Practice Test #1 - AWS Certified Solutions Architect Associate - Results

Return to review

Attempt 1

All knowledge areas

All questions

Question 1: **Skipped**

The development team at an e-commerce startup has set up multiple microservices running on 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?

- Host-based Routing
- Query string parameter-based routing
- Path-based Routing **(Correct)**
- HTTP header-based routing

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 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 `/img/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

Question 2: **Skipped**

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

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

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

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

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

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

Explanation

Correct option:

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

Exam Alert:

Please review this comparison vis-a-vis Multi-AZ vs Read Replica for 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 within a single region. Read replicas follow synchronous replication and can be within an Availability Zone, Cross-AZ, or Cross-Region

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

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

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

References:

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

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

Question 3: **Skipped**

As part of a pilot program, a biotechnology company wants to integrate data files from its on-premises analytical application with AWS Cloud via an NFS interface.

Which of the following AWS service is the MOST efficient solution for the given use-case?

- ◯ **AWS Storage Gateway - File Gateway** **(Correct)**
- ◯ **AWS Site-to-Site VPN**
- ◯ **AWS Storage Gateway - Volume Gateway**
- ◯ **AWS Storage Gateway - Tape Gateway**

Explanation

Correct option:

AWS Storage Gateway - File Gateway

AWS Storage Gateway is a hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage. The service provides three different types of gateways – Tape Gateway, File Gateway, and Volume Gateway – that seamlessly connect on-premises applications to cloud storage, caching data locally for low-latency access.

AWS Storage Gateway's file interface, or file gateway, offers you a seamless way to connect to the cloud in order to store application data files and backup images as durable objects on Amazon S3 cloud storage. File gateway offers SMB or NFS-based access to data in Amazon S3 with local caching. As the company wants to integrate data files from its analytical instruments into AWS via an NFS interface, therefore AWS Storage Gateway - File Gateway is the correct answer.

File Gateway Overview:  via - https://docs.aws.amazon.com/storagegateway/latest/userguide/StorageGatewayConcepts.html

Incorrect options:

AWS Storage Gateway - Volume Gateway - You can configure the AWS Storage Gateway service as a Volume Gateway to present cloud-based iSCSI block storage volumes to your on-premises applications. Volume Gateway does not support NFS interface, so this option is not correct.

AWS Storage Gateway - Tape Gateway - AWS Storage Gateway - Tape Gateway allows moving tape backups to the cloud. Tape Gateway does not support NFS interface, so this option is not correct.

AWS Site-to-Site VPN - 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 (Site-to-Site VPN) connection. It uses internet protocol security (IPSec) communications to create encrypted VPN tunnels between two locations. You cannot use AWS Site-to-Site VPN to integrate data files via the NFS interface, so this option is not correct.

References:

https://aws.amazon.com/storagegateway/

https://aws.amazon.com/storagegateway/volume/

https://aws.amazon.com/storagegateway/file/

https://aws.amazon.com/storagegateway/vtl/

Question 4: **Skipped**

Which of the following features of an Amazon S3 bucket can only be suspended once they have been enabled?

- ○ **Server Access Logging**
- ○ **Static Website Hosting**
- ○ **Requester Pays**
- ○ **Versioning** **(Correct)**

Explanation

Correct option:

Versioning

Once you version-enable a bucket, it can never return to an unversioned state. Versioning can only be suspended once it has been enabled.

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).

Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite. For example:

- 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. For more information, see Deleting object versions.
- If you overwrite an object, it results in a new object version in the bucket. You can always restore the previous version.

Important

If you have an object expiration lifecycle policy in your non-versioned bucket and you want to maintain the same permanent delete behavior when you enable versioning, you must add a noncurrent expiration policy. The noncurrent expiration lifecycle policy will manage the deletes of the noncurrent object versions in the version-enabled bucket. (A version-enabled bucket maintains one current and zero or more noncurrent object versions.) For more information, see How Do I Create a Lifecycle Policy for an S3 Bucket? in the *Amazon Simple Storage Service Console User Guide*.

Buckets can be in one of three states: unversioned (the default), versioning-enabled, or versioning-suspended.

Important

Once you version-enable a bucket, it can never return to an unversioned state. You can, however, suspend versioning on that bucket.

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

Incorrect options:

Server Access Logging

Static Website Hosting

Requester Pays

Server Access Logging, Static Website Hosting and Requester Pays features can be disabled even after they have been enabled.

Reference:

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

Question 5: **Skipped**

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?

◯ **Use VPC transit gateway to establish a connection between the data center and AWS Cloud**

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

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

○ **Use AWS Direct Connect plus VPN to establish a connection between the data center and AWS Cloud** **(Correct)**

Explanation

Correct option:

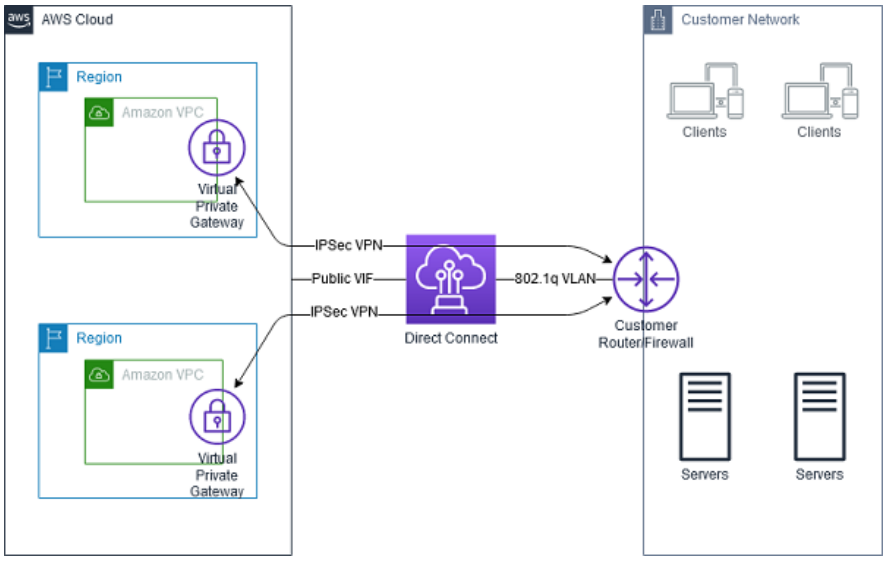
Use AWS Direct Connect plus VPN to establish a connection between the data center and AWS Cloud

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 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 VPC transit gateway to establish a connection between the data center and AWS Cloud - A transit gateway is a network transit hub that you can use to interconnect your virtual private clouds (VPC) and on-premises networks. A 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

Question 6: **Skipped**

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 Elastic Container Service (ECS) with the EC2 launch type compared to the Elastic Container Service (ECS) with the Fargate launch type.

Which of the following is correct regarding the pricing for these two services?

- ○ **Both ECS with EC2 launch type and ECS with Fargate launch type are charged based on vCPU and memory resources that the containerized application requests**
- ○ **Both ECS with EC2 launch type and ECS with Fargate launch type are charged based on EC2 instances and EBS volumes used**
- ○ **Both ECS with EC2 launch type and ECS with Fargate launch type are just charged based on Elastic Container Service used per hour**
- ○ **ECS with EC2 launch type is charged based on EC2 instances and EBS volumes used. ECS with Fargate launch type is charged based on vCPU and memory resources that the containerized application requests** **(Correct)**

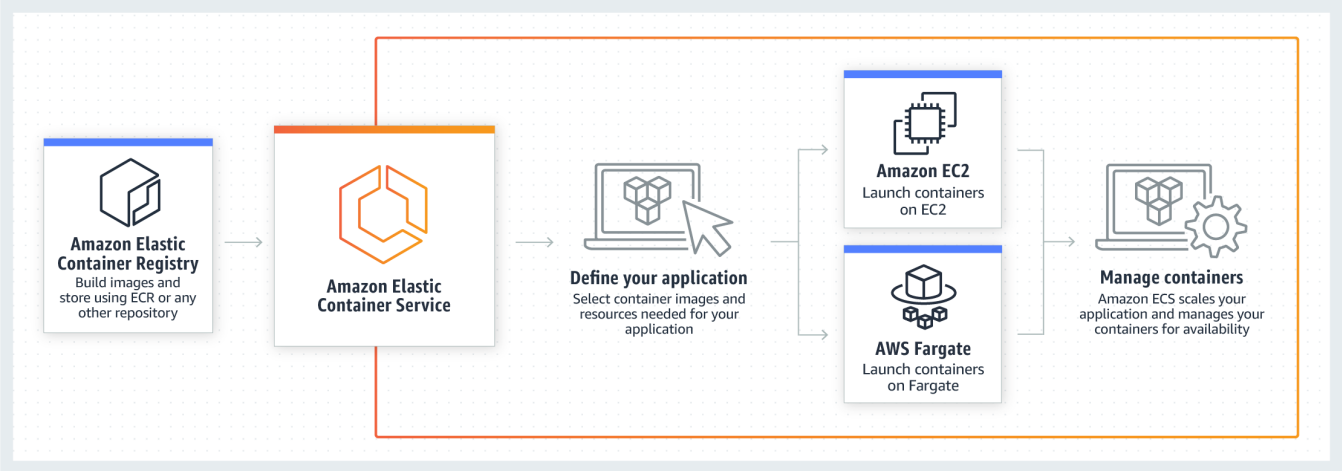
Explanation

Correct option:

ECS with EC2 launch type is charged based on EC2 instances and EBS volumes used. 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.

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 ECS with EC2 launch type and ECS with Fargate launch type are charged based on vCPU and memory resources that the containerized application requests

Both ECS with EC2 launch type and ECS with Fargate launch type are charged based on EC2 instances and EBS 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 ECS with EC2 launch type and 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/

Question 7: **Skipped**

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 EC2 spot instances to run the workflow processes** **(Correct)**

◯ **Use AWS Lambda function to run the workflow processes**

Use EC2 on-demand instances to run the workflow processes

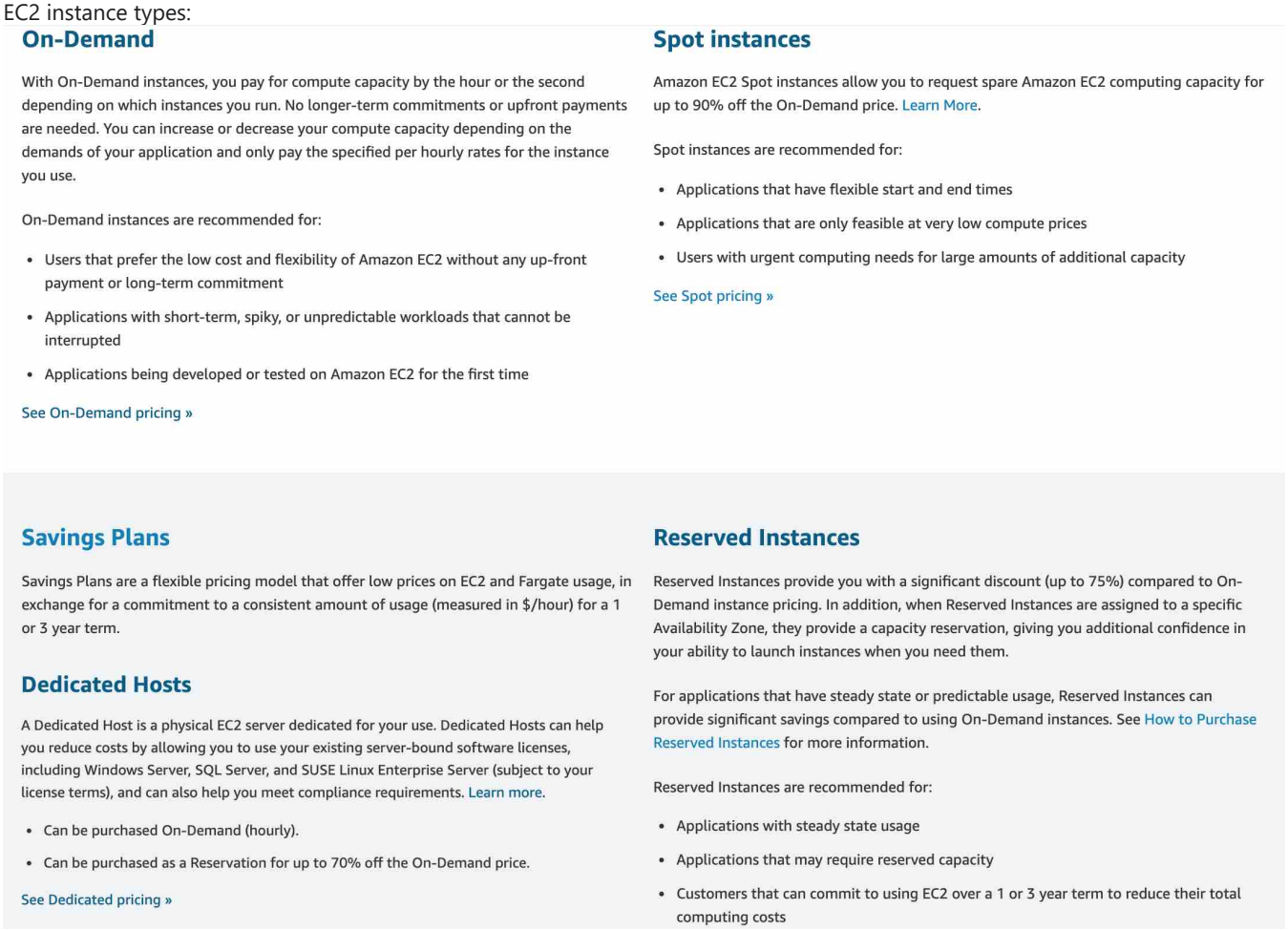
Use EC2 reserved instances to run the workflow processes

Explanation

Correct option:

Use EC2 spot instances to run the workflow processes

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).

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 EC2 on-demand instances to run the workflow processes

Use 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/

Question 8: **Skipped**

A news network uses 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 S3 bucket.

Which of the following are the MOST cost-effective options to improve the file upload speed into 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 S3

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

Create multiple 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 S3

Use multipart uploads for faster file uploads into the destination S3 bucket **(Correct)**

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

Explanation

Correct options:

Use Amazon S3 Transfer Acceleration 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. 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.

Use multipart uploads for faster file uploads into the destination 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 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 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 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 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

Question 9: **Skipped**

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?

- ○ **Migrate the website to Amazon S3. Use cross-Region replication between AWS Regions in the US and Asia**
- ○ **Use Amazon CloudFront with a custom origin pointing to the on-premises servers** **(Correct)**
- ○ **Leverage a Route 53 geo-proximity routing policy pointing to on-premises servers**
- ○ **Use Amazon CloudFront with a custom origin pointing to the DNS record of the website on Route 53**

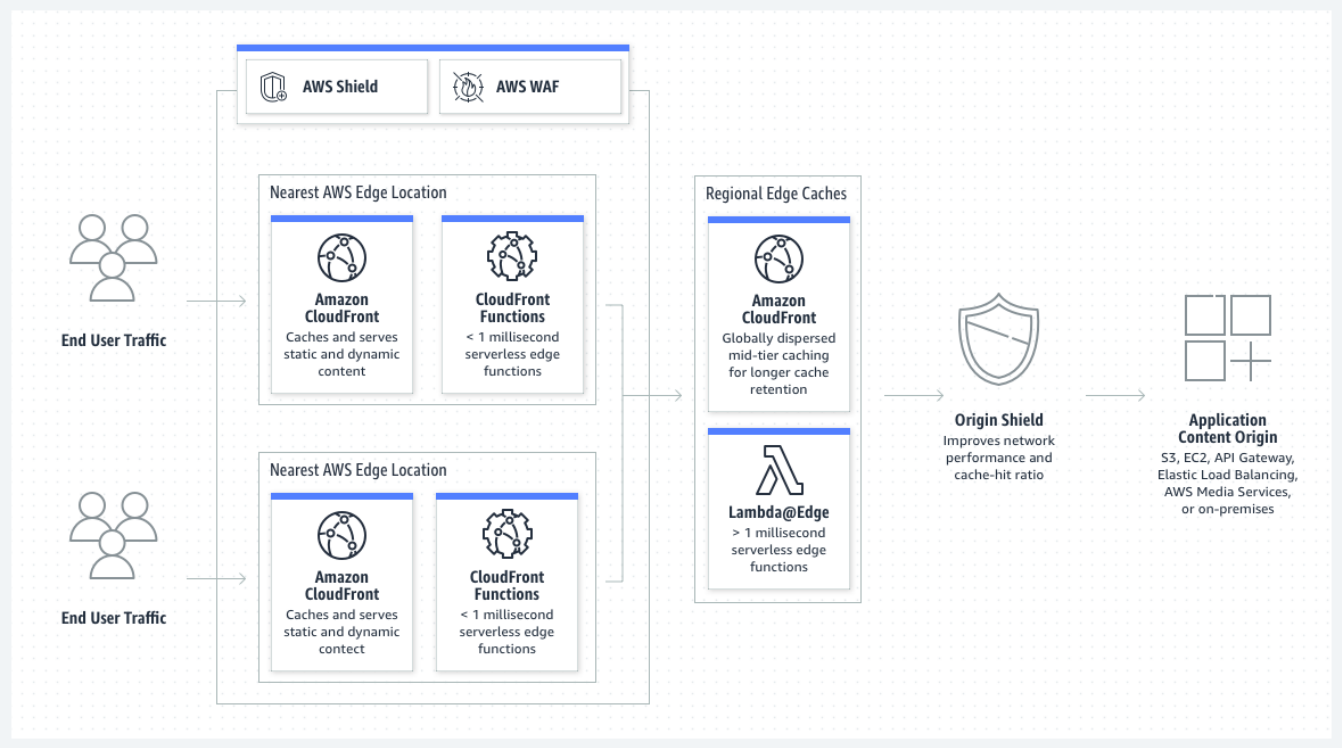
Explanation

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:

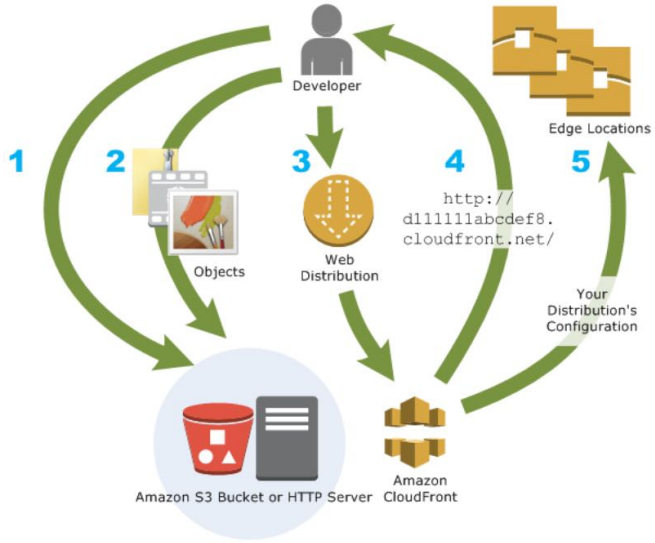


via - https://aws.amazon.com/cloudfront/

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 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 cross-Region replication 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 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

Question 10: **Skipped**

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 DynamoDB with DynamoDB Accelerator (DAX) as it meets the in-memory, high availability, low latency requirements **(Correct)**

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

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

Power the on-demand, live leaderboard using ElastiCache Redis as it meets the in-memory, high availability, low latency requirements **(Correct)**

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

Explanation

Correct options:

Power the on-demand, live leaderboard using ElastiCache 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.

ElastiCache for Redis Overview:

Power the on-demand, live leaderboard using 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 AWS 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 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 RDS 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/

Question 11: **Skipped**

A Big Data analytics company wants to set up an AWS cloud architecture that throttles requests in case of sudden traffic spikes. The company is looking for AWS services that can be used for buffering or throttling to handle such traffic variations.

Which of the following services can be used to support this requirement?

- **Amazon API Gateway, Amazon SQS and Amazon Kinesis** **(Correct)**
- **Amazon Gateway Endpoints, Amazon SQS and Amazon Kinesis**
- **Elastic Load Balancer, Amazon SQS, AWS Lambda**
- **Amazon SQS, Amazon SNS and AWS Lambda**

Explanation

Correct option:

Throttling is the process of limiting the number of requests an authorized program can submit to a given operation in a given amount of time.

Amazon API Gateway, Amazon SQS and Amazon Kinesis - To prevent your API from being overwhelmed by too many requests, Amazon API Gateway throttles requests to your API using the token bucket algorithm, where a token counts for a request. Specifically, API Gateway sets a limit on a steady-state rate and a burst of request submissions against all APIs in your account. In the token bucket algorithm, the burst is the maximum bucket size.

Amazon SQS - 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. Amazon SQS offers buffer capabilities to smooth out temporary volume spikes without losing messages or increasing latency.

Amazon Kinesis - Amazon Kinesis is a fully managed, scalable service that can ingest, buffer, and process streaming data in real-time.

Incorrect options:

Amazon SQS, Amazon SNS and AWS Lambda - Amazon SQS has the ability to buffer its messages. Amazon Simple Notification Service (SNS) cannot buffer messages and is generally used with SQS to provide the buffering facility. When requests come in faster than your Lambda function can scale, or when your function is at maximum concurrency, additional requests fail as the Lambda throttles those requests with error code 429 status code. So, this combination of services is incorrect.

Amazon Gateway Endpoints, Amazon SQS and Amazon Kinesis - A Gateway Endpoint is a gateway that you specify as a target for a route in your route table for traffic destined to a supported AWS service. This cannot help in throttling or buffering of requests. Amazon SQS and Kinesis can buffer incoming data. Since Gateway Endpoint is an incorrect service for throttling or buffering, this option is incorrect.

Elastic Load Balancer, Amazon SQS, AWS Lambda - Elastic Load Balancer cannot throttle requests. Amazon SQS can be used to buffer messages. When requests come in faster than your Lambda function can scale, or when your function is at maximum concurrency, additional requests fail as the Lambda throttles those requests with error code 429 status code. So, this combination of services is incorrect.

References:

https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-request-throttling.html

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

Question 12: **Skipped**

CloudFront offers a multi-tier cache in the form of regional edge caches that improve latency. However, there are certain content types that bypass the regional edge cache, and go directly to the origin.

Which of the following content types skip the regional edge cache? (Select two)

User-generated videos	
Proxy methods PUT/POST/PATCH/OPTIONS/DELETE go directly to the origin	**(Correct)**
Dynamic content, as determined at request time (cache-behavior configured to forward all headers)	**(Correct)**
E-commerce assets such as product photos	

Static content such as style sheets, JavaScript files

Explanation

Correct options:

Dynamic content, as determined at request time (cache-behavior configured to forward all headers)

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.

CloudFront points of presence (POPs) (edge locations) make sure that popular content can be served quickly to your viewers. CloudFront also has regional edge caches that bring more of your content closer to your viewers, even when the content is not popular enough to stay at a POP, to help improve performance for that content.

Dynamic content, as determined at request time (cache-behavior configured to forward all headers), does not flow through regional edge caches, but goes directly to the origin. So this option is correct.

Proxy methods PUT/POST/PATCH/OPTIONS/DELETE go directly to the origin

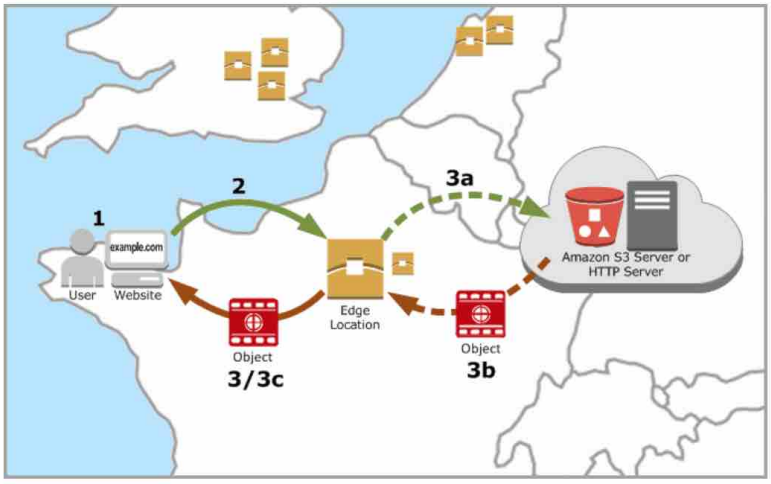
Proxy methods PUT/POST/PATCH/OPTIONS/DELETE go directly to the origin from the POPs and do not proxy through the regional edge caches. So this option is also correct.

How CloudFront Delivers Content:

How CloudFront Delivers Content to Your Users

After you configure CloudFront to deliver your content, here's what happens when users request your files:

1. A user accesses your website or application and requests one or more files, such as an image file and an HTML file.
2. DNS routes the request to the CloudFront POP (edge location) that can best serve the request—typically the nearest CloudFront POP in terms of latency—and routes the request to that edge location.
3. In the POP, CloudFront checks its cache for the requested files. If the files are in the cache, CloudFront returns them to the user. If the files are *not* in the cache, it does the following:
   a. CloudFront compares the request with the specifications in your distribution and forwards the request for the files to your origin server for the corresponding file type—for example, to your Amazon S3 bucket for image files and to your HTTP server for HTML files.
   b. The origin servers send the files back to the edge location.
   c. As soon as the first byte arrives from the origin, CloudFront begins to forward the files to the user. CloudFront also adds the files to the cache in the edge location for the next time someone requests those files.



via - https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/HowCloudFrontWorks.html

Incorrect Options:

E-commerce assets such as product photos

User-generated videos

Static content such as style sheets, JavaScript files

The following type of content flows through the regional edge caches - user-generated content, such as video, photos, or artwork; e-commerce assets such as product photos and videos and static content such as style sheets, JavaScript files. Hence these three options are not correct.

Reference:

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/HowCloudFrontWorks.html

Question 13: **Skipped**

A video analytics organization has been acquired by a leading media company. The analytics organization has 10 independent applications with an on-premises data footprint of about 70TB for each application. The CTO of the media company has set a timeline of two weeks to carry out the data migration from on-premises data center to AWS Cloud and establish connectivity.

Which of the following are the MOST cost-effective options for completing the data transfer and establishing connectivity? (Select two)

Order 10 Snowball Edge Storage Optimized devices to complete the one-time data transfer **(Correct)**

Order 70 Snowball Edge Storage Optimized devices to complete the one-time data transfer

Order 1 Snowmobile to complete the one-time data transfer

Setup AWS direct connect to establish connectivity between the on-premises data center and AWS Cloud

Setup Site-to-Site VPN to establish on-going connectivity between the on-premises data center and AWS Cloud **(Correct)**

Explanation

Correct options:

Order 10 Snowball Edge Storage Optimized devices to complete the one-time data transfer

Snowball Edge Storage Optimized is the optimal choice if you need to securely and quickly transfer dozens of terabytes to petabytes of data to AWS. It provides up to 80 TB of usable HDD storage, 40 vCPUs, 1 TB of SATA SSD storage, and up to 40 Gb network connectivity to address large scale data transfer and pre-processing use cases.

As each Snowball Edge Storage Optimized device can handle 80TB of data, you can order 10 such devices to take care of the data transfer for all applications.

Exam Alert:

The original Snowball devices were transitioned out of service and Snowball Edge Storage Optimized are now the primary devices used for data transfer. You may see the Snowball device on the exam, just remember that the original Snowball device had 80TB of storage space.

Setup Site-to-Site VPN to establish on-going connectivity between the on-premises 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). 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 can be configured in minutes and 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.

Therefore this option is the right fit for the given use-case as the connectivity can be easily established within the given timeframe.

Incorrect options:

Order 1 Snowmobile to complete the one-time data transfer - Each Snowmobile has a total capacity of up to 100 petabytes. To migrate large datasets of 10PB or more in a single location, you should use Snowmobile. For datasets less than 10PB or distributed in multiple locations, you should use Snowball. So Snowmobile is not the right fit for this use-case.

Setup AWS direct connect to establish connectivity between the on-premises data center and AWS Cloud - AWS Direct Connect lets you establish a dedicated network connection between your network and one of the AWS Direct Connect locations. Using industry-standard 802.1q VLANs, this dedicated connection can be partitioned into multiple virtual interfaces. AWS Direct Connect does not involve the Internet; instead, it uses dedicated, private network connections between your intranet and Amazon VPC. Direct Connect involves significant monetary investment and takes at least a month to set up, therefore it's not the correct fit for this use-case.

Order 70 Snowball Edge Storage Optimized devices to complete the one-time data transfer - As the data-transfer can be completed with just 10 Snowball Edge Storage Optimized devices, there is no need to order 70 devices.

References:

https://aws.amazon.com/snowball/faqs/

https://aws.amazon.com/vpn/

https://aws.amazon.com/snowmobile/faqs/

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

Question 14: **Skipped**

A retail company uses Amazon EC2 instances, API Gateway, Amazon RDS, Elastic Load Balancer and CloudFront services. To improve the security of these services, the Risk Advisory group has suggested a feasibility check for using the Amazon GuardDuty service.

Which of the following would you identify as data sources supported by GuardDuty?

- ○ **CloudFront logs, API Gateway logs, CloudTrail events**
- ○ **ELB logs, DNS logs, CloudTrail events**
- ○ **VPC Flow Logs, API Gateway logs, S3 access logs**
- ○ **VPC Flow Logs, DNS logs, CloudTrail events** **(Correct)**

Explanation

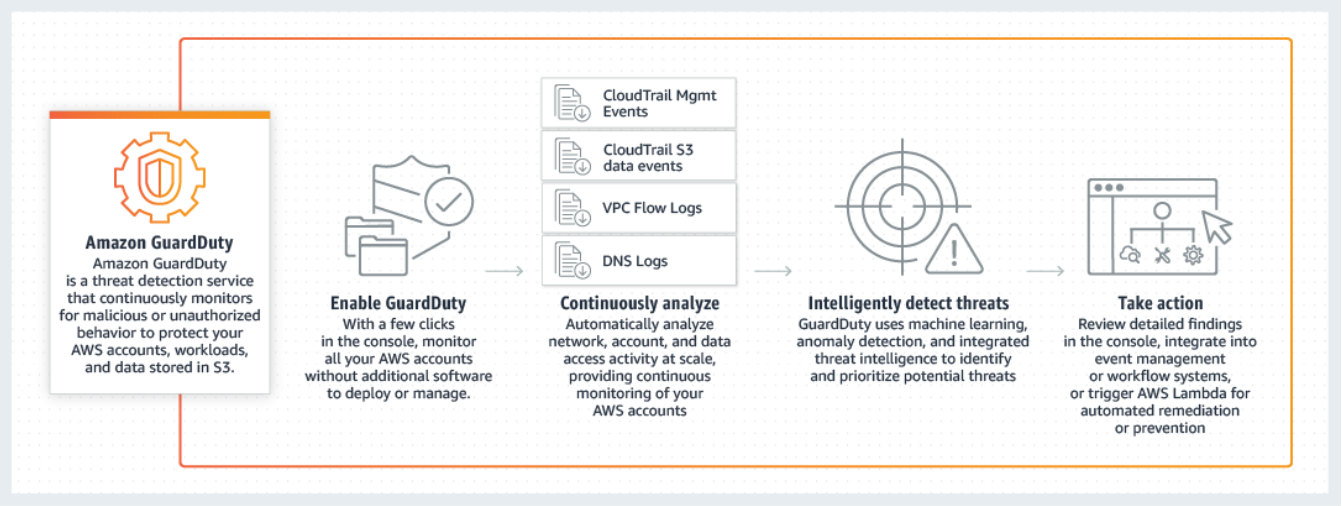
Correct option:

VPC Flow Logs, DNS logs, CloudTrail events - 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. With the cloud, the collection and aggregation of account and network activities is simplified, but it can be time-consuming for security teams to continuously analyze event log data for potential threats. With GuardDuty, you now have an intelligent and cost-effective option for continuous threat detection in AWS. The service uses machine learning, anomaly detection, and integrated threat intelligence to identify and prioritize potential threats.

GuardDuty analyzes tens of billions of events across multiple AWS data sources, such as AWS CloudTrail events, Amazon VPC Flow Logs, and DNS logs.

With a few clicks in the AWS Management Console, GuardDuty can be enabled with no software or hardware to deploy or maintain. By integrating with Amazon EventBridge Events, GuardDuty alerts are actionable, easy to aggregate across multiple accounts, and straightforward to push into existing event management and workflow systems.

How GuardDuty works:



via - https://aws.amazon.com/guardduty/

Incorrect options:

VPC Flow Logs, API Gateway logs, S3 access logs

ELB logs, DNS logs, CloudTrail events

CloudFront logs, API Gateway logs, CloudTrail events

These three options contradict the explanation provided above, so these options are incorrect.

Reference:

https://aws.amazon.com/guardduty/

Question 15: **Skipped**

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?

- **Amazon FSx for Windows File Server**
- **Amazon EMR**
- **AWS Glue**
- **Amazon FSx for Lustre** **(Correct)**

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/

Question 16: **Skipped**

A geological research agency maintains the seismological data for the last 100 years. The data has a velocity of 1GB per minute. You would like to store the data with only the most relevant attributes to build a predictive model for earthquakes.

What AWS services would you use to build the most cost-effective solution with the LEAST amount of infrastructure maintenance?

- ○ **Ingest the data in Kinesis Data Analytics and use SQL queries to filter and transform the data before writing to S3**

- ○ **Ingest the data in Kinesis Data Streams and use an intermediary Lambda function to filter and transform the incoming stream before the output is dumped on S3**

○ **Ingest the data in Kinesis Data Firehose and use an intermediary Lambda function to filter and transform the incoming stream before the output is dumped on S3** **(Correct)**

○ **Ingest the data in a Spark Streaming Cluster on EMR use Spark Streaming transformations before writing to S3**

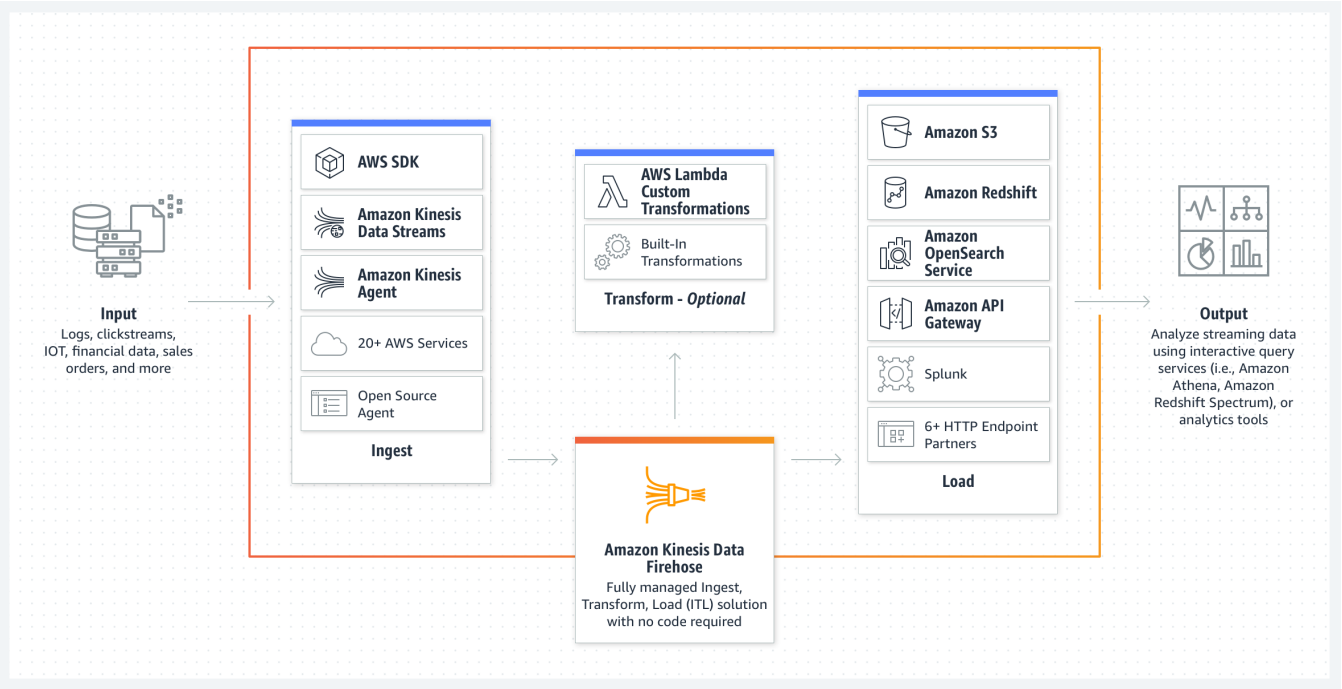
Explanation

Correct option:

Ingest the data in Kinesis Data Firehose and use an intermediary Lambda function to filter and transform the incoming stream before the output is dumped on S3

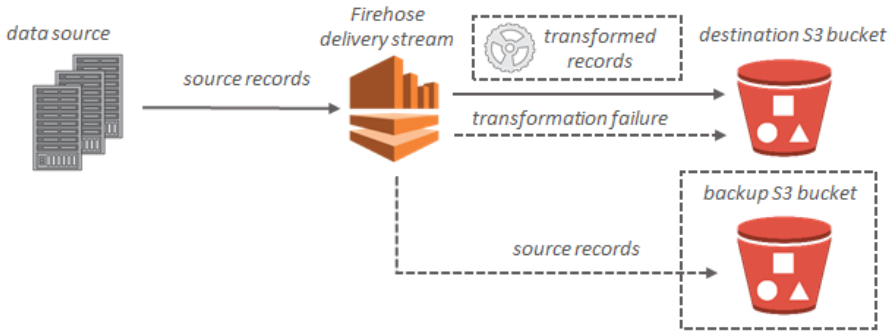
Amazon Kinesis Data Firehose is the easiest way to load streaming data into data stores and analytics tools. It can capture, transform, and load streaming data into Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk, enabling near real-time analytics with existing business intelligence tools and dashboards you're already using today. It is a fully managed service that automatically scales to match the throughput of your data and requires no ongoing administration. It can also batch, compress, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security.

Kinesis Data Firehose Overview



via - https://aws.amazon.com/kinesis/data-firehose/

The correct option is to ingest the data in Kinesis Data Firehose and use a Lambda function to filter and transform the incoming data before the output is dumped on S3. This way you only need to store a sliced version of the data with only the relevant data attributes required for your model. Also it should be noted that this solution is entirely serverless and requires no infrastructure maintenance.

Kinesis Data Firehose to S3: 

via

- https://docs.aws.amazon.com/firehose/latest/dev/what-is-this-service.html

Incorrect options:

Ingest the data in Kinesis Data Analytics and use SQL queries to filter and transform the data before writing to S3 - Amazon Kinesis Data Analytics is the easiest way to analyze streaming data in real-time. Kinesis Data Analytics enables you to easily and quickly build queries and sophisticated streaming applications in three simple steps: setup your streaming data sources, write your queries or streaming applications, and set up your destination for processed data. Kinesis Data Analytics cannot directly ingest data from the source as it ingests data either from Kinesis Data Streams or Kinesis Data Firehose, so this option is ruled out.

Ingest the data in Kinesis Data Streams and use an intermediary Lambda function to filter and transform the incoming stream before the output is dumped on S3 - Amazon Kinesis Data Streams (KDS) is a massively scalable, highly durable data ingestion and processing service optimized for streaming data. Amazon Kinesis Data Streams is integrated with a number of AWS services, including Amazon Kinesis Data Firehose for near real-time transformation.

Kinesis Data Streams cannot directly write the output to S3. Unlike Firehose, KDS does not offer a ready-made integration via an intermediary Lambda function to reliably dump data into S3. You will need to do a lot of custom coding to get the Lambda function to process the incoming stream and then store the transformed output to S3 with the constraint that the buffer is maintained reliably and no transformed data is lost. So this option is incorrect.

Ingest the data in a Spark Streaming Cluster on EMR use Spark Streaming transformations before writing to S3 - 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. Using an EMR cluster would imply managing the underlying infrastructure so it's ruled out because the correct solution for the given use-case should require the least amount of infrastructure maintenance.

Reference:

https://aws.amazon.com/kinesis/data-firehose/

Question 17: **Skipped**

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 EFS file system 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 EFS file systems of other AWS regions as EFS is a regional service and it does not allow access from other AWS regions**

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

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

○ **The spreadsheet on the EFS file system can be accessed in other AWS regions by using an inter-region VPC peering connection** **(Correct)**

Explanation

Correct option:

The spreadsheet on the EFS file system 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 RDS MySQL database which can then be accessed from any AWS region

Copying the spreadsheet into S3 or RDS database is not the correct solution as it involves a lot of operational overhead. For 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 collabaration as per the use-case. So both these options are ruled out.

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

Question 18: **Skipped**

The solo founder at a tech startup has just created a brand new AWS account. The founder has provisioned an EC2 instance 1A which is running in region A. Later, he takes a snapshot of the instance 1A and then creates a new 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?

- ○ 1 EC2 instance and 1 snapshot exist in region B
- ○ 1 EC2 instance and 1 AMI exist in region B
- ○ 1 EC2 instance, 1 AMI and 1 snapshot exist in region B **(Correct)**
- ○ 1 EC2 instance and 2 AMIs exist in region B

Explanation

Correct option:

1 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 EC2 instance, 1 AMI and 1 snapshot in region B.

AMI Overview:

Amazon Machine Images (AMI)

PDF | Kindle | RSS

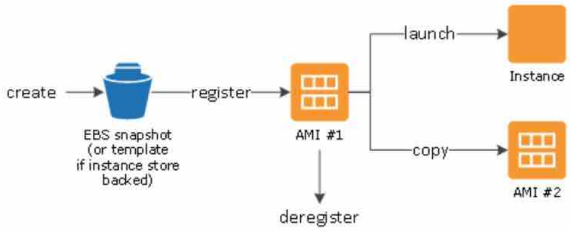
An Amazon Machine Image (AMI) provides the information required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. You can use different AMIs to launch instances 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 EC2 instance and 1 AMI exist in region B

1 EC2 instance and 2 AMIs exist in region B

1 EC2 instance and 1 snapshot 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 EC2 instance, 1 AMI and 1 snapshot in region B. Hence all three options are incorrect.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html

Question 19: **Skipped**

A financial services company recently launched an initiative to improve the security of its AWS resources and it had enabled AWS Shield Advanced across multiple AWS accounts owned by the company. Upon analysis, the company has found that the costs incurred are much higher than expected.

Which of the following would you attribute as the underlying reason for the unexpectedly high costs for AWS Shield Advanced service?

- ◯ **Consolidated billing has not been enabled. All the AWS accounts should fall under a single consolidated billing for the monthly fee to be charged only once** **(Correct)**

- ◯ **Savings Plans has not been enabled for the AWS Shield Advanced service across all the AWS accounts**

- ◯ **AWS Shield Advanced also covers AWS Shield Standard plan, thereby resulting in increased costs**

- ◯ **AWS Shield Advanced is being used for custom servers, that are not part of AWS Cloud, thereby resulting in increased costs**

Explanation

Correct option:

Consolidated billing has not been enabled. All the AWS accounts should fall under a single consolidated billing for the monthly fee to be charged only once - If your organization has multiple AWS accounts, then you can subscribe multiple AWS Accounts to AWS Shield Advanced by individually enabling it on each account using the AWS Management Console or API. You will pay the monthly fee once as long as the AWS accounts are all under a single consolidated billing, and you own all the AWS accounts and resources in those accounts.

Incorrect options:

AWS Shield Advanced is being used for custom servers, that are not part of AWS Cloud, thereby resulting in increased costs - AWS Shield Advanced does offer protection to resources outside of AWS. This should not cause unexpected spike in billing costs.

AWS Shield Advanced also covers AWS Shield Standard plan, thereby resulting in increased costs - AWS Shield Standard is automatically enabled for all AWS customers at no additional cost. AWS Shield Advanced is an optional paid service.

Savings Plans has not been enabled for the AWS Shield Advanced service across all the AWS accounts - This option has been added as a distractor. Savings Plans is a flexible pricing model that offers low prices on EC2, Lambda, and Fargate usage, in exchange for a commitment to a consistent amount of usage (measured in $/hour) for a 1 or 3 year term. Savings Plans is not applicable for the AWS Shield Advanced service.

References:

https://aws.amazon.com/shield/faqs/

https://aws.amazon.com/savingsplans/faq/

Question 20: **Skipped**

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 QuickSight with Redshift
- Leverage Amazon Athena with S3
- Leverage Amazon API Gateway with Kinesis Data Analytics **(Correct)**
- Leverage Amazon API Gateway with AWS Lambda

Explanation

Correct option:

Leverage Amazon API Gateway with 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.

Kinesis Data Analytics:

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

Incorrect options:

Leverage Amazon Athena with 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 QuickSight with 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/

Question 21: **Skipped**

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 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?

- ◯ **Use permissions boundary to control the maximum permissions employees can grant to the IAM principals** **(Correct)**
- ◯ **The CTO should review the permissions for each new developer's IAM user so that such incidents don't recur**
- ◯ **Only root user should have full database access in the organization**
- ◯ **Remove full database access for all IAM users in the organization**

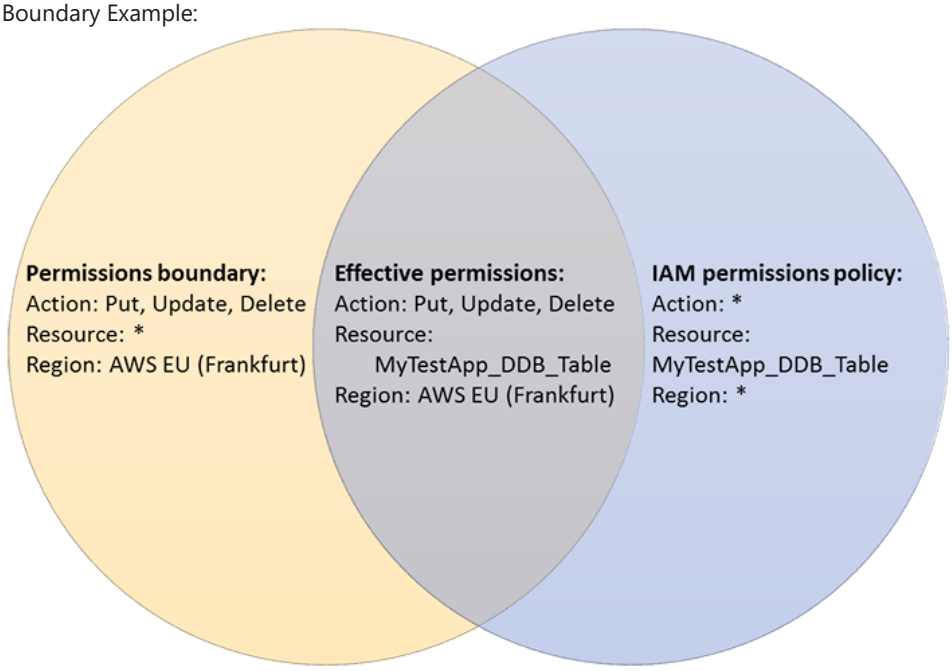
Explanation

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/

Question 22: **Skipped**

The engineering team at a data analytics company has observed that its flagship application functions at its peak performance when the underlying EC2 instances have a CPU utilization of about 50%. The application is built on a fleet of 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 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 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 a Cloudwatch alarm triggered on a CPU utilization threshold of 50%**

- ◯ **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%** **(Correct)**

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%

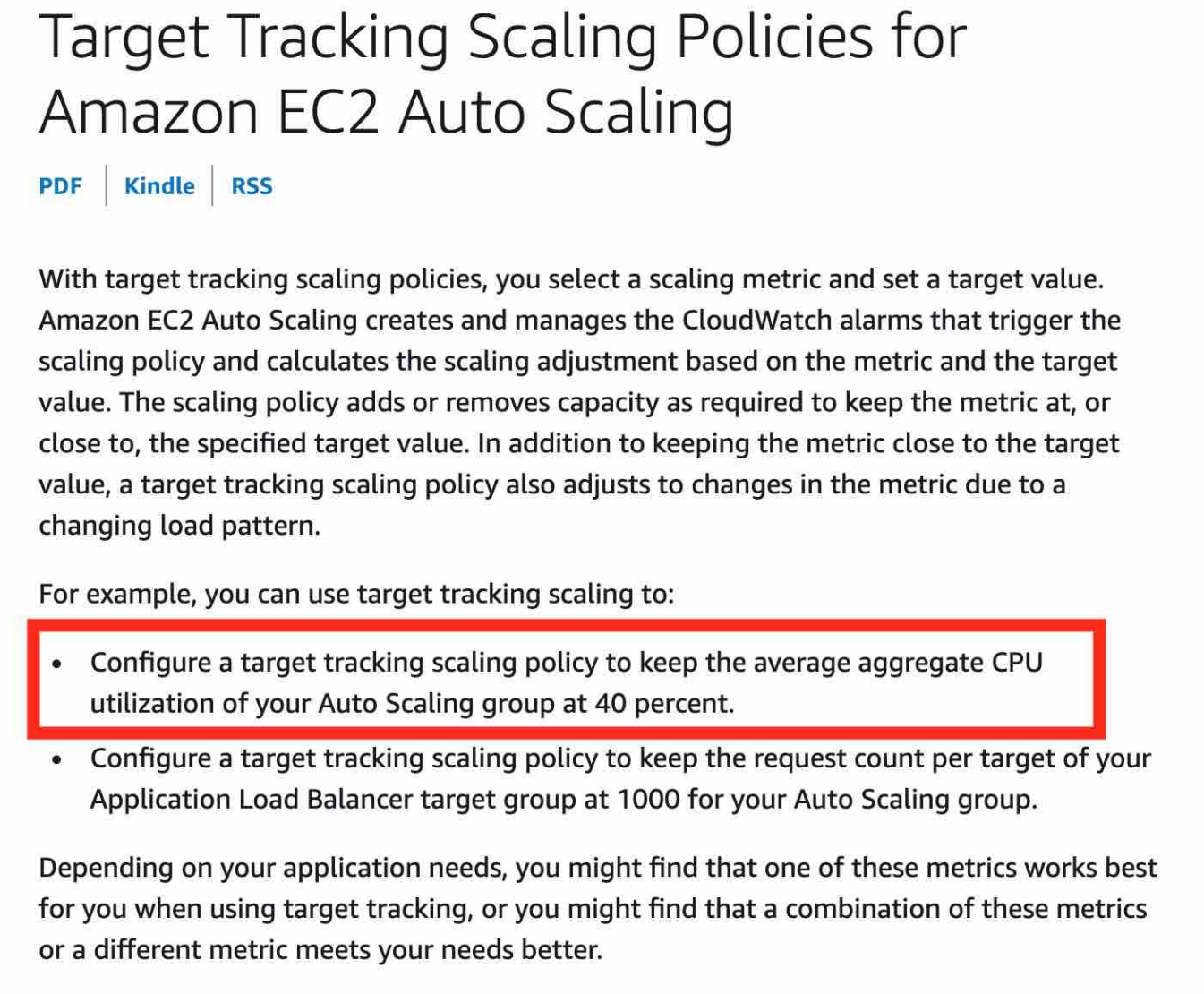
An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also enables you to use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies.

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 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 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

Question 23: **Skipped**

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 3GB. The junior scientist is using S3 Transfer Acceleration (S3TA) for faster image upload. It turns out that S3TA did not result in an accelerated transfer.

Given this scenario, which of the following is correct regarding the charges for this image transfer?

- ○ **The junior scientist needs to pay both S3 transfer charges and S3TA transfer charges for the image upload**
- ○ **The junior scientist only needs to pay S3TA transfer charges for the image upload**
- ○ **The junior scientist does not need to pay any transfer charges for the image upload** **(Correct)**
- ○ **The junior scientist only needs to pay S3 transfer charges for the image upload**

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.

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 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 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/

Question 24: **Skipped**

A company is in the process of migrating its on-premises SMB file shares to AWS so the company can get out of the business of managing multiple file servers across dozens of offices. The company has 200TB of data in its file servers. The existing on-premises applications and native Windows workloads should continue to have low latency access to this data without any disruptions after the migration. The company also wants any new applications deployed on AWS to have access to this migrated data.

Which of the following is the best solution to meet this requirement?

- **Use Amazon FSx File Gateway to provide low-latency, on-premises access to fully managed file shares in Amazon EFS. The applications deployed on AWS can access this data directly from Amazon EFS**

- **Use Amazon Storage Gateway's File Gateway to provide low-latency, on-premises access to fully managed file shares in Amazon S3. The applications deployed on AWS can access this data directly from Amazon S3**

- **Use Amazon FSx File Gateway to provide low-latency, on-premises access to fully managed file shares in Amazon FSx for Windows File Server. The applications deployed on AWS can access this data directly from Amazon FSx in AWS** **(Correct)**

- **Use Amazon Storage Gateway's File Gateway to provide low-latency, on-premises access to fully managed file shares in Amazon FSx for Windows File Server. The applications deployed on AWS can access this data directly from Amazon FSx in AWS**

Explanation

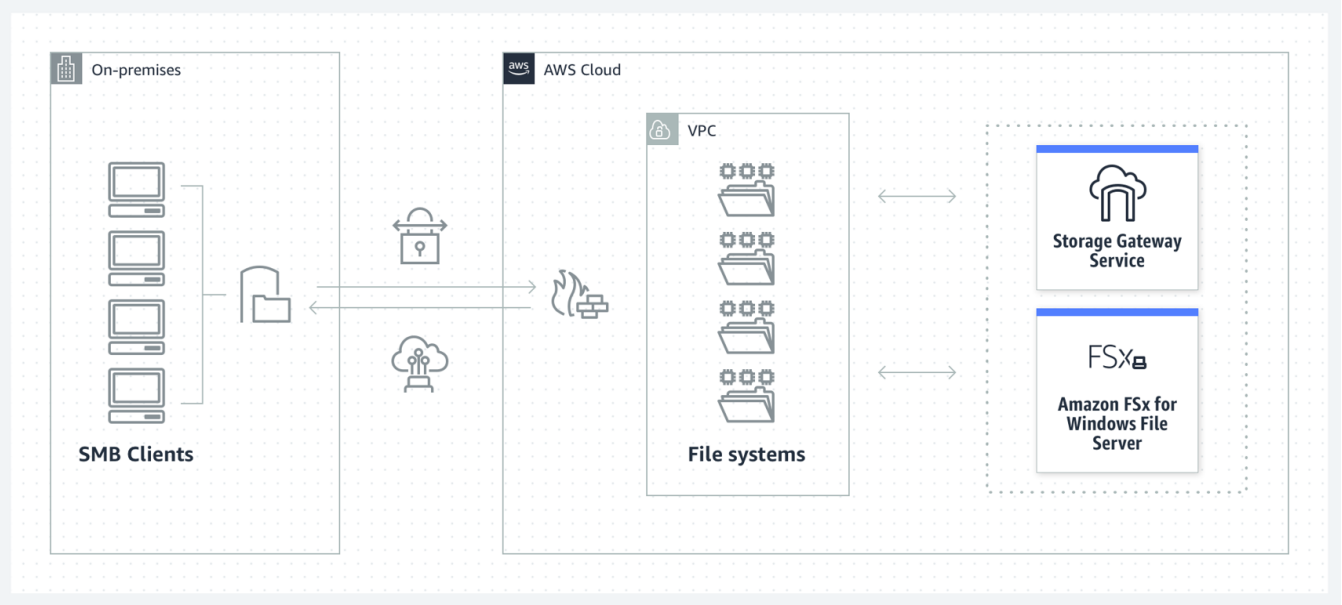
Correct option:

Use Amazon FSx File Gateway to provide low-latency, on-premises access to fully managed file shares in Amazon FSx for Windows File Server. The applications deployed on AWS can access this data directly from Amazon FSx in AWS

For user or team file shares, and file-based application migrations, Amazon FSx File Gateway provides low-latency, on-premises access to fully managed file shares in Amazon FSx for Windows File Server. For applications deployed on AWS, you may access your file shares directly from Amazon FSx in AWS.

For your native Windows workloads and users, or your SMB clients, Amazon FSx for Windows File Server provides all of the benefits of a native Windows SMB environment that is fully managed and secured and scaled like any other AWS service. You get detailed reporting, replication, backup, failover, and support for native Windows tools like DFS and Active Directory.

Amazon FSx File Gateway:



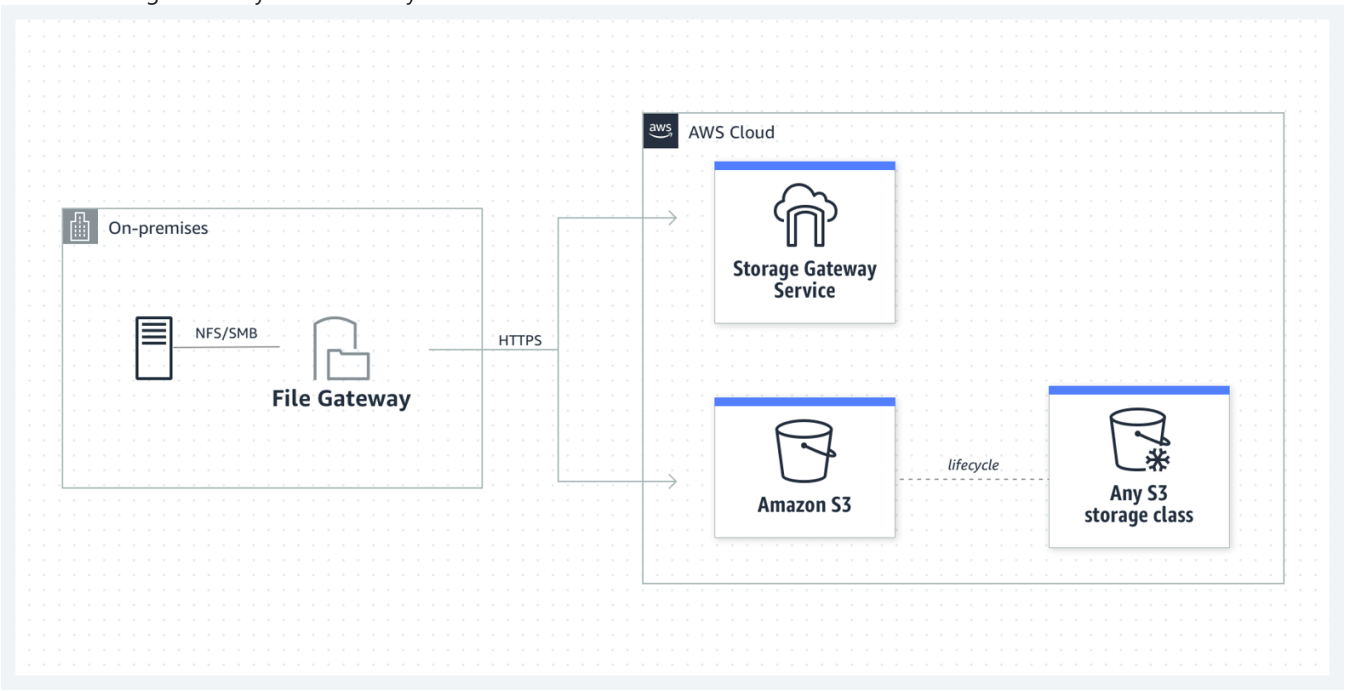
via - https://aws.amazon.com/storagegateway/file/

Incorrect options:

Use Amazon Storage Gateway's File Gateway to provide low-latency, on-premises access to fully managed file shares in Amazon FSx for Windows File Server. The applications deployed on AWS can access this data directly from Amazon FSx in AWS - When you need to access S3 using a file system protocol, you should use File Gateway. You get a local cache in the gateway that provides high throughput and low latency over SMB.

Amazon Storage Gateway's File Gateway does not support file shares for native Windows workloads, so this option is incorrect.

Amazon Storage Gateway's File Gateway:



Use Amazon Storage Gateway's File Gateway to provide low-latency, on-premises access to fully managed file shares in Amazon S3. The applications deployed on AWS can access this data directly from Amazon S3 - - When you need to access S3 using a file system protocol, you should use File Gateway. You get a local cache in the gateway that provides high throughput and low latency over SMB.

The given use case requires native Windows support for the applications. File Gateway can only be used to access S3 objects using a file system protocol, so this option is incorrect.

Use Amazon FSx File Gateway to provide low-latency, on-premises access to fully managed file shares in Amazon EFS. The applications deployed on AWS can access this data directly from Amazon EFS - Amazon FSx File Gateway provides access to fully managed file shares in Amazon FSx for Windows File Server and it does not support EFS. You should also note that EFS uses the Network File System version 4 (NFS v4) protocol and it does not support SMB protocol. Therefore this option is incorrect for the given use case.

References:

https://aws.amazon.com/storagegateway/file/fsx/

https://aws.amazon.com/storagegateway/faqs/

https://aws.amazon.com/blogs/storage/aws-reinvent-recap-choosing-storage-for-on-premises-file-based-workloads/

Question 25: **Skipped**

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 Kinesis Data Streams which uses a Lambda function to process these updates and then store these processed updates in DynamoDB** **(Correct)**

○ **Push score updates to Kinesis Data Streams which uses a fleet of EC2 instances (with Auto Scaling) to process the updates in Kinesis Data Streams and then store these processed updates in DynamoDB**

○ **Push score updates to an SQS queue which uses a fleet of EC2 instances (with Auto Scaling) to process these updates in the SQS queue and then store these processed updates in an RDS MySQL database**

○ **Push score updates to an SNS topic, subscribe a Lambda function to this SNS topic to process the updates and then store these processed updates in a SQL database running on Amazon EC2**

Explanation

Correct option:

Push score updates to Kinesis Data Streams which uses a Lambda function to process these updates and then store these processed updates in 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.

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 DynamoDB.

Incorrect options:

Push score updates to an SQS queue which uses a fleet of EC2 instances (with Auto Scaling) to process these updates in the SQS queue and then store these processed updates in an RDS MySQL database

Push score updates to Kinesis Data Streams which uses a fleet of EC2 instances (with Auto Scaling) to process the updates in Kinesis Data Streams and then store these processed updates in DynamoDB

Push score updates to an SNS topic, subscribe a Lambda function to this SNS topic to process the updates, and then store these processed updates in a SQL database running on Amazon EC2

These three options use EC2 instances as part of the solution architecture. The use-case seeks to minimize the management overhead required to maintain the solution. However, 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/

Question 26: **Skipped**

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 an Amazon Aurora Global Database for the `games` table and use Amazon Aurora for the `users` and `games_played` tables** **(Correct)**
- ◯ **Use an Amazon Aurora Global Database for the `games` table and use DynamoDB tables for the `users` and `games_played` tables**
- ◯ **Use a DynamoDB global table for the `games` table and use Amazon Aurora for the `users` and `games_played` tables**
- ◯ **Use a DynamoDB global table for the `games` table and use DynamoDB tables for the `users` and `games_played` tables**

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 DynamoDB tables for the `users` and `games_played` tables

Use a DynamoDB global table for the `games` table and use Amazon Aurora for the `users` and `games_played` tables

Use a DynamoDB global table for the `games` table and use DynamoDB tables for the `users` and `games_played` tables

Here, we want minimal application refactoring. DynamoDB and Aurora have a completely different API, due to Aurora being SQL and DynamoDB being NoSQL. So all three options are incorrect, as they have DynamoDB as one of the components.

Reference:

https://aws.amazon.com/rds/aurora/faqs/

Question 27: **Skipped**

A software engineering intern at an e-commerce company is documenting the process flow to provision EC2 instances via the Amazon EC2 API. These instances are to be used for an internal application that processes HR 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)

Instance Store

Throughput Optimized HDD (st1) **(Correct)**

General Purpose SSD (gp2)

Cold HDD (sc1) **(Correct)**

Provisioned IOPS SSD (io1)

Explanation

Correct options:

Throughput Optimized HDD (st1)

Cold HDD (sc1)

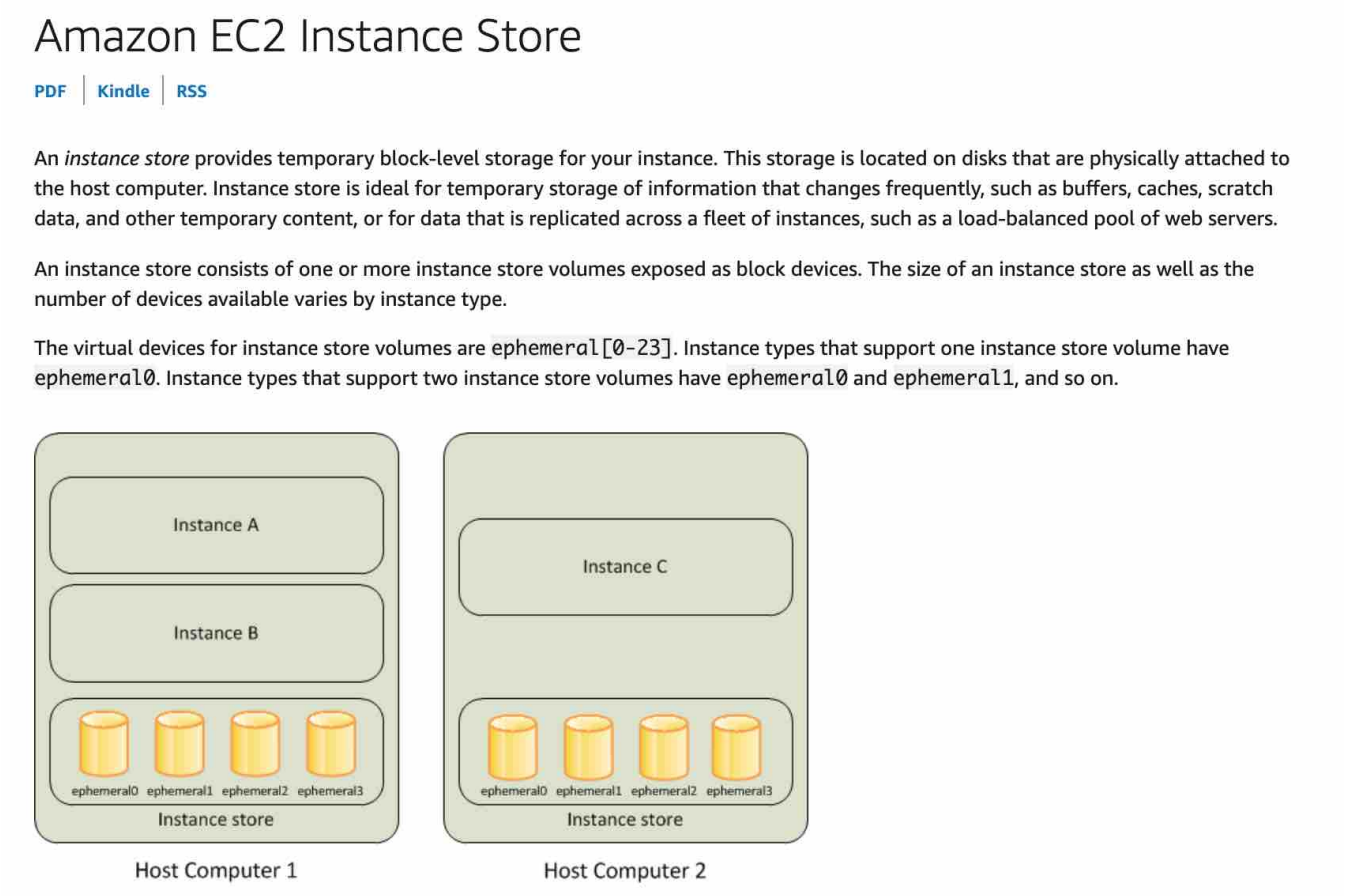
The EBS volume types fall into two categories:

SSD-backed volumes optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS.

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 EBS volumes.



via - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html

Incorrect options:

General Purpose SSD (gp2)

Provisioned IOPS SSD (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

Question 28: **Skipped**

The DevOps team at an e-commerce company wants to perform some maintenance work on a specific 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)

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

Take a snapshot of the instance, create a new 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

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 **(Correct)**

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 you can manually set the instance's health status back to healthy and activate the ScheduledActions process type again

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 (Correct)

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.

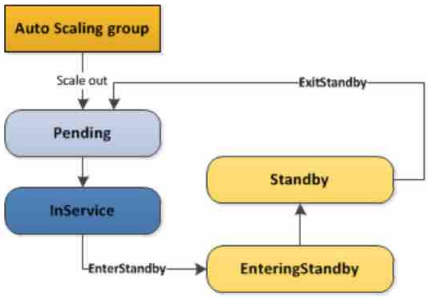
How Standby State Works:

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 ReplaceUnhealthly 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 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 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

Question 29: **Skipped**

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 EC2 instance topologies should this application be deployed on?

- ○ **The EC2 instances should be deployed in a spread placement group so that there are no correlated failures**
- ○ **The 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** **(Correct)**
- ○ **The EC2 instances should be deployed in an Auto Scaling group so that application meets high availability requirements**
- ○ **The EC2 instances should be deployed in a partition placement group so that distributed workloads can be handled effectively**

Explanation

Correct option:

The 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:

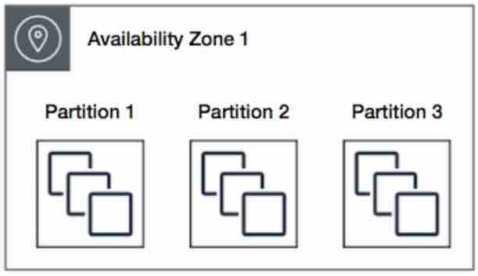
The EC2 instances should be deployed in a partition placement group so that distributed workloads can be handled effectively - A partition placement group spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of 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

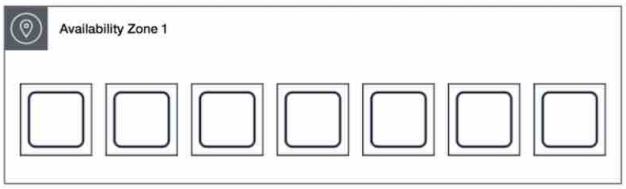
The EC2 instances should be deployed in a spread placement group so that there are no correlated failures - A spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source. The instances are placed across distinct underlying 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 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

Question 30: **Skipped**

A media company runs a photo-sharing web application that is accessed across three different countries. The application is deployed on several 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?

- ◯ **Use Geo Restriction feature of Amazon CloudFront in a VPC**
- ◯ **Configure the security group on the Application Load Balancer**
- ◯ **Configure the security group for the EC2 instances**
- ◯ **Configure AWS WAF on the Application Load Balancer in a VPC** **(Correct)**

Explanation

Correct option:

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 WAF on the Application Load Balancer in a 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 VPC - Geo Restriction feature of 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 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/

https://aws.amazon.com/about-aws/whats-new/2016/12/AWS-WAF-now-available-on-Application-Load-Balancer/

Question 31: **Skipped**

A retail company has set up a Network Load Balancer (NLB) having a target group that is configured to use an Amazon EC2 Auto Scaling group with multiple EC2 instances (across 3 Availability Zones) that run the web service. The company is getting poor feedback from its customers regarding the application's availability as the NLB is unable to detect HTTP errors for the application. These HTTP errors require a manual restart of the EC2 instances that run the web service.

The company has hired you as an AWS Certified Solutions Architect Associate to build the best-fit solution that does not require custom development/scripting effort. Which of the following will you suggest?

○ **Configure HTTP health checks on the Network Load Balancer (NLB) by pointing to the URL of the application. Leverage the Auto Scaling group to replace unhealthy instances**

○ **Replace the Network Load Balancer (NLB) with an Application Load Balancer (ALB) and configure HTTP health checks on the ALB by pointing to the URL of the application. Leverage the Auto Scaling group to replace unhealthy instances** **(Correct)**

- ◯ **Set up a CloudWatch alarm to monitor the UnhealthyHostCount metric for the NLB. Leverage the Auto Scaling group to replace unhealthy instances when the alarm is in the ALARM state**

- ◯ **Set up a cron job on the EC2 instances to inspect the web application's logs at a regular frequency. When HTTP errors are detected, force an application restart**

Explanation

Correct option:

Replace the Network Load Balancer (NLB) with an Application Load Balancer (ALB) and configure HTTP health checks on the ALB by pointing to the URL of the application. Leverage the Auto Scaling group to replace unhealthy instances

A Network Load Balancer (NLB) functions at the fourth layer of the Open Systems Interconnection (OSI) model. It can handle millions of requests per second. After the load balancer receives a connection request, it selects a target from the target group for the default rule. It attempts to open a TCP connection to the selected target on the port specified in the listener configuration.

A load balancer serves as the single point of contact for clients. The load balancer distributes incoming traffic across multiple targets, such as Amazon EC2 instances. This increases the availability of your application. You add one or more listeners to your load balancer.

A listener checks for connection requests from clients, using the protocol and port that you configure, and forwards requests to a target group. Each target group routes requests to one or more registered targets, such as EC2 instances, using the TCP protocol and the port number that you specify.

Health check settings

You configure active health checks for the targets in a target group using the following settings. If the health checks exceed **UnhealthyThresholdCount** consecutive failures, the load balancer takes the target out of service. When the health checks exceed **HealthyThresholdCount** consecutive successes, the load balancer puts the target back in service.

Setting	Description	Default
HealthCheckProtocol	The protocol the load balancer uses when performing health checks on targets. The possible protocols are HTTP, HTTPS, and TCP. The default is the TCP protocol. If the target type is `alb`, the supported health check protocols are HTTP and HTTPS.	TCP
HealthCheckPort	The port the load balancer uses when performing health checks on targets. The default is to use the port on which each target receives traffic from the load balancer.	Port on which each target receives traffic from the load balancer.
HealthCheckPath	[HTTP/HTTPS health checks] The ping path that is the destination on the targets for health checks. The default is /.	/
HealthCheckTimeoutSeconds	The amount of time, in seconds, during which no response from a target means a failed health check. This value must be 6 seconds for HTTP health checks and 10 seconds for TCP and HTTPS health checks.	6 seconds for HTTP health checks and 10 seconds for TCP and HTTPS health checks.
HealthCheckIntervalSeconds	The approximate amount of time, in seconds, between health checks of an individual target. This value can be 10 seconds or 30 seconds. The default is 30 seconds. ⚠ **Important** Health checks for a Network Load Balancer are distributed and use a consensus mechanism to determine target health. Therefore, targets receive more than the configured number of health checks. To reduce the impact to your targets if you are using HTTP health checks, use a simpler destination on the targets, such as a static HTML file, or switch to TCP health checks.	30 seconds
HealthyThresholdCount	The number of consecutive successful health checks required before considering an unhealthy target healthy. The range is 2 to 10. The default is 3.	3
UnhealthyThresholdCount	The number of consecutive failed health checks required before considering a target unhealthy. This value must be the same as the healthy threshold count. The default is 3.	3
Matcher	[HTTP/HTTPS health checks] The HTTP codes to use when checking for a successful response from a target. This value must be 200 to 399.	200-399

via - https://docs.aws.amazon.com/elasticloadbalancing/latest/network/target-group-health-checks.html

For the given use case, you need to swap out the NLB with an ALB. This would allow you to use HTTP-based health checks to detect when the web application faces errors. You can then leverage the Auto Scaling group to use the ALB's health checks to identify and replace unhealthy instances.

Add Elastic Load Balancing health checks to an Auto Scaling group

PDF | RSS

The default health checks for an Auto Scaling group are EC2 status checks only. If an instance fails these status checks, it is marked unhealthy and is terminated while Amazon EC2 Auto Scaling launches a new replacement instance.

You can attach one or more load balancer target groups, one or more Classic Load Balancers, or both to your Auto Scaling group. However, by default, the Auto Scaling group does not consider an instance unhealthy and replace it if it fails the Elastic Load Balancing health checks.

To ensure that your Auto Scaling group can determine instance health based on additional load balancer tests, configure the Auto Scaling group to use Elastic Load Balancing (ELB) health checks. The load balancer periodically sends pings, attempts connections, or sends requests to test the EC2 instances and determines if an instance is unhealthy. If you configure the Auto Scaling group to use Elastic Load Balancing health checks, it considers the instance unhealthy if it fails either the EC2 status checks or the Elastic Load Balancing health checks. If you attach multiple load balancer target groups or Classic Load Balancers to the group, all of them must report that an instance is healthy in order for it to consider the instance healthy. If any one of them reports an instance as unhealthy, the Auto Scaling group replaces the instance, even if others report it as healthy.

via - https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-add-elb-healthcheck.html

Incorrect options:

Set up a CloudWatch alarm to monitor the UnhealthyHostCount metric for the NLB. Leverage the Auto Scaling group to replace unhealthy instances when the alarm is in the ALARM state - The Elastic Load Balancing (ELB) service provides you with Amazon CloudWatch metrics (HealthyHostCount and UnhealthyHostCount) to monitor the targets behind your load balancers. Although the unhealthy host count metric gives the aggregate number of failed hosts, there is a common pain point when you create an alarm for unhealthy hosts based on these metrics. This is because there is no easy way for you to tell which target was or is unhealthy. Building a solution using the Cloudwatch alarm requires significant development/scripting effort to identify the unhealthy target, so this option is incorrect.

Configure HTTP health checks on the Network Load Balancer (NLB) by pointing to the URL of the application. Leverage the Auto Scaling group to replace unhealthy instances - The NLB uses HTTP, HTTPS, and TCP as possible protocols when performing health checks on targets. The default is the TCP protocol. If the target type is ALB, the supported health check protocols are HTTP and HTTPS. Although it is now possible to configure an ALB as a target of an NLB, it would end up being a costlier and inefficient solution than just swapping out the NLB with the ALB, so this solution is not the best fit.

Set up a cron job on the EC2 instances to inspect the web application's logs at a regular frequency. When HTTP errors are detected, force an application restart - This option requires significant development/scripting effort to identify the unhealthy target. It's not as elegant a solution as directly leveraging the HTTP health check capabilities of the ALB. So this option is incorrect.

References:

https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html

https://docs.aws.amazon.com/elasticloadbalancing/latest/network/target-group-health-checks.html

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-add-elb-healthcheck.html

https://docs.aws.amazon.com/elasticloadbalancing/latest/network/load-balancer-cloudwatch-metrics.html

https://docs.aws.amazon.com/elasticloadbalancing/latest/network/load-balancer-troubleshooting.html

https://aws.amazon.com/blogs/networking-and-content-delivery/identifying-unhealthy-targets-of-elastic-load-balancer/

Question 32: **Skipped**

An IT security consultancy is working on a solution to protect data stored in S3 from any malicious activity as well as check for any vulnerabilities on EC2 instances.

As a solutions architect, which of the following solutions would you suggest to help address the given requirement?

○ **Use Amazon GuardDuty to monitor any malicious activity on data stored in S3. Use security assessments provided by Amazon**

GuardDuty to check for vulnerabilities on EC2 instances

- ○ Use Amazon Inspector to monitor any malicious activity on data stored in S3. Use security assessments provided by Amazon Inspector to check for vulnerabilities on EC2 instances
- ○ Use Amazon GuardDuty to monitor any malicious activity on data stored in S3. Use security assessments provided by Amazon Inspector to check for vulnerabilities on EC2 instances **(Correct)**
- ○ Use Amazon Inspector to monitor any malicious activity on data stored in S3. Use security assessments provided by Amazon GuardDuty to check for vulnerabilities on EC2 instances

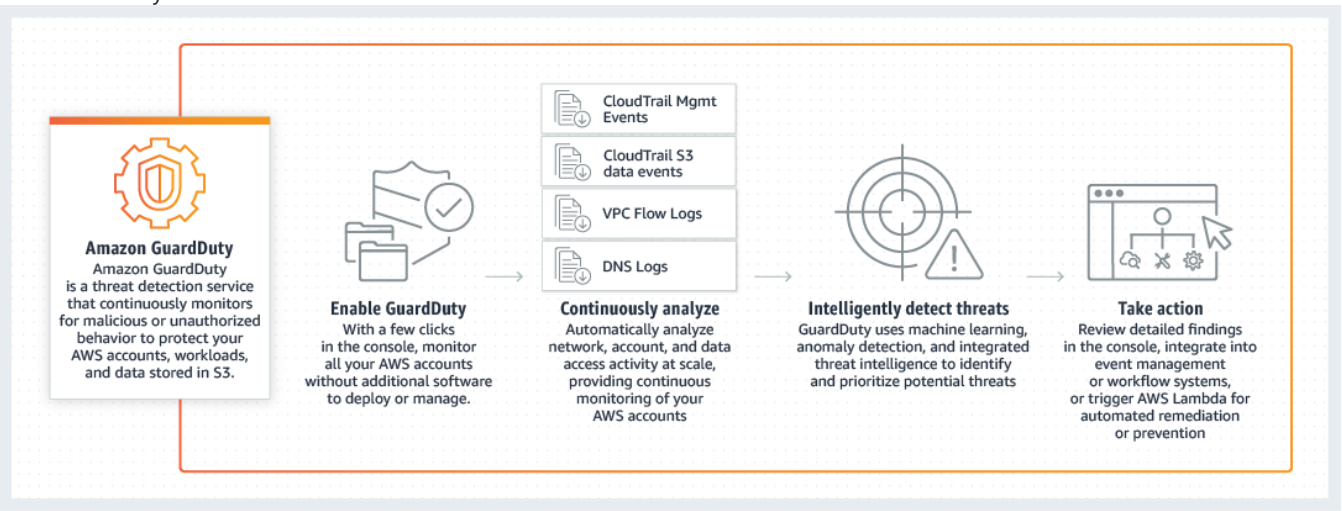
Explanation

Correct option:

Use Amazon GuardDuty to monitor any malicious activity on data stored in S3. Use security assessments provided by Amazon Inspector to check for vulnerabilities on 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 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 S3. Use security assessments provided by Amazon GuardDuty to check for vulnerabilities on EC2 instances

Use Amazon Inspector to monitor any malicious activity on data stored in S3. Use security assessments provided by Amazon Inspector to check for vulnerabilities on EC2 instances

Use Amazon Inspector to monitor any malicious activity on data stored in S3. Use security assessments provided by Amazon GuardDuty to check for vulnerabilities on 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/

Question 33: **Skipped**

A retail company has developed a REST API which is deployed in an Auto Scaling group behind an Application Load Balancer. The API stores the user data in DynamoDB and any static content, such as images, are served via 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 ElastiCache Memcached for S3**
- **Enable DAX for DynamoDB and ElastiCache Memcached for S3**
- **Enable DynamoDB Accelerator (DAX) for DynamoDB and CloudFront for S3** **(Correct)**
- **Enable ElastiCache Redis for DynamoDB and CloudFront for S3**

Explanation

Correct option:

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

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.

DAX is tightly integrated with DynamoDB—you simply provision a DAX cluster, use the DAX client SDK to point your existing DynamoDB API calls at the DAX cluster, and let DAX handle the rest. Because DAX is API-compatible with DynamoDB, you don't

have to make any functional application code changes. DAX is used to natively cache DynamoDB reads.

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 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 S3 bucket where you've stored your content.

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

Incorrect options:

Enable ElastiCache Redis for DynamoDB and CloudFront for 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.

ElastiCache for Redis Overview:

via - https://aws.amazon.com/elasticache/redis/

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

Enable DAX for DynamoDB and ElastiCache Memcached for S3

Enable ElastiCache Redis for DynamoDB and ElastiCache Memcached for 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.

ElastiCache Memcached cannot be used as a cache to serve static content from 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/elasticache/redis/

Question 34: **Skipped**

A company uses DynamoDB as a data store for various kinds of customer data, such as user profiles, user events, clicks, and visited links. Some of these use-cases require a high request rate (millions of requests per second), low predictable latency, and reliability. The company now wants to add a caching layer to support high read volumes.

As a solutions architect, which of the following AWS services would you recommend as a caching layer for this use-case? (Select two)

RDS

ElastiCache (Correct)

Elasticsearch

Redshift

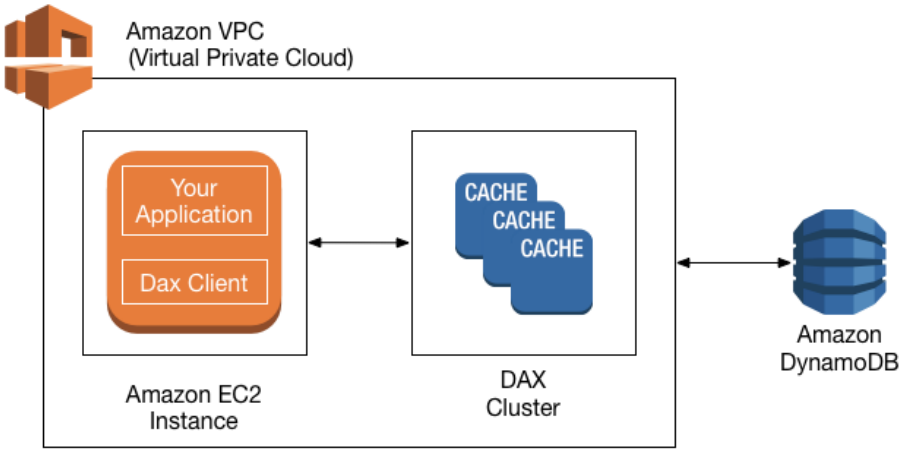
DynamoDB Accelerator (DAX) (Correct)

Explanation

Correct options:

DynamoDB Accelerator (DAX) - Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache for DynamoDB that delivers up to a 10x performance improvement – from milliseconds to microseconds – even at millions of requests per second. DAX does all the heavy lifting required to add in-memory acceleration to your DynamoDB tables, without requiring developers to manage cache invalidation, data population, or cluster management. Therefore, this is a correct option.

DAX Overview:



via

- https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DAX.concepts.html

ElastiCache - Amazon ElastiCache for Memcached is an ideal front-end for data stores like Amazon RDS or Amazon DynamoDB, providing a high-performance middle tier for applications with extremely high request rates and/or low latency requirements. Therefore, this is also a correct option.

Incorrect options:

RDS - Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching, and backups. RDS cannot be used as a caching layer for DynamoDB.

Elasticsearch - Elasticsearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. It cannot be used as a caching layer for DynamoDB.

Redshift - Amazon Redshift is a fully-managed petabyte-scale cloud-based data warehouse product designed for large scale data set storage and analysis. It cannot be used as a caching layer for DynamoDB.

References:

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

https://aws.amazon.com/elasticache/faqs/

Question 35: **Skipped**

A leading video streaming service delivers billions of hours of content from 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 S3 One Zone-Infrequent Access storage class**

○ **Store the intermediary query results in S3 Standard storage class** **(Correct)**

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

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

Explanation

Correct option:

Store the intermediary query results in S3 Standard storage class

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 S3 Glacier Instant Retrieval storage class - 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 S3 Standard-Infrequent Access storage class - 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 S3 One Zone-Infrequent Access storage class - 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/

Question 36: **Skipped**

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

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

- Amazon Route 53
- Amazon CloudFront
- AWS Global Accelerator (Correct)
- AWS Elastic Load Balancing (ELB)

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.

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/

Question 37: **Skipped**

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 S3.

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

S3 Standard-IA => S3 One Zone-IA

S3 Intelligent-Tiering => S3 Standard **(Correct)**

S3 Standard-IA => S3 Intelligent-Tiering

S3 Standard => S3 Intelligent-Tiering

S3 One Zone-IA => S3 Standard-IA **(Correct)**

Explanation

Correct options:

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

S3 Intelligent-Tiering => S3 Standard

S3 One Zone-IA => S3 Standard-IA

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

Incorrect options:

S3 Standard => S3 Intelligent-Tiering

S3 Standard-IA => S3 Intelligent-Tiering

S3 Standard-IA => 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. via - https://docs.aws.amazon.com/AmazonS3/latest/dev/lifecycle-transition-general-considerations.html

Reference:

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

Question 38: **Skipped**

A telecom company operates thousands of hardware devices like switches, routers, cables, etc. The real-time status data for these devices must be fed into a communications application for notifications. Simultaneously, another analytics application needs to read the same real-time status data and analyze all the connecting lines that may go down because of any device failures.

As a Solutions Architect, which of the following solutions would you suggest, so that both the applications can consume the real-time status data concurrently?

- ◯ **Amazon Simple Notification Service (SNS)**
- ◯ **Amazon Kinesis Data Streams** **(Correct)**
- ◯ **Amazon Simple Queue Service (SQS) with Amazon Simple Email Service (Amazon SES)**
- ◯ **Amazon Simple Queue Service (SQS) with Amazon Simple Notification Service (SNS)**

Explanation

Correct option:

Amazon Kinesis Data Streams - Amazon Kinesis Data Streams enables real-time processing of streaming big data. It provides ordering of records, as well as the ability to read and/or replay records in the same order to multiple Amazon Kinesis Applications. The Amazon Kinesis Client Library (KCL) delivers all records for a given partition key to the same record processor, making it easier to build multiple applications reading from the same Amazon Kinesis data stream (for example, to perform counting, aggregation, and filtering).

AWS recommends Amazon Kinesis Data Streams for use cases with requirements that are similar to the following:

1. Routing related records to the same record processor (as in streaming MapReduce). For example, counting and aggregation are simpler when all records for a given key are routed to the same record processor.
2. Ordering of records. For example, you want to transfer log data from the application host to the processing/archival host while maintaining the order of log statements.
3. Ability for multiple applications to consume the same stream concurrently. For example, you have one application that updates a real-time dashboard and another that archives data to Amazon Redshift. You want both applications to consume data from the same stream concurrently and independently.
4. Ability to consume records in the same order a few hours later. For example, you have a billing application and an audit application that runs a few hours behind the billing application. Because Amazon Kinesis Data Streams stores data for up to 365 days, you can run the audit application up to 365 days behind the billing application.

Incorrect options:

Amazon Simple Notification Service (SNS) - Amazon Simple Notification Service (SNS) is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications. Amazon SNS provides topics for high-throughput, push-based, many-to-many messaging. SNS is a notification service and cannot be used for real-time processing of data.

Amazon Simple Queue Service (SQS) with Amazon Simple Notification Service (SNS) - Amazon Simple Queue Service (Amazon SQS) offers a reliable, highly scalable hosted queue for storing messages as they travel between computers. Amazon SQS lets you easily move data between distributed application components and helps you build applications in which messages are processed independently (with message-level ack/fail semantics), such as automated workflows. Since multiple applications need to consume the same data stream concurrently, Kinesis is a better choice when compared to the combination of SQS with SNS.

Amazon Simple Queue Service (SQS) with Amazon Simple Email Service (Amazon SES) - As discussed above, Kinesis is a better option for this use case in comparison to SQS. Also, SES does not fit this use-case. Hence, this option is an incorrect answer.

Reference:

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

Question 39: **Skipped**

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 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 SSE-S3 to encrypt the user data on S3**

◯ **Use SSE-KMS to encrypt the user data on S3** **(Correct)**

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

○ **Use SSE-C to encrypt the user data on S3**

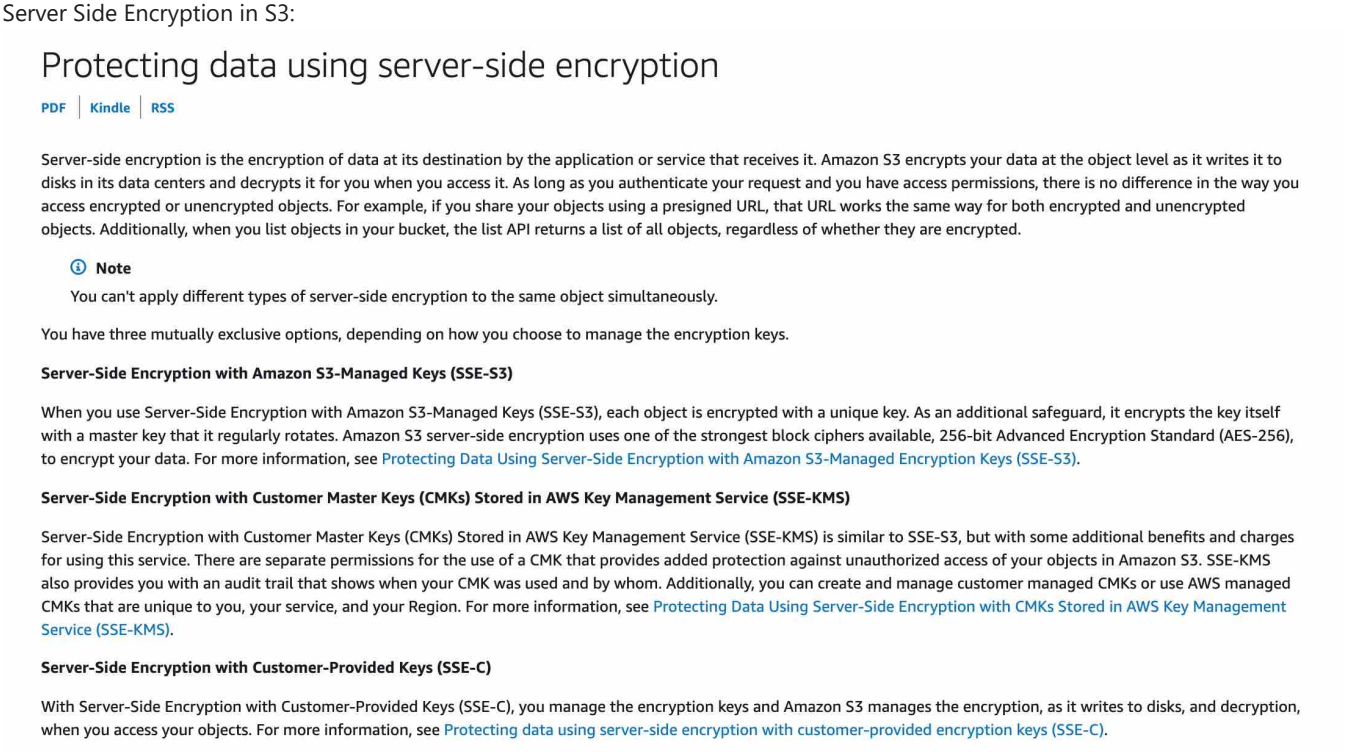
Explanation

Correct option:

Use SSE-KMS to encrypt the user data on 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 SSE-S3 to encrypt the user data on 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 SSE-C to encrypt the user data on 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 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

Question 40: **Skipped**

A research group needs a fleet of 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. Because of the resilient application architecture, the specialized task would continue to be processed even if any instance goes down, as the underlying application architecture 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 EC2 instances?

- ◯ **Use EC2 instances with access to S3 based storage**
- ◯ **Use EBS based EC2 instances**
- ◯ **Use EC2 instances with EFS mount points**
- ◯ **Use Instance Store based EC2 instances** **(Correct)**

Explanation

Correct option:

Use Instance Store based 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 EC2 instances for this use-case.

EC2 Instance Store Overview:

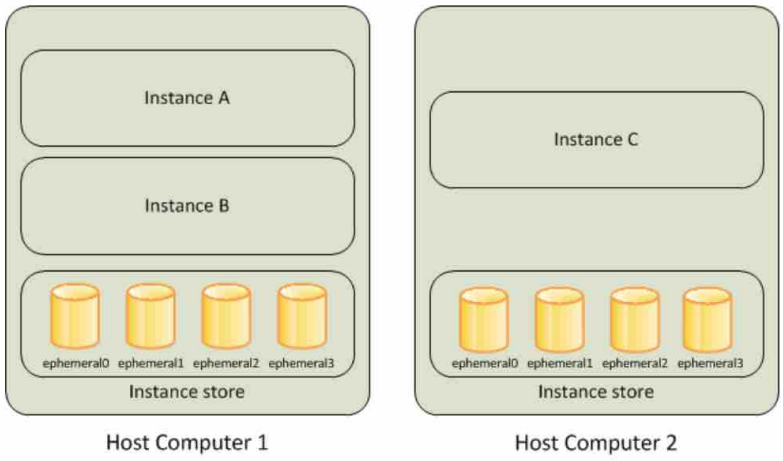
Amazon EC2 Instance Store

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 EBS based EC2 instances - 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 EC2 instances with EFS mount points - Using 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 EC2 instances with access to S3 based storage - Using EC2 instances with access to 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 41: **Skipped**

A company manages a multi-tier social media application that runs on EC2 instances behind an Application Load Balancer. The instances run in an EC2 Auto Scaling group across multiple Availability Zones and use an Amazon Aurora database. As a solutions architect, you have been tasked to make the application more resilient to periodic spikes in request rates.

Which of the following solutions would you recommend for the given use-case? (Select two)

Use CloudFront distribution in front of the Application Load Balancer (Correct)

Use AWS Direct Connect

Use AWS Shield

Use Aurora Replica (Correct)

Use AWS Global Accelerator

Explanation

Correct options:

You can use Aurora replicas and CloudFront distribution to make the application more resilient to spikes in request rates.

Use Aurora Replica

Aurora Replicas have two main purposes. You can issue queries to them to scale the read operations for your application. You typically do so by connecting to the reader endpoint of the cluster. That way, Aurora can spread the load for read-only connections across as many Aurora Replicas as you have in the cluster. Aurora Replicas also help to increase availability. If the writer instance in a cluster becomes unavailable, Aurora automatically promotes one of the reader instances to take its place as the new writer. Up to 15 Aurora Replicas can be distributed across the Availability Zones that a DB cluster spans within an AWS Region.

Use CloudFront distribution in front of the Application Load Balancer

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. CloudFront points of presence (POPs) (edge locations) make sure that popular content can be served quickly to your viewers. CloudFront also has regional edge caches that bring more of your content closer to your viewers, even when the content is not popular enough to stay at a POP, to help improve performance for that content.

CloudFront offers an origin failover feature to help support your data resiliency needs. CloudFront is a global service that delivers your content through a worldwide network of data centers called edge locations or points of presence (POPs). If your content is not already cached in an edge location, CloudFront retrieves it from an origin that you've identified as the source for the definitive version of the content.

Incorrect options:

** Use AWS Shield** - AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS. AWS Shield provides always-on detection and automatic inline mitigations that minimize application downtime and latency. There are two tiers of AWS Shield - Standard and Advanced. Shield cannot be used to improve application resiliency to handle spikes in traffic.

Use AWS Global Accelerator - 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. 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. Since CloudFront is better for improving application resiliency to handle spikes in traffic, so this option is ruled out.

Use AWS Direct Connect - AWS Direct Connect lets you establish a dedicated network connection between your network and one of the AWS Direct Connect locations. Using industry-standard 802.1q VLANs, this dedicated connection can be partitioned into multiple virtual interfaces. AWS Direct Connect does not involve the Internet; instead, it uses dedicated, private network connections between your intranet and Amazon VPC. Direct Connect cannot be used to improve application resiliency to handle spikes in traffic.

References:

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/disaster-recovery-resiliency.html

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Replication.html

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

https://docs.aws.amazon.com/global-accelerator/latest/dg/disaster-recovery-resiliency.html

Question 42: **Skipped**

A file-hosting service uses Amazon S3 under the hood to power its storage offerings. Currently all the customer files are uploaded directly under a single 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 S3 bucket for each customer and then upload each customer's files directly under the respective buckets**

- ◯ **Change the application architecture to create customer-specific custom prefixes within the single bucket and then upload the daily files into those prefixed locations** **(Correct)**

- ◯ **Change the application architecture to use EFS instead of Amazon S3 for storing the customers' uploaded files**

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

Explanation

Correct option:

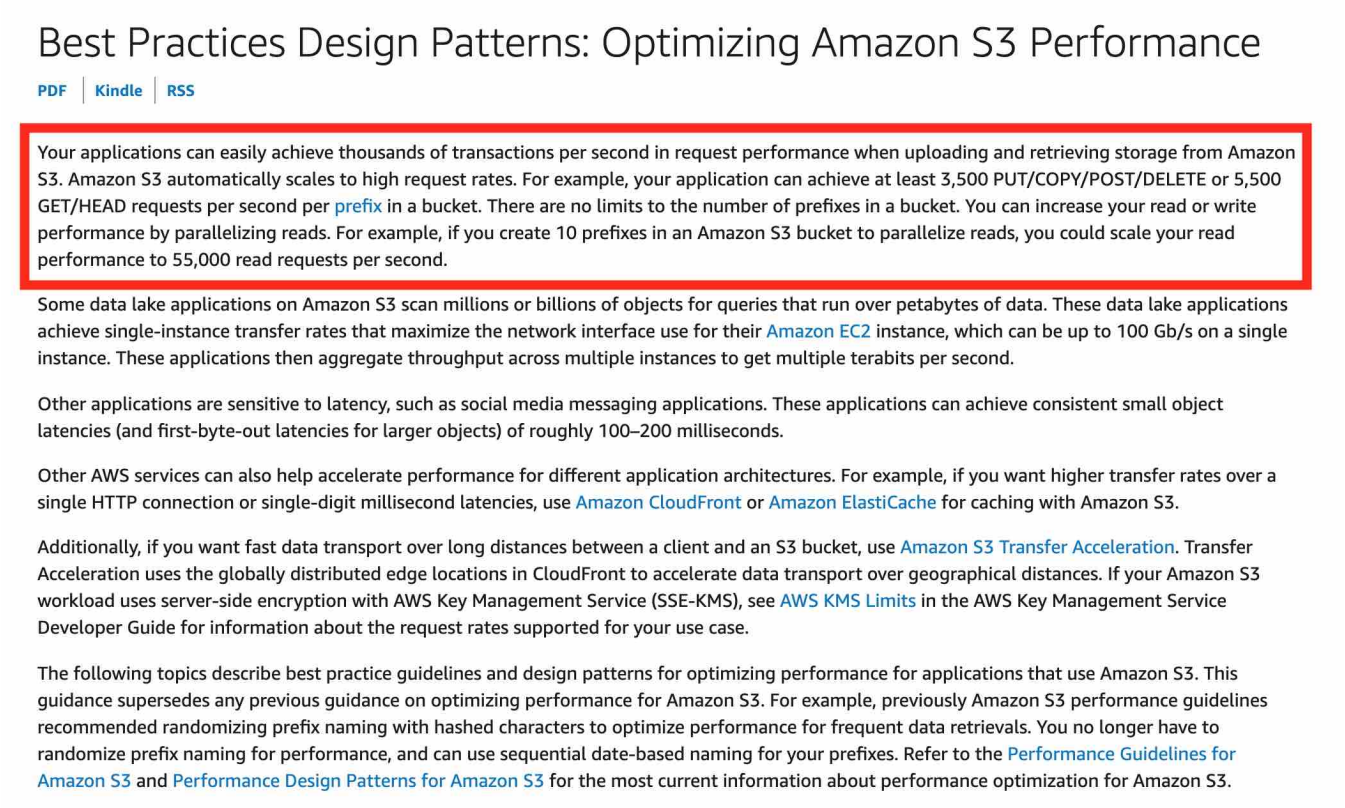
Change the application architecture to create customer-specific custom prefixes within the single bucket and then upload the daily files into those prefixed locations

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 S3 bucket for each customer and then upload each customer's files directly under the respective buckets - Creating a new 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 S3 bucket for each day's data and then upload the daily files directly under that day's bucket - Creating a new 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 EFS instead of Amazon S3 for storing the customers' uploaded files - EFS is a costlier storage option compared to S3, so it is ruled out.

Reference:

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

Question 43: **Skipped**

The engineering team at a Spanish professional football club has built a notification system for its website using Amazon SNS notifications which are then handled by a 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 Lambda service**

- ◯ **Amazon SNS message deliveries to AWS Lambda have crossed the account concurrency quota for Lambda, so the team needs to contact AWS support to raise the account limit** **(Correct)**

- ◯ **The engineering team needs to provision more servers running the SNS service**

- ◯ **Amazon SNS has hit a scalability limit, so the team needs to contact AWS support to raise the account limit**

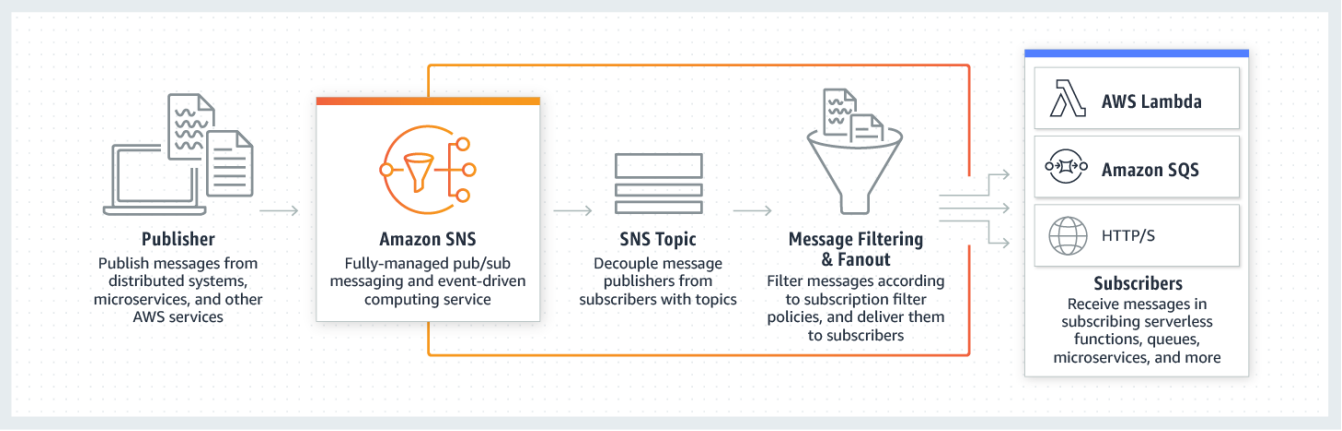
Explanation

Correct option:

Amazon SNS message deliveries to AWS Lambda have crossed the account concurrency quota for Lambda, so the team needs to contact AWS support to raise the account limit

Amazon Simple Notification Service (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 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 SNS service

The engineering team needs to provision more servers running the Lambda service

As both Lambda and 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/

Question 44: **Skipped**

A gaming company uses Amazon Aurora as its primary database service. The company has now deployed 5 multi-AZ read replicas to increase the read throughput and for use as failover target. The replicas have been assigned the following failover priority tiers and corresponding instance sizes are given in parentheses: tier-1 (16TB), tier-1 (32TB), tier-10 (16TB), tier-15 (16TB), tier-15 (32TB).

In the event of a failover, Amazon Aurora will promote which of the following read replicas?

- ○ **Tier-15 (32TB)**
- ○ **Tier-1 (32TB)** **(Correct)**
- ○ **Tier-1 (16TB)**
- ○ **Tier-10 (16TB)**

Explanation

Correct option:

Tier-1 (32TB)

Amazon Aurora features a distributed, fault-tolerant, self-healing storage system that auto-scales up to 128TB per database instance. It delivers high performance and availability with up to 15 low-latency read replicas, point-in-time recovery, continuous backup to Amazon S3, and replication across three Availability Zones (AZs).

For Amazon Aurora, each Read Replica is associated with a priority tier (0-15). In the event of a failover, Amazon Aurora will promote the Read Replica that has the highest priority (the lowest numbered tier). If two or more Aurora Replicas share the same priority, then Amazon RDS promotes the replica that is largest in size. If two or more Aurora Replicas share the same priority and size, then Amazon Aurora promotes an arbitrary replica in the same promotion tier.

Therefore, for this problem statement, the Tier-1 (32TB) replica will be promoted.

Incorrect options:

Tier-15 (32TB)

Tier-1 (16TB)

Tier-10 (16TB)

Given the failover rules discussed earlier in the explanation, these three options are incorrect.

References:

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Managing.Backups.html

https://docs.amazonaws.cn/en_us/AmazonRDS/latest/AuroraUserGuide/Aurora.Managing.Backups.html#Aurora.Managing.FaultTolerance

Question 45: **Skipped**

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)

Create a strong password for the AWS account root user **(Correct)**

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

Encrypt the access keys and save them on Amazon S3

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

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

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

Question 46: **Skipped**

A development team requires permissions to list an 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"
}
```

(Correct)

○

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

○

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

○

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

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 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 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/

Question 47: **Skipped**

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)

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

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

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

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

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

Question 48: **Skipped**

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 S3, runs into hundreds of Terabytes and should be available with millisecond latency.

As a solutions architect, which is the MOST cost-effective storage class that you would recommend to be used for this use-case?

○ **Amazon S3 Standard**

○ **Amazon S3 Intelligent-Tiering (S3 Intelligent-Tiering)**

○ **Amazon S3 Standard-Infrequent Access (S3 Standard-IA)** **(Correct)**

○ **Amazon S3 Glacier Deep Archive**

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 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. Standard-IA is designed for 99.9% availability compared to 99.99% availability of 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 S3 Standard-IA, the job will be auto re-invoked till data is successfully retrieved. Therefore this is the correct option.

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.%	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 - S3 Standard offers high durability, availability, and performance object storage for frequently accessed data. As described above, S3 Standard-IA storage is a better fit than 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, S3 Intelligent-Tiering monitors access patterns and automatically moves objects that have not been accessed to lower-cost access tiers. The 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 S3 Intelligent-Tiering. So, it's cost-efficient to use S3 Standard-IA instead of S3 Intelligent-Tiering.

Amazon S3 Glacier Deep Archive - S3 Glacier Deep Archive is a secure, durable, and low-cost storage class for data archiving. 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 49: **Skipped**

A company has a web application that runs 24*7 in the production environment. The development team at the company runs a clone of the same application in the dev environment for up to 8 hours every day. The company wants to build the MOST cost-optimal solution by deploying these applications using the best-fit pricing options for EC2 instances.

What would you recommend?

- ○ **Use reserved EC2 instances for the production application and spot block instances for the dev application**
- ○ **Use on-demand EC2 instances for the production application and spot instances for the dev application**
- ○ **Use reserved EC2 instances for the production application and on-demand instances for the dev application** **(Correct)**
- ○ **Use reserved EC2 instances for the production application and spot instances for the dev application**

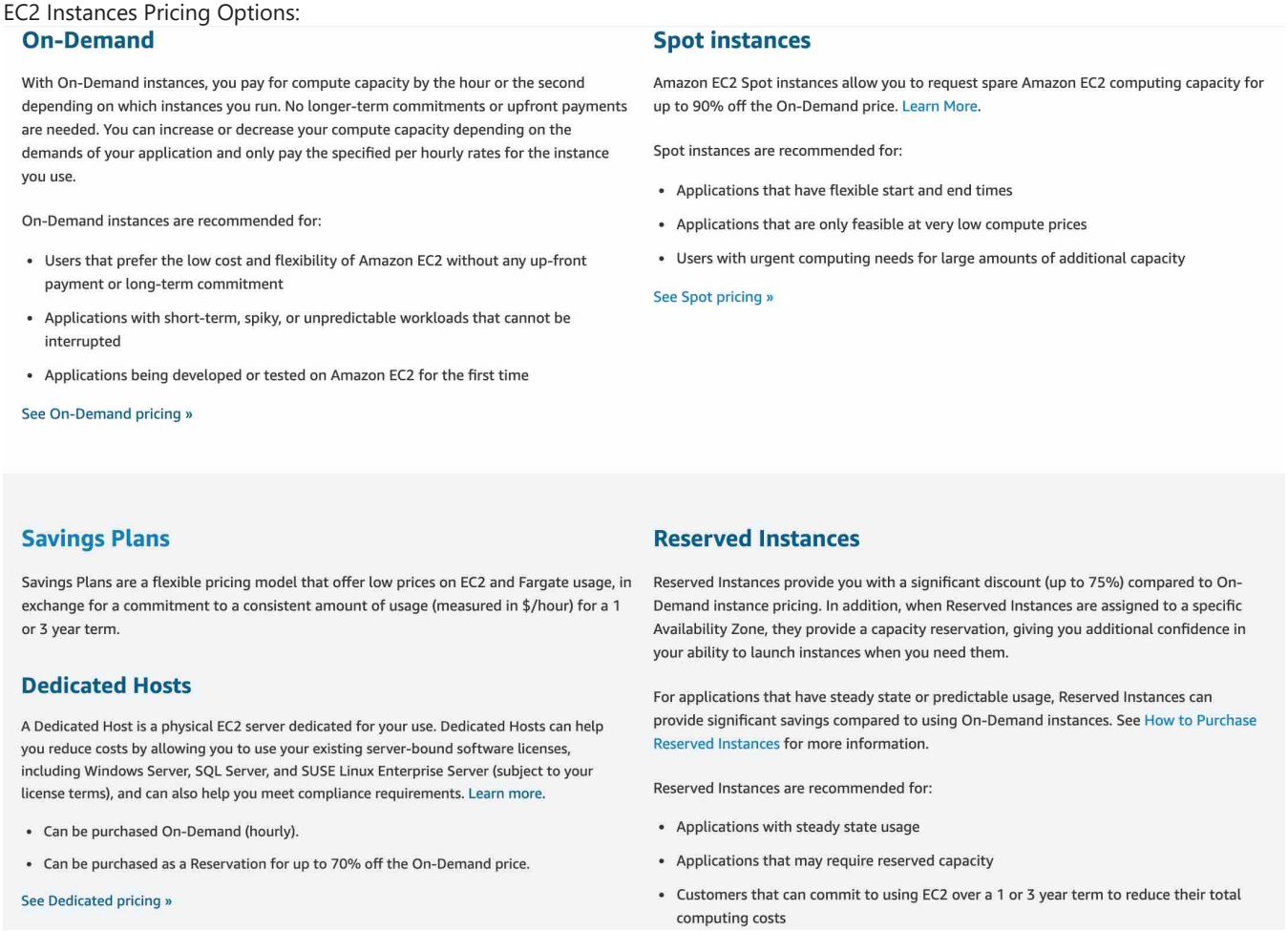
Explanation

Correct option:

Use reserved EC2 instances for the production application and on-demand instances for the dev application

There are multiple pricing options for EC2 instances, such as On-Demand, Savings Plans, Reserved Instances, and Spot Instances.

EC2 Instances Pricing Options:



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 Reserved Instances (RI) provide a significant discount (up to 72%) compared to On-Demand pricing and provide a capacity reservation when used in a specific Availability Zone. RIs provide you with a significant discount (up to 72%) compared to On-Demand instance pricing. You have the flexibility to change families, OS types, and tenancies while benefitting from RI pricing when you use Convertible RIs.

Standard and Convertible RI Features

The following summarizes features of all RIs.

Provide a significant discount compared to running instances On-Demand.

- Can apply to usage across all Availability Zones in an AWS region, or can provide a capacity reservation when assigned to a specific Availability Zone.
- Are offered under three upfront payment options to provide you with payment flexibility at the point of purchase.
- Can be shared between multiple accounts within a consolidated billing family.

The following table summarizes the differences between Standard and Convertible RIs.

Characteristic	Standard	Convertible
Terms (avg. discount off On-Demand)	1yr (40%), 3yr (60%)	1yr (31%), 3yr (54%)
Change Availability Zone, instance size (for Linux OS), networking type	Yes (Using ModifyReservedInstances API and console)	Yes (Using ExchangeReservedInstances API and console)
Change instance families, operating system, tenancy, and payment option		Yes
Benefit from Price Reductions		Yes

Standard and Convertible RI Payment Attributes

- **Offering class:** There are two classes of RIs: Convertible and Standard. Convertible RIs can be exchanged for different Convertible RIs of equal or greater value.
- **Term:** AWS offers Standard RIs for 1-year or 3-year terms. Reserved Instance Marketplace sellers also offer RIs often with shorter terms. AWS offers Convertible RIs for 1-year or 3-year terms.
- **Payment option:** You can choose between three payment options: All Upfront, Partial Upfront, and No Upfront. If you choose the Partial or No Upfront payment option, the remaining balance will be due in monthly increments over the term.

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

For the given use case, you can use reserved EC2 instances for the production application as it is run 24*7. This way you can get a 72% discount if you avail a 3-year term. You can use on-demand instances for the dev application since it is only used for up to 8 hours per day. On-demand offers the flexibility to only pay for the EC2 instance when it is being used (0 to 8 hours for the given use case).

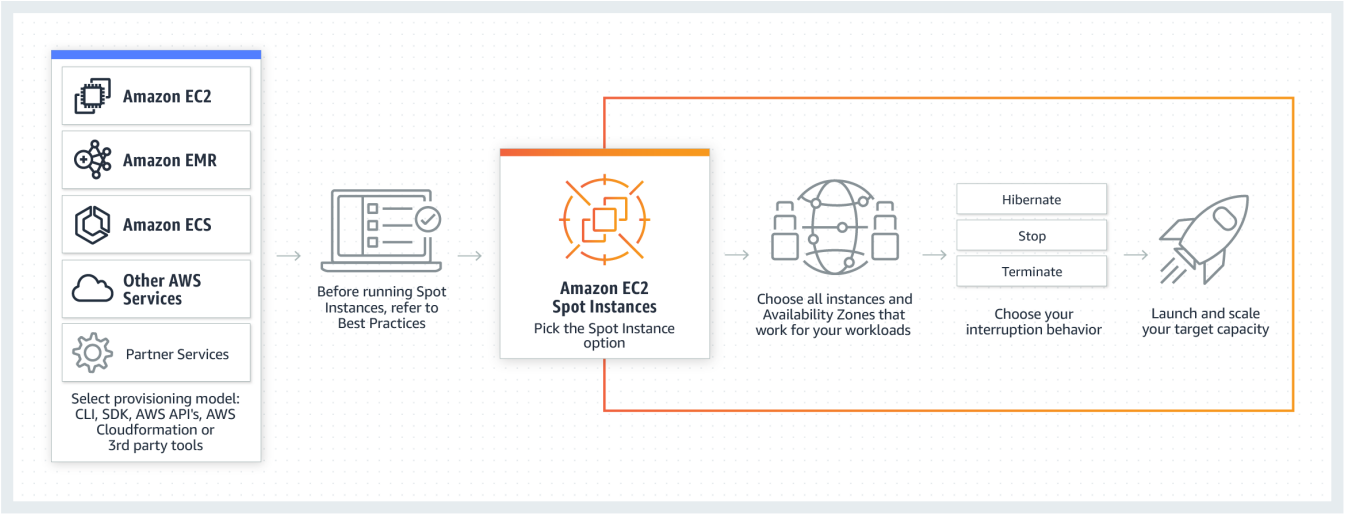
Incorrect options:

Use reserved EC2 instances for the production application and spot block instances for the dev application - Spot blocks can only be used for a span of up to 6 hours, so this option does not meet the requirements of the given use case where the dev application can be up and running up to 8 hours. You should also note that AWS has stopped offering Spot blocks to new customers.

Use reserved EC2 instances for the production application and spot instances for the dev application

Use on-demand EC2 instances for the production application and spot instances for the dev application

Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity in the AWS cloud. Spot Instances are available at up to a 90% discount compared to On-Demand prices. You can use Spot Instances for various stateless, fault-tolerant, or flexible applications.



via - https://aws.amazon.com/ec2/spot/

Spot instances can be taken back by AWS with two minutes of notice, so spot instances cannot be reliably used for running the dev application (which can be up and running for up to 8 hours). So both these options are incorrect.

References:

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

https://aws.amazon.com/blogs/aws/new-ec2-spot-blocks-for-defined-duration-workloads/

https://aws.amazon.com/ec2/spot/

Question 50: **Skipped**

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?

- **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**

◯ **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** **(Correct)**

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/

Question 51: **Skipped**

A major bank is using 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 FIFO queue in batch mode of 4 messages per operation to process the messages at the peak rate** **(Correct)**

◯ **Use Amazon SQS FIFO queue in batch mode of 2 messages per operation to process the messages at the peak rate**

◯ **Use Amazon SQS FIFO queue to process the messages**

◯ **Use Amazon SQS standard queue to process the messages**

Explanation

Correct option:

Use Amazon SQS FIFO 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:

FIFO (First-In-First-Out) queues are designed to enhance messaging between applications when the order of operations and events is critical, or where duplicates can't be tolerated, for example:

- Ensure that user-entered commands are executed 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.

FIFO queues also provide exactly-once processing but have a limited number of transactions per second (TPS):

- If you use batching, FIFO queues support up to 3,000 transactions per second, per API method (`SendMessageBatch`, `ReceiveMessage`, or `DeleteMessageBatch`). The 3000 transactions 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 (`SendMessage`, `ReceiveMessage`, or `DeleteMessage`).

Note

- Amazon SNS isn't currently compatible with FIFO queues.
- The name of a FIFO queue must end with the `.fifo` suffix. The suffix counts towards the 80-character queue name quota. To determine whether a queue is FIFO, you can check whether the queue name ends with the suffix.

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 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 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/

Question 52: **Skipped**

One of the biggest football leagues in Europe has granted the distribution rights for live streaming its matches in the US to a silicon valley based streaming services company. As per the terms of distribution, the company must make sure that only users from the US 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)

Use Route 53 based weighted routing policy to restrict 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 CloudFront web distribution	**(Correct)**
Use Route 53 based geolocation routing policy to restrict distribution of content to only the locations in which you have distribution rights	**(Correct)**

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

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

Explanation

Correct options:

Use 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 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 CloudFront web distribution. When a user requests your content, 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.

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 Route 53 based latency 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 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 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 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

Question 53: **Skipped**

A social photo-sharing company uses Amazon S3 to store the images uploaded by the users. These images are kept encrypted in S3 by using AWS-KMS and the company manages its own Customer Master Key (CMK) for encryption. A member of the DevOps team accidentally deleted the CMK a day ago, thereby rendering the user's photo data unrecoverable. You have been contacted by the company to consult them on possible solutions to this crisis.

As a solutions architect, which of the following steps would you recommend to solve this issue?

- ○ **The company should issue a notification on its web application informing the users about the loss of their data**
- ○ **The CMK can be recovered by the AWS root account user**
- ○ **Contact AWS support to retrieve the CMK from their backup**
- ○ **As the CMK was deleted a day ago, it must be in the 'pending deletion' status and hence you can just cancel the CMK deletion and recover the key** **(Correct)**

Explanation

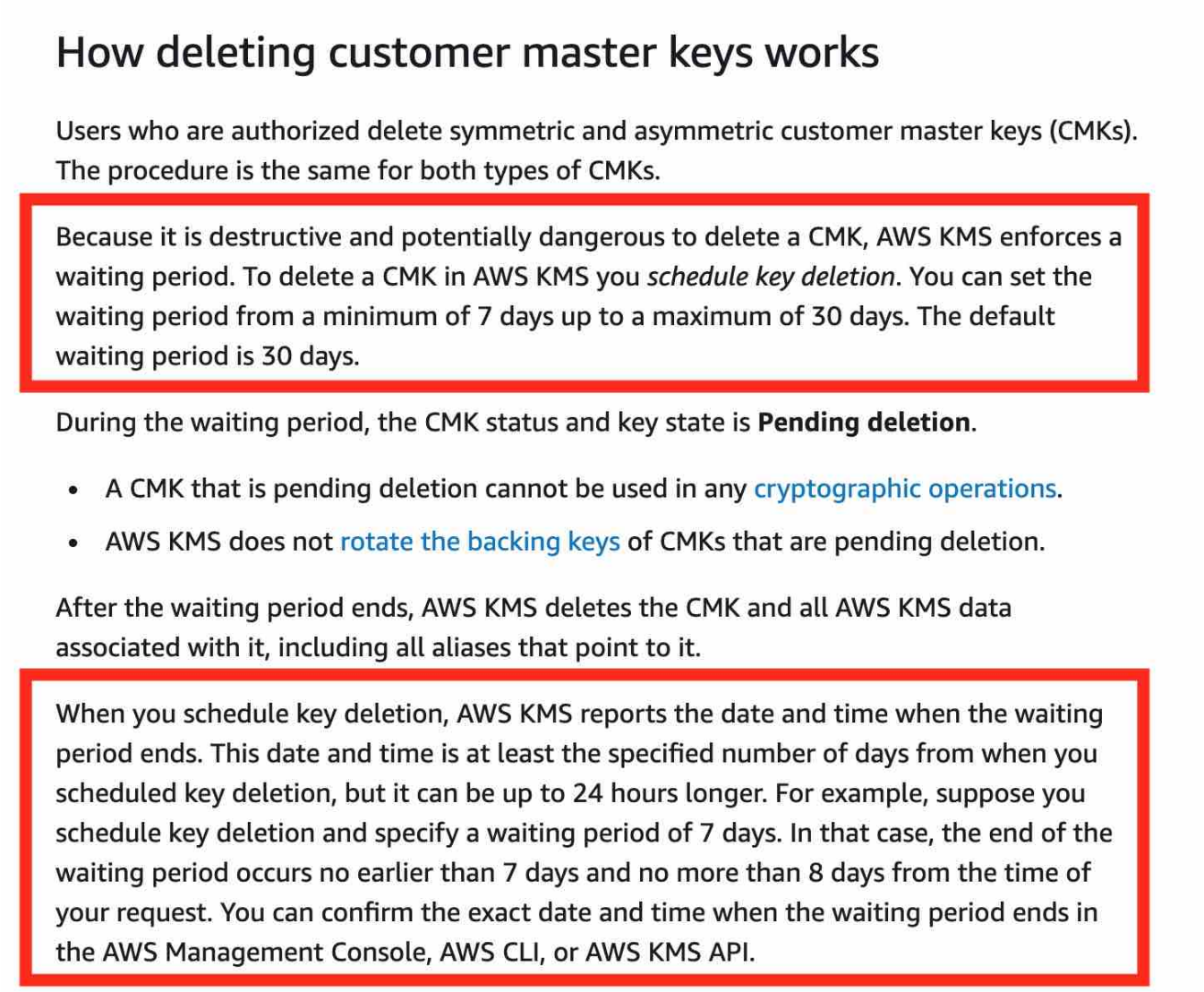
Correct option:

As the CMK was deleted a day ago, it must be in the 'pending deletion' status and hence you can just cancel the CMK deletion and recover the key

AWS Key Management Service (KMS) makes it easy for you to create and manage cryptographic keys and control their use across a wide range of AWS services and in your applications. AWS KMS is a secure and resilient service that uses hardware security modules that have been validated under FIPS 140-2.

Deleting a customer master key (CMK) in AWS Key Management Service (AWS KMS) is destructive and potentially dangerous. Therefore, AWS KMS enforces a waiting period. To delete a CMK in AWS KMS you schedule key deletion. You can set the waiting period from a minimum of 7 days up to a maximum of 30 days. The default waiting period is 30 days. During the waiting period, the CMK status and key state is Pending deletion. To recover the CMK, you can cancel key deletion before the waiting period ends. After the waiting period ends you cannot cancel key deletion, and AWS KMS deletes the CMK.

How Deleting Customer Master Keys Works:



How deleting customer master keys works

Users who are authorized delete symmetric and asymmetric customer master keys (CMKs). The procedure is the same for both types of CMKs.

Because it is destructive and potentially dangerous to delete a CMK, AWS KMS enforces a waiting period. To delete a CMK in AWS KMS you *schedule key deletion*. You can set the waiting period from a minimum of 7 days up to a maximum of 30 days. The default waiting period is 30 days.

During the waiting period, the CMK status and key state is **Pending deletion**.

- A CMK that is pending deletion cannot be used in any cryptographic operations.
- AWS KMS does not rotate the backing keys of CMKs that are pending deletion.

After the waiting period ends, AWS KMS deletes the CMK and all AWS KMS data associated with it, including all aliases that point to it.

When you schedule key deletion, AWS KMS reports the date and time when the waiting period ends. This date and time is at least the specified number of days from when you scheduled key deletion, but it can be up to 24 hours longer. For example, suppose you schedule key deletion and specify a waiting period of 7 days. In that case, the end of the waiting period occurs no earlier than 7 days and no more than 8 days from the time of your request. You can confirm the exact date and time when the waiting period ends in the AWS Management Console, AWS CLI, or AWS KMS API.

via - https://docs.aws.amazon.com/kms/latest/developerguide/deleting-keys.html

Incorrect options:

Contact AWS support to retrieve the CMK from their backup

The CMK can be recovered by the AWS root account user

The AWS root account user cannot recover CMK and the AWS support does not have access to CMK via any backups. Both these options just serve as distractors.

The company should issue a notification on its web application informing the users about the loss of their data - This option is not required as the data can be recovered via the cancel key deletion feature.

Reference:

https://docs.aws.amazon.com/kms/latest/developerguide/deleting-keys.html

Question 54: **Skipped**

A large financial institution operates an on-premises data center with hundreds of PB 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?

- Microsoft SQL Server on Amazon
- Amazon FSx for Lustre
- AWS Managed Microsoft AD
- Amazon FSx for Windows File Server (Correct)

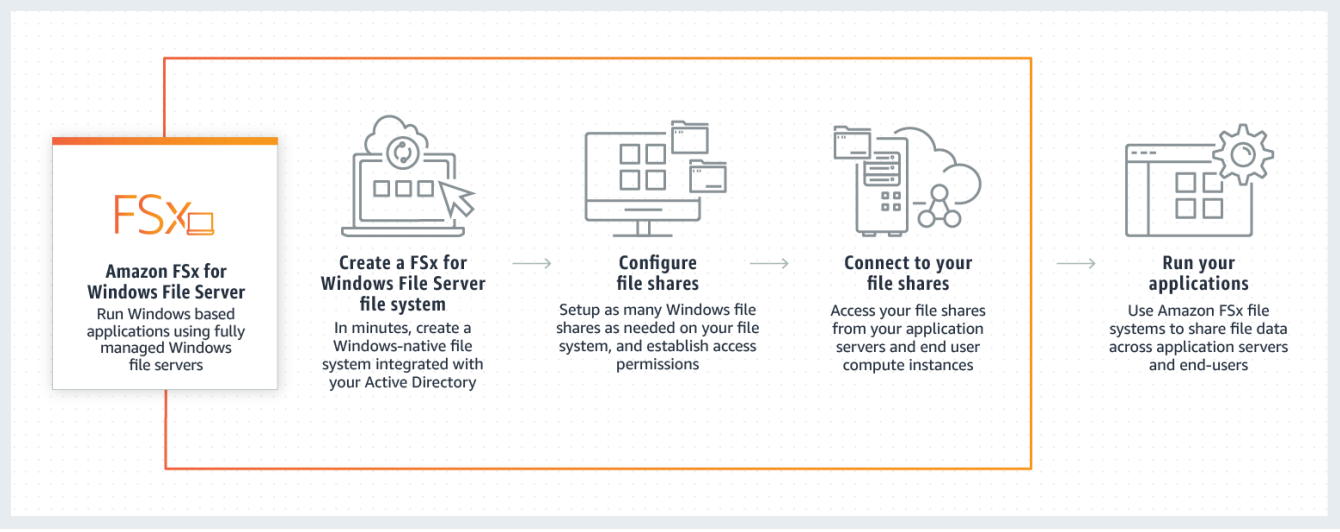
Explanation

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 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 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 Amazon

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/

Question 55: **Skipped**

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)

Create an event trigger on deleting any S3 object. The event invokes an SNS notification via email to the IT manager

Establish a process to get managerial approval for deleting S3 objects

Enable versioning on the bucket **(Correct)**

Enable MFA delete on the bucket **(Correct)**

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

Explanation

Correct options:

Enable versioning on the 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.

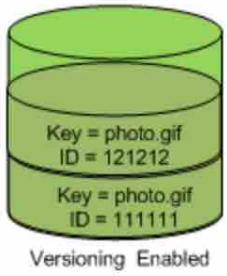
Versioning Overview:

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 MFA delete on the 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 S3 object. The event invokes an 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 S3 objects - This option for getting managerial approval is just a distractor.

Change the configuration on AWS S3 console so that the user needs to provide additional confirmation while deleting any S3 object - There is no provision to set up 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

Question 56: **Skipped**

The product team at a startup has figured out a market need to support both stateful and stateless client-server communications via the 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 AWS API Gateway.

Which of the following would you identify as correct?

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

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

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

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

Explanation

Correct option:

API Gateway creates RESTful APIs that enable stateless client-server communication and 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 API Gateway Works:

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

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.

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

Incorrect options:

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

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

API Gateway creates RESTful APIs that enable stateful client-server communication and 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, 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

Question 57: **Skipped**

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 SQS standard queue, which is polled by an application running on an EC2 instance and the data is written into an auto-scaled DynamoDB table for downstream processing**

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

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

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

Explanation

Correct option:

Ingest the sensor data in an Amazon SQS standard queue, which is polled by a Lambda function in batches and the data is written into an auto-scaled 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 Kinesis Data Firehose, which directly writes the data into an auto-scaled 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 SQS standard queue, which is polled by an application running on an EC2 instance and the data is written into an auto-scaled DynamoDB table for downstream processing

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

Using an application on an 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/

Question 58: **Skipped**

The payroll department at a company initiates several computationally intensive workloads on 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 EC2 instances and making sure that 10 EC2 instances are available during this peak usage hour. For normal operations only 2 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 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**

- ○ **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 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** **(Correct)**

- ○ **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**

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

Question 59: **Skipped**

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.

As a solutions architect, which best practices would you recommend (Select two)?

- [ ] **Configure AWS CloudTrail to log all IAM actions** **(Correct)**

- [ ] **Enable MFA for privileged users** **(Correct)**

- [ ] **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

Explanation

Correct options:

Enable 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 record all account activity - AWS recommends to turn on 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/

Question 60: **Skipped**

A financial services company uses Amazon GuardDuty for analyzing its AWS account metadata to meet the compliance guidelines. However, the company has now decided to stop using GuardDuty service. All the existing findings have to be deleted and cannot persist anywhere on AWS Cloud.

Which of the following techniques will help the company meet this requirement?

- ○ Raise a service request with Amazon to completely delete the data from all their backups
- ○ De-register the service under services tab
- ○ Suspend the service in the general settings

○ **Disable the service in the general settings** **(Correct)**

Explanation

Correct option:

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.

Disable the service in the general settings - Disabling the service will delete all remaining data, including your findings and configurations before relinquishing the service permissions and resetting the service. So, this is the correct option for our use case.

Incorrect options:

Suspend the service in the general settings - You can stop Amazon GuardDuty from analyzing your data sources at any time by choosing to suspend the service in the general settings. This will immediately stop the service from analyzing data, but does not delete your existing findings or configurations.

De-register the service under services tab - This is a made-up option, used only as a distractor.

Raise a service request with Amazon to completely delete the data from all their backups - There is no need to create a service request as you can delete the existing findings by disabling the service.

Reference:

https://aws.amazon.com/guardduty/faqs/

Question 61: **Skipped**

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?

○ **Leverage multi-AZ configuration of RDS for Oracle that allows the database administrators to access and customize the database environment and the underlying operating system**

○ **Leverage cross AZ read-replica configuration of RDS for Oracle that allows the database administrators to access and customize the database environment and the underlying operating system**

○ **Deploy the Oracle database layer on multiple EC2 instances spread across two Availability Zones (AZ). This deployment configuration**

guarantees high availability and also allows the database administrators to access and customize the database environment and the underlying operating system

○ **Leverage multi-AZ configuration of RDS Custom for Oracle that allows the database administrators to access and customize the database environment and the underlying operating system** **(Correct)**

Explanation

Correct option:

Leverage multi-AZ configuration of RDS Custom for Oracle that allows the database administrators 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 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. 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.

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 RDS for Oracle that allows the database administrators to access and customize the database environment and the underlying operating system

Leverage cross AZ read-replica configuration of RDS for Oracle that allows the database administrators to access and customize the database environment and the underlying operating system

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 EC2 instances spread across two Availability Zones (AZ). This deployment configuration guarantees high availability and also allows the database administrators 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 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 EC2 instances across two Availability Zones, 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/

Question 62: **Skipped**

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 GB.

Which of the following is the fastest way to upload the daily compressed file into S3?

- ○ **Upload the compressed file in a single operation**
- ○ **FTP the compressed file into an EC2 instance that runs in the same region as the S3 bucket. Then transfer the file from the EC2 instance into the S3 bucket**
- ○ **Upload the compressed file using multipart upload with S3 transfer acceleration** **(Correct)**
- ○ **Upload the compressed file using multipart upload**

Explanation

Correct option:

Upload the compressed file using multipart upload with S3 transfer acceleration

Amazon S3 Transfer Acceleration 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 MB, 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 S3 transfer acceleration would further improve the transfer speed. Therefore just using multipart upload is not the correct option.

FTP the compressed file into an EC2 instance that runs in the same region as the S3 bucket. Then transfer the file from the EC2 instance into the S3 bucket - This is a roundabout process of getting the file into 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 S3.

References:

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

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

Question 63: **Skipped**

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 1GB with some random data. Next he copies this test file into AWS S3 Standard storage class, provisions an EBS volume (General Purpose SSD (gp2)) with 100GB of provisioned storage and copies the test file into the EBS volume, and lastly copies the test file into an 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 S3 Standard < Cost of test file storage on EFS < Cost of test file storage on EBS** **(Correct)**
- ◯ **Cost of test file storage on S3 Standard < Cost of test file storage on EBS < Cost of test file storage on EFS**
- ◯ **Cost of test file storage on EFS < Cost of test file storage on S3 Standard < Cost of test file storage on EBS**
- ◯ **Cost of test file storage on EBS < Cost of test file storage on S3 Standard < Cost of test file storage on EFS**

Explanation

Correct option:

Cost of test file storage on S3 Standard < Cost of test file storage on EFS < Cost of test file storage on EBS

With Amazon EFS, you pay only for the resources that you use. The 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 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*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 S3 Standard < Cost of test file storage on EBS < Cost of test file storage on EFS

Cost of test file storage on EFS < Cost of test file storage on S3 Standard < Cost of test file storage on EBS

Cost of test file storage on EBS < Cost of test file storage on S3 Standard < Cost of test file storage on 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/

Question 64: **Skipped**

A media agency stores its re-creatable assets on 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 S3 storage is turning out to be very expensive and the agency is looking at reducing costs as much as possible.

As a Solutions Architect, can you suggest a way to lower the storage costs while fulfilling the business requirements?

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

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 - 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. 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 S3 Standard or S3 Standard-IA. The minimum storage duration is 30 days before you can transition objects from S3 Standard to S3 One Zone-IA.

S3 One Zone-IA offers the same high durability, high throughput, and low latency of 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 S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, and 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.

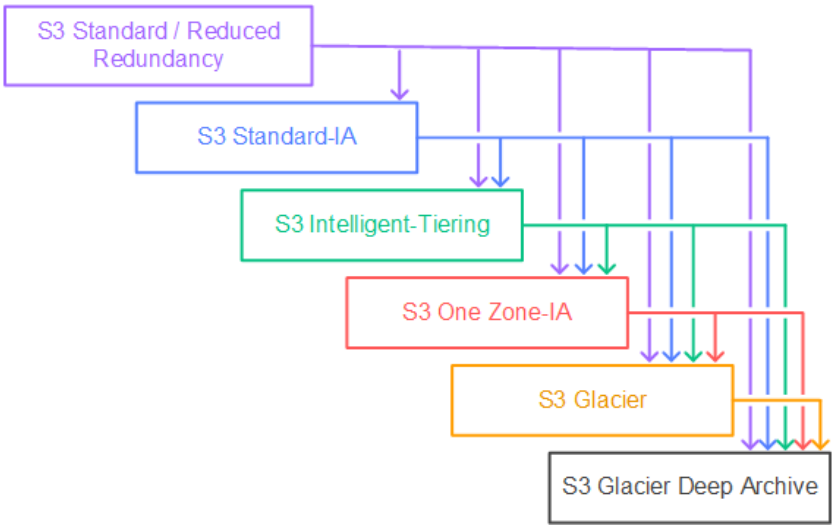
Similarly, if you are transitioning noncurrent objects (in versioned buckets), you can transition only objects that are at least 30 days noncurrent to S3 Standard-IA or S3 One Zone-IA storage.

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 S3 lifecycle transitions: 

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 S3 Standard to S3 One Zone-IA or 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 - 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 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. But, it costs more than S3 One Zone-IA because of the redundant storage across availability zones. 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

Question 65: **Skipped**

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 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 EC2 instances spread across different Availability Zones. Configure a Public IP address to mask any failure of an instance**

○ **Use an Auto Scaling group for distributing traffic to the EC2 instances spread across different Availability Zones. Configure an Elastic IP address to mask any failure of an instance**

○ **Use an Application Load Balancer for distributing traffic to the EC2 instances spread across different Availability Zones. Configure Auto Scaling group to mask any failure of an instance** **(Correct)**

○ **Use a Network Load Balancer for distributing traffic to the EC2 instances spread across different Availability Zones. Configure a Private IP address to mask any failure of an instance**

Explanation

Correct option:

Use an Application Load Balancer for distributing traffic to the EC2 instances spread across different Availability Zones. 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.

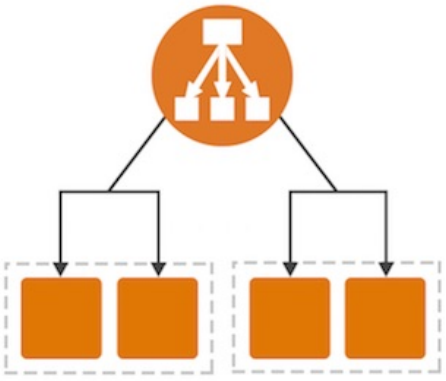
This is the correct option since the question has a specific requirement for content-based routing which can be configured via the Application Load Balancer. Different AZs provide high availability to the overall architecture and Auto Scaling group will help mask any instance failures.

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.

- https://aws.amazon.com/blogs/aws/new-aws-application-load-balancer/ via

Incorrect options:

Use a Network Load Balancer for distributing traffic to the EC2 instances spread across different Availability Zones. 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 EC2 instances spread across different Availability Zones. Configure an Elastic IP address to mask any failure of an instance

Use an Auto Scaling group for distributing traffic to the EC2 instances spread across different Availability Zones. 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 EC2 instances.

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.

More info on Elastic Load Balancer:

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.

via

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.

- 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 within a 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-zone.html

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