Question 1:
A medium-sized business has a taxi dispatch application deployed on an EC2 instance. Because of an unknown bug, the application causes the instance to freeze regularly. Then, the instance has to be manually restarted via the AWS management console.

Which of the following is the MOST cost-optimal and resource-efficient way to implement an automated solution until a permanent fix is delivered by the development team?

- Use CloudWatch events to trigger a Lambda function to reboot the instance status every 5 minutes
- Setup a CloudWatch alarm to monitor the health status of the instance. In case of an Instance Health Check failure, CloudWatch Alarm can publish to an SNS event which can then trigger a lambda function. The lambda function can use AWS EC2 API to reboot the instance
- Use CloudWatch events to trigger a Lambda function to check the instance status every 5 minutes. In the case of Instance Health Check failure, the lambda function can use AWS EC2 API to reboot the instance
- Setup a CloudWatch alarm to monitor the health status of the instance. In case of an Instance Health Check failure, an EC2 Reboot CloudWatch Alarm Action can be used to reboot the instance
  (Correct)

Explanation
Correct option:

Setup a CloudWatch alarm to monitor the health status of the instance. In case of an Instance Health Check failure, an EC2 Reboot CloudWatch Alarm Action can be used to reboot the instance

Using Amazon CloudWatch alarm actions, you can create alarms that automatically stop, terminate, reboot, or recover your EC2 instances. You can use the stop or terminate actions to help you save money when you no longer need an instance to be running. You can use the reboot and recover actions to automatically reboot those instances or recover them onto new hardware if a system impairment occurs.

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically reboots the instance. The reboot alarm action is recommended for Instance Health Check failures (as opposed to the recover alarm action, which is suited for System Health Check failures).

Incorrect options:

Setup a CloudWatch alarm to monitor the health status of the instance. In case of an Instance Health Check failure, CloudWatch Alarm can publish to an SNS event which can then trigger a lambda function. The lambda function can use AWS EC2 API to reboot the instance

Use CloudWatch events to trigger a Lambda function to check the instance status every 5 minutes. In the case of Instance Health Check failure, the lambda function can use AWS EC2 API to reboot the instance

Use CloudWatch events to trigger a Lambda function to reboot the instance status every 5 minutes

Using CloudWatch event or CloudWatch alarm to trigger a lambda function, directly or indirectly, is wasteful of resources. You should just use the EC2 Reboot CloudWatch Alarm Action to reboot the instance. So all the options that trigger the lambda function are incorrect.

Question 2:
An Internet-of-Things (IoT) company is looking for a database solution on AWS Cloud that has Auto Scaling capabilities and is highly available. The database should be able to handle any changes in data attributes over time, in case the company updates the data feed from its IoT devices. The database must provide the capability to output a continuous stream with details of any changes to the underlying data.

As a Solutions Architect, which database will you recommend?

- Amazon DynamoDB **(Correct)**
- Amazon Aurora
- Amazon Relational Database Service (Amazon RDS)
- Amazon Redshift

Explanation
Correct option:

Amazon DynamoDB - Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-Region, multi-master, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second. DynamoDB is serverless with no servers to provision, patch, or manage and no software to install, maintain, or operate.

A DynamoDB stream is an ordered flow of information about changes to items in a DynamoDB table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table.

Whenever an application creates, updates, or deletes items in the table, DynamoDB Streams writes a stream record with the primary key attributes of the items that were modified. A stream record contains information about a data modification to a single item in a DynamoDB table. You can configure the stream so that the stream records capture additional information, such as the "before" and "after" images of modified items.

DynamoDB is horizontally scalable, has a DynamoDB streams capability and is multi-AZ by default. On top of it, we can adjust the RCU and WCU automatically using Auto Scaling. This is the right choice for current requirements.

Incorrect options:

Amazon Relational Database Service (Amazon RDS) - Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. Schema changes on relational databases are not straight forward and are hard to maintain if the schema requirements change often.

Amazon Aurora - 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 64TB per database instance. Aurora is not an in-memory database. Schema changes on relational databases are not straight forward and are hard to maintain if the schema requirements change often.

Amazon Redshift - Amazon Redshift is a fully-managed petabyte-scale cloud based data warehouse product designed for large scale data set storage and analysis. It is a powerful warehousing service from Amazon. The current requirement, however, is not looking for a warehousing solution and hence Redshift is not an option here.

Question 3:
Which of the following is true regarding cross-zone load balancing as seen in Application Load Balancer versus Network Load Balancer?

- By default, cross-zone load balancing is disabled for both Application Load Balancer and Network Load Balancer
- By default, cross-zone load balancing is enabled for Application Load Balancer and disabled for Network Load Balancer **(Correct)**
- By default, cross-zone load balancing is disabled for Application Load Balancer and enabled for Network Load Balancer
- By default, cross-zone load balancing is enabled for both Application Load Balancer and Network Load Balancer

Explanation

Correct option:

By default, cross-zone load balancing is enabled for Application Load Balancer and disabled for Network Load Balancer

By default, cross-zone load balancing is enabled for Application Load Balancer and disabled for Network Load Balancer. When cross-zone load balancing is enabled, each load balancer node distributes traffic across the registered targets in all the enabled Availability Zones. When cross-zone load balancing is disabled, each load balancer node distributes traffic only across the registered targets in its Availability Zone.

How cross-zone load balancing
works:

Cross-Zone Load Balancing

The nodes for your load balancer distribute requests from clients to registered targets. When cross-zone load balancing is enabled, each load balancer node distributes traffic across the registered targets in all enabled Availability Zones. When cross-zone load balancing is disabled, each load balancer node distributes traffic only across the registered targets in its Availability Zone.

The following diagrams demonstrate the effect of cross-zone load balancing. There are two enabled Availability Zones, with two targets in Availability Zone A and eight targets in Availability Zone B. Clients send requests, and Amazon Route 53 responds to each request with the IP address of one of the load balancer nodes. This distributes traffic such that each load balancer node receives 50% of the traffic from the clients. Each load balancer node distributes its share of the traffic across the registered targets in its scope.

If cross-zone load balancing is enabled, each of the 10 targets receives 10% of the traffic. This is because each load balancer node can route its 50% of the client traffic to all 10 targets.

If cross-zone load balancing is disabled:

- Each of the two targets in Availability Zone A receives 25% of the traffic.
- Each of the eight targets in Availability Zone B receives 6.25% of the traffic.

This is because each load balancer node can route its 50% of the client traffic only to targets in its Availability Zone.

With Application Load Balancers, cross-zone load balancing is always enabled.

With Network Load Balancers, cross-zone load balancing is disabled by default. After you create a Network Load Balancer, you can enable or disable cross-zone load balancing at any time. For more information, see Cross-Zone Load Balancing in the *User*

When you create a Classic Load Balancer, the default for cross-zone load balancing depends on how you create the load balancer. With the API or CLI, cross-zone load balancing is disabled by default. With the AWS Management Console, the option to enable cross-zone load balancing is selected by default. After you create a Classic Load Balancer, you can enable or disable cross-zone load balancing at any time. For more information, see Enable Cross-Zone Load Balancing in the *User Guide for Classic Load Balancers*.

via - https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/how-elastic-load-balancing-works.html

Incorrect Options:

By default, cross-zone load balancing is disabled for both Application Load Balancer and Network Load Balancer

By default, cross-zone load balancing is enabled for both Application Load Balancer and Network Load Balancer

By default, cross-zone load balancing is disabled for Application Load Balancer and enabled for Network Load Balancer

Per the default cross-zone load balancing settings described earlier in the explanation, these three options are incorrect.

Question 4:
A health-care company manages its web application on Amazon EC2 instances running behind Auto Scaling group (ASG). The company provides ambulances for critical patients and needs the application to be reliable. The workload of the company can be managed on 2 EC2 instances and can peak up to 6 instances when traffic increases.

As a Solutions Architect, which of the following configurations would you select as the best fit for these requirements?

- The ASG should be configured with the minimum capacity set to 4, with 2 instances each in two different Availability Zones. The maximum capacity of the ASG should be set to 6
  (Correct)
- The ASG should be configured with the minimum capacity set to 4, with 2 instances each in two different AWS Regions. The maximum capacity of the ASG should be set to 6
- The ASG should be configured with the minimum capacity set to 2 and the maximum capacity set to 6 in a single Availability Zone
- The ASG should be configured with the minimum capacity set to 2, with 1 instance each in two different Availability Zones. The maximum capacity of the ASG should be set to 6

Explanation
Correct option:

The ASG should be configured with the minimum capacity set to 4, with 2 instances each in two different Availability Zones. The maximum capacity of the ASG should be set to 6 - You configure the size of your Auto Scaling group by setting the minimum, maximum, and desired capacity. The minimum and maximum capacity are required to create an Auto Scaling group, while the desired capacity is optional. If you do not define your desired capacity upfront, it defaults to your minimum capacity.

Amazon EC2 Auto Scaling enables you to take advantage of the safety and reliability of geographic redundancy by spanning Auto Scaling groups across multiple Availability Zones within a Region. When one Availability Zone becomes unhealthy or unavailable, Auto Scaling launches new instances in an unaffected Availability Zone. When the unhealthy Availability Zone returns to a healthy state, Auto Scaling automatically redistributes the application instances evenly across all of the designated Availability Zones. Since the application is extremely critical and needs to have a reliable architecture to support it, the EC2 instances should be maintained in at least two Availability Zones (AZs) for uninterrupted service.

Amazon EC2 Auto Scaling attempts to distribute instances evenly between the Availability Zones that are enabled for your Auto Scaling group. This is why the minimum capacity should be 4 instances and not 2. ASG will launch 2 instances each in both the AZs and this redundancy is needed to keep the service available always.

Incorrect options:

The ASG should be configured with the minimum capacity set to 2, with 1 instance each in two different Availability Zones. The maximum capacity of the ASG should be set to 6

The ASG should be configured with the minimum capacity set to 2 and the maximum capacity set to 6 in a single Availability Zone

The explanation above gives the correct rationale for minimum capacity as well as the instance distribution across AZs, so both these options are incorrect.

The ASG should be configured with the minimum capacity set to 4, with 2 instances each in two different AWS Regions. The maximum capacity of the ASG should be set to 6 - An Auto Scaling group can contain EC2 instances in one or more Availability Zones within the same region. However, Auto Scaling groups cannot span multiple Regions.

Question 5:
A company runs its EC2 servers behind an Application Load Balancer along with an Auto Scaling group. The engineers at the company want to be able to install proprietary tools on each instance and perform a pre-activation status check of these tools whenever an instance is provisioned because of a scale-out event from an auto-scaling policy.

Which of the following options can be used to enable this custom action?

- Use the Auto Scaling group scheduled action to put the instance in a wait state and launch a custom script that installs the proprietary forensic tools and performs a pre-activation status check
- Use the EC2 instance user data to put the instance in a wait state and launch a custom script that installs the proprietary forensic tools and performs a pre-activation status check
- Use the Auto Scaling group lifecycle hook to put the instance in a wait state and launch a custom script that installs the proprietary forensic tools and performs a pre-activation status check **(Correct)**
- Use the EC2 instance meta data to put the instance in a wait state and launch a custom script that installs the proprietary forensic tools and performs a pre-activation status check

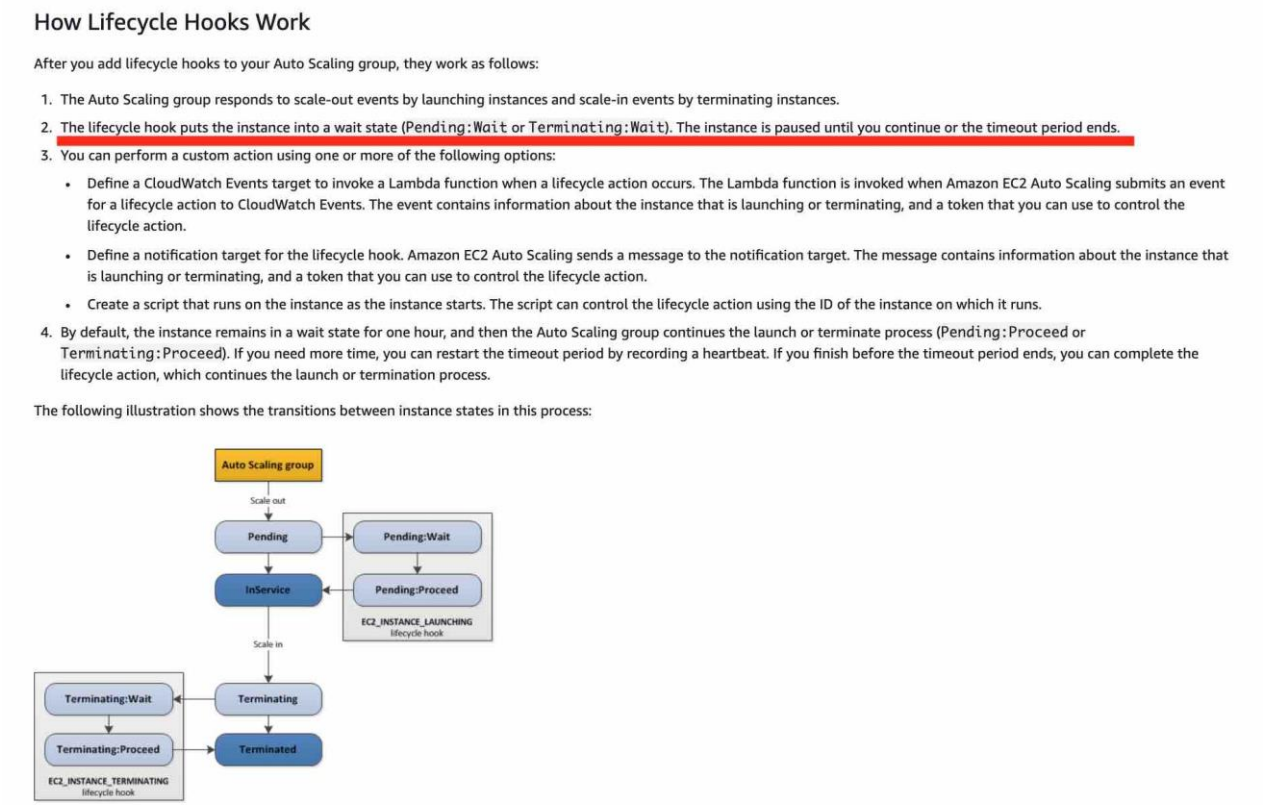
Explanation
Correct option:

Use the Auto Scaling group lifecycle hook to put the instance in a wait state and launch a custom script that installs the proprietary forensic tools and performs a pre-activation status check

An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for automatic scaling and management.

Auto Scaling group lifecycle hooks enable you to perform custom actions as the Auto Scaling group launches or terminates instances. Lifecycle hooks enable you to perform custom actions by pausing instances as an Auto Scaling group launches or terminates them. When an instance is paused, it remains in a wait state either until you complete the lifecycle action using the complete-lifecycle-action command or the CompleteLifecycleAction operation, or until the timeout period ends (one hour by default). For example, you could install or configure software on newly launched instances, or download log files from an instance before it terminates.

How lifecycle hooks work:



How Lifecycle Hooks Work

After you add lifecycle hooks to your Auto Scaling group, they work as follows:

1. The Auto Scaling group responds to scale-out events by launching instances and scale-in events by terminating instances.
2. The lifecycle hook puts the instance into a wait state (`Pending:Wait` or `Terminating:Wait`). The instance is paused until you continue or the timeout period ends.
3. You can perform a custom action using one or more of the following options:
   - Define a CloudWatch Events target to invoke a Lambda function when a lifecycle action occurs. The Lambda function is invoked when Amazon EC2 Auto Scaling submits an event for a lifecycle action to CloudWatch Events. The event contains information about the instance that is launching or terminating, and a token that you can use to control the lifecycle action.
   - Define a notification target for the lifecycle hook. Amazon EC2 Auto Scaling sends a message to the notification target. The message contains information about the instance that is launching or terminating, and a token that you can use to control the lifecycle action.
   - Create a script that runs on the instance as the instance starts. The script can control the lifecycle action using the ID of the instance on which it runs.
4. By default, the instance remains in a wait state for one hour, and then the Auto Scaling group continues the launch or terminate process (`Pending:Proceed` or `Terminating:Proceed`). If you need more time, you can restart the timeout period by recording a heartbeat. If you finish before the timeout period ends, you can complete the lifecycle action, which continues the launch or termination process.

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

via - https://docs.aws.amazon.com/autoscaling/ec2/userguide/lifecycle-hooks.html

Incorrect options:

Use the Auto Scaling group scheduled action to put the instance in a wait state and launch a custom script that installs the proprietary forensic tools and performs a pre-activation status check - To configure your Auto Scaling group to scale based on a schedule, you create a scheduled action. The scheduled action tells Amazon EC2 Auto Scaling to perform a scaling action at specified times. You cannot use scheduled action to carry out custom actions when the Auto Scaling group launches or terminates an instance.

Use the EC2 instance meta data to put the instance in a wait state and launch a custom script that installs the proprietary forensic tools and performs a pre-activation status check - EC2 instance metadata is data about your instance that you can use to configure

or manage the running instance. You cannot use EC2 instance metadata to put the instance in wait state.

Use the EC2 instance user data to put the instance in a wait state and launch a custom script that installs the proprietary forensic tools and performs a pre-activation status check - EC2 instance user data is the data that you specified in the form of a configuration script while launching your instance. You cannot use EC2 instance user data to put the instance in wait state.

Question 6:
A junior developer is learning to build websites using HTML, CSS, and JavaScript. He has created a static website and then deployed it on Amazon S3. Now he can't seem to figure out the endpoint for his super cool website.

As a solutions architect, can you help him figure out the allowed formats for the Amazon S3 website endpoints? (Select two)

- http://bucket-name.s3-website.Region.amazonaws.com **(Correct)**
- http://bucket-name.Region.s3-website.amazonaws.com
- http://s3-website.Region.bucket-name.amazonaws.com
- http://bucket-name.s3-website-Region.amazonaws.com **(Correct)**
- http://s3-website-Region.bucket-name.amazonaws.com

Explanation
Correct options:

http://bucket-name.s3-website.Region.amazonaws.com

http://bucket-name.s3-website-Region.amazonaws.com

To host a static website on Amazon S3, you configure an Amazon S3 bucket for website hosting and then upload your website content to the bucket. When you configure a bucket as a static website, you enable static website hosting, set permissions, and add an index document. Depending on your website requirements, you can also configure other options, including redirects, web traffic logging, and custom error documents.

When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket.

Depending on your Region, your Amazon S3 website endpoints follow one of these two formats.

s3-website dash (-) Region - http://bucket-name.s3-website.Region.amazonaws.com

s3-website dot (.) Region - http://bucket-name.s3-website-Region.amazonaws.com

These URLs return the default index document that you configure for the website.

Incorrect options:

http://s3-website-Region.bucket-name.amazonaws.com

http://s3-website.Region.bucket-name.amazonaws.com

http://bucket-name.Region.s3-website.amazonaws.com

These three options do not meet the specifications for the Amazon S3 website endpoints format, so these are incorrect.

Question 7:
As a Solutions Architect, you have been hired to work with the engineering team at a company to create a REST API using the serverless architecture.

Which of the following solutions will you recommend to move the company to the serverless architecture?

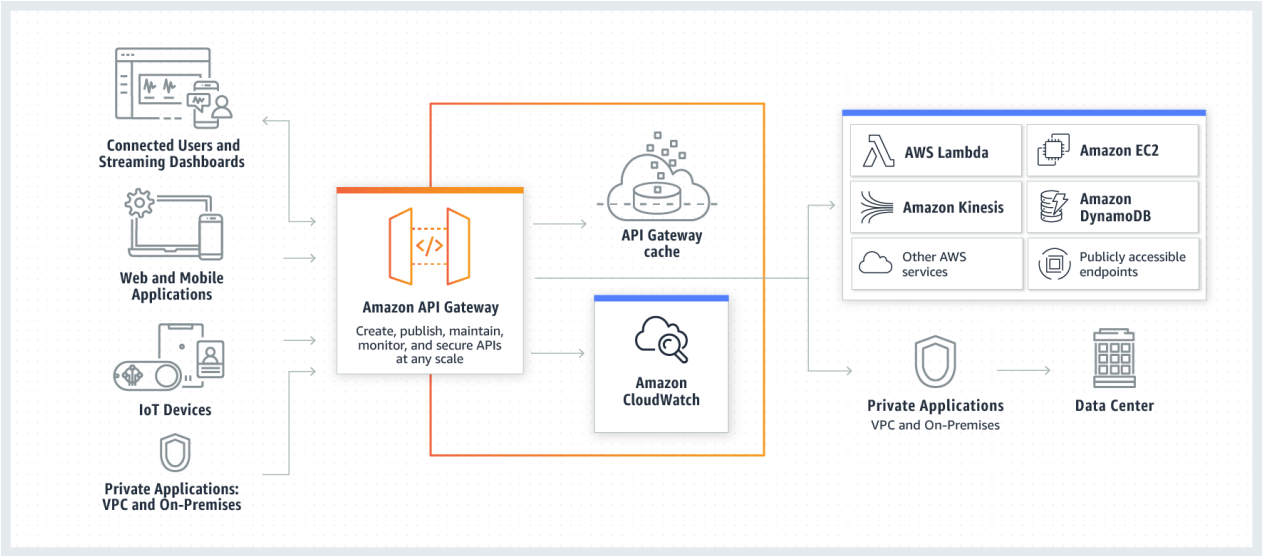
- Route 53 with EC2 as backend
- Public-facing Application Load Balancer with ECS on Amazon EC2
- API Gateway exposing Lambda Functionality **(Correct)**
- Fargate with Lambda at the front

Explanation
Correct option:

API Gateway exposing Lambda Functionality

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.

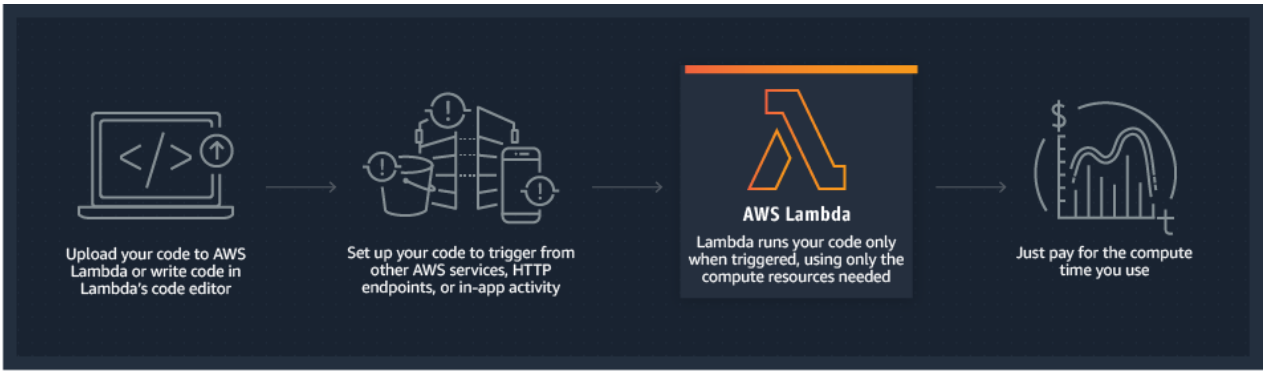
How API Gateway
Works:



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

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume.

How Lambda function works:



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

API Gateway can expose Lambda functionality through RESTful APIs. Both are serverless options offered by AWS and hence the right choice for this scenario, considering all the functionality they offer.

Incorrect options:

Fargate with Lambda at the front - Lambda cannot directly handle RESTful API requests. You can invoke a Lambda function over HTTPS by defining a custom RESTful API using Amazon API Gateway. So, Fargate with Lambda as the front-facing service is a wrong combination, though both Fargate and Lambda are serverless.

Public-facing Application Load Balancer with ECS on Amazon EC2 - ECS on Amazon EC2 does not come under serverless and hence cannot be considered for this use case.

Route 53 with EC2 as backend - Amazon EC2 is not a serverless service and hence cannot be considered for this use case.

Question 8:
A company's cloud architect has set up a solution that uses Route 53 to configure the DNS records for the primary website with the domain pointing to the Application Load Balancer (ALB). The company wants a solution where users will be directed to a static error page, configured as a backup, in case of unavailability of the primary website.

Which configuration will meet the company's requirements, while keeping the changes to a bare minimum?

- Use Route 53 Weighted routing to give minimum weight to Amazon S3 bucket that holds the error page to be displayed. In case of primary failure, the requests get routed to the error page
- Set up a Route 53 active-passive type of failover routing policy. If Route 53 health check determines the ALB endpoint as unhealthy, the traffic will be diverted to a static error page, hosted on Amazon S3 bucket **(Correct)**
- Use Route 53 Latency-based routing. Create a latency record to point to the Amazon S3 bucket that holds the error page to be displayed
- Set up a Route 53 active-active type of failover routing policy. If Route 53 health check determines the ALB endpoint as unhealthy, the traffic will be diverted to a static error page, hosted on Amazon S3 bucket

Explanation
Correct option:

Set up a Route 53 active-passive type of failover routing policy. If Route 53 health check determines the ALB endpoint as unhealthy, the traffic will be diverted to a static error page, hosted on Amazon S3 bucket

Use an active-passive failover configuration when you want a primary resource or group of resources to be available the majority of the time and you want a secondary resource or group of resources to be on standby in case all the primary resources become unavailable. When responding to queries, Route 53 includes only healthy primary resources. If all the primary resources are unhealthy, Route 53 begins to include only the healthy secondary resources in response to DNS queries.

Incorrect options:

Set up a Route 53 active-active type of failover routing policy. If Route 53 health check determines the ALB endpoint as unhealthy, the traffic will be diverted to a static error page, hosted on Amazon S3 bucket - This option has been added as a distractor as there is no such thing as an active-active failover routing policy in Route 53. You can configure active-active failover using any routing policy (or combination of routing policies) other than failover routing policy and you configure active-passive failover only using the failover routing policy. In active-active failover configuration, all the records that have the same name, the same type (such as A or AAAA), and the same routing

policy (such as weighted or latency) are active unless Route 53 considers them unhealthy. Route 53 can respond to a DNS query using any healthy record.

Use Route 53 Latency-based routing. Create a latency record to point to the Amazon S3 bucket that holds the error page to be displayed - If your application is hosted in multiple AWS Regions, you can improve performance for your users by serving their requests from the AWS Region that provides the lowest latency - this is Latency-based routing and is not helpful for the current use case.

Use Route 53 Weighted routing to give minimum weight to Amazon S3 bucket that holds the error page to be displayed. In case of primary failure, the requests get routed to the error page - Weighted routing lets you associate multiple resources with a single domain name (example.com) or subdomain name (acme.example.com) and choose how much traffic is routed to each resource. This can be useful for a variety of purposes, including load balancing and testing new versions of the software. This is not useful for the current use case.

Question 9:
A silicon valley based startup helps its users legally sign highly confidential contracts. To meet the compliance guidelines, the startup must ensure that the signed contracts are encrypted using the AES-256 algorithm via an encryption key that is generated internally. The startup is now migrating to AWS Cloud and would like you to advise them on the encryption scheme to adopt. The startup wants to continue using their existing encryption key generation mechanism.

What do you recommend?

- SSE-KMS
- SSE-C **(Correct)**
- SSE-S3
- Client-Side Encryption

Explanation
Correct option:

SSE-C - With Server-Side Encryption with Customer-Provided Keys (SSE-C), you manage the encryption keys and Amazon S3 manages the encryption, as it writes to disks, and decryption when you access your objects. With SSE-C, the startup can still provide the encryption key but let AWS do the encryption. Therefore, this is the correct option.

Incorrect options:

SSE-KMS - AWS Key Management Service (AWS KMS) is a service that combines secure, highly available hardware and software to provide a key management system scaled for the cloud. When you use server-side encryption with AWS KMS (SSE-KMS),

you can specify a customer-managed CMK that you have already created. But, you never get to know the actual key here.

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

Client-Side Encryption - Client-side encryption is the act of encrypting data before sending it to Amazon S3. To enable client-side encryption, you have the following options: Use a customer master key (CMK) stored in AWS Key Management Service (AWS KMS), Use a master key you store within your application. Since the customer wants to use AWS provided facility, this is not an option.

Question 10:
An IT company has built a custom data warehousing solution for a retail organization by using Amazon Redshift. As part of the cost optimizations, the company wants to move any historical data (any data older than a year) into S3, as the daily analytical reports consume data for just the last one year. However the analysts want to retain the ability to cross-reference this historical data along with the daily reports.

The company wants to develop a solution with the LEAST amount of effort and MINIMUM cost. As a solutions architect, which option would you recommend to facilitate this use-case?

- Use the Redshift COPY command to load the S3 based historical data into Redshift. Once the ad-hoc queries are run for the historic data, it can be removed from Redshift
- Setup access to the historical data via Athena. The analytics team can run historical data queries on Athena and continue the daily reporting on Redshift. In case the reports need to be cross-referenced, the analytics team need to export these in flat files and then do further analysis
- Use Glue ETL job to load the S3 based historical data into Redshift. Once the ad-hoc queries are run for the historic data, it can be removed from Redshift
- Use Redshift Spectrum to create Redshift cluster tables pointing to the underlying historical data in S3. The analytics team can then query this historical data to cross-reference with the daily reports from Redshift
  (Correct)

Explanation
Correct option:

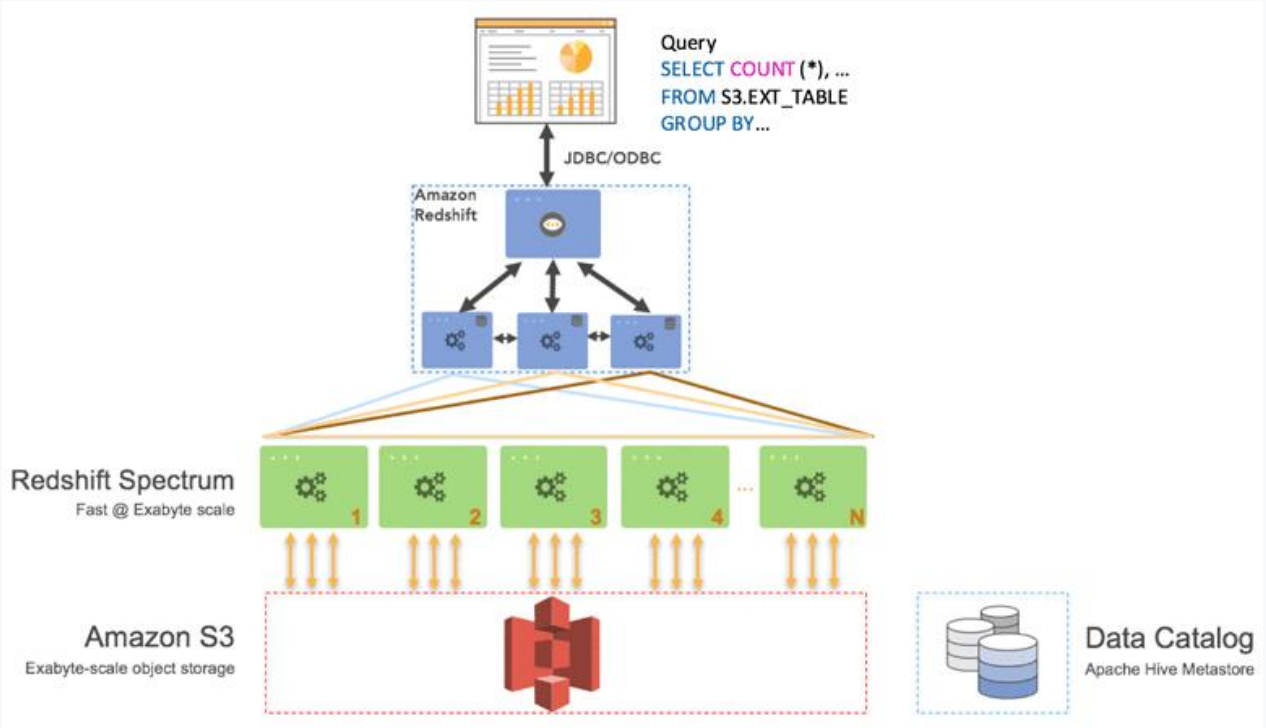
Use Redshift Spectrum to create Redshift cluster tables pointing to the underlying historical data in S3. The analytics team can then query this historical data to cross-reference with the daily reports from Redshift

Amazon Redshift is a fully-managed petabyte-scale cloud-based data warehouse product designed for large scale data set storage and analysis.

Using Amazon Redshift Spectrum, you can efficiently query and retrieve structured and semistructured data from files in Amazon S3 without having to load the data into Amazon Redshift tables.

Amazon Redshift Spectrum resides on dedicated Amazon Redshift servers that are independent of your cluster. Redshift Spectrum pushes many compute-intensive tasks, such as predicate filtering and aggregation, down to the Redshift Spectrum layer. Thus, Redshift Spectrum queries use much less of your cluster's processing capacity than other queries.

Redshift Spectrum
Overview



via - https://aws.amazon.com/blogs/big-data/amazon-redshift-spectrum-extends-data-warehousing-out-to-exabytes-no-loading-required/

Incorrect options:

Setup access to the historical data via Athena. The analytics team can run historical data queries on Athena and continue the daily reporting on Redshift. In case the reports need to be cross-referenced, the analytics team need to export these in flat files and then do further analysis Amazon Athena is an interactive query service that makes it easy to analyze data directly in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to set up or manage, and customers pay only for the queries they run. You can use Athena to process logs, perform ad-hoc analysis, and run interactive queries. Providing access to historical data via Athena would mean that historical data reconciliation would become difficult as the daily report would still be produced via Redshift. Such a setup is cumbersome to maintain on a day to day basis. Hence the option to use Athena is ruled out.

Use the Redshift COPY command to load the S3 based historical data into Redshift. Once the ad-hoc queries are run for the historic data, it can be removed from Redshift

Use Glue ETL job to load the S3 based historical data into Redshift. Once the ad-hoc queries are run for the historic data, it can be removed from Redshift

Loading historical data into Redshift via COPY command or Glue ETL job would cost heavy for a one-time ad-hoc process. The same result can be achieved more cost-efficiently by using Redshift Spectrum. Therefore both these options to load historical data into Redshift are also incorrect for the given use-case.

Question 11:
You are a cloud architect at an IT company. The company has multiple enterprise customers that manage their own mobile apps that capture and send data to Amazon Kinesis Data Streams. They have been getting a `ProvisionedThroughputExceededException` exception. You have been contacted to help and upon analysis, you notice that messages are being sent one by one at a high rate.

Which of the following options will help with the exception while keeping costs at a minimum?

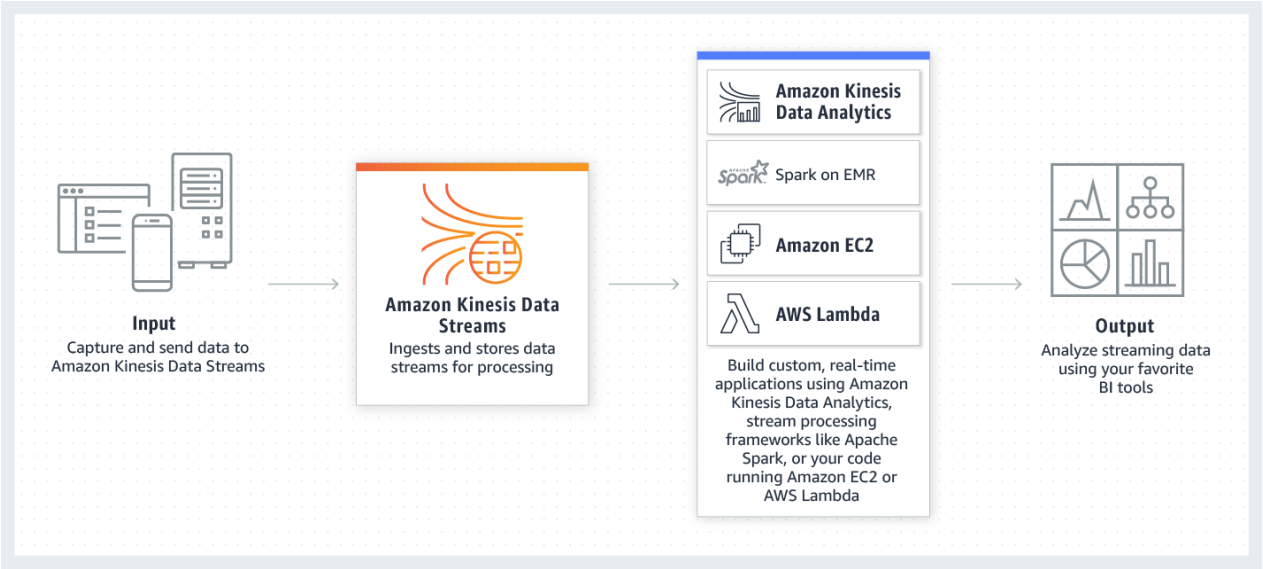
- Increase the number of shards
- Use Exponential Backoff
- Use batch messages **(Correct)**
- Decrease the Stream retention duration

Explanation
Correct options:

Use batch messages

Amazon Kinesis Data Streams (KDS) is a massively scalable and durable real-time data streaming service. KDS can continuously capture gigabytes of data per second from hundreds of thousands of sources such as website clickstreams, database event streams, financial transactions, social media feeds, IT logs, and location-tracking events. The data collected is available in milliseconds to enable real-time analytics use cases such as real-time dashboards, real-time anomaly detection, dynamic pricing, and more.

Kinesis Data Streams
Overview:



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

When a host needs to send many records per second (RPS) to Amazon Kinesis, simply calling the basic PutRecord API action in a loop is inadequate. To reduce overhead and increase throughput, the application must batch records and implement parallel HTTP requests. This will increase the efficiency overall and ensure you are optimally using the shards.

Incorrect options:

Use Exponential Backoff: While this may help in the short term, as soon as the request rate increases, you will see the `ProvisionedThroughputExceededException` exception again.

Increase the number of shards - Increasing shards could be a short term fix but will substantially increase the cost, so this option is ruled out.

Decrease the Stream retention duration - This operation may result in data loss and won't help with the exceptions, so this option is incorrect.

Question 12:
A social media analytics company uses a fleet of EC2 servers to manage its analytics workflow. These EC2 servers operate under an Auto Scaling group. The engineers at the company want to be able to download log files whenever an instance terminates because of a scale-in event from an auto-scaling policy.

Which of the following features can be used to enable this custom action?

- Auto Scaling group lifecycle hook **(Correct)**
- EC2 instance user data
- EC2 instance meta data

- Auto Scaling group scheduled action

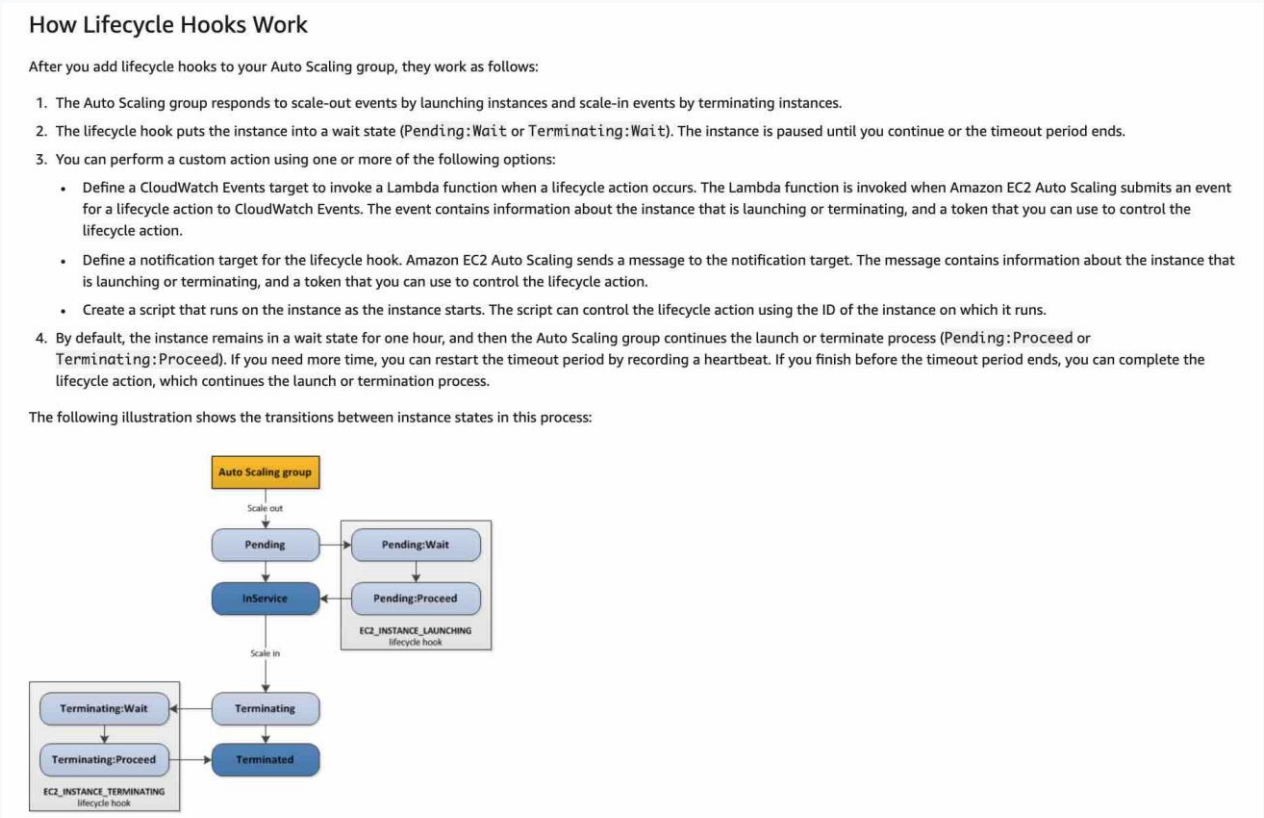
Explanation
Correct option:

Auto Scaling group lifecycle hook

An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for automatic scaling and management.

Auto Scaling group lifecycle hooks enable you to perform custom actions as the Auto Scaling group launches or terminates instances. Lifecycle hooks enable you to perform custom actions by pausing instances as an Auto Scaling group launches or terminates them. For example, you could install or configure software on newly launched instances, or download log files from an instance before it terminates.

How lifecycle hooks work:



via - https://docs.aws.amazon.com/autoscaling/ec2/userguide/lifecycle-hooks.html

Incorrect options:

EC2 instance meta data - EC2 instance metadata is data about your instance that you can use to configure or manage the running instance. You cannot use EC2 instance metadata to download log files whenever an instance terminates because of a scale-in event from an auto-scaling policy.

EC2 instance user data - EC2 instance user data is the data that you specified in the form of a configuration script while launching your instance. You cannot use EC2

instance user data to download log files whenever an instance terminates because of a scale-in event from an auto-scaling policy.

Auto Scaling group scheduled action - To configure your Auto Scaling group to scale based on a schedule, you create a scheduled action. The scheduled action tells Amazon EC2 Auto Scaling to perform a scaling action at specified times. You cannot use scheduled action to download log files whenever an instance terminates because of a scale-in event from an auto-scaling policy.

Question 13:
A leading media company wants to do an accelerated online migration of hundreds of terabytes of files from their on-premises data center to Amazon S3 and then establish a mechanism to access the migrated data for ongoing updates from the on-premises applications.

As a solutions architect, which of the following would you select as the MOST performant solution for the given use-case?

- Use AWS DataSync to migrate existing data to Amazon S3 as well as access the S3 data for ongoing updates
- Use S3 Transfer Acceleration to migrate existing data to Amazon S3 and then use DataSync for ongoing updates from the on-premises applications
- Use File Gateway configuration of AWS Storage Gateway to migrate data to Amazon S3 and then use S3 Transfer Acceleration for ongoing updates from the on-premises applications
- Use AWS DataSync to migrate existing data to Amazon S3 and then use File Gateway to retain access to the migrated data for ongoing updates from the on-premises applications **(Correct)**

Explanation
Correct options:

Use AWS DataSync to migrate existing data to Amazon S3 and then use File Gateway to retain access to the migrated data for ongoing updates from the on-premises applications

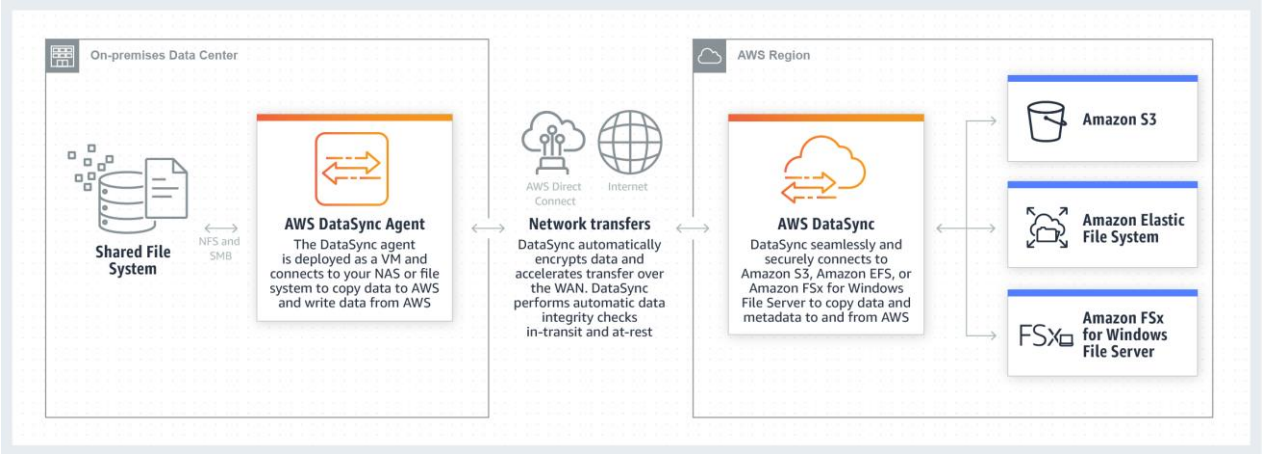
AWS DataSync is an online data transfer service that simplifies, automates, and accelerates copying large amounts of data to and from AWS storage services over the internet or AWS Direct Connect.

AWS DataSync fully automates and accelerates moving large active datasets to AWS, up to 10 times faster than command-line tools. It is natively integrated with Amazon S3, Amazon EFS, Amazon FSx for Windows File Server, Amazon CloudWatch, and AWS CloudTrail, which provides seamless and secure access to your storage services, as well as detailed monitoring of the transfer. DataSync uses a purpose-built network protocol

and scale-out architecture to transfer data. A single DataSync agent is capable of saturating a 10 Gbps network link.

DataSync fully automates the data transfer. It comes with retry and network resiliency mechanisms, network optimizations, built-in task scheduling, monitoring via the DataSync API and Console, and CloudWatch metrics, events, and logs that provide granular visibility into the transfer process. DataSync performs data integrity verification both during the transfer and at the end of the transfer.

How DataSync
Works:



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

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. File gateway offers SMB or NFS-based access to data in Amazon S3 with local caching.

The combination of DataSync and File Gateway is the correct solution. AWS DataSync enables you to automate and accelerate online data transfers to AWS storage services. File Gateway then provides your on-premises applications with low latency access to the migrated data.

Incorrect options:

Use AWS DataSync to migrate existing data to Amazon S3 as well as access the S3 data for ongoing updates - AWS DataSync is used to easily transfer data to and from AWS with up to 10x faster speeds. It is used to transfer data and cannot be used to facilitate ongoing updates to the migrated files from the on-premises applications.

Use File Gateway configuration of AWS Storage Gateway to migrate data to Amazon S3 and then use S3 Transfer Acceleration for ongoing updates from the on-premises applications - File Gateway can be used to move on-premises data to AWS Cloud, but it not an optimal solution for high volumes. Migration services such as DataSync are best suited for this purpose. S3 Transfer Acceleration cannot facilitate ongoing updates to the migrated files from the on-premises applications.

Use S3 Transfer Acceleration to migrate existing data to Amazon S3 and then use DataSync for ongoing updates from the on-premises applications - If your application is already integrated with the Amazon S3 API, and you want higher throughput for transferring large files to S3, S3 Transfer Acceleration can be used. However DataSync cannot be used to facilitate ongoing updates to the migrated files from the on-premises applications.

Question 14:
A CRM application is facing user experience issues with users reporting frequent sign-in requests from the application. The application is currently hosted on multiple EC2 instances behind an Application Load Balancer. The engineering team has identified the root cause as unhealthy servers causing session data to be lost. The team would like to implement a distributed in-memory cache-based session management solution.

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

- Use Elasticache for distributed in-memory cache based session management **(Correct)**
- Use DynamoDB for distributed in-memory cache based session management
- Use Application Load Balancer sticky sessions
- Use RDS for distributed in-memory cache based session management

Explanation
Correct option:

Use Elasticache for distributed cache-based session management

Amazon ElastiCache can be used as a distributed in-memory cache for session management. Amazon ElastiCache allows you to seamlessly set up, run, and scale popular open-Source compatible in-memory data stores in the cloud. Session stores can be set up using both Memcached or Redis for ElastiCache.

Amazon ElastiCache for Redis is a great choice for real-time transactional and analytical processing use cases such as caching, chat/messaging, gaming leaderboards, geospatial, machine learning, media streaming, queues, real-time analytics, and session store.

Amazon ElastiCache for Memcached is a Memcached-compatible in-memory key-value store service that can be used as a cache or a data store. Session stores are easy to create with Amazon ElastiCache for Memcached.

How ElastiCache
Works:

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

Incorrect options:

Use RDS for distributed in-memory cache-based session management - Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It cannot be used as a distributed in-memory cache for session management, hence this option is incorrect.

Use DynamoDB for distributed in-memory cache-based session management - Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. DynamoDB is a NoSQL database and is not the right fit for a distributed in-memory cache-based session management solution.

Use Application Load Balancer sticky sessions - Although sticky sessions enable each user to interact with one server and one server only, however, in case of an unhealthy server, all the session data is gone as well. Therefore Elasticache powered distributed in-memory cache-based session management is a better solution.

Question 15:
A media company is evaluating the possibility of moving its IT infrastructure to the AWS Cloud. The company needs at least 10 TB of storage with the maximum possible I/O performance for processing certain files which are mostly large videos. The company also needs close to 450 TB of very durable storage for storing media content and almost double of it, i.e. 900 TB for archival of legacy data.

As a Solutions Architect, which set of services will you recommend to meet these requirements?

- Amazon EC2 instance store for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage **(Correct)**
- Amazon EC2 instance store for maximum performance, AWS Storage Gateway for on-premises durable data access and Amazon S3 Glacier Deep Archive for archival storage
-

Amazon S3 standard storage for maximum performance, Amazon S3 Intelligent-Tiering for intelligent, durable storage, and Amazon S3 Glacier Deep Archive for archival storage

- Amazon EBS for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage

Explanation
Correct option:

Amazon EC2 instance store for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage - 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.

You can specify instance store volumes for an instance only when you launch it. You can't detach an instance store volume from one instance and attach it to a different instance.

Some instance types use NVMe or SATA-based solid-state drives (SSD) to deliver high random I/O performance. This is a good option when you need storage with very low latency, but you don't need the data to persist when the instance terminates or you can take advantage of fault-tolerant architectures.

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.

S3 Glacier is a secure, durable, and low-cost storage class for data archiving. You can reliably store any amount of data at costs that are competitive with or cheaper than on-premises solutions. To keep costs low yet suitable for varying needs, S3 Glacier provides three retrieval options that range from a few minutes to hours. You can upload objects directly to S3 Glacier, or use S3 Lifecycle policies to transfer data between any of the S3 Storage Classes for active data (S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, and S3 One Zone-IA) and S3 Glacier.

Incorrect options:

Amazon S3 standard storage for maximum performance, Amazon S3 Intelligent-Tiering for intelligent, durable storage, and Amazon S3 Glacier Deep Archive for archival storage - Amazon EC2 instance store volumes provide the best I/O performance for low latency requirement, as in the current use case. 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 Glacier Deep Archive is Amazon S3's lowest-cost storage class and supports long-term retention and digital preservation for data that may be accessed once or twice a

year. It is designed for customers — particularly those in highly-regulated industries, such as the Financial Services, Healthcare, and Public Sectors — that retain data sets for 7-10 years or longer to meet regulatory compliance requirements.

Amazon EBS for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage - Amazon Elastic Block Store (Amazon EBS) provides block-level storage volumes for use with EC2 instances. EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. For high I/O performance, instance store volumes are a better option.

Amazon EC2 instance store for maximum performance, AWS Storage Gateway for on-premises durable data access and Amazon S3 Glacier Deep Archive for archival storage - AWS Storage Gateway is a hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage. Storage Gateway will be the right answer if the customer wanted to retain the on-premises data storage and just move the applications to AWS Cloud. In the absence of such requirements, instance store is a better option for high performance and Amazon S3 for durable storage.

Question 16:
A development team has deployed a microservice to the ECS. The application layer is in a Docker container that provides both static and dynamic content through an Application Load Balancer. With increasing load, the ECS cluster is experiencing higher network usage. The development team has looked into the network usage and found that 90% of it is due to distributing static content of the application.

As a Solutions Architect, what do you recommend to improve the application's network usage and decrease costs?

- Distribute the static content through Amazon S3 **(Correct)**
- Distribute the dynamic content through Amazon S3
- Distribute the static content through Amazon EFS
- Distribute the dynamic content through Amazon EFS

Explanation
Correct option:

Distribute the static content through Amazon S3 -

You can use Amazon S3 to host a static website. On a static website, individual web pages include static content. They might also contain client-side scripts. To host a static website on Amazon S3, you configure an Amazon S3 bucket for website hosting and then upload your website content to the bucket. When you configure a bucket as a static website, you must enable website hosting, set permissions, and create and add an index

document. Depending on your website requirements, you can also configure redirects, web traffic logging, and a custom error document.

Distributing the static content through S3 allows us to offload most of the network usage to S3 and free up our applications running on ECS.

Incorrect options:

Distribute the dynamic content through Amazon S3 - By contrast, a dynamic website relies on server-side processing, including server-side scripts such as PHP, JSP, or ASP.NET. Amazon S3 does not support server-side scripting, but AWS has other resources for hosting dynamic websites.

Distribute the static content through Amazon EFS

Distribute the dynamic content through Amazon EFS

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. Using EFS for static or dynamic content will not change anything as static content on EFS would still have to be distributed by the ECS instances.

Question 17:
A leading video streaming provider is migrating to AWS Cloud infrastructure for delivering its content to users across the world. The company wants to make sure that the solution supports at least a million requests per second for its EC2 server farm.

As a solutions architect, which type of Elastic Load Balancer would you recommend as part of the solution stack?

- Infrastructure Load Balancer
- Network Load Balancer **(Correct)**
- Classic Load Balancer
- Application Load Balancer

Explanation
Correct option:

Network Load Balancer

Network Load Balancer is best suited for use-cases involving low latency and high throughput workloads that involve scaling to millions of requests per second. Network Load Balancer operates at the connection level (Layer 4), routing connections to targets - Amazon EC2 instances, microservices, and containers – within Amazon Virtual Private Cloud (Amazon VPC) based on IP protocol data.

Incorrect options:

Application Load Balancer - Application Load Balancer operates at the request level (layer 7), routing traffic to targets – EC2 instances, containers, IP addresses, and Lambda functions based on the content of the request. Ideal for advanced load balancing of HTTP and HTTPS traffic, Application Load Balancer provides advanced request routing targeted at delivery of modern application architectures, including microservices and container-based applications. Application Load Balancer is not a good fit for the low latency and high throughput scenario mentioned in the given use-case.

Classic Load Balancer - Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network. Classic Load Balancer is not a good fit for the low latency and high throughput scenario mentioned in the given use-case.

Infrastructure Load Balancer - There is no such thing as Infrastructure Load Balancer and this option just acts as a distractor.

Question 18:
A financial services firm uses a high-frequency trading system and wants to write the log files into Amazon S3. The system will also read these log files in parallel on a near real-time basis. The engineering team wants to address any data discrepancies that might arise when the trading system overwrites an existing log file and then tries to read that specific log file.

Which of the following options BEST describes the capabilities of Amazon S3 relevant to this scenario?

- A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, Amazon S3 does not return any data
- A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, Amazon S3 might return the new data
- A process replaces an existing object and immediately tries to read it. Amazon S3 always returns the latest version of the object **(Correct)**
- A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, Amazon S3 might return the previous data

Explanation
Correct option:

A process replaces an existing object and immediately tries to read it. Amazon S3 always returns the latest version of the object

Amazon S3 delivers strong read-after-write consistency automatically, without changes to performance or availability, without sacrificing regional isolation for applications, and at no additional cost.

After a successful write of a new object or an overwrite of an existing object, any subsequent read request immediately receives the latest version of the object. S3 also provides strong consistency for list operations, so after a write, you can immediately perform a listing of the objects in a bucket with any changes reflected.

Strong read-after-write consistency helps when you need to immediately read an object after a write. For example, strong read-after-write consistency when you often read and list immediately after writing objects.

To summarize, all S3 GET, PUT, and LIST operations, as well as operations that change object tags, ACLs, or metadata, are strongly consistent. What you write is what you will read, and the results of a LIST will be an accurate reflection of what's in the bucket.

Incorrect options:

A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, Amazon S3 might return the previous data

A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, Amazon S3 does not return any data

A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, Amazon S3 might return the new data

These three options contradict the earlier details provided in the explanation.

Question 19:
A company wants to store business-critical data on EBS volumes which provide persistent storage independent of EC2 instances. During a test run, the development team found that on terminating an EC2 instance, the attached EBS volume was also lost, which was contrary to their assumptions.

As a solutions architect, could you explain this issue?

- The EBS volumes were not backed up on Amazon S3 storage, resulting in the loss of volume
- The EBS volumes were not backed up on EFS file system storage, resulting in the loss of volume
- On termination of an EC2 instance, all the attached EBS volumes are always terminated
-

The EBS volume was configured as the root volume of Amazon EC2 instance. On termination of the instance, the default behavior is to also terminate the attached root volume

(Correct)

Explanation
Correct option:

The EBS volume was configured as the root volume of the Amazon EC2 instance. On termination of the instance, the default behavior is to also terminate the attached root volume

Amazon Elastic Block Store (EBS) is an easy to use, high-performance block storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction-intensive workloads at any scale.

When you launch an instance, the root device volume contains the image used to boot the instance. You can choose between AMIs backed by Amazon EC2 instance store and AMIs backed by Amazon EBS.

By default, the root volume for an AMI backed by Amazon EBS is deleted when the instance terminates. You can change the default behavior to ensure that the volume persists after the instance terminates. Non-root EBS volumes remain available even after you terminate an instance to which the volumes were attached. Therefore, this option is correct.

Incorrect options:

The EBS volumes were not backed up on Amazon S3 storage, resulting in the loss of volume

The EBS volumes were not backed up on EFS file system storage, resulting in the loss of volume

EBS volumes do not need to back up the data on Amazon S3 or EFS filesystem. Both these options are added as distractors.

On termination of an EC2 instance, all the attached EBS volumes are always terminated - As mentioned earlier, non-root EBS volumes remain available even after you terminate an instance to which the volumes were attached. Hence this option is incorrect.

Question 20:
A company is looking for a technology that allows its mobile app users to connect through a Google login and have the capability to turn on MFA (Multi-Factor Authentication) to have maximum security. Ideally, the solution should be fully managed by AWS.

Which technology do you recommend for managing the users' accounts?

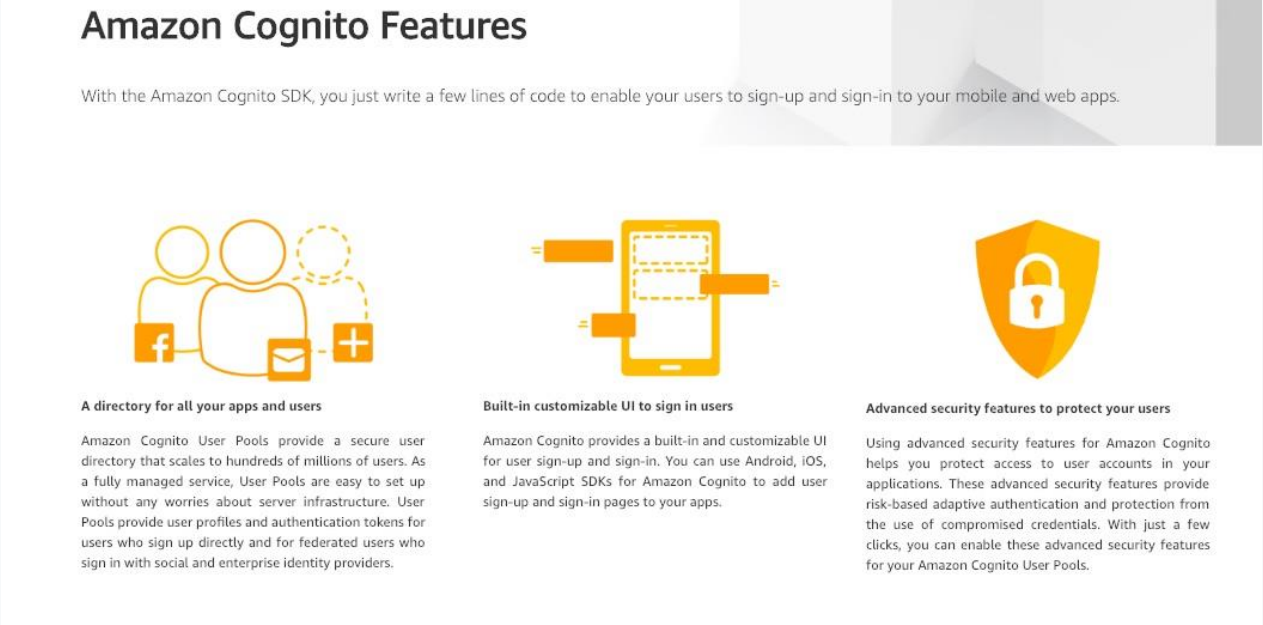
- Enable the AWS Google Login Service

- Amazon Cognito
  (Correct)
- Write a Lambda function with Auth0 3rd party integration
- AWS Identity and Access Management (IAM)

Explanation
Correct option:

Amazon Cognito - Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0. Here Cognito is the best technology choice for managing mobile user accounts.

Amazon Cognito
Features:



via - https://aws.amazon.com/cognito/details/

Incorrect options:

Write a Lambda function with Auth0 3rd party integration - AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume. Using Lambda would require code maintenance for user management functionality, therefore this option is ruled out.

AWS Identity and Access Management (IAM) - AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources. IAM cannot be used to manage mobile user accounts.

Enable the AWS Google Login Service - There is no such thing as AWS Google Login service. This option is just added as a distractor.

Question 21:
A company wants to ensure high availability for its RDS database. The development team wants to opt for Multi-AZ deployment and they would like to understand what happens when the primary instance of the Multi-AZ configuration goes down.

As a Solutions Architect, which of the following will you identify as the outcome of the scenario?

- The URL to access the database will change to the standby DB
- The CNAME record will be updated to point to the standby DB **(Correct)**
- The application will be down until the primary database has recovered itself
- An email will be sent to the System Administrator asking for manual intervention

Explanation
Correct option:

The CNAME record will be updated to point to the standby DB - Amazon RDS provides high availability and failover support for DB instances using Multi-AZ deployments. Amazon RDS uses several different technologies to provide failover support. Multi-AZ deployments for MariaDB, MySQL, Oracle, and PostgreSQL DB instances use Amazon's failover technology. SQL Server DB instances use SQL Server Database Mirroring (DBM) or Always On Availability Groups (AGs).

In a Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone. The primary DB instance is synchronously replicated across Availability Zones to a standby replica to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups. Running a DB instance with high availability can enhance availability during planned system maintenance, and help protect your databases against DB instance failure and Availability Zone disruption.

Failover is automatically handled by Amazon RDS so that you can resume database operations as quickly as possible without administrative intervention. When failing over, Amazon RDS simply flips the canonical name record (CNAME) for your DB instance to point at the standby, which is in turn promoted to become the new primary. Multi-AZ means the URL is the same, the failover is automated, and the CNAME will automatically be updated to point to the standby database.

Incorrect options:

The URL to access the database will change to the standby DB - As discussed above, URL remains the same.

An email will be sent to the System Administrator asking for manual intervention - This option is incorrect and it has been added as a distractor.

The application will be down until the primary database has recovered itself - This option is incorrect and it has been added as a distractor.

Question 22:
The infrastructure team at a company maintains 5 different VPCs (let's call these VPCs A, B, C, D, E) for resource isolation. Due to the changed organizational structure, the team wants to interconnect all VPCs together. To facilitate this, the team has set up VPC peering connections between VPC A and all other VPCs in a hub and spoke model with VPC A at the center. However, the team has still failed to establish connectivity between all VPCs.

As a solutions architect, which of the following would you recommend as the MOST resource-efficient and scalable solution?

- Use an internet gateway to interconnect the VPCs
- Use a VPC endpoint to interconnect the VPCs
- Use a transit gateway to interconnect the VPCs **(Correct)**
- Establish VPC peering connections between all VPCs

Explanation
Correct option:

Use a transit gateway to interconnect the VPCs

A transit gateway is a network transit hub that you can use to interconnect your virtual private clouds (VPC) and on-premises networks.

Transit Gateway
Overview:

What is a transit gateway?

PDF

A *transit gateway* is a network transit hub that you can use to interconnect your virtual private clouds (VPC) and on-premises networks.

For more information, see AWS Transit Gateway.

Transit gateway concepts

The following are the key concepts for transit gateways:

- **attachment** — You can attach a VPC, an AWS Direct Connect gateway, a peering connection with another transit gateway, or a VPN connection to a transit gateway.
- **transit gateway Maximum Transmission Unit (MTU)** — The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The larger the MTU of a connection, the more data can be passed in a single packet. A transit gateway supports an MTU of 8500 bytes for traffic between VPCs, Direct Connect and peering attachments. Traffic over VPN connections can have an MTU of 1500 bytes.
- **transit gateway route table** — A transit gateway has a default route table and can optionally have additional route tables. A route table includes dynamic and static routes that decide the next hop based on the destination IP address of the packet. The target of these routes could be a VPC or a VPN connection. By default, transit gateway attachments are associated with the default transit gateway route table.
- **associations** — Each attachment is associated with exactly one route table. Each route table can be associated with zero to many attachments.
- **route propagation** — A VPC or VPN connection can dynamically propagate routes to a transit gateway route table. With a VPC, you must create static routes to send traffic to the transit gateway. With a VPN connection, routes are propagated from the transit gateway to your on-premises router using Border Gateway Protocol (BGP). With a peering attachment, you must create a static route in the transit gateway route table to point to the peering attachment.

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

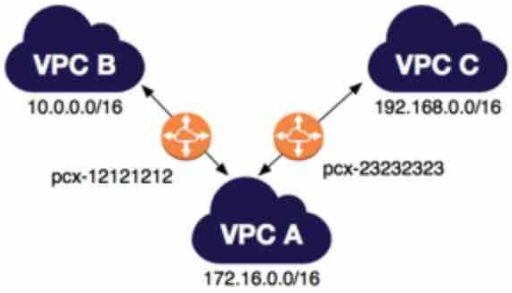
A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Transitive Peering does not work for VPC peering connections. So, if you have a VPC peering connection between VPC A and VPC B (pcx-aaaabbbb), and between VPC A and VPC C (pcx-aaaacccc). Then, there is no VPC peering connection between VPC B and VPC C. Instead of using VPC peering, you can use an AWS Transit Gateway that acts as a network transit hub, to interconnect your VPCs or connect your VPCs with on-premises networks. Therefore this is the correct option.

VPC Peering Connections
Overview:

Multiple VPC peering connections

A VPC peering connection is a one to one relationship between two VPCs. You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported. You do not have any peering relationship with VPCs that your VPC is not directly peered with.

The following diagram is an example of one VPC peered to two different VPCs. There are two VPC peering connections: VPC A is peered with both VPC B and VPC C. VPC B and VPC C are not peered, and you cannot use VPC A as a transit point for peering between VPC B and VPC C. If you want to enable routing of traffic between VPC B and VPC C, you must create a unique VPC peering connection between them.



via - https://docs.aws.amazon.com/vpc/latest/peering/vpc-peering-basics.html

Incorrect options:

Use an internet gateway to interconnect the VPCs - An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It, therefore, imposes no availability risks or bandwidth constraints on your network traffic. You cannot use an internet gateway to interconnect your VPCs and on-premises networks, hence this option is incorrect.

Use a VPC endpoint to interconnect the VPCs - A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by AWS PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. You cannot use a VPC endpoint to interconnect your VPCs and on-premises networks, hence this option is incorrect.

Establish VPC peering connections between all VPCs - Establishing VPC peering between all VPCs is an inelegant and clumsy way to establish connectivity between all VPCs. Instead, you should use a Transit Gateway that acts as a network transit hub to interconnect your VPCs and on-premises networks.

Question 23:
An Internet-of-Things (IoT) company is planning on distributing a master sensor in people's homes to measure the key metrics from its smart devices. In order to provide adjustment commands for these devices, the company would like to have a streaming system that supports ordered data based on the sensor's key, and also sustains high throughput messages (thousands of messages per second).

As a solutions architect, which of the following AWS services would you recommend for this use-case?

- Amazon Simple Notification Service (SNS)
- Amazon Simple Queue Service (SQS)
- AWS Lambda
- Amazon Kinesis Data Streams (KDS) **(Correct)**

Explanation
Correct option:

Amazon Kinesis Data Streams (KDS) - Amazon Kinesis Data Streams (KDS) is a massively scalable and durable real-time data streaming service. KDS can continuously capture gigabytes of data per second from hundreds of thousands of sources such as website clickstreams, database event streams, financial transactions, social media feeds, IT logs, and location-tracking events. The throughput of an Amazon Kinesis data stream is designed to scale without limits via increasing the number of shards within a data stream.

However, there are certain limits you should keep in mind while using Amazon Kinesis Data Streams:

By default, records of a stream are accessible for up to 24 hours from the time they are added to the stream. You can raise this limit to up to 7 days by enabling extended data retention.

The maximum size of a data blob (the data payload before Base64-encoding) within one record is 1 megabyte (MB). Each shard can support up to 1000 PUT records per second.

Kinesis is the right answer here, as by providing a partition key in your message, you can guarantee ordered messages for a specific sensor, even if your stream is sharded.

Incorrect options:

Amazon Simple Queue Service (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. SQS eliminates the complexity and overhead associated with managing and operating message-oriented

middleware, and empowers developers to focus on differentiating work. Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available. Kinesis is better for streaming data since queues aren't meant for real-time streaming of data.

Amazon Simple Notification Service (SNS) - Amazon Simple Notification Service (SNS) is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications. Amazon SNS provides topics for high-throughput, push-based, many-to-many messaging. SNS cannot be used for data streaming. Therefore this option is not the best fit for the given use-case.

AWS Lambda - AWS Lambda lets you run code without provisioning or managing servers. It cannot be used for production-grade serverless log analytics. Lambda isn't meant to retain data either. Therefore this option is not the best fit for the given use-case.

Question 24:
A company hires experienced specialists to analyze the customer service calls attended by its call center representatives. Now, the company wants to move to AWS Cloud and is looking at an automated solution to analyze customer service calls for sentiment analysis and security.

As a Solutions Architect, which of the following solutions would you recommend?

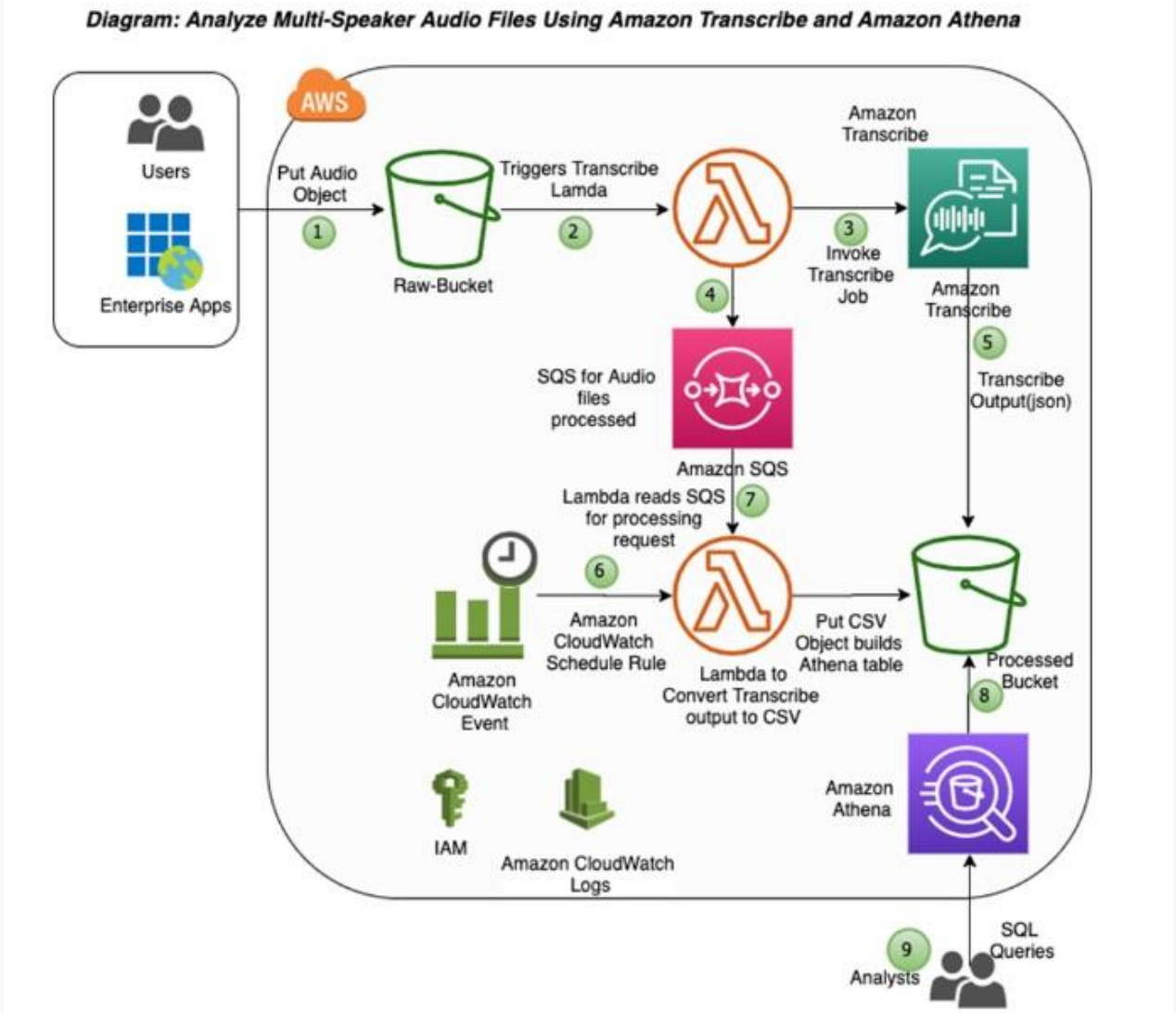
- Use Kinesis Data Streams to read the audio files and machine learning (ML) algorithms to convert the audio files into text and run customer sentiment analysis
- Use Kinesis Data Streams to read the audio files and Amazon Alexa to convert them into text. Kinesis Data Analytics can be used to analyze these files and Amazon Quicksight can be used to visualize and display the output
- Use Amazon Transcribe to convert audio files to text and Amazon Quicksight to run analysis on these text files to understand the underlying patterns. Visualize and display them onto user Dashboards for human analysis
- Use Amazon Transcribe to convert audio files to text and Amazon Athena to understand the underlying customer sentiments **(Correct)**

Explanation
Correct option:

Use Amazon Transcribe to convert audio files to text and Amazon Athena to understand the underlying customer sentiments - Amazon Transcribe is an automatic speech recognition (ASR) service that makes it easy to convert audio to text. One key feature of the service is called speaker identification, which you can use to label each individual speaker when transcribing multi-speaker audio files. You can specify Amazon Transcribe to identify 2–10 speakers in the audio clip.

Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run. To leverage Athena, you can simply point to your data in Amazon S3, define the schema, and start querying using standard SQL. Most results are delivered within seconds.

Analyzing multi-speaker audio files using Amazon Transcribe and Amazon Athena:



via - https://aws.amazon.com/blogs/machine-learning/automating-the-analysis-of-multi-speaker-audio-files-using-amazon-transcribe-and-amazon-athena

Incorrect options:

Use Kinesis Data Streams to read the audio files and machine learning (ML) algorithms to convert the audio files into text and run customer sentiment analysis - Amazon Kinesis can be used to stream real-time data for further analysis and storage. Kinesis Data Streams cannot read audio files. You will still need to use AWS Transcribe for ASR services.

Use Kinesis Data Streams to read the audio files and Amazon Alexa to convert them into text. Kinesis Data Analytics can be used to analyze these files and Amazon Quicksight can be used to visualize and display the output - Kinesis Data Streams

cannot read audio files. Amazon Alexa cannot be used as an Automatic Speech Recognition (ASR) service, though Alexa internally uses ASR for its working.

Use Amazon Transcribe to convert audio files to text and Amazon Quicksight to run analysis on these text files to understand the underlying patterns. Visualize and display them onto user Dashboards for human analysis - Amazon Quicksight is for the visual representation of data through Dashboards, graphs and various other modes. It has a rich feature set that helps analyze data and the complex relationships that exist between different data features. It is, however, not a powerful analysis tool like Amazon Athena.

Question 25:
A big data analytics company is using Kinesis Data Streams (KDS) to process IoT data from the field devices of an agricultural sciences company. Multiple consumer applications are using the incoming data streams and the engineers have noticed a performance lag for the data delivery speed between producers and consumers of the data streams.

As a solutions architect, which of the following would you recommend for improving the performance for the given use-case?

- Swap out Kinesis Data Streams with SQS Standard queues
- Use Enhanced Fanout feature of Kinesis Data Streams **(Correct)**
- Swap out Kinesis Data Streams with Kinesis Data Firehose
- Swap out Kinesis Data Streams with SQS FIFO queues

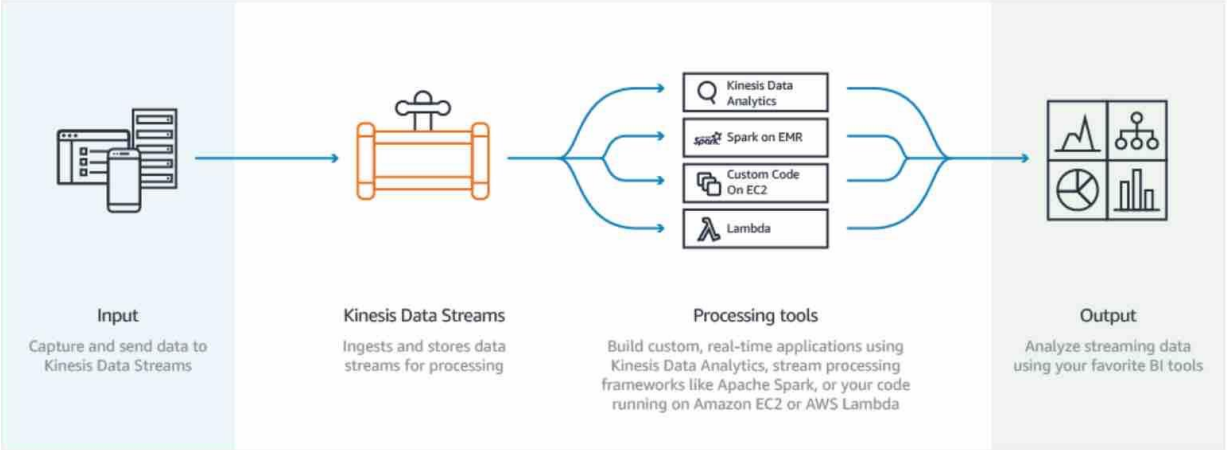
Explanation
Correct option:

Use Enhanced Fanout feature of Kinesis Data Streams

Amazon Kinesis Data Streams (KDS) is a massively scalable and durable real-time data streaming service. KDS can continuously capture gigabytes of data per second from hundreds of thousands of sources such as website clickstreams, database event streams, financial transactions, social media feeds, IT logs, and location-tracking events.

By default, the 2MB/second/shard output is shared between all of the applications consuming data from the stream. You should use enhanced fan-out if you have multiple consumers retrieving data from a stream in parallel. With enhanced fan-out developers can register stream consumers to use enhanced fan-out and receive their own 2MB/second pipe of read throughput per shard, and this throughput automatically scales with the number of shards in a stream.

Kinesis Data Streams
Fanout



Kinesis Data Streams are scaled using the concept of a **shard**. One shard provides an ingest capacity of 1MB/second or 1000 records/second and an output capacity of 2MB/second. It's not uncommon for customers to have thousands or tens of thousands of shards supporting 10s of GB/sec of ingest and egress. Before the enhanced fan-out capability, that 2MB/second/shard output was shared between all of the applications consuming data from the stream. With enhanced fan-out developers can register stream consumers to use enhanced fan-out and receive their own 2MB/second pipe of read throughput per shard, and this throughput automatically scales with the number of shards in a stream. Prior to the launch of Enhanced Fan-out customers would frequently fan-out their data out to multiple streams to support their desired read throughput for their downstream applications. That sounds like undifferentiated heavy lifting to us, and that's something we decided our customers shouldn't need to worry about. Customers pay for enhanced fan-out based on the amount of data retrieved from the stream using enhanced fan-out and the number of consumers registered per-shard. You can find additional info on the pricing page.

via - https://aws.amazon.com/blogs/aws/kds-enhanced-fanout/

Incorrect options:

Swap out Kinesis Data Streams with Kinesis Data Firehose - Amazon Kinesis Data Firehose is the easiest way to reliably load streaming data into data lakes, data stores, and analytics tools. It is a fully managed service that automatically scales to match the throughput of your data and requires no ongoing administration. It can also batch, compress, transform, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security. Kinesis Data Firehose can only write to S3, Redshift, Elasticsearch or Splunk. You can't have applications consuming data streams from Kinesis Data Firehose, that's the job of Kinesis Data Streams. Therefore this option is not correct.

Swap out Kinesis Data Streams with SQS Standard queues

Swap out Kinesis Data Streams with SQS FIFO queues

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. As multiple applications are consuming the same stream concurrently, both SQS Standard and SQS FIFO are not the right fit for the given use-case.

Exam Alert:

Please understand the differences between the capabilities of Kinesis Data Streams vs SQS, as you may be asked scenario-based questions on this topic in the exam.

Q: When should I use Amazon Kinesis Data Streams, and when should I use Amazon SQS?

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

- Routing related records to the same record processor (as in streaming MapReduce). For example, counting and aggregation are simpler when all records for a given key are routed to the same record processor.
- Ordering of records. For example, you want to transfer log data from the application host to the processing/archival host while maintaining the order of log statements.
- Ability for multiple applications to consume the same stream concurrently. For example, you have one application that updates a real-time dashboard and another that archives data to Amazon Redshift. You want both applications to consume data from the same stream concurrently and independently.
- Ability to consume records in the same order a few hours later. For example, you have a billing application and an audit application that runs a few hours behind the billing application. Because Amazon Kinesis Data Streams stores data for up to 7 days, you can run the audit application up to 7 days behind the billing application.

We recommend Amazon SQS for use cases with requirements that are similar to the following:

- Messaging semantics (such as message-level ack/fail) and visibility timeout. For example, you have a queue of work items and want to track the successful completion of each item independently. Amazon SQS tracks the ack/fail, so the application does not have to maintain a persistent checkpoint/cursor. Amazon SQS will delete acked messages and redeliver failed messages after a configured visibility timeout.
- Individual message delay. For example, you have a job queue and need to schedule individual jobs with a delay. With Amazon SQS, you can configure individual messages to have a delay of up to 15 minutes.
- Dynamically increasing concurrency/throughput at read time. For example, you have a work queue and want to add more readers until the backlog is cleared. With Amazon Kinesis Data Streams, you can scale up to a sufficient number of shards (note, however, that you'll need to provision enough shards ahead of time).
- Leveraging Amazon SQS's ability to scale transparently. For example, you buffer requests and the load changes as a result of occasional load spikes or the natural growth of your business. Because each buffered request can be processed independently, Amazon SQS can scale transparently to handle the load without any provisioning instructions from you.

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

Question 26:
The engineering team at a weather tracking company wants to enhance the performance of its relation database and is looking for a caching solution that supports geospatial data.

As a solutions architect, which of the following solutions will you suggest?

- Use Amazon ElastiCache for Redis **(Correct)**
- Use Amazon DynamoDB Accelerator (DAX)
- Use Amazon ElastiCache for Memcached
- Use AWS Global Accelerator

Explanation

Correct option:

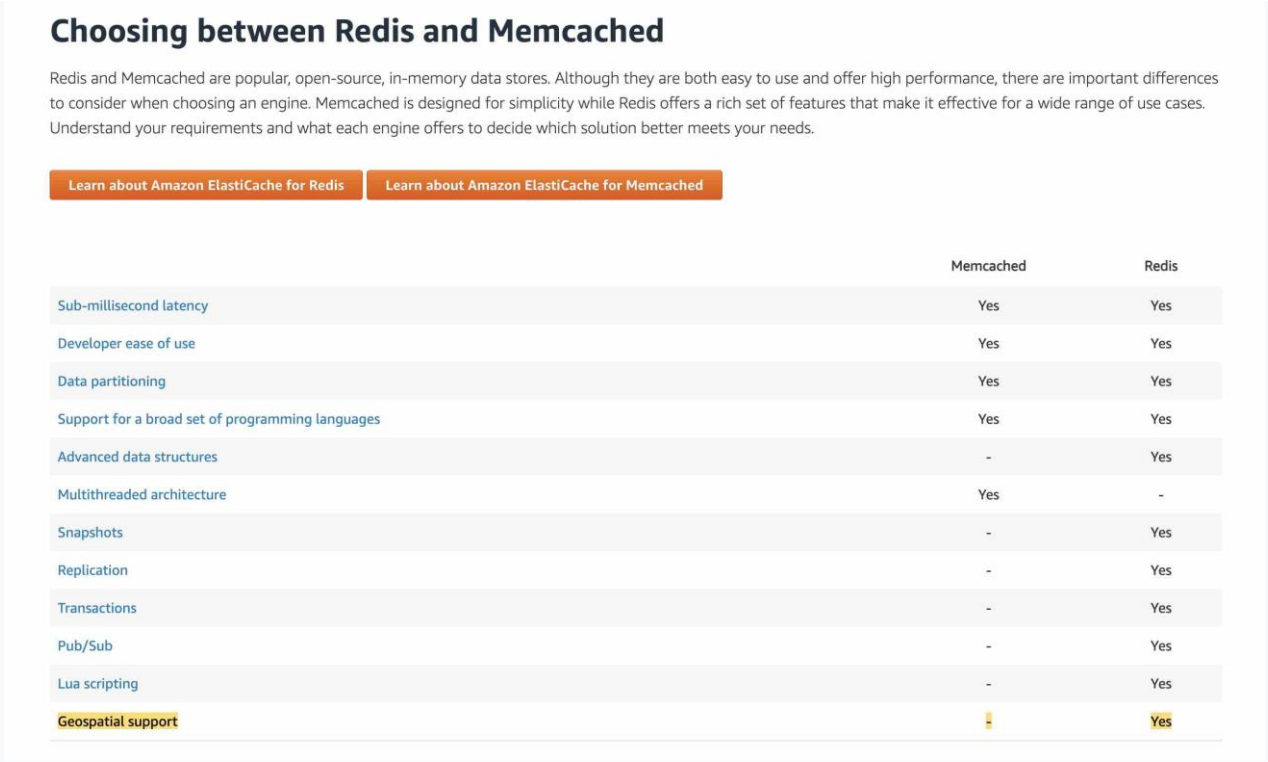
Use Amazon ElastiCache for Redis - Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale a distributed in-memory data store or cache environment in the cloud. Redis, which stands for Remote Dictionary Server, is a fast, open-source, in-memory key-value data store for use as a database, cache, message broker, and queue. Redis now delivers sub-millisecond response times enabling millions of requests per second for real-time applications in Gaming, Ad-Tech, Financial Services, Healthcare, and IoT. Redis is a popular choice for caching, session management, gaming, leaderboards, real-time analytics, geospatial, ride-hailing, chat/messaging, media streaming, and pub/sub apps.

All Redis data resides in the server's main memory, in contrast to databases such as PostgreSQL, Cassandra, MongoDB and others that store most data on disk or on SSDs. In comparison to traditional disk based databases where most operations require a roundtrip to disk, in-memory data stores such as Redis don't suffer the same penalty. They can therefore support an order of magnitude more operations and faster response times. The result is – blazing fast performance with average read or write operations taking less than a millisecond and support for millions of operations per second.

Redis has purpose-built commands for working with real-time geospatial data at scale. You can perform operations like finding the distance between two elements (for example people or places) and finding all elements within a given distance of a point.

Incorrect options:

Use Amazon ElastiCache for Memcached - Both Redis and MemCached are in-memory, open-source data stores. Memcached, a high-performance distributed memory cache service, is designed for simplicity while Redis offers a rich set of features that make it effective for a wide range of use cases. Memcached does not offer support for geospatial data.



Choosing between Redis and Memcached

Redis and Memcached are popular, open-source, in-memory data stores. Although they are both easy to use and offer high performance, there are important differences to consider when choosing an engine. Memcached is designed for simplicity while Redis offers a rich set of features that make it effective for a wide range of use cases. Understand your requirements and what each engine offers to decide which solution better meets your needs.

Learn about Amazon ElastiCache for Redis | Learn about Amazon ElastiCache for Memcached

	Memcached	Redis
Sub-millisecond latency	Yes	Yes
Developer ease of use	Yes	Yes
Data partitioning	Yes	Yes
Support for a broad set of programming languages	Yes	Yes
Advanced data structures	-	Yes
Multithreaded architecture	Yes	-
Snapshots	-	Yes
Replication	-	Yes
Transactions	-	Yes
Pub/Sub	-	Yes
Lua scripting	-	Yes
Geospatial support	-	Yes

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

Amazon DynamoDB Accelerator (DAX) - Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache for Amazon DynamoDB. DAX does not support relational databases.

AWS Global Accelerator - AWS Global Accelerator is a networking service that helps you improve the availability and performance of the applications that you offer to your global users. This option has been added as a distractor, it has nothing to do with database caching.

Question 27:
A retail company wants to establish encrypted network connectivity between its on-premises data center and AWS Cloud. The company wants to get the solution up and running in the fastest possible time and it should also support encryption in transit.

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

- Use AWS Secrets Manager to establish encrypted network connectivity between the on-premises data center and AWS Cloud
- Use AWS Direct Connect to establish encrypted network connectivity between the on-premises data center and AWS Cloud
- Use AWS Data Sync to establish encrypted network connectivity between the on-premises data center and AWS Cloud
-

Use Site-to-Site VPN to establish encrypted network connectivity between the on-premises data center and AWS Cloud
(Correct)

Explanation
Correct options:

Use Site-to-Site VPN to establish encrypted network connectivity between the on-premises data center and AWS Cloud

AWS Site-to-Site VPN enables you to securely connect your on-premises network or branch office site to your Amazon Virtual Private Cloud (Amazon VPC). You can securely extend your data center or branch office network to the cloud with an AWS Site-to-Site VPN connection. A VPC VPN Connection utilizes IPSec to establish encrypted network connectivity between your on-premises network and Amazon VPC over the Internet. IPsec is a protocol suite for securing IP communications by authenticating and encrypting each IP packet in a data stream.

Incorrect options:

Use AWS Direct Connect to establish encrypted network connectivity between the on-premises data center and AWS Cloud - AWS Direct Connect lets you establish a dedicated network connection between your network and one of the AWS Direct Connect locations. Using industry-standard 802.1q VLANs, this dedicated connection can be partitioned into multiple virtual interfaces. AWS Direct Connect does not encrypt your traffic that is in transit. To encrypt the data in transit that traverses AWS Direct Connect, you must use the transit encryption options for that service. As AWS Direct Connect does not support encrypted network connectivity between an on-premises data center and AWS Cloud, therefore this option is incorrect.

Use AWS Data Sync to establish encrypted network connectivity between the on-premises data center and AWS Cloud - AWS DataSync makes it simple and fast to move large amounts of data online between on-premises storage and AWS. DataSync eliminates or automatically handles many of these tasks, including scripting copy jobs, scheduling, and monitoring transfers, validating data, and optimizing network utilization. As AWS Data Sync cannot be used to establish network connectivity between an on-premises data center and AWS Cloud, therefore this option is incorrect.

Use AWS Secrets Manager to establish encrypted network connectivity between the on-premises data center and AWS Cloud - AWS Secrets Manager helps you protect secrets needed to access your applications, services, and IT resources. The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle. As AWS Secrets Manager cannot be used to establish network connectivity between an on-premises data center and AWS Cloud, therefore this option is incorrect.

Question 28:
A multi-national company is looking at optimizing their AWS resources across various countries and regions. They want to understand the best practices on cost optimization, performance, and security for their system architecture spanning across multiple business units.

Which AWS service is the best fit for their requirements?

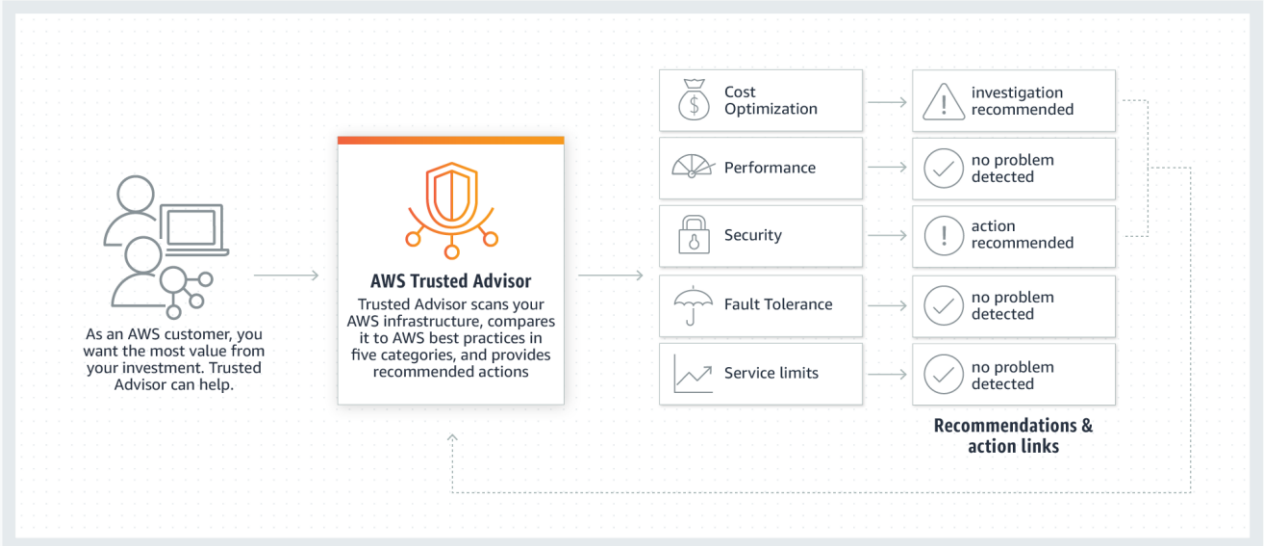
- AWS Config
- AWS Management Console
- AWS Trusted Advisor **(Correct)**
- AWS Systems Manager

Explanation
Correct option:

AWS Trusted Advisor

AWS Trusted Advisor is an online tool that draws upon best practices learned from AWS's aggregated operational history of serving hundreds of thousands of AWS customers. Trusted Advisor inspects your AWS environment and makes recommendations for saving money, improving system performance, or closing security gaps. It scans your AWS infrastructure and compares it to AWS Best practices in five categories (Cost Optimization, Performance, Security, Fault Tolerance, Service limits) and then provides recommendations.

How Trusted Advisor Works:



via - https://aws.amazon.com/premiumsupport/technology/trusted-advisor/

Incorrect options:

AWS Config - AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. You can use Config to answer questions such as - "What did my AWS resource look like at xyz point in time?". It does not offer any feedback about architectural best practices.

AWS Management Console - The AWS Management Console is a web application that comprises and refers to a broad collection of service consoles for managing Amazon Web Services. You log into your AWS account using the AWS Management console. It does not offer any feedback about architectural best practices.

AWS Systems Manager - AWS Systems Manager is an AWS service that you can use to view and control your infrastructure on AWS. Using the Systems Manager console, you can view operational data from multiple AWS services and automate operational tasks across your AWS resources. With Systems Manager, you can group resources, like Amazon EC2 instances, Amazon S3 buckets, or Amazon RDS instances, by application, view operational data for monitoring and troubleshooting, and take action on your groups of resources. It does not offer any feedback about architectural best practices.

Question 29:
The engineering team at an online fashion retailer uses AWS Cloud to manage its technology infrastructure. The EC2 server fleet is behind an Application Load Balancer and the fleet strength is managed by an Auto Scaling group. Based on the historical data, the team is anticipating a huge traffic spike during the upcoming Thanksgiving sale.

As an AWS solutions architect, what feature of the Auto Scaling group would you leverage so that the potential surge in traffic can be preemptively addressed?

- Auto Scaling group lifecycle hook
- Auto Scaling group scheduled action **(Correct)**
- Auto Scaling group target tracking scaling policy
- Auto Scaling group step scaling policy

Explanation
Correct option:

Auto Scaling group scheduled action

The engineering team can create a scheduled action for the Auto Scaling group to pre-emptively provision additional instances for the sale duration. This makes sure that adequate instances are ready before the sale goes live. The scheduled action tells Amazon EC2 Auto Scaling to perform a scaling action at specified times. To create a scheduled scaling action, you specify the start time when the scaling action should take

effect, and the new minimum, maximum, and desired sizes for the scaling action. At the specified time, Amazon EC2 Auto Scaling updates the group with the values for minimum, maximum, and desired size that are specified by the scaling action.

Incorrect options:

Auto Scaling group target tracking scaling policy - With target tracking scaling policies, you choose a scaling metric and set a target value. Application Auto Scaling creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value.

Auto Scaling group step scaling policy - With step scaling, you choose scaling metrics and threshold values for the CloudWatch alarms that trigger the scaling process as well as define how your scalable target should be scaled when a threshold is in breach for a specified number of evaluation periods.

Both the target tracking as well as step scaling policies entail a lag wherein the instances will be provisioned only when the underlying CloudWatch alarms go off. Therefore these two options are not pre-emptive in nature and ruled out for the given use-case.

Auto Scaling group lifecycle hook - Auto Scaling group lifecycle hooks enable you to perform custom actions as the Auto Scaling group launches or terminates instances. For example, you could install or configure software on newly launched instances, or download log files from an instance before it terminates. Lifecycle hooks cannot be used to pre-emptively provision additional instances for a specific period such as the sale duration.

Question 30:
A DevOps engineer at an IT company was recently added to the admin group of the company's AWS account. The `AdministratorAccess` managed policy is attached to this group.

Can you identify the AWS tasks that the DevOps engineer CANNOT perform even though he has full Administrator privileges (Select two)?

- [ ] Configure an Amazon S3 bucket to enable MFA (Multi Factor Authentication) delete **(Correct)**
- [ ] Close the company's AWS account **(Correct)**
- [ ] Delete the IAM user for his manager
- [ ] Delete an S3 bucket from the production environment
- [ ] Change the password for his own IAM user account

Explanation

Correct options:

Configure an Amazon S3 bucket to enable MFA (Multi Factor Authentication) delete

Close the company's AWS account

An IAM user with full administrator access can perform almost all AWS tasks except a few tasks designated only for the root account user. Some of the AWS tasks that only a root account user can do are as follows: change account name or root password or root email address, change AWS support plan, close AWS account, enable MFA on S3 bucket delete, create Cloudfront key pair, register for GovCloud. Even though the DevOps engineer is part of the admin group, he cannot configure an Amazon S3 bucket to enable MFA delete or close the company's AWS account.

Incorrect Options:

Delete the IAM user for his manager

Delete an S3 bucket from the production environment

Change the password for his own IAM user account

The DevOps engineer is part of the admin group, so he can delete any IAM user, delete the S3 bucket, and change the password for his own IAM user account.

Question 31:
A startup has created a cost-effective backup solution in another AWS Region. The application is running in warm standby mode and has Application Load Balancer (ALB) to support it from the front. The current failover process is manual and requires updating the DNS alias record to point to the secondary ALB in another Region in case of failure of the primary ALB.

As a Solutions Architect, what will you recommend to automate the failover process?

- Configure Trusted Advisor to check on unhealthy instances
- Enable an Amazon Route 53 health check **(Correct)**
- Enable an EC2 instance health check
- Enable an ALB health check

Explanation
Correct option:

Enable an Amazon Route 53 health check - Determining the health of an ELB endpoint is more complex than health checking a single IP address. For example, what if your application is running fine on EC2, but the load balancer itself isn't reachable? Or if your load balancer and your EC2 instances are working correctly, but a bug in your code

causes your application to crash? Or how about if the EC2 instances in one Availability Zone of a multi-AZ ELB are experiencing problems?

Route 53 DNS Failover handles all of these failure scenarios by integrating with ELB behind the scenes. Once enabled, Route 53 automatically configures and manages health checks for individual ELB nodes. Route 53 also takes advantage of the EC2 instance health checking that ELB performs (information on configuring your ELB health checks is available here). By combining the results of health checks of your EC2 instances and your ELBs, Route 53 DNS Failover can evaluate the health of the load balancer and the health of the application running on the EC2 instances behind it. In other words, if any part of the stack goes down, Route 53 detects the failure and routes traffic away from the failed endpoint.

Using Route 53 DNS Failover, you can run your primary application simultaneously in multiple AWS regions around the world and failover across regions. Your end-users will be routed to the closest (by latency), healthy region for your application. Route 53 automatically removes from service any region where your application is unavailable - it will pull an endpoint out of service if there is region-wide connectivity or operational issue, if your application goes down in that region, or if your ELB or EC2 instances go down in that region.

Incorrect options:

Enable an ALB health check - ELB health check verifies that a specified TCP port on an instance is accepting connections or a specified page has returned an error code of 200. It is not useful for the given failover scenario.

Enable an EC2 instance health check - Instance status checks monitor the software and network configuration of your instance. It is not intelligent enough to understand if the application on the instance is working correctly. Hence, this is not the right choice for the given use-case.

Configure Trusted Advisor to check on unhealthy instances - AWS Trusted Advisor examines the health check configuration for Auto Scaling groups. If Elastic Load Balancing is being used for an Auto Scaling group, the recommended configuration is to enable an Elastic Load Balancing health check. Trusted Advisor recommends certain configuration changes by comparing your system configurations to AWS Best practices. It cannot handle a failover the way Route 53 does.

Question 32:
You have built an application that is deployed with an Elastic Load Balancer and an Auto Scaling Group. As a Solutions Architect, you have configured aggressive CloudWatch alarms, making your Auto Scaling Group (ASG) scale in and out very quickly, renewing your fleet of Amazon EC2 instances on a daily basis. A production bug appeared two days ago, but the team is unable to SSH into the instance to debug the issue, because the instance has already been terminated by the ASG. The log files are saved on the EC2 instance.

How will you resolve the issue and make sure it doesn't happen again?

- Install a CloudWatch Logs agents on the EC2 instances to send logs to CloudWatch **(Correct)**
- Disable the Termination from the ASG any time a user reports an issue
- Use AWS Lambda to regularly SSH into the EC2 instances and copy the log files to S3
- Make a snapshot of the EC2 instance just before it gets terminated

Explanation
Correct option:

Install a CloudWatch Logs agents on the EC2 instances to send logs to CloudWatch

You can use the CloudWatch Logs agent installer on an existing EC2 instance to install and configure the CloudWatch Logs agent. After installation is complete, logs automatically flow from the instance to the log stream you create while installing the agent. The agent confirms that it has started and it stays running until you disable it.

Here, the natural and by far the easiest solution would be to use the CloudWatch Logs agents on the EC2 instances to automatically send log files into CloudWatch, so we can analyze them in the future easily should any problem arise.

To control whether an Auto Scaling group can terminate a particular instance when scaling in, use instance scale-in protection. You can enable the instance scale-in protection setting on an Auto Scaling group or on an individual Auto Scaling instance. When the Auto Scaling group launches an instance, it inherits the instance scale-in protection setting of the Auto Scaling group. You can change the instance scale-in protection setting for an Auto Scaling group or an Auto Scaling instance at any time.

Incorrect options:

Disable the Termination from the ASG any time a user reports an issue - Disabling the Termination from the ASG would prevent our ASG from being Elastic and impact our costs. Therefore this option is incorrect.

Make a snapshot of the EC2 instance just before it gets terminated - Making a snapshot of the EC2 instance before it gets terminated *could* work but it's tedious, not elastic and

very expensive, since our interest is just the log files. Therefore this option is not the best fit for the given use-case.

You can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data.

Use AWS Lambda to regularly SSH into the EC2 instances and copy the log files to S3 - AWS Lambda lets you run code without provisioning or managing servers. It cannot be used for production-grade serverless log analytics. Using AWS Lambda would be extremely hard to use for this task. Therefore this option is not the best fit for the given use-case.

Question 33:
A company wants to publish an event into an SQS queue whenever a new object is uploaded on S3.

Which of the following statements are true regarding this functionality?

- Neither Standard SQS queue nor FIFO SQS queue are allowed as an Amazon S3 event notification destination
- Only Standard SQS queue is allowed as an Amazon S3 event notification destination, whereas FIFO SQS queue is not allowed **(Correct)**
- Only FIFO SQS queue is allowed as an Amazon S3 event notification destination, whereas Standard SQS queue is not allowed
- Both Standard SQS queue and FIFO SQS queue are allowed as an Amazon S3 event notification destination

Explanation
Correct option:

Only Standard SQS queue is allowed as an Amazon S3 event notification destination, whereas FIFO SQS queue is not allowed

The Amazon S3 notification feature enables you to receive notifications when certain events happen in your bucket. To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

Amazon S3 supports the following destinations where it can publish events:

Amazon Simple Notification Service (Amazon SNS) topic

Amazon Simple Queue Service (Amazon SQS) queue

AWS Lambda

Currently, the Standard SQS queue is only allowed as an Amazon S3 event notification destination, whereas the FIFO SQS queue is not allowed.

Incorrect options:

Both Standard SQS queue and FIFO SQS queue are allowed as an Amazon S3 event notification destination

Neither Standard SQS queue nor FIFO SQS queue is allowed as an Amazon S3 event notification destination

Only FIFO SQS queue is allowed as an Amazon S3 event notification destination, whereas Standard SQS queue is not allowed

These three options contradict the details provided in the explanation above. To summarize, the Standard SQS queue is only allowed as an Amazon S3 event notification destination, whereas the FIFO SQS queue is not allowed. Hence these three options are incorrect.

Question 34:
Computer vision researchers at a university are trying to optimize the I/O bound processes for a proprietary algorithm running on EC2 instances. The ideal storage would facilitate high-performance IOPS when doing file processing in a temporary storage space before uploading the results back into Amazon S3.

As a solutions architect, which of the following AWS storage options would you recommend as the MOST performant as well as cost-optimal?

- Use EC2 instances with EBS Throughput Optimized HDD (st1) as the storage option
- Use EC2 instances with EBS Provisioned IOPS SSD (io1) as the storage option
- Use EC2 instances with Instance Store as the storage option **(Correct)**
- Use EC2 instances with EBS General Purpose SSD (gp2) as the storage option

Explanation
Correct option:

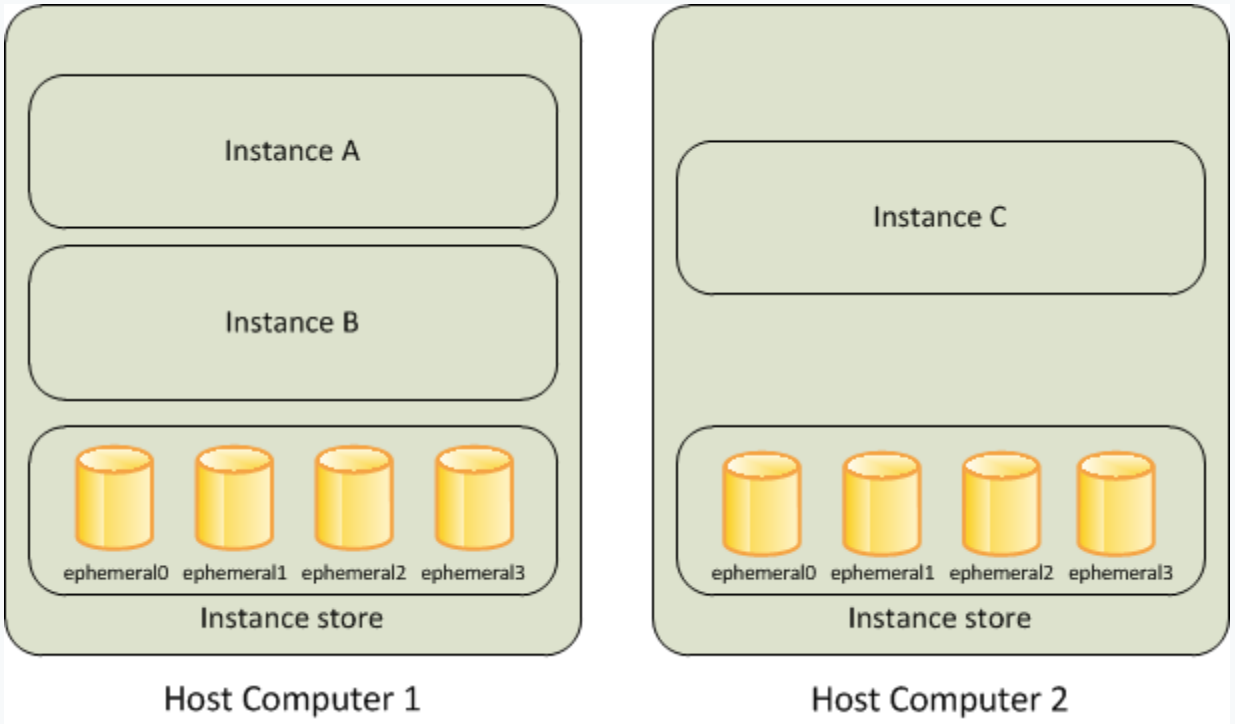
Use EC2 instances with Instance Store as the storage type

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. Some instance types use NVMe or SATA-based solid-state drives (SSD) to deliver high random I/O performance. This is a good option when you need storage with very low latency, but you don't need the data to persist when the instance terminates or you can take advantage of fault-tolerant architectures.

As Instance Store delivers high random I/O performance, it can act as a temporary storage space, and these volumes are included as part of the instance's usage cost, therefore this is the correct option.

Amazon EC2 Instance Store



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

Incorrect options:

Use EC2 instances with EBS General Purpose SSD (gp2) as the storage option - General Purpose SSD (gp2) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 16,000 IOPS (at 5,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. AWS designs gp2 volumes to deliver its provisioned performance 99% of the time. A gp2 volume can range in size from 1 GiB to 16 TiB. EBS gp2 is persistent storage and costlier than Instance Stores (the cost of the storage volume is in addition to that of the EC2 instance), therefore this option is not correct.

Use EC2 instances with EBS Provisioned IOPS SSD (io1) as the storage option - Provisioned IOPS SSD (io1) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency. Unlike gp2, which uses a bucket and credit model to calculate performance, an io1 volume allows you to specify a consistent IOPS rate when you create the volume, and Amazon EBS delivers the provisioned performance 99.9 percent

of the time. EBS io1 is persistent storage and costlier than Instance Stores (the cost of the storage volume is in addition to that of the EC2 instance), therefore this option is not correct.

Use EC2 instances with EBS Throughput Optimized HDD (st1) as the storage option - Throughput Optimized HDD (st1) are low-cost HDD volumes designed for frequently accessed, throughput-intensive workloads such as Big data and Data warehouses. EBS st1 is persistent storage and costlier than Instance Stores (the cost of the storage volume is in addition to that of the EC2 instance), therefore this option is not correct.

Question 35:
An application with global users across AWS Regions had suffered an issue when the Elastic Load Balancer (ELB) in a Region malfunctioned thereby taking down the traffic with it. The manual intervention cost the company significant time and resulted in major revenue loss.

What should a solutions architect recommend to reduce internet latency and add automatic failover across AWS Regions?

- Set up AWS Direct Connect as the backbone for each of the AWS Regions where the application is deployed
- Create S3 buckets in different AWS Regions and configure CloudFront to pick the nearest edge location to the user
- Set up an Amazon Route 53 geoproximity routing policy to route traffic
- Set up AWS Global Accelerator and add endpoints to cater to users in different geographic locations **(Correct)**

Explanation
Correct option:

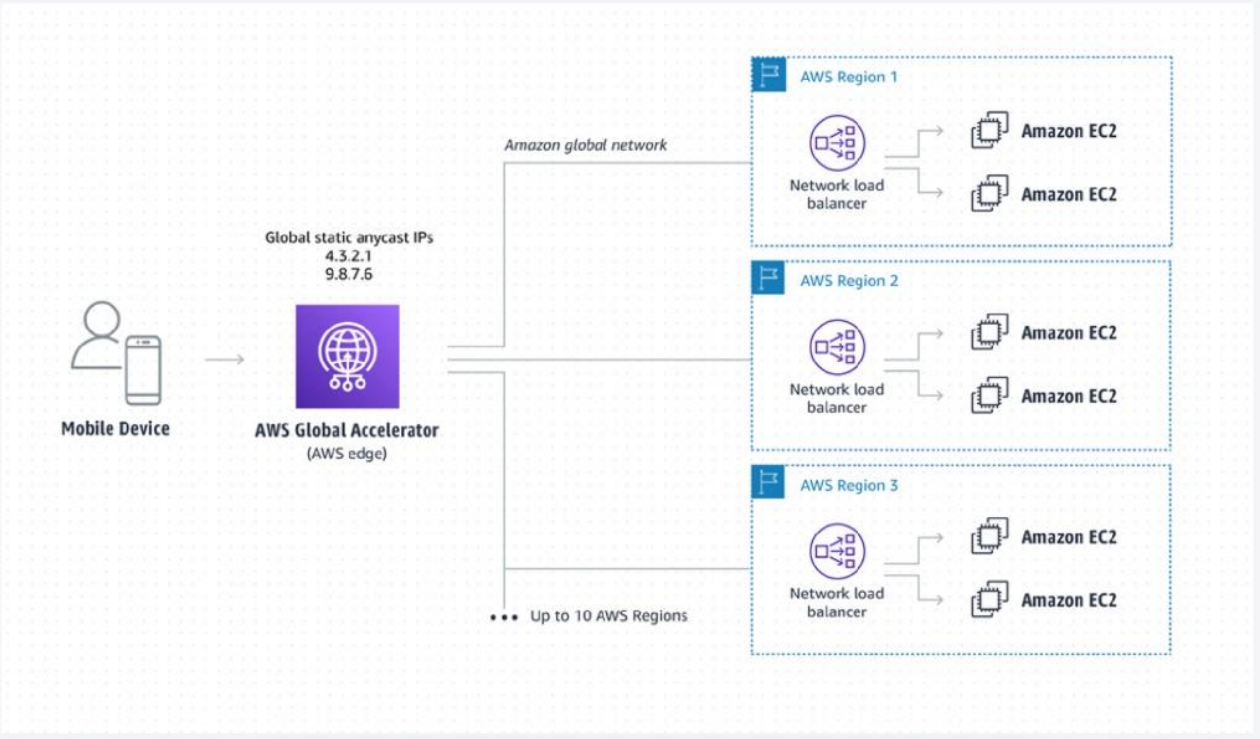
Set up AWS Global Accelerator and add endpoints to cater to users in different geographic locations

As your application architecture grows, so does the complexity, with longer user-facing IP lists and more nuanced traffic routing logic. AWS Global Accelerator solves this by providing you with two static IPs that are anycast from our globally distributed edge locations, giving you a single entry point to your application, regardless of how many AWS Regions it's deployed in. This allows you to add or remove origins, Availability Zones or Regions without reducing your application availability. Your traffic routing is managed manually, or in console with endpoint traffic dials and weights. If your application endpoint has a failure or availability issue, AWS Global Accelerator will automatically redirect your new connections to a healthy endpoint within seconds.

By using AWS Global Accelerator, you can:

1. Associate the static IP addresses provided by AWS Global Accelerator to regional AWS resources or endpoints, such as Network Load Balancers, Application Load Balancers, EC2 Instances, and Elastic IP addresses. The IP addresses are anycast from AWS edge locations so they provide onboarding to the AWS global network close to your users.
2. Easily move endpoints between Availability Zones or AWS Regions without needing to update your DNS configuration or change client-facing applications.
3. Dial traffic up or down for a specific AWS Region by configuring a traffic dial percentage for your endpoint groups. This is especially useful for testing performance and releasing updates.
4. Control the proportion of traffic directed to each endpoint within an endpoint group by assigning weights across the endpoints.

AWS Global Accelerator for Multi-Region applications:



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

Incorrect options:

Set up AWS Direct Connect as the backbone for each of the AWS Regions where the application is deployed - AWS Direct Connect can reduce latency to great extent. Direct Connect is used to connect on-premises systems to AWS Cloud for extremely low latency use cases. It cannot be used to serve users directly.

Create S3 buckets in different AWS Regions and configure CloudFront to pick the nearest edge location to the user - If most of the content is static, we can configure CloudFront to improve performance. In the current scenario, the architecture has ELBs, EC2 instances too that need to be covered in the automatic failover plan.

*Set up an Amazon Route 53 geoproximity routing policy to route traffic** - Geoproximity routing lets Amazon Route 53 route traffic to your resources based on the geographic location of your users and your resources. Unlike Global Accelerator, managing and

routing to different instances, ELBs and other AWS resources will become an operational overhead as the resource count reaches into the hundreds. With inbuilt features like Static anycast IP addresses, fault tolerance using network zones, Global performance-based routing, TCP Termination at the Edge - Global Accelerator is the right choice for multi-region, low latency use cases.

Question 36:
An e-commerce company uses Amazon SQS queues to decouple their application architecture. The engineering team has observed message processing failures for some customer orders.

As a solutions architect, which of the following solutions would you recommend for handling such message failures?

- Use a dead-letter queue to handle message processing failures **(Correct)**
- Use long polling to handle message processing failures
- Use a temporary queue to handle message processing failures
- Use short polling to handle message processing failures

Explanation
Correct option:

Use a dead-letter queue to handle message processing failures

Dead-letter queues can be used by other queues (source queues) as a target for messages that can't be processed (consumed) successfully. Dead-letter queues are useful for debugging your application or messaging system because they let you isolate problematic messages to determine why their processing doesn't succeed. Sometimes, messages can’t be processed because of a variety of possible issues, such as when a user comments on a story but it remains unprocessed because the original story itself is deleted by the author while the comments were being posted. In such a case, the dead-letter queue can be used to handle message processing failures.

How do dead-letter queues
work?

How do dead-letter queues work?

Sometimes, messages can't be processed because of a variety of possible issues, such as erroneous conditions within the producer or consumer application or an unexpected state change that causes an issue with your application code. For example, if a user places a web order with a particular product ID, but the product ID is deleted, the web store's code fails and displays an error, and the message with the order request is sent to a dead-letter queue.

Occasionally, producers and consumers might fail to interpret aspects of the protocol that they use to communicate, causing message corruption or loss. Also, the consumer's hardware errors might corrupt message payload.

The *redrive policy* specifies the *source queue*, the *dead-letter queue*, and the conditions under which Amazon SQS moves messages from the former to the latter if the consumer of the source queue fails to process a message a specified number of times. When the `ReceiveCount` for a message exceeds the `maxReceiveCount` for a queue, Amazon SQS moves the message to a dead-letter queue (with its original message ID). For example, if the source queue has a redrive policy with `maxReceiveCount` set to 5, and the consumer of the source queue receives a message 6 times without ever deleting it, Amazon SQS moves the message to the dead-letter queue.

via
- https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-dead-letter-queues.html

Incorrect options:

Use a temporary queue to handle message processing failures - The most common use case for temporary queues is the request-response messaging pattern (for example, processing a login request), where a requester creates a temporary queue for receiving each response message. To avoid creating an Amazon SQS queue for each response message, the Temporary Queue Client lets you create and delete multiple temporary queues without making any Amazon SQS API calls. Temporary queues cannot be used to handle message processing failures.

Use short polling to handle message processing failures

Use long polling to handle message processing failures

Amazon SQS provides short polling and long polling to receive messages from a queue. By default, queues use short polling. With short polling, Amazon SQS sends the response right away, even if the query found no messages. With long polling, Amazon SQS sends a response after it collects at least one available message, up to the maximum number of messages specified in the request. Amazon SQS sends an empty response only if the polling wait time expires. Neither short polling nor long polling can be used to handle message processing failures.

Question 37:
A media company wants to get out of the business of owning and maintaining its own IT infrastructure. As part of this digital transformation, the media company wants to archive about 5PB of data in its on-premises data center to durable long term storage.

As a solutions architect, what is your recommendation to migrate this data in the MOST cost-optimal way?

- Transfer the on-premises data into multiple Snowball Edge Storage Optimized devices. Copy the Snowball Edge data into Amazon S3 and create a lifecycle policy to transition the data into AWS Glacier
(Correct)
- Setup Site-to-Site VPN connection between the on-premises data center and AWS Cloud. Use this connection to transfer the data into AWS Glacier
- Transfer the on-premises data into multiple Snowball Edge Storage Optimized devices. Copy the Snowball Edge data into AWS Glacier
- Setup AWS direct connect between the on-premises data center and AWS Cloud. Use this connection to transfer the data into AWS Glacier

Explanation
Correct option:

Transfer the on-premises data into multiple Snowball Edge Storage Optimized devices. Copy the Snowball Edge data into Amazon S3 and create a lifecycle policy to transition the data into AWS Glacier

Snowball Edge Storage Optimized is the optimal choice if you need to securely and quickly transfer dozens of terabytes to petabytes of data to AWS. It provides up to 80 TB of usable HDD storage, 40 vCPUs, 1 TB of SATA SSD storage, and up to 40 Gb network connectivity to address large scale data transfer and pre-processing use cases. The data stored on the Snowball Edge device can be copied into the S3 bucket and later transitioned into AWS Glacier via a lifecycle policy. You can't directly copy data from Snowball Edge devices into AWS Glacier.

Incorrect options:

Transfer the on-premises data into multiple Snowball Edge Storage Optimized devices. Copy the Snowball Edge data into AWS Glacier - As mentioned earlier, you can't directly copy data from Snowball Edge devices into AWS Glacier. Hence, this option is incorrect.

Setup AWS direct connect between the on-premises data center and AWS Cloud. Use this connection to transfer the data into AWS Glacier - 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. Direct Connect involves significant monetary investment and takes more than a month to set up, therefore it's not the correct fit for this use-case where just a one-time data transfer has to be done.

Setup Site-to-Site VPN connection between the on-premises data center and AWS Cloud. Use this connection to transfer the data into AWS Glacier - 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). VPN Connections are a good solution if you have an immediate need, and have low to modest bandwidth requirements. Because of the high data volume for the given use-case, Site-to-Site VPN is not the correct choice.

Question 38:
Your firm has implemented a multi-tiered networking structure within the VPC - with two public and two private subnets. The public subnets are used to deploy the Application Load Balancers, while the two private subnets are used to deploy the application on Amazon EC2 instances. The development team wants the EC2 instances to have access to the internet. The solution has to be fully managed by AWS and needs to work over IPv4.

What will you recommend?

- Internet Gateways deployed in your private subnet
- NAT Gateways deployed in your public subnet **(Correct)**
- Egress-Only Internet Gateways deployed in your private subnet
- NAT Instances deployed in your public subnet

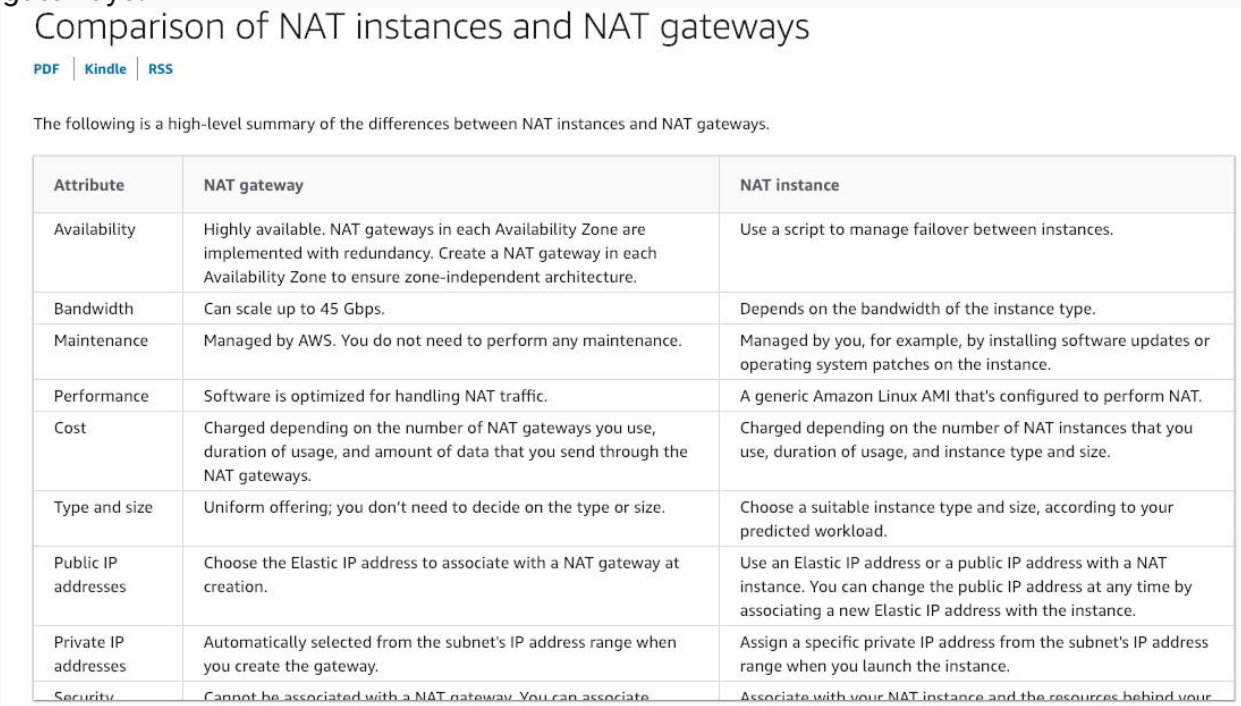
Explanation
Correct option:

NAT Gateways deployed in your public subnet - You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances. A NAT gateway has the following characteristics and limitations:

1. A NAT gateway supports 5 Gbps of bandwidth and automatically scales up to 45 Gbps.
2. You can associate exactly one Elastic IP address with a NAT gateway.
3. A NAT gateway supports the following protocols: TCP, UDP, and ICMP.
4. You cannot associate a security group with a NAT gateway.
5. You can use a network ACL to control the traffic to and from the subnet in which the NAT gateway is located.
6. A NAT gateway can support up to 55,000 simultaneous connections to each unique destination.

Therefore you must use a NAT Gateway in your public subnet in order to provide internet access to your instances in your private subnets. You are charged for creating and using a NAT gateway in your account. NAT gateway hourly usage and data processing rates apply.

Comparison of NAT instances and NAT gateways:



Comparison of NAT instances and NAT gateways

PDF | Kindle | RSS

The following is a high-level summary of the differences between NAT instances and NAT gateways.

Attribute	NAT gateway	NAT instance
Availability	Highly available. NAT gateways in each Availability Zone are implemented with redundancy. Create a NAT gateway in each Availability Zone to ensure zone-independent architecture.	Use a script to manage failover between instances.
Bandwidth	Can scale up to 45 Gbps.	Depends on the bandwidth of the instance type.
Maintenance	Managed by AWS. You do not need to perform any maintenance.	Managed by you, for example, by installing software updates or operating system patches on the instance.
Performance	Software is optimized for handling NAT traffic.	A generic Amazon Linux AMI that's configured to perform NAT.
Cost	Charged depending on the number of NAT gateways you use, duration of usage, and amount of data that you send through the NAT gateways.	Charged depending on the number of NAT instances that you use, duration of usage, and instance type and size.
Type and size	Uniform offering; you don't need to decide on the type or size.	Choose a suitable instance type and size, according to your predicted workload.
Public IP addresses	Choose the Elastic IP address to associate with a NAT gateway at creation.	Use an Elastic IP address or a public IP address with a NAT instance. You can change the public IP address at any time by associating a new Elastic IP address with the instance.
Private IP addresses	Automatically selected from the subnet's IP address range when you create the gateway.	Assign a specific private IP address from the subnet's IP address range when you launch the instance.
Security	Cannot be associated with a NAT gateway. You can associate	Associate with your NAT instance and the resources behind your

via - https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-comparison.html

Incorrect options:

NAT Instances deployed in your public subnet - You can use a network address translation (NAT) instance in a public subnet in your VPC to enable instances in the private subnet to initiate outbound IPv4 traffic to the Internet or other AWS services, but prevent the instances from receiving inbound traffic initiated by someone on the Internet. Amazon provides Amazon Linux AMIs that are configured to run as NAT instances. These AMIs include the string amzn-ami-vpc-nat in their names, so you can search for them in the Amazon EC2 console. NAT Instances would work but won't scale and you would have to manage them (as they're nothing but EC2 instances).

Internet Gateways deployed in your private subnet - An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It, therefore, imposes no availability risks or bandwidth constraints on your network traffic. Internet Gateways must be deployed in a public subnet, hence not an option here.

Egress-Only Internet Gateways deployed in your private subnet - An Egress-Only Internet Gateway is a horizontally scaled, redundant, and highly available VPC component that allows outbound communication over IPv6 from instances in your VPC to the Internet, and prevents the Internet from initiating an IPv6 connection with your instances. Egress-Only Internet Gateways are for IPv6, not IPv4. Therefore, this option is incorrect.

Question 39:
A silicon valley based healthcare startup uses AWS Cloud for its IT infrastructure. The startup stores patient health records on Amazon S3. The engineering team needs to implement an archival solution based on Amazon S3 Glacier to enforce regulatory and compliance controls on data access.

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

- Use S3 Glacier vault to store the sensitive archived data and then use an S3 Access Control List to enforce compliance controls
- Use S3 Glacier vault to store the sensitive archived data and then use a vault lock policy to enforce compliance controls **(Correct)**
- Use S3 Glacier to store the sensitive archived data and then use an S3 lifecycle policy to enforce compliance controls
- Use S3 Glacier to store the sensitive archived data and then use an S3 Access Control List to enforce compliance controls

Explanation
Correct option:

Use S3 Glacier vault to store the sensitive archived data and then use a vault lock policy to enforce compliance controls

Amazon S3 Glacier is a secure, durable, and extremely low-cost Amazon S3 cloud storage class for data archiving and long-term backup. It is designed to deliver 99.999999999% durability, and provide comprehensive security and compliance capabilities that can help meet even the most stringent regulatory requirements.

An S3 Glacier vault is a container for storing archives. When you create a vault, you specify a vault name and the AWS Region in which you want to create the vault. S3 Glacier Vault Lock allows you to easily deploy and enforce compliance controls for individual S3 Glacier vaults with a vault lock policy. You can specify controls such as “write once read many” (WORM) in a vault lock policy and lock the policy from future edits. Therefore, this is the correct option.

Incorrect options:

Use S3 Glacier to store the sensitive archived data and then use an S3 lifecycle policy to enforce compliance controls - You can use lifecycle policy to define actions you want Amazon S3 to take during an object's lifetime. For example, use a lifecycle policy to transition objects to another storage class, archive them, or delete them after a specified period. It cannot be used to enforce compliance controls. Therefore, this option is incorrect.

Use S3 Glacier vault to store the sensitive archived data and then use an S3 Access Control List to enforce compliance controls - Amazon S3 access control lists (ACLs) enable you to manage access to buckets and objects. It cannot be used to enforce compliance controls. Therefore, this option is incorrect.

Use S3 Glacier to store the sensitive archived data and then use an S3 Access Control List to enforce compliance controls - Amazon S3 access control lists (ACLs) enable you

to manage access to buckets and objects. It cannot be used to enforce compliance controls. Therefore, this option is incorrect.

Question 40:
A pharma company is working on developing a vaccine for the COVID-19 virus. The researchers at the company want to process the reference healthcare data in an in-memory database that is highly available as well as HIPAA compliant.

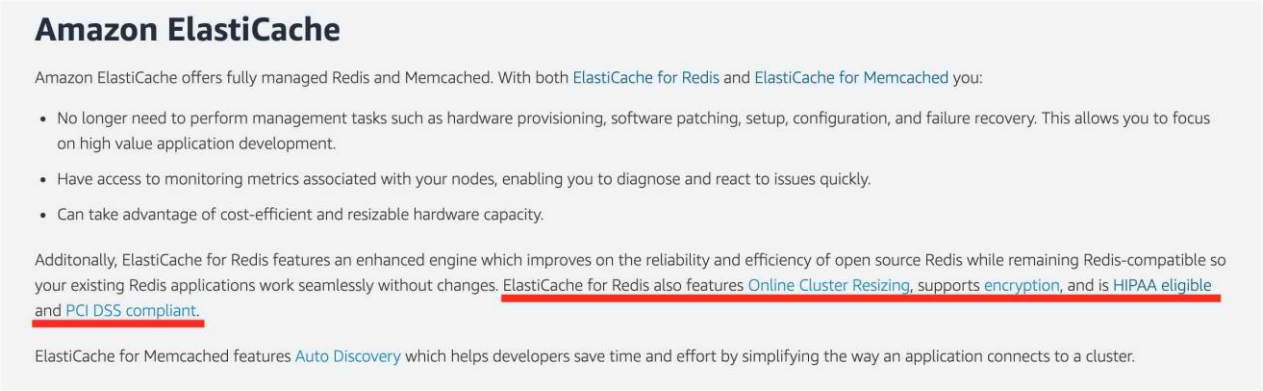
As a solutions architect, which of the following AWS services would you recommend for this task?

- DynamoDB
- DocumentDB
- ElastiCache for Redis **(Correct)**
- ElastiCache for Memcached

Explanation
Correct option:

ElastiCache for Redis

ElastiCache
Overview:



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

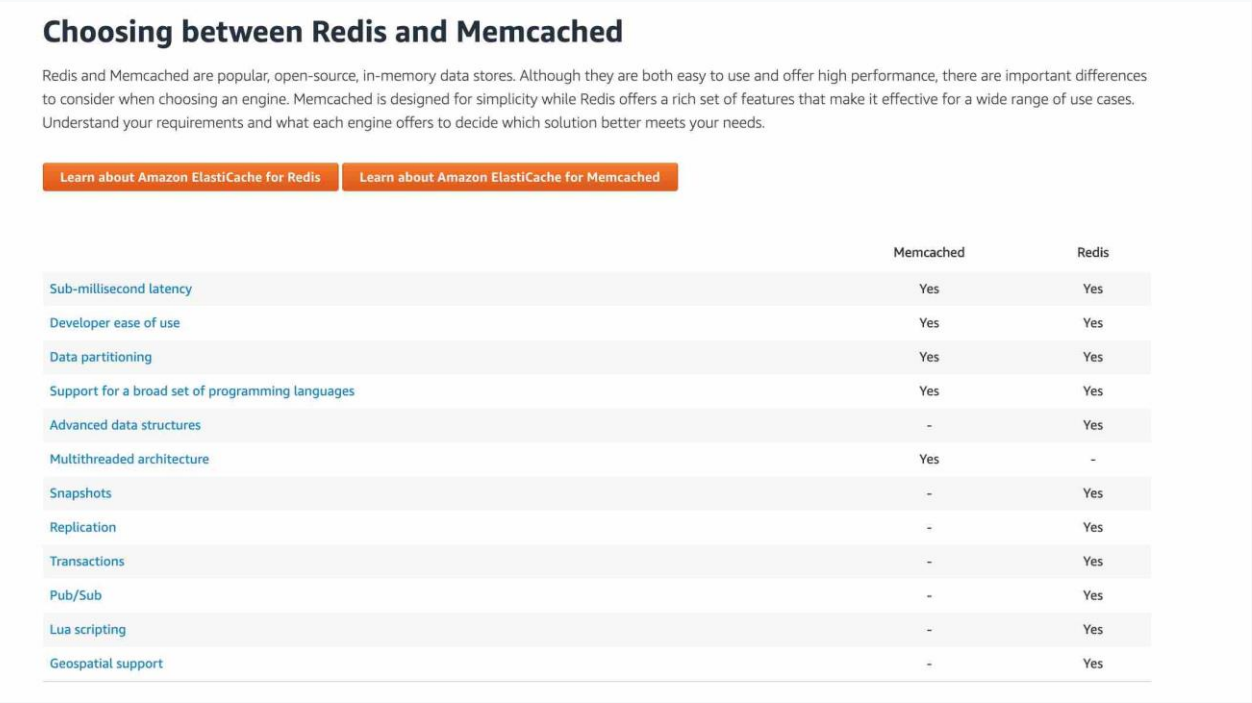
Amazon ElastiCache for Redis is a blazing fast in-memory data store that provides sub-millisecond latency to power internet-scale real-time applications. Amazon ElastiCache for Redis is a great choice for real-time transactional and analytical processing use cases such as caching, chat/messaging, gaming leaderboards, geospatial, machine learning, media streaming, queues, real-time analytics, and session store. ElastiCache for Redis supports replication, high availability, and cluster sharding right out of the box. Amazon ElastiCache for Redis is also HIPAA Eligible Service. Therefore, this is the correct option.

ElastiCache for Redis
Overview:

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

Exam Alert:

Please review this comparison sheet for Redis vs Memcached features:



Choosing between Redis and Memcached

Redis and Memcached are popular, open-source, in-memory data stores. Although they are both easy to use and offer high performance, there are important differences to consider when choosing an engine. Memcached is designed for simplicity while Redis offers a rich set of features that make it effective for a wide range of use cases. Understand your requirements and what each engine offers to decide which solution better meets your needs.

Learn about Amazon ElastiCache for Redis | Learn about Amazon ElastiCache for Memcached

	Memcached	Redis
Sub-millisecond latency	Yes	Yes
Developer ease of use	Yes	Yes
Data partitioning	Yes	Yes
Support for a broad set of programming languages	Yes	Yes
Advanced data structures	-	Yes
Multithreaded architecture	Yes	-
Snapshots	-	Yes
Replication	-	Yes
Transactions	-	Yes
Pub/Sub	-	Yes
Lua scripting	-	Yes
Geospatial support	-	Yes

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

Incorrect Options:

ElastiCache for Memcached - Amazon ElastiCache for Memcached is a Memcached-compatible in-memory key-value store service that can be used as a cache or a data store. Amazon ElastiCache for Memcached is a great choice for implementing an in-memory cache to decrease access latency, increase throughput, and ease the load off your relational or NoSQL database. Session stores are easy to create with Amazon ElastiCache for Memcached. Elasticache for Memcached is not HIPAA eligible, so this option is incorrect.

DynamoDB - Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-region,

multi-master, durable database with built-in security, backup and restore, and in-memory caching (via DAX) for internet-scale applications. DynamoDB is not an in-memory database, so this option is incorrect.

DocumentDB - Amazon DocumentDB is a fast, scalable, highly available, and fully managed document database service that supports MongoDB workloads. As a document database, Amazon DocumentDB makes it easy to store, query, and index JSON data. DocumentDB is not an in-memory database, so this option is incorrect.

Question 41:
A mobile chat application uses DynamoDB as its database service to provide low latency chat updates. A new developer has joined the team and is reviewing the configuration settings for DynamoDB which have been tweaked for certain technical requirements. CloudTrail service has been enabled on all the resources used for the project. Yet, DynamoDB encryption details are nowhere to be found.

Which of the following options can explain the root cause for the given issue?

- By default, all DynamoDB tables are encrypted using Data keys, which do not write to CloudTrail logs
- By default, all DynamoDB tables are encrypted under AWS managed CMKs, which do not write to CloudTrail logs
- By default, all DynamoDB tables are encrypted under Customer managed CMKs, which do not write to CloudTrail logs
- By default, all DynamoDB tables are encrypted under an AWS owned customer master key (CMK), which do not write to CloudTrail logs **(Correct)**

Explanation
Correct option:

By default, all DynamoDB tables are encrypted under an AWS owned customer master key (CMK), which do not write to CloudTrail logs - AWS owned CMKs are a collection of CMKs that an AWS service owns and manages for use in multiple AWS accounts. Although AWS owned CMKs are not in your AWS account, an AWS service can use its AWS owned CMKs to protect the resources in your account.

You do not need to create or manage the AWS owned CMKs. However, you cannot view, use, track, or audit them. You are not charged a monthly fee or usage fee for AWS owned CMKs and they do not count against the AWS KMS quotas for your account.

The key rotation strategy for an AWS owned CMK is determined by the AWS service that creates and manages the CMK.

All DynamoDB tables are encrypted. There is no option to enable or disable encryption for new or existing tables. By default, all tables are encrypted under an AWS owned customer master key (CMK) in the DynamoDB service account. However, you can select

an option to encrypt some or all of your tables under a customer-managed CMK or the AWS managed CMK for DynamoDB in your account.

Incorrect options:

By default, all DynamoDB tables are encrypted under AWS managed CMKs, which do not write to CloudTrail logs

By default, all DynamoDB tables are encrypted under Customer managed CMKs, which do not write to CloudTrail logs

By default, all DynamoDB tables are encrypted using Data keys, which do not write to CloudTrail logs

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

Question 42:
A global media company uses a fleet of EC2 instances (behind an Application Load Balancer) to power its video streaming application. To improve the performance of the application, the engineering team has also created a CloudFront distribution with the Application Load Balancer as the custom origin. The security team at the company has noticed a spike in the number and types of SQL injection and cross-site scripting attack vectors on the application.

As a solutions architect, which of the following solutions would you recommend as the MOST effective in countering these malicious attacks?

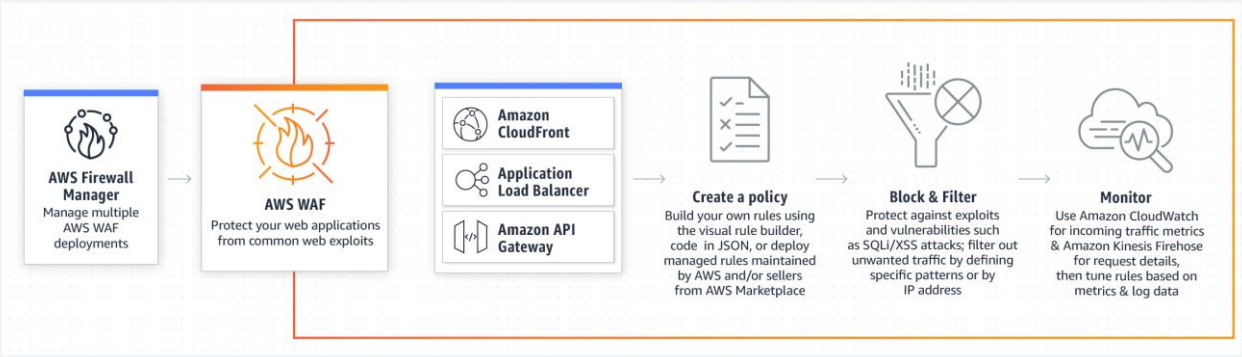
- Use Security Hub with CloudFront distribution
- Use Route 53 with CloudFront distribution
- Use Web Application Firewall (WAF) with CloudFront distribution **(Correct)**
- Use AWS Firewall Manager with CloudFront distribution

Explanation
Correct option:

Use Web Application Firewall (WAF) with CloudFront distribution

AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits that may affect availability, compromise security, or consume excessive resources. AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that block common attack patterns, such as SQL injection or cross-site scripting, and rules that filter out specific traffic patterns you define.

How WAF Works:



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

A web access control list (web ACL) gives you fine-grained control over the web requests that your Amazon CloudFront distribution, Amazon API Gateway API, or Application Load Balancer responds to.

When you create a web ACL, you can specify one or more CloudFront distributions that you want AWS WAF to inspect. AWS WAF starts to allow, block, or count web requests for those distributions based on the conditions that you identify in the web ACL. Therefore, combining WAF with CloudFront can prevent SQL injection and cross-site scripting attacks. So this is the correct option.

Incorrect options:

Use Route 53 with CloudFront distribution - Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. You cannot use Route 53 to prevent SQL injection and cross-site scripting attacks. So this option is incorrect.

Use Security Hub with CloudFront distribution - AWS Security Hub gives you a comprehensive view of your high-priority security alerts and security posture across your AWS accounts. With Security Hub, you have a single place that aggregates, organizes, and prioritizes your security alerts, or findings, from multiple AWS services, such as Amazon GuardDuty, Amazon Inspector, Amazon Macie, AWS Identity and Access Management (IAM) Access Analyzer, and AWS Firewall Manager, as well as from AWS Partner solutions. You cannot use Security Hub to prevent SQL injection and cross-site scripting attacks. So this option is incorrect.

Use AWS Firewall Manager with CloudFront distribution - AWS Firewall Manager is a security management service that allows you to centrally configure and manage firewall rules across your accounts and applications in AWS Organization. You cannot use Firewall Manager to prevent SQL injection and cross-site scripting attacks. So this option is incorrect.

Question 43:
Your company is evolving towards a microservice approach for their website. The company plans to expose the website from the same load balancer, linked to different target groups with different URLs, that are similar to these - checkout.mycorp.com, www.mycorp.com, mycorp.com/profile, and mycorp.com/search.

As a Solutions Architect, which Load Balancer type do you recommend to achieve this routing feature with MINIMUM configuration and development effort?

- Create a Network Load Balancer
- Create a Classic Load Balancer
- Create an NGINX based load balancer on an EC2 instance to have advanced routing capabilities
- Create an Application Load Balancer **(Correct)**

Explanation
Correct option:

Create an Application Load Balancer

Application Load Balancer can automatically distribute incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.

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. You can use host conditions to define rules that route requests based on the hostname in the host header (also known as host-based routing). This enables you to support multiple domains using a single load balancer. Example hostnames: example.com test.example.com *.example.com The rule *.example.com matches test.example.com but doesn't match example.com.

Path-based Routing: You can route a client request based on the URL path of the HTTP header. You can use path conditions to define rules that route requests based on the URL in the request (also known as path-based routing). 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. The path pattern is applied only to the path of the URL, not to its query parameters.

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 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 and host based routing are only available for the Application Load Balancer (ALB). Therefore this is the correct option for the given use-case.

Incorrect options:

Create an NGINX based load balancer on an EC2 instance to have advanced routing capabilities - Although it is technically possible to set up NGINX based load balancer, however, this option involves a lot of configuration effort, so this option is ruled out for the given use-case. So, deploying an NGINX load balancer on EC2 would work but would suffer management and scaling issues.

Create a Network Load Balancer - Network Load Balancer is best suited for use-cases involving low latency and high throughput workloads that involve scaling to millions of requests per second. Network Load Balancer operates at the connection level (Layer 4), routing connections to targets - Amazon EC2 instances, microservices, and containers – within Amazon Virtual Private Cloud (Amazon VPC) based on IP protocol data.

Create a Classic Load Balancer - Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network.

As mentioned in the description above, these two options are incorrect for the given use-case.

Question 44:
A developer in your team has set up a classic 3 tier architecture composed of an Application Load Balancer, an Auto Scaling group managing a fleet of EC2 instances, and an Aurora database. As a Solutions Architect, you would like to adhere to the security pillar of the well-architected framework.

How do you configure the security group of the Aurora database to only allow traffic coming from the EC2 instances?

- Add a rule authorizing the Aurora security group
- Add a rule authorizing the EC2 security group **(Correct)**
- Add a rule authorizing the ELB security group
- Add a rule authorizing the ASG's subnets CIDR

Explanation
Correct option:

Add a rule authorizing the EC2 security group

A security group acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you can specify one or more security groups; otherwise, we use the default security group. You can add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.

The following are the characteristics of security group rules:

By default, security groups allow all outbound traffic.

Security group rules are always permissive; you can't create rules that deny access.

Security groups are stateful.

For the given scenario, the EC2 instances that are part of the ASG are the ones accessing the database layer. The correct response is to add a rule to the security group attached to Aurora authorizing the EC2 instance's security group.

Incorrect options:

Add a rule authorizing the Aurora security group - Adding a rule, authorizing the Aurora security group, is just a distractor. Since it has no bearing on traffic allowed from the EC2 instances.

Add a rule authorizing the ASG's subnets CIDR - Authorizing the entire CIDR of the ASG's subnets is overkill and would allow non-ASG instances, access Aurora if they were part of the same CIDR.

Add a rule authorizing the ELB security group - Adding a rule authorizing the ELB security group would dilute the security for the Aurora databases because only the EC2 instances that are part of the ASG are the ones accessing the database layer. Therefore, it is not the correct option.

Question 45:
A company hosts a Network File System on its on-premises data center but it is now looking to adopt a hybrid cloud strategy to connect the on-premise applications to an AWS NFS that is backed by Amazon S3.

Which service do you recommend?

- Amazon Elastic File System (Amazon EFS)
-

Tape Gateway

- File Gateway **(Correct)**
- Volume Gateway

Explanation
Correct option:

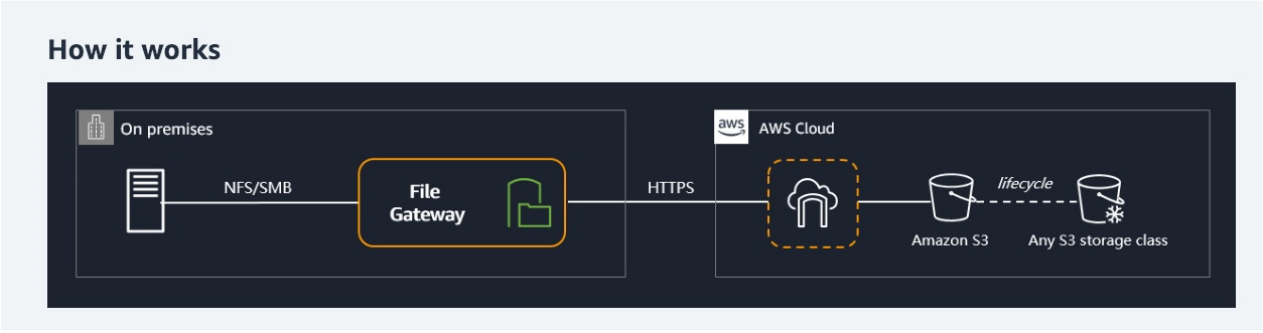
AWS Storage Gateway is a hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage. The service provides three different types of gateways – Tape Gateway, File Gateway, and Volume Gateway – that seamlessly connect on-premises applications to cloud storage, caching data locally for low-latency access. Your applications connect to the service through a virtual machine or hardware gateway appliance using standard storage protocols, such as NFS, SMB, and iSCSI.

File Gateway - File Gateway provides a seamless way to connect to the cloud in order to store application data files and backup images as durable objects in Amazon S3 cloud storage. File Gateway offers SMB or NFS-based access to data in Amazon S3 with local caching. It can be used for on-premises applications, and for Amazon EC2-based applications that need file protocol access to S3 object storage.

File Gateway - How it works:

Nearly all enterprises, regardless of industry, have to store files, whether they are backups, media content, or files generated by specialized industry applications. Managing and scaling on-premises infrastructure to provide online storage and distribution of such backup or content files is often burdensome and costly, requiring expensive hardware refreshes, data center expansion, and software licensing. These large file data repositories can be siloed in specialized file servers, NAS units, or backup systems, limiting access for big data analytics or media processing applications.

File Gateway provides a seamless way to connect to the cloud in order to store application data files and backup images as durable objects in Amazon S3 cloud storage. File Gateway offers SMB or NFS-based access to data in Amazon S3 with local caching. It can be used for on-premises applications, and for Amazon EC2-based applications that need file protocol access to S3 object storage.



via - https://aws.amazon.com/storagegateway/file/?nc=sn&loc=2&dn=2

Incorrect options:

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. The Volume Gateway provides either a local cache or full volumes on-premises while also storing full copies of your volumes in the AWS cloud. Volume Gateway also provides Amazon EBS Snapshots of your data for backup, disaster recovery, and migration. It's easy to get started with the Volume Gateway: Deploy it as a

virtual machine or hardware appliance, give it local disk resources, connect it to your applications, and start using your hybrid cloud storage for block data.

Tape Gateway - Tape Gateway enables you to replace using physical tapes on-premises with virtual tapes in AWS without changing existing backup workflows. Tape Gateway supports all leading backup applications and caches virtual tapes on-premises for low-latency data access. Tape Gateway encrypts data between the gateway and AWS for secure data transfer and compresses data and transitions virtual tapes between Amazon S3 and Amazon S3 Glacier, or Amazon S3 Glacier Deep Archive, to minimize storage costs.

Amazon Elastic File System (Amazon EFS) - Amazon EFS is a file storage service for use with Amazon EC2. Amazon EFS provides a file system interface, file system access semantics (such as strong consistency and file locking), and concurrently-accessible storage for up to thousands of Amazon EC2 instances. Amazon S3 is an object storage service. These are different storage systems and EFS is not backed by S3, so this is not the right fit for the given use-case. EFS, however, can be used for on-premises applications.

Question 46:
Your company is building a music sharing platform on which users can upload the songs of their choice. As a solutions architect for the platform, you have designed an architecture that will leverage a Network Load Balancer linked to an Auto Scaling Group across multiple availability zones. You are currently running with 100 Amazon EC2 instances with an Auto Scaling Group that needs to be able to share the storage layer for the music files.

Which technology do you recommend?

- EBS volumes mounted in RAID 1
- Instance Store
- EBS volumes mounted in RAID 0
- Amazon Elastic File System (Amazon EFS) **(Correct)**

Explanation
Correct option:

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. It is built to scale on-demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files, eliminating the need to provision and manage capacity to accommodate growth.

Here, we need a network file system (NFS), which is exactly what EFS is designed for. So, EFS is the correct option.

Incorrect options:

EBS volumes mounted in RAID 1

EBS volumes mounted in RAID 0

Amazon Elastic Block Store (EBS) is an easy to use, high-performance block storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction-intensive workloads at any scale. With Amazon EBS, you can use any of the standard RAID configurations that you can use with a traditional bare metal server, as long as that particular RAID configuration is supported by the operating system for your instance. This is because all RAID is accomplished at the software level. For greater I/O performance than you can achieve with a single volume, RAID 0 can stripe multiple volumes together; for on-instance redundancy, RAID 1 can mirror two volumes together.

EBS volumes (irrespective of the RAID types) are local disks and cannot be shared across instances (io1 or io2 type EBS volumes can be shared on Nitro EC2 instances but even this configuration only supports up to 16 instances).

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 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 stores are local disks and cannot be shared across instances.

Question 47:
A cyber security company is running a mission critical application using a single Spread placement group of EC2 instances. The company needs 15 Amazon EC2 instances for optimal performance.

How many Availability Zones (AZs) will the company need to deploy these EC2 instances per the given use-case?

- 3 **(Correct)**
- 15
- 7
- 14

Explanation
Correct option:

3

When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures. You can use placement groups to influence the placement of a group of interdependent instances to meet the needs of your workload. Depending on the type of workload, you can create a placement group using one of the following placement strategies:

Cluster placement group

Partition placement group

Spread placement group.

A Spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source.

Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other. Launching instances in a spread placement group reduces the risk of simultaneous failures that might occur when instances share the same racks.

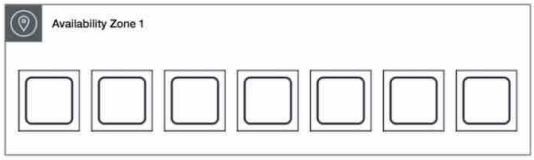
A spread placement group can span multiple Availability Zones in the same Region. You can have a maximum of seven running instances per Availability Zone per group. Therefore, to deploy 15 EC2 instances in a single Spread placement group, the company needs to use 3 AZs.

Spread placement group
overview:

Spread placement groups

A spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source.

The following image shows seven instances in a single Availability Zone that are placed into a spread placement group. The seven instances are placed on seven different racks.



Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other. Launching instances in a spread placement group reduces the risk of simultaneous failures that might occur when instances share the same racks. Spread placement groups provide access to distinct racks, and are therefore suitable for mixing instance types or launching instances over time.

A spread placement group can span multiple Availability Zones in the same Region. You can have a maximum of seven running instances per Availability Zone per group.

If you start or launch an instance in a spread placement group and there is insufficient unique hardware to fulfill the request, the request fails. Amazon EC2 makes more distinct hardware available over time, so you can try your request again later.

via - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html

Incorrect options:

7

14

15

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

Question 48:
A solutions architect has been tasked to design a low-latency solution for a static, single-page application, accessed by users through a custom domain name. The solution must be serverless, provide in-transit data encryption and needs to be cost-effective.

Which AWS services can be combined to build the simplest possible solution for the company's requirement?

- Host the application on AWS Fargate and front it with an Elastic Load Balancer for an improved performance
- Configure Amazon S3 to store the static data and use AWS Fargate for hosting the application
- Use Amazon S3 to host the static website and Amazon CloudFront to distribute the content for low latency access **(Correct)**
- Host the application on Amazon EC2 instance with instance store volume for high performance and low latency access to users

Explanation
Correct option:

Use Amazon S3 to host the static website and Amazon CloudFront to distribute the content for low latency access

To host a static website on Amazon S3, you configure an Amazon S3 bucket for website hosting and then upload your website content to the bucket. When you configure a bucket as a static website, you must enable website hosting, set permissions, and create and add an index document. Depending on your website requirements, you can also configure redirects, web traffic logging, and a custom error document.

After you configure your bucket as a static website, you can access the bucket through the AWS Region-specific Amazon S3 website endpoints for your bucket. Website endpoints are different from the endpoints where you send REST API requests. Amazon S3 doesn't support HTTPS access for website endpoints. If you want to use HTTPS, you can use CloudFront to serve a static website hosted on Amazon S3.

You can use Amazon CloudFront to improve the performance of your website. CloudFront makes your website files (such as HTML, images, and video) available from data centers around the world (called edge locations). When a visitor requests a file from your website, CloudFront automatically redirects the request to a copy of the file at the nearest edge location. This results in faster download times than if the visitor had requested the content from a data center that is located farther away.

CloudFront caches content at edge locations for a period of time that you specify. If a visitor requests content that has been cached for longer than the expiration date, CloudFront checks the origin server to see if a newer version of the content is available. If a newer version is available, CloudFront copies the new version to the edge location. Changes that you make to the original content are replicated to edge locations as visitors request the content.

Incorrect options:

Host the application on Amazon EC2 instance with instance store volume for high performance and low latency access to users - Since the use case speaks about a serverless solution, Amazon EC2 cannot be the answer, since EC2 is not serverless.

Host the application on AWS Fargate and front it with an Elastic Load Balancer for an improved performance - AWS Fargate is a serverless compute engine for containers that works with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS). Elastic Load Balancer can spread the incoming requests across a fleet of EC2 instances. This added complexity is not needed since we are looking at a static single-page webpage.

Configure Amazon S3 to store the static data and use AWS Fargate for hosting the application - Fargate is overkill for hosting a static single-page webpage.

Question 49:
A medical devices company uses S3 buckets to store critical data. Hundreds of buckets are used to keep the data segregated and well organized. Recently, the development team noticed that the lifecycle policies on the S3 buckets have not been applied optimally, resulting in higher costs.

As a Solutions Architect, can you recommend a solution to reduce storage costs on S3 while keeping the IT team's involvement to a minimum?

- Configure Amazon EFS to provide a fast, cost-effective and sharable storage service
- Use S3 One Zone-Infrequent Access, to reduce the costs on S3 storage
- Use S3 Outposts storage class to reduce the costs on S3 storage by storing the data on-premises
- Use S3 Intelligent-Tiering storage class to optimize the S3 storage costs **(Correct)**

Explanation
Correct option:

Use S3 Intelligent-Tiering storage class to optimize the S3 storage costs - 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. It works by storing objects in two access tiers: one tier that is optimized for frequent access and another lower-cost tier that is optimized for infrequent access.

For a small monthly monitoring and automation fee per object, Amazon S3 monitors access patterns of the objects in S3 Intelligent-Tiering and moves the ones that have not been accessed for 30 consecutive days to the infrequent access tier. If an object in the infrequent access tier is accessed, it is automatically moved back to the frequent access tier. There are no retrieval fees when using the S3 Intelligent-Tiering storage class, and no additional tiering fees when objects are moved between access tiers. It is the ideal storage class for long-lived data with access patterns that are unknown or unpredictable.

S3 Storage Classes can be configured at the object level and a single bucket can contain objects stored in S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, and S3 One Zone-IA. You can upload objects directly to S3 Intelligent-Tiering, or use S3 Lifecycle policies to transfer objects from S3 Standard and S3 Standard-IA to S3 Intelligent-Tiering. You can also archive objects from S3 Intelligent-Tiering to S3 Glacier.

Incorrect options:

Configure Amazon Elastic File System (Amazon EFS) to provide a fast, cost-effective and sharable storage service - 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. EFS offers sharable service, unlike Amazon Elastic Block Storage (EBS) that cannot be shared by instances. EFS is costlier than storing data in Amazon S3. Also, EFS needs an Amazon EC2 instance or an AWS Direct Connect network connection. Hence, this is not the correct option.

Use S3 One Zone-Infrequent Access, to reduce the costs on S3 storage - S3 One Zone-IA is for data that is accessed less frequently but requires rapid access when needed. Unlike other S3 Storage Classes which store data in a minimum of three Availability Zones (AZs), S3 One Zone-IA stores data in a single AZ and costs 20% less than S3 Standard-IA. S3 One Zone-IA is ideal for customers who want a lower-cost option for infrequently accessed data but do not require the availability and resilience of S3 Standard or S3 Standard-IA. Not a right option, since data stored is business-critical and cannot be risked by using S3 One Zone-IA.

Use S3 Outposts storage class to reduce the costs on S3 storage by storing the data on-premises - This is a distractor as Amazon S3 Outposts delivers object storage to your on-premises AWS Outposts environment. It is used in conjunction with AWS Outposts and has no relevance to the current use case.

Question 50:
A company needs a massive PostgreSQL database and the engineering team would like to retain control over managing the patches, version upgrades for the database, and

consistent performance with high IOPS. The team wants to install the database on an EC2 instance with the optimal storage type on the attached EBS volume.

As a solutions architect, which of the following configurations would you suggest to the engineering team?

- Amazon EC2 with EBS volume of General Purpose SSD (gp2) type
- Amazon EC2 with EBS volume of Provisioned IOPS SSD (io1) type **(Correct)**
- Amazon EC2 with EBS volume of cold HDD (sc1) type
- Amazon EC2 with EBS volume of Throughput Optimized HDD (st1) type

Explanation
Correct option:

Amazon EC2 with EBS volume of Provisioned IOPS SSD (io1) type

Amazon EBS provides the following volume types, which differ in performance characteristics and price so that you can tailor your storage performance and cost to the needs of your applications.

The volumes types fall into two categories:

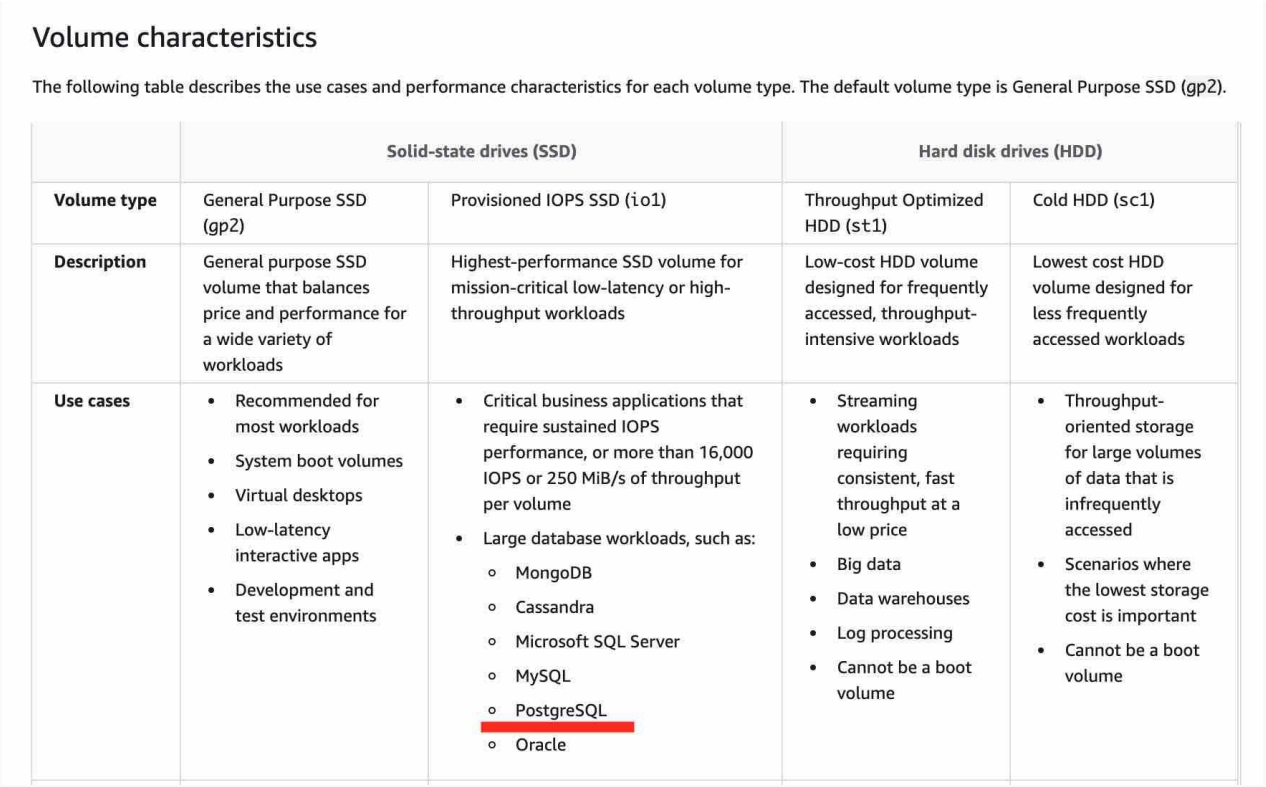
SSD-backed volumes optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS

HDD-backed volumes optimized for large streaming workloads where throughput (measured in MiB/s) is a better performance measure than IOPS

Provision IOPS type supports critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume. Examples are large database workloads, such as: MongoDB Cassandra Microsoft SQL Server MySQL PostgreSQL Oracle

Therefore, Amazon EC2 with EBS volume of Provisioned IOPS SSD (io1) type is the right fit for the given use-case.

Please see this detailed overview of the volume types for EBS volumes.



Volume characteristics

The following table describes the use cases and performance characteristics for each volume type. The default volume type is General Purpose SSD (gp2).

	Solid-state drives (SSD)		Hard disk drives (HDD)	
Volume type	General Purpose SSD (gp2)	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use cases	• Recommended for most workloads • System boot volumes • Virtual desktops • Low-latency interactive apps • Development and test environments	• Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume • Large database workloads, such as: ◦ MongoDB ◦ Cassandra ◦ Microsoft SQL Server ◦ MySQL ◦ PostgreSQL ◦ Oracle	• Streaming workloads requiring consistent, fast throughput at a low price • Big data • Data warehouses • Log processing • Cannot be a boot volume	• Throughput-oriented storage for large volumes of data that is infrequently accessed • Scenarios where the lowest storage cost is important • Cannot be a boot volume

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

Incorrect options:

Amazon EC2 with EBS volume of General Purpose SSD (gp2) type

Amazon EC2 with EBS volume of Throughput Optimized HDD (st1) type

Amazon EC2 with EBS volume of cold HDD (sc1) type

Per the explanation in the detailed overview provided above, these three options are incorrect.

Question 51:
The Development team at an e-commerce company is working on securing their databases.

Which of the following AWS database engines can be configured with IAM Database Authentication? (Select two)

- ☐ RDS Sequel Server
- ☐ RDS Maria DB
- ☐ RDS Oracle
- ☐ RDS PostGreSQL **(Correct)**

- RDS MySQL
  (Correct)

Explanation
Correct options:

You can authenticate to your DB instance using AWS Identity and Access Management (IAM) database authentication. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token. An authentication token is a unique string of characters that Amazon RDS generates on request. Each token has a lifetime of 15 minutes. You don't need to store user credentials in the database, because authentication is managed externally using IAM.

RDS MySQL - IAM database authentication works with MySQL and PostgreSQL.

RDS PostGreSQL - IAM database authentication works with MySQL and PostgreSQL.

Incorrect options:

RDS Oracle

RDS Maria DB

RDS Sequel Server

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

Question 52:
The data engineering team at an e-commerce company has set up a workflow to ingest the clickstream data into the raw zone of the S3 data lake. The team wants to run some SQL based data sanity checks on the raw zone of the data lake.

What AWS services would you recommend for this use-case such that the solution is cost-effective and easy to maintain?

- Load the incremental raw zone data into RDS on an hourly basis and run the SQL based sanity checks
- Load the incremental raw zone data into Redshift on an hourly basis and run the SQL based sanity checks
- Use Athena to run SQL based analytics against S3 data
  (Correct)
- Load the incremental raw zone data into an EMR based Spark Cluster on an hourly basis and use SparkSQL to run the SQL based sanity checks

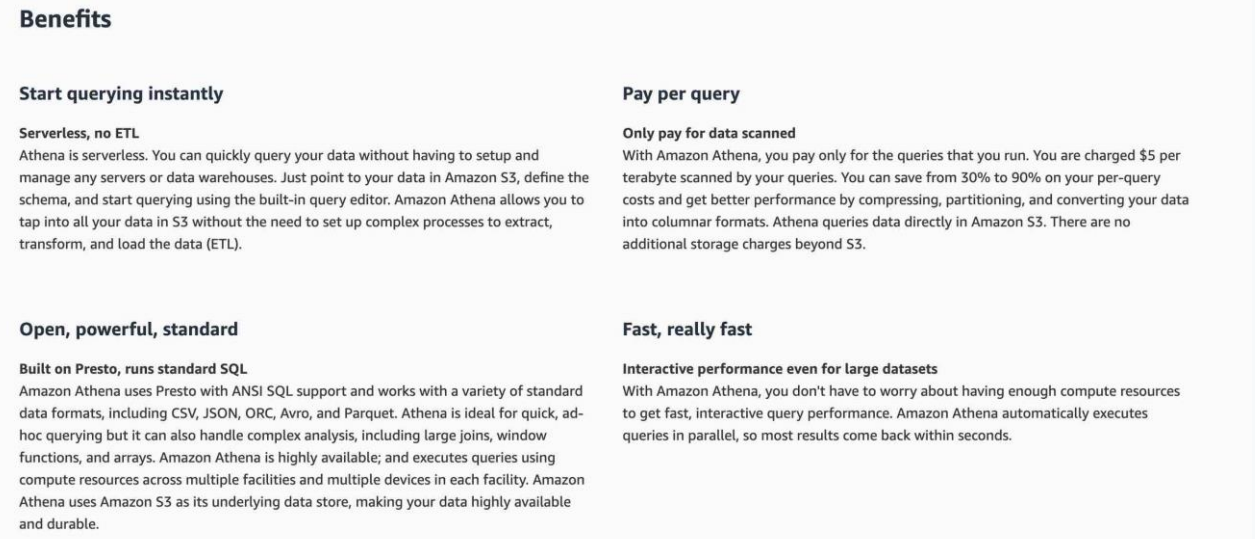
Explanation

Correct option:

Use Athena to run SQL based analytics against S3 data

Amazon Athena is an interactive query service that makes it easy to analyze data directly in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to set up or manage, and customers pay only for the queries they run. You can use Athena to process logs, perform ad-hoc analysis, and run interactive queries.

AWS Athena
Benefits:



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

Incorrect options:

Load the incremental raw zone data into Redshift on an hourly basis and run the SQL based sanity checks - Amazon Redshift is a fully-managed petabyte-scale cloud-based data warehouse product designed for large scale data set storage and analysis. As the development team would have to maintain and monitor the Redshift cluster size and would require significant development time to set up the processes to consume the data periodically, so this option is ruled out.

Load the incremental raw zone data into an EMR based Spark Cluster on an hourly basis and use SparkSQL to run the SQL based sanity checks - Amazon EMR is the industry-leading cloud big data platform for processing vast amounts of data using open source tools such as Apache Spark, Apache Hive, Apache HBase, Apache Flink, Apache Hudi, and Presto. Amazon EMR uses Hadoop, an open-source framework, to distribute your data and processing across a resizable cluster of Amazon EC2 instances. Using an EMR cluster would imply managing the underlying infrastructure so it's ruled out because the correct solution for the given use-case should require the least amount of development effort and ongoing maintenance.

Load the incremental raw zone data into RDS on an hourly basis and run the SQL based sanity checks - Loading the incremental data into RDS implies data migration jobs will have to be written via a Lambda function or an EC2 based process. This goes against the requirement that the solution should involve the least amount of development effort and ongoing maintenance. Hence this option is not correct.

Question 53:
A financial services company is moving its IT infrastructure to AWS Cloud and wants to enforce adequate data protection mechanisms on Amazon S3 to meet compliance guidelines. The engineering team has hired you as a solutions architect to build a solution for this requirement.

Can you help the team identify the INCORRECT option from the choices below?

- S3 can encrypt object metadata by using Server-Side Encryption **(Correct)**
- S3 can encrypt data in transit using HTTPS (TLS)
- S3 can protect data at rest using Client-Side Encryption
- S3 can protect data at rest using Server-Side Encryption

Explanation
Correct option:

S3 can encrypt object metadata by using Server-Side Encryption

Amazon S3 is a simple key-value store designed to store as many objects as you want. You store these objects in one or more buckets, and each object can be up to 5 TB in size.

An object consists of the following:

Key – The name that you assign to an object. You use the object key to retrieve the object.

Version ID – Within a bucket, a key and version ID uniquely identify an object.

Value – The content that you are storing.

Metadata – A set of name-value pairs with which you can store information regarding the object.

Subresources – Amazon S3 uses the subresource mechanism to store object-specific additional information.

Access Control Information – You can control access to the objects you store in Amazon S3.

Metadata, which can be included with the object, is not encrypted while being stored on Amazon S3. Therefore, AWS recommends that customers not place sensitive information in Amazon S3 metadata.

Incorrect options:

S3 can protect data at rest using Server-Side Encryption - This is possible and AWS provides three different ways of doing this - Server-side encryption with Amazon S3-managed keys (SSE-S3), Server-side encryption with customer master keys stored in AWS Key Management Service (SSE-KMS), Server-side encryption with customer-provided keys (SSE-C).

S3 can protect data at rest using Client-Side Encryption - This is a possible scenario too. You can encrypt data on the client-side and upload the encrypted data to Amazon S3. In this case, the client manages the encryption process, the encryption keys, and related tools.

S3 can encrypt data in transit using HTTPS (TLS) - This is also possible and you can use HTTPS (TLS) to help prevent potential attackers from eavesdropping on or manipulating network traffic using person-in-the-middle or similar attacks.

Question 54:
A pharmaceutical company is considering moving to AWS Cloud to accelerate the research and development process. Most of the daily workflows would be centered around running batch jobs on EC2 instances with storage on EBS volumes. The CTO is concerned about meeting HIPAA compliance norms for sensitive data stored on EBS.

Which of the following options outline the correct capabilities of an encrypted EBS volume? (Select three)

- ☐ Data moving between the volume and the instance is NOT encrypted
- ☐ Data at rest inside the volume is encrypted **(Correct)**
- ☐ Any snapshot created from the volume is NOT encrypted
- ☐ Any snapshot created from the volume is encrypted **(Correct)**
- ☐ Data moving between the volume and the instance is encrypted **(Correct)**
- ☐ Data at rest inside the volume is NOT encrypted

Explanation
Correct options:

Data at rest inside the volume is encrypted

Any snapshot created from the volume is encrypted

Data moving between the volume and the instance is encrypted

Amazon Elastic Block Store (Amazon EBS) provides block-level storage volumes for use with EC2 instances. When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, data moving between the volume and the instance, snapshots created from the volume and volumes created from those snapshots are all encrypted. It uses AWS Key Management Service (AWS KMS) customer master keys (CMK) when creating encrypted volumes and snapshots. Encryption operations occur on the servers that host EC2 instances, ensuring the security of both data-at-rest and data-in-transit between an instance and its attached EBS storage.

Therefore, the incorrect options are:

Data moving between the volume and the instance is NOT encrypted

Any snapshot created from the volume is NOT encrypted

Data at rest inside the volume is NOT encrypted

Question 55:
The data science team at a mobility company wants to analyze real-time location data of rides. The company is using Kinesis Data Firehose for delivering the location-specific streaming data into targets for downstream analytics.

Which of the following targets are NOT supported by Kinesis Data Firehose?

- Amazon Elasticsearch
- Amazon EMR **(Correct)**
- Amazon RedShift
- Amazon S3

Explanation
Correct option:

Amazon EMR

You can use Amazon Kinesis Data Firehose to load streaming data into data lakes, data stores, and analytics tools. It can capture, transform, and load streaming data into Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk.

How Kinesis Data Firehose
works

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

Amazon EMR is the industry-leading cloud big data platform for processing vast amounts of data using open source tools such as Apache Spark, Apache Hive, Apache HBase, Apache Flink, Apache Hudi, and Presto. Amazon EMR uses Hadoop, an open-source framework, to distribute your data and processing across a resizable cluster of Amazon EC2 instances. Firehose does not support Amazon EMR as a target for delivering the streaming data.

Incorrect options:

Amazon S3 - Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Your applications can easily achieve thousands of transactions per second in request performance when uploading and retrieving storage from Amazon S3.

Amazon RedShift - Amazon Redshift is a fully-managed petabyte-scale cloud-based data warehouse product designed for large scale data set storage and analysis.

Amazon Elasticsearch - Amazon Elasticsearch Service is a fully managed service that makes it easy for you to deploy, secure, and run Elasticsearch cost-effectively at scale. Elasticsearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.

As mentioned, Firehose can deliver streaming data into Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk.

Question 56:
An automobile company is running its flagship application on a fleet of EC2 instances behind an Auto Scaling Group (ASG). The ASG has been configured more than a year ago. A young developer has just joined the development team and wants to understand the best practices to manage and configure an ASG.

As a Solutions Architect, which of these would you identify as the key characteristics that the developer needs to understand regarding ASG configurations? (Select three)

- [ ]

Amazon EC2 Auto Scaling can automatically add a volume when the existing one is approaching capacity. This, however, is a configuration parameter and needs to be set explicitly

- If you configure the ASG to a certain base capacity, you cannot use a combined purchasing model to fulfill the instance requirements. You will need to choose either On-Demand instances or Reserved Instances only
- EC2 Auto Scaling groups can span Availability Zones, but not AWS regions **(Correct)**
- If you have an EC2 Auto Scaling group (ASG) with running instances and you choose to delete the ASG, the instances will be terminated and the ASG will be deleted **(Correct)**
- You can only specify one launch configuration for an EC2 Auto Scaling group at a time. But, you can modify a launch configuration after you've created it
- Data is not automatically copied from existing instances to a new dynamically created instance **(Correct)**

Explanation
Correct options:

Amazon EC2 Auto Scaling is a fully managed service designed to launch or terminate Amazon EC2 instances automatically to help ensure you have the correct number of Amazon EC2 instances available to handle the load for your application.

If you have an EC2 Auto Scaling group (ASG) with running instances and you choose to delete the ASG, the instances will be terminated and the ASG will be deleted This statement is correct.

EC2 Auto Scaling groups can span Availability Zones, but not AWS regions - EC2 Auto Scaling groups are regional constructs. They can span Availability Zones, but not AWS regions.

Data is not automatically copied from existing instances to a new dynamically created instance - Data is not automatically copied from existing instances to new instances. You can use lifecycle hooks to copy the data.

Incorrect options:

If you configure the ASG to a certain base capacity, you cannot use a combined purchasing model to fulfill the instance requirements. You will need to choose either On-Demand instances or Reserved Instances only - When setting up an ASG with a combined purchasing model, you can specify the base capacity of the group to be fulfilled by On-Demand instances. As the ASG scales in or scales out, EC2 Auto Scaling ensures the base capacity is fulfilled using On-Demand instances and anything beyond

that be fulfilled with either only Spot instances or a specified percentage mix of On-Demand or Spot instances.

Amazon EC2 Auto Scaling can automatically add a volume when the existing one is approaching capacity. This, however, is a configuration parameter and needs to be set explicitly - Amazon EC2 Auto Scaling doesn't automatically add a volume when the existing one is approaching capacity. You can use the EC2 API to add a volume to an existing instance.

You can only specify one launch configuration for an EC2 Auto Scaling group at a time. But, you can modify a launch configuration after you've created it - You can only specify one launch configuration for an EC2 Auto Scaling group at a time, and you can't modify a launch configuration after you've created it.

Question 57:
Reporters at a news agency upload/download video files (about 500MB each) to/from an S3 bucket as part of their daily work. As the agency has started offices in remote locations, it has resulted in poor latency for uploading and accessing data to/from S3. The agency wants to continue using S3 but wants to improve the performance.

As a solutions architect, which of the following solutions do you propose to address this issue? (Select two)

- ☐ Enable Amazon S3 Transfer Acceleration for the S3 bucket. This would speed up uploads as well as downloads for the video files **(Correct)**
- ☐ Create new S3 buckets in every region where the agency has a remote office, so that each office can maintain its storage for the media assets
- ☐ Use Amazon CloudFront distribution with origin as the S3 bucket. This would speed up uploads as well as downloads for the video files **(Correct)**
- ☐ Spin up EC2 instances in each region where the agency has a remote office. Create a daily job to transfer S3 data into EBS volumes attached to the EC2 instances
- ☐ Move S3 data into EFS file system created in a US region, connect to EFS file system from EC2 instances in other AWS regions using an inter-region VPC peering connection

Explanation
Correct options:

Use Amazon CloudFront distribution with origin as the S3 bucket. This would speed up uploads as well as downloads for the video files

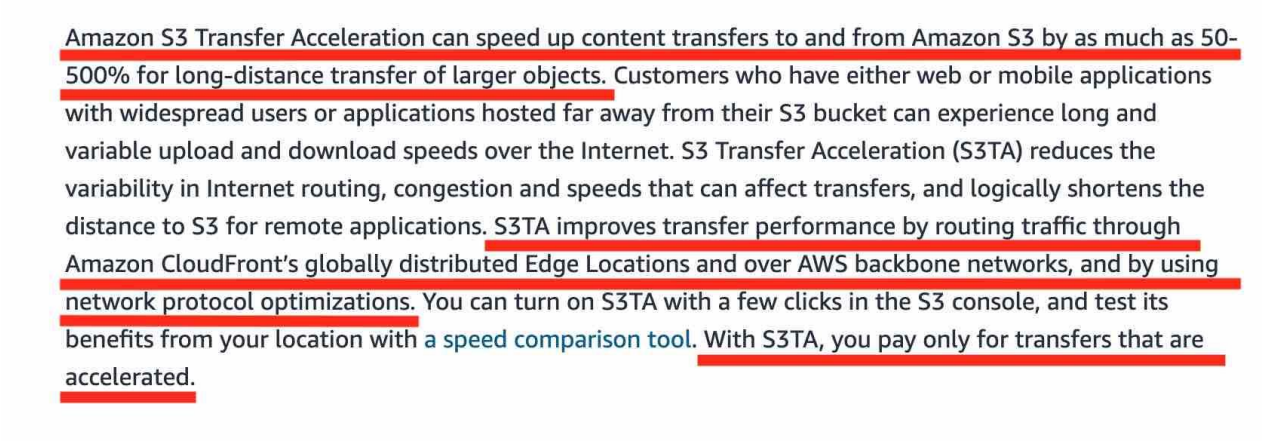
Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high

transfer speeds, within a developer-friendly environment. When an object from S3 that is set up with CloudFront CDN is requested, the request would come through the Edge Location transfer paths only for the first request. Thereafter, it would be served from the nearest edge location to the users until it expires. So in this way, you can speed up uploads as well as downloads for the video files.

Enable Amazon S3 Transfer Acceleration for the S3 bucket. This would speed up uploads as well as downloads for the video files

Amazon S3 Transfer Acceleration can speed up content transfers to and from Amazon S3 by as much as 50-500% for long-distance transfer of larger objects. 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. So this option is also correct.

Transfer Acceleration
Overview:


Amazon S3 Transfer Acceleration can speed up content transfers to and from Amazon S3 by as much as 50-500% for long-distance transfer of larger objects. Customers who have either web or mobile applications with widespread users or applications hosted far away from their S3 bucket can experience long and variable upload and download speeds over the Internet. S3 Transfer Acceleration (S3TA) reduces the variability in Internet routing, congestion and speeds that can affect transfers, and logically shortens the distance to S3 for remote applications. S3TA improves transfer performance by routing traffic through Amazon CloudFront's globally distributed Edge Locations and over AWS backbone networks, and by using network protocol optimizations. You can turn on S3TA with a few clicks in the S3 console, and test its benefits from your location with a speed comparison tool. With S3TA, you pay only for transfers that are accelerated.

via - https://aws.amazon.com/s3/transfer-acceleration/

Incorrect options:

Create new S3 buckets in every region where the agency has a remote office, so that each office can maintain its storage for the media assets - Creating new S3 buckets in every region is not an option, since the agency maintains centralized storage. Hence this option is incorrect.

Move S3 data into EFS file system created in a US region, connect to EFS file system from EC2 instances in other AWS regions using an inter-region VPC peering connection

Spin up EC2 instances in each region where the agency has a remote office. Create a daily job to transfer S3 data into EBS volumes attached to the EC2 instances

Both these options using EC2 instances are not correct for the given use-case, as the agency wants a serverless storage solution.

Question 58:
While troubleshooting, a cloud architect realized that the Amazon EC2 instance is unable to connect to the internet using the Internet Gateway.

Which conditions should be met for internet connectivity to be established? (Select two)

- The instance's subnet is not associated with any route table
- The instance's subnet is associated with multiple route tables with conflicting configurations
- The network ACLs associated with the subnet must have rules to allow inbound and outbound traffic **(Correct)**
- The route table in the instance's subnet should have a route to an Internet Gateway **(Correct)**
- The subnet has been configured to be public and has no access to the internet

Explanation
Correct options:

The network ACLs associated with the subnet must have rules to allow inbound and outbound traffic - The network access control lists (ACLs) that are associated with the subnet must have rules to allow inbound and outbound traffic on port 80 (for HTTP traffic) and port 443 (for HTTPs traffic). This is a necessary condition for Internet Gateway connectivity

The route table in the instance's subnet should have a route to an Internet Gateway - A route table contains a set of rules, called routes, that are used to determine where network traffic from your subnet or gateway is directed. The route table in the instance's subnet should have a route defined to the Internet Gateway.

Incorrect options:

The instance's subnet is not associated with any route table - This is an incorrect statement. A subnet is implicitly associated with the main route table if it is not explicitly associated with a particular route table. So, a subnet is always associated with some route table.

The instance's subnet is associated with multiple route tables with conflicting configurations - This is an incorrect statement. A subnet can only be associated with one route table at a time.

The subnet has been configured to be public and has no access to internet - This is an incorrect statement. Public subnets have access to the internet via Internet Gateway.

Question 59:
A company has noticed that its application performance has deteriorated after a new Auto Scaling group was deployed a few days back. Upon investigation, the team found out that the Launch Configuration selected for the Auto Scaling group is using the incorrect instance type that is not optimized to handle the application workflow.

As a solutions architect, what would you recommend to provide a long term resolution for this issue?

- No need to modify the launch configuration. Just modify the Auto Scaling group to use the correct instance type
- Create a new launch configuration to use the correct instance type. Modify the Auto Scaling group to use this new launch configuration. Delete the old launch configuration as it is no longer needed **(Correct)**
- No need to modify the launch configuration. Just modify the Auto Scaling group to use more number of existing instance types. More instances may offset the loss of performance
- Modify the launch configuration to use the correct instance type and continue to use the existing Auto Scaling group

Explanation
Correct option:

Create a new launch configuration to use the correct instance type. Modify the Auto Scaling group to use this new launch configuration. Delete the old launch configuration as it is no longer needed

A launch configuration is an instance configuration template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances. Include the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping.

It is not possible to modify a launch configuration once it is created. The correct option is to create a new launch configuration to use the correct instance type. Then modify the Auto Scaling group to use this new launch configuration. Lastly to clean-up, just delete the old launch configuration as it is no longer needed.

Incorrect options:

Modify the launch configuration to use the correct instance type and continue to use the existing Auto Scaling group - As mentioned earlier, it is not possible to modify a launch configuration once it is created. Hence, this option is incorrect.

No need to modify the launch configuration. Just modify the Auto Scaling group to use the correct instance type - You cannot use an Auto Scaling group to directly modify the instance type of the underlying instances. Hence, this option is incorrect.

No need to modify the launch configuration. Just modify the Auto Scaling group to use more number of existing instance types. More instances may offset the loss of performance - Using the Auto Scaling group to increase the number of instances to cover up for the performance loss is not recommended as it does not address the root cause of the problem. The Machine Learning workflow requires a certain instance type

that is optimized to handle Machine Learning computations. Hence, this option is incorrect.

Question 60:
An online gaming company wants to block access to its application from specific countries; however, the company wants to allow its remote development team (from one of the blocked countries) to have access to the application. The application is deployed on EC2 instances running under an Application Load Balancer (ALB) with AWS WAF.

As a solutions architect, which of the following solutions can be combined to address the given use-case? (Select two)

- [ ] Use WAF geo match statement listing the countries that you want to block **(Correct)**
- [ ] Use ALB IP set statement that specifies the IP addresses that you want to allow through
- [ ] Use ALB geo match statement listing the countries that you want to block
- [ ] Create a deny rule for the blocked countries in the NACL associated with each of the EC2 instances
- [ ] Use WAF IP set statement that specifies the IP addresses that you want to allow through **(Correct)**

Explanation
Correct options:

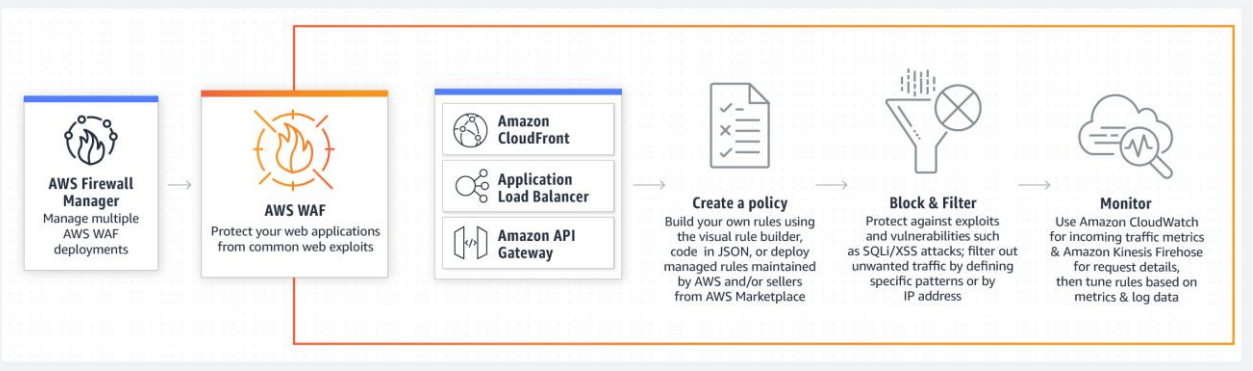
Use WAF geo match statement listing the countries that you want to block

Use WAF IP set statement that specifies the IP addresses that you want to allow through

AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits that may affect availability, compromise security, or consume excessive resources. AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that block common attack patterns and rules that filter out specific traffic patterns you define.

You can deploy AWS WAF on Amazon CloudFront as part of your CDN solution, the Application Load Balancer that fronts your web servers or origin servers running on EC2, or Amazon API Gateway for your APIs.

AWS WAF - How it
Works



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

To block specific countries, you can create a WAF geo match statement listing the countries that you want to block, and to allow traffic from IPs of the remote development team, you can create a WAF IP set statement that specifies the IP addresses that you want to allow through. You can combine the two rules as shown below:

Geographic match rule statement

PDF | Kindle | RSS

To allow or block web requests based on country of origin, create one or more geographical, or geo, match statements.

Note

If you use the CloudFront geo restriction feature to block a country from accessing your content, any request from that country is blocked and is not forwarded to AWS WAF. So if you want to allow or block requests based on geography plus other AWS WAF criteria, you should *not* use the CloudFront geo restriction feature. Instead, you should use an AWS WAF geo match condition.

You can use this to block access to your site from specific countries or to only allow access from specific countries. If you want to allow some web requests and block others based on country of origin, add a geo match statement for the countries that you want to allow and add a second one for the countries that you want to block.

You can use geo match statements with other AWS WAF statements to build sophisticated filtering. For example, to block certain countries, but still allow requests from a specific set of IP addresses in that country, you could create a rule with the action set to `Block` and the following nested statements:

- AND statement
  - Geo match statement listing the countries that you want to block
  - NOT statement
    - IP set statement that specifies the IP addresses that you want to allow through

As another example, if you want to prioritize resources for users in a particular country, you could create a different rate-based rules statement for each geo match condition. Set a higher rate limit for users in the preferred country and set a lower rate limit for all other users.

via - https://docs.aws.amazon.com/waf/latest/developerguide/waf-rule-statement-type-geo-match.html

Incorrect options:

Create a deny rule for the blocked countries in the NACL associated to each of the EC2 instances - A network access control list (ACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. NACL does not have the capability to block traffic based on geographic match conditions.

Use ALB geo match statement listing the countries that you want to block

Use ALB IP set statement that specifies the IP addresses that you want to allow through

An Application Load Balancer (ALB) operates at the request level (layer 7), routing traffic to targets – EC2 instances, containers, IP addresses, and Lambda functions based on the content of the request. Ideal for advanced load balancing of HTTP and HTTPS traffic, Application Load Balancer provides advanced request routing targeted at delivery of modern application architectures, including microservices and container-based applications.

An ALB cannot block or allow traffic based on geographic match conditions or IP based conditions. Both these options have been added as distractors.

Question 61:
A streaming solutions company is building a video streaming product by using an Application Load Balancer (ALB) that routes the requests to the underlying EC2 instances. The engineering team has noticed a peculiar pattern. The ALB removes an instance whenever it is detected as unhealthy but the Auto Scaling group fails to kick-in and provision the replacement instance.

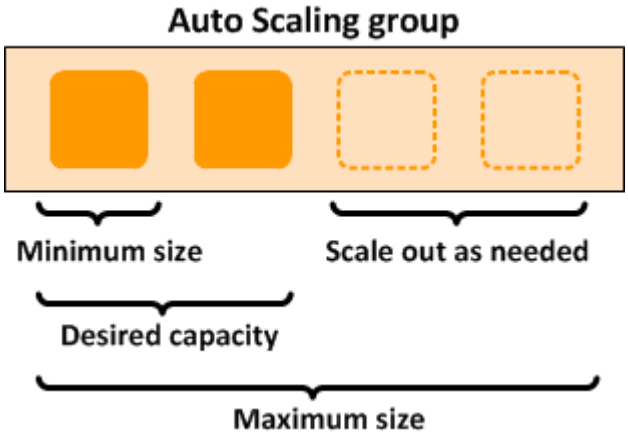
What could explain this anomaly?

- Both the Auto Scaling group and Application Load Balancer are using ALB based health check
- The Auto Scaling group is using ALB based health check and the Application Load Balancer is using EC2 based health check
- Both the Auto Scaling group and Application Load Balancer are using EC2 based health check
- The Auto Scaling group is using EC2 based health check and the Application Load Balancer is using ALB based health check **(Correct)**

Explanation
Correct option:

The Auto Scaling group is using EC2 based health check and the Application Load Balancer is using ALB based health check

An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for automatic scaling and management.



Auto Scaling Group Overview: via - https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html

Application Load Balancer automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.

If the Auto Scaling group (ASG) is using EC2 as the health check type and the Application Load Balancer (ALB) is using its in-built health check, there may be a situation where the ALB health check fails because the health check pings fail to receive a response from the instance. At the same time, ASG health check can come back as successful because it is based on EC2 based health check. Therefore, in this scenario, the ALB will remove the instance from its inventory, however, the ASG will fail to provide the replacement instance. This can lead to the scaling issues mentioned in the problem statement.

Incorrect options:

The Auto Scaling group is using ALB based health check and the Application Load Balancer is using EC2 based health check - ALB cannot use EC2 based health checks, so this option is incorrect.

Both the Auto Scaling group and Application Load Balancer are using ALB based health check - It is recommended to use ALB based health checks for both Auto Scaling group and Application Load Balancer. If both the Auto Scaling group and Application Load Balancer use ALB based health checks, then you will be able to avoid the scenario mentioned in the question.

Both the Auto Scaling group and Application Load Balancer are using EC2 based health check - ALB cannot use EC2 based health checks, so this option is incorrect.

Question 62:
An application hosted on Amazon EC2 contains sensitive personal information about all its customers and needs to be protected from all types of cyber-attacks. The company is considering using the AWS Web Application Firewall (WAF) to handle this requirement.

Can you identify the correct solution leveraging the capabilities of WAF?

- AWS WAF can be directly configured on Amazon EC2 instances for ensuring the security of the underlying application data
- AWS WAF can be directly configured only on an ALB or an Amazon API Gateway. One of these two services can then be configured with Amazon EC2 to build the needed secure architecture
- Configure an ALB to balance the workload for all the EC2 instances. Configure CloudFront to distribute from an ALB since WAF cannot be directly configured on ALBs. This configuration not only provides necessary safety but is scalable too
- Create a CloudFront distribution for the application on Amazon EC2 instances. Deploy AWS WAF on Amazon CloudFront to provide the necessary safety measures **(Correct)**

Explanation
Correct option:

Create a CloudFront distribution for the application on Amazon EC2 instances. Deploy AWS WAF on Amazon CloudFront to provide the necessary safety measures

When you use AWS WAF with CloudFront, you can protect your applications running on any HTTP webserver, whether it's a webserver that's running in Amazon Elastic Compute Cloud (Amazon EC2) or a web server that you manage privately. You can also configure CloudFront to require HTTPS between CloudFront and your own webserver, as well as between viewers and CloudFront.

AWS WAF is tightly integrated with Amazon CloudFront and the Application Load Balancer (ALB), services that AWS customers commonly use to deliver content for their websites and applications. When you use AWS WAF on Amazon CloudFront, your rules run in all AWS Edge Locations, located around the world close to your end-users. This means security doesn't come at the expense of performance. Blocked requests are stopped before they reach your web servers. When you use AWS WAF on Application Load Balancer, your rules run in the region and can be used to protect internet-facing as well as internal load balancers.

Incorrect options:

Configure an Application Load Balancer (ALB) to balance the workload for all the EC2 instances. Configure CloudFront to distribute from an ALB since WAF cannot be directly configured on ALBs. This configuration not only provides necessary safety but is scalable too - This statement is wrong. You can configure WAF on Application Load Balancers (ALB).

*AWS WAF can be directly configured on Amazon EC2 instances for ensuring the security of the underlying application data** - AWS WAF can be deployed on Amazon CloudFront, the Application Load Balancer (ALB), and Amazon API Gateway. It cannot be configured directly on an EC2 instance.

AWS WAF can be directly configured only on an Application Load Balancer (ALB) or an Amazon API Gateway. One of these two services can then be configured with Amazon EC2 to build the needed secure architecture - This statement is only partially correct. WAF can also be deployed on Amazon CloudFront service.

Question 63:
A DevOps engineer at an organization is debugging issues related to an Amazon EC2 instance. The engineer has SSH'ed into the instance and he needs to retrieve the instance public IP from within a shell script running on the instance command line.

Can you identify the correct URL path to get the instance public IP?

- http://169.254.169.254/latest/user-data/public-ipv4
- http://254.169.254.169/latest/meta-data/public-ipv4
- http://169.254.169.254/latest/meta-data/public-ipv4 **(Correct)**
- http://254.169.254.169/latest/user-data/public-ipv4

Explanation
Correct option:

http://169.254.169.254/latest/meta-data/public-ipv4

Instance metadata is the data about your instance that you can use to configure or manage the running instance.

Instance user data is the data that you specified in the form of a configuration script while launching your instance.

The following URL paths can be used to get the instance meta data and user data from within the instance: http://169.254.169.254/latest/meta-data/

http://169.254.169.254/latest/user-data/

Further, you can get the instance public IP via the URL - http://169.254.169.254/latest/meta-data/public-ipv4

Incorrect options:

http://169.254.169.254/latest/user-data/public-ipv4

http://254.169.254.169/latest/meta-data/public-ipv4

http://254.169.254.169/latest/user-data/public-ipv4

These three options do not meet the specification for the URL path to get the instance public IP, so these are incorrect.

Question 64:
A financial services company has to retain the activity logs for each of their customers to meet compliance guidelines. Depending on the business line, the company wants to retain the logs for 5-10 years in highly available and durable storage on AWS. The overall data size is expected to be in PetaBytes. In case of an audit, the data would need to be accessible within a timeframe of up to 48 hours.

Which AWS storage option is the MOST cost-effective for the given compliance requirements?

- Third party tape storage
- Amazon S3 Glacier
- Amazon S3 Standard storage
- Amazon S3 Glacier Deep Archive **(Correct)**

Explanation
Correct option:

Amazon S3 Glacier Deep Archive

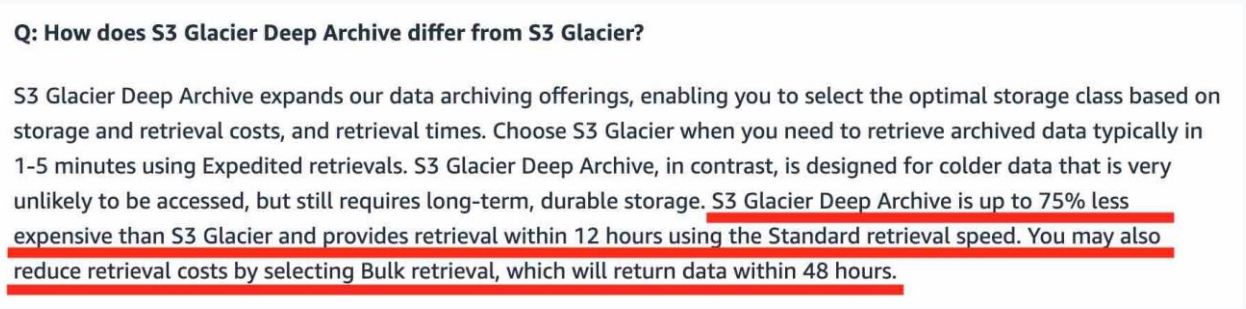
Amazon S3 Glacier and S3 Glacier Deep Archive are secure, durable, and extremely low-cost Amazon S3 cloud storage classes for data archiving and long-term backup. They are designed to deliver 99.999999999% durability, and provide comprehensive security and compliance capabilities that can help meet even the most stringent regulatory requirements.

S3 Glacier Deep Archive is a new Amazon S3 storage class that provides secure and durable object storage for long-term retention of data that is accessed once or twice in a year. From just $0.00099 per GB-month (less than one-tenth of one cent, or about $1 per TB-month), S3 Glacier Deep Archive offers the lowest cost storage in the cloud, at prices significantly lower than storing and maintaining data in on-premises magnetic tape libraries or archiving data off-site.

S3 Glacier Deep Archive is up to 75% less expensive than S3 Glacier and provides retrieval within 12 hours using the Standard retrieval speed. You may also reduce retrieval costs by selecting Bulk retrieval, which will return data within 48 hours.

Therefore, Amazon S3 Glacier Deep Archive is the correct choice.

S3 Glacier vs S3 Glacier Deep Archive:



Q: How does S3 Glacier Deep Archive differ from S3 Glacier?

S3 Glacier Deep Archive expands our data archiving offerings, enabling you to select the optimal storage class based on storage and retrieval costs, and retrieval times. Choose S3 Glacier when you need to retrieve archived data typically in 1-5 minutes using Expedited retrievals. S3 Glacier Deep Archive, in contrast, is designed for colder data that is very unlikely to be accessed, but still requires long-term, durable storage. S3 Glacier Deep Archive is up to 75% less expensive than S3 Glacier and provides retrieval within 12 hours using the Standard retrieval speed. You may also reduce retrieval costs by selecting Bulk retrieval, which will return data within 48 hours.

via - https://aws.amazon.com/s3/faqs/

Incorrect options:

Amazon S3 Glacier - As mentioned earlier, S3 Glacier Deep Archive is up to 75% less expensive than S3 Glacier and provides retrieval within 12 hours. So using Amazon S3 Glacier is not the correct choice.

Third-party tape storage

Amazon S3 Standard storage

Given the relaxed retrieval times, S3 standard storage would be much costlier than the S3 Glacier Deep Archive, so S3 standard storage is not the correct option. Using Third-party tape storage is ruled out as the company wants to use an AWS storage service. Therefore, both of these options are incorrect.

Question 65:
You have just terminated an instance in the us-west-1a availability zone. The attached EBS volume is now available for attachment to other instances. An intern launches a new Linux EC2 instance in the us-west-1b availability zone and is attempting to attach the EBS volume. The intern informs you that it is not possible and needs your help.

Which of the following explanations would you provide to them?

- The required IAM permissions are missing
- EBS volumes are AZ locked **(Correct)**
- The EBS volume is encrypted
- EBS volumes are region locked

Explanation
Correct option:

EBS volumes are AZ locked

An Amazon EBS volume is a durable, block-level storage device that you can attach to your instances. After you attach a volume to an instance, you can use it as you would use a physical hard drive. EBS volumes are flexible. For current-generation volumes attached to current-generation instance types, you can dynamically increase size, modify the provisioned IOPS capacity, and change volume type on live production volumes.

When you create an EBS volume, it is automatically replicated within its Availability Zone to prevent data loss due to the failure of any single hardware component. You can attach an EBS volume to an EC2 instance in the same Availability Zone.

Incorrect options:

EBS volumes are region locked - It's confined to an Availability Zone and not by region.

The required IAM permissions are missing - This is a possibility as well but if permissions are not an issue then you are still confined to an availability zone.

The EBS volume is encrypted - This doesn't affect the ability to attach an EBS volume.