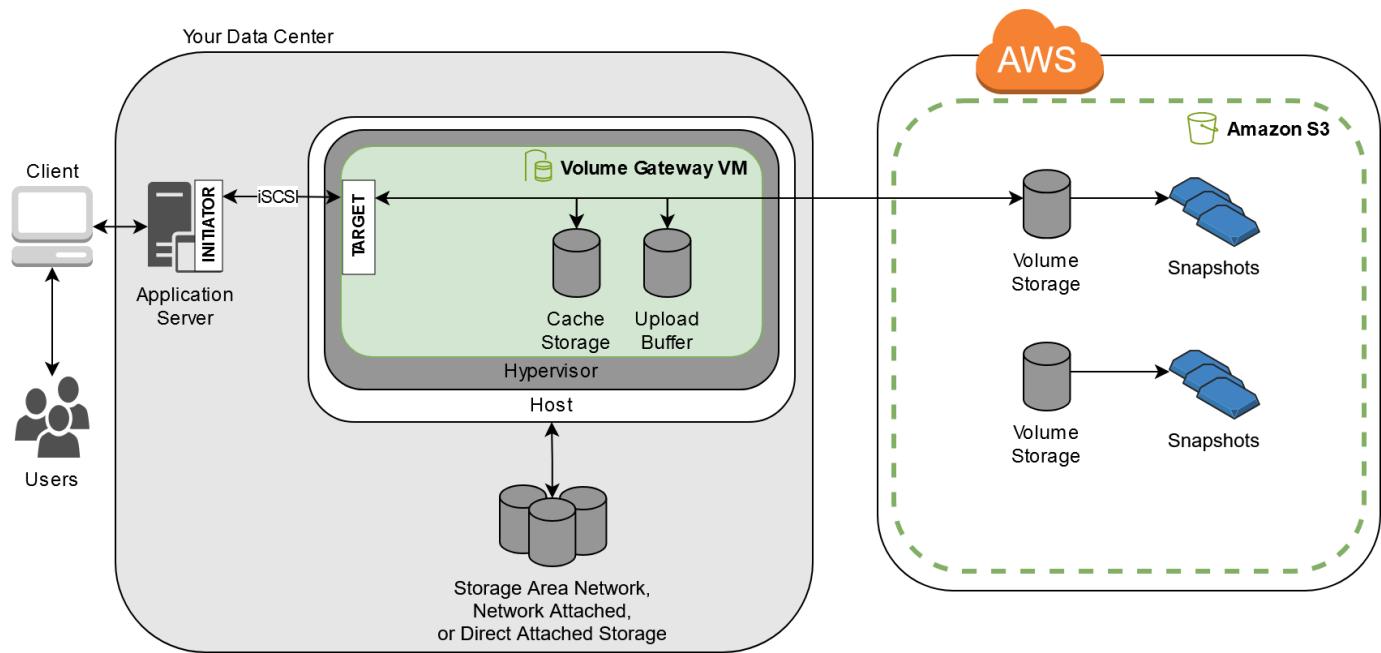
Incorrect options:

Provision an Amazon Elastic File System (Amazon EFS) file system with the One Zone–IA storage class. Use AWS DataSync to migrate the NFS data to EFS. Configure the application to mount the file system over NFS and activate lifecycle management to tier infrequently accessed files - Amazon EFS is designed for shared file systems in the cloud, and while it supports NFS and lifecycle management, it is not cost-optimal for hybrid

setups where on-prem NFS compatibility is required. Additionally, One Zone-IA stores data in a single Availability Zone, making it less resilient. This solution also assumes a full data migration and reconfiguration of the applications, increasing complexity and operational overhead.

Deploy an AWS Storage Gateway Volume Gateway in cached mode. Attach it as a block device to an on-premises file server and mount NFS on top. Store snapshots in Amazon S3 Glacier Deep Archive, and use AWS Backup to manage recovery operations and tiering - Volume Gateway provides iSCSI block storage, not file-based access, meaning you would have to mount NFS over a block volume, which is not ideal for compatibility or scalability. Although Volume Gateway can store snapshots in S3 and Glacier, it's not optimized for NFS workloads or direct object access via S3. It also lacks native support for S3 lifecycle tiering, relying instead on backup and snapshot mechanisms.

How Volume Gateway Works:



via -

<https://docs.aws.amazon.com/storagegateway/latest/vgw/StorageGatewayConcepts.html>

Use Amazon FSx for Windows File Server to replace the NFS workload. Enable data deduplication and automatic backups. Use Amazon S3 Glacier to move snapshots to a cost-efficient storage tier. Reconfigure the analytics application to access files using SMB protocol - Amazon FSx for Windows File Server is intended for Windows-based SMB workloads, not NFS-based environments. It offers useful features like deduplication and backups, but it requires the application to be rewritten or reconfigured to use SMB instead of NFS. Moreover, its integration with Glacier is only applicable to backups and not individual files, and it does not support S3 lifecycle tiering, making it less cost-efficient.

References:

<https://docs.aws.amazon.com/filegateway/latest/files3/what-is-file-s3.html>

<https://docs.aws.amazon.com/filegateway/latest/files3/file-gateway-concepts.html>

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lifecycle-mgmt.html>

<https://docs.aws.amazon.com/fsx/latest/WindowsGuide/what-is.html>

Domain

Design Cost-Optimized Architectures

Question 2 Incorrect

The engineering team at an e-commerce company wants to establish a dedicated, encrypted, low latency, and high throughput connection between its data center and AWS Cloud. The engineering team has set aside sufficient time to account for the operational overhead of establishing this connection.

As a solutions architect, which of the following solutions would you recommend to the company?

Correct answer

Use AWS Direct Connect plus virtual private network (VPN) to establish a connection between the data center and AWS Cloud

Use AWS Transit Gateway to establish a connection between the data center and AWS Cloud

Use AWS site-to-site VPN to establish a connection between the data center and AWS Cloud

Your answer is incorrect

Use AWS Direct Connect to establish a connection between the data center and AWS Cloud

Overall explanation

Correct option:

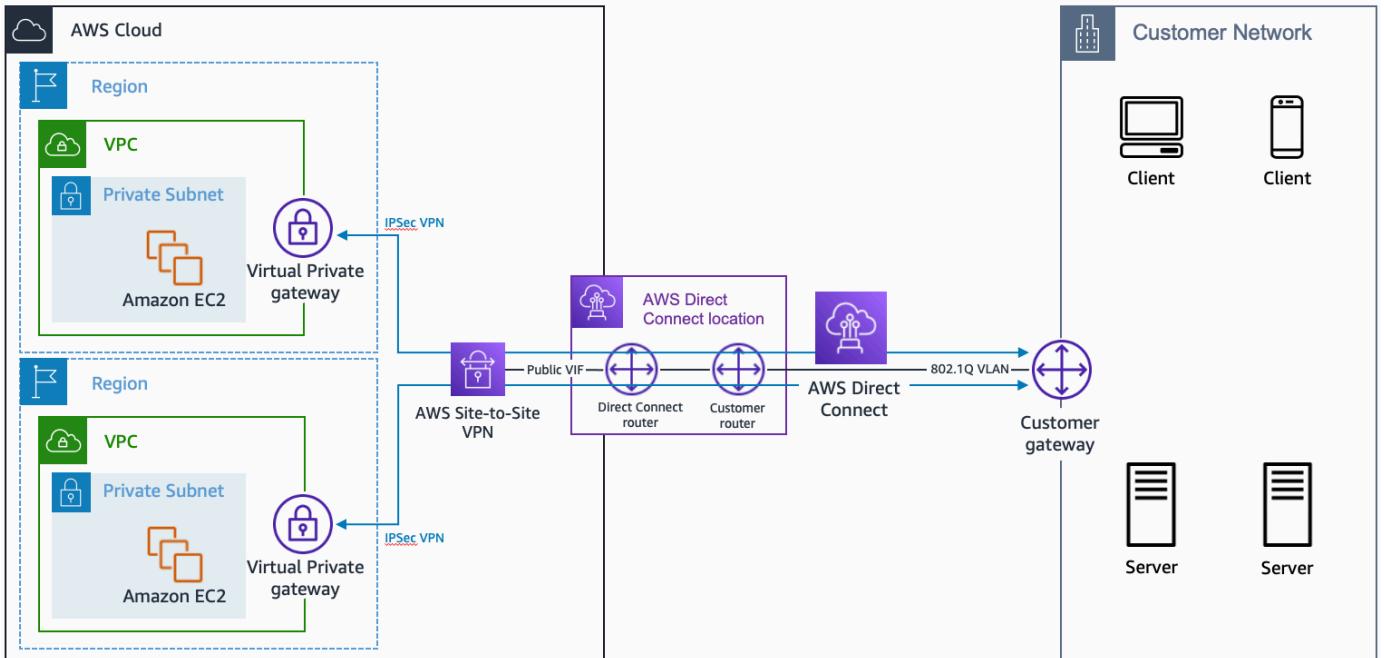
Use AWS Direct Connect plus virtual private network (VPN) to establish a connection between the data center and AWS Cloud

AWS Direct Connect is a cloud service solution that makes it easy to establish a dedicated network connection from your premises to AWS. AWS Direct Connect lets you establish a dedicated network connection between your network and one of the AWS Direct Connect locations.

With AWS Direct Connect plus VPN, you can combine one or more AWS Direct Connect dedicated network connections with the Amazon VPC VPN. This combination provides an IPsec-encrypted private connection that also reduces network costs, increases bandwidth throughput, and provides a more consistent network experience than internet-based VPN connections.

This solution combines the AWS managed benefits of the VPN solution with low latency, increased bandwidth, more consistent benefits of the AWS Direct Connect solution, and an end-to-end, secure IPsec connection. Therefore, AWS Direct Connect plus VPN is the correct solution for this use-case.

AWS Direct Connect Plus VPN:



via - <https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/aws-direct-connect-vpn.html>

Incorrect options:

Use AWS site-to-site VPN to establish a connection between the data center and AWS Cloud - AWS Site-to-Site VPN enables you to securely connect your on-premises network or branch office site to your Amazon Virtual Private Cloud (Amazon VPC). A VPC VPN Connection utilizes IPSec to establish encrypted network connectivity between your intranet and Amazon VPC over the Internet. VPN Connections are a good solution if you have an immediate need, have low to modest bandwidth requirements, and can tolerate the inherent variability in Internet-based connectivity. However, Site-to-site VPN cannot provide low latency and high throughput connection, therefore this option is ruled out.

Use AWS Transit Gateway to establish a connection between the data center and AWS Cloud - AWS Transit Gateway is a network transit hub that you can use to interconnect your virtual private clouds (VPC) and on-premises networks. AWS Transit Gateway by itself cannot establish a low latency and high throughput connection between a data center and AWS Cloud. Hence this option is incorrect.

Use AWS Direct Connect to establish a connection between the data center and AWS Cloud - AWS Direct Connect by itself cannot provide an encrypted connection between a data center and AWS Cloud, so this option is ruled out.

References:

<https://aws.amazon.com/directconnect/>

<https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/aws-direct-connect-plus-vpn-network-to-amazon.html>

Domain

Design High-Performing Architectures

Question 3 Incorrect

A healthcare startup needs to enforce compliance and regulatory guidelines for objects stored in Amazon S3. One of the key requirements is to provide adequate protection against accidental deletion of objects.

As a solutions architect, what are your recommendations to address these guidelines? (Select two) ?

Your selection is correct

Enable multi-factor authentication (MFA) delete on the Amazon S3 bucket

Change the configuration on Amazon S3 console so that the user needs to provide additional confirmation while deleting any Amazon S3 object

Create an event trigger on deleting any Amazon S3 object. The event invokes an Amazon Simple Notification Service (Amazon SNS) notification via email to the IT manager

Correct selection

Establish a process to get managerial approval for deleting Amazon S3 objects

Overall explanation

Correct options:

Enable versioning on the Amazon S3 bucket

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite.

For example:

If you overwrite an object, it results in a new object version in the bucket. You can always restore the previous version. If you delete an object, instead of removing it permanently, Amazon S3 inserts a delete marker, which becomes the current object version. You can always restore the previous version. Hence, this is the correct option.

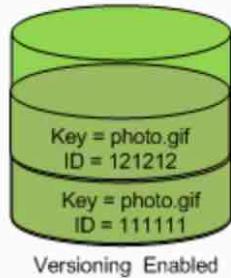
Versioning Overview:

Using versioning

[PDF](#) | [Kindle](#) | [RSS](#)

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. When you enable versioning for a bucket, if Amazon S3 receives multiple write requests for the same object simultaneously, it stores all of the objects.

If you enable versioning for a bucket, Amazon S3 automatically generates a unique version ID for the object being stored. In one bucket, for example, you can have two objects with the same key, but different version IDs, such as `photo.gif` (version 111111) and `photo.gif` (version 121212).



via - <https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html>

Enable multi-factor authentication (MFA) delete on the Amazon S3 bucket

To provide additional protection, multi-factor authentication (MFA) delete can be enabled. MFA delete requires secondary authentication to take place before objects can be permanently deleted from an Amazon S3 bucket. Hence, this is the correct option.

Incorrect options:

Create an event trigger on deleting any Amazon S3 object. The event invokes an Amazon Simple Notification Service (Amazon SNS) notification via email to the IT manager - Sending an event trigger after object deletion does not meet the objective of preventing object deletion by mistake because the object has already been deleted. So, this option is incorrect.

Establish a process to get managerial approval for deleting Amazon S3 objects - This option for getting managerial approval is just a distractor.

Change the configuration on Amazon S3 console so that the user needs to provide additional confirmation while deleting any Amazon S3 object - There is no provision to set up Amazon S3 configuration to ask for additional confirmation before deleting an object. This option is incorrect.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingMFADelete.html>

Domain

Design Resilient Architectures

Question 4 Correct

A research group runs its flagship application on a fleet of Amazon EC2 instances for a specialized task that must deliver high random I/O performance. Each instance in the fleet would have access to a dataset that is replicated across the instances by the application itself. Because of the resilient application architecture, the specialized task would continue to be processed even if any instance goes down, as the underlying application would ensure the replacement instance has access to the required dataset.

Which of the following options is the MOST cost-optimal and resource-efficient solution to build this fleet of Amazon EC2 instances?

Use Amazon EC2 instances with access to Amazon S3 based storage

Your answer is correct

Use Instance Store based Amazon EC2 instances

Use Amazon EC2 instances with Amazon EFS mount points

Use Amazon Elastic Block Store (Amazon EBS) based EC2 instances

Overall explanation

Correct option:

Use Instance Store based Amazon EC2 instances

An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host instance. Instance store is ideal for the temporary storage of information that changes frequently such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers. Instance store volumes are included as part of the instance's usage cost.

As Instance Store based volumes provide high random I/O performance at low cost (as the storage is part of the instance's usage cost) and the resilient architecture can adjust for the loss of any instance, therefore you should use Instance Store based Amazon EC2 instances for this use-case.

Amazon EC2 Instance Store Overview:

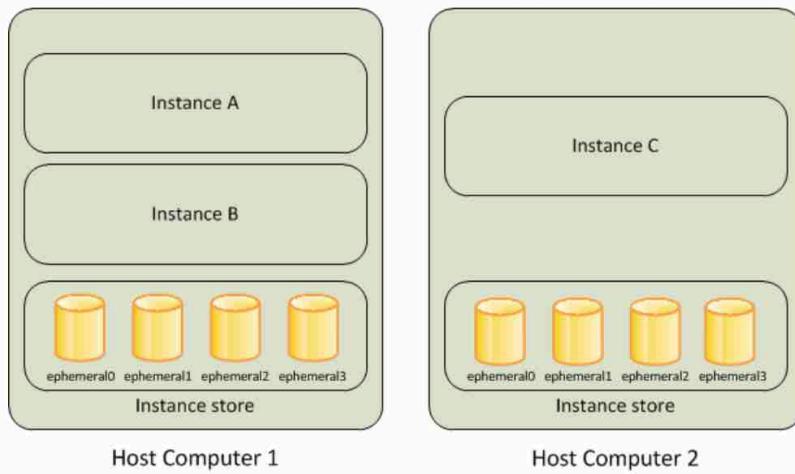
Amazon EC2 Instance Store

[PDF](#) | [Kindle](#) | [RSS](#)

An *instance store* provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

An instance store consists of one or more instance store volumes exposed as block devices. The size of an instance store as well as the number of devices available varies by instance type.

The virtual devices for instance store volumes are `ephemeral[0-23]`. Instance types that support one instance store volume have `ephemeral0`. Instance types that support two instance store volumes have `ephemeral0` and `ephemeral1`, and so on.



via - <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html>

Incorrect options:

Use Amazon Elastic Block Store (Amazon EBS) based EC2 instances - Amazon Elastic Block Store (Amazon EBS) based volumes would need to use provisioned IOPS (io1) as the storage type and that would incur additional costs. As we are looking for the most cost-optimal solution, this option is ruled out.

Use Amazon EC2 instances with Amazon EFS mount points - Using Amazon Elastic File System (Amazon EFS) implies that extra resources would have to be provisioned (compared to using instance store where the storage is located on disks that are physically attached to the host instance itself). As we are looking for the most resource-efficient solution, this option is also ruled out.

Use Amazon EC2 instances with access to Amazon S3 based storage - Using Amazon EC2 instances with access to Amazon S3 based storage does not deliver high random I/O performance, this option is just added as a distractor.

Reference:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html>

Question 5 Incorrect

A new DevOps engineer has joined a large financial services company recently. As part of his onboarding, the IT department is conducting a review of the checklist for tasks related to AWS Identity and Access Management (AWS IAM).

As an AWS Certified Solutions Architect – Associate, which best practices would you recommend (Select two)?

Your selection is incorrect

Use user credentials to provide access specific permissions for Amazon EC2 instances

Create a minimum number of accounts and share these account credentials among employees

Grant maximum privileges to avoid assigning privileges again

Correct selection

Enable AWS Multi-Factor Authentication (AWS MFA) for privileged users

Your selection is correct

Configure AWS CloudTrail to log all AWS Identity and Access Management (AWS IAM) actions

Overall explanation

Correct options:

Enable AWS Multi-Factor Authentication (AWS MFA) for privileged users

As per the AWS best practices, it is better to enable Multi Factor Authentication (MFA) for privileged users via an MFA-enabled mobile device or hardware MFA token.

Configure AWS CloudTrail to log all AWS Identity and Access Management (AWS IAM) actions

AWS recommends to turn on AWS CloudTrail to log all IAM actions for monitoring and audit purposes.

Incorrect options:

Create a minimum number of accounts and share these account credentials among employees - AWS recommends that user account credentials should not be shared between users. So, this option is incorrect.

Grant maximum privileges to avoid assigning privileges again - AWS recommends granting the least privileges required to complete a certain job and avoid giving excessive privileges which can be misused. So, this option is incorrect.

Use user credentials to provide access specific permissions for Amazon EC2 instances - It is highly recommended to use roles to grant access permissions for EC2 instances working on different AWS services. So, this option is incorrect.

References:

<https://aws.amazon.com/iam/>

<https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html>

<https://aws.amazon.com/cloudtrail/faqs/>

Domain

Design Secure Architectures

Question 6 Correct

The sourcing team at the US headquarters of a global e-commerce company is preparing a spreadsheet of the new product catalog. The spreadsheet is saved on an Amazon Elastic File System (Amazon EFS) created in `us-east-1` region. The sourcing team counterparts from other AWS regions such as Asia Pacific and Europe also want to collaborate on this spreadsheet.

As a solutions architect, what is your recommendation to enable this collaboration with the LEAST amount of operational overhead?

The spreadsheet will have to be copied into Amazon EFS file systems of other AWS regions as Amazon EFS is a regional service and it does not allow access from other AWS regions

Your answer is correct

The spreadsheet on the Amazon Elastic File System (Amazon EFS) can be accessed in other AWS regions by using an inter-region VPC peering connection

The spreadsheet will have to be copied in Amazon S3 which can then be accessed from any AWS region

The spreadsheet data will have to be moved into an Amazon RDS for MySQL database which can then be accessed from any AWS region

Overall explanation

Correct option:

The spreadsheet on the Amazon Elastic File System (Amazon EFS) can be accessed in other AWS regions by using an inter-region VPC peering connection

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system for use with AWS Cloud services and on-premises resources.

Amazon EFS is a regional service storing data within and across multiple Availability Zones (AZs) for high availability and durability. Amazon EC2 instances can access your file system across AZs, regions, and VPCs, while on-premises servers can access using AWS Direct Connect or AWS VPN.

You can connect to Amazon EFS file systems from EC2 instances in other AWS regions using an inter-region VPC peering connection, and from on-premises servers using an AWS VPN connection. So this is the correct option.

Incorrect options:

The spreadsheet will have to be copied in Amazon S3 which can then be accessed from any AWS region

The spreadsheet data will have to be moved into an Amazon RDS for MySQL database which can then be accessed from any AWS region

Copying the spreadsheet into Amazon S3 or Amazon RDS for MySQL database is not the correct solution as it involves a lot of operational overhead. For Amazon RDS, one would need to write custom code to replicate the spreadsheet functionality running off of the database. S3 does not allow in-place edit of an object. Additionally, it's also not POSIX compliant. So one would need to develop a custom application to "simulate in-place edits" to support collaboration as per the use-case. So both these options are ruled out.

The spreadsheet will have to be copied into Amazon EFS file systems of other AWS regions as Amazon EFS is a regional service and it does not allow access from other AWS regions - Creating copies of the spreadsheet into Amazon EFS file systems of other

AWS regions would mean no collaboration would be possible between the teams. In this case, each team would work on "its own file" instead of a single file accessed and updated by all teams. Hence this option is incorrect.

Reference:

<https://aws.amazon.com/efs/>

Domain

Design High-Performing Architectures

Question 7 Correct

An e-commerce company manages a digital catalog of consumer products submitted by third-party sellers. Each product submission includes a description stored as a text file in an Amazon S3 bucket. These descriptions may include ingredient information for consumable products like snacks, supplements, or beverages. The company wants to build a fully automated solution that extracts ingredient names from the uploaded product descriptions and uses those names to query an Amazon DynamoDB table, which returns precomputed health and safety scores for each ingredient. Non-food items and invalid submissions can be ignored without affecting application logic. The company has no in-house machine learning (ML) experts and is looking for the most cost-effective solution with minimal operational overhead.

Which solution meets these requirements MOST cost-effectively?

Create a workflow where Amazon Transcribe is used to convert synthetic audio versions (created from text of the product descriptions) back into text. Analyze the transcripts manually or using simple keyword matching within a Lambda function. Use Amazon SNS to notify the content moderation team for each processed file

Use Amazon SageMaker with a custom-trained NLP model to identify ingredients from the uploaded descriptions. Use Amazon EventBridge to invoke a Lambda function that forwards the document content to a SageMaker endpoint and stores the results in DynamoDB. Fine-tune the model using labeled ingredient datasets from open-source repositories and retrain it monthly

Use Amazon Lookout for Vision to scan the uploaded text files in the S3 bucket and extract entities. Invoke this workflow using an S3-triggered Lambda function. Parse the output and use Amazon API Gateway to push updates to the frontend in real time

Your answer is correct

Configure S3 Event Notifications to trigger an AWS Lambda function whenever a new product description is uploaded. Inside the function, use Amazon Comprehend's custom entity recognition feature to extract ingredient names. Store these names in the DynamoDB table and let the front-end application query for health scores

Overall explanation

Correct option:

Configure S3 Event Notifications to trigger an AWS Lambda function whenever a new product description is uploaded. Inside the function, use Amazon Comprehend's custom entity recognition feature to extract ingredient names. Store these names in the DynamoDB table and let the front-end application query for health scores

This option uses Amazon Comprehend's custom entity recognition feature to detect and extract ingredients from unstructured text. Since Comprehend is fully managed and doesn't require ML expertise, it aligns with the company's constraints. It integrates easily into a serverless pipeline using S3 Event Notifications and AWS Lambda. The ingredient names can then be used as lookup keys in a DynamoDB table to retrieve nutrition or safety scores.

This architecture is event-driven, scalable, cost-efficient, and requires minimal operational effort.

Amazon Comprehend insights

Amazon Comprehend uses a pre-trained model to examine and analyze a document or set of documents to gather insights about it. This model is continuously trained on a large body of text so that there is no need for you to provide training data.

Amazon Comprehend analyzes the following types of insights:

- **Entities** – References to the names of people, places, items, and locations contained in a document.
- **Key phrases** – Phrases that appear in a document. For example, a document about a basketball game might return the names of the teams, the name of the venue, and the final score.
- **Personally Identifiable Information (PII)** – Personal data that can identify an individual, such as an address, bank account number, or phone number.
- **Language** – The dominant language of a document.
- **Sentiment** – The dominant sentiment of a document, which can be positive, neutral, negative, or mixed.
- **Targeted sentiment** – The sentiments associated with specific entities in a document. The sentiment for each entity occurrence can be positive, negative, neutral or mixed.
- **Syntax** – The parts of speech for each word in the document.

via - <https://docs.aws.amazon.com/comprehend/latest/dg/what-is.html>

Incorrect options:

Use Amazon SageMaker with a custom-trained NLP model to identify ingredients from the uploaded descriptions. Use Amazon EventBridge to invoke a Lambda function that forwards the document content to a SageMaker endpoint and stores the results in DynamoDB. Fine-tune the model using labeled ingredient datasets from open-source repositories and retrain it monthly - While SageMaker is a powerful and flexible ML service, it requires ML expertise to train, deploy, and manage models. Fine-tuning and maintaining a custom NLP model, especially for entity extraction, involves substantial effort and cost. It also contradicts the company's requirement to avoid ML development overhead. This solution is over-engineered for the use case and not the most cost-effective.

Create a workflow where Amazon Transcribe is used to convert synthetic audio versions (created from text of the product descriptions) back into text. Analyze the transcripts manually or using simple keyword matching within a Lambda function. Use Amazon SNS to notify the content moderation team for each processed file - This approach is convoluted and inefficient. Generating synthetic audio from text, transcribing it back, and relying on manual or keyword-based analysis introduces redundant processing steps and delays. Additionally, using Amazon SNS to notify human reviewers increases operational burden and violates the requirement for a fully automated solution.

Use Amazon Lookout for Vision to scan the uploaded text files in the S3 bucket and extract entities. Invoke this workflow using an S3-triggered Lambda function. Parse the output and use Amazon API Gateway to push updates to the frontend in real time - Amazon Lookout for Vision is designed for image-based anomaly detection, not text processing. It cannot extract entities or analyze structured/unstructured text documents,

making it completely unsuitable for this use case. Using API Gateway for pushing updates is unrelated to the core requirement of ingredient extraction and further increases complexity.

References:

<https://docs.aws.amazon.com/comprehend/latest/dg/what-is.html>

<https://docs.aws.amazon.com/comprehend/latest/dg/custom-entity-recognition.html>

<https://aws.amazon.com/blogs/machine-learning/build-a-custom-entity-recognizer-using-amazon-comprehend/>

<https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html>

<https://docs.aws.amazon.com/transcribe/latest/dg/what-is.html>

<https://docs.aws.amazon.com/lookout-for-vision/latest/developer-guide/what-is.html>

Domain

Design High-Performing Architectures

Question 8 Incorrect

A company uses Amazon S3 buckets for storing sensitive customer data. The company has defined different retention periods for different objects present in the Amazon S3 buckets, based on the compliance requirements. But, the retention rules do not seem to work as expected.

Which of the following options represent a valid configuration for setting up retention periods for objects in Amazon S3 buckets? (Select two)

Your selection is correct

When you apply a retention period to an object version explicitly, you specify a **Retain Until Date** for the object version

Your selection is correct

When you use bucket default settings, you specify a **Retain Until Date** for the object version

The bucket default settings will override any explicit retention mode or period you request on an object version

Your selection is incorrect

You cannot place a retention period on an object version through a bucket default setting

Overall explanation

Correct options:

When you apply a retention period to an object version explicitly, you specify a **Retain Until Date** for the object version

You can place a retention period on an object version either explicitly or through a bucket default setting. When you apply a retention period to an object version explicitly, you specify a **Retain Until Date** for the object version. Amazon S3 stores the Retain Until Date setting in the object version's metadata and protects the object version until the retention period expires.

Different versions of a single object can have different retention modes and periods

Like all other Object Lock settings, retention periods apply to individual object versions.

Different versions of a single object can have different retention modes and periods.

For example, suppose that you have an object that is 15 days into a 30-day retention period, and you PUT an object into Amazon S3 with the same name and a 60-day retention period. In this case, your PUT succeeds, and Amazon S3 creates a new version of the object with a 60-day retention period. The older version maintains its original retention period and becomes deletable in 15 days.

Incorrect options:

You cannot place a retention period on an object version through a bucket default setting - You can place a retention period on an object version either explicitly or through a bucket default setting.

When you use bucket default settings, you specify a Retain Until Date for the object version - When you use bucket default settings, you don't specify a Retain Until Date. Instead, you specify a duration, in either days or years, for which every object version placed in the bucket should be protected.

The bucket default settings will override any explicit retention mode or period you request on an object version - If your request to place an object version in a bucket contains an explicit retention mode and period, those settings override any bucket default settings for that object version.

Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lock-overview.html>

Domain

Design Secure Architectures

Question 9 Incorrect

A new DevOps engineer has just joined a development team and wants to understand the replication capabilities for Amazon RDS Multi-AZ deployment as well as Amazon RDS Read-replicas.

Which of the following correctly summarizes these capabilities for the given database?

Multi-AZ follows asynchronous replication and spans at least two Availability Zones (AZs) within a single region. Read replicas follow asynchronous replication and can be within an Availability Zone (AZ), Cross-AZ, or Cross-Region

Your answer is incorrect

Multi-AZ follows asynchronous replication and spans at least two Availability Zones (AZs) within a single region. Read replicas follow synchronous replication and can be within an Availability Zone (AZ), Cross-AZ, or Cross-Region

Correct answer

Multi-AZ follows synchronous replication and spans at least two Availability Zones (AZs) within a single region. Read replicas follow asynchronous replication and can be within an Availability Zone (AZ), Cross-AZ, or Cross-Region

Multi-AZ follows asynchronous replication and spans one Availability Zone (AZ) within a single region. Read replicas follow synchronous replication and can be within an Availability Zone (AZ), Cross-AZ, or Cross-Region

Overall explanation

Correct option:

Multi-AZ follows synchronous replication and spans at least two Availability Zones (AZs) within a single region. Read replicas follow asynchronous replication and can be within an Availability Zone (AZ), Cross-AZ, or Cross-Region

Amazon RDS Multi-AZ deployments provide enhanced availability and durability for RDS database (DB) instances, making them a natural fit for production database workloads.

When you provision a Multi-AZ DB Instance, Amazon RDS automatically creates a primary DB Instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ). Multi-AZ spans at least two Availability Zones (AZs) within a single region.

Amazon RDS Read Replicas provide enhanced performance and durability for RDS database (DB) instances. They make it easy to elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads. For the MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server database engines, Amazon RDS creates a second DB instance using a snapshot of the source DB instance. It then uses the engines' native asynchronous replication to update the read replica whenever there is a change to the source DB instance.

Amazon RDS replicates all databases in the source DB instance. Read replicas can be within an Availability Zone (AZ), Cross-AZ, or Cross-Region.

Exam Alert:

Please review this comparison vis-a-vis Multi-AZ vs Read Replica for Amazon RDS:

Multi-AZ deployments, multi-region deployments, and read replicas

Amazon RDS Multi-AZ deployments complement multi-region deployments and [read replicas](#). While all three features increase availability and durability by maintaining additional copies of your data, there are differences between them:

Multi-AZ deployments	Multi-Region deployments	Read replicas
Main purpose is high availability	Main purpose is disaster recovery and local performance	Main purpose is scalability
Non-Aurora: synchronous replication; Aurora: asynchronous replication	Asynchronous replication	Asynchronous replication
Non-Aurora: only the primary instance is active; Aurora: all instances are active	All regions are accessible and can be used for reads	All read replicas are accessible and can be used for readscaling
Non-Aurora: automated backups are taken from standby; Aurora: automated backups are taken from shared storage layer	Automated backups can be taken in each region	No backups configured by default
Always span at least two Availability Zones within a single region	Each region can have a Multi-AZ deployment	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Non-Aurora: database engine version upgrades happen on primary; Aurora: all instances are updated together	Non-Aurora: database engine version upgrade is independent in each region; Aurora: all instances are updated together	Non-Aurora: database engine version upgrade is independent from source instance; Aurora: all instances are updated together
Automatic failover to standby (non-Aurora) or read replica (Aurora) when a problem is detected	Aurora allows promotion of a secondary region to be the master	Can be manually promoted to a standalone database instance (non-Aurora) or to be the primary instance (Aurora)

via - <https://aws.amazon.com/rds/features/multi-az/>

Incorrect Options:

Multi-AZ follows asynchronous replication and spans one Availability Zone (AZ) within a single region. Read replicas follow synchronous replication and can be within an Availability Zone (AZ), Cross-AZ, or Cross-Region

Multi-AZ follows asynchronous replication and spans at least two Availability Zones (AZs) within a single region. Read replicas follow synchronous replication and can be within an Availability Zone (AZ), Cross-AZ, or Cross-Region

Multi-AZ follows asynchronous replication and spans at least two Availability Zones (AZs) within a single region. Read replicas follow asynchronous replication and can be within an Availability Zone (AZ), Cross-AZ, or Cross-Region

These three options contradict the earlier details provided in the explanation. To summarize, Multi-AZ deployment follows synchronous replication for Amazon RDS. Hence these options are incorrect.

References:

<https://aws.amazon.com/rds/features/multi-az/>

<https://aws.amazon.com/rds/features/read-replicas/>

Domain

Design Resilient Architectures

Question 10 Incorrect

A financial services company operates a containerized microservices architecture using Kubernetes in its on-premises data center. Due to strict industry regulations and internal security policies, all application data and workloads must remain physically within the on-premises environment. The company's infrastructure team wants to modernize its Kubernetes stack and take advantage of AWS-managed services and APIs, including automated Kubernetes upgrades, Amazon CloudWatch integration, and access to AWS IAM features — but without migrating any data or compute resources to the cloud.

Which AWS solution will best meet the company's requirements for modernization while ensuring that all data remains on premises?

Correct answer

Install an AWS Outposts rack in the company's data center. Use Amazon EKS Anywhere on Outposts to run containerized workloads locally while integrating with AWS APIs

Your answer is incorrect

Set up a dedicated AWS Direct Connect connection between the on-premises environment and an AWS Region. Deploy Amazon EKS in the cloud and connect it to the local Kubernetes cluster. Use IAM roles and API Gateway to integrate authentication and traffic flow for hybrid workloads

Use an AWS Snowball Edge Compute Optimized device to run EKS-compatible Docker containers on-site. Periodically export application logs and container snapshots to Amazon S3 using Snowball's offline data transfer features. Use the Snowball console to orchestrate workloads in batches

Deploy Amazon ECS with Fargate in a nearby AWS Local Zone. Use CloudWatch Logs to forward events to the primary region. Connect the Local Zone to the company's data center over a VPN. Configure containers to pull data from on-premises storage through a mounted file share

Overall explanation

Correct option:

Install an AWS Outposts rack in the company's data center. Use Amazon EKS Anywhere on Outposts to run containerized workloads locally while integrating with AWS APIs

This is the only option that allows the company to run Amazon EKS-compatible Kubernetes clusters entirely on premises while maintaining full data residency and regulatory compliance. AWS Outposts Rack brings native AWS services, including EKS, EC2, S3, and CloudWatch, into the customer's local data center. Amazon EKS Anywhere is a deployment option for Amazon Elastic Kubernetes Service (EKS) that allows you to create and manage Kubernetes clusters on your own infrastructure — such as on-premises servers or edge locations, outside of AWS. With EKS Anywhere running on Outposts, the company can

modernize its Kubernetes stack with familiar AWS tooling without sending any data to the cloud.

Fully private local clusters for Amazon EKS on AWS Outposts powered by VPC Endpoints

by Kobi Biton and Ovidiu Valeanu | on 24 MAR 2023 | in [Amazon Elastic Kubernetes Service](#), [Amazon VPC](#), [AWS Outposts](#), [Containers](#) | [Permalink](#) |  Share

Introduction

Recently, Amazon Elastic Kubernetes Service ([Amazon EKS](#)) added support for local clusters on AWS Outposts racks. In a nutshell, this [deployment option](#) allows our customers to run the entire Kubernetes cluster (i.e., control plane and worker nodes) on [AWS Outposts](#) racks. The rationale behind this deployment option is often described as **static stability**.

In practice, this means that stability is translated into continued operations during temporary, unplanned network disconnects of the AWS Outposts rack network connection from its Parent Region.

Outposts remains a [fully connected offering](#), which acts as an extension of the AWS Cloud in your datacenter. In the event of a network disconnect between your AWS Outposts service and the AWS Parent Region, your top priority and focus should be troubleshooting your service link connection in aim to restore the connectivity as early as possible.

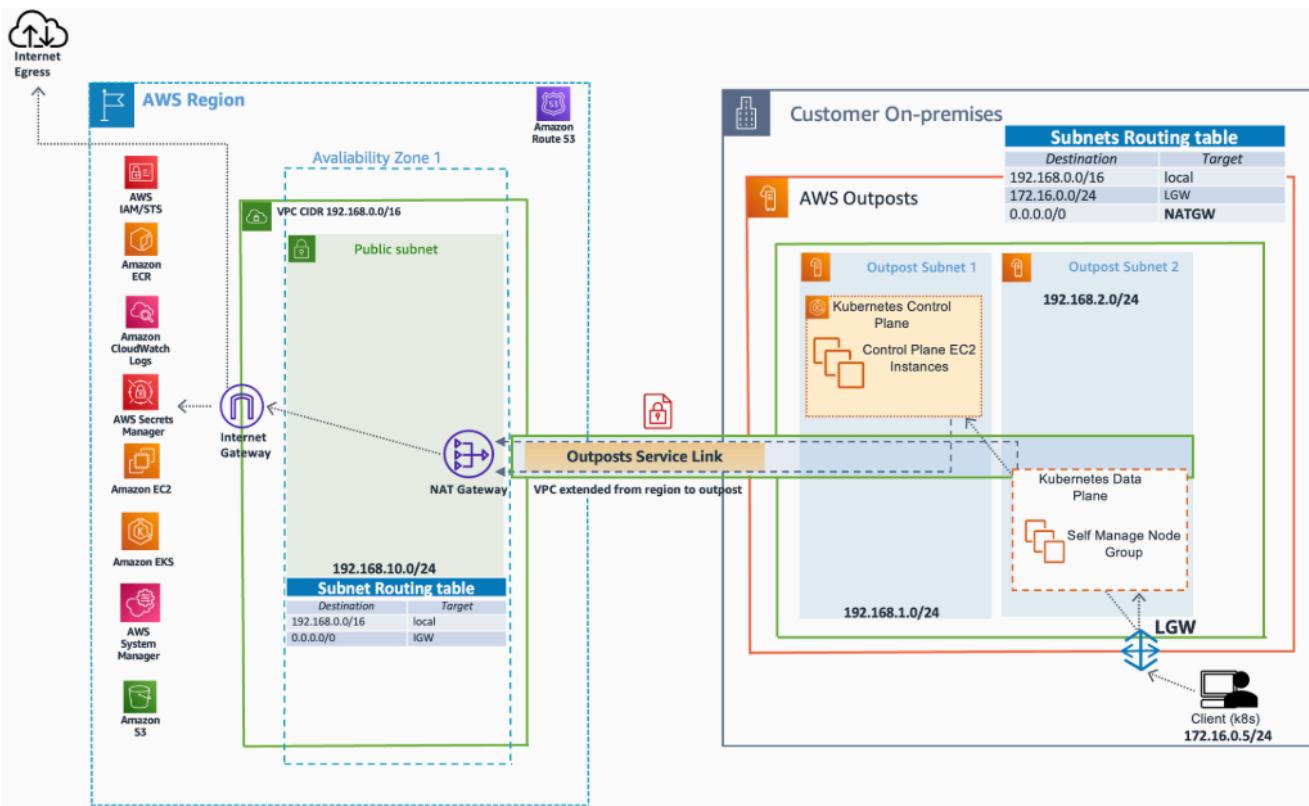
If your use case requires a primarily disconnected environment or your environment isn't equipped with a highly available and reliable consistent connectivity to the AWS Parent Region, then we advise to look into [Amazon EKS Anywhere](#) or [Amazon EKS Distro](#) as the deployment options.

In this post we'll show two Amazon EKS local clusters for AWS Outposts design patterns, (i.e., pattern A and pattern B). We'll focus on *pattern B*, which enables a deployment of a local cluster communicating **privately** with the mandatory AWS in-region service endpoints, powered by Virtual Private Cloud ([VPC Endpoint Services](#)). Privately means that all network traffic stays within the VPC boundaries.

via - <https://aws.amazon.com/blogs/big-data/create-train-and-deploy-machine-learning-models-in-amazon-redshift-using-sql-with-amazon-redshift-ml/>

The following diagram depicts the service endpoints access (public, via in-region NAT Gateway):

Design Pattern A – Public Egress



Solution overview

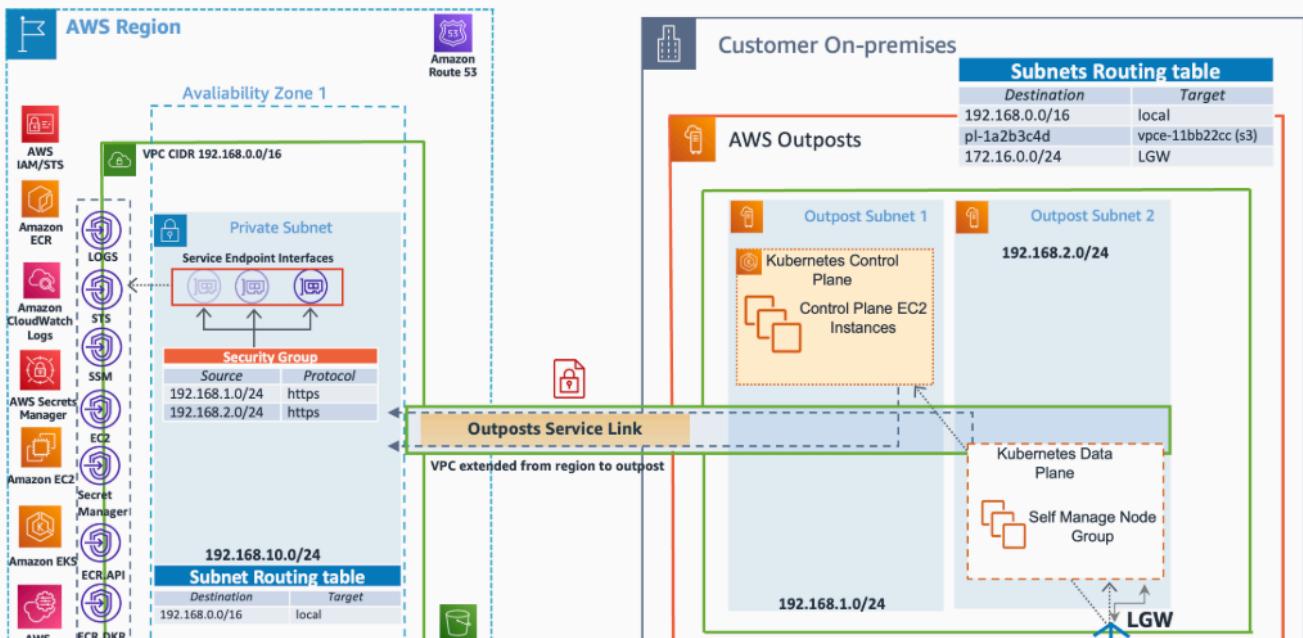
Working backward from our customers, we [recently](#) added support for fully private Amazon EKS local clusters on Outposts leveraging VPC endpoints.

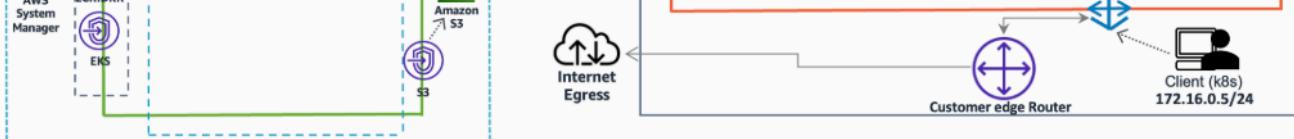
This essentially means that you're no longer required to create an in-region public subnet and NAT Gateway for the sake of connecting the local cluster to its mandatory regional service endpoints.

In simple words, this new design pattern allows the in-outpost Kubernetes control plane and data plane nodes to privately communicate with the regional AWS service endpoints. This is only possible given that the VPC is deployed with the [required](#) VPC service endpoints, [prior](#) to the deployment of the local cluster.

The following diagram depicts the private access (via in-region VPC Endpoint Services):

Design Pattern B – Fully Private





via - <https://aws.amazon.com/blogs/big-data/create-train-and-deploy-machine-learning-models-in-amazon-redshift-using-sql-with-amazon-redshift-ml/>

Incorrect options:

Set up a dedicated AWS Direct Connect connection between the on-premises environment and an AWS Region. Deploy Amazon EKS in the cloud and connect it to the local Kubernetes cluster. Use IAM roles and API Gateway to integrate authentication and traffic flow for hybrid workloads - Although AWS Direct Connect provides a private, low-latency connection between an on-prem data center and AWS, it does not meet data residency requirements, as workloads and data would still reside in the cloud. The use of Amazon EKS in the cloud violates the requirement to keep all workloads and data on premises. This hybrid model is suitable for low-sensitivity environments but not for compliance-restricted industries.

Deploy Amazon ECS with Fargate in a nearby AWS Local Zone. Use CloudWatch Logs to forward events to the primary region. Connect the Local Zone to the company's data center over a VPN. Configure containers to pull data from on-premises storage through a mounted file share - AWS Local Zones are designed to place compute, storage, and select services closer to end-users in metro areas but still operate outside the customer's data center and are managed by AWS. Using ECS in a Local Zone does not satisfy the strict requirement that all data remain physically in the company's data center. Additionally, ECS is not Kubernetes-based, and it doesn't support the same operational model as EKS.

Use an AWS Snowball Edge Compute Optimized device to run EKS-compatible Docker containers on-site. Periodically export application logs and container snapshots to Amazon S3 using Snowball's offline data transfer features. Use the Snowball console to orchestrate workloads in batches - While Snowball Edge can run containerized applications using AWS IoT Greengrass or custom scripts, it is not intended for persistent production workloads or as a long-term infrastructure solution. Its primary use cases include offline data collection, edge processing, and bulk transfer to AWS. It lacks the scalability, lifecycle management, and Kubernetes support needed for this scenario.

References:

<https://aws.amazon.com/blogs/containers/fully-private-local-clusters-for-amazon-eks-on-aws-outposts-powered-by-vpc-endpoints/>

<https://aws.amazon.com/eks/eks-anywhere/>

<https://aws.amazon.com/outposts/rack/>

<https://aws.amazon.com/about-aws/global-infrastructure/localzones/>

<https://docs.aws.amazon.com/snowball/latest/developer-guide/whatisedge.html>

Domain

Design Secure Architectures

Question 11 Correct

A software company has a globally distributed team of developers, that requires secure and compliant access to AWS environments. The company manages multiple AWS accounts under AWS Organizations and uses an on-premises Microsoft Active Directory for user authentication. To simplify access control and identity governance across projects and accounts, the company wants a centrally managed solution that integrates with their existing infrastructure. The solution should require the least amount of ongoing operational management.

Which approach best meets the company's requirements?

Use AWS Control Tower to enable account access for developers. Create AWS IAM roles in each member account and manually assign permissions. Instruct developers to assume roles across accounts using the AWS CLI

Deploy an open-source identity provider (IdP) on Amazon EC2. Synchronize it with the on-premises Active Directory and use SAML to federate access to AWS accounts. Assign IAM roles to federated users based on SAML assertions

Your answer is correct

Use AWS Directory Service AD Connector to connect AWS to the on-premises Active Directory. Integrate AD Connector with AWS IAM Identity Center. Use permission sets to assign access to AWS accounts and resources based on Active Directory group membership

Deploy AWS Directory Service for Microsoft Active Directory in AWS. Establish a trust relationship with the on-premises Active Directory. Use IAM roles linked to AD groups to control access to AWS resources

Overall explanation

Correct option:

Use AWS Directory Service AD Connector to connect AWS to the on-premises Active Directory. Integrate AD Connector with AWS IAM Identity Center. Use permission sets to assign access to AWS accounts and resources based on Active Directory group membership

This solution provides a centralized, low-maintenance, and scalable identity management solution that integrates seamlessly with the company's existing on-premises Active Directory. AD Connector acts as a proxy to the on-prem AD, eliminating the need to deploy or manage a separate directory in the cloud. By integrating IAM Identity Center (formerly AWS SSO) with AD Connector, the company can use existing Active Directory group memberships to assign permission sets that define what AWS accounts and resources each user can access. This allows fine-grained, role-based access control across multiple accounts managed by AWS Organizations. IAM Identity Center also provides a single sign-on portal, reducing friction for users while maintaining compliance and security. The entire solution requires minimal operational overhead and scales well with growing teams and AWS accounts.

Incorrect options:

Deploy AWS Directory Service for Microsoft Active Directory in AWS. Establish a trust relationship with the on-premises Active Directory. Use IAM roles linked to AD groups to control access to AWS resources - While this solution provides centralized identity integration via trust relationships, managing a self-hosted or AWS-managed AD in the cloud adds significant operational overhead. You must maintain the AWS Directory Service

instance, monitor trust health, and manage role mappings manually. It does meet the technical need, but it's less efficient than a fully managed identity federation service such as IAM Identity Center (formerly AWS SSO). It introduces extra cost and maintenance that goes against the requirement for low operational effort.

Deploy an open-source identity provider (IdP) on Amazon EC2. Synchronize it with the on-premises Active Directory and use SAML to federate access to AWS accounts.

Assign IAM roles to federated users based on SAML assertions - While this solution is technically feasible, it introduces significant operational overhead, which contradicts the question's requirement. Managing an open-source IdP on EC2 - including scaling, patching, high availability, and synchronization with Active Directory is labor-intensive. Although it allows for SAML federation and AWS access, it lacks the native integration, multi-account visibility, and permission set management offered by IAM Identity Center. This approach may suit specific customization needs but fails the test of being simple and low maintenance.

Use AWS Control Tower to enable account access for developers. Create AWS IAM roles in each member account and manually assign permissions. Instruct developers to assume roles across accounts using the AWS CLI - Although AWS Control Tower helps set up a multi-account environment, this approach involves manual creation and assignment of IAM roles in each account, which is not scalable. Developers must use CLI-based role assumption, and administrators must maintain custom trust policies and permission boundaries. There's no centralized management of access or visibility across accounts, making this solution error-prone and operationally complex compared to using IAM Identity Center with group-based permission sets and automatic propagation across accounts.

References:

https://docs.aws.amazon.com/directoryservice/latest/admin-guide/directory_ad_connector.html

https://docs.aws.amazon.com/directoryservice/latest/admin-guide/directory_microsoft_ad.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_saml.html

<https://docs.aws.amazon.com/singlesignon/latest/userguide/what-is.html>

Domain

Design Secure Architectures

Question 12 Incorrect

A biotech research company needs to perform data analytics on real-time lab results provided by a partner organization. The partner stores these lab results in an Amazon RDS for MySQL instance within the partner's own AWS account. The research company has a private VPC that does not have internet access, Direct Connect, or a VPN connection. However, the company must establish secure and private connectivity to the RDS database in the partner's VPC. The solution must allow the research company to connect from its VPC while minimizing complexity and complying with data security requirements.

Which solution will meet these requirements?

Your answer is incorrect

Set up VPC peering between the company's VPC and the partner's VPC. Use AWS Transit Gateway in the partner's account to route traffic from the company's VPC to the database. Modify the RDS subnet route tables to allow access from the company's CIDR block

Correct answer

Instruct the partner to create a Network Load Balancer (NLB) in front of the Amazon RDS for MySQL instance. Use AWS PrivateLink to expose the NLB as an interface VPC endpoint in the research company's VPC

Instruct the partner to enable public access on the Amazon RDS instance and add a security group rule to allow inbound access from the company's IP range. The company accesses the database over the public internet through a NAT Gateway configured in a private subnet

Configure a client VPN endpoint in the company's account. Have researchers connect to the VPN from their local machines. Establish a Direct Connect gateway to the partner's VPC and route RDS traffic via this connection

Overall explanation

Correct option:

Instruct the partner to create a Network Load Balancer (NLB) in front of the Amazon RDS for MySQL instance. Use AWS PrivateLink to expose the NLB as an interface VPC endpoint in the research company's VPC

This is the correct and most secure solution. Since the RDS instance cannot be directly exposed via PrivateLink, the partner can create a Network Load Balancer (NLB) in front of a proxy or custom endpoint that connects to RDS (e.g., using RDS Proxy or an EC2-based MySQL proxy). The partner then creates a PrivateLink service (interface endpoint service) that the research company can connect to privately via an interface VPC endpoint in its own VPC. This architecture keeps traffic within the AWS network without requiring an internet gateway, VPN, or Direct Connect.

What is AWS PrivateLink?

[Download PDF](#)[RSS](#)

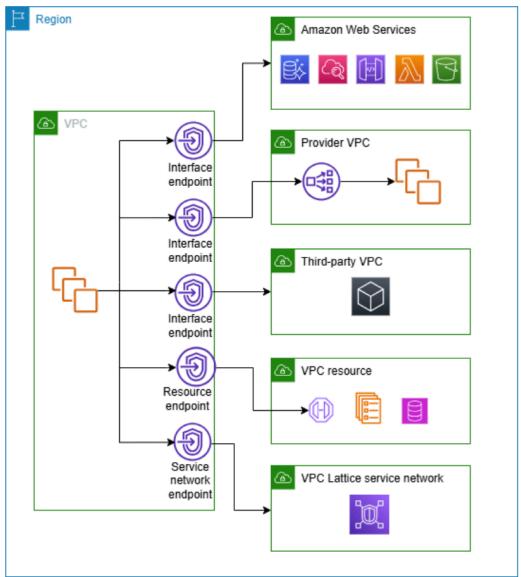
Focus mode

AWS PrivateLink is a highly available, scalable technology that you can use to privately connect your VPC to services and resources as if they were in your VPC. You do not need to use an internet gateway, NAT device, public IP address, AWS Direct Connect connection, or AWS Site-to-Site VPN connection to allow communication with the service or resource from your private subnets. Therefore, you control the specific API endpoints, sites, services, and resources that are reachable from your VPC.

Use cases

You can create VPC endpoints to connect clients in your VPC to services and resources that integrate with AWS PrivateLink. You can create your own VPC endpoint service and make it available to other AWS customers. For more information, see [AWS PrivateLink concepts](#).

In the following diagram, the VPC on the left has several Amazon EC2 instances in a private subnet and five VPC endpoints - three interface VPC endpoints, a resource VPC endpoint and a service-network VPC endpoint. The first interface VPC endpoint connects to an AWS service. The second interface VPC endpoint connects to a service hosted by another AWS account (a VPC endpoint service). The third interface VPC endpoint connects to an AWS Marketplace partner service. The resource VPC endpoint connects to a database. The service network VPC endpoint connects to a service network.

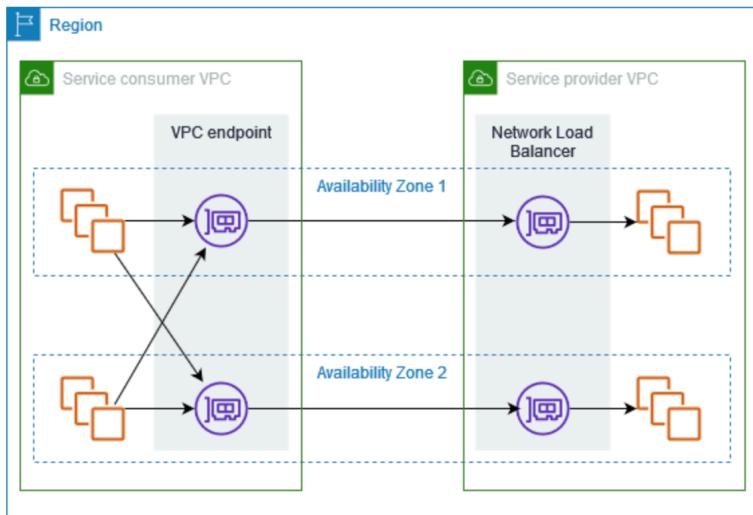


via - <https://docs.aws.amazon.com/vpc/latest/privatelink/what-is-privatelink.html>

Share your services through AWS PrivateLink:

Overview

The following diagram shows how you share your service that's hosted in AWS with other AWS customers, and how those customers connect to your service. As the service provider, you create a Network Load Balancer in your VPC as the service front end. You then select this load balancer when you create the VPC endpoint service configuration. You grant permission to specific AWS principals so that they can connect to your service. As a service consumer, the customer creates an interface VPC endpoint, which establishes connections between the subnets that they select from their VPC and your endpoint service. The load balancer receives requests from the service consumer and routes them to the targets hosting your service.



For low latency and high availability, we recommend that you make your service available in at least two Availability Zones.

via - <https://docs.aws.amazon.com/vpc/latest/privatelink/privatelink-share-your-services.html>

Incorrect options:

Instruct the partner to enable public access on the Amazon RDS instance and add a security group rule to allow inbound access from the company's IP range. The company accesses the database over the public internet through a NAT Gateway configured in a private subnet - This option violates the requirement for no internet access in the company's VPC. Publicly exposing RDS by enabling public access and accessing it via NAT over the internet also raises serious security and compliance concerns. Furthermore, a NAT Gateway cannot exist in a subnet that has no internet gateway, making this solution infeasible in the given environment.

Set up VPC peering between the company's VPC and the partner's VPC. Use AWS Transit Gateway in the partner's account to route traffic from the company's VPC to the database. Modify the RDS subnet route tables to allow access from the company's CIDR block - VPC peering allows direct private communication, but you cannot use Transit Gateway and VPC peering together for a single pair of VPCs—this is an architectural contradiction. You either use Transit Gateway to route traffic between VPCs or use VPC peering—not both. Additionally, RDS traffic routing through Transit Gateway requires more complex configuration and is unnecessary in this scenario.

Configure a client VPN endpoint in the company's account. Have researchers connect to the VPN from their local machines. Establish a Direct Connect gateway to the partner's VPC and route RDS traffic via this connection - This option introduces multiple incorrect assumptions. First, Direct Connect is not available, as per the question's conditions. Second, Client VPN is for user-based access, not VPC-to-VPC communication. This solution involves unnecessary complexity, adds VPN management, and fails to satisfy the requirement of connecting from the VPC itself to the RDS instance.

References:

<https://docs.aws.amazon.com/vpc/latest/privatelink/what-is-privatelink.html>

<https://docs.aws.amazon.com/vpc/latest/privatelink/privatelink-share-your-services.html>

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html>

<https://docs.aws.amazon.com/vpc/latest/tgw/what-is-transit-gateway.html>

<https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/what-is.html>

Domain

Design Secure Architectures

Question 13 Incorrect

A file-hosting service uses Amazon Simple Storage Service (Amazon S3) under the hood to power its storage offerings. Currently all the customer files are uploaded directly under a single Amazon S3 bucket. The engineering team has started seeing scalability issues where customer file uploads have started failing during the peak access hours with more than 5000 requests per second.

Which of the following is the MOST resource efficient and cost-optimal way of addressing this issue?

Change the application architecture to create a new Amazon S3 bucket for each day's data and then upload the daily files directly under that day's bucket

Your answer is incorrect

Change the application architecture to create a new Amazon S3 bucket for each customer and then upload each customer's files directly under the respective buckets

Correct answer

Change the application architecture to create customer-specific custom prefixes within the single Amazon S3 bucket and then upload the daily files into those prefixed locations

Change the application architecture to use Amazon Elastic File System (Amazon EFS) instead of Amazon S3 for storing the customers' uploaded files

Overall explanation

Correct option:

Change the application architecture to create customer-specific custom prefixes within the single Amazon S3 bucket and then upload the daily files into those prefixed locations

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Your applications can easily achieve thousands of transactions per second in request performance when uploading and retrieving storage from Amazon S3. Amazon S3 automatically scales to high request rates. For example, your application can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per prefix in a bucket.

There are no limits to the number of prefixes in a bucket. You can increase your read or write performance by parallelizing reads. For example, if you create 10 prefixes in an

Amazon S3 bucket to parallelize reads, you could scale your read performance to 55,000 read requests per second. Please see this example for more clarity on prefixes: if you have a file f1 stored in an S3 object path like so `s3://your_bucket_name/folder1/sub_folder_1/f1`, then `/folder1/sub_folder_1/` becomes the prefix for file f1.

Some data lake applications on Amazon S3 scan millions or billions of objects for queries that run over petabytes of data. These data lake applications achieve single-instance transfer rates that maximize the network interface used for their Amazon EC2 instance, which can be up to 100 Gb/s on a single instance. These applications then aggregate throughput across multiple instances to get multiple terabits per second. Therefore creating customer-specific custom prefixes within the single bucket and then uploading the daily files into those prefixed locations is the BEST solution for the given constraints.

Optimizing Amazon S3 Performance:

Best Practices Design Patterns: Optimizing Amazon S3 Performance

[PDF](#) | [Kindle](#) | [RSS](#)

Your applications can easily achieve thousands of transactions per second in request performance when uploading and retrieving storage from Amazon S3. Amazon S3 automatically scales to high request rates. For example, your application can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per [prefix](#) in a bucket. There are no limits to the number of prefixes in a bucket. You can increase your read or write performance by parallelizing reads. For example, if you create 10 prefixes in an Amazon S3 bucket to parallelize reads, you could scale your read performance to 55,000 read requests per second.

Some data lake applications on Amazon S3 scan millions or billions of objects for queries that run over petabytes of data. These data lake applications achieve single-instance transfer rates that maximize the network interface use for their [Amazon EC2](#) instance, which can be up to 100 Gb/s on a single instance. These applications then aggregate throughput across multiple instances to get multiple terabits per second.

Other applications are sensitive to latency, such as social media messaging applications. These applications can achieve consistent small object latencies (and first-byte-out latencies for larger objects) of roughly 100–200 milliseconds.

Other AWS services can also help accelerate performance for different application architectures. For example, if you want higher transfer rates over a single HTTP connection or single-digit millisecond latencies, use [Amazon CloudFront](#) or [Amazon ElastiCache](#) for caching with Amazon S3.

Additionally, if you want fast data transport over long distances between a client and an S3 bucket, use [Amazon S3 Transfer Acceleration](#). Transfer Acceleration uses the globally distributed edge locations in CloudFront to accelerate data transport over geographical distances. If your Amazon S3 workload uses server-side encryption with AWS Key Management Service (SSE-KMS), see [AWS KMS Limits](#) in the AWS Key Management Service Developer Guide for information about the request rates supported for your use case.

The following topics describe best practice guidelines and design patterns for optimizing performance for applications that use Amazon S3. This guidance supersedes any previous guidance on optimizing performance for Amazon S3. For example, previously Amazon S3 performance guidelines recommended randomizing prefix naming with hashed characters to optimize performance for frequent data retrievals. You no longer have to randomize prefix naming for performance, and can use sequential date-based naming for your prefixes. Refer to the [Performance Guidelines for Amazon S3](#) and [Performance Design Patterns for Amazon S3](#) for the most current information about performance optimization for Amazon S3.

via - <https://docs.aws.amazon.com/AmazonS3/latest/dev/optimizing-performance.html>

Incorrect options:

Change the application architecture to create a new Amazon S3 bucket for each customer and then upload each customer's files directly under the respective buckets - Creating a new Amazon S3 bucket for each new customer is an inefficient way of handling resource availability (S3 buckets need to be globally unique) as some customers may use the service sparingly but the bucket name is locked for them forever. Moreover, this is really not required as we can use S3 prefixes to improve the performance.

Change the application architecture to create a new Amazon S3 bucket for each day's data and then upload the daily files directly under that day's bucket - Creating a new

Amazon S3 bucket for each new day's data is also an inefficient way of handling resource availability (S3 buckets need to be globally unique) as some of the bucket names may not be available for daily data processing. Moreover, this is really not required as we can use S3 prefixes to improve the performance.

Change the application architecture to use Amazon Elastic File System (Amazon EFS) instead of Amazon S3 for storing the customers' uploaded files - Amazon EFS is a costlier storage option compared to Amazon S3, so it is ruled out.

Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/optimizing-performance.html>

Domain

Design High-Performing Architectures

Question 14 Incorrect

A large financial institution operates an on-premises data center with hundreds of petabytes of data managed on Microsoft's Distributed File System (DFS). The CTO wants the organization to transition into a hybrid cloud environment and run data-intensive analytics workloads that support DFS.

Which of the following AWS services can facilitate the migration of these workloads?

**AWS Directory Service for Microsoft Active Directory
(AWS Managed Microsoft AD)**

Correct answer

Amazon FSx for Windows File Server

Your answer is incorrect

Amazon FSx for Lustre

Microsoft SQL Server on AWS

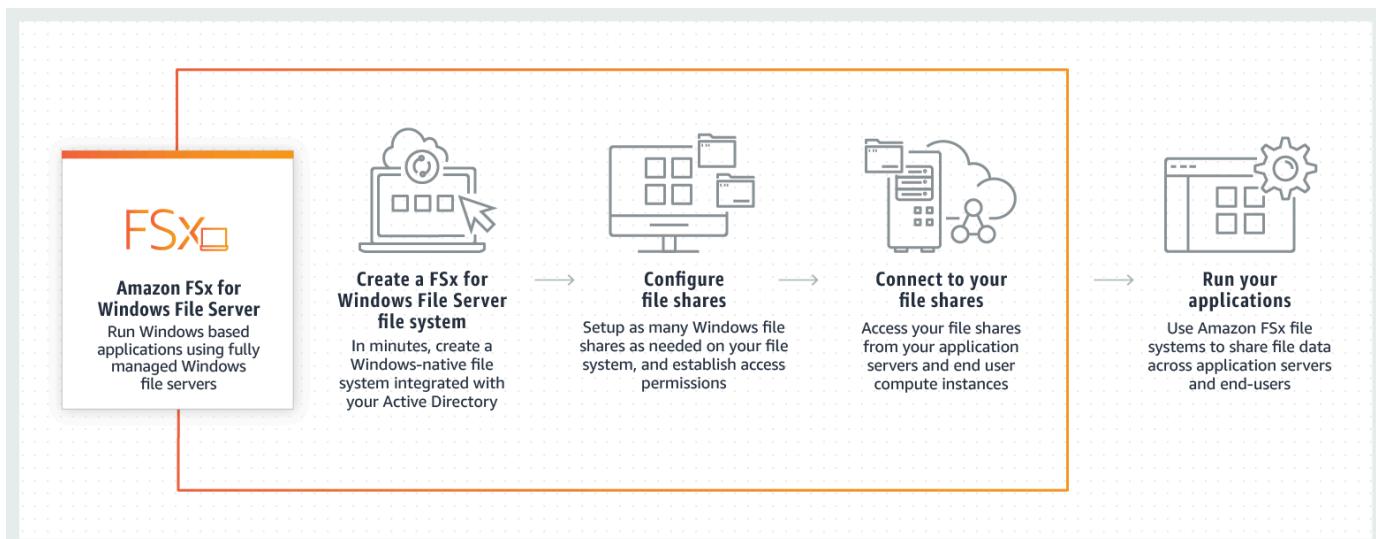
Overall explanation

Correct option:

Amazon FSx for Windows File Server

Amazon FSx for Windows File Server provides fully managed, highly reliable file storage that is accessible over the industry-standard Service Message Block (SMB) protocol. It is built on Windows Server, delivering a wide range of administrative features such as user quotas, end-user file restore, and Microsoft Active Directory (AD) integration. Amazon FSx supports the use of Microsoft's Distributed File System (DFS) to organize shares into a single folder structure up to hundreds of PB in size. So this option is correct.

How Amazon FSx for Windows File Server Works:



via - <https://aws.amazon.com/fsx/windows/>

Incorrect options:

Amazon FSx for Lustre

Amazon FSx for Lustre makes it easy and cost-effective to launch and run the world's most popular high-performance file system. It is used for workloads such as machine learning, high-performance computing (HPC), video processing, and financial modeling. Amazon FSx enables you to use Lustre file systems for any workload where storage speed matters. FSx for Lustre does not support Microsoft's Distributed File System (DFS), so this option is incorrect.

AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD)

AWS Directory Service for Microsoft Active Directory, also known as AWS Managed Microsoft AD, enables your directory-aware workloads and AWS resources to use managed Active Directory in the AWS Cloud. AWS Managed Microsoft AD is built on the actual Microsoft Active Directory and does not require you to synchronize or replicate data from your existing Active Directory to the cloud. AWS Managed Microsoft AD does not support Microsoft's Distributed File System (DFS), so this option is incorrect.

Microsoft SQL Server on AWS

Microsoft SQL Server on AWS offers you the flexibility to run Microsoft SQL Server database on AWS Cloud. Microsoft SQL Server on AWS does not support Microsoft's Distributed File System (DFS), so this option is incorrect.

Reference:

<https://aws.amazon.com/fsx/windows/>

Domain

Design High-Performing Architectures

Question 15 Incorrect

One of the biggest football leagues in Europe has granted the distribution rights for live streaming its matches in the USA to a silicon valley based streaming services company. As per the terms of distribution, the company must make sure that only users from the USA are able to live stream the matches on their platform. Users from other countries in the world must be denied access to these live-streamed matches.

Which of the following options would allow the company to enforce these streaming restrictions? (Select two)

Your selection is correct

Use Amazon Route 53 based geolocation routing policy to restrict distribution of content to only the locations in which you have distribution rights

Use Amazon Route 53 based weighted routing policy to restrict distribution of content to only the locations in which you have distribution rights

Use Amazon Route 53 based latency-based routing policy to restrict distribution of content to only the locations in which you have distribution rights

Your selection is correct

Use georestriction to prevent users in specific geographic locations from accessing content that you're distributing through a Amazon CloudFront web distribution

Your selection is incorrect

Use Amazon Route 53 based failover routing policy to restrict distribution of content to only the locations in which you have distribution rights

Overall explanation

Correct options:

Use Amazon Route 53 based geolocation routing policy to restrict distribution of content to only the locations in which you have distribution rights

Geolocation routing lets you choose the resources that serve your traffic based on the geographic location of your users, meaning the location that DNS queries originate from. For example, you might want all queries from Europe to be routed to an ELB load balancer in the Frankfurt region. You can also use geolocation routing to restrict the distribution of content to only the locations in which you have distribution rights.

Use georestriction to prevent users in specific geographic locations from accessing content that you're distributing through a Amazon CloudFront web distribution

You can use georestriction, also known as geo-blocking, to prevent users in specific geographic locations from accessing content that you're distributing through a Amazon CloudFront web distribution. When a user requests your content, Amazon CloudFront typically serves the requested content regardless of where the user is located. If you need to prevent users in specific countries from accessing your content, you can use the CloudFront geo restriction feature to do one of the following: Allow your users to access your content only if they're in one of the countries on a whitelist of approved countries. Prevent your users from accessing your content if they're in one of the countries on a blacklist of banned countries. So this option is also correct.

Amazon Route 53 Routing Policy Overview:

Choosing a routing policy

[PDF](#) | [Kindle](#) | [RSS](#)

When you create a record, you choose a routing policy, which determines how Amazon Route 53 responds to queries:

- **Simple routing policy** – Use for a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.
- **Failover routing policy** – Use when you want to configure active-passive failover.
- **Geolocation routing policy** – Use when you want to route traffic based on the location of your users.
- **Geoproximity routing policy** – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.
- **Latency routing policy** – Use when you have resources in multiple AWS Regions and you want to route traffic to the region that provides the best latency.
- **Multivalue answer routing policy** – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.
- **Weighted routing policy** – Use to route traffic to multiple resources in proportions that you specify.

via - <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html>

Incorrect options:

Use Amazon Route 53 based latency-based routing policy to restrict distribution of content to only the locations in which you have distribution rights - Use latency-based routing when you have resources in multiple AWS Regions and you want to route traffic to the region that provides the lowest latency. To use latency-based routing, you create latency records for your resources in multiple AWS Regions. When Amazon Route 53 receives a DNS query for your domain or subdomain (example.com or acme.example.com), it determines which AWS Regions you've created latency records for, determines which region gives the user the lowest latency, and then selects a latency record for that region. Route 53 responds with the value from the selected record, such as the IP address for a web server.

Use Amazon Route 53 based weighted routing policy to restrict distribution of content to only the locations in which you have distribution rights - Weighted routing lets you associate multiple resources with a single domain name (example.com) or subdomain name (acme.example.com) and choose how much traffic is routed to each resource. This

can be useful for a variety of purposes, including load balancing and testing new versions of the software.

Use Amazon Route 53 based failover routing policy to restrict distribution of content to only the locations in which you have distribution rights - Failover routing lets you route traffic to a resource when the resource is healthy or to a different resource when the first resource is unhealthy. The primary and secondary records can route traffic to anything from an Amazon S3 bucket that is configured as a website to a complex tree of records.

Weighted routing or failover routing or latency routing cannot be used to restrict the distribution of content to only the locations in which you have distribution rights. So all three options above are incorrect.

References:

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-geo>

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-geo>

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-geo>

Domain

Design Secure Architectures

Question 16 Correct

The product team at a startup has figured out a market need to support both stateful and stateless client-server communications via the application programming interface (APIs) developed using its platform. You have been hired by the startup as a solutions architect to build a solution to fulfill this market need using Amazon API Gateway.

Which of the following would you identify as correct?

Your answer is correct

Amazon API Gateway creates RESTful APIs that enable stateless client-server communication and Amazon API Gateway also creates WebSocket APIs that adhere to

the WebSocket protocol, which enables stateful, full-duplex communication between client and server

Amazon API Gateway creates RESTful APIs that enable stateful client-server communication and Amazon API Gateway also creates WebSocket APIs that adhere to the WebSocket protocol, which enables stateless, full-duplex communication between client and server

Amazon API Gateway creates RESTful APIs that enable stateless client-server communication and Amazon API Gateway also creates WebSocket APIs that adhere to the WebSocket protocol, which enables stateless, full-duplex communication between client and server

Amazon API Gateway creates RESTful APIs that enable stateful client-server communication and Amazon API Gateway also creates WebSocket APIs that adhere to the WebSocket protocol, which enables stateful, full-duplex communication between client and server

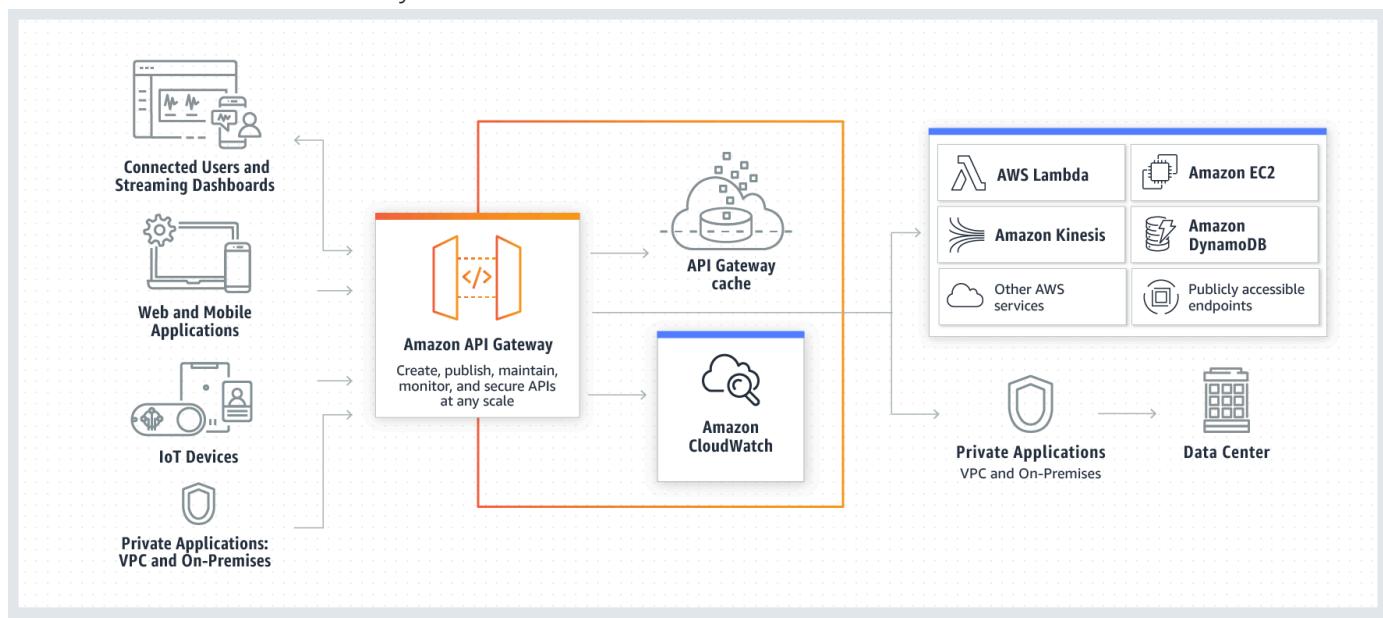
Overall explanation

Correct option:

Amazon API Gateway creates RESTful APIs that enable stateless client-server communication and Amazon API Gateway also creates WebSocket APIs that adhere to the WebSocket protocol, which enables stateful, full-duplex communication between client and server

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the front door for applications to access data, business logic, or functionality from your backend services. Using API Gateway, you can create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications.

How Amazon API Gateway Works:



via - <https://aws.amazon.com/api-gateway/>

Amazon API Gateway creates RESTful APIs that:

Are HTTP-based.

Enable stateless client-server communication.

Implement standard HTTP methods such as GET, POST, PUT, PATCH, and DELETE.

Amazon API Gateway creates WebSocket APIs that:

Adhere to the WebSocket protocol, which enables stateful, full-duplex communication between client and server. Route incoming messages based on message content.

So Amazon API Gateway supports stateless RESTful APIs as well as stateful WebSocket APIs. Therefore this option is correct.

Incorrect options:

Amazon API Gateway creates RESTful APIs that enable stateful client-server communication and Amazon API Gateway also creates WebSocket APIs that adhere to the WebSocket protocol, which enables stateful, full-duplex communication between client and server

Amazon API Gateway creates RESTful APIs that enable stateless client-server communication and Amazon API Gateway also creates WebSocket APIs that adhere to the WebSocket protocol, which enables stateless, full-duplex communication between client and server

Amazon API Gateway creates RESTful APIs that enable stateful client-server communication and Amazon API Gateway also creates WebSocket APIs that adhere to the WebSocket protocol, which enables stateless, full-duplex communication between client and server

These three options contradict the earlier details provided in the explanation. To summarize, Amazon API Gateway supports stateless RESTful APIs and stateful WebSocket APIs. Hence these options are incorrect.

Reference:

<https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html>

Domain

Design High-Performing Architectures

Question 17 Correct

A logistics company is building a multi-tier application to track the location of its trucks during peak operating hours. The company wants these data points to be accessible in real-time in its analytics platform via a REST API. The company has hired you as an AWS Certified Solutions Architect Associate to build a multi-tier solution to store and retrieve this location data for analysis.

Which of the following options addresses the given use case?

Leverage Amazon QuickSight with Amazon Redshift

Leverage Amazon API Gateway with AWS Lambda

Leverage Amazon Athena with Amazon S3

Your answer is correct

Leverage Amazon API Gateway with Amazon Kinesis Data Analytics

Overall explanation

Correct option:

Leverage Amazon API Gateway with Amazon Kinesis Data Analytics

You can use Kinesis Data Analytics to transform and analyze streaming data in real-time with Apache Flink. Kinesis Data Analytics enables you to quickly build end-to-end stream processing applications for log analytics, clickstream analytics, Internet of Things (IoT), ad tech, gaming, etc. The four most common use cases are streaming extract-transform-load (ETL), continuous metric generation, responsive real-time analytics, and interactive querying of data streams. Kinesis Data Analytics for Apache Flink applications provides your application 50 GB of running application storage per Kinesis Processing Unit (KPU).

Amazon API Gateway is a fully managed service that allows you to publish, maintain, monitor, and secure APIs at any scale. Amazon API Gateway offers two options to create RESTful APIs, HTTP APIs and REST APIs, as well as an option to create WebSocket APIs.

Amazon API Gateway:

Q: What API types are supported by Amazon API Gateway?

Amazon API Gateway offers two options to create RESTful APIs, HTTP APIs and REST APIs, as well as an option to create WebSocket APIs.

HTTP API: HTTP APIs are optimized for building APIs that proxy to AWS Lambda functions or HTTP backends, making them ideal for serverless workloads. They do not currently offer API management functionality.

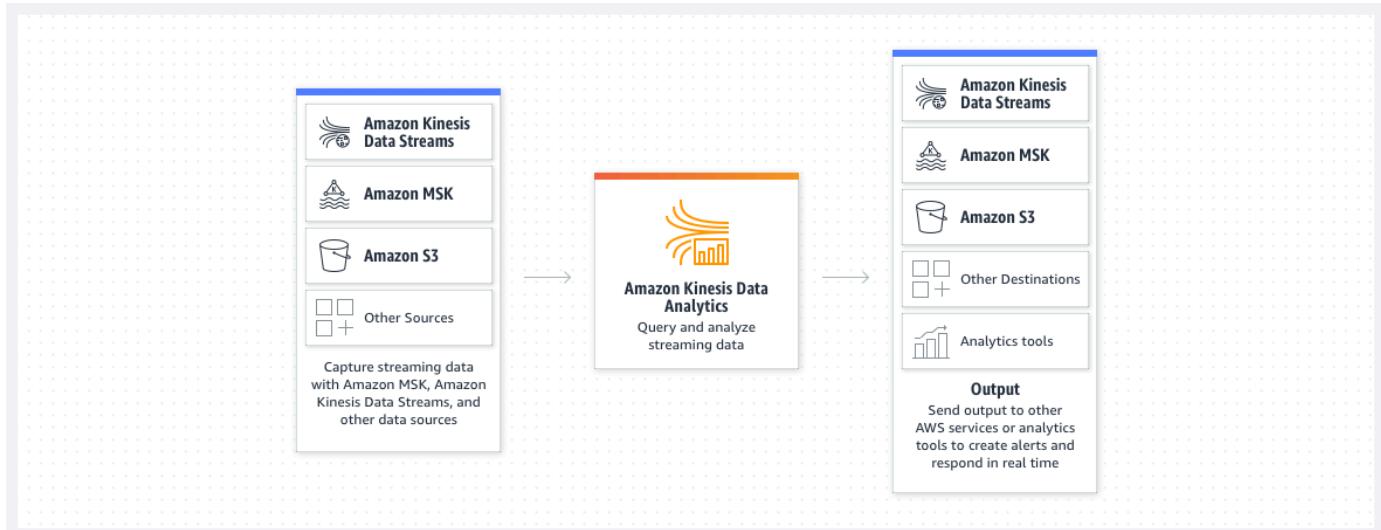
REST API: REST APIs offer API proxy functionality and API management features in a single solution. REST APIs offer API management features such as usage plans, API keys, publishing, and monetizing APIs.

WebSocket API: WebSocket APIs maintain a persistent connection between connected clients to enable real-time message communication. With WebSocket APIs in API Gateway, you can define backend integrations with AWS Lambda functions, Amazon Kinesis, or any HTTP endpoint to be invoked when messages are received from the connected clients.

via - <https://aws.amazon.com/blogs/aws/amazon-rds-custom-for-oracle-new-control-capabilities-in-database-environment/>

For the given use case, you can use Amazon API Gateway to create a REST API that handles incoming requests having location data from the trucks and sends it to the Kinesis Data Analytics application on the back end.

Amazon Kinesis Data Analytics:



via - <https://aws.amazon.com/kinesis/data-analytics/>

Incorrect options:

Leverage Amazon Athena with Amazon S3 - Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Athena

cannot be used to build a REST API to consume data from the source. So this option is incorrect.

Leverage Amazon QuickSight with Amazon Redshift - QuickSight is a cloud-native, serverless business intelligence service. Quicksight cannot be used to build a REST API to consume data from the source. Redshift is a fully managed AWS cloud data warehouse. So this option is incorrect.

Leverage Amazon API Gateway with AWS Lambda - You cannot use Lambda to store and retrieve the location data for analysis, so this option is incorrect.

References:

<https://docs.aws.amazon.com/apigateway/latest/developerguide/integrating-api-with-aws-services-kinesis.html>

<https://aws.amazon.com/kinesis/data-analytics/>

<https://aws.amazon.com/kinesis/data-analytics/faqs/>

Domain

Design High-Performing Architectures

Question 18 Correct

A US-based healthcare startup is building an interactive diagnostic tool for COVID-19 related assessments. The users would be required to capture their personal health records via this tool. As this is sensitive health information, the backup of the user data must be kept encrypted in Amazon Simple Storage Service (Amazon S3). The startup does not want to provide its own encryption keys but still wants to maintain an audit trail of when an encryption key was used and by whom.

Which of the following is the BEST solution for this use-case?

Use server-side encryption with customer-provided keys (SSE-C) to encrypt the user data on Amazon S3

Use server-side encryption with Amazon S3 managed keys (SSE-S3) to encrypt the user data on Amazon S3

Your answer is correct

Use server-side encryption with AWS Key Management Service keys (SSE-KMS) to encrypt the user data on Amazon S3

Use client-side encryption with client provided keys and then upload the encrypted user data to Amazon S3

Overall explanation

Correct option:

Use server-side encryption with AWS Key Management Service keys (SSE-KMS) to encrypt the user data on Amazon S3

AWS Key Management Service (AWS KMS) is a service that combines secure, highly available hardware and software to provide a key management system scaled for the cloud. When you use server-side encryption with AWS KMS (SSE-KMS), you can specify a customer-managed CMK that you have already created. SSE-KMS provides you with an audit trail that shows when your CMK was used and by whom. Therefore SSE-KMS is the correct solution for this use-case.

Server Side Encryption in S3:

Protecting data using server-side encryption

[PDF](#) | [Kindle](#) | [RSS](#)

Server-side encryption is the encryption of data at its destination by the application or service that receives it. Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypts it for you when you access it. As long as you authenticate your request and you have access permissions, there is no difference in the way you access encrypted or unencrypted objects. For example, if you share your objects using a presigned URL, that URL works the same way for both encrypted and unencrypted objects. Additionally, when you list objects in your bucket, the list API returns a list of all objects, regardless of whether they are encrypted.

 **Note**

You can't apply different types of server-side encryption to the same object simultaneously.

You have three mutually exclusive options, depending on how you choose to manage the encryption keys.

Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)

When you use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3), each object is encrypted with a unique key. As an additional safeguard, it encrypts the key itself with a master key that it regularly rotates. Amazon S3 server-side encryption uses one of the strongest block ciphers available, 256-bit Advanced Encryption Standard (AES-256), to encrypt your data. For more information, see [Protecting Data Using Server-Side Encryption with Amazon S3-Managed Encryption Keys \(SSE-S3\)](#).

Server-Side Encryption with Customer Master Keys (CMKs) Stored in AWS Key Management Service (SSE-KMS)

Server-Side Encryption with Customer Master Keys (CMKs) Stored in AWS Key Management Service (SSE-KMS) is similar to SSE-S3, but with some additional benefits and charges for using this service. There are separate permissions for the use of a CMK that provides added protection against unauthorized access of your objects in Amazon S3. SSE-KMS also provides you with an audit trail that shows when your CMK was used and by whom. Additionally, you can create and manage customer managed CMKs or use AWS managed CMKs that are unique to you, your service, and your Region. For more information, see [Protecting Data Using Server-Side Encryption with CMKs Stored in AWS Key Management Service \(SSE-KMS\)](#).

Server-Side Encryption with Customer-Provided Keys (SSE-C)

With Server-Side Encryption with Customer-Provided Keys (SSE-C), you manage the encryption keys and Amazon S3 manages the encryption, as it writes to disks, and decryption, when you access your objects. For more information, see [Protecting data using server-side encryption with customer-provided encryption keys \(SSE-C\)](#).

via - <https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html>

Incorrect options:

Use server-side encryption with Amazon S3 managed keys (SSE-S3) to encrypt the user data on Amazon S3 - When you use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3), each object is encrypted with a unique key. However this option does not provide the ability to audit trail the usage of the encryption keys.

Use server-side encryption with customer-provided keys (SSE-C) to encrypt the user data on Amazon S3 - With Server-Side Encryption with Customer-Provided Keys (SSE-C), you manage the encryption keys and Amazon S3 manages the encryption, as it writes to disks, and decryption when you access your objects. However this option does not provide the ability to audit trail the usage of the encryption keys.

Use client-side encryption with client provided keys and then upload the encrypted user data to Amazon S3 - Using client-side encryption is ruled out as the startup does not want to provide the encryption keys.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingKMSEncryption.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html>

Domain

Question 19 Incorrect

A gaming company is looking at improving the availability and performance of its global flagship application which utilizes User Datagram Protocol and needs to support fast regional failover in case an AWS Region goes down. The company wants to continue using its own custom Domain Name System (DNS) service.

Which of the following AWS services represents the best solution for this use-case?

Amazon Route 53

AWS Elastic Load Balancing (ELB)

Your answer is incorrect

Amazon CloudFront

Correct answer

AWS Global Accelerator

Overall explanation

Correct option:

AWS Global Accelerator

AWS Global Accelerator utilizes the Amazon global network, allowing you to improve the performance of your applications by lowering first-byte latency (the round trip time for a packet to go from a client to your endpoint and back again) and jitter (the variation of latency), and increasing throughput (the amount of time it takes to transfer data) as compared to the public internet.

AWS Global Accelerator improves performance for a wide range of applications over TCP or UDP by proxying packets at the edge to applications running in one or more AWS Regions. Global Accelerator is a good fit for non-HTTP use cases, such as gaming (UDP), IoT (MQTT), or Voice over IP, as well as for HTTP use cases that specifically require static IP addresses or deterministic, fast regional failover.

Incorrect options:

Amazon CloudFront - Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.

AWS Global Accelerator and Amazon CloudFront are separate services that use the AWS global network and its edge locations around the world. CloudFront improves performance for both cacheable content (such as images and videos) and dynamic content (such as API acceleration and dynamic site delivery), while Global Accelerator improves performance for a wide range of applications over TCP or UDP.

AWS Elastic Load Balancing (ELB) - Both of the services, ELB and Global Accelerator solve the challenge of routing user requests to healthy application endpoints. AWS Global Accelerator relies on ELB to provide the traditional load balancing features such as support for internal and non-AWS endpoints, pre-warming, and Layer 7 routing. However, while ELB provides load balancing within one Region, AWS Global Accelerator provides traffic management across multiple Regions.

A regional ELB load balancer is an ideal target for AWS Global Accelerator. By using a regional ELB load balancer, you can precisely distribute incoming application traffic across backends, such as Amazon EC2 instances or Amazon ECS tasks, within an AWS Region.

If you have workloads that cater to a global client base, AWS recommends that you use AWS Global Accelerator. If you have workloads hosted in a single AWS Region and used by

clients in and around the same Region, you can use an Application Load Balancer or Network Load Balancer to manage your resources.

Amazon Route 53 - Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost-effective way to route end users to Internet applications by translating names like `www.example.com` into the numeric IP addresses like `192.0.2.1` that computers use to connect to each other. Route 53 is ruled out as the company wants to continue using its own custom DNS service.

Reference:

<https://aws.amazon.com/global-accelerator/faqs/>

Domain

Design High-Performing Architectures

Question 20 Incorrect

A leading video streaming service delivers billions of hours of content from Amazon Simple Storage Service (Amazon S3) to customers around the world. Amazon S3 also serves as the data lake for its big data analytics solution. The data lake has a staging zone where intermediary query results are kept only for 24 hours. These results are also heavily referenced by other parts of the analytics pipeline.

Which of the following is the MOST cost-effective strategy for storing this intermediary query data?

Store the intermediary query results in Amazon S3 Glacier Instant Retrieval storage class

Correct answer

Store the intermediary query results in Amazon S3 Standard storage class

Store the intermediary query results in Amazon S3 One Zone-Infrequent Access storage class

Your answer is incorrect

Store the intermediary query results in Amazon S3 Standard-Infrequent Access storage class

Overall explanation

Correct option:

Store the intermediary query results in Amazon S3 Standard storage class

Amazon S3 Standard offers high durability, availability, and performance object storage for frequently accessed data. Because it delivers low latency and high throughput, S3 Standard is appropriate for a wide variety of use cases, including cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics. As there is no minimum storage duration charge and no retrieval fee (remember that intermediary query results are heavily referenced by other parts of the analytics pipeline), this is the MOST cost-effective storage class amongst the given options.

Incorrect options:

Store the intermediary query results in Amazon S3 Glacier Instant Retrieval storage class - Amazon S3 Glacier Instant Retrieval delivers the fastest access to archive storage, with the same throughput and milliseconds access as the S3 Standard and S3 Standard-IA storage classes. S3 Glacier Instant Retrieval is ideal for archive data that needs immediate access, such as medical images, news media assets, or user-generated content archives.

The minimum storage duration charge is 90 days, so this option is NOT cost-effective because intermediary query results need to be kept only for 24 hours. Hence this option is not correct.

Store the intermediary query results in Amazon S3 Standard-Infrequent Access storage class - Amazon S3 Standard-IA is for data that is accessed less frequently but requires rapid access when needed. S3 Standard-IA offers high durability, high throughput, and low latency of S3 Standard, with a low per GB storage price and per GB retrieval fee. This combination of low cost and high performance makes S3 Standard-IA ideal for long-term storage, backups, and as a data store for disaster recovery files. The minimum storage duration charge is 30 days, so this option is NOT cost-effective because intermediary query results need to be kept only for 24 hours. Hence this option is not correct.

Store the intermediary query results in Amazon S3 One Zone-Infrequent Access storage class - Amazon S3 One Zone-IA is for data that is accessed less frequently but requires rapid access when needed. Unlike other S3 Storage Classes which store data in a minimum of three Availability Zones (AZs), S3 One Zone-IA stores data in a single AZ and costs 20% less than S3 Standard-IA. The minimum storage duration charge is 30 days, so this option is NOT cost-effective because intermediary query results need to be kept only for 24 hours. Hence this option is not correct.

To summarize again, S3 Standard-IA and S3 One Zone-IA have a minimum storage duration charge of 30 days (so instead of 24 hours, you end up paying for 30 days). S3 Standard-IA and S3 One Zone-IA also have retrieval charges (as the results are heavily referenced by other parts of the analytics pipeline, so the retrieval costs would be pretty high). Therefore, these storage classes are not cost optimal for the given use-case.

Reference:

<https://aws.amazon.com/s3/storage-classes/>

Domain

Design Cost-Optimized Architectures

Question 21 Correct

An IT consultant is helping the owner of a medium-sized business set up an AWS account. What are the security recommendations he must follow while creating the AWS account root user? (Select two)

Your selection is correct

Enable Multi Factor Authentication (MFA) for the AWS account root user account

Encrypt the access keys and save them on Amazon S3

Create AWS account root user access keys and share those keys only with the business owner

Send an email to the business owner with details of the login username and password for the AWS root user. This will help the business owner to troubleshoot any login issues in future

Your selection is correct

Create a strong password for the AWS account root user

Overall explanation

Correct options:

Create a strong password for the AWS account root user

Enable Multi Factor Authentication (MFA) for the AWS account root user account

Here are some of the best practices while creating an AWS account root user:

- 1) Use a strong password to help protect account-level access to the AWS Management Console.
- 2) Never share your AWS account root user password or access keys with anyone.
- 3) If you do have an access key for your AWS account root user, delete it. If you must keep it, rotate (change) the access key regularly. You should not encrypt the access keys and save them on Amazon S3.
- 4) If you don't already have an access key for your AWS account root user, don't create one unless you absolutely need to.
- 5) Enable AWS multi-factor authentication (MFA) on your AWS account root user account.

AWS Root Account Security Best Practices:

- If you don't already have an access key for your AWS account root user, don't create one unless you absolutely need to. Instead, use your account email address and password to sign in to the AWS Management Console and [create an IAM user for yourself](#) that has administrative permissions.
- If you do have an access key for your AWS account root user, delete it. If you must keep it, rotate (change) the access key regularly. To delete or rotate your root user access keys, go to the [My Security Credentials page](#) in the AWS Management Console and sign in with your account's email address and password. You can manage your access keys in the **Access keys** section. For more information about rotating access keys, see [Rotating Access Keys](#).
- Never share your AWS account root user password or access keys with anyone. The remaining sections of this document discuss various ways to avoid having to share your AWS account root user credentials with other users. They also explain how to avoid having to embed them in an application.
- Use a strong password to help protect account-level access to the AWS Management Console. For information about managing your AWS account root user password, see [Changing the AWS Account Root User Password](#).
- Enable AWS multi-factor authentication (MFA) on your AWS account root user account. For more information, see [Using Multi-Factor Authentication \(MFA\) in AWS](#).

via - <https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html>

Incorrect options:

Encrypt the access keys and save them on Amazon S3 - AWS recommends that if you don't already have an access key for your AWS account root user, don't create one unless you absolutely need to. Even an encrypted access key for the root user poses a significant security risk. Therefore, this option is incorrect.

Create AWS account root user access keys and share those keys only with the business owner - AWS recommends that if you don't already have an access key for your AWS account root user, don't create one unless you absolutely need to. Hence, this option is incorrect.

Send an email to the business owner with details of the login username and password for the AWS root user. This will help the business owner to troubleshoot any login issues in future - AWS recommends that you should never share your AWS account root

user password or access keys with anyone. Sending an email with AWS account root user credentials creates a security risk as it can be misused by anyone reading the email. Hence, this option is incorrect.

Reference:

<https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#create-iam-users>

Domain

Design Secure Architectures

Question 22 Correct

A healthcare analytics company centralizes clinical and operational datasets in an Amazon S3-based data lake. Incoming data is ingested in Apache Parquet format from multiple hospitals and wearable health devices. To ensure quality and standardization, the company applies several transformation steps: anomaly filtering, datetime normalization, and aggregation by patient cohort. The company needs a solution to support a code-free interface that enables data engineers and business analysts to collaborate on data preparation workflows. The company also requires data lineage tracking, data profiling capabilities, and an easy way to share transformation logic across teams without writing or managing code.

Which AWS solution best meets these requirements?

Use AWS Glue Studio's visual canvas to design data transformation workflows on top of the Parquet files in Amazon S3. Configure Glue Studio jobs to run these transformations without writing code. Share the job definitions with team members for reuse. Use the visual job editor to track transformation progress and inspect profiling statistics for each dataset column

Your answer is correct

Use AWS Glue DataBrew to visually build transformation workflows on top of the raw Parquet files in S3. Use DataBrew recipes to track, audit, and share the transformation steps with others. Enable data profiling to inspect column statistics, null values, and data types across datasets

Use Amazon AppFlow to move and transform Parquet files in S3. Configure AppFlow transformations and mappings within the visual interface. Share flows with collaborators through AWS IAM policies and scheduled executions

Create Amazon Athena SQL queries to perform transformation steps directly on S3. Store queries in AWS Glue Data Catalog and share saved queries with other users through Amazon Athena's query editor

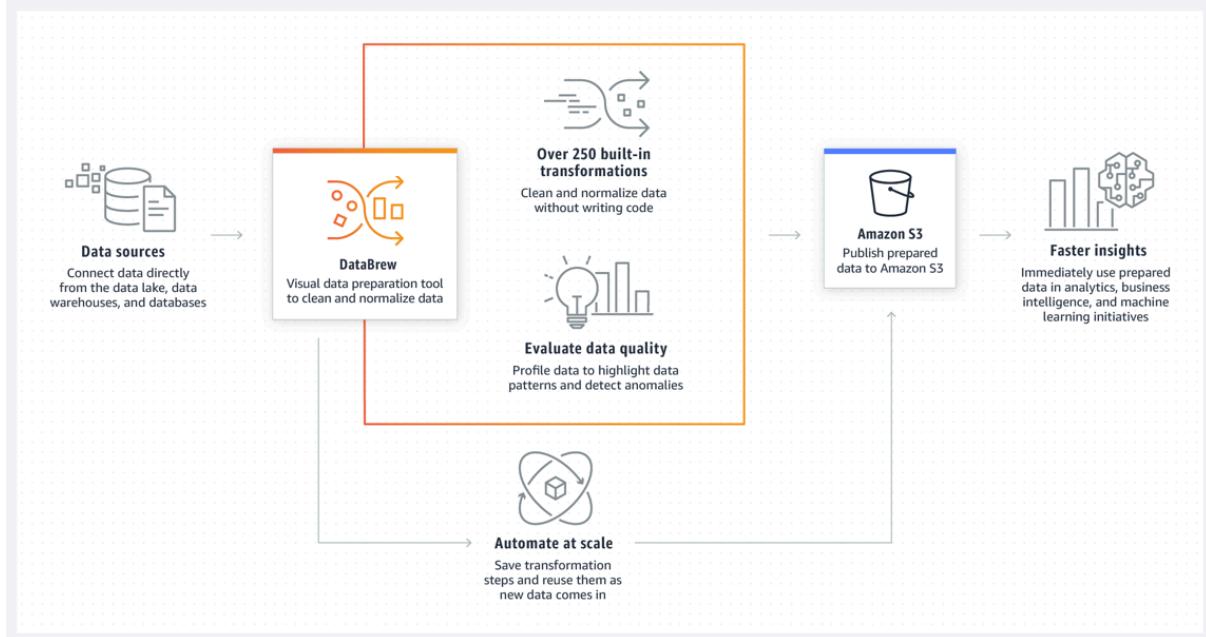
Overall explanation

Correct option:

Use AWS Glue DataBrew to visually build transformation workflows on top of the raw Parquet files in S3. Use DataBrew recipes to track, audit, and share the transformation steps with others. Enable data profiling to inspect column statistics, null values, and data types across datasets

AWS Glue DataBrew is a fully managed, serverless data preparation tool that enables users to visually clean, transform, and normalize raw datasets—all without writing any code. It is specifically designed for data analysts, business users, and data engineers who need to prepare data for analytics or machine learning workflows using an intuitive point-and-click interface.

What is AWS Glue DataBrew?:



via - <https://docs.aws.amazon.com/databrew/latest/dg/what-is.html>

In the scenario, the company ingests Apache Parquet files into Amazon S3 and needs to apply multiple transformation steps such as anomaly filtering, standardizing date formats, and computing aggregates. With DataBrew, users can perform these tasks through a drag-and-drop UI and configure complex workflows using reusable “recipes”—collections of ordered transformation steps that can be saved, versioned, and shared with others in the organization.

Furthermore, DataBrew provides built-in data profiling capabilities. It automatically generates column-level statistics such as distributions, cardinality, null value counts, standard deviations, and outlier detection. This visibility enables better data quality assessment before the data is used for reporting or machine learning.

DataBrew’s recipe model supports auditability and traceability, satisfying the requirement for data lineage. All transformation steps are recorded and can be exported or replicated across projects. The output of a DataBrew job is written back to Amazon S3, keeping the transformed data accessible for downstream tools like Amazon Athena, Redshift, or QuickSight.

Because it's serverless, there's no infrastructure to manage, and it integrates seamlessly with other AWS services. This makes it ideal for teams that want a no-code, collaborative, auditable, and profile-rich data transformation tool for Parquet-based data lakes in S3.

Incorrect options:

Use Amazon AppFlow to move and transform Parquet files in S3. Configure AppFlow transformations and mappings within the visual interface. Share flows with collaborators through AWS IAM policies and scheduled executions - Amazon AppFlow is used for data movement and basic transformation between SaaS applications (like Salesforce, Zendesk) and AWS services, but it does not support transforming or managing

datasets already in S3 at scale. It also lacks comprehensive transformation workflows, data profiling, and data lineage capabilities required for complex ETL pipelines involving Parquet files.

Create Amazon Athena SQL queries to perform transformation steps directly on S3. Store queries in AWS Glue Data Catalog and share saved queries with other users through Amazon Athena's query editor - Athena enables running ad hoc SQL queries directly on S3-stored data and is suitable for technical users. However, it requires SQL skills, lacks a visual transformation interface, and does not provide built-in data profiling or visual lineage. While users can save and share queries, Athena is better suited for exploration and reporting than for building shareable, no-code transformation pipelines.

Use AWS Glue Studio's visual canvas to design data transformation workflows on top of the Parquet files in Amazon S3. Configure Glue Studio jobs to run these transformations without writing code. Share the job definitions with team members for reuse. Use the visual job editor to track transformation progress and inspect profiling statistics for each dataset column - While AWS Glue Studio does offer a low-code visual interface for building ETL workflows, it is still primarily intended for developers and data engineers. The canvas generates and manages Apache Spark scripts, which may require customization and still involve code for advanced logic. More importantly, Glue Studio does not natively provide built-in data profiling or column-level statistics, unlike AWS Glue DataBrew. Additionally, Glue Studio is less suited for collaboration among non-technical users, and lacks the ability to share visual "recipes" with reusable, human-readable steps like DataBrew offers.

References:

<https://docs.aws.amazon.com/databrew/latest/dg/what-is.html>

<https://aws.amazon.com/blogs/big-data/use-aws-glue-databrew-recipes-in-your-aws-glue-studio-visual-etl-jobs/>

<https://docs.aws.amazon.com/appflow/latest/userguide/what-is-appflow.html>

<https://docs.aws.amazon.com/athena/latest/ug/querying.html>

<https://docs.aws.amazon.com/emr/latest/EMR-Serverless-UserGuide/emr-serverless.html>

Domain

Design Resilient Architectures

Question 23 Correct

A retail company has developed a REST API which is deployed in an Auto Scaling group behind an Application Load Balancer. The REST API stores the user data in Amazon DynamoDB and any static content, such as images, are served via Amazon Simple Storage Service (Amazon S3). On analyzing the usage trends, it is found that 90% of the read requests are for commonly accessed data across all users.

As a Solutions Architect, which of the following would you suggest as the MOST efficient solution to improve the application performance?

Enable ElastiCache Redis for DynamoDB and Amazon CloudFront for Amazon S3

Enable Amazon DynamoDB Accelerator (DAX) for Amazon DynamoDB and ElastiCache Memcached for Amazon S3

Enable ElastiCache Redis for DynamoDB and ElastiCache Memcached for Amazon S3

Your answer is correct

Enable Amazon DynamoDB Accelerator (DAX) for Amazon DynamoDB and Amazon CloudFront for Amazon S3

Overall explanation

Correct option:

Enable Amazon DynamoDB Accelerator (DAX) for Amazon DynamoDB and Amazon CloudFront for Amazon S3

Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache for Amazon DynamoDB that delivers up to a 10 times performance improvement—from milliseconds to microseconds—even at millions of requests per second.

Amazon DynamoDB Accelerator (DAX) is tightly integrated with Amazon DynamoDB—you simply provision a DAX cluster, use the DAX client SDK to point your existing Amazon DynamoDB API calls at the DAX cluster, and let DAX handle the rest. Because DAX is API-compatible with Amazon DynamoDB, you don't have to make any functional application code changes. DAX is used to natively cache Amazon DynamoDB reads.

Amazon CloudFront is a content delivery network (CDN) service that delivers static and dynamic web content, video streams, and APIs around the world, securely and at scale. By design, delivering data out of Amazon CloudFront can be more cost-effective than delivering it from S3 directly to your users.

When a user requests content that you serve with CloudFront, their request is routed to a nearby Edge Location. If CloudFront has a cached copy of the requested file, CloudFront delivers it to the user, providing a fast (low-latency) response. If the file they've requested isn't yet cached, CloudFront retrieves it from your origin – for example, the Amazon S3 bucket where you've stored your content.

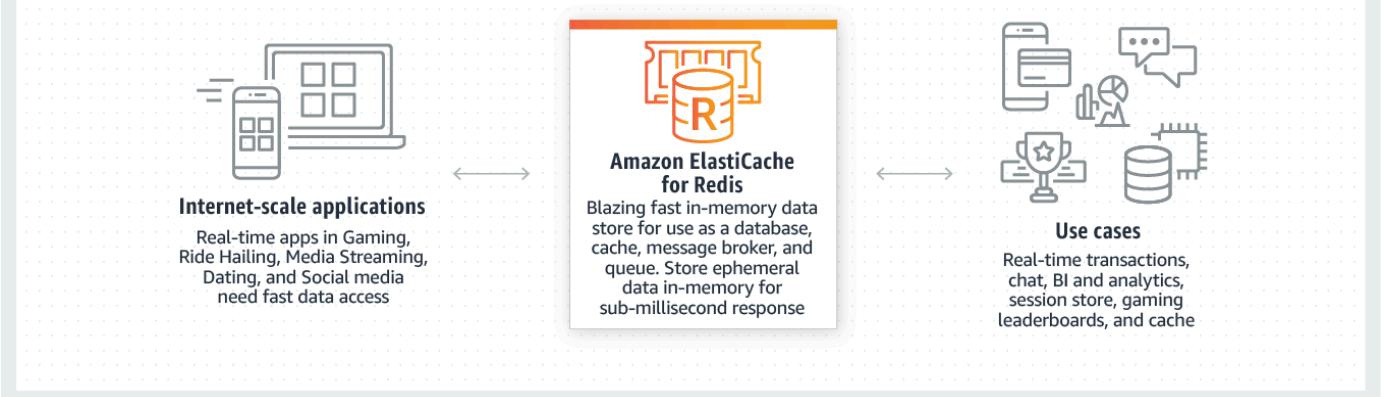
So, you can use Amazon CloudFront to improve application performance to serve static content from Amazon S3.

Incorrect options:

Enable ElastiCache Redis for DynamoDB and Amazon CloudFront for Amazon S3

Amazon ElastiCache for Redis is a blazing fast in-memory data store that provides sub-millisecond latency to power internet-scale real-time applications. Amazon ElastiCache for Redis is a great choice for real-time transactional and analytical processing use cases such as caching, chat/messaging, gaming leaderboards, geospatial, machine learning, media streaming, queues, real-time analytics, and session store.

Amazon ElastiCache for Redis Overview:



via - <https://aws.amazon.com/elasticsearch/redis/>

Although you can integrate Redis with DynamoDB, it's much more involved than using DAX which is a much better fit.

Enable Amazon DynamoDB Accelerator (DAX) for Amazon DynamoDB and ElastiCache Memcached for Amazon S3

Enable ElastiCache Redis for DynamoDB and ElastiCache Memcached for Amazon S3

Amazon ElastiCache for Memcached is a Memcached-compatible in-memory key-value store service that can be used as a cache or a data store. Amazon ElastiCache for Memcached is a great choice for implementing an in-memory cache to decrease access latency, increase throughput, and ease the load off your relational or NoSQL database.

Amazon ElastiCache Memcached cannot be used as a cache to serve static content from Amazon S3, so both these options are incorrect.

References:

<https://aws.amazon.com/dynamodb/dax/>

<https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/>

<https://aws.amazon.com/elasticsearch/redis/>

Domain

Design Cost-Optimized Architectures

Question 24 Correct

A digital event-ticketing platform hosts its core transaction-processing service on AWS. The service runs on Amazon EC2 instances and stores finalized transactions in an Amazon Aurora

PostgreSQL database. During periods of high user activity - such as flash ticket sales or holiday promotions - the application begins timing out, causing failed or delayed purchases. A solutions architect has been asked to redesign the backend for scalability and cost-efficiency, without reengineering the database layer.

Which combination of actions will meet these goals in the most cost-effective and scalable manner? (Select two)

Your selection is correct

Implement Amazon RDS Proxy between the application and the Aurora PostgreSQL cluster. Deploy EC2 instances in an Auto Scaling group to retry transactions as needed

Your selection is correct

Modify the application to publish purchase events to an Amazon SQS queue. Launch an Auto Scaling group of EC2 workers that poll the queue and process purchases asynchronously

Deploy read replicas for the Aurora database in another Region and configure EC2 instances to read and write from the nearest replica based on latency

Deploy an Amazon API Gateway with throttling and usage plans to slow down incoming purchase requests during peak times and maintain application stability

Use an Amazon ElastiCache cluster to cache database queries. Configure the application to store purchase transactions in the cache before writing to the database

Overall explanation

Correct options:

Implement Amazon RDS Proxy between the application and the Aurora PostgreSQL cluster. Deploy EC2 instances in an Auto Scaling group to retry transactions as needed

Amazon RDS Proxy helps manage connection pooling and throttling, which are common causes of Aurora timeouts during high-load scenarios. It maintains a pool of database connections and reuses them across clients, reducing the overhead of frequent connection creation. Pairing this with an Auto Scaling group of EC2 instances ensures that application compute can scale dynamically based on demand. This combination improves application resiliency, reduces Aurora connection stress, and addresses peak usage efficiently and cost-effectively.

Modify the application to publish purchase events to an Amazon SQS queue. Launch an Auto Scaling group of EC2 workers that poll the queue and process purchases asynchronously

Introducing Amazon SQS decouples the web tier from the processing logic, allowing the system to buffer bursts in transaction load without overwhelming the database. EC2 workers in an Auto Scaling group can be dynamically scaled to consume and process queue messages based on throughput needs. This pattern ensures durability, resiliency, and elastic scalability, especially during peak usage. It's also cost-effective because compute usage grows and shrinks with actual demand.

Incorrect Options:

Use an Amazon ElastiCache cluster to cache database queries. Configure the application to store purchase transactions in the cache before writing to the database - While ElastiCache is effective for read-heavy workloads, it's not designed for durable transactional writes. Caching transactional data—especially before it is persisted in a database—risks data loss if the cache node fails. In addition, caching will not prevent database connection saturation under write-heavy load. This approach does not solve the scalability issues related to timeouts during concurrent write operations.

Deploy an Amazon API Gateway with throttling and usage plans to slow down incoming purchase requests during peak times and maintain application stability - While API Gateway usage plans and throttling can help protect backend services from overload, they do so by delaying or rejecting traffic, which could worsen the customer experience during peak demand. This does not address the need for backend scalability, and it doesn't improve throughput or processing capacity. Instead, it suppresses demand rather than meeting it.

Deploy read replicas for the Aurora database in another Region and configure EC2 instances to read and write from the nearest replica based on latency - This option acts as a distractor. While read replicas improve read scalability, they are read-only and cannot handle write traffic.

References:

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/rds-proxy.html>

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/welcome.html>

<https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/WhatIs.html>

Domain

Design High-Performing Architectures

Question 25 Incorrect

An Electronic Design Automation (EDA) application produces massive volumes of data that can be divided into two categories. The 'hot data' needs to be both processed and stored quickly in a parallel and distributed fashion. The 'cold data' needs to be kept for reference with quick access for reads and updates at a low cost.

Which of the following AWS services is BEST suited to accelerate the aforementioned chip design process?

Correct answer

Amazon FSx for Lustre

AWS Glue

Your answer is incorrect

Amazon EMR

Overall explanation

Correct option:

Amazon FSx for Lustre

Amazon FSx for Lustre makes it easy and cost-effective to launch and run the world's most popular high-performance file system. It is used for workloads such as machine learning, high-performance computing (HPC), video processing, and financial modeling. The open-source Lustre file system is designed for applications that require fast storage – where you want your storage to keep up with your compute. FSx for Lustre integrates with Amazon S3, making it easy to process data sets with the Lustre file system. When linked to an S3 bucket, an FSx for Lustre file system transparently presents S3 objects as files and allows you to write changed data back to S3.

FSx for Lustre provides the ability to both process the 'hot data' in a parallel and distributed fashion as well as easily store the 'cold data' on Amazon S3. Therefore this option is the BEST fit for the given problem statement.

Incorrect options:

Amazon FSx for Windows File Server - Amazon FSx for Windows File Server provides fully managed, highly reliable file storage that is accessible over the industry-standard Service Message Block (SMB) protocol. It is built on Windows Server, delivering a wide range of administrative features such as user quotas, end-user file restore, and Microsoft Active Directory (AD) integration. FSx for Windows does not allow you to present S3 objects as files and does not allow you to write changed data back to S3. Therefore you cannot reference the "cold data" with quick access for reads and updates at low cost. Hence this option is not correct.

Amazon EMR - Amazon EMR is the industry-leading cloud big data platform for processing vast amounts of data using open source tools such as Apache Spark, Apache Hive, Apache HBase, Apache Flink, Apache Hudi, and Presto. Amazon EMR uses Hadoop, an open-source framework, to distribute your data and processing across a resizable cluster of Amazon EC2 instances. EMR does not offer the same storage and processing speed as FSx for Lustre. So it is not the right fit for the given high-performance workflow scenario.

AWS Glue - AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy for customers to prepare and load their data for analytics. AWS Glue job is meant to be used for batch ETL data processing. AWS Glue does not offer the same storage and processing speed as FSx for Lustre. So it is not the right fit for the given high-performance workflow scenario.

References:

<https://aws.amazon.com/fsx/lustre/>

<https://aws.amazon.com/fsx/windows/faqs/>

Domain

Design High-Performing Architectures

Question 26 Correct

A company runs a data processing workflow that takes about 60 minutes to complete. The workflow can withstand disruptions and it can be started and stopped multiple times.

Which is the most cost-effective solution to build a solution for the workflow?

Use Amazon EC2 reserved instances to run the workflow processes

Your answer is correct

Use Amazon EC2 spot instances to run the workflow processes

Use Amazon EC2 on-demand instances to run the workflow processes

Use AWS Lambda function to run the workflow processes

Overall explanation

Correct option:

Use Amazon EC2 spot instances to run the workflow processes

Amazon EC2 instance types:

On-Demand

With On-Demand instances, you pay for compute capacity by the hour or the second depending on which instances you run. No longer-term commitments or upfront payments are needed. You can increase or decrease your compute capacity depending on the demands of your application and only pay the specified per hourly rates for the instance you use.

On-Demand instances are recommended for:

- Users that prefer the low cost and flexibility of Amazon EC2 without any up-front payment or long-term commitment
- Applications with short-term, spiky, or unpredictable workloads that cannot be interrupted
- Applications being developed or tested on Amazon EC2 for the first time

[See On-Demand pricing »](#)

Spot instances

Amazon EC2 Spot instances allow you to request spare Amazon EC2 computing capacity for up to 90% off the On-Demand price. [Learn More](#).

Spot instances are recommended for:

- Applications that have flexible start and end times
- Applications that are only feasible at very low compute prices
- Users with urgent computing needs for large amounts of additional capacity

[See Spot pricing »](#)

Savings Plans

Savings Plans are a flexible pricing model that offer low prices on EC2 and Fargate usage, in exchange for a commitment to a consistent amount of usage (measured in \$/hour) for a 1 or 3 year term.

Dedicated Hosts

A Dedicated Host is a physical EC2 server dedicated for your use. Dedicated Hosts can help you reduce costs by allowing you to use your existing server-bound software licenses, including Windows Server, SQL Server, and SUSE Linux Enterprise Server (subject to your license terms), and can also help you meet compliance requirements. [Learn more](#).

- Can be purchased On-Demand (hourly).
- Can be purchased as a Reservation for up to 70% off the On-Demand price.

[See Dedicated pricing »](#)

Reserved Instances

Reserved Instances provide you with a significant discount (up to 75%) compared to On-Demand instance pricing. In addition, when Reserved Instances are assigned to a specific Availability Zone, they provide a capacity reservation, giving you additional confidence in your ability to launch instances when you need them.

For applications that have steady state or predictable usage, Reserved Instances can provide significant savings compared to using On-Demand instances. See [How to Purchase Reserved Instances](#) for more information.

Reserved Instances are recommended for:

- Applications with steady state usage
- Applications that may require reserved capacity
- Customers that can commit to using EC2 over a 1 or 3 year term to reduce their total computing costs

via - <https://aws.amazon.com/ec2/pricing/>

Amazon EC2 Spot instances allow you to request spare Amazon EC2 computing capacity for up to 90% off the On-Demand price.

Spot instances are recommended for:

Applications that have flexible start and end times
Applications that are feasible only at very low compute prices
Users with urgent computing needs for large amounts of additional capacity

For the given use case, spot instances offer the most cost-effective solution as the workflow can withstand disruptions and can be started and stopped multiple times.

For example, considering a process that runs for an hour and needs about 1024 MB of memory, spot instance pricing for a t2.micro instance (having 1024 MB of RAM) is \$0.0035 per hour.

Spot instance pricing:

Spot Instance Prices

Spot Instances

Note: T4g and T3 instances launch as unlimited by default. If you launch T4g or T3 Spot Instances as unlimited and plan to use them immediately and for a short duration, with no idle time for accruing CPU credits, you will incur charges for surplus credits. If the average CPU usage over a 24-hour period exceeds the baseline, you will also incur charges for surplus credits. We recommend that you launch your T4g or T3 Spot Instances in [standard mode](#) to avoid paying higher costs. For more information, see [Surplus Credits Can Incur Charges](#) and [Spot Instance limits](#).

Region: US East (Ohio)

US East (Ohio)

Linux/UNIX Usage

Windows Usage

General Purpose - Current Generation

a1.medium	\$0.0049 per Hour	N/A*
a1.large	\$0.0099 per Hour	N/A*
a1.xlarge	\$0.0197 per Hour	N/A*
a1.2xlarge	\$0.0394 per Hour	N/A*
a1.4xlarge	\$0.0788 per Hour	N/A*
a1.metal	\$0.0788 per Hour	N/A*
t2.micro	\$0.0035 per Hour	\$0.0081 per Hour
t2.small	\$0.0069 per Hour	\$0.0159 per Hour
t2.medium	\$0.0139 per Hour	\$0.0319 per Hour
t2.large	\$0.0278 per Hour	\$0.0558 per Hour

via - <https://aws.amazon.com/ec2/spot/pricing/>

Contrast this with the pricing of a Lambda function (having 1024 MB of allocated memory), which comes out to \$0.0000000167 per 1ms or \$0.06 per hour ($\$0.0000000167 * 1000 * 60 * 60$ per hour).

AWS Lambda function pricing:

AWS Lambda Pricing

Region: US East (Ohio) ▾

Architecture	Duration	Requests
x86 Price		
First 6 Billion GB-seconds / month	\$0.0000166667 for every GB-second	\$0.20 per 1M requests
Next 9 Billion GB-seconds / month	\$0.000015 for every GB-second	\$0.20 per 1M requests
Over 15 Billion GB-seconds / month	\$0.0000133334 for every GB-second	\$0.20 per 1M requests
Arm Price		
First 7.5 Billion GB-seconds / month	\$0.0000133334 for every GB-second	\$0.20 per 1M requests
Next 11.25 Billion GB-seconds / month	\$0.0000120001 for every GB-second	\$0.20 per 1M requests
Over 18.75 Billion GB-seconds / month	\$0.0000106667 for every GB-second	\$0.20 per 1M requests

Duration cost depends on the amount of memory you allocate to your function. You can allocate any amount of memory to your function between 128 MB and 10,240 MB, in 1 MB increments. The table below contains a few examples of the price per 1 ms associated with different memory sizes, for usage falling within the first pricing tier – for example, up to 6 billion GB-seconds / month in US East (Ohio).

x86 Price	Arm Price
Region: US East (Ohio) ▾	
Memory (MB)	Price per 1ms
128	\$0.0000000021
512	\$0.0000000083
1024	\$0.0000000167
1536	\$0.0000000250

via - <https://aws.amazon.com/lambda/pricing/>

Thus, a spot instance turns out to be about 20 times cost effective than a Lambda function to meet the requirements of the given use case.

Incorrect options:

Use AWS Lambda function to run the workflow processes - As mentioned in the explanation above, a Lambda function turns out to be 20 times more expensive than a spot instance to meet the workflow requirements of the given use case, so this option is incorrect. You should also note that the maximum execution time of a Lambda function is 15 minutes, so the workflow process would be disrupted for sure. On the other hand, it is

certainly possible that the workflow process can be completed in a single run on the spot instance (the average frequency of stop instance interruption across all Regions and instance types is <10%).

Use Amazon EC2 on-demand instances to run the workflow processes

Use Amazon EC2 reserved instances to run the workflow processes

You should note that both on-demand and reserved instances are more expensive than spot instances. In addition, reserved instances have a term of 1 year or 3 years, so they are not suited for the given workflow. Therefore, both these options are incorrect.

References:

<https://aws.amazon.com/ec2/pricing/>

<https://aws.amazon.com/ec2/spot/pricing/>

<https://aws.amazon.com/lambda/pricing/>

<https://aws.amazon.com/ec2/spot/instance-advisor/>

Domain

Design Cost-Optimized Architectures

Question 27 Incorrect

An audit department generates and accesses the audit reports only twice in a financial year. The department uses AWS Step Functions to orchestrate the report creating process that has failover and retry scenarios built into the solution. The underlying data to create these audit reports is stored on Amazon S3, runs into hundreds of Terabytes and should be available with millisecond latency.

As an AWS Certified Solutions Architect – Associate, which is the MOST cost-effective storage class that you would recommend to be used for this use-case?

Your answer is incorrect

Amazon S3 Intelligent-Tiering (S3 Intelligent-Tiering)

Amazon S3 Standard

Amazon S3 Glacier Deep Archive

Correct answer

Amazon S3 Standard-Infrequent Access (S3 Standard-IA)

Overall explanation

Correct option:

Amazon S3 Standard-Infrequent Access (S3 Standard-IA)

Since the data is accessed only twice in a financial year but needs rapid access when required, the most cost-effective storage class for this use-case is Amazon S3 Standard-IA. S3 Standard-IA storage class is for data that is accessed less frequently but requires rapid access when needed. S3 Standard-IA matches the high durability, high throughput, and low latency of S3 Standard, with a low per GB storage price and per GB retrieval fee. Amazon Standard-IA is designed for 99.9% availability compared to 99.99% availability of Amazon S3 Standard. However, the report creation process has failover and retry scenarios built into the workflow, so in case the data is not available owing to the 99.9% availability of Amazon S3 Standard-IA, the job will be auto re-invoked till data is successfully retrieved. Therefore this is the correct option.

Amazon S3 Storage Classes Overview:

	S3 Standard	S3 Intelligent-Tiering*	S3 Standard-IA	S3 One Zone-IA†	S3 Glacier Instant Retrieval	S3 Glacier Flexible Retrieval	S3 Glacier Deep Archive
Designed for durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Designed for availability	99.99%	99.9%	99.9%	99.5%	99.9%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99%	99.9%	99.9%
Availability Zones	≥3	≥3	≥3	1	≥3	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128 KB	128 KB	128 KB	40 KB	40 KB
Minimum storage duration charge	N/A	N/A	30 days	30 days	90 days	90 days	180 days
Retrieval charge	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds	milliseconds	minutes or hours	hours
Storage type	Object	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes	Yes

via - <https://aws.amazon.com/s3/storage-classes/>

Incorrect options:

Amazon S3 Standard - Amazon S3 Standard offers high durability, availability, and performance object storage for frequently accessed data. As described above, Amazon S3 Standard-IA storage is a better fit than Amazon S3 Standard, hence using S3 standard is ruled out for the given use-case.

Amazon S3 Intelligent-Tiering (S3 Intelligent-Tiering) - For a small monthly object monitoring and automation charge, Amazon S3 Intelligent-Tiering monitors access patterns and automatically moves objects that have not been accessed to lower-cost access tiers. The Amazon S3 Intelligent-Tiering storage class is designed to optimize costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead. S3 Standard-IA matches the high durability, high throughput, and low latency of S3 Intelligent-Tiering, with a low per GB storage price and per GB retrieval fee. Moreover, Standard-IA has the same availability as that of Amazon S3 Intelligent-Tiering. So, it's cost-efficient to use S3 Standard-IA instead of S3 Intelligent-Tiering.

Amazon S3 Glacier Deep Archive - Amazon S3 Glacier Deep Archive is a secure, durable, and low-cost storage class for data archiving. Amazon S3 Glacier Deep Archive does not support millisecond latency, so this option is ruled out.

For more details on the durability, availability, cost and access latency - please review this reference link: <https://aws.amazon.com/s3/storage-classes>

Question 28 Incorrect

A media company runs a photo-sharing web application that is accessed across three different countries. The application is deployed on several Amazon Elastic Compute Cloud (Amazon EC2) instances running behind an Application Load Balancer. With new government regulations, the company has been asked to block access from two countries and allow access only from the home country of the company.

Which configuration should be used to meet this changed requirement?

Configure the security group for the Amazon EC2 instances

Correct answer

Configure AWS Web Application Firewall (AWS WAF) on the Application Load Balancer in a Amazon Virtual Private Cloud (Amazon VPC)

Use Geo Restriction feature of Amazon CloudFront in a Amazon Virtual Private Cloud (Amazon VPC)

Your answer is incorrect

Configure the security group on the Application Load Balancer

Overall explanation

Correct option:

AWS Web Application Firewall (AWS WAF) is a web application firewall service that lets you monitor web requests and protect your web applications from malicious requests. Use AWS WAF to block or allow requests based on conditions that you specify, such as the IP addresses. You can also use AWS WAF preconfigured protections to block common attacks like SQL injection or cross-site scripting.

Configure AWS Web Application Firewall (AWS WAF) on the Application Load Balancer in a Amazon Virtual Private Cloud (Amazon VPC)

You can use AWS WAF with your Application Load Balancer to allow or block requests based on the rules in a web access control list (web ACL). Geographic (Geo) Match Conditions in AWS WAF allows you to use AWS WAF to restrict application access based on the geographic location of your viewers. With geo match conditions you can choose the countries from which AWS WAF should allow access.

Geo match conditions are important for many customers. For example, legal and licensing requirements restrict some customers from delivering their applications outside certain countries. These customers can configure a whitelist that allows only viewers in those countries. Other customers need to prevent the downloading of their encrypted software by users in certain countries. These customers can configure a blacklist so that end-users from those countries are blocked from downloading their software.

Incorrect options:

Use Geo Restriction feature of Amazon CloudFront in a Amazon Virtual Private Cloud (Amazon VPC) - Geo Restriction feature of Amazon CloudFront helps in restricting traffic based on the user's geographic location. But, CloudFront works from edge locations and doesn't belong to a VPC. Hence, this option itself is incorrect and given only as a distractor.

Configure the security group on the Application Load Balancer

Configure the security group for the Amazon EC2 instances

Security Groups cannot restrict access based on the user's geographic location.

References:

<https://aws.amazon.com/about-aws/whats-new/2017/10/aws-waf-now-supports-geographic-match/>

<https://aws.amazon.com/blogs/aws/aws-web-application-firewall-waf-for-application-load-balancers/>

Domain

Design Secure Architectures

Question 29 Correct

A healthcare company uses its on-premises infrastructure to run legacy applications that require specialized customizations to the underlying Oracle database as well as its host operating system (OS). The company also wants to improve the availability of the Oracle database layer. The company has hired you as an AWS Certified Solutions Architect – Associate to build a solution on AWS that meets these requirements while minimizing the underlying infrastructure maintenance effort.

Which of the following options represents the best solution for this use case?

Your answer is correct

Leverage multi-AZ configuration of Amazon RDS Custom for Oracle that allows the Database Administrator (DBA) to access and customize the database environment and the underlying operating system

Leverage cross AZ read-replica configuration of Amazon RDS for Oracle that allows the Database Administrator (DBA) to access and customize the database environment and the underlying operating system

Deploy the Oracle database layer on multiple Amazon EC2 instances spread across two Availability Zones (AZs). This deployment configuration guarantees high availability and also allows the Database Administrator (DBA) to access and customize the database environment and the underlying operating system

Leverage multi-AZ configuration of Amazon RDS for Oracle that allows the Database Administrator (DBA) to access and customize the database environment and the underlying operating system

Overall explanation

Correct option:

Leverage multi-AZ configuration of Amazon RDS Custom for Oracle that allows the Database Administrator (DBA) to access and customize the database environment and the underlying operating system

Amazon RDS is a managed service that makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks. Amazon RDS can automatically back up your database and keep your database software up to date with the latest version. However, RDS does not allow you to access the host OS of the database.

For the given use-case, you need to use Amazon RDS Custom for Oracle as it allows you to access and customize your database server host and operating system, for example by applying special patches and changing the database software settings to support third-party applications that require privileged access. Amazon RDS Custom for Oracle facilitates these functionalities with minimum infrastructure maintenance effort. You need to set up the RDS Custom for Oracle in multi-AZ configuration for high availability.

Amazon RDS Custom for Oracle:

Amazon RDS Custom for Oracle – New Control Capabilities in Database Environment

by Channy Yun | on 26 OCT 2021 | in Amazon RDS, Database, Launch, News, RDS For Oracle | Permalink | [Share](#)

▶ 0:00 / 0:00



Voiced by [Amazon Polly](#)

Managing databases in self-managed environments such as on premises or Amazon Elastic Compute Cloud (Amazon EC2) requires customers to spend time and resources doing database administration tasks such as provisioning, scaling, patching, backups, and configuring for high availability. So, hundreds of thousands of AWS customers use [Amazon Relational Database Service](#) (Amazon RDS) because it automates these undifferentiated administration tasks.

However, there are some legacy and packaged applications that require customers to make specialized customizations to the underlying database and the operating system (OS), such as Oracle industry specialized applications for healthcare and life sciences, telecom, retail, banking, and hospitality. Customers with these specific customization requirements cannot get the benefits of a fully managed database service like Amazon RDS, and they end up deploying their databases on premises or on EC2 instances.

Today, I am happy to announce the general availability of [Amazon RDS Custom for Oracle](#), new capabilities that enable database administrators to access and customize the database environment and operating system. **With RDS Custom for Oracle, you can now access and customize your database server host and operating system, for example by applying special patches and changing the database software settings to support third-party applications that require privileged access.**

You can easily move your existing self-managed database for these applications to Amazon RDS and automate time-consuming database management tasks, such as software installation, patching, and backups. Here is a simple comparison of features and responsibilities between Amazon EC2, RDS Custom for Oracle, and RDS.

Features and Responsibilities	Amazon EC2	RDS Custom for Oracle	Amazon RDS
Application optimization	Customer	Customer	Customer
Scaling/high availability	Customer	Shared	AWS
DB backups	Customer	Shared	AWS
DB software maintenance	Customer	Shared	AWS
OS maintenance	Customer	Shared	AWS
Server maintenance	AWS	AWS	AWS

via - <https://aws.amazon.com/blogs/aws/amazon-rds-custom-for-oracle-new-control-capabilities-in-database-environment/>

Incorrect options:

Leverage multi-AZ configuration of Amazon RDS for Oracle that allows the Database Administrator (DBA) to access and customize the database environment and the underlying operating system

Leverage cross AZ read-replica configuration of Amazon RDS for Oracle that allows the Database Administrator (DBA) to access and customize the database environment and the underlying operating system

Amazon RDS for Oracle does not allow you to access and customize your database server host and operating system. Therefore, both these options are incorrect.

Deploy the Oracle database layer on multiple Amazon EC2 instances spread across two Availability Zones (AZs). This deployment configuration guarantees high availability and also allows the Database Administrator (DBA) to access and customize the database environment and the underlying operating system - The use case requires that the best solution should involve minimum infrastructure maintenance effort. When you use Amazon EC2 instances to host the databases, you need to manage the server health, server maintenance, server patching, and database maintenance tasks yourself. In addition, you will also need to manage the multi-AZ configuration by deploying Amazon EC2 instances across two Availability Zones (AZs), perhaps by using an Auto Scaling group. These steps entail significant maintenance effort. Hence this option is incorrect.

References:

<https://aws.amazon.com/blogs/aws/amazon-rds-custom-for-oracle-new-control-capabilities-in-database-environment/>

<https://aws.amazon.com/rds/faqs/>

Domain

Design Resilient Architectures

Question 30 Correct

A biotechnology firm runs genomics data analysis workloads using AWS Lambda functions deployed inside a VPC in their central AWS account. The input data for these workloads consists of large files stored in an Amazon Elastic File System (Amazon EFS) that resides in a separate AWS account managed by a research partner. The firm wants the Lambda function in their account to access the shared EFS storage directly. The access pattern and file volume are expected to grow as additional research datasets are added over time, so the solution must be scalable and cost-efficient, and should require minimal operational overhead.

Which solution best meets these requirements in the MOST cost-effective way?

Package the genomic input data as a Lambda layer and publish it in the research partner's account. Share the layer across accounts by modifying its resource policy and attach the layer to the Lambda function in the central account to access the data during execution

Your answer is correct

Use Amazon EFS resource policies to allow cross-account access to the file system from the central account. Attach the EFS mount target to a shared VPC or peered VPC, and mount the file system in the Lambda function configuration using an EFS access point

Set up an Amazon S3 bucket in the research partner's account and periodically copy EFS contents into the bucket using scheduled AWS DataSync jobs. Use Amazon S3 Access Points to expose the data to the Lambda function in the central account, allowing access via S3 API calls instead of file system mounts

Create a second Lambda function in the research partner's account that mounts the EFS file system locally. Have the main Lambda function in the central account invoke this secondary Lambda via Amazon API Gateway for data access and computation. Use IAM cross-account permissions to allow invocation

Overall explanation

Correct option:

Use Amazon EFS resource policies to allow cross-account access to the file system from the central account. Attach the EFS mount target to a shared VPC or peered VPC, and mount the file system in the Lambda function configuration using an EFS access point

This solution uses native AWS support for cross-account access to EFS via resource policies, combined with EFS access points and VPC-level networking (VPC peering or shared VPC). The Lambda function in the central account can mount the file system directly using the EFS mount target and access point as if it were in the same account. This method supports automatic scaling, secure access, and avoids any need for data duplication or proxy functions. It is the most cost-effective and operationally efficient solution.

Incorrect options:

Set up an Amazon S3 bucket in the research partner's account and periodically copy EFS contents into the bucket using scheduled AWS DataSync jobs. Use Amazon S3 Access Points to expose the data to the Lambda function in the central account, allowing access via S3 API calls instead of file system mounts - While this approach enables cross-account access to data using Amazon S3 Access Points, it introduces significant storage duplication and recurring transfer costs. EFS contents must be periodically copied to S3 using AWS DataSync, which adds operational complexity and delay. This solution also does not support real-time or low-latency file system operations, which may be required by the workloads.

Create a second Lambda function in the research partner's account that mounts the EFS file system locally. Have the main Lambda function in the central account invoke this secondary Lambda via Amazon API Gateway for data access and computation. Use IAM cross-account permissions to allow invocation - This option introduces additional latency and architectural complexity by involving a proxy Lambda function in a separate account, which acts as an intermediary for file access. It breaks the direct file system access model, limits performance, and requires managing a custom API layer. It also increases operational overhead, and any scaling of the secondary function will incur further compute cost and complexity.

Package the genomic input data as a Lambda layer and publish it in the research partner's account. Share the layer across accounts by modifying its resource policy and attach the layer to the Lambda function in the central account to access the data during execution - Lambda layers are designed for shared code and dependencies, not for large or dynamic datasets like genomic files. There are strict size limits (250 MB unzipped) on Lambda layers, and they are immutable once published. Sharing large or growing data sets via Lambda layers is infeasible and not scalable. This method is both cost-ineffective and technically unsuitable for the given use case.

References:

<https://docs.aws.amazon.com/efs/latest/ug/whatisefs.html>

<https://repost.aws/questions/QUCxMGeJpWRKKuvibwxWfP7Q/cross-account-efs-access-point-mount-on-lambda-with-root-path-as>

<https://docs.aws.amazon.com/efs/latest/ug/efs-access-points.html>

<https://docs.aws.amazon.com/datasync/latest/userguide/what-is-datasync.html>

<https://docs.aws.amazon.com/lambda/latest/dg/chapter-layers.html>

<https://docs.aws.amazon.com/lambda/latest/dg/adding-layers.html>

Domain

Design Cost-Optimized Architectures

Question 31 Incorrect

A media agency stores its re-creatable assets on Amazon Simple Storage Service (Amazon S3) buckets. The assets are accessed by a large number of users for the first few days and the frequency of access falls down drastically after a week. Although the assets would be accessed occasionally after the first week, but they must continue to be immediately accessible when required. The cost of maintaining all the assets on Amazon S3 storage is turning out to be very expensive and the agency is looking at reducing costs as much as possible.

As an AWS Certified Solutions Architect – Associate, can you suggest a way to lower the storage costs while fulfilling the business requirements?

Your answer is incorrect

Configure a lifecycle policy to transition the objects to Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA) after 7 days

Configure a lifecycle policy to transition the objects to Amazon S3 Standard-Infrequent Access (S3 Standard-IA) after 7 days

Correct answer

Configure a lifecycle policy to transition the objects to Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA) after 30 days

Configure a lifecycle policy to transition the objects to Amazon S3 Standard-Infrequent Access (S3 Standard-IA) after 30 days

Overall explanation

Correct option:

Configure a lifecycle policy to transition the objects to Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA) after 30 days

Amazon S3 One Zone-IA is for data that is accessed less frequently, but requires rapid access when needed. Unlike other S3 Storage Classes which store data in a minimum of three Availability Zones (AZs), Amazon S3 One Zone-IA stores data in a single Availability Zone (AZ) and costs 20% less than Amazon S3 Standard-IA. Amazon S3 One Zone-IA is ideal for customers who want a lower-cost option for infrequently accessed and re-creatable data but do not require the availability and resilience of Amazon S3 Standard or Amazon S3 Standard-IA. The minimum storage duration is 30 days before you can transition objects from Amazon S3 Standard to Amazon S3 One Zone-IA.

Amazon S3 One Zone-IA offers the same high durability, high throughput, and low latency of Amazon S3 Standard, with a low per GB storage price and per GB retrieval fee. S3 Storage Classes can be configured at the object level, and a single bucket can contain objects stored across Amazon S3 Standard, Amazon S3 Intelligent-Tiering, Amazon S3 Standard-IA, and Amazon S3 One Zone-IA. You can also use S3 Lifecycle policies to automatically transition objects between storage classes without any application changes.

Constraints for Lifecycle storage class transitions:

Constraints

Lifecycle storage class transitions have the following constraints:

Object size and transitions from S3 Standard or S3 Standard-IA to S3 Intelligent-Tiering, S3 Standard-IA, or S3 One Zone-IA

When you transition objects from the S3 Standard or S3 Standard-IA storage classes to S3 Intelligent-Tiering, S3 Standard-IA, or S3 One Zone-IA, the following object size constraints apply:

- **Larger objects** - For the following transitions, there is a cost benefit to transitioning larger objects:
 - From the S3 Standard or S3 Standard-IA storage classes to S3 Intelligent-Tiering.
 - From the S3 Standard storage class to S3 Standard-IA or S3 One Zone-IA.
- **Objects smaller than 128 KB** - For the following transitions, Amazon S3 does not transition objects that are smaller than 128 KB because it's not cost effective:
 - From the S3 Standard or S3 Standard-IA storage classes to S3 Intelligent-Tiering.
 - From the S3 Standard storage class to S3 Standard-IA or S3 One Zone-IA.

Minimum days for transition from S3 Standard or S3 Standard-IA to S3 Standard-IA or S3 One Zone-IA

Before you transition objects from the S3 Standard or S3 Standard-IA storage classes to S3 Standard-IA or S3 One Zone-IA, you must store them at least 30 days in the S3 Standard storage class. For example, you cannot create a Lifecycle rule to transition objects to the S3 Standard-IA storage class one day after you create them. Amazon S3 doesn't transition objects within the first 30 days because newer objects are often accessed more frequently or deleted sooner than is suitable for S3 Standard-IA or S3 One Zone-IA storage.

Similarly, if you are transitioning noncurrent objects (in versioned buckets), you can transition only objects that are at least 30 days noncurrent to S3 Standard-IA or S3 One Zone-IA storage.

Minimum 30-Day storage charge for S3 Intelligent-Tiering, S3 Standard-IA, and S3 One Zone-IA

The S3 Intelligent-Tiering, S3 Standard-IA, and S3 One Zone-IA storage classes have a minimum 30-day storage charge. Therefore, you can't specify a single Lifecycle rule for both an S3 Intelligent-Tiering, S3 Standard-IA, or S3 One Zone-IA transition and a S3 Glacier or S3 Glacier Deep Archive transition when the S3 Glacier or S3 Glacier Deep Archive transition occurs less than 30 days after the S3 Intelligent-Tiering, S3 Standard-IA, or S3 One Zone-IA transition.

The same 30-day minimum applies when you specify a transition from S3 Standard-IA storage to S3 One Zone-IA or S3 Intelligent-Tiering storage. You can specify two rules to accomplish this, but you pay minimum storage charges. For more information about cost considerations, see [Amazon S3 pricing](#).

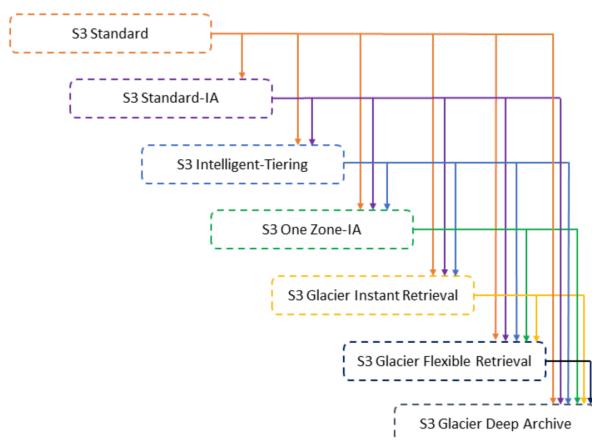
via - <https://docs.aws.amazon.com/AmazonS3/latest/dev/lifecycle-transition-general-considerations.html>

Supported Amazon S3 lifecycle transitions:

Supported transitions and related constraints

In an S3 Lifecycle configuration, you can define rules to transition objects from one storage class to another to save on storage costs. When you don't know the access patterns of your objects, or if your access patterns are changing over time, you can transition the objects to the S3 Intelligent-Tiering storage class for automatic cost savings. For information about storage classes, see [Using Amazon S3 storage classes](#).

Amazon S3 supports a waterfall model for transitioning between storage classes, as shown in the following diagram.



via - <https://docs.aws.amazon.com/AmazonS3/latest/dev/lifecycle-transition-general-considerations.html>

Incorrect options:

Configure a lifecycle policy to transition the objects to Amazon S3 Standard-Infrequent Access (S3 Standard-IA) after 7 days

Configure a lifecycle policy to transition the objects to Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA) after 7 days

As mentioned earlier, the minimum storage duration is 30 days before you can transition objects from Amazon S3 Standard to Amazon S3 One Zone-IA or Amazon S3 Standard-IA, so both these options are added as distractors.

Configure a lifecycle policy to transition the objects to Amazon S3 Standard-Infrequent Access (S3 Standard-IA) after 30 days - Amazon S3 Standard-IA is for data that is accessed less frequently, but requires rapid access when needed. S3 Standard-IA offers the high durability, high throughput, and low latency of Amazon S3 Standard, with a low per GB storage price and per GB retrieval fee. This combination of low cost and high performance makes Amazon S3 Standard-IA ideal for long-term storage, backups, and as a data store for disaster recovery files. But, it costs more than Amazon S3 One Zone-IA because of the redundant storage across Availability Zones (AZs). As the data is re-creatable, so you don't need to incur this additional cost.

References:

<https://aws.amazon.com/s3/storage-classes/>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/lifecycle-transition-general-considerations.html>

Domain

Design Cost-Optimized Architectures

Question 32 Correct

The engineering team at a data analytics company has observed that its flagship application functions at its peak performance when the underlying Amazon Elastic Compute Cloud (Amazon EC2) instances have a CPU utilization of about 50%. The application is built on a fleet of Amazon EC2 instances managed under an Auto Scaling group. The workflow requests are handled by an internal Application Load Balancer that routes the requests to the instances.

As a solutions architect, what would you recommend so that the application runs near its peak performance state?

Configure the Auto Scaling group to use step scaling policy and set the CPU utilization as the target metric with a target value of 50%

Configure the Auto Scaling group to use simple scaling policy and set the CPU utilization as the target metric with a target value of 50%

Configure the Auto Scaling group to use a Amazon Cloudwatch alarm triggered on a CPU utilization threshold of 50%

Your answer is correct

Configure the Auto Scaling group to use target tracking policy and set the CPU utilization as the target metric with a target value of 50%

Overall explanation

Correct option:

Configure the Auto Scaling group to use target tracking policy and set the CPU utilization as the target metric with a target value of 50%

An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also enables you to use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies.

With target tracking scaling policies, you select a scaling metric and set a target value. Amazon EC2 Auto Scaling creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value.

For example, you can use target tracking scaling to:

Configure a target tracking scaling policy to keep the average aggregate CPU utilization of your Auto Scaling group at 50 percent. This meets the requirements specified in the given use-case and therefore, this is the correct option.

Target Tracking Policy Overview:

Target Tracking Scaling Policies for Amazon EC2 Auto Scaling

[PDF](#) | [Kindle](#) | [RSS](#)

With target tracking scaling policies, you select a scaling metric and set a target value. Amazon EC2 Auto Scaling creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value. In addition to keeping the metric close to the target value, a target tracking scaling policy also adjusts to changes in the metric due to a changing load pattern.

For example, you can use target tracking scaling to:

- Configure a target tracking scaling policy to keep the average aggregate CPU utilization of your Auto Scaling group at 40 percent.
- Configure a target tracking scaling policy to keep the request count per target of your Application Load Balancer target group at 1000 for your Auto Scaling group.

Depending on your application needs, you might find that one of these metrics works best for you when using target tracking, or you might find that a combination of these metrics or a different metric meets your needs better.

via - <https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-target-tracking.html>

Incorrect options:

Configure the Auto Scaling group to use step scaling policy and set the CPU utilization as the target metric with a target value of 50%

Configure the Auto Scaling group to use simple scaling policy and set the CPU utilization as the target metric with a target value of 50%

With step scaling and simple scaling, you choose scaling metrics and threshold values for the Amazon CloudWatch alarms that trigger the scaling process. Neither step scaling nor simple scaling can be configured to use a target metric for CPU utilization, hence both these options are incorrect.

Configure the Auto Scaling group to use a Amazon Cloudwatch alarm triggered on a CPU utilization threshold of 50% - An Auto Scaling group cannot directly use a Cloudwatch alarm as the source for a scale-in or scale-out event, hence this option is incorrect.

References:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroup.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-target-tracking.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-simple-step.html>

Domain

Design High-Performing Architectures

Question 33 Incorrect

A major bank is using Amazon Simple Queue Service (Amazon SQS) to migrate several core banking applications to the cloud to ensure high availability and cost efficiency while simplifying administrative complexity and overhead. The development team at the bank expects a peak rate of about 1000 messages per second to be processed via SQS. It is important that the messages are processed in order.

Which of the following options can be used to implement this system?

Use Amazon SQS standard queue to process the messages

Your answer is incorrect

Use Amazon SQS FIFO (First-In-First-Out) queue to process the messages

Use Amazon SQS FIFO (First-In-First-Out) queue in batch mode of 2 messages per operation to process the messages at the peak rate

Correct answer

Use Amazon SQS FIFO (First-In-First-Out) queue in batch mode of 4 messages per operation to process the messages at the peak rate

Overall explanation

Correct option:

Use Amazon SQS FIFO (First-In-First-Out) queue in batch mode of 4 messages per operation to process the messages at the peak rate

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS offers two types of message queues - Standard queues vs FIFO queues.

For FIFO queues, the order in which messages are sent and received is strictly preserved (i.e. First-In-First-Out). On the other hand, the standard SQS queues offer best-effort ordering. This means that occasionally, messages might be delivered in an order different from which they were sent.

By default, FIFO queues support up to 300 messages per second (300 send, receive, or delete operations per second). When you batch 10 messages per operation (maximum),

FIFO queues can support up to 3,000 messages per second. Therefore you need to process 4 messages per operation so that the FIFO queue can support up to 1200 messages per second, which is well within the peak rate.

FIFO Queues Overview:

Amazon SQS queue types

[PDF](#) | [RSS](#)

Amazon SQS supports two types of queues – **standard queues** and **FIFO queues**. Use the information from the following table to choose the right queue for your situation. To learn more about Amazon SQS queues, see [Getting started with Amazon SQS standard queues](#) and [Getting started with Amazon SQS FIFO queues](#).

Standard queues	FIFO queues
<p>Unlimited Throughput – Standard queues support a nearly unlimited number of API calls per second, per API action (<code>SendMessage</code>, <code>ReceiveMessage</code>, or <code>DeleteMessage</code>).</p> <p>At-Least-Once Delivery – A message is delivered at least once, but occasionally more than one copy of a message is delivered.</p> <p>Best-Effort Ordering – Occasionally, messages are delivered in an order different from which they were sent.</p>  <p></p>	<p>High Throughput – If you use batching, FIFO queues support up to 3,000 messages per second, per API method (<code>SendMessageBatch</code>, <code>ReceiveMessage</code>, or <code>DeleteMessageBatch</code>). The 3,000 messages per second represent 300 API calls, each with a batch of 10 messages. To request a quota increase, submit a support request. Without batching, FIFO queues support up to 300 API calls per second, per API method (<code>SendMessage</code>, <code>ReceiveMessage</code>, or <code>DeleteMessage</code>).</p> <p>Exactly-Once Processing – A message is delivered once and remains available until a consumer processes and deletes it. Duplicates aren't introduced into the queue.</p>  <p></p>
<p>Send data between applications when the throughput is important, for example:</p> <ul style="list-style-type: none">Decouple live user requests from intensive background work: let users upload media while resizing or encoding it.Allocate tasks to multiple worker nodes: process a high number of credit card validation requests.Batch messages for future processing: schedule multiple entries to be added to a database.	<p>Send data between applications when the order of events is important, for example:</p> <ul style="list-style-type: none">Make sure that user-entered commands are run in the right order.Display the correct product price by sending price modifications in the right order.Prevent a student from enrolling in a course before registering for an account.

via -

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/FIFO-queues.html>

Incorrect options:

Use Amazon SQS standard queue to process the messages - As messages need to be processed in order, therefore standard queues are ruled out.

Use Amazon SQS FIFO (First-In-First-Out) queue to process the messages - By default, FIFO queues support up to 300 messages per second and this is not sufficient to meet the message processing throughput per the given use-case. Hence this option is incorrect.

Use Amazon SQS FIFO (First-In-First-Out) queue in batch mode of 2 messages per operation to process the messages at the peak rate - As mentioned earlier in the explanation, you need to use FIFO queues in batch mode and process 4 messages per operation, so that the FIFO queue can support up to 1200 messages per second. With 2 messages per operation, you can only support up to 600 messages per second.

References:

<https://aws.amazon.com/sqs/>

<https://aws.amazon.com/sqs/features/>

Domain

Design Resilient Architectures

Question 34 Incorrect

The DevOps team at an e-commerce company wants to perform some maintenance work on a specific Amazon EC2 instance that is part of an Auto Scaling group using a step scaling policy. The team is facing a maintenance challenge - every time the team deploys a maintenance patch, the instance health check status shows as out of service for a few minutes. This causes the Auto Scaling group to provision another replacement instance immediately.

As a solutions architect, which are the MOST time/resource efficient steps that you would recommend so that the maintenance work can be completed at the earliest? (Select two)

Take a snapshot of the instance, create a new Amazon Machine Image (AMI) and then launch a new instance using this AMI. Apply the maintenance patch to this new instance and then add it back to the Auto Scaling Group by using the manual scaling policy. Terminate the earlier instance that had the maintenance issue

Suspend the `ScheduledActions` process type for the Auto Scaling group and apply the maintenance patch to the instance. Once the instance is ready, you can manually set the instance's health status back to healthy and activate the `ScheduledActions` process type again

Correct selection

Suspend the `ReplaceUnhealthy` process type for the Auto Scaling group and apply the maintenance patch to the instance. Once the instance is ready, you can manually set the instance's health status back to healthy and activate the `ReplaceUnhealthy` process type again

Your selection is correct

Put the instance into the Standby state and then update the instance by applying the maintenance patch. Once the instance is ready, you can exit the Standby state and then return the instance to service

Delete the Auto Scaling group and apply the maintenance fix to the given instance. Create a new Auto Scaling group and add all the instances again using the manual scaling policy

Overall explanation

Correct options:

Put the instance into the Standby state and then update the instance by applying the maintenance patch. Once the instance is ready, you can exit the Standby state and then return the instance to service - You can put an instance that is in the `InService` state into the `Standby` state, update some software or troubleshoot the instance, and then return the instance to service. Instances that are on standby are still part of the Auto Scaling group, but they do not actively handle application traffic.

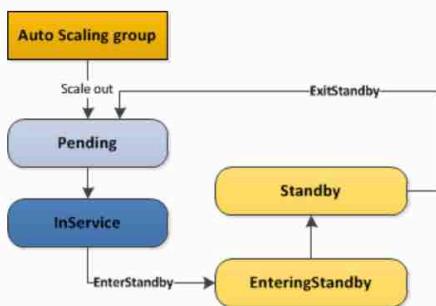
How Standby State Works:

How the Standby State Works

The standby state works as follows to help you temporarily remove an instance from your Auto Scaling group:

1. You put the instance into the standby state. The instance remains in this state until you exit the standby state.
2. If there is a load balancer or target group attached to your Auto Scaling group, the instance is deregistered from the load balancer or target group.
3. By default, the value that you specified as your desired capacity is decremented when you put an instance on standby. This prevents the launch of an additional instance while you have this instance on standby. Alternatively, you can specify that your desired capacity is not decremented. If you specify this option, the Auto Scaling group launches an instance to replace the one on standby. The intention is to help you maintain capacity for your application while one or more instances are on standby.
4. You can update or troubleshoot the instance.
5. You return the instance to service by exiting the standby state.
6. After you put an instance that was on standby back in service, the desired capacity is incremented. If you did not decrement the capacity when you put the instance on standby, the Auto Scaling group detects that you have more instances than you need. It applies the termination policy in effect to reduce the size of the group. For more information, see [Controlling Which Auto Scaling Instances Terminate During Scale In](#).
7. If there is a load balancer or target group attached to your Auto Scaling group, the instance is registered with the load balancer or target group.

The following illustration shows the transitions between instance states in this process:



via - <https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-enter-exit-standby.html>

Suspend the `ReplaceUnhealthy` process type for the Auto Scaling group and apply the maintenance patch to the instance. Once the instance is ready, you can manually set the instance's health status back to healthy and activate the `ReplaceUnhealthy` process type again - The `ReplaceUnhealthy` process terminates instances that are marked as unhealthy and then creates new instances to replace them. Amazon EC2 Auto Scaling stops replacing instances that are marked as unhealthy. Instances that fail EC2 or Elastic Load Balancing health checks are still marked as unhealthy. As soon as you resume the `ReplaceUnhealthy` process, Amazon EC2 Auto Scaling replaces instances that were marked unhealthy while this process was suspended.

Incorrect options:

Take a snapshot of the instance, create a new Amazon Machine Image (AMI) and then launch a new instance using this AMI. Apply the maintenance patch to this new instance and then add it back to the Auto Scaling Group by using the manual scaling policy. Terminate the earlier instance that had the maintenance issue - Taking the snapshot of the existing instance to create a new AMI and then creating a new instance in order to apply the maintenance patch is not time/resource optimal, hence this option is ruled out.

Delete the Auto Scaling group and apply the maintenance fix to the given instance. Create a new Auto Scaling group and add all the instances again using the manual scaling policy - It's not recommended to delete the Auto Scaling group just to apply a maintenance patch on a specific instance.

Suspend the `ScheduledActions` process type for the Auto Scaling group and apply the maintenance patch to the instance. Once the instance is ready, you can manually set the instance's health status back to healthy and activate the `ScheduledActions` process type again - Amazon EC2 Auto Scaling does not execute scaling actions that are scheduled to run during the suspension period. This option is not relevant to the given use-case.

References:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-enter-exit-standby.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-suspend-resume-processes.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/health-checks-overview.html>

Domain

Design Resilient Architectures

Question 35 Correct

A digital wallet company plans to launch a new cloud-based service for processing user cash transfers and peer-to-peer payments. The application will receive transaction requests from mobile clients via a secure endpoint. Each transaction request must go through a lightweight validation step before being forwarded for backend processing, which includes fraud detection, ledger updates, and notifications. The backend workload is compute- and memory-intensive, requires scaling based on volume, and must run for a longer duration than typical short-lived tasks. The engineering team prefers a fully managed solution that minimizes infrastructure maintenance, including provisioning and patching of virtual machines or containers.

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

Build a REST API using Amazon API Gateway. Integrate it with an AWS Step Functions state machine for validation. Launch the backend application using Amazon EKS with self-managed nodes, and use

Kubernetes Jobs to handle transaction processing workflows. Manually scale the cluster based on demand

Create an Amazon API Gateway endpoint to receive transaction requests from mobile devices. Use AWS Lambda to validate the transactions. For backend processing, deploy the application on Amazon EKS Anywhere, running on on-premises servers in the company's data center. Use a custom provisioning script to scale Kubernetes worker nodes based on transaction volume

Configure Amazon SQS to receive encrypted payment notifications from mobile devices. Use Amazon EventBridge rules to extract the payload and perform validation. Route the messages to a backend system hosted on Amazon Lightsail instances with dynamic scaling policies based on memory thresholds and instance health checks

Your answer is correct

Expose an Amazon API Gateway REST API endpoint to receive transaction requests from mobile clients. Integrate the API with AWS Lambda to perform basic validation. For backend processing, deploy the long-running application to Amazon ECS using the Fargate launch type, allowing ECS to manage compute and memory provisioning automatically, with no server management required

Overall explanation

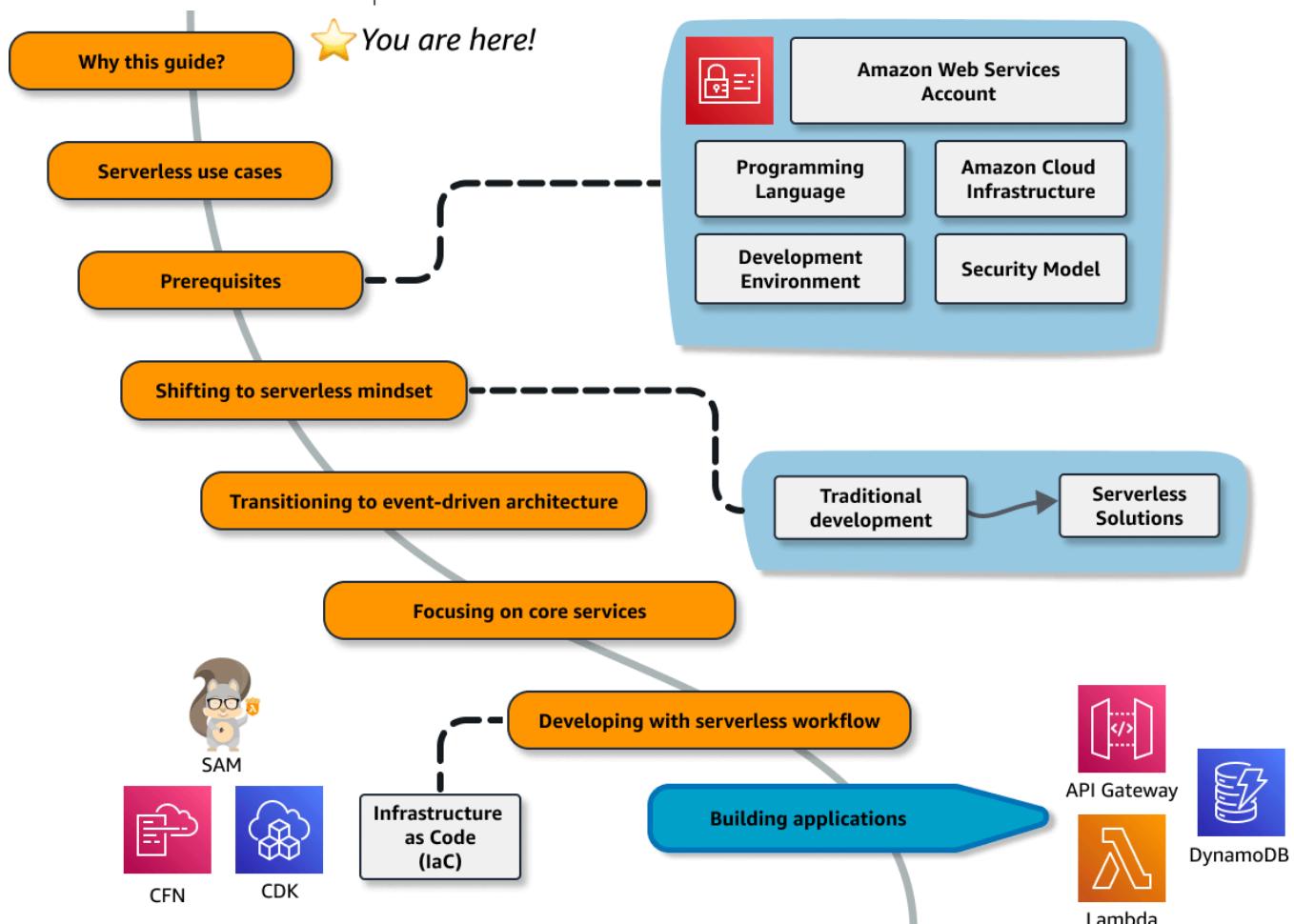
Correct option:

Expose an Amazon API Gateway REST API endpoint to receive transaction requests from mobile clients. Integrate the API with AWS Lambda to perform basic validation.

For backend processing, deploy the long-running application to Amazon ECS using the Fargate launch type, allowing ECS to manage compute and memory provisioning automatically, with no server management required

This architecture uses Amazon API Gateway to securely expose a RESTful endpoint, which integrates seamlessly with AWS Lambda for the lightweight validation logic. Once validated, the data is passed to Amazon ECS running on Fargate, a serverless container orchestration service that automatically provisions compute and memory resources based on task definitions. ECS with Fargate eliminates the need for cluster management, OS patching, or infrastructure provisioning. This makes the solution fully managed, highly scalable, and the most operationally efficient for long-running backend tasks.

What is serverless development?



via - <https://docs.aws.amazon.com/serverless/latest/devguide/welcome.html>

Incorrect options:

Build a REST API using Amazon API Gateway. Integrate it with an AWS Step Functions state machine for validation. Launch the backend application using Amazon EKS with self-managed nodes, and use Kubernetes Jobs to handle transaction processing workflows. Manually scale the cluster based on demand - Although Step Functions can manage workflows and EKS supports containerized apps, EKS with self-managed nodes introduces infrastructure complexity. You must provision and manage the worker nodes, monitor Kubernetes components, and maintain security patches. This goes against the

company's requirement to avoid managing infrastructure. Step Functions is also less ideal for handling synchronous validation of lightweight logic, where Lambda is more cost-effective.

Configure Amazon SQS to receive encrypted payment notifications from mobile devices. Use Amazon EventBridge rules to extract the payload and perform validation. Route the messages to a backend system hosted on Amazon Lightsail instances with dynamic scaling policies based on memory thresholds and instance health checks - This approach combines messaging and routing but uses Amazon Lightsail, which is designed for simplified hosting of websites and smaller applications. While Lightsail supports scaling policies, it lacks the advanced orchestration, scalability, and managed features needed for enterprise-grade backend processing. Additionally, using EventBridge for validation is not correct — it's a routing service, not a transformation or logic execution engine.

Create an Amazon API Gateway endpoint to receive transaction requests from mobile devices. Use AWS Lambda to validate the transactions. For backend processing, deploy the application on Amazon EKS Anywhere, running on on-premises servers in the company's data center. Use a custom provisioning script to scale Kubernetes worker nodes based on transaction volume - This solution introduces unnecessary infrastructure management, which contradicts the company's requirement to avoid managing infrastructure. Amazon EKS Anywhere is intended for organizations that need to run Kubernetes on-premises, often for latency, compliance, or hybrid cloud scenarios. However, it requires you to manage the entire infrastructure stack, including hardware, networking, OS patches, Kubernetes upgrades, and worker node scaling. That adds significant operational overhead, making it unsuitable for a team prioritizing low-maintenance, cloud-native solutions.

References:

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/AWS_Fargate.html

<https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html>

<https://docs.aws.amazon.com/serverless/latest/devguide/welcome.html>

<https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-set-up-simple-proxy.html>

<https://aws.amazon.com/eks/eks-anywhere/>

Domain

Question 36 Correct

While consolidating logs for the weekly reporting, a development team at an e-commerce company noticed that an unusually large number of illegal AWS application programming interface (API) queries were made sometime during the week. Due to the off-season, there was no visible impact on the systems. However, this event led the management team to seek an automated solution that can trigger near-real-time warnings in case such an event recurs.

Which of the following represents the best solution for the given scenario?

AWS Trusted Advisor publishes metrics about check results to Amazon CloudWatch. Create an alarm to track status changes for checks in the Service Limits category for the APIs. The alarm will then notify when the service quota is reached or exceeded

Run Amazon Athena SQL queries against AWS CloudTrail log files stored in Amazon S3 buckets. Use Amazon QuickSight to generate reports for managerial dashboards

Your answer is correct

Create an Amazon CloudWatch metric filter that processes AWS CloudTrail logs having API call details and looks at any errors by factoring in all the error codes that need to be tracked. Create an alarm based on this metric's rate to send an Amazon SNS notification to the required team

Configure AWS CloudTrail to stream event data to Amazon Kinesis. Use Amazon Kinesis stream-level metrics in the Amazon CloudWatch to trigger an AWS Lambda function that will trigger an error workflow

Overall explanation

Correct option:

Create an Amazon CloudWatch metric filter that processes AWS CloudTrail logs having API call details and looks at any errors by factoring in all the error codes that need to be tracked. Create an alarm based on this metric's rate to send an Amazon SNS notification to the required team

AWS CloudTrail log data can be ingested into Amazon CloudWatch to monitor and identify your AWS account activity against security threats, and create a governance framework for security best practices. You can analyze log trail event data in CloudWatch using features such as Logs Insight, Contributor Insights, Metric filters, and CloudWatch Alarms.

AWS CloudTrail integrates with the Amazon CloudWatch service to publish the API calls being made to resources or services in the AWS account. The published event has invaluable information that can be used for compliance, auditing, and governance of your AWS accounts. Below we introduce several features available in CloudWatch to monitor API activity, analyze the logs at scale, and take action when malicious activity is discovered, without provisioning your infrastructure.

For the AWS Cloudtrail logs available in Amazon CloudWatch Logs, you can begin searching and filtering the log data by creating one or more metric filters. Use these metric filters to turn log data into numerical CloudWatch metrics that you can graph or set a CloudWatch Alarm on.

Note: AWS CloudTrail Insights helps AWS users identify and respond to unusual activity associated with **write** API calls by continuously analyzing CloudTrail management events.

Insights events are logged when AWS CloudTrail detects unusual **write** management API activity in your account. If you have AWS CloudTrail Insights enabled and CloudTrail detects unusual activity, Insights events are delivered to the destination Amazon S3 bucket for your trail. You can also see the type of insight and the incident time when you view Insights events on the CloudTrail console. Unlike other types of events captured in a CloudTrail trail, Insights events are logged only when CloudTrail detects changes in your account's API usage that differ significantly from the account's typical usage patterns.

Incorrect options:

Configure AWS CloudTrail to stream event data to Amazon Kinesis. Use Amazon Kinesis stream-level metrics in the Amazon CloudWatch to trigger an AWS Lambda function that will trigger an error workflow - AWS CloudTrail cannot stream data to Amazon Kinesis. Amazon S3 buckets and Amazon CloudWatch logs are the only destinations possible.

Run Amazon Athena SQL queries against AWS CloudTrail log files stored in Amazon S3 buckets. Use Amazon QuickSight to generate reports for managerial dashboards - Generating reports and visualizations help in understanding and analyzing patterns but is not useful as a near-real-time automatic solution for the given problem.

AWS Trusted Advisor publishes metrics about check results to Amazon CloudWatch. Create an alarm to track status changes for checks in the Service Limits category for the APIs. The alarm will then notify when the service quota is reached or exceeded - When AWS Trusted Advisor refreshes your checks, Trusted Advisor publishes metrics about your check results to Amazon CloudWatch. You can view the metrics in CloudWatch. You can also create alarms to detect status changes to Trusted Advisor checks and status changes for resources, and service quota usage (formerly referred to as limits). The alarm will then notify you when you reach or exceed a service quota for your AWS account. However, the alarm is triggered only when the service limit is reached. We need a solution that raises an alarm when the number of API calls randomly increases or an abnormal pattern is detected. Hence, this option is not the right fit for the given use case.

References:

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html#cloudwatch-alarms-for-cloudtrail-authorization-failures>

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-concepts.html>

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/logging-insights-events-with-cloudtrail.html>

<https://docs.aws.amazon.com/awssupport/latest/user/cloudwatch-metrics-ta.html>

Domain

Design Resilient Architectures

Question 37 Correct

A news network uses Amazon Simple Storage Service (Amazon S3) to aggregate the raw video footage from its reporting teams across the US. The news network has recently expanded into new geographies in Europe and Asia. The technical teams at the overseas branch offices have reported huge delays in uploading large video files to the destination Amazon S3 bucket.

Which of the following are the MOST cost-effective options to improve the file upload speed into Amazon S3 (Select two)

Create multiple AWS Direct Connect connections between the AWS Cloud and branch offices in Europe and Asia. Use the direct connect connections for faster file uploads into Amazon S3

Your selection is correct

Use multipart uploads for faster file uploads into the destination Amazon S3 bucket

Use AWS Global Accelerator for faster file uploads into the destination Amazon S3 bucket

Create multiple AWS Site-to-Site VPN connections between the AWS Cloud and branch offices in Europe and Asia. Use these VPN connections for faster file uploads into Amazon S3

Your selection is correct

Use Amazon S3 Transfer Acceleration (Amazon S3TA) to enable faster file uploads into the destination S3 bucket

Overall explanation

Correct options:

Use Amazon S3 Transfer Acceleration (Amazon S3TA) to enable faster file uploads into the destination S3 bucket

Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. Amazon S3TA takes advantage of Amazon CloudFront's globally distributed edge locations. As the data arrives at an edge location, data is routed to Amazon S3 over an optimized network path.

Use multipart uploads for faster file uploads into the destination Amazon S3 bucket

Multipart upload allows you to upload a single object as a set of parts. Each part is a contiguous portion of the object's data. You can upload these object parts independently and in any order. If transmission of any part fails, you can retransmit that part without affecting other parts. After all parts of your object are uploaded, Amazon S3 assembles these parts and creates the object. In general, when your object size reaches 100 MB, you should consider using multipart uploads instead of uploading the object in a single operation. Multipart upload provides improved throughput, therefore it facilitates faster file uploads.

Incorrect options:

Create multiple AWS Direct Connect connections between the AWS Cloud and branch offices in Europe and Asia. Use the direct connect connections for faster file uploads into Amazon S3 - AWS Direct Connect is a cloud service solution that makes it easy to establish a dedicated network connection from your premises to AWS. AWS Direct Connect lets you establish a dedicated network connection between your network and one of the AWS Direct Connect locations. Direct connect takes significant time (several months) to be provisioned and is an overkill for the given use-case.

Create multiple AWS Site-to-Site VPN connections between the AWS Cloud and branch offices in Europe and Asia. Use these VPN connections for faster file uploads into Amazon S3 - AWS Site-to-Site VPN enables you to securely connect your on-premises network or branch office site to your Amazon Virtual Private Cloud (Amazon VPC). You can securely extend your data center or branch office network to the cloud with an AWS Site-to-Site VPN connection. A VPC VPN Connection utilizes IPSec to establish encrypted network connectivity between your intranet and Amazon VPC over the Internet. VPN Connections are a good solution if you have low to modest bandwidth requirements and can tolerate the inherent variability in Internet-based connectivity. Site-to-site VPN will not help in accelerating the file transfer speeds into S3 for the given use-case.

Use AWS Global Accelerator for faster file uploads into the destination Amazon S3 bucket - AWS Global Accelerator is a service that improves the availability and performance of your applications with local or global users. It provides static IP addresses

that act as a fixed entry point to your application endpoints in a single or multiple AWS Regions, such as your Application Load Balancers, Network Load Balancers or Amazon EC2 instances. AWS Global Accelerator will not help in accelerating the file transfer speeds into S3 for the given use-case.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/transfer-acceleration.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/uploadobjusingmpu.html>

Domain

Design Cost-Optimized Architectures

Question 38 Correct

A leading carmaker would like to build a new car-as-a-sensor service by leveraging fully serverless components that are provisioned and managed automatically by AWS. The development team at the carmaker does not want an option that requires the capacity to be manually provisioned, as it does not want to respond manually to changing volumes of sensor data.

Given these constraints, which of the following solutions is the BEST fit to develop this car-as-a-sensor service?

Ingest the sensor data in an Amazon Simple Queue Service (Amazon SQS) standard queue, which is polled by an application running on an Amazon EC2 instance and the data is written into an auto-scaled Amazon DynamoDB table for downstream processing

Ingest the sensor data in Amazon Kinesis Data Streams, which is polled by an application running on an Amazon EC2 instance and the data is written into an auto-scaled Amazon DynamoDB table for downstream processing

Your answer is correct

Ingest the sensor data in an Amazon Simple Queue Service (Amazon SQS) standard queue, which is polled by an AWS Lambda function in batches and the data is written into an auto-scaled Amazon DynamoDB table for downstream processing

Ingest the sensor data in Amazon Kinesis Data Firehose, which directly writes the data into an auto-scaled Amazon DynamoDB table for downstream processing

Overall explanation

Correct option:

Ingest the sensor data in an Amazon Simple Queue Service (Amazon SQS) standard queue, which is polled by an AWS Lambda function in batches and the data is written into an auto-scaled Amazon DynamoDB table for downstream processing

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume. Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

AWS manages all ongoing operations and underlying infrastructure needed to provide a highly available and scalable message queuing service. With SQS, there is no upfront cost, no need to acquire, install, and configure messaging software, and no time-consuming build-out and maintenance of supporting infrastructure. SQS queues are dynamically created and scale automatically so you can build and grow applications quickly and efficiently.

As there is no need to manually provision the capacity, so this is the correct option.

Incorrect options:

Ingest the sensor data in Amazon Kinesis Data Firehose, which directly writes the data into an auto-scaled Amazon DynamoDB table for downstream processing -Amazon Kinesis Data Firehose is a fully managed service for delivering real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon OpenSearch Service, Splunk, and any custom HTTP endpoint or HTTP endpoints owned by supported third-party service providers, including Datadog, Dynatrace, LogicMonitor, MongoDB, New Relic, and Sumo Logic.

Firehose cannot directly write into a DynamoDB table, so this option is incorrect.

Ingest the sensor data in an Amazon Simple Queue Service (Amazon SQS) standard queue, which is polled by an application running on an Amazon EC2 instance and the data is written into an auto-scaled Amazon DynamoDB table for downstream processing

Ingest the sensor data in Amazon Kinesis Data Streams, which is polled by an application running on an Amazon EC2 instance and the data is written into an auto-scaled Amazon DynamoDB table for downstream processing

Using an application on an Amazon EC2 instance is ruled out as the carmaker wants to use fully serverless components. So both these options are incorrect.

References:

<https://aws.amazon.com/sqs/>

<https://docs.aws.amazon.com/lambda/latest/dg/with-kinesis.html>

<https://docs.aws.amazon.com/lambda/latest/dg/with-sqs.html>

<https://aws.amazon.com/kinesis/data-streams/faqs/>

Domain

Design Resilient Architectures

Question 39 Incorrect

A company has moved its business critical data to Amazon Elastic File System (Amazon EFS) which will be accessed by multiple Amazon EC2 instances.

As an AWS Certified Solutions Architect - Associate, which of the following would you recommend to exercise access control such that only the permitted Amazon EC2 instances can read from the Amazon EFS file system? (Select two)

Your selection is correct

Use an IAM policy to control access for clients who can mount your file system with the required permissions

Your selection is incorrect

Use Amazon GuardDuty to curb unwanted access to Amazon EFS file system

Set up the IAM policy root credentials to control and configure the clients accessing the Amazon EFS file system

Use network access control list (network ACL) to control the network traffic to and from your Amazon EC2 instance

Your selection is correct

Use VPC security groups to control the network traffic to and from your file system

Overall explanation

Correct options:

Use VPC security groups to control the network traffic to and from your file system

Use an IAM policy to control access for clients who can mount your file system with the required permissions

You control which Amazon EC2 instances can access your Amazon EFS file system by using VPC security group rules and AWS Identity and Access Management (IAM) policies. Use VPC security groups to control the network traffic to and from your file system. Attach an IAM policy to your file system to control which clients can mount your file system and with what permissions, and you may use Amazon EFS Access Points to manage application access. Control access to files and directories with POSIX-compliant user and group-level permissions.

Files and directories in an Amazon EFS file system support standard Unix-style read, write, and execute permissions based on the user ID and group IDs. When an NFS client mounts an Amazon EFS file system without using an access point, the user ID and group ID provided by the client is trusted. You can also use Amazon EFS access points to override user ID and group IDs used by the NFS client. When users attempt to access files and directories, Amazon EFS checks their user IDs and group IDs to verify that each user has permission to access the objects.

Incorrect options:

Use network access control list (network ACL) to control the network traffic to and from your Amazon EC2 instance - Network ACLs operate at the subnet level and not at the instance level.

Set up the IAM policy root credentials to control and configure the clients accessing the Amazon EFS file system - There is no such thing as an IAM policy root credentials and this statement has been added as a distractor.

Use Amazon GuardDuty to curb unwanted access to Amazon EFS file system - Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts, workloads, and data stored in Amazon S3. It cannot be used for access control to the Amazon EFS file system.

References:

https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Security.html#VPC_Security_Comparison

<https://docs.aws.amazon.com/efs/latest/ug/accessing-fs-nfs-permissions.html>

<https://docs.aws.amazon.com/efs/latest/ug/iam-access-control-nfs-efs.html>

Domain

Design Secure Architectures

Question 40 Correct

The IT department at a consulting firm is conducting a training workshop for new developers. As part of an evaluation exercise on Amazon S3, the new developers were asked to identify the invalid storage class lifecycle transitions for objects stored on Amazon S3.

Can you spot the INVALID lifecycle transitions from the options below? (Select two)

Amazon S3 Standard-IA => Amazon S3 One Zone-IA

Amazon S3 Standard-IA => Amazon S3 Intelligent-Tiering

Amazon S3 Standard => Amazon S3 Intelligent-Tiering

Your selection is correct

Amazon S3 One Zone-IA => Amazon S3 Standard-IA

Your selection is correct

Amazon S3 Intelligent-Tiering => Amazon S3 Standard

Overall explanation

Correct options:

As the question wants to know about the INVALID lifecycle transitions, the following options are the correct answers -

Amazon S3 Intelligent-Tiering => Amazon S3 Standard

Amazon S3 One Zone-IA => Amazon S3 Standard-IA

Following are the unsupported life cycle transitions for S3 storage classes - Any storage class to the Amazon S3 Standard storage class. Any storage class to the Reduced Redundancy storage class. The Amazon S3 Intelligent-Tiering storage class to the Amazon S3 Standard-IA storage class. The Amazon S3 One Zone-IA storage class to the Amazon S3 Standard-IA or Amazon S3 Intelligent-Tiering storage classes.

Incorrect options:

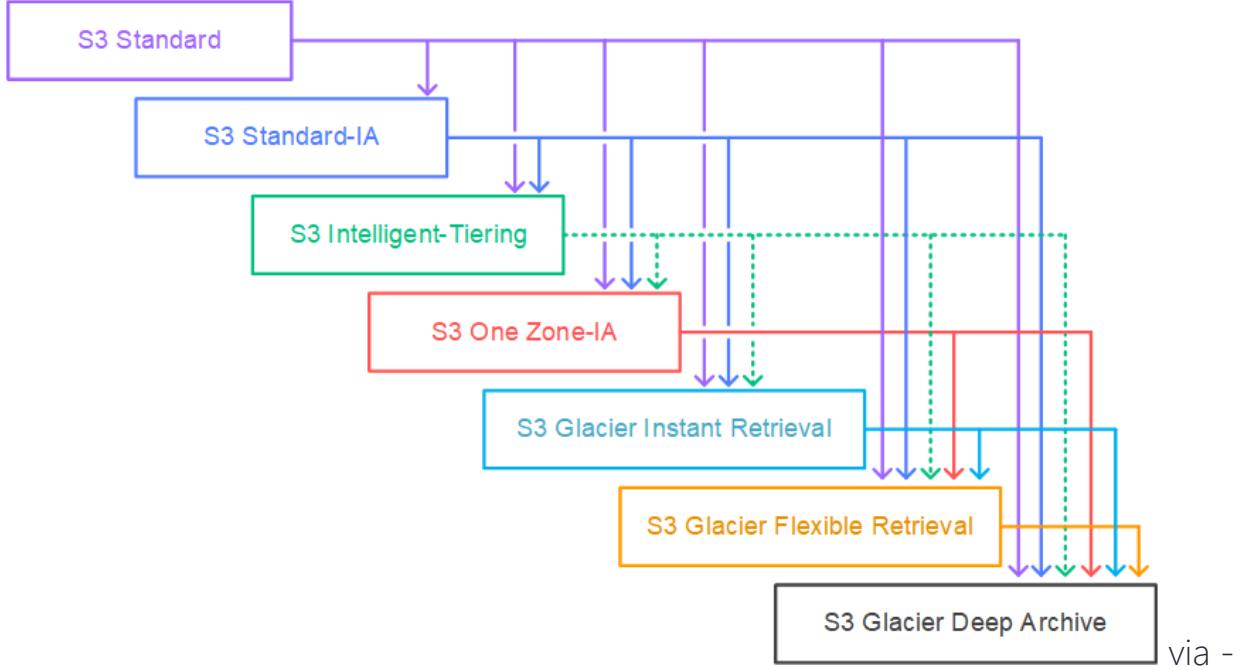
Amazon S3 Standard => Amazon S3 Intelligent-Tiering

Amazon S3 Standard-IA => Amazon S3 Intelligent-Tiering

Amazon S3 Standard-IA => Amazon S3 One Zone-IA

Here are the supported life cycle transitions for S3 storage classes - The S3 Standard storage class to any other storage class. Any storage class to the S3 Glacier or S3 Glacier Deep Archive storage classes. The S3 Standard-IA storage class to the S3 Intelligent-Tiering or S3 One Zone-IA storage classes. The S3 Intelligent-Tiering storage class to the S3 One Zone-IA storage class. The S3 Glacier storage class to the S3 Glacier Deep Archive storage class.

Amazon S3 supports a waterfall model for transitioning between storage classes, as shown in the diagram below:



<https://docs.aws.amazon.com/AmazonS3/latest/userguide/lifecycle-transition-general-considerations.html>

Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/lifecycle-transition-general-considerations.html>

Domain

Design Cost-Optimized Architectures

Question 41 Incorrect

The engineering team at an in-home fitness company is evaluating multiple in-memory data stores with the ability to power its on-demand, live leaderboard. The company's leaderboard requires high availability, low latency, and real-time processing to deliver customizable user data for the community of users working out together virtually from the comfort of their home.

As a solutions architect, which of the following solutions would you recommend? (Select two)

Power the on-demand, live leaderboard using Amazon RDS for Aurora as it meets the in-memory, high availability, low latency requirements

Correct selection

Power the on-demand, live leaderboard using Amazon DynamoDB with DynamoDB Accelerator (DAX) as it meets the in-memory, high availability, low latency requirements

Your selection is correct

Power the on-demand, live leaderboard using Amazon ElastiCache for Redis as it meets the in-memory, high availability, low latency requirements

Power the on-demand, live leaderboard using Amazon Neptune as it meets the in-memory, high availability, low latency requirements

Power the on-demand, live leaderboard using Amazon DynamoDB as it meets the in-memory, high availability, low latency requirements

Overall explanation

Correct options:

Power the on-demand, live leaderboard using Amazon ElastiCache for Redis as it meets the in-memory, high availability, low latency requirements

Amazon ElastiCache for Redis is a blazing fast in-memory data store that provides sub-millisecond latency to power internet-scale real-time applications. Amazon ElastiCache for Redis is a great choice for real-time transactional and analytical processing use cases such as caching, chat/messaging, gaming leaderboards, geospatial, machine learning, media

streaming, queues, real-time analytics, and session store. ElastiCache for Redis can be used to power the live leaderboard, so this option is correct.

Amazon ElastiCache for Redis Overview:



Power the on-demand, live leaderboard using Amazon DynamoDB with DynamoDB Accelerator (DAX) as it meets the in-memory, high availability, low latency requirements

Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multiregion, multimaster, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DAX is a DynamoDB-compatible caching service that enables you to benefit from fast in-memory performance for demanding applications. So DynamoDB with DAX can be used to power the live leaderboard.

Incorrect options:

Power the on-demand, live leaderboard using Amazon Neptune as it meets the in-memory, high availability, low latency requirements - Amazon Neptune is a fast, reliable, fully-managed graph database service that makes it easy to build and run applications that work with highly connected datasets. Neptune is not an in-memory database, so this option is not correct.

Power the on-demand, live leaderboard using Amazon DynamoDB as it meets the in-memory, high availability, low latency requirements - DynamoDB is not an in-memory database, so this option is not correct.

Power the on-demand, live leaderboard using Amazon RDS for Aurora as it meets the in-memory, high availability, low latency requirements - Amazon Aurora is a MySQL and PostgreSQL-compatible relational database built for the cloud, that combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases. Amazon Aurora features a distributed, fault-tolerant, self-healing storage system that auto-scales up to 128TB per database instance. Aurora is not an in-memory database, so this option is not correct.

References:

<https://aws.amazon.com/elasticache/>

<https://aws.amazon.com/elasticache/redis/>

<https://aws.amazon.com/dynamodb/dax/>

Domain

Design High-Performing Architectures

Question 42 Incorrect

A healthcare company is developing a secure internal web portal hosted on AWS. The application must communicate with legacy systems that reside in the company's on-premises data centers. These data centers are connected to AWS via a site-to-site VPN. The company uses Amazon Route 53 as its DNS solution and requires the application to resolve private DNS records for the on-premises services from within its Amazon VPC.

What is the MOST secure and appropriate way to meet these DNS resolution requirements?

Configure a Route 53 Resolver inbound endpoint and create a DNS forwarding rule. Enable recursive DNS resolution in the VPC to access on-premises services

Your answer is incorrect

Create a Route 53 private hosted zone for the on-premises domain. Associate the hosted zone with the VPC to allow the application to resolve DNS names of the on-premises services

Create a hybrid connectivity gateway and attach the on-premises DNS servers to Route 53 as authoritative zones for internal domains

Correct answer

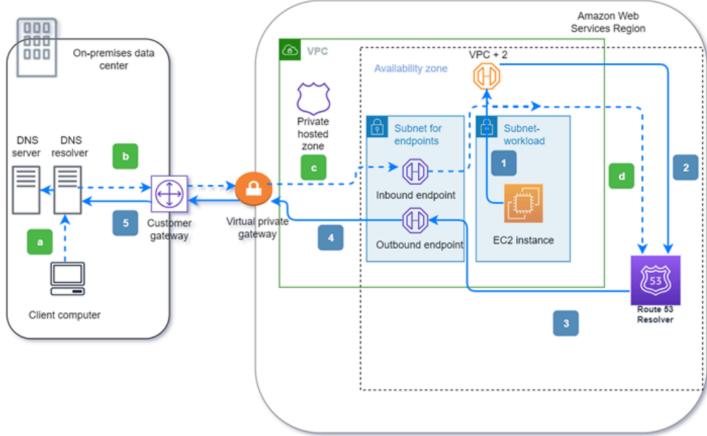
Create a Route 53 Resolver outbound endpoint. Define a forwarding rule that routes DNS queries for on-premises domains to the on-premises DNS server. Associate the rule with the VPC

Overall explanation

Correct option:

Create a Route 53 Resolver outbound endpoint. Define a forwarding rule that routes DNS queries for on-premises domains to the on-premises DNS server. Associate the rule with the VPC

This solution allows DNS queries from AWS-based workloads (inside the VPC) to resolve private on-premises domain names using a Route 53 Resolver outbound endpoint. The outbound endpoint forwards DNS queries from the VPC to the on-premises DNS server across the site-to-site VPN. The forwarding rules determine which domain names are sent to the on-premises DNS server. Associating the rule with the correct VPC ensures that the application can securely resolve on-premises DNS records. This method follows AWS best practices for hybrid DNS resolution.



via - <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resolver.html>

Incorrect options:

Configure a Route 53 Resolver inbound endpoint and create a DNS forwarding rule.
Enable recursive DNS resolution in the VPC to access on-premises services - An inbound endpoint allows DNS queries from the on-premises network to resolve AWS-hosted private domains, not the other way around. The application's requirement is to resolve on-premises DNS names from within AWS, which requires an outbound endpoint instead. Furthermore, there is no VPC configuration called "recursive DNS resolution" in this context. This option misapplies the purpose of inbound endpoints.

Create a hybrid connectivity gateway and attach the on-premises DNS servers to Route 53 as authoritative zones for internal domains - There is no such construct as a "hybrid connectivity gateway" in AWS that allows attaching DNS servers directly to Route 53. Route 53 does not support attaching on-premises DNS servers as authoritative sources. Instead, forwarding rules and Resolver endpoints are the proper mechanisms to integrate on-premises DNS resolution with AWS. This option misrepresents available AWS DNS and networking capabilities.

Create a Route 53 private hosted zone for the on-premises domain. Associate the hosted zone with the VPC to allow the application to resolve DNS names of the on-

on-premises services - A Route 53 private hosted zone is designed to resolve DNS names within AWS resources for domains hosted on AWS, not for external or on-premises DNS domains. Creating a private hosted zone for an on-premises domain name does not enable resolution of actual on-premises DNS records unless those records are manually duplicated in Route 53, which would introduce operational overhead and potential inconsistency. The correct approach for hybrid DNS resolution is to use Route 53 Resolver outbound endpoints with forwarding rules that point to the on-premises DNS servers.

References:

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resolver.html>

<https://docs.aws.amazon.com/whitepapers/latest/hybrid-cloud-dns-options-for-vpc/route-53-resolver-endpoints-and-forwarding-rules.html>

Domain

Design Secure Architectures

Question 43 Correct

A development team requires permissions to list an Amazon S3 bucket and delete objects from that bucket. A systems administrator has created the following IAM policy to provide access to the bucket and applied that policy to the group. The group is not able to delete objects in the bucket. The company follows the principle of least privilege.

```
"Version": "2021-10-17",
"Statement": [
    {
        "Action": [
            "s3>ListBucket",
            "s3>DeleteObject"
        ],
        "Resource": [
            "arn:aws:s3:::example-bucket"
        ],
        "Effect": "Allow"
    }
]
```

Which statement should a solutions architect add to the policy to address this issue?

```
{  
    "Action": [  
        "s3>DeleteObject"  
    ],  
    "Resource": [  
        "arn:aws:s3:::example-bucket*"  
    ],  
    "Effect": "Allow"  
}
```

```
{  
    "Action": [  
        "s3:*"  
    ],  
    "Resource": [  
        "arn:aws:s3:::example-bucket/*"  
    ],  
    "Effect": "Allow"  
}
```

```
{  
    "Action": [  
        "s3:*Object"  
    ],  
    "Resource": [  
        "arn:aws:s3:::example-bucket/*"  
    ],  
    "Effect": "Allow"  
}
```

Your answer is correct

```
{  
    "Action": [  
        "s3>DeleteObject"  
    ],  
    "Resource": [  
        "arn:aws:s3:::example-bucket/*"  
    ],  
    "Effect": "Allow"  
}
```

Overall explanation

Correct option:

**

```
{  
    "Action": [  
        "s3>DeleteObject"  
    ],  
    "Resource": [  
        "arn:aws:s3:::example-bucket/*"  
    ],  
    "Effect": "Allow"  
}
```

**

The main elements of a policy statement are:

1. Effect: Specifies whether the statement will Allow or Deny an action (**Allow** is the effect defined here).
2. Action: Describes a specific action or actions that will either be allowed or denied to run based on the Effect entered. API actions are unique to each service (**DeleteObject** is the action defined here).
3. Resource: Specifies the resources—for example, an Amazon S3 bucket or objects—that the policy applies to in Amazon Resource Name (ARN) format (**example-bucket/*** is the resource defined here).

This policy provides the necessary delete permissions on the resources of the Amazon S3 bucket to the group.

Incorrect options:

**

```
{  
    "Action": [  
        "s3:*Object"  
    ],  
    "Resource": [  
        "arn:aws:s3::::example-bucket/*"  
    ],  
    "Effect": "Allow"  
}
```

** - This policy is incorrect as the action value is invalid

**

```
{  
    "Action": [  
        "s3:*"  
    ],  
    "Resource": [  
        "arn:aws:s3::::example-bucket/*"  
    ],  
    "Effect": "Allow"  
}
```

** - This policy is incorrect since it allows all actions on the resource, which violates the principle of least privilege, as required by the given use case.

**

```
{  
    "Action": [  
        "s3>DeleteObject"  
    ],  
    "Resource": [  
        "arn:aws:s3::::example-bucket*"  
    ],  
    "Effect": "Allow"  
}
```

** - This is incorrect, as the resource name is incorrect. It should have a `/*` after the bucket name.

Reference:

<https://aws.amazon.com/blogs/security/techniques-for-writing-least-privilege-iam-policies/>

Domain

Design Secure Architectures

Question 44 Correct

A retail company runs a customer management system backed by a Microsoft SQL Server database. The system is tightly integrated with applications that rely on T-SQL queries. The company wants to modernize its infrastructure by migrating to Amazon Aurora PostgreSQL, but it needs to avoid major modifications to the existing application logic.

Which combination of actions should the company take to achieve this goal with minimal application refactoring? (Select two)

Your selection is correct

Use AWS Schema Conversion Tool (AWS SCT) along with AWS Database Migration Service (AWS DMS) to migrate the schema and data

Configure Amazon Aurora PostgreSQL with a custom endpoint that emulates Microsoft SQL Server behavior

Use Amazon Aurora Global Database to replicate data across regions for compatibility

Use AWS Glue to convert T-SQL queries to PostgreSQL-compatible SQL during the migration

Your selection is correct

Deploy Babelfish for Aurora PostgreSQL to enable support for T-SQL commands

Overall explanation

Correct options:

Deploy Babelfish for Aurora PostgreSQL to enable support for T-SQL commands

Babelfish allows Aurora PostgreSQL to understand T-SQL (Microsoft SQL Server's query language) and SQL Server wire protocol, enabling applications to communicate with Aurora using SQL Server-style queries with minimal code changes. This is ideal for minimizing application code refactoring.

Using Babelfish for Aurora PostgreSQL

[PDF](#)[RSS](#) Focus mode

Babelfish for Aurora PostgreSQL extends your Aurora PostgreSQL DB cluster with the ability to accept database connections from SQL Server clients. With Babelfish, applications that were originally built for SQL Server can work directly with Aurora PostgreSQL with few code changes compared to a traditional migration and without changing database drivers. For more information about migrating, see [Migrating a SQL Server database to Babelfish for Aurora PostgreSQL](#).

Babelfish provides an additional endpoint for an Aurora PostgreSQL database cluster that allows it to understand the SQL Server wire-level protocol and commonly used SQL Server statements. Client applications that use the Tabular Data Stream (TDS) wire protocol can connect natively to the TDS listener port on Aurora PostgreSQL. To learn more about TDS, see [\[MS-TDS\]: Tabular Data Stream Protocol](#) on the Microsoft website.

 **Note**

Babelfish for Aurora PostgreSQL supports TDS versions 7.1 through 7.4.

Babelfish also provides access to data using the PostgreSQL connection. By default, both SQL dialects supported by Babelfish are available through their native wire protocols at the following ports:

- SQL Server dialect (T-SQL), clients connect to port 1433.
- PostgreSQL dialect (PL/pgSQL), clients connect to port 5432.

Babelfish runs the Transact-SQL (T-SQL) language with some differences. For more information, see [Differences between Babelfish for Aurora PostgreSQL and SQL Server](#).

via - <https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/babelfish.html>

Use AWS Schema Conversion Tool (AWS SCT) along with AWS Database Migration Service (AWS DMS) to migrate the schema and data

The AWS Schema Conversion Tool (SCT) helps convert the SQL Server schema to PostgreSQL-compatible syntax, and AWS Database Migration Service (DMS) can move the actual data with minimal downtime. These tools are designed for database migration and are essential for schema and data transfer.

Incorrect options:

Configure Amazon Aurora PostgreSQL with a custom endpoint that emulates Microsoft SQL Server behavior - Aurora endpoints do not provide protocol-level emulation of SQL Server unless Babelfish is explicitly enabled. There is no feature to make Aurora natively emulate SQL Server behavior without Babelfish.

Use Amazon Aurora Global Database to replicate data across regions for compatibility - Aurora Global Database helps with cross-region disaster recovery and read scalability, but it does not assist with SQL Server compatibility or reduce application code changes.

Use AWS Glue to convert T-SQL queries to PostgreSQL-compatible SQL during the migration - AWS Glue is primarily used for ETL and data transformation, not for application SQL query conversion. It cannot translate T-SQL into PostgreSQL syntax.

References:

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/babelfish.html>

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/babelfish-compatibility.html>

Domain

Design High-Performing Architectures

Question 45 Correct

An e-commerce company is looking for a solution with high availability, as it plans to migrate its flagship application to a fleet of Amazon Elastic Compute Cloud (Amazon EC2) instances. The solution should allow for content-based routing as part of the architecture.

As a Solutions Architect, which of the following will you suggest for the company?

Use an Auto Scaling group for distributing traffic to the Amazon EC2 instances spread across different Availability Zones (AZs). Configure an elastic IP address (EIP) to mask any failure of an instance

Use a Network Load Balancer for distributing traffic to the Amazon EC2 instances spread across different Availability Zones (AZs). Configure a Private IP address to mask any failure of an instance

Your answer is correct

Use an Application Load Balancer for distributing traffic to the Amazon EC2 instances spread across different Availability Zones (AZs). Configure Auto Scaling group to mask any failure of an instance

Use an Auto Scaling group for distributing traffic to the Amazon EC2 instances spread across different Availability Zones (AZs). Configure a Public IP address to mask any failure of an instance

Overall explanation

Correct option:

Use an Application Load Balancer for distributing traffic to the Amazon EC2 instances spread across different Availability Zones (AZs). Configure Auto Scaling group to mask any failure of an instance

The Application Load Balancer (ALB) is best suited for load balancing HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers. Operating at the individual request level (Layer 7), the Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request.

This is the correct option since the question has a specific requirement for content-based routing which can be configured via the Application Load Balancer. Different Availability Zones (AZs) provide high availability to the overall architecture and Auto Scaling group will help mask any instance failures.

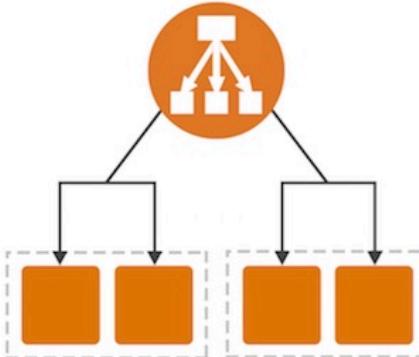
More info on Application Load Balancer:

A Layer 7 load balancer is more sophisticated and more powerful. It inspects packets, has access to HTTP and HTTPS headers, and (armed with more information) can do a more intelligent job of spreading the load out to the target.

Application Load Balancing for AWS

Today we are launching a new Application Load Balancer option for ELB. This option runs at Layer 7 and supports a number of advanced features. The original option (now called a Classic Load Balancer) is still available to you and continues to offer Layer 4 and Layer 7 functionality.

Application Load Balancers support content-based routing, and supports applications that run in containers. They support a pair of industry-standard protocols (WebSocket and HTTP/2) and also provide additional visibility into the health of the target instances and containers. Web sites and mobile apps, running in containers or on EC2 instances, will benefit from the use of Application Load Balancers.



Let's take a closer look at each of these features and then create a new Application Load Balancer of our very own!

Content-Based Routing

An Application Load Balancer has access to HTTP headers and allows you to route requests to different backend services accordingly. For example, you might want to send requests that include `/api` in the URL path to one group of servers (we call these target groups) and requests that include `/mobile` to another. Routing requests in this fashion allows you to build applications that are composed of multiple microservices that can run and be scaled independently.

As you will see in a moment, each Application Load Balancer allows you to define up to 10 URL-based rules to route requests to target groups. Over time, we plan to give you access to other routing methods.

via -

<https://aws.amazon.com/blogs/aws/new-aws-application-load-balancer/>

Incorrect options:

Use a Network Load Balancer for distributing traffic to the Amazon EC2 instances spread across different Availability Zones (AZs). Configure a Private IP address to mask any failure of an instance - Network Load Balancer cannot facilitate content-based routing so this option is incorrect.

Use an Auto Scaling group for distributing traffic to the Amazon EC2 instances spread across different Availability Zones (AZs). Configure an elastic IP address (EIP) to mask any failure of an instance

Use an Auto Scaling group for distributing traffic to the Amazon EC2 instances spread across different Availability Zones (AZs). Configure a Public IP address to mask any failure of an instance

Both these options are incorrect as you cannot use the Auto Scaling group to distribute traffic to the Amazon EC2 instances.

An elastic IP address (EIP) is a static, public, IPv4 address allocated to your AWS account. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account. Elastic IPs do not change and remain allocated to your account until you delete them.

More info on Elastic Load Balancing (ELB):

Elastic IP Addresses

An *Elastic IP address* is a static, public, IPv4 address allocated to your AWS account. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account. Elastic IPs do not change and remain allocated to your account until you delete them.

An Elastic IP address is allocated from the public AWS IPv4 network ranges in a specific region. If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet; for example, to connect to your instance from your local computer. Elastic IP addresses are mapped via an Internet Gateway to the private address of the instance. Once you associate an Elastic IP address with an instance, it remains associated until you remove the association or associate the address with another resource.

Elastic IP addresses are one method for handling failover, especially for legacy type applications that cannot be scaled horizontally. In the event of a failure of a single server with an associated Elastic IP address, the failover mechanism can re-associate the Elastic IP address to a replacement instance, ideally in an automated fashion. While this scenario may experience downtime for the application, the time may be limited to the time it takes to detect the failure and quickly re-associate the Elastic IP address to the replacement resource.

Where higher availability levels are required, you can use multiple instances and an Elastic Load Balancer.

Elastic Load Balancing

Elastic Load Balancing is an AWS service that automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions, and ensures only healthy targets receive traffic. It can handle the varying load of your application traffic in a single Availability Zone or across multiple AZs, and supports the ability to load balance across AWS and on-premises resources in the same load balancer.

Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant.

via -

<https://docs.aws.amazon.com/whitepapers/latest/fault-tolerant-components/fault-tolerant-components.pdf>

You can span your Auto Scaling group across multiple Availability Zones (AZs) within an AWS Region and then attaching a load balancer to distribute incoming traffic across those zones.

Expanding your scaled and load-balanced application to an additional Availability Zone

[PDF](#) | [Kindle](#) | [RSS](#)

You can take advantage of the safety and reliability of geographic redundancy by spanning your Auto Scaling group across multiple Availability Zones within a Region and then attaching a load balancer to distribute incoming traffic across those zones. Incoming traffic is distributed equally across all Availability Zones enabled for your load balancer.

When one Availability Zone becomes unhealthy or unavailable, Amazon EC2 Auto Scaling launches new instances in an unaffected zone. When the unhealthy Availability Zone returns to a healthy state, Amazon EC2 Auto Scaling automatically redistributes the application instances evenly across all of the zones for your Auto Scaling group. Amazon EC2 Auto Scaling does this by attempting to launch new instances in the Availability Zone with the fewest instances. If the attempt fails, however, Amazon EC2 Auto Scaling attempts to launch in other Availability Zones until it succeeds.

You can expand the availability of your scaled and load-balanced application by adding an Availability Zone to your Auto Scaling group and then enabling that zone for your load balancer. After you've enabled the new Availability Zone, the load balancer begins to route traffic equally among all the enabled zones.

Limitations:

- An Auto Scaling group can contain Amazon EC2 instances from multiple Availability Zones within the same Region. However, an Auto Scaling group can't contain instances from multiple Regions.
- When you enable an Availability Zone for your load balancer, you specify one subnet from that Availability Zone. Note that you can select at most one subnet per Availability Zone.
- When editing Application Load Balancers, you must specify subnets from at least two Availability Zones.
- When editing Network Load Balancers, you cannot disable the enabled Availability Zones, but you can enable additional ones.
- For internet-facing load balancers, the subnets that you specify for the load balancer must have at least 8 available IP addresses.
- For Gateway Load Balancers, you cannot change the Availability Zones or subnets that were added when the load balancer was created.

via - <https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-add-availability->

References:

<https://aws.amazon.com/blogs/aws/new-aws-application-load-balancer/>

<https://docs.aws.amazon.com/whitepapers/latest/fault-tolerant-components/fault-tolerant-components.pdf>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-add-availability-zone.html>

Domain

Design Resilient Architectures

Question 46 Incorrect

An enterprise runs a microservices-based application on Amazon EKS, deployed on EC2 worker nodes. The application includes a frontend UI service that interacts with Amazon DynamoDB and a data-processing service that stores and retrieves files from Amazon S3. The organization needs to strictly enforce least privilege access: the UI Pods must access only DynamoDB, and the data-processing Pods must access only S3.

Which solution will best enforce these access controls within the EKS cluster?

Correct answer

Create separate Kubernetes service accounts for the UI and data services. Use IAM Roles for Service Accounts (IRSA) to map each service account to an IAM role with only the required permissions. Assign DynamoDB access to the UI Pods and S3 access to the data Pods

Create one Kubernetes service account shared across all Pods. Attach a single IAM role to this account with both AmazonS3FullAccess and AmazonDynamoDBFullAccess policies

Create IAM policies for DynamoDB and S3 access, and attach both to the EC2 instance profile used by the EKS nodes. Use Kubernetes role-based access control (RBAC) to control service-level permissions within the cluster

Your answer is incorrect

Attach an IAM policy directly to each Pod using Kubernetes annotations. Assign the S3 policy to data-service Pods and the DynamoDB policy to UI Pods

Overall explanation

Correct option:

Create separate Kubernetes service accounts for the UI and data services. Use IAM Roles for Service Accounts (IRSA) to map each service account to an IAM role with only the required permissions. Assign DynamoDB access to the UI Pods and S3 access to the data Pods

This option is correct because it uses IAM Roles for Service Accounts (IRSA), which is the AWS-recommended method for providing fine-grained IAM permissions to individual Pods in an Amazon EKS cluster. By creating separate Kubernetes service accounts—one for the UI service and one for the data-processing service—and associating each with an IAM role scoped to only the required AWS permissions, the company ensures that each service can access only its designated AWS resource (DynamoDB for the UI, S3 for the data service). IRSA works by leveraging the OpenID Connect (OIDC) identity provider that EKS integrates with, allowing Pods to assume roles dynamically without sharing EC2 instance-level credentials. This approach enforces least privilege, ensures Pod-level isolation, and avoids the security and operational drawbacks of broader node-level roles.

Incorrect options:

Create one Kubernetes service account shared across all Pods. Attach a single IAM role to this account with both AmazonS3FullAccess and AmazonDynamoDBFullAccess policies - This option defeats the purpose of enforcing least privilege. Sharing a single Kubernetes service account with a broad IAM role across all Pods gives both the UI and

data service components unnecessary access. Even though IRSA could be used here, the identity boundary is not separated, so the UI Pods can access S3 and the data Pods can access DynamoDB—both of which violate the access requirements.

Attach an IAM policy directly to each Pod using Kubernetes annotations. Assign the S3 policy to data-service Pods and the DynamoDB policy to UI Pods - IAM policies cannot be directly attached to Pods using annotations. AWS IAM does not natively support Pod-level identity mappings outside of IAM Roles for Service Accounts (IRSA). Without IRSA, Pods only inherit permissions from the underlying node's IAM role, meaning all Pods on the same EC2 node will share the same set of permissions. This option is not technically feasible in EKS.

Create IAM policies for DynamoDB and S3 access, and attach both to the EC2 instance profile used by the EKS nodes. Use Kubernetes role-based access control (RBAC) to control service-level permissions within the cluster - Attaching both IAM policies to the EC2 instance profile means that all Pods running on that node will inherit all permissions, violating least privilege. Kubernetes RBAC controls API access within the cluster, not access to AWS services like S3 or DynamoDB. Therefore, RBAC cannot enforce IAM policy-level restrictions, making this approach insecure and non-compliant with the requirements.

References:

<https://docs.aws.amazon.com/eks/latest/userguide/iam-roles-for-service-accounts.html>

<https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html>

Domain

Design Secure Architectures

Question 47 Correct

A gaming company is developing a mobile game that streams score updates to a backend processor and then publishes results on a leaderboard. The company has hired you as an AWS Certified Solutions Architect Associate to design a solution that can handle major traffic spikes, process the mobile game updates in the order of receipt, and store the processed updates in a highly available database. The company wants to minimize the management overhead required to maintain the solution.

Which of the following will you recommend to meet these requirements?

Push score updates to an Amazon Simple Notification Service (Amazon SNS) topic, subscribe an AWS Lambda function to this Amazon SNS topic to process the updates and then store these processed updates in a SQL database running on Amazon EC2 instance

Push score updates to Amazon Kinesis Data Streams which uses a fleet of Amazon EC2 instances (with Auto Scaling) to process the updates in Amazon Kinesis Data Streams and then store these processed updates in Amazon DynamoDB

Push score updates to an Amazon Simple Queue Service (Amazon SQS) queue which uses a fleet of Amazon EC2 instances (with Auto Scaling) to process these updates in the Amazon SQS queue and then store these processed updates in an Amazon RDS MySQL database

Your answer is correct

Push score updates to Amazon Kinesis Data Streams which uses an AWS Lambda function to process these updates and then store these processed updates in Amazon DynamoDB

Overall explanation

Correct option:

Push score updates to Amazon Kinesis Data Streams which uses an AWS Lambda function to process these updates and then store these processed updates in Amazon DynamoDB

To help ingest real-time data or streaming data at large scales, you can use Amazon Kinesis Data Streams (KDS). KDS can continuously capture gigabytes of data per second from hundreds of thousands of sources. The data collected is available in milliseconds, enabling

real-time analytics. KDS provides ordering of records, as well as the ability to read and/or replay records in the same order to multiple Amazon Kinesis Applications.

AWS Lambda integrates natively with Kinesis Data Streams. The polling, checkpointing, and error handling complexities are abstracted when you use this native integration. The processed data can then be configured to be saved in Amazon DynamoDB.

Incorrect options:

Push score updates to an Amazon Simple Queue Service (Amazon SQS) queue which uses a fleet of Amazon EC2 instances (with Auto Scaling) to process these updates in the Amazon SQS queue and then store these processed updates in an Amazon RDS MySQL database

Push score updates to Amazon Kinesis Data Streams which uses a fleet of Amazon EC2 instances (with Auto Scaling) to process the updates in Amazon Kinesis Data Streams and then store these processed updates in Amazon DynamoDB

Push score updates to an Amazon Simple Notification Service (Amazon SNS) topic, subscribe an AWS Lambda function to this Amazon SNS topic to process the updates and then store these processed updates in a SQL database running on Amazon EC2 instance

These three options use Amazon EC2 instances as part of the solution architecture. The use-case seeks to minimize the management overhead required to maintain the solution. However, Amazon EC2 instances involve several maintenance activities such as managing the guest operating system and software deployed to the guest operating system, including updates and security patches, etc. Hence these options are incorrect.

Reference:

<https://aws.amazon.com/blogs/big-data/best-practices-for-consuming-amazon-kinesis-data-streams-using-aws-lambda/>

Domain

Design High-Performing Architectures

Question 48 Correct

An IT company wants to review its security best-practices after an incident was reported where a new developer on the team was assigned full access to Amazon DynamoDB. The developer

accidentally deleted a couple of tables from the production environment while building out a new feature.

Which is the MOST effective way to address this issue so that such incidents do not recur?

Your answer is correct

Use permissions boundary to control the maximum permissions employees can grant to the IAM principals

The CTO should review the permissions for each new developer's IAM user so that such incidents don't recur

Remove full database access for all IAM users in the organization

Only root user should have full database access in the organization

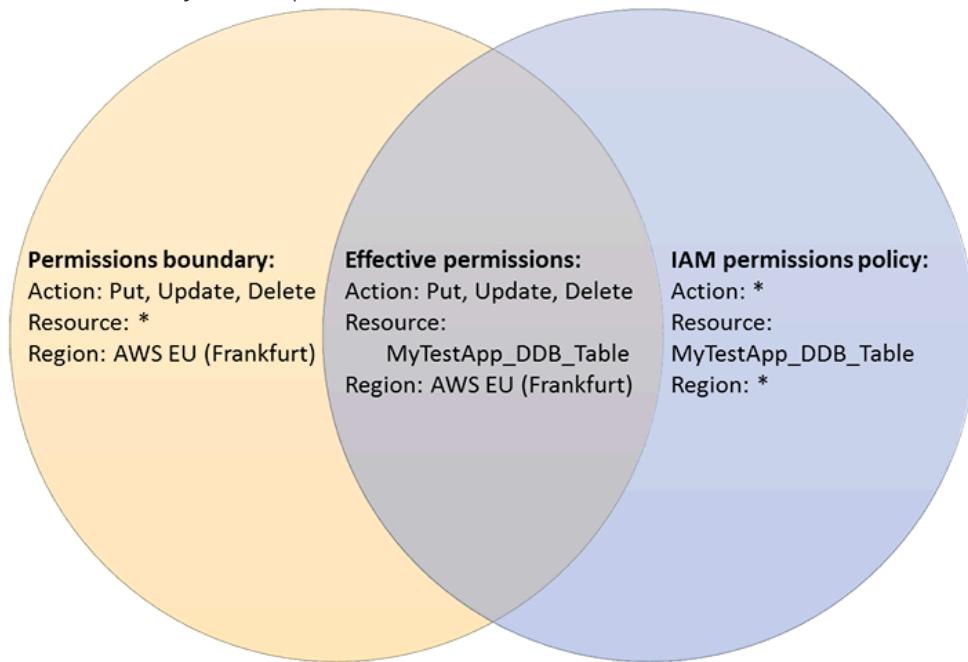
Overall explanation

Correct option:

Use permissions boundary to control the maximum permissions employees can grant to the IAM principals

A permissions boundary can be used to control the maximum permissions employees can grant to the IAM principals (that is, users and roles) that they create and manage. As the IAM administrator, you can define one or more permissions boundaries using managed policies and allow your employee to create a principal with this boundary. The employee can then attach a permissions policy to this principal. However, the effective permissions of the principal are the intersection of the permissions boundary and permissions policy. As a result, the new principal cannot exceed the boundary that you defined. Therefore, using the permissions boundary offers the right solution for this use-case.

Permission Boundary Example:



via - <https://aws.amazon.com/blogs/security/delegate-permission-management-to-developers-using-iam-permissions-boundaries/>

Incorrect options:

Remove full database access for all IAM users in the organization - It is not practical to remove full access for all IAM users in the organization because a select set of users need this access for database administration. So this option is not correct.

The CTO should review the permissions for each new developer's IAM user so that such incidents don't recur - Likewise the CTO is not expected to review the permissions for each new developer's IAM user, as this is best done via an automated procedure. This option has been added as a distractor.

Only root user should have full database access in the organization - As a best practice, the root user should not access the AWS account to carry out any administrative procedures. So this option is not correct.

Reference:

<https://aws.amazon.com/blogs/security/delegate-permission-management-to-developers-using-iam-permissions-boundaries/>

Domain

Design Secure Architectures

Question 49 Correct

An IT security consultancy is working on a solution to protect data stored in Amazon S3 from any malicious activity as well as check for any vulnerabilities on Amazon EC2 instances.

As a solutions architect, which of the following solutions would you suggest to help address the given requirement?

Use Amazon Inspector to monitor any malicious activity on data stored in Amazon S3. Use security assessments provided by Amazon Inspector to check for vulnerabilities on Amazon EC2 instances

Use Amazon GuardDuty to monitor any malicious activity on data stored in Amazon S3. Use security assessments provided by Amazon GuardDuty to check for vulnerabilities on Amazon EC2 instances

Your answer is correct

Use Amazon GuardDuty to monitor any malicious activity on data stored in Amazon S3. Use security assessments provided by Amazon Inspector to check for vulnerabilities on Amazon EC2 instances

Use Amazon Inspector to monitor any malicious activity on data stored in Amazon S3. Use security assessments provided by Amazon GuardDuty to check for vulnerabilities on Amazon EC2 instances

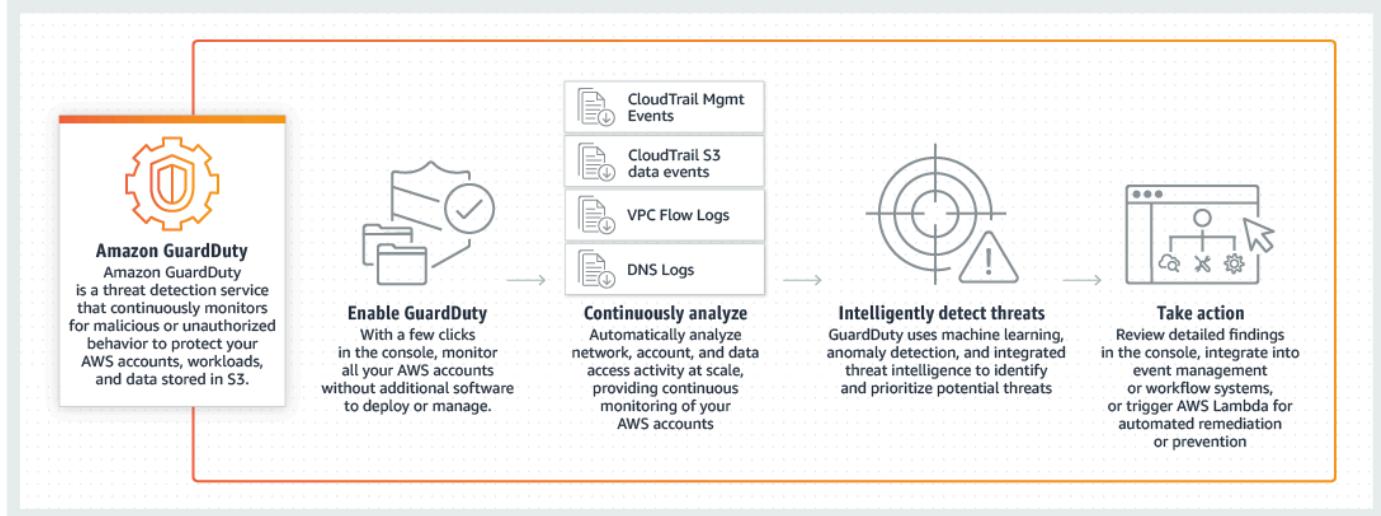
Overall explanation

Correct option:

Use Amazon GuardDuty to monitor any malicious activity on data stored in Amazon S3. Use security assessments provided by Amazon Inspector to check for vulnerabilities on Amazon EC2 instances

Amazon GuardDuty offers threat detection that enables you to continuously monitor and protect your AWS accounts, workloads, and data stored in Amazon S3. GuardDuty analyzes continuous streams of meta-data generated from your account and network activity found in AWS CloudTrail Events, Amazon VPC Flow Logs, and DNS Logs. It also uses integrated threat intelligence such as known malicious IP addresses, anomaly detection, and machine learning to identify threats more accurately.

How Amazon GuardDuty works:



via - <https://aws.amazon.com/guardduty/>

Amazon Inspector security assessments help you check for unintended network accessibility of your Amazon EC2 instances and for vulnerabilities on those EC2 instances. Amazon Inspector assessments are offered to you as pre-defined rules packages mapped to common security best practices and vulnerability definitions.

Incorrect options:

Use Amazon GuardDuty to monitor any malicious activity on data stored in Amazon S3. Use security assessments provided by Amazon GuardDuty to check for vulnerabilities on Amazon EC2 instances

Use Amazon Inspector to monitor any malicious activity on data stored in Amazon S3. Use security assessments provided by Amazon Inspector to check for vulnerabilities on Amazon EC2 instances

Use Amazon Inspector to monitor any malicious activity on data stored in Amazon S3. Use security assessments provided by Amazon GuardDuty to check for vulnerabilities on Amazon EC2 instances

These three options contradict the explanation provided above, so these options are incorrect.

References:

<https://aws.amazon.com/guardduty/>

<https://aws.amazon.com/inspector/>

Domain

Design Secure Architectures

Question 50 Correct

An ivy-league university is assisting NASA to find potential landing sites for exploration vehicles of unmanned missions to our neighboring planets. The university uses High Performance Computing (HPC) driven application architecture to identify these landing sites.

Which of the following Amazon EC2 instance topologies should this application be deployed on?

The Amazon EC2 instances should be deployed in a partition placement group so that distributed workloads can be handled effectively

The Amazon EC2 instances should be deployed in a spread placement group so that there are no correlated failures

Your answer is correct

The Amazon EC2 instances should be deployed in a cluster placement group so that the underlying workload can benefit from low network latency and high network throughput

The Amazon EC2 instances should be deployed in an Auto Scaling group so that application meets high availability requirements

Overall explanation

Correct option:

The Amazon EC2 instances should be deployed in a cluster placement group so that the underlying workload can benefit from low network latency and high network throughput

The key thing to understand in this question is that HPC workloads need to achieve low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of HPC applications. Cluster placement groups pack instances close together inside an Availability Zone. These are recommended for applications that benefit from low network latency, high network throughput, or both. Therefore this option is the correct answer.

Cluster Placement Group:

Cluster placement groups

A cluster placement group is a logical grouping of instances within a single Availability Zone. A cluster placement group can span peered VPCs in the same Region. Instances in the same cluster placement group enjoy a higher per-flow throughput limit of up to 10 Gbps for TCP/IP traffic and are placed in the same high-bisection bandwidth segment of the network.

The following image shows instances that are placed into a cluster placement group.



via - <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html>

Incorrect options:

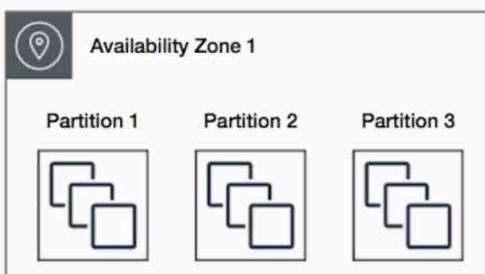
The Amazon EC2 instances should be deployed in a partition placement group so that distributed workloads can be handled effectively - A partition placement group spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads, such as Hadoop, Cassandra, and Kafka. A partition placement group can have a maximum of seven partitions per Availability Zone. Since a partition placement group can have partitions in multiple Availability Zones in the same region, therefore instances will not have low-latency network performance. Hence the partition placement group is not the right fit for HPC applications.

Partition Placement Group:

Partition placement groups

Partition placement groups help reduce the likelihood of correlated hardware failures for your application. When using partition placement groups, Amazon EC2 divides each group into logical segments called partitions. Amazon EC2 ensures that each partition within a placement group has its own set of racks. Each rack has its own network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure within your application.

The following image is a simple visual representation of a partition placement group in a single Availability Zone. It shows instances that are placed into a partition placement group with three partitions—**Partition 1**, **Partition 2**, and **Partition 3**. Each partition comprises multiple instances. The instances in a partition do not share racks with the instances in the other partitions, allowing you to contain the impact of a single hardware failure to only the associated partition.



via - <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html>

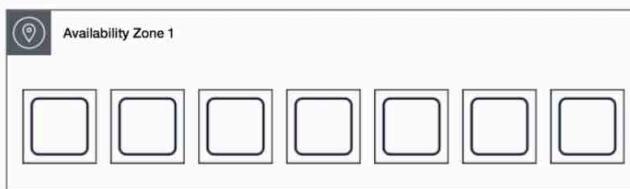
The Amazon EC2 instances should be deployed in a spread placement group so that there are no correlated failures - A spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source. The instances are placed across distinct underlying hardware to reduce correlated failures. You can have a maximum of seven running instances per Availability Zone per group. Since a spread placement group can span multiple Availability Zones in the same Region, therefore instances will not have low-latency network performance. Hence spread placement group is not the right fit for HPC applications.

Spread Placement Group:

Spread placement groups

A spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source.

The following image shows seven instances in a single Availability Zone that are placed into a spread placement group. The seven instances are placed on seven different racks.



via - <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html>

The Amazon EC2 instances should be deployed in an Auto Scaling group so that application meets high availability requirements - An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling. You do not use Auto Scaling groups per se to meet HPC requirements.

Reference:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html>

Domain

Design High-Performing Architectures

Question 51 Correct

A government agency is developing a online application to assist users in submitting permit requests through a web-based interface. The system architecture consists of a front-end web application tier and a background processing tier that handles the validation and submission of

the forms. The application is expected to see high traffic and it must ensure that every submitted request is processed exactly once, with no loss of data.

Which design choice best satisfies these requirements?

Implement an Amazon SQS standard queue to reliably buffer and deliver form submissions from the web application layer to the processing tier

Your answer is correct

Implement an Amazon SQS FIFO queue to reliably buffer and deliver form submissions from the web application layer to the processing tier

Leverage Amazon API Gateway to pass the form submissions to AWS Lambda for processing in real time

Leverage Amazon EventBridge to send events from the web application to the processing tier for asynchronous form handling

Overall explanation

Correct option:

Implement an Amazon SQS FIFO queue to reliably buffer and deliver form submissions from the web application layer to the processing tier

SQS FIFO (First-In-First-Out) queues are designed for exactly-once message processing with order preservation. They prevent duplicate deliveries, making them ideal for transactional workflows such as registration or permit submissions. FIFO queues guarantee that each message is delivered and processed once and only once, which satisfies the application's reliability and data integrity requirements.

Incorrect options:

Implement an Amazon SQS standard queue to reliably buffer and deliver form submissions from the web application layer to the processing tier - The option to leverage an Amazon Simple Queue Service (Amazon SQS) standard queue between the web and processing tiers is incorrect for this use case because SQS standard queues do not guarantee exactly-once message delivery. Instead, they provide at-least-once delivery and best-effort ordering. In the context of form submissions or permit requests—where each submission must be processed exactly once and in the correct order—standard queues introduce the risk of duplicate messages or out-of-order processing

Leverage Amazon EventBridge to send events from the web application to the processing tier for asynchronous form handling - Amazon EventBridge is a powerful event bus that enables loosely coupled architectures and event-driven applications. However, it does not guarantee exactly-once delivery. Instead, EventBridge provides at-least-once delivery semantics, which means events can be delivered more than once in certain failure scenarios. So, the key shortcomings for this use case are - no built-in ordering of events and possible duplicate event delivery.

Leverage Amazon API Gateway to pass the form submissions to AWS Lambda for processing in real time - While API Gateway with Lambda is effective for real-time synchronous processing, it does not inherently offer decoupling between tiers or a durable queue. It also lacks native support for guaranteed exactly-once processing across asynchronous workloads.

References:

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDriverGuide/welcome.html>

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDriverGuide/sqs-fifo-queues.html>

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDriverGuide/standard-queues.html>

Domain

Question 52 Correct

The payroll department at a company initiates several computationally intensive workloads on Amazon EC2 instances at a designated hour on the last day of every month. The payroll department has noticed a trend of severe performance lag during this hour. The engineering team has figured out a solution by using Auto Scaling Group for these Amazon EC2 instances and making sure that 10 Amazon EC2 instances are available during this peak usage hour. For normal operations only 2 Amazon EC2 instances are enough to cater to the workload.

As a solutions architect, which of the following steps would you recommend to implement the solution?

Configure your Auto Scaling group by creating a target tracking policy and setting the instance count to 10 at the designated hour. This causes the scale-out to happen before peak traffic kicks in at the designated hour

Configure your Auto Scaling group by creating a scheduled action that kicks-off at the designated hour on the last day of the month. Set the min count as well as the max count of instances to 10. This causes the scale-out to happen before peak traffic kicks in at the designated hour

Configure your Auto Scaling group by creating a simple tracking policy and setting the instance count to 10 at the designated hour. This causes the scale-out to happen before peak traffic kicks in at the designated hour

Your answer is correct

Configure your Auto Scaling group by creating a scheduled action that kicks-off at the designated hour on the last day of the month. Set the desired capacity

of instances to 10. This causes the scale-out to happen before peak traffic kicks in at the designated hour

Overall explanation

Correct option:

Configure your Auto Scaling group by creating a scheduled action that kicks-off at the designated hour on the last day of the month. Set the desired capacity of instances to 10. This causes the scale-out to happen before peak traffic kicks in at the designated hour

Scheduled scaling allows you to set your own scaling schedule. For example, let's say that every week the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday. You can plan your scaling actions based on the predictable traffic patterns of your web application. Scaling actions are performed automatically as a function of time and date.

A scheduled action sets the minimum, maximum, and desired sizes to what is specified by the scheduled action at the time specified by the scheduled action. For the given use case, the correct solution is to set the desired capacity to 10. When we want to specify a range of instances, then we must use min and max values.

Incorrect options:

Configure your Auto Scaling group by creating a scheduled action that kicks-off at the designated hour on the last day of the month. Set the min count as well as the max count of instances to 10. This causes the scale-out to happen before peak traffic kicks in at the designated hour - As mentioned earlier in the explanation, only when we want to specify a range of instances, then we must use min and max values. As the given use-case requires exactly 10 instances to be available during the peak hour, so we must set the desired capacity to 10. Hence this option is incorrect.

Configure your Auto Scaling group by creating a target tracking policy and setting the instance count to 10 at the designated hour. This causes the scale-out to happen before peak traffic kicks in at the designated hour

Configure your Auto Scaling group by creating a simple tracking policy and setting the instance count to 10 at the designated hour. This causes the scale-out to happen before peak traffic kicks in at the designated hour

Target tracking policy or simple tracking policy cannot be used to effect a scaling action at a certain designated hour. Both these options have been added as distractors.

Reference:

https://docs.aws.amazon.com/autoscaling/ec2/userguide/schedule_time.html

Domain

Design Resilient Architectures

Question 53 Correct

The development team at an e-commerce startup has set up multiple microservices running on Amazon EC2 instances under an Application Load Balancer. The team wants to route traffic to multiple back-end services based on the URL path of the HTTP header. So it wants requests for <https://www.example.com/orders> to go to a specific microservice and requests for <https://www.example.com/products> to go to another microservice.

Which of the following features of Application Load Balancers can be used for this use-case?

HTTP header-based routing

Query string parameter-based routing

Host-based Routing

Your answer is correct

Path-based Routing

Overall explanation

Correct option:

Path-based Routing

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and AWS Lambda functions.

If your application is composed of several individual services, an Application Load Balancer can route a request to a service based on the content of the request. Here are the different types -

Host-based Routing:

You can route a client request based on the Host field of the HTTP header allowing you to route to multiple domains from the same load balancer.

Path-based Routing:

You can route a client request based on the URL path of the HTTP header.

HTTP header-based routing:

You can route a client request based on the value of any standard or custom HTTP header.

HTTP method-based routing:

You can route a client request based on any standard or custom HTTP method.

Query string parameter-based routing:

You can route a client request based on the query string or query parameters.

Source IP address CIDR-based routing:

You can route a client request based on source IP address CIDR from where the request originates.

Path-based Routing Overview:

You can use path conditions to define rules that route requests based on the URL in the request (also known as path-based routing).

The path pattern is applied only to the path of the URL, not to its query parameters.

Path Conditions

You can use path conditions to define rules that route requests based on the URL in the request (also known as *path-based routing*).

The path pattern is applied only to the path of the URL, not to its query parameters.

A path pattern is case-sensitive, can be up to 128 characters in length, and can contain any of the following characters.

- A–Z, a–z, 0–9
- _ - . \$ / ~ " '@ : +
- & (using &)
- * (matches 0 or more characters)
- ? (matches exactly 1 character)

Example path patterns

- /img/*
- /img/*/pics

The path pattern is used to route requests but does not alter them. For example, if a rule has a path pattern of /img/*, the rule would forward a request for /img/picture.jpg to the specified target group as a request for /picture.jpg.

via - <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-listeners.html#path-conditions>

Incorrect options:

Query string parameter-based routing

HTTP header-based routing

Host-based Routing

As mentioned earlier in the explanation, none of these three types of routing support requests based on the URL path of the HTTP header. Hence these three are incorrect.

Reference:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-listeners.html>

Domain

Question 54 Correct

A retail company's dynamic website is hosted using on-premises servers in its data center in the United States. The company is launching its website in Asia, and it wants to optimize the website loading times for new users in Asia. The website's backend must remain in the United States. The website is being launched in a few days, and an immediate solution is needed.

What would you recommend?

Leverage a Amazon Route 53 geo-proximity routing policy pointing to on-premises servers

Your answer is correct

Use Amazon CloudFront with a custom origin pointing to the on-premises servers

Use Amazon CloudFront with a custom origin pointing to the DNS record of the website on Amazon Route 53

Migrate the website to Amazon S3. Use S3 cross-region replication (S3 CRR) between AWS Regions in the US and Asia

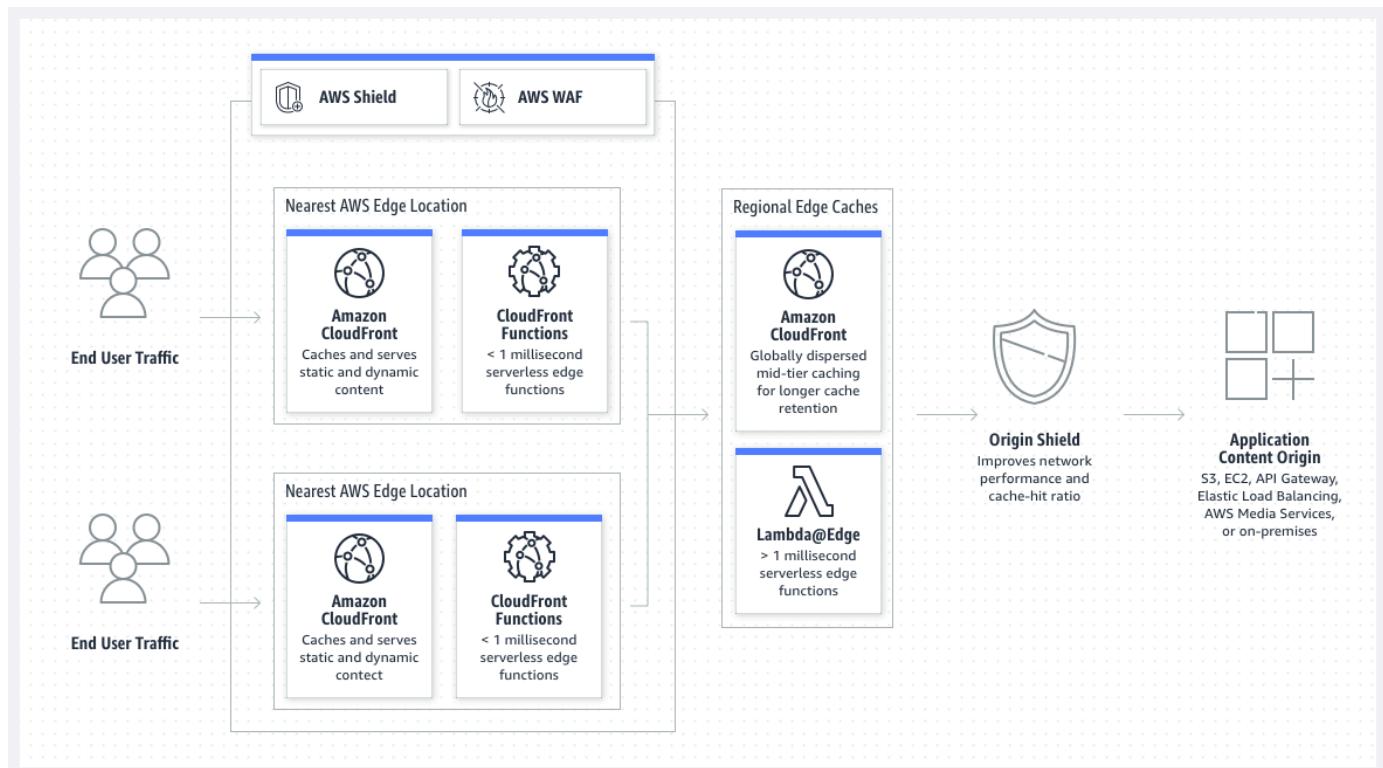
Overall explanation

Correct option:

Use Amazon CloudFront with a custom origin pointing to the on-premises servers

Amazon CloudFront is a web service that gives businesses and web application developers an easy and cost-effective way to distribute content with low latency and high data transfer speeds. Amazon CloudFront uses standard cache control headers you set on your files to identify static and dynamic content. You can use different origins for different types of content on a single site – e.g. Amazon S3 for static objects, Amazon EC2 for dynamic content, and custom origins for third-party content.

Amazon CloudFront:

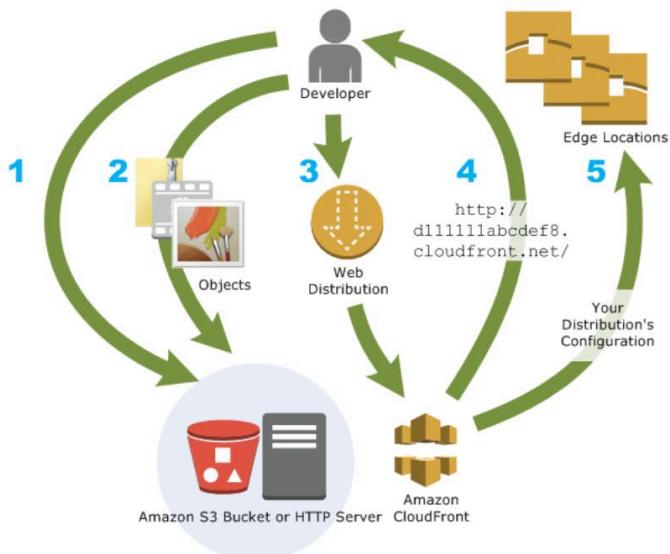


via - <https://aws.amazon.com/cloudfront/>

An origin server stores the original, definitive version of your objects. If you're serving content over HTTP, your origin server is either an Amazon S3 bucket or an HTTP server, such as a web server. Your HTTP server can run on an Amazon Elastic Compute Cloud (Amazon EC2) instance or on a server that you manage; these servers are also known as custom origins.

How you set up CloudFront to deliver content

You create a CloudFront distribution to tell CloudFront where you want content to be delivered from, and the details about how to track and manage content delivery. Then CloudFront uses computers—edge servers—that are close to your viewers to deliver that content quickly when someone wants to see it or use it.



How you configure CloudFront to deliver your content

1. You specify *origin servers*, like an Amazon S3 bucket or your own HTTP server, from which CloudFront gets your files which will then be distributed from CloudFront edge locations all over the world.
An origin server stores the original, definitive version of your objects. If you're serving content over HTTP, your origin server is either an Amazon S3 bucket or an HTTP server, such as a web server. Your HTTP server can run on an Amazon Elastic Compute Cloud (Amazon EC2) instance or on a server that you manage; these servers are also known as *custom origins*.
2. You upload your files to your origin servers. Your files, also known as *objects*, typically include web pages, images, and media files, but can be anything that can be served over HTTP.
If you're using an Amazon S3 bucket as an origin server, you can make the objects in your bucket publicly readable, so that anyone who knows the CloudFront URLs for your objects can access them. You also have the option of keeping objects private and controlling who accesses them. See [Serving private content with signed URLs and signed cookies](#).
3. You create a CloudFront *distribution*, which tells CloudFront which origin servers to get your files from when users request the files through your web site or application. At the same time, you specify details such as whether you want CloudFront to log all requests and whether you want the distribution to be enabled as soon as it's created.
4. CloudFront assigns a domain name to your new distribution that you can see in the CloudFront console, or that is returned in the response to a programmatic request, for example, an API request. If you like, you can add an alternate domain name to use instead.
5. CloudFront sends your distribution's configuration (but not your content) to all of its *edge locations* or *points of presence* (POPs)— collections of servers in geographically-dispersed data centers where CloudFront caches copies of your files.

via -

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html>

Amazon CloudFront employs a global network of edge locations and regional edge caches that cache copies of your content close to your viewers. Amazon CloudFront ensures that end-user requests are served by the closest edge location. As a result, viewer requests travel a short distance, improving performance for your viewers. Therefore for the given use case, the users in Asia will enjoy a low latency experience while using the website even though the on-premises servers continue to be in the US.

Incorrect options:

Use Amazon CloudFront with a custom origin pointing to the DNS record of the website on Amazon Route 53 - This option has been added as a distractor. CloudFront cannot have a custom origin pointing to the DNS record of the website on Route 53.

Migrate the website to Amazon S3. Use S3 cross-region replication (S3 CRR) between AWS Regions in the US and Asia - The use case states that the company operates a

dynamic website. You can use Amazon S3 to host a static website. On a static website, individual web pages include static content. They might also contain client-side scripts. By contrast, a dynamic website relies on server-side processing, including server-side scripts, such as PHP, JSP, or ASP.NET. Amazon S3 does not support server-side scripting, but AWS has other resources for hosting dynamic websites. So this option is incorrect.

Leverage a Amazon Route 53 geo-proximity routing policy pointing to on-premises servers - Since the on-premises servers continue to be in the US, so even using a Route 53 geo-proximity routing policy that directs the users in Asia to the on-premises servers in the US would not reduce the latency for the users in Asia. So this option is incorrect.

References:

<https://aws.amazon.com/cloudfront/>

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html>

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteHosting.html>

Domain

Design High-Performing Architectures

Question 55 Correct

The flagship application for a gaming company connects to an Amazon Aurora database and the entire technology stack is currently deployed in the United States. Now, the company has plans to expand to Europe and Asia for its operations. It needs the `games` table to be accessible globally but needs the `users` and `games_played` tables to be regional only.

How would you implement this with minimal application refactoring?

Use a Amazon DynamoDB global table for the `games` table and use Amazon Aurora for the `users` and `games_played` tables

Use a Amazon DynamoDB global table for the `games` table and use Amazon DynamoDB tables for the `users` and `games_played` tables

Use an Amazon Aurora Global Database for the `games` table and use Amazon DynamoDB tables for the `users` and `games_played` tables

Your answer is correct

Use an Amazon Aurora Global Database for the `games` table and use Amazon Aurora for the `users` and `games_played` tables

Overall explanation

Correct option:

Use an Amazon Aurora Global Database for the `games` table and use Amazon Aurora for the `users` and `games_played` tables

Amazon Aurora is a MySQL and PostgreSQL-compatible relational database built for the cloud, that combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases. Amazon Aurora features a distributed, fault-tolerant, self-healing storage system that auto-scales up to 128TB per database instance. Aurora is not an in-memory database.

Amazon Aurora Global Database is designed for globally distributed applications, allowing a single Amazon Aurora database to span multiple AWS regions. It replicates your data with no impact on database performance, enables fast local reads with low latency in each region, and provides disaster recovery from region-wide outages. Amazon Aurora Global Database is the correct choice for the given use-case.

For the given use-case, we, therefore, need to have two Aurora clusters, one for the global table (games table) and the other one for the local tables (users and games_played tables).

Incorrect options:

Use an Amazon Aurora Global Database for the `games` table and use Amazon DynamoDB tables for the `users` and `games_played` tables

Use a Amazon DynamoDB global table for the `games` table and use Amazon Aurora for the `users` and `games_played` tables

Use a Amazon DynamoDB global table for the `games` table and use Amazon DynamoDB tables for the `users` and `games_played` tables

Here, we want minimal application refactoring. Amazon DynamoDB and Amazon Aurora have a completely different APIs, due to Amazon Aurora being SQL and Amazon DynamoDB being NoSQL. So all three options are incorrect, as they have Amazon DynamoDB as one of the components.

Reference:

<https://aws.amazon.com/rds/aurora/faqs/>

Domain

Design Secure Architectures

Question 56 Incorrect

A junior scientist working with the Deep Space Research Laboratory at NASA is trying to upload a high-resolution image of a nebula into Amazon S3. The image size is approximately 3 gigabytes. The junior scientist is using Amazon S3 Transfer Acceleration (Amazon S3TA) for faster image upload. It turns out that Amazon S3TA did not result in an accelerated transfer.

Given this scenario, which of the following is correct regarding the charges for this image transfer?

Correct answer

The junior scientist does not need to pay any transfer charges for the image upload

The junior scientist only needs to pay Amazon S3 transfer charges for the image upload

The junior scientist only needs to pay S3TA transfer charges for the image upload

Your answer is incorrect

The junior scientist needs to pay both S3 transfer charges and S3TA transfer charges for the image upload

Overall explanation

Correct option:

The junior scientist does not need to pay any transfer charges for the image upload

There are no S3 data transfer charges when data is transferred in from the internet. Also with S3TA, you pay only for transfers that are accelerated. Therefore the junior scientist does not need to pay any transfer charges for the image upload because S3TA did not result in an accelerated transfer.

Amazon S3 Transfer Acceleration (S3TA) Overview:

Amazon S3 Transfer Acceleration can speed up content transfers to and from Amazon S3 by as much as 50-500% for long-distance transfer of larger objects. Customers who have either web or mobile applications with widespread users or applications hosted far away from their S3 bucket can experience long and variable upload and download speeds over the Internet. S3 Transfer Acceleration (S3TA) reduces the variability in Internet routing, congestion and speeds that can affect transfers, and logically shortens the distance to S3 for remote applications. S3TA improves transfer performance by routing traffic through Amazon CloudFront's globally distributed Edge Locations and over AWS backbone networks, and by using network protocol optimizations. You can turn on S3TA with a few clicks in the S3 console, and test its benefits from your location with a speed comparison tool. With S3TA, you pay only for transfers that are accelerated.

via - <https://aws.amazon.com/s3/transfer-acceleration/>

Incorrect options:

The junior scientist only needs to pay S3TA transfer charges for the image upload - Since S3TA did not result in an accelerated transfer, there are no S3TA transfer charges to be paid.

The junior scientist only needs to pay Amazon S3 transfer charges for the image upload - There are no S3 data transfer charges when data is transferred in from the internet. So this option is incorrect.

The junior scientist needs to pay both S3 transfer charges and S3TA transfer charges for the image upload - There are no Amazon S3 data transfer charges when data is transferred in from the internet. Since S3TA did not result in an accelerated transfer, there are no S3TA transfer charges to be paid.

References:

<https://aws.amazon.com/s3/transfer-acceleration/>

<https://aws.amazon.com/s3/pricing/>

Domain

Design High-Performing Architectures

Question 57 Correct

A data analytics company measures what the consumers watch and what advertising they're exposed to. This real-time data is ingested into its on-premises data center and subsequently, the daily data feed is compressed into a single file and uploaded on Amazon S3 for backup. The typical compressed file size is around 2 gigabytes.

Which of the following is the fastest way to upload the daily compressed file into Amazon S3?

Upload the compressed file in a single operation

Your answer is correct

Upload the compressed file using multipart upload with Amazon S3 Transfer Acceleration (Amazon S3TA)

Upload the compressed file using multipart upload

FTP the compressed file into an Amazon EC2 instance that runs in the same region as the Amazon S3 bucket. Then transfer the file from the Amazon EC2 instance into the Amazon S3 bucket

Overall explanation

Correct option:

Upload the compressed file using multipart upload with Amazon S3 Transfer Acceleration (Amazon S3TA)

Amazon S3 Transfer Acceleration (Amazon S3TA) enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. Transfer Acceleration takes

advantage of Amazon CloudFront's globally distributed edge locations. As the data arrives at an edge location, data is routed to Amazon S3 over an optimized network path.

Multipart upload allows you to upload a single object as a set of parts. Each part is a contiguous portion of the object's data. You can upload these object parts independently and in any order. If transmission of any part fails, you can retransmit that part without affecting other parts. After all parts of your object are uploaded, Amazon S3 assembles these parts and creates the object. If you're uploading large objects over a stable high-bandwidth network, use multipart uploading to maximize the use of your available bandwidth by uploading object parts in parallel for multi-threaded performance. If you're uploading over a spotty network, use multipart uploading to increase resiliency to network errors by avoiding upload restarts.

Incorrect options:

Upload the compressed file in a single operation - In general, when your object size reaches 100 megabytes, you should consider using multipart uploads instead of uploading the object in a single operation. Multipart upload provides improved throughput - you can upload parts in parallel to improve throughput. Therefore, this option is not correct.

Upload the compressed file using multipart upload - Although using multipart upload would certainly speed up the process, combining with Amazon S3 Transfer Acceleration (Amazon S3TA) would further improve the transfer speed. Therefore just using multipart upload is not the correct option.

FTP the compressed file into an Amazon EC2 instance that runs in the same region as the Amazon S3 bucket. Then transfer the file from the Amazon EC2 instance into the Amazon S3 bucket - This is a roundabout process of getting the file into Amazon S3 and added as a distractor. Although it is technically feasible to follow this process, it would involve a lot of scripting and certainly would not be the fastest way to get the file into Amazon S3.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/transfer-acceleration.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/uploadobjusingmpu.html>

Domain

Design High-Performing Architectures

Question 58 Incorrect

A software engineering intern at an e-commerce company is documenting the process flow to provision Amazon EC2 instances via the Amazon EC2 API. These instances are to be used for an internal application that processes Human Resources payroll data. He wants to highlight those volume types that cannot be used as a boot volume.

Can you help the intern by identifying those storage volume types that CANNOT be used as boot volumes while creating the instances? (Select two)

Correct selection

Throughput Optimized Hard disk drive (st1)

Your selection is correct

Cold Hard disk drive (sc1)

General Purpose Solid State Drive (gp2)

Instance Store

Your selection is incorrect

Overall explanation

Correct options:

Throughput Optimized Hard disk drive (st1)

Cold Hard disk drive (sc1)

The Amazon EBS volume types fall into two categories:

Solid state drive (SSD) backed volumes optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS.

Hard disk drive (HDD) backed volumes optimized for large streaming workloads where throughput (measured in MiB/s) is a better performance measure than IOPS.

Throughput Optimized HDD (st1) and Cold HDD (sc1) volume types CANNOT be used as a boot volume, so these two options are correct.

Please see this detailed overview of the volume types for Amazon EBS volumes.

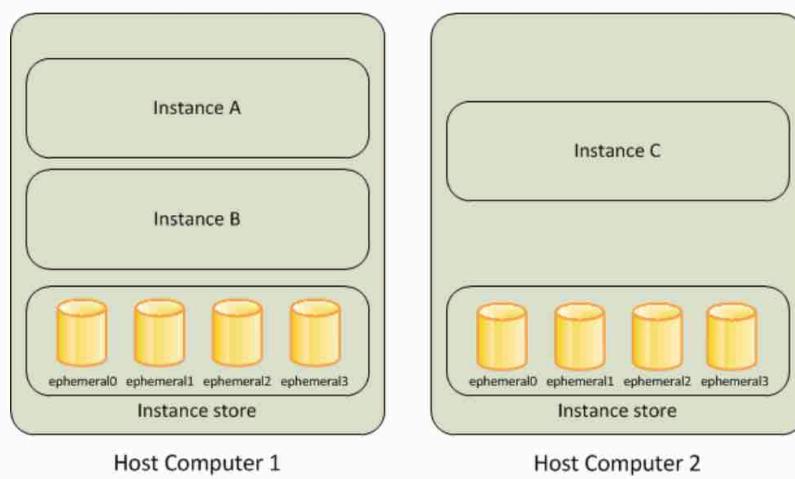
Amazon EC2 Instance Store

[PDF](#) | [Kindle](#) | [RSS](#)

An *instance store* provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

An instance store consists of one or more instance store volumes exposed as block devices. The size of an instance store as well as the number of devices available varies by instance type.

The virtual devices for instance store volumes are `ephemeral[0-23]`. Instance types that support one instance store volume have `ephemeral0`. Instance types that support two instance store volumes have `ephemeral0` and `ephemeral1`, and so on.



via - <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html>

Incorrect options:

General Purpose Solid State Drive (gp2)

Provisioned IOPS Solid state drive (io1)

Instance Store

General Purpose SSD (gp2), Provisioned IOPS SSD (io1), and Instance Store can be used as a boot volume.

References:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/RootDeviceStorage.html>

Domain

Design High-Performing Architectures

Question 59 Incorrect

The solo founder at a tech startup has just created a brand new AWS account. The founder has provisioned an Amazon EC2 instance 1A which is running in AWS Region A. Later, he takes a snapshot of the instance 1A and then creates a new Amazon Machine Image (AMI) in Region A from this snapshot. This AMI is then copied into another Region B. The founder provisions an instance 1B in Region B using this new AMI in Region B.

At this point in time, what entities exist in Region B?

Correct answer

1 Amazon EC2 instance, 1 AMI and 1 snapshot exist in Region B

Your answer is incorrect

1 Amazon EC2 instance and 1 AMI exist in Region B

1 Amazon EC2 instance and 2 AMIs exist in Region B

1 Amazon EC2 instance and 1 snapshot exist in Region B

Overall explanation

Correct option:

1 Amazon EC2 instance, 1 AMI and 1 snapshot exist in Region B

An Amazon Machine Image (AMI) provides the information required to launch an instance. You must specify an AMI when you launch an instance. When the new AMI is copied from Region A into Region B, it automatically creates a snapshot in Region B because AMIs are based on the underlying snapshots. Further, an instance is created from this AMI in Region B. Hence, we have 1 Amazon EC2 instance, 1 AMI and 1 snapshot in Region B.

Amazon Machine Image (AMI) Overview:

Amazon Machine Images (AMI)

[PDF](#) | [Kindle](#) | [RSS](#)

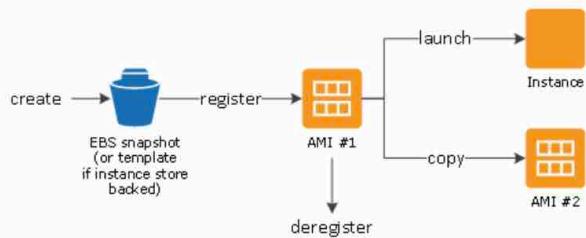
An Amazon Machine Image (AMI) provides the information required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. You can use different AMIs to launch instances when you need instances with different configurations.

An AMI includes the following:

- One or more EBS snapshots, or, for instance-store-backed AMIs, a template for the root volume of the instance (for example, an operating system, an application server, and applications).
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

Using an AMI

The following diagram summarizes the AMI lifecycle. After you create and register an AMI, you can use it to launch new instances. (You can also launch instances from an AMI if the AMI owner grants you launch permissions.) You can copy an AMI within the same Region or to different Regions. When you no longer require an AMI, you can deregister it.



via - <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html>

Incorrect options:

1 Amazon EC2 instance and 1 AMI exist in Region B

1 Amazon EC2 instance and 1 snapshot exist in Region B

1 Amazon EC2 instance and 2 AMIs exist in Region B

As mentioned earlier in the explanation, when the new AMI is copied from Region A into Region B, it also creates a snapshot in Region B because AMIs are based on the underlying snapshots. In addition, an instance is created from this AMI in Region B. So, we have 1 Amazon EC2 instance, 1 AMI and 1 snapshot in Region B. Hence all three options are incorrect.

Reference:

Domain

Design High-Performing Architectures

Question 60 Correct

The engineering team at a Spanish professional football club has built a notification system for its website using Amazon Simple Notification Service (Amazon SNS) notifications which are then handled by an AWS Lambda function for end-user delivery. During the off-season, the notification systems need to handle about 100 requests per second. During the peak football season, the rate touches about 5000 requests per second and it is noticed that a significant number of the notifications are not being delivered to the end-users on the website.

As a solutions architect, which of the following would you suggest as the BEST possible solution to this issue?

The engineering team needs to provision more servers running the Amazon SNS service

Amazon SNS has hit a scalability limit, so the team needs to contact AWS support to raise the account limit

Your answer is correct

Amazon SNS message deliveries to AWS Lambda have crossed the account concurrency quota for AWS Lambda, so the team needs to contact AWS support to raise the account limit

The engineering team needs to provision more servers running the AWS Lambda service

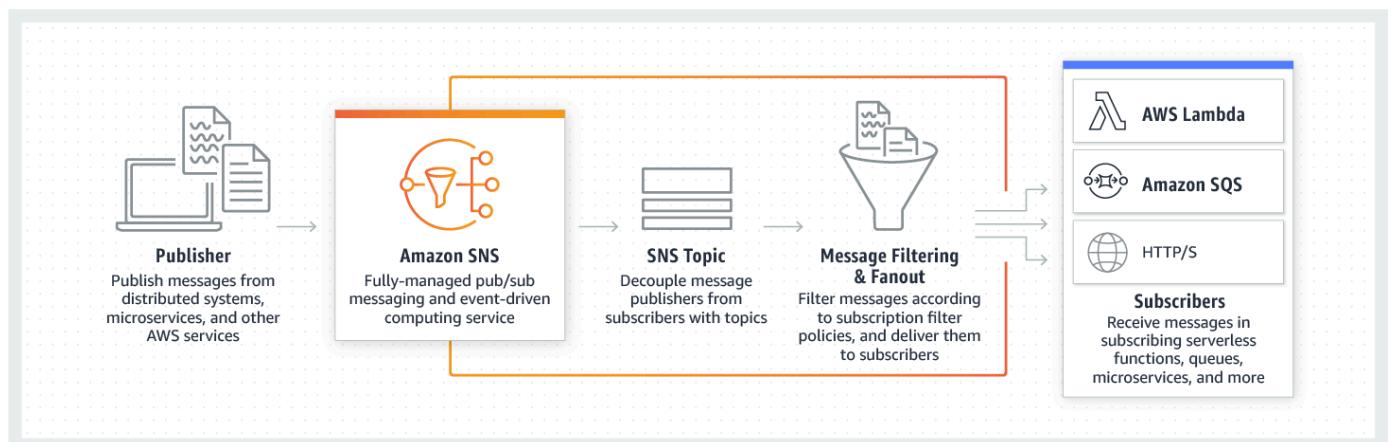
Overall explanation

Correct option:

Amazon SNS message deliveries to AWS Lambda have crossed the account concurrency quota for AWS Lambda, so the team needs to contact AWS support to raise the account limit

Amazon Simple Notification Service (Amazon SNS) is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications.

How Amazon SNS Works:



via - <https://aws.amazon.com/sns/>

With AWS Lambda, you can run code without provisioning or managing servers. You pay only for the compute time that you consume—there's no charge when your code isn't running.

AWS Lambda currently supports 1000 concurrent executions per AWS account per region. If your Amazon SNS message deliveries to AWS Lambda contribute to crossing these concurrency quotas, your Amazon SNS message deliveries will be throttled. You need to contact AWS support to raise the account limit. Therefore this option is correct.

Incorrect options:

Amazon SNS has hit a scalability limit, so the team needs to contact AWS support to raise the account limit - Amazon SNS leverages the proven AWS cloud to dynamically

scale with your application. You don't need to contact AWS support, as SNS is a fully managed service, taking care of the heavy lifting related to capacity planning, provisioning, monitoring, and patching. Therefore, this option is incorrect.

The engineering team needs to provision more servers running the Amazon SNS service

The engineering team needs to provision more servers running the AWS Lambda service

As both AWS Lambda and Amazon SNS are serverless and fully managed services, the engineering team cannot provision more servers. Both of these options are incorrect.

References:

<https://aws.amazon.com/sns/>

<https://aws.amazon.com/sns/faqs/>

Domain

Design Resilient Architectures

Question 61 Correct

A leading social media analytics company is contemplating moving its dockerized application stack into AWS Cloud. The company is not sure about the pricing for using Amazon Elastic Container Service (Amazon ECS) with the EC2 launch type compared to the Amazon Elastic Container Service (Amazon ECS) with the Fargate launch type.

Which of the following is correct regarding the pricing for these two services?

Your answer is correct

Amazon ECS with EC2 launch type is charged based on EC2 instances and EBS volumes used. Amazon ECS with Fargate launch type is charged based on vCPU and memory resources that the containerized application requests

Both Amazon ECS with EC2 launch type and Amazon ECS with Fargate launch type are charged based on Amazon EC2 instances and Amazon EBS Elastic Volumes used

Both Amazon ECS with EC2 launch type and Amazon ECS with Fargate launch type are charged based on vCPU and memory resources that the containerized application requests

Both Amazon ECS with EC2 launch type and Amazon ECS with Fargate launch type are just charged based on Elastic Container Service used per hour

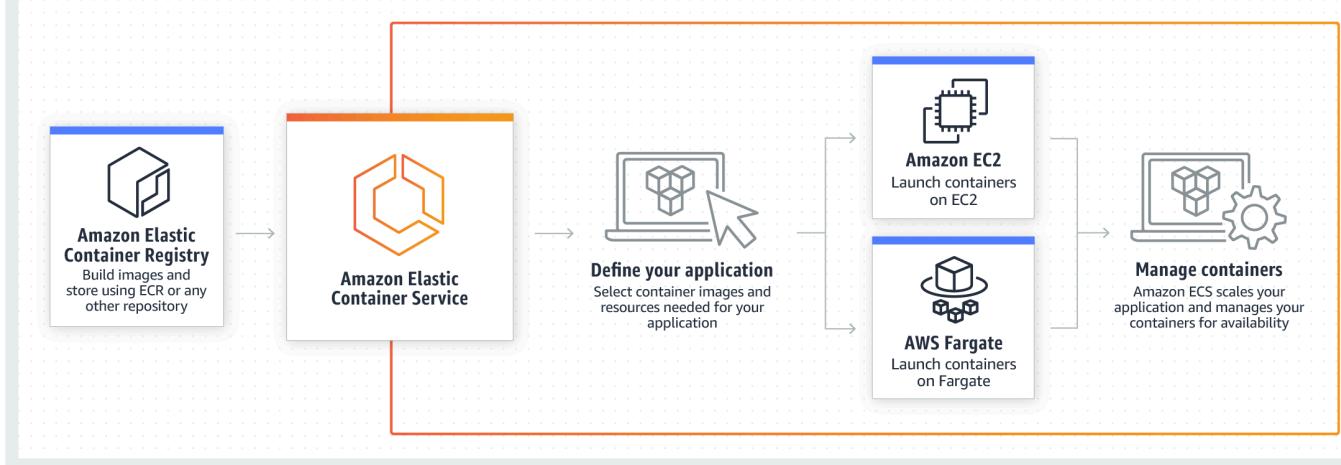
Overall explanation

Correct option:

Amazon ECS with EC2 launch type is charged based on EC2 instances and EBS volumes used. Amazon ECS with Fargate launch type is charged based on vCPU and memory resources that the containerized application requests

Amazon Elastic Container Service (Amazon ECS) is a fully managed container orchestration service. ECS allows you to easily run, scale, and secure Docker container applications on AWS.

Amazon ECS Overview:



via - <https://aws.amazon.com/ecs/>

With the Fargate launch type, you pay for the amount of vCPU and memory resources that your containerized application requests. vCPU and memory resources are calculated from the time your container images are pulled until the Amazon ECS Task terminates, rounded up to the nearest second. With the EC2 launch type, there is no additional charge for the EC2 launch type. You pay for AWS resources (e.g. EC2 instances or EBS volumes) you create to store and run your application.

Incorrect options:

Both Amazon ECS with EC2 launch type and Amazon ECS with Fargate launch type are charged based on vCPU and memory resources that the containerized application requests

Both Amazon ECS with EC2 launch type and Amazon ECS with Fargate launch type are charged based on Amazon EC2 instances and Amazon EBS Elastic Volumes used

As mentioned above - with the Fargate launch type, you pay for the amount of vCPU and memory resources. With EC2 launch type, you pay for AWS resources (e.g. EC2 instances or EBS volumes). Hence both these options are incorrect.

Both Amazon ECS with EC2 launch type and Amazon ECS with Fargate launch type are just charged based on Elastic Container Service used per hour

This is a made-up option and has been added as a distractor.

References:

<https://aws.amazon.com/ecs/pricing/>

Domain

Design Cost-Optimized Architectures

Question 62 Incorrect

An organization wants to delegate access to a set of users from the development environment so that they can access some resources in the production environment which is managed under another AWS account.

As a solutions architect, which of the following steps would you recommend?

Correct answer

Create a new IAM role with the required permissions to access the resources in the production environment. The users can then assume this IAM role while accessing the resources from the production environment

Your answer is incorrect

It is not possible to access cross-account resources

Create new IAM user credentials for the production environment and share these credentials with the set of users from the development environment

Both IAM roles and IAM users can be used interchangeably for cross-account access

Overall explanation

Correct option:

Create a new IAM role with the required permissions to access the resources in the production environment. The users can then assume this IAM role while accessing the resources from the production environment

IAM roles allow you to delegate access to users or services that normally don't have access to your organization's AWS resources. IAM users or AWS services can assume a role to obtain temporary security credentials that can be used to make AWS API calls. Consequently, you don't have to share long-term credentials for access to a resource. Using IAM roles, it is possible to access cross-account resources.

Incorrect options:

Create new IAM user credentials for the production environment and share these credentials with the set of users from the development environment - There is no need to create new IAM user credentials for the production environment, as you can use IAM roles to access cross-account resources.

It is not possible to access cross-account resources - You can use IAM roles to access cross-account resources.

Both IAM roles and IAM users can be used interchangeably for cross-account access - IAM roles and IAM users are separate IAM entities and should not be mixed. Only IAM roles can be used to access cross-account resources.

Reference:

<https://aws.amazon.com/iam/features/manage-roles/>

Domain

Design Secure Architectures

Question 63 Incorrect

The DevOps team at an e-commerce company has deployed a fleet of Amazon EC2 instances under an Auto Scaling group (ASG). The instances under the ASG span two Availability Zones (AZ) within the `us-east-1` region. All the incoming requests are handled by an Application Load Balancer (ALB) that routes the requests to the Amazon EC2 instances under the Auto Scaling Group. As part of a test run, two instances (instance 1 and 2, belonging to AZ A) were manually terminated by the DevOps team causing the Availability Zones (AZ) to have unbalanced resources. Later that day, another instance (belonging to AZ B) was detected as unhealthy by the Application Load Balancer's health check.

Can you identify the correct outcomes for these events? (Select two)

Amazon EC2 Auto Scaling creates a new scaling activity for launching a new instance to replace the unhealthy instance. Later, Amazon EC2 Auto Scaling creates a new scaling activity for terminating the unhealthy instance and then terminates it

Correct selection

Amazon EC2 Auto Scaling creates a new scaling activity for terminating the unhealthy instance and then terminates it. Later, another scaling activity launches a new instance to replace the terminated instance

Correct selection

As the resources are unbalanced in the Availability Zones, Amazon EC2 Auto Scaling will compensate by rebalancing the Availability Zones. When rebalancing, Amazon EC2 Auto Scaling launches new instances before terminating the old ones, so that rebalancing does not compromise the performance or availability of your application

Your selection is incorrect

As the resources are unbalanced in the Availability Zones, Amazon EC2 Auto Scaling will compensate by rebalancing the Availability Zones. When rebalancing, Amazon EC2 Auto Scaling terminates old instances before launching new instances, so that rebalancing does not cause extra instances to be launched

Amazon EC2 Auto Scaling creates a new scaling activity to terminate the unhealthy instance and launch the new instance simultaneously

Overall explanation

Correct options:

As the resources are unbalanced in the Availability Zones, Amazon EC2 Auto Scaling will compensate by rebalancing the Availability Zones. When rebalancing, Amazon EC2 Auto Scaling launches new instances before terminating the old ones, so that rebalancing does not compromise the performance or availability of your application

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called Auto Scaling groups. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. Actions such as changing the Availability Zones (AZ) for your group or explicitly terminating or detaching instances can lead to the Auto Scaling group becoming unbalanced between Availability Zones. Amazon EC2 Auto Scaling compensates by rebalancing the Availability Zones.

When rebalancing, Amazon EC2 Auto Scaling launches new instances before terminating the old ones, so that rebalancing does not compromise the performance or availability of your application. Therefore, this option is correct.

Availability Zone Rebalancing Overview:

Rebalancing Activities

After certain actions occur, your Auto Scaling group can become unbalanced between Availability Zones. Amazon EC2 Auto Scaling compensates by rebalancing the Availability Zones. The following actions can lead to rebalancing activity:

- You change the Availability Zones for your group.
- You explicitly terminate or detach instances and the group becomes unbalanced.
- An Availability Zone that previously had insufficient capacity recovers and has additional capacity available.
- An Availability Zone that previously had a Spot price above your maximum price now has a Spot price below your maximum price.

When rebalancing, Amazon EC2 Auto Scaling launches new instances before terminating the old ones, so that rebalancing does not compromise the performance or availability of your application.

Because Amazon EC2 Auto Scaling attempts to launch new instances before terminating the old ones, being at or near the specified maximum capacity could impede or completely halt rebalancing activities. To avoid this problem, the system can temporarily exceed the specified maximum capacity of a group by a 10 percent margin (or by a 1-instance margin, whichever is greater) during a rebalancing activity. The margin is extended only if the group is at or near maximum capacity and needs rebalancing, either because of user-requested rezoning or to compensate for zone availability issues. The extension lasts only as long as needed to rebalance the group typically a few minutes.

via - <https://docs.aws.amazon.com/autoscaling/ec2/userguide/auto-scaling-benefits.html>

Amazon EC2 Auto Scaling creates a new scaling activity for terminating the unhealthy instance and then terminates it. Later, another scaling activity launches a new instance to replace the terminated instance

However, the scaling activity of Auto Scaling works in a different sequence compared to the rebalancing activity. Auto Scaling creates a new scaling activity for terminating the unhealthy instance and then terminates it. Later, another scaling activity launches a new instance to replace the terminated instance.

Incorrect options:

Amazon EC2 Auto Scaling creates a new scaling activity for launching a new instance to replace the unhealthy instance. Later, Amazon EC2 Auto Scaling creates a new scaling activity for terminating the unhealthy instance and then terminates it - This option contradicts the correct sequence of events outlined earlier for scaling activity created by Amazon EC2 Auto Scaling. Actually, Auto Scaling first terminates the unhealthy instance and then launches a new instance. Hence this is incorrect.

As the resources are unbalanced in the Availability Zones, Amazon EC2 Auto Scaling will compensate by rebalancing the Availability Zones. When rebalancing, Amazon EC2 Auto Scaling terminates old instances before launching new instances, so that rebalancing does not cause extra instances to be launched - This option contradicts the correct sequence of events outlined earlier for rebalancing activity. When rebalancing, Amazon EC2 Auto Scaling launches new instances before terminating the old ones. Hence this is incorrect.

Amazon EC2 Auto Scaling creates a new scaling activity to terminate the unhealthy instance and launch the new instance simultaneously - This is a made-up option as both the terminate and launch activities can't happen simultaneously. This option has been added as a distractor.

References:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/auto-scaling-benefits.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-instance-termination.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/healthcheck.html>

Domain

Design High-Performing Architectures

Question 64 Correct

A retail analytics company operates a large-scale data lake on Amazon S3, where they store daily logs of customer transactions, product views, and inventory updates. Each morning, they need to transform and load the data into a data warehouse to support fast analytical queries. The company also wants to enable data analysts to build and train machine learning (ML) models using familiar SQL syntax without writing custom Python code. The architecture must support massively parallel processing (MPP) for fast data aggregation and scoring, and must use serverless AWS services wherever possible to reduce infrastructure management and operational overhead.

Which solution best meets these requirements?

Use an AWS Glue job to transform and load data into Amazon RDS for PostgreSQL. Allow analysts to run machine learning models using Amazon Aurora ML integrated with PostgreSQL, leveraging Amazon SageMaker endpoints behind the scenes

Run a daily AWS Glue job to process and transform the raw files in S3 and register the outputs as Amazon Athena tables in AWS Glue Data Catalog. Allow analysts to build ML models using Amazon Athena ML, with SQL-based predictions on top of S3 data without moving it to a warehouse

Your answer is correct

Use a daily AWS Glue job to transform and clean the data stored in Amazon S3. Load the transformed dataset into Amazon Redshift Serverless, which offers MPP capabilities in a serverless model. Enable analysts to use Amazon Redshift ML to build and train ML models

Provision and run a daily Amazon EMR cluster with Apache Spark to process and transform the S3 data. Load the results into Amazon Redshift (provisioned). Enable ML model development by integrating Redshift with Amazon SageMaker notebooks for advanced modeling tasks

Overall explanation

Correct option:

Use a daily AWS Glue job to transform and clean the data stored in Amazon S3. Load the transformed dataset into Amazon Redshift Serverless, which offers MPP capabilities in a serverless model. Enable analysts to use Amazon Redshift ML to build and train ML models

This solution is fully serverless and optimized for high-performance analytics and ML. AWS Glue provides serverless ETL, and Amazon Redshift Serverless offers a MPP data warehouse that scales compute independently. Amazon Redshift ML lets analysts create, train, and deploy ML models using SQL by integrating with Amazon SageMaker under the hood — abstracting away the complexity of building models. This architecture meets all core requirements: serverless, SQL-based ML, and scalable MPP analytics.

Create, train, and deploy machine learning models in Amazon Redshift using SQL with Amazon Redshift ML

by Debu Panda, Phil Bates, Murali Balakrishnan Narayanaswamy, Rohit Bansal, Sriram Krishnamurthy, Stefano Stefani, Sudipta Sengupta, and Yannis Papakonstantinou | on 08 DEC 2020 | in [Amazon Redshift](#), [AWS Big Data](#) | [Permalink](#) | [Comments](#) | [Share](#)

December 2022: Post was reviewed and updated to announce support of Prediction Probabilities for Classification problems using Amazon Redshift ML.

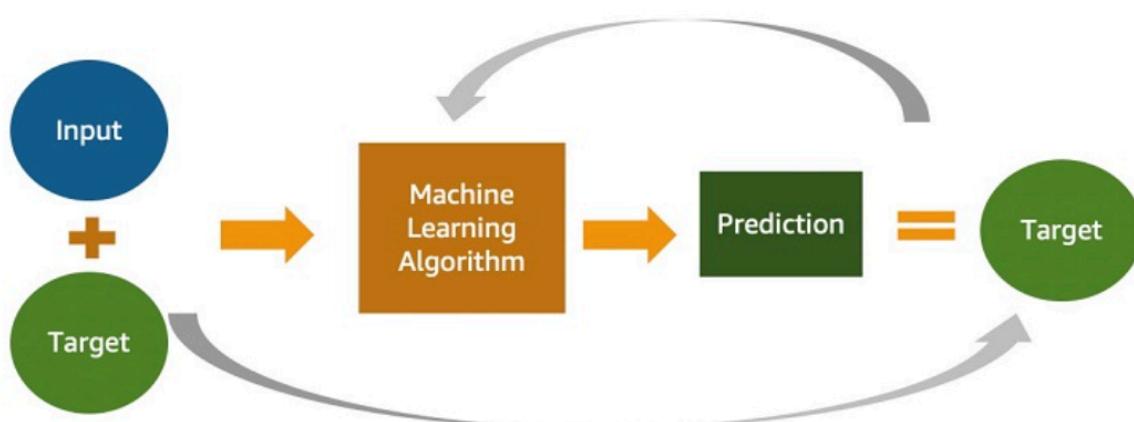
[Amazon Redshift](#) is a fast, petabyte-scale cloud data warehouse data warehouse delivering the best price–performance. Tens of thousands of customers use Amazon Redshift to process exabytes of data every day to power their analytics workloads. Data analysts and database developers want to use this data to train machine learning (ML) models, which can then be used to generate insights on new data for use cases such as forecasting revenue, predicting customer churn, and detecting anomalies.

[Amazon Redshift ML](#) makes it easy for SQL users to create, train, and deploy ML models using familiar SQL commands. Redshift ML allows you to use your data in Amazon Redshift with [Amazon SageMaker](#), a fully managed ML service, without requiring you to become an expert in ML.

This post shows you how to use familiar SQL statements to create and train ML models from data in Amazon Redshift and use these models to make in-database predictions on new data for use cases such as churn prediction and fraud risk scoring.

ML use cases relevant to data warehousing

You may use different ML approaches according to what's relevant for your business, such as supervised, unsupervised, and reinforcement learning. With this release, Redshift ML supports supervised learning, which is most commonly used in enterprises for advanced analytics. As evident in the following diagram, supervised learning is preferred when you have a training dataset and an understanding of how specific input data predicts various business outcomes. The inputs used for the ML model are often referred to as *features*, and the outcomes or results are called *targets* or *labels*. Your training dataset is a table or a query whose attributes or columns comprise features, and targets are extracted from your data warehouse. The following diagram illustrates this architecture.



Incorrect options:

Provision and run a daily Amazon EMR cluster with Apache Spark to process and transform the S3 data. Load the results into Amazon Redshift (provisioned). Enable ML model development by integrating Redshift with Amazon SageMaker notebooks for advanced modeling tasks - While EMR is suitable for large-scale transformation using Spark and Redshift supports MPP, this solution requires managing both EMR clusters and Redshift clusters. Additionally, ML modeling via SageMaker notebooks requires Python coding skills and breaks the requirement for SQL-based ML. This option introduces higher operational overhead and deviates from the serverless and SQL-only requirements.

Run a daily AWS Glue job to process and transform the raw files in S3 and register the outputs as Amazon Athena tables in AWS Glue Data Catalog. Allow analysts to build ML models using Amazon Athena ML, with SQL-based predictions on top of S3 data without moving it to a warehouse - Although AWS Glue offers serverless ETL and RDS for PostgreSQL is widely used, RDS is not a MPP data warehouse, and performance may not scale well for large, concurrent analytical queries. While Aurora ML allows calling SageMaker models from SQL, it's designed for transactional workloads, not large-scale ML workflows. This approach doesn't align with the analytics and MPP scale required by the company.

Use an AWS Glue job to transform and load data into Amazon RDS for PostgreSQL. Allow analysts to run machine learning models using Amazon Aurora ML integrated with PostgreSQL, leveraging Amazon SageMaker endpoints behind the scenes - Amazon Athena is serverless and can run SQL queries on S3-based data using the Glue Data Catalog, and Athena ML allows applying SageMaker models to query outputs. However, Athena does not support MPP-level performance for large-scale data warehousing workloads and is primarily designed for ad hoc querying. Additionally, Athena ML is limited to inference only, not for training ML models via SQL.

References:

<https://aws.amazon.com/blogs/big-data/create-train-and-deploy-machine-learning-models-in-amazon-redshift-using-sql-with-amazon-redshift-ml/>

<https://docs.aws.amazon.com/redshift/latest/mgmt/serverless-whatis.html>

<https://docs.aws.amazon.com/glue/latest/dg/what-is-glue.html>

<https://github.com/awslabs/aws-athena-query-federation/wiki/AthenaML-Tutorial>

Domain

Design High-Performing Architectures

Question 65 Correct

A technology blogger wants to write a review on the comparative pricing for various storage types available on AWS Cloud. The blogger has created a test file of size 1 gigabytes with some random data. Next he copies this test file into AWS S3 Standard storage class, provisions an Amazon EBS volume (General Purpose SSD (gp2)) with 100 gigabytes of provisioned storage and copies the test file into the Amazon EBS volume, and lastly copies the test file into an Amazon EFS Standard Storage filesystem. At the end of the month, he analyses the bill for costs incurred on the respective storage types for the test file.

What is the correct order of the storage charges incurred for the test file on these three storage types?

Cost of test file storage on Amazon S3 Standard < Cost of test file storage on Amazon EBS < Cost of test file storage on Amazon EFS

Cost of test file storage on Amazon EBS < Cost of test file storage on Amazon S3 Standard < Cost of test file storage on Amazon EFS

Cost of test file storage on Amazon EFS < Cost of test file storage on Amazon S3 Standard < Cost of test file storage on Amazon EBS

Your answer is correct

Cost of test file storage on Amazon S3 Standard < Cost of test file storage on Amazon EFS < Cost of test file storage on Amazon EBS

Overall explanation

Correct option:

Cost of test file storage on Amazon S3 Standard < Cost of test file storage on Amazon EFS < Cost of test file storage on Amazon EBS

With Amazon EFS, you pay only for the resources that you use. The Amazon EFS Standard Storage pricing is \$0.30 per GB per month. Therefore the cost for storing the test file on EFS is \$0.30 for the month.

For Amazon EBS General Purpose SSD (gp2) volumes, the charges are \$0.10 per GB-month of provisioned storage. Therefore, for a provisioned storage of 100GB for this use-case, the monthly cost on EBS is $\$0.10 \times 100 = \10 . This cost is irrespective of how much storage is actually consumed by the test file.

For S3 Standard storage, the pricing is \$0.023 per GB per month. Therefore, the monthly storage cost on S3 for the test file is \$0.023.

Therefore this is the correct option.

Incorrect options:

Cost of test file storage on Amazon S3 Standard < Cost of test file storage on Amazon EBS < Cost of test file storage on Amazon EFS

Cost of test file storage on Amazon EFS < Cost of test file storage on Amazon S3 Standard < Cost of test file storage on Amazon EBS

Cost of test file storage on Amazon EBS < Cost of test file storage on Amazon S3 Standard < Cost of test file storage on Amazon EFS

Following the computations shown earlier in the explanation, these three options are incorrect.

References:

<https://aws.amazon.com/ebs/pricing/>

<https://aws.amazon.com/s3/pricing/> (<https://aws.amazon.com/s3/pricing/>)

<https://aws.amazon.com/efs/pricing/>

Domain

Design Cost-Optimized Architectures