DEMO

Exam Dumps

AWS Certified Solutions Architect Associate

AWS Certified Solutions Architect Associate Exam Dumps
CLOUD EXAM DUMPS

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

Return to review
Chart
Pie chart with 3 slices.
End of interactive chart.
Attempt 1
All knowledge areas
All questions
Question 1:
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

- ○

  Amazon SQS, Amazon SNS and AWS Lambda

- ○

  Elastic Load Balancer, Amazon SQS, 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 2:

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?

- ○

  Configure AWS WAF on the Application Load Balancer in a VPC

  (Correct)

- ○

  Configure the security group for the EC2 instances

- ○

  Use Geo Restriction feature of Amazon CloudFront in a VPC

- ○

  Configure the security group on the Application Load Balancer

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 3:

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 Global Accelerator

- □

  Use AWS Shield

- □

  Use Aurora Replica

  (Correct)

- □

  Use AWS Direct Connect

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 4:

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 DynamoDB tables 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

- ○

  Use an Amazon Aurora Global Database for the `games` table and use Amazon Aurora for the `users` and `games_played` tables

  (Correct)

- ○

  Use a DynamoDB global table for the `games` table and use Amazon Aurora 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 5:

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 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

- ○

  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 EFS. The applications deployed on AWS can access this data directly from Amazon EFS

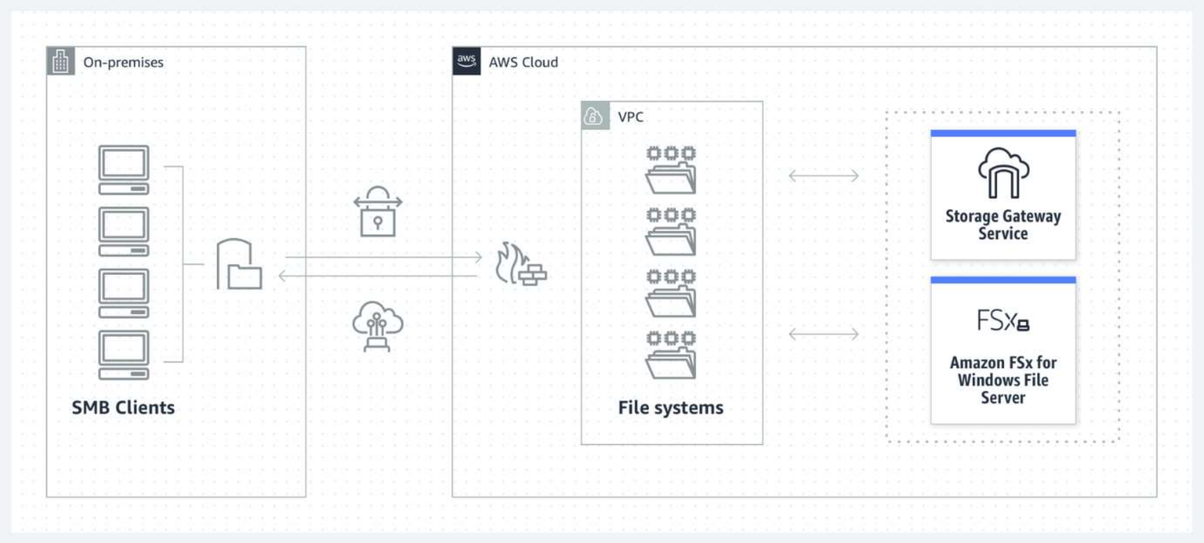
Explanation
Correct option:

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:



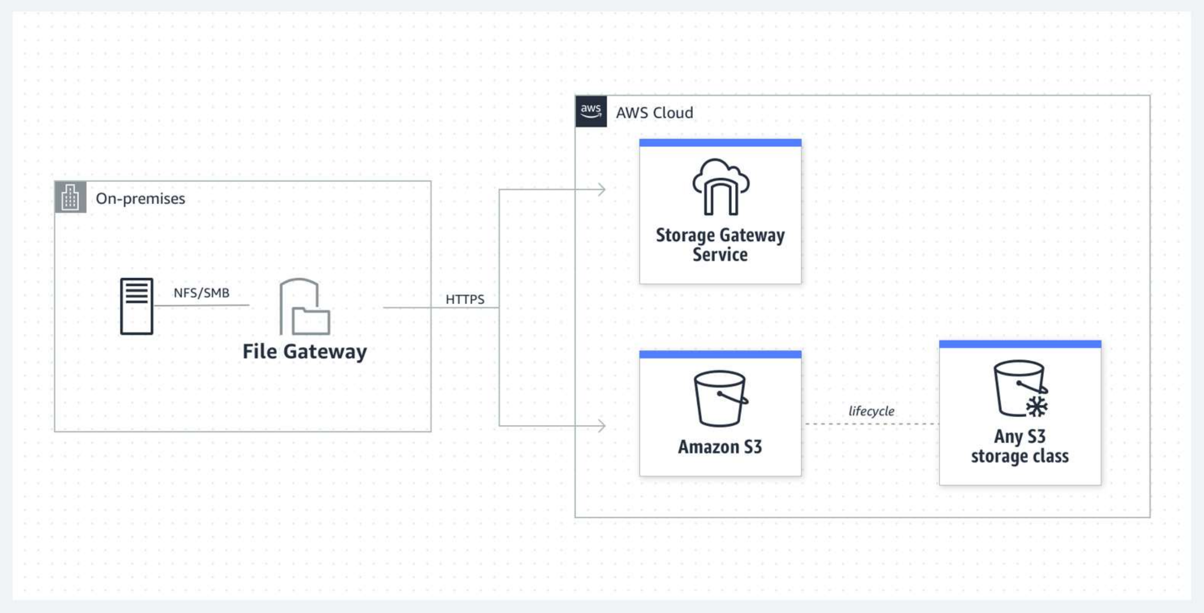
vi
a - https://aws.amazon.com/storagegateway/file/

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 6:

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)

- □

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

- □

  Enable MFA delete on the bucket

  (Correct)

- □

  Enable versioning on the bucket

  (Correct)

- □

  Establish a process to get managerial approval for deleting S3 objects

- □

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

Explanation

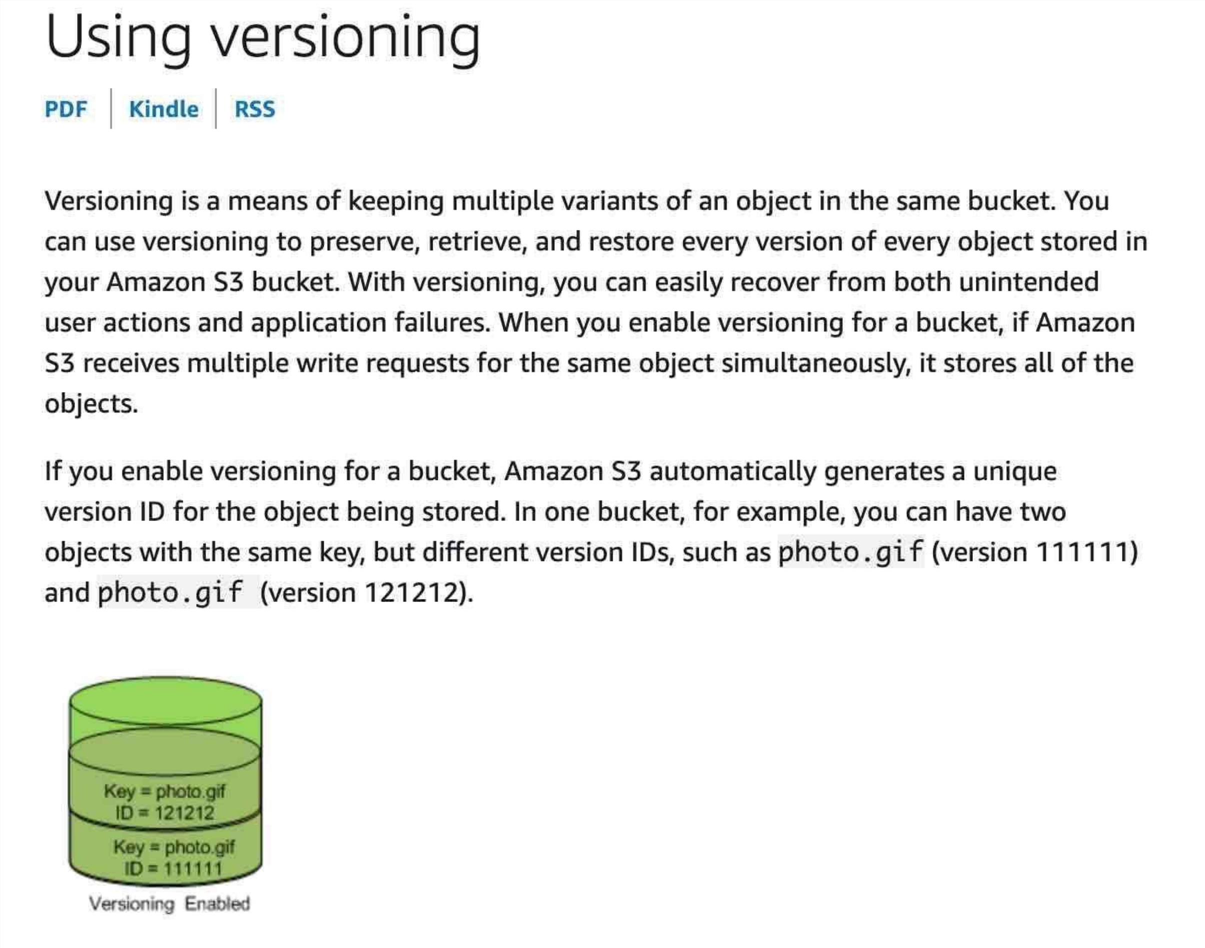
Correct options:

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

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).

vi
a - 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 7:

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 EFS mount points

- ○

  Use EBS based EC2 instances

- ○

  Use EC2 instances with access to S3 based storage

- ○

  Use Instance Store based EC2 instances

  (Correct)

Explanation

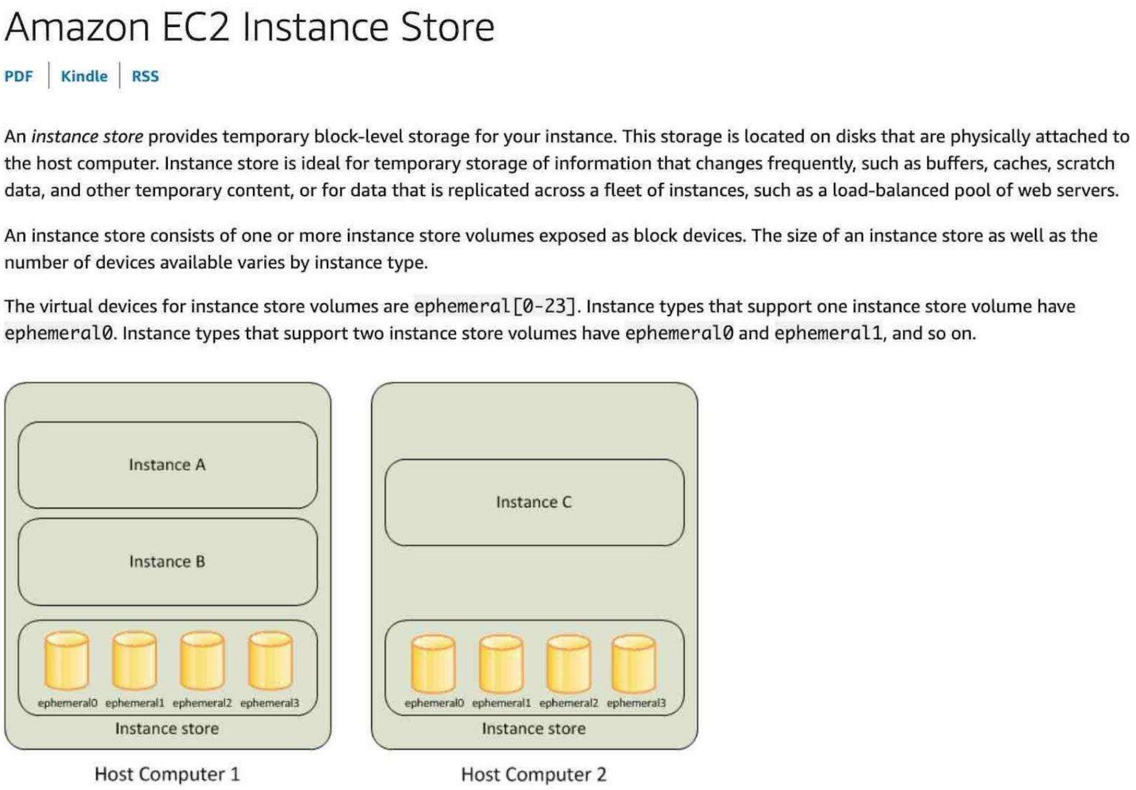
Correct option:

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 computer. 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.

vi
a - 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. 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 8:

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 Intelligent-Tiering => S3 Standard

  (Correct)

- □

  S3 Standard => S3 Intelligent-Tiering

- □

  S3 Standard-IA => S3 One Zone-IA

- □

  S3 Standard-IA => 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 9:
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 Intelligent-Tiering (S3 Intelligent-Tiering)

- ○

  Amazon S3 Standard

- ○

  Amazon S3 Glacier Deep Archive

- ○

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

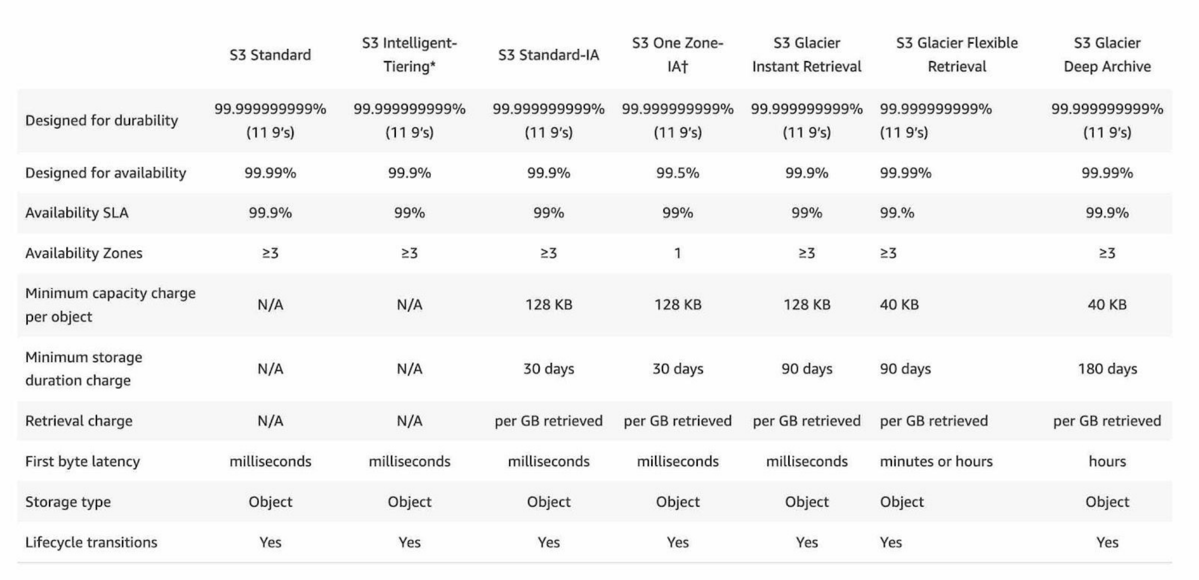
  (Correct)

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

vi
a - 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 10:

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 Glacier Instant Retrieval storage class

- ○

  Store the intermediary query results in S3 Standard storage class

  (Correct)

- ○

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

- ○

  Store the intermediary query results in S3 One Zone-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 11:

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?

- ○

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)

- ○

  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

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 12:
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?

- ○

  Path-based Routing

  (Correct)
- ○

  Query string parameter-based routing
- ○

  HTTP header-based routing
- ○

  Host-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.

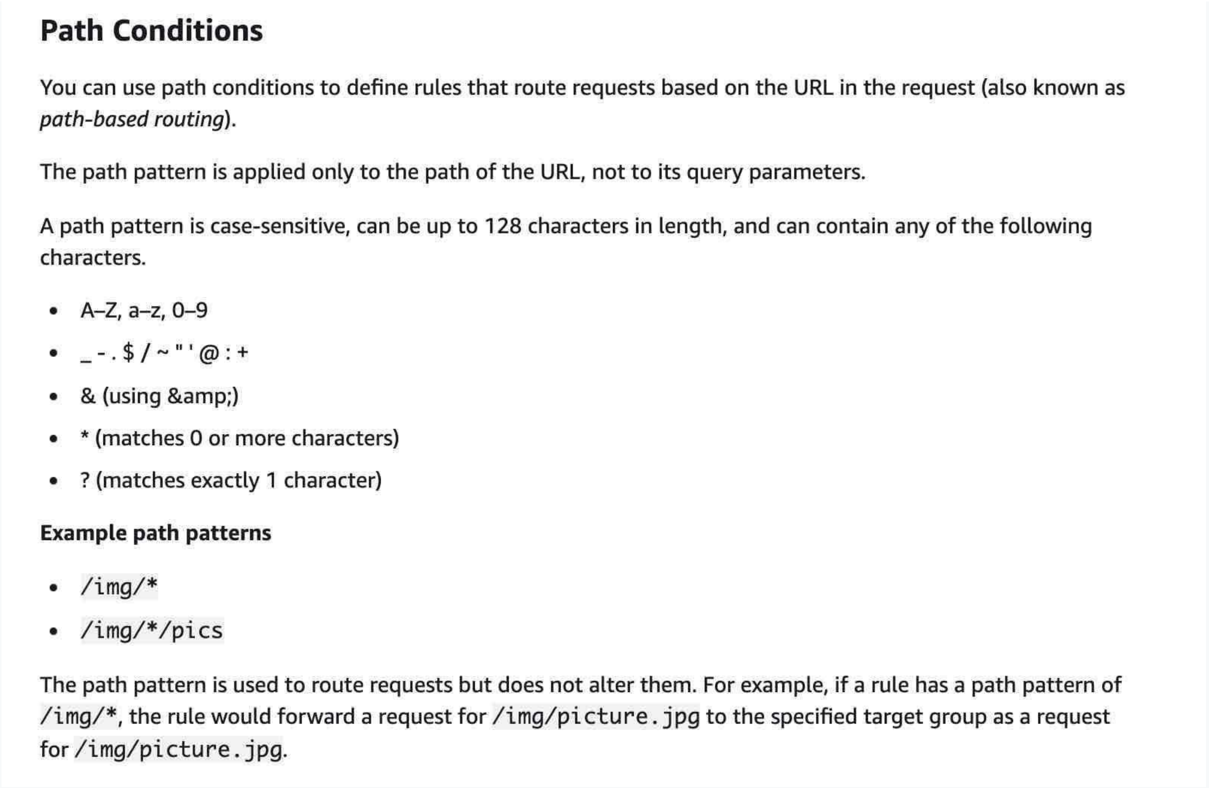
Source IP address CIDR-based routing:

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 &amp;)
- * (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`.

vi

a - 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 13:

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 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 stateful, 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

- ○

  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

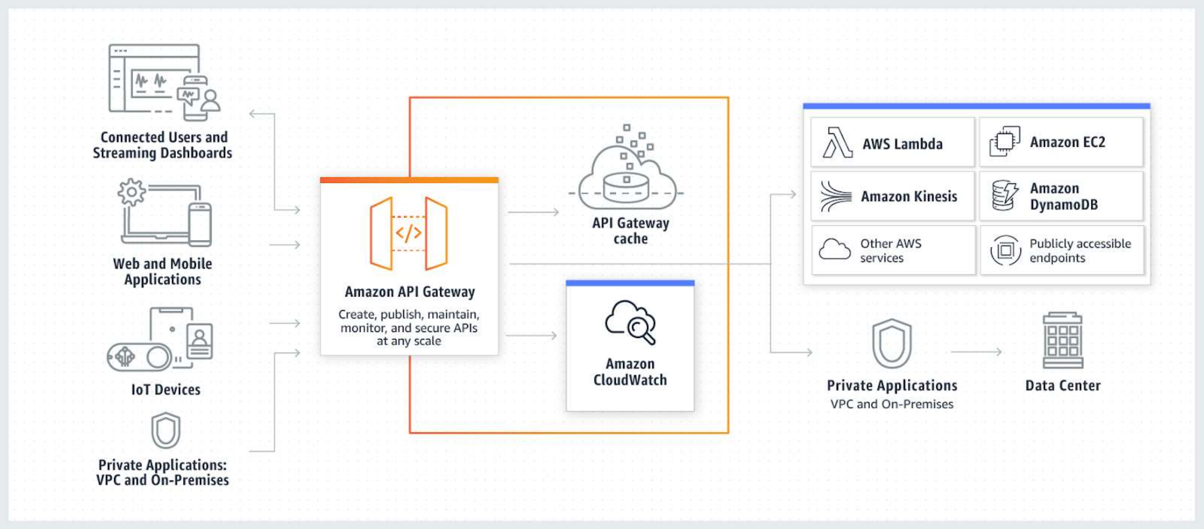
  (Correct)

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:


Connected Users and Streaming Dashboards
Web and Mobile Applications
IoT Devices
Private Applications: VPC and On-Premises
Amazon API Gateway
Create, publish, maintain, monitor, and secure APIs at any scale
API Gateway cache
Amazon CloudWatch
AWS Lambda
Amazon EC2
Amazon Kinesis
Amazon DynamoDB
Other AWS services
Publicly accessible endpoints
Private Applications
VPC and On-Premises
Data Center

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 14:

Skipped

An IT Company wants to move all the compute components of its AWS Cloud infrastructure into serverless architecture. Their development stack comprises a mix of backend programming languages and the company would like to explore the support offered by the AWS Lambda runtime for their programming languages stack.

Can you identify the programming languages supported by the Lambda runtime? (Select two)

- □

  R

- □

  PHP

- □

  C#/.NET

  (Correct)

- □

  Go

  (Correct)

- □

C

Explanation

Correct options:

C#/.NET

Go

A runtime is a version of a programming language or framework that you can use to write Lambda functions. AWS Lambda supports runtimes for the following languages:

C#/.NET

Go

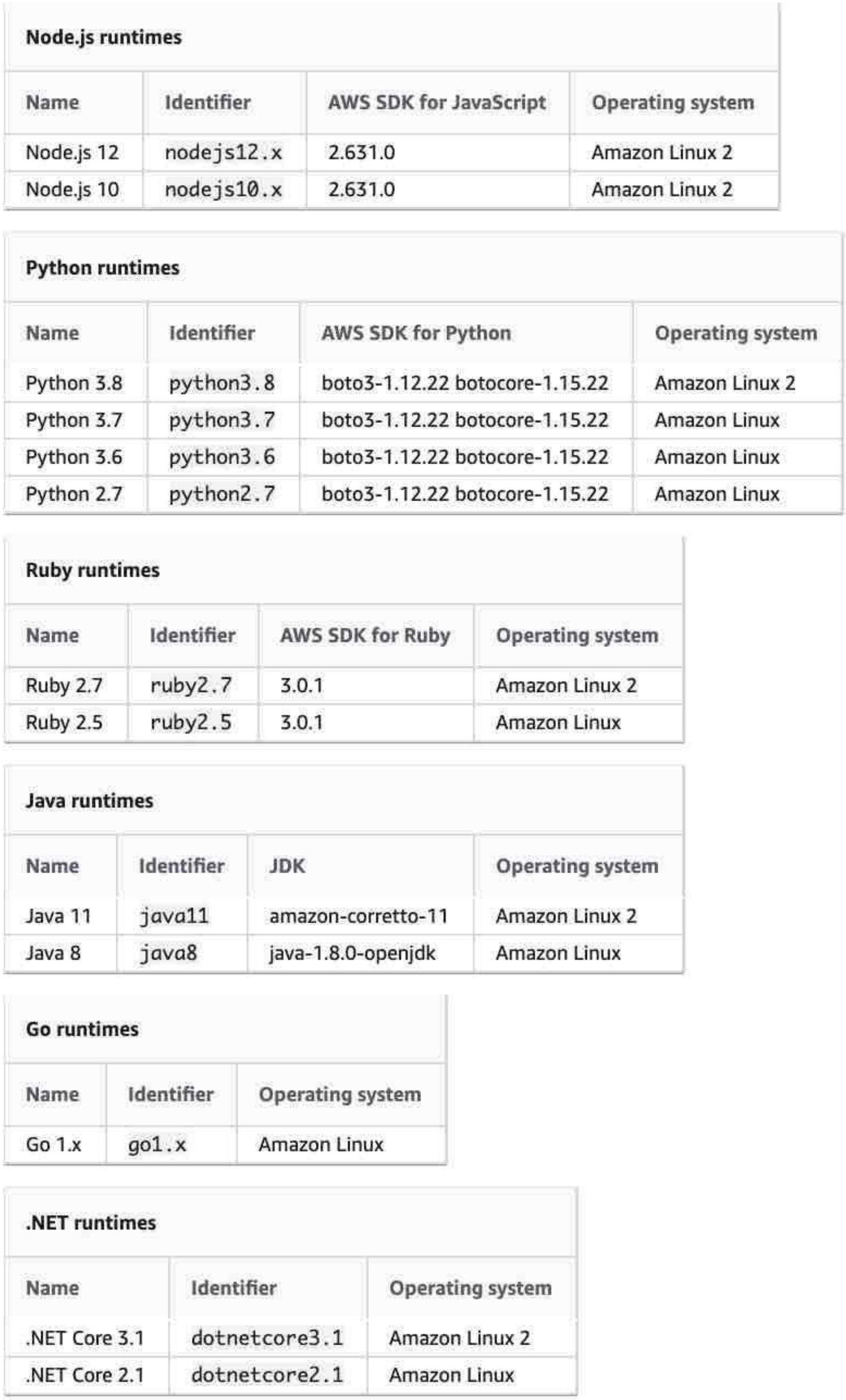
Java

Node.js

Python

Ruby

AWS Lambda
runtimes:



Node.js runtimes

Name	Identifier	AWS SDK for JavaScript	Operating system
Node.js 12	`nodejs12.x`	2.631.0	Amazon Linux 2
Node.js 10	`nodejs10.x`	2.631.0	Amazon Linux 2

Python runtimes

Name	Identifier	AWS SDK for Python	Operating system
Python 3.8	`python3.8`	boto3-1.12.22 botocore-1.15.22	Amazon Linux 2
Python 3.7	`python3.7`	boto3-1.12.22 botocore-1.15.22	Amazon Linux
Python 3.6	`python3.6`	boto3-1.12.22 botocore-1.15.22	Amazon Linux
Python 2.7	`python2.7`	boto3-1.12.22 botocore-1.15.22	Amazon Linux

Ruby runtimes

Name	Identifier	AWS SDK for Ruby	Operating system
Ruby 2.7	`ruby2.7`	3.0.1	Amazon Linux 2
Ruby 2.5	`ruby2.5`	3.0.1	Amazon Linux

Java runtimes

Name	Identifier	JDK	Operating system
Java 11	`java11`	amazon-corretto-11	Amazon Linux 2
Java 8	`java8`	java-1.8.0-openjdk	Amazon Linux

Go runtimes

Name	Identifier	Operating system
Go 1.x	`go1.x`	Amazon Linux

.NET runtimes

Name	Identifier	Operating system
.NET Core 3.1	`dotnetcore3.1`	Amazon Linux 2
.NET Core 2.1	`dotnetcore2.1`	Amazon Linux

vi

a - https://docs.aws.amazon.com/lambda/latest/dg/lambda-runtimes.html

Incorrect options:

C

PHP

R

Given the list of supported runtimes above, these three options are incorrect.

Reference:

https://docs.aws.amazon.com/lambda/latest/dg/lambda-runtimes.html

Question 15:

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

- □

  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)

- □

  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)

- □

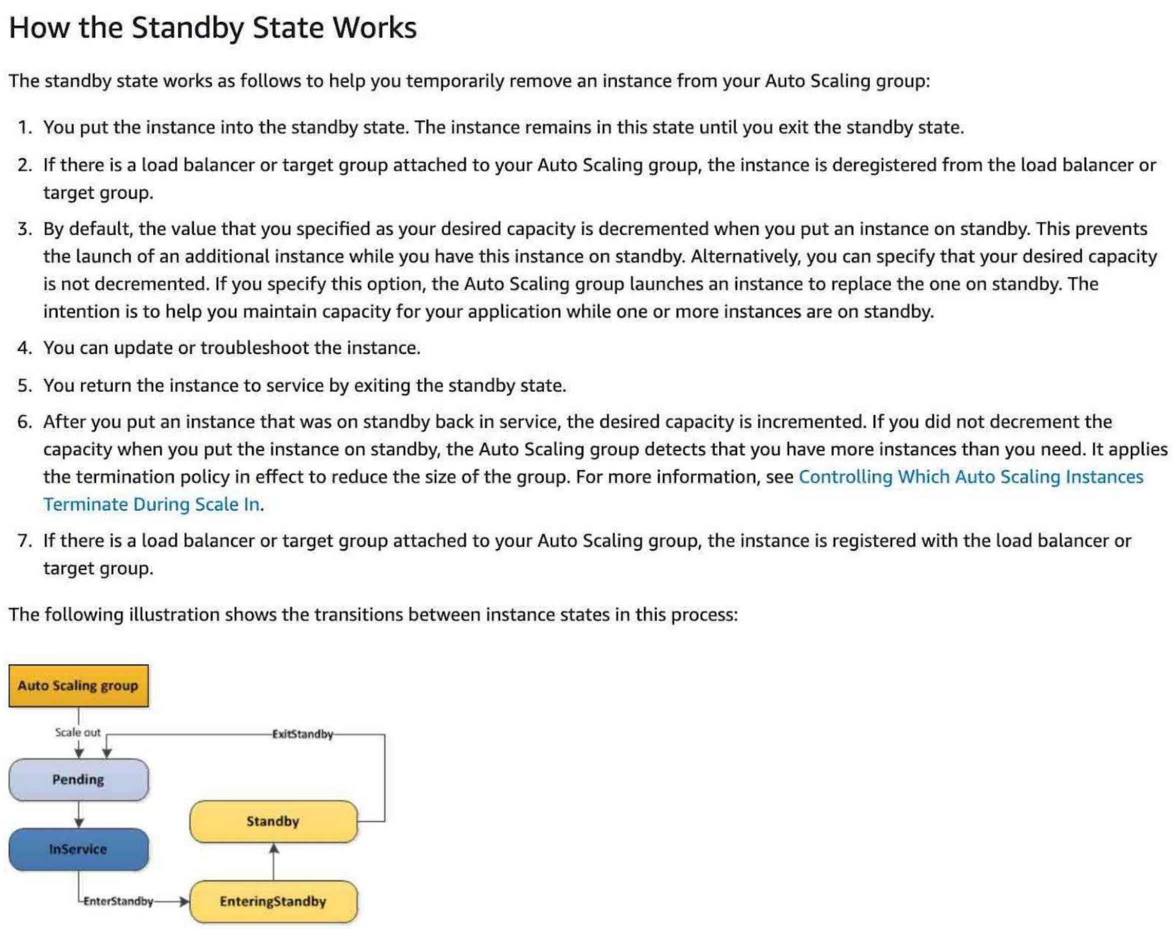
  Suspend the ScheduledActions process type for the Auto Scaling group and apply the maintenance patch to the instance. Once the instance is ready, you can you can manually set the instance's health status back to healthy and activate the ScheduledActions process type again

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:

vi
a - 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 16:

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

- ○

  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

- ○

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 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)

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 17:

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?

- ○

  Amazon FSx for Windows File Server

  (Correct)

- ○

  Amazon FSx for Lustre

- ○

  Microsoft SQL Server on Amazon

- **AWS Managed Microsoft AD**

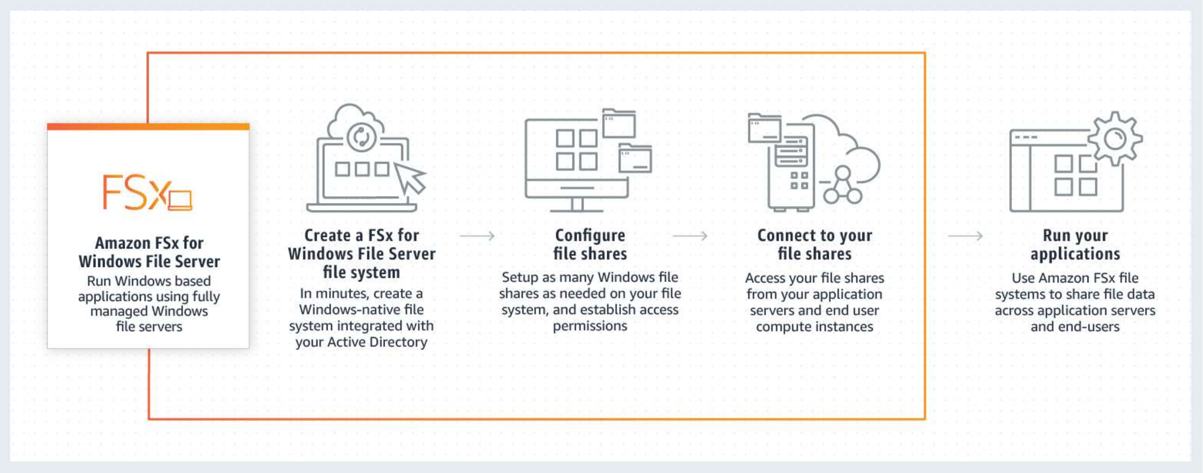
Explanation

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:


FSx
Amazon FSx for Windows File Server
Run Windows based applications using fully managed Windows file servers
Create a FSx for Windows File Server file system
In minutes, create a Windows-native file system integrated with your Active Directory
Configure file shares
Setup as many Windows file shares as needed on your file system, and establish access permissions
Connect to your file shares
Access your file shares from your application servers and end user compute instances
Run your applications
Use Amazon FSx file systems to share file data across application servers and end-users

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 18:

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 just charged based on Elastic Container Service used per hour

- ○

  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

- ○

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

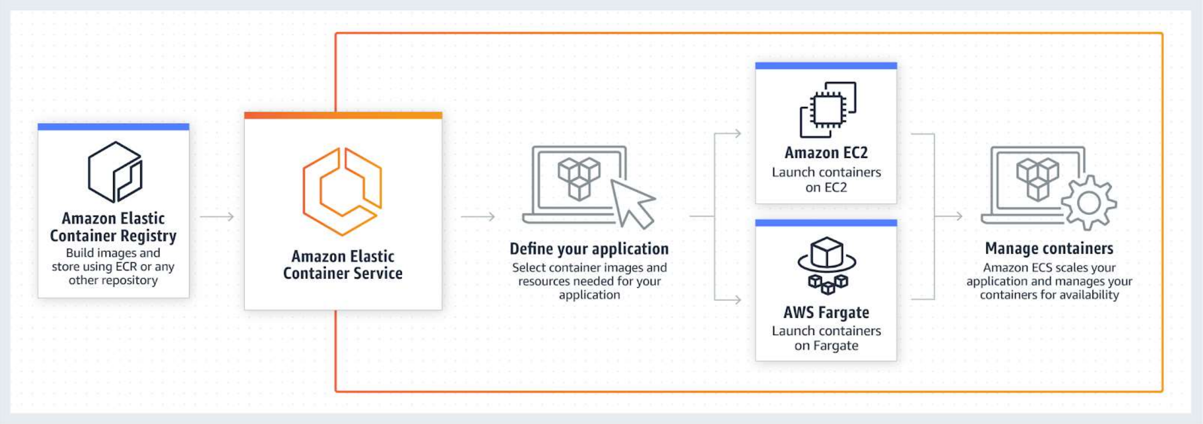
(Correct)

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:


Amazon Elastic Container Registry
Build images and store using ECR or any other repository
Amazon Elastic Container Service
Define your application
Select container images and resources needed for your application
Amazon EC2
Launch containers on EC2
AWS Fargate
Launch containers on Fargate
Manage containers
Amazon ECS scales your application and manages your containers for availability

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 19:

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 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

- ○

  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 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 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

Explanation

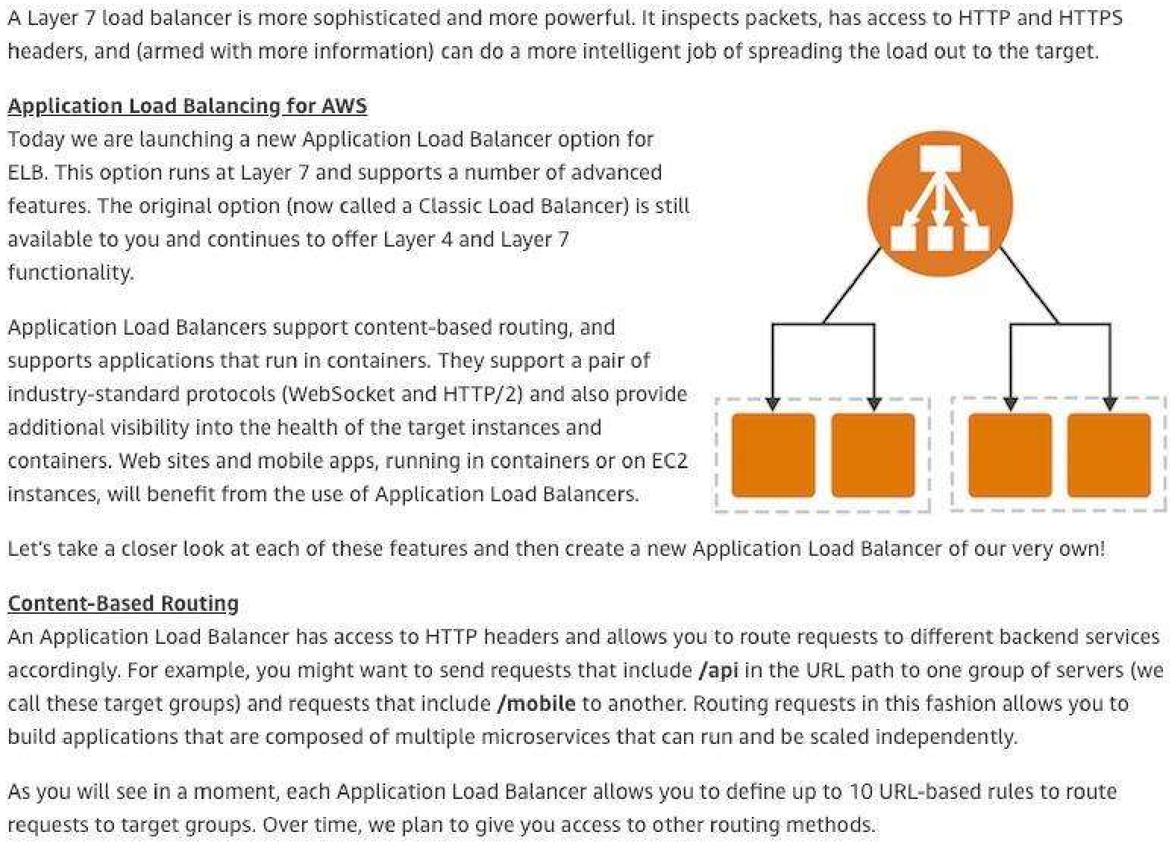
Correct option:

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.

vi

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

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.

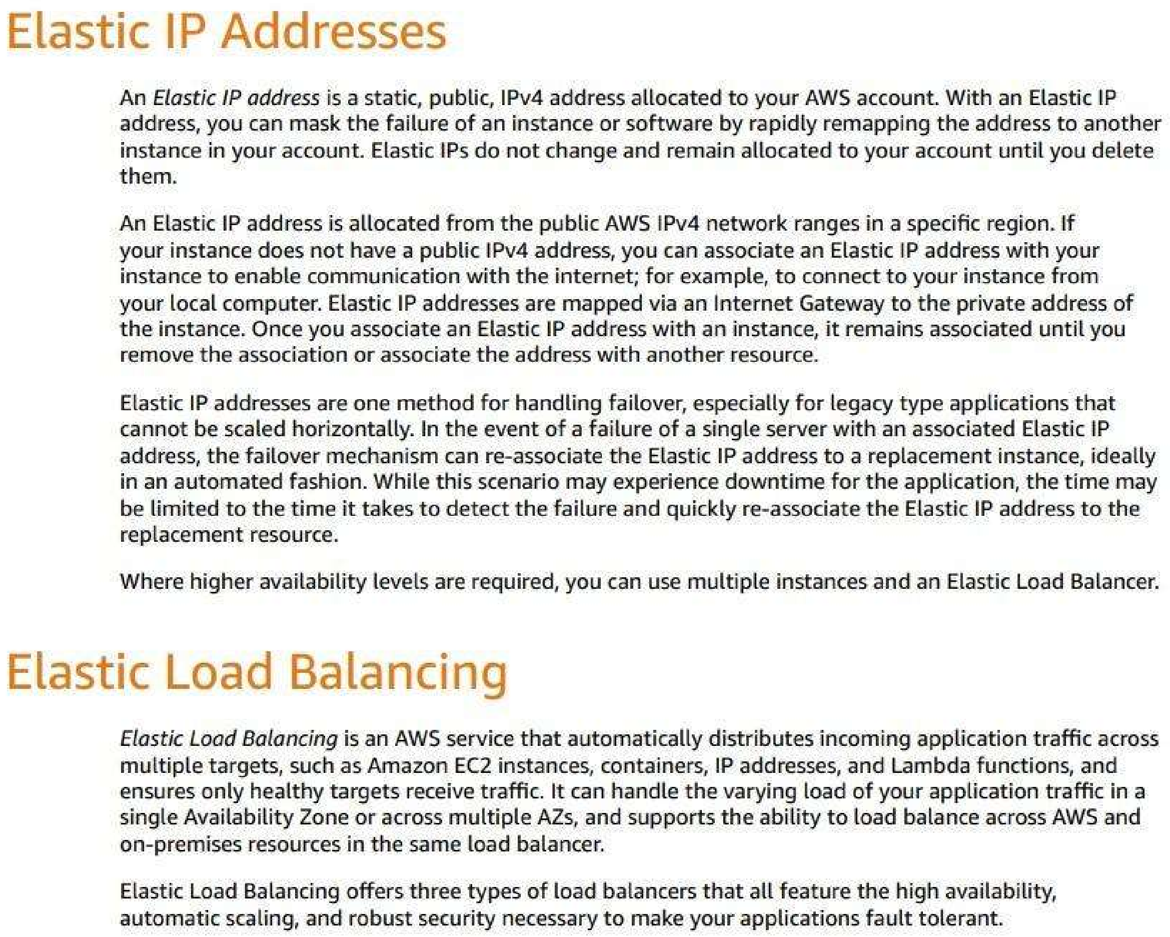
Use an Auto Scaling group for distributing traffic to the EC2 instances spread across different Availability Zones. Configure an Elastic IP address to mask any failure of an instance

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.

Elastic Load Balancing

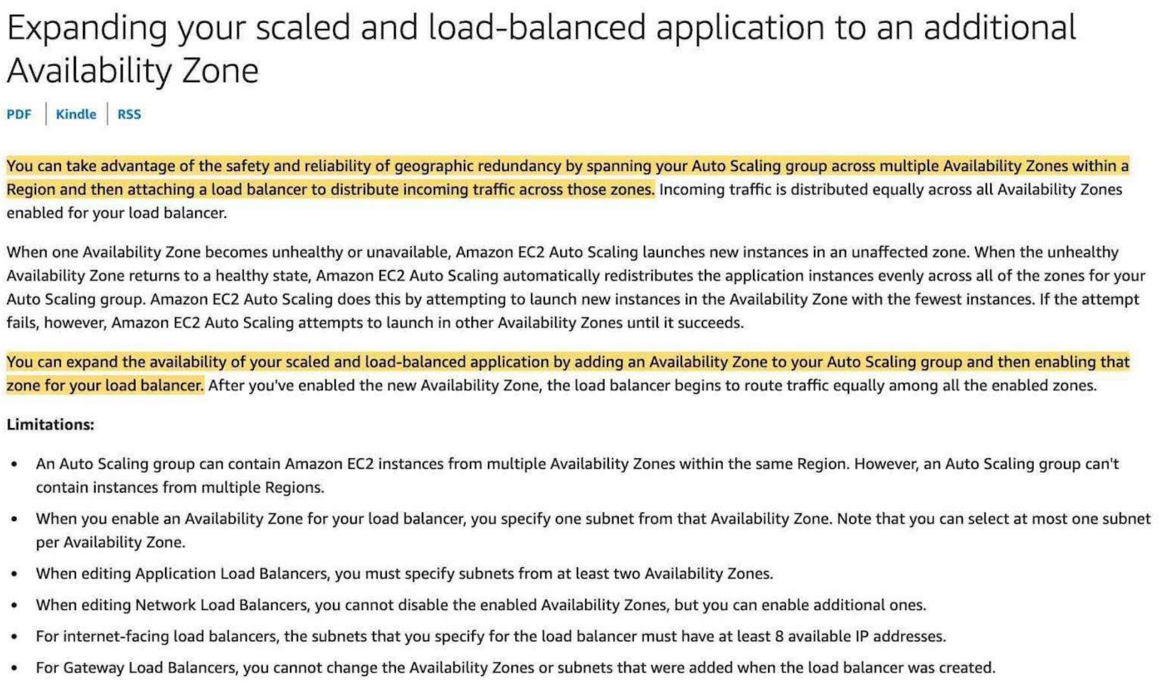
Elastic Load Balancing is an AWS service that automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions, and ensures only healthy targets receive traffic. It can handle the varying load of your application 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.

vi

a - 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.

vi

a - 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

Question 20:

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 - Tape Gateway

- ○

  AWS Storage Gateway - File Gateway

(Correct)

- ○

AWS Storage Gateway - Volume Gateway

- ○

AWS Site-to-Site VPN

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 21:

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

- □

  General Purpose SSD (gp2)

- □

  Provisioned IOPS SSD (io1)

- □

  Throughput Optimized HDD (st1)

  (Correct)

- □

Cold HDD (sc1)

(Correct)

Explanation
Correct options:

Throughput Optimized HDD (st1)

Cold HDD (sc1)

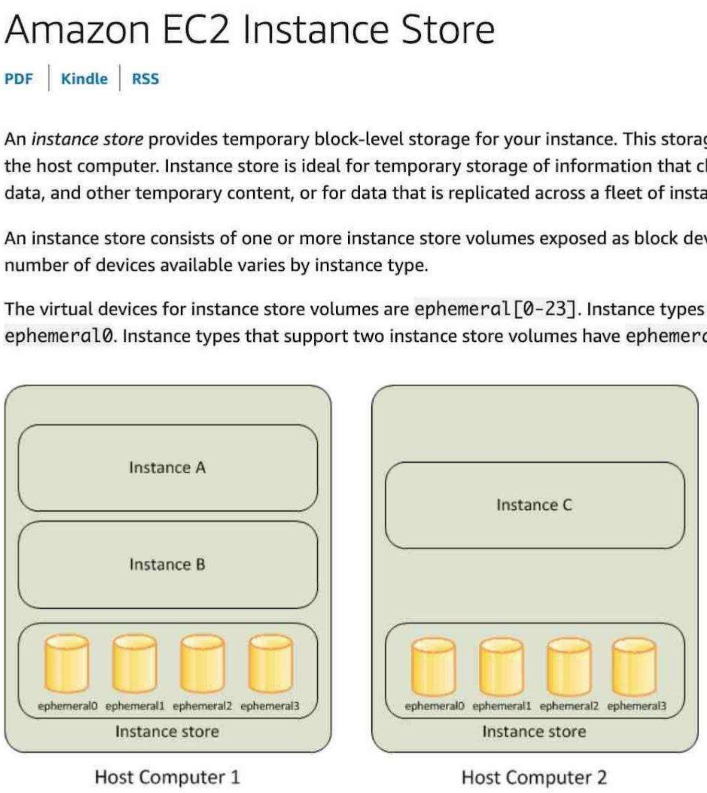
The EBS volume types fall into two categories:

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.



Amazon EC2 Instance Store

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.

vi

a - 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 22:

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 multipart uploads for faster file uploads into the destination S3 bucket

  (Correct)

- □

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

  (Correct)

- □

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

- □

  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

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 23:

Skipped

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

- ○

  Versioning

  (Correct)

- ○

  Server Access Logging

- ○

  Requester Pays

- ○

  Static Website Hosting

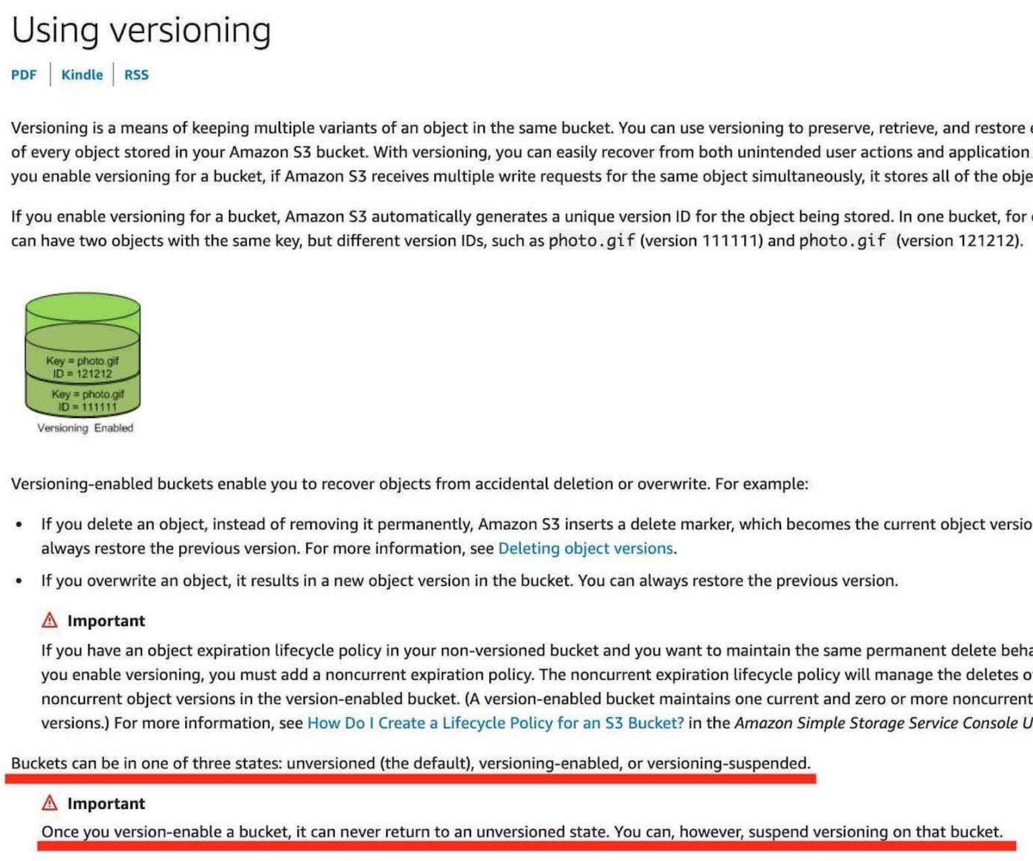
Explanation

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.

vi

a - 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 24:

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?

- ○

  AWS Elastic Load Balancing (ELB)

- ○

  Amazon CloudFront

- ○

  Amazon Route 53

- ○

  AWS Global Accelerator

  (Correct)

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 25:

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 on the EFS file system can be accessed in other AWS regions by using an inter-region VPC peering connection

(Correct)

- ○

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

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/