Notes Practice Test – 6

Question: The content division at a digital media agency has an application that generates a large number of files on Amazon S3, each approximately 10 megabytes in size. The agency mandates that the files be stored for 5 years before they can be deleted. The files are frequently accessed in the first 30 days of the object creation but are rarely accessed after the first 30 days. The files contain critical business data that is not easy to reproduce, therefore, immediate accessibility is always required. Which solution is the MOST cost-effective for the given use case?

Correct option:

Set up an Amazon S3 bucket lifecycle policy to move files from Amazon S3 Standard to Amazon S3 Standard-IA 30 days after object creation. Delete the files 5 years after object creation.

Explanation behind correct option:

1. Amazon S3 Standard-IA class is for data that is accessed less frequently but requires rapid access when needed.
2. Amazon S3 Standard-IA offers high durability, high throughput, and low latency similar to S3 Standard, with a lower per gigabyte storage price and per GB retrieval charge.
3. You can set up an Amazon S3 lifecycle configuration to transition objects from S3 Standard to S3 Standard-IA 30 days after object creation.
4. Set an expiration action to delete the object 5 years after object creation.

Incorrect options:

1. **Set up an Amazon S3 bucket lifecycle policy to move files from Amazon S3 Standard to Amazon S3 Glacier Flexible Retrieval 30 days after object creation. Delete the files 5 years after object creation.**
   - Amazon S3 Glacier Flexible Retrieval storage class has a retrieval time of minutes, which is not suitable for the requirement of immediate accessibility.
2. **Set up an Amazon S3 bucket lifecycle policy to move files from Amazon S3 Standard to Amazon S3 One Zone-IA 30 days after object creation. Delete the files 5 years after object creation.**
   - Amazon S3 One Zone-IA stores data in a single Availability Zone (AZ) and is less durable compared to other S3 storage classes, making it unsuitable for critical business data that is not easy to reproduce.
3. **Set up an Amazon S3 bucket lifecycle policy to move files from Amazon S3 Standard to Amazon S3 Standard-IA 30 days after object creation. Archive the files to Amazon S3 Glacier Deep Archive 5 years after object creation.**
   - Archiving files to Amazon S3 Glacier Deep Archive is unnecessary and incurs additional costs. The files can simply be deleted after 5 years.

Conclusion / Points to memorize:

1. Amazon S3 Standard-IA is ideal for data that is accessed less frequently but requires quick access when needed.
2. Transition objects from S3 Standard to S3 Standard-IA after 30 days for cost savings.
3. Delete objects after the required retention period to avoid unnecessary storage costs.
4. S3 Glacier classes are suitable for long-term archival where retrieval times can be in minutes to hours, not for immediate access.
5. S3 One Zone-IA is not suitable for critical data as it is stored in a single AZ.

Question 2 - (correct)

Question: Reporters at a news agency upload/download video files (about 500 megabytes each) to/from an Amazon 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 the given Amazon S3 bucket. The agency wants to continue using a serverless storage solution such as Amazon 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)

Correct option/s:

- Enable Amazon S3 Transfer Acceleration (Amazon S3TA) for the Amazon S3 bucket. This would speed up uploads as well as downloads for the video files.
- Use Amazon CloudFront distribution with origin as the Amazon S3 bucket. This would speed up uploads as well as downloads for the video files.

Explanation behind correct option/s:

1. **Enable Amazon S3 Transfer Acceleration (Amazon S3TA) for the Amazon 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.
   - It improves transfer performance by routing traffic through Amazon CloudFront's globally distributed Edge Locations and AWS backbone networks using network protocol optimizations.
   - You pay only for transfers that are accelerated.
2. **Use Amazon CloudFront distribution with origin as the Amazon S3 bucket. This would speed up uploads as well as downloads for the video files**

- Amazon CloudFront is a fast content delivery network (CDN) that securely delivers data, videos, applications, and APIs to customers globally with low latency and high transfer speeds.
- When an object from Amazon S3 set up with Amazon CloudFront is requested, the request is served from the nearest edge location to the users until it expires, thus speeding up uploads and downloads.

Incorrect options:

1. **Move Amazon S3 data into Amazon Elastic File System (Amazon EFS) created in a US region, connect to Amazon EFS file system from Amazon EC2 instances in other AWS regions using an inter-region VPC peering connection**
   - This option does not address the requirement for a serverless storage solution.
2. **Spin up Amazon EC2 instances in each region where the agency has a remote office. Create a daily job to transfer Amazon S3 data into Amazon EBS volumes attached to the Amazon EC2 instances**
   - Using Amazon EC2 instances is not a serverless solution and adds unnecessary complexity and cost.
3. **Create new Amazon 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 feasible as the agency maintains centralized storage.

Conclusion / Points to memorize:

1. **Amazon S3 Transfer Acceleration** is effective for speeding up long-distance transfers of large objects.
2. **Amazon CloudFront** distribution can improve upload and download speeds by serving requests from the nearest edge location.
3. Serverless storage solutions like Amazon S3 with optimizations such as S3TA and CloudFront are preferable for reducing latency and improving performance.
4. Avoid using EC2 instances for a serverless storage solution.
5. Centralized storage should not be replaced with regional S3 buckets unless decentralization is required.

Question 3 - (correct)

Question: A development team has deployed a microservice to the Amazon Elastic Container Service (Amazon 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 Amazon 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?

Correct option/s:

- Distribute the static content through Amazon S3

Explanation behind correct option/s:

1. **Distribute the static content through Amazon S3**
   - Amazon S3 is ideal for hosting static websites. On a static website, individual web pages include static content, which can also contain client-side scripts.
   - To host a static website on Amazon S3, configure an S3 bucket for website hosting, upload your website content, enable website hosting, set permissions, and create and add an index document.
   - This setup offloads the majority of the network usage to Amazon S3, freeing up resources on the Amazon ECS instances.

Incorrect options:

1. **Distribute the dynamic content through Amazon S3**
   - Dynamic websites rely on server-side processing, including server-side scripts such as PHP, JSP, or ASP.NET.
   - Amazon S3 does not support server-side scripting, making it unsuitable for hosting dynamic content.
2. **Distribute the static content through Amazon EFS**
   - Amazon Elastic File System (Amazon EFS) provides a scalable, fully managed NFS file system for use with AWS Cloud services and on-premises resources.
   - Using Amazon EFS for static content will not change the network load as the static content on EFS would still need to be distributed by the Amazon ECS instances.
3. **Distribute the dynamic content through Amazon EFS**
   - Similar to the previous point, using Amazon EFS for dynamic content will not reduce the network usage load on the Amazon ECS instances as the distribution still relies on the ECS cluster.

Conclusion / Points to memorize:

1. **Amazon S3** is ideal for distributing static content, reducing network load on ECS instances.
2. **Static websites** hosted on Amazon S3 offload the network usage from application servers.

3. **Amazon EFS** is not suitable for offloading static or dynamic content distribution in this context.
4. **Dynamic content** requires server-side processing, which Amazon S3 does not support.

Question 4 - (correct)

Question: 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 Amazon EC2 instances with storage on Amazon Elastic Block Store (Amazon EBS) volumes. The CTO is concerned about meeting HIPAA compliance norms for sensitive data stored on Amazon EBS. Which of the following options outline the correct capabilities of an encrypted Amazon EBS volume? (Select three)

Correct option/s:

- Data at rest inside the volume is encrypted
- Data moving between the volume and the instance is encrypted
- Any snapshot created from the volume is encrypted

Explanation behind correct option/s:

1. **Data at rest inside the volume is encrypted**
   - When you create an encrypted Amazon EBS volume, the data stored at rest on the volume is encrypted using AWS Key Management Service (AWS KMS) customer master keys (CMK).
2. **Data moving between the volume and the instance is encrypted**
   - Data moving between the Amazon EBS volume and the attached Amazon EC2 instance is encrypted, ensuring the security of data-in-transit.
3. **Any snapshot created from the volume is encrypted**
   - Snapshots created from an encrypted Amazon EBS volume are also encrypted. Similarly, any volumes created from these snapshots will be encrypted.

Incorrect options:

1. **Any snapshot created from the volume is NOT encrypted**
   - This is incorrect because any snapshot created from an encrypted Amazon EBS volume is encrypted.
2. **Data at rest inside the volume is NOT encrypted**
   - This is incorrect because data at rest inside an encrypted Amazon EBS volume is encrypted.
3. **Data moving between the volume and the instance is NOT encrypted**
   - This is incorrect because data moving between an encrypted Amazon EBS volume and the attached instance is encrypted.

Conclusion / Points to memorize:

1. **Amazon EBS encryption** ensures that data at rest, data in transit between the volume and instance, and snapshots are all encrypted.
2. **AWS KMS** customer master keys are used for encrypting Amazon EBS volumes and snapshots.
3. Ensuring data encryption helps meet **HIPAA compliance** norms for sensitive data.
4. Always verify that both data at rest and data in transit are encrypted when dealing with sensitive information.

Question 5 - (incorrect)

Question: 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?

Correct option/s:

- Amazon S3 Glacier Deep Archive

Explanation behind correct option/s:

1. **Amazon S3 Glacier Deep Archive**
   - Amazon S3 Glacier Deep Archive is designed for long-term retention of data that is accessed infrequently, such as once or twice a year.
   - It offers the lowest cost storage in the cloud, at prices significantly lower than on-premises magnetic tape libraries or off-site data archiving.
   - The storage class delivers 99.999999999% durability and provides comprehensive security and compliance capabilities.
   - Retrieval within 48 hours can be achieved using the Bulk retrieval option, while Standard retrieval provides access within 12 hours.
   - This makes it the most cost-effective solution for retaining petabytes of data with the ability to access it within the required timeframe for audits.

Incorrect options:

1. **Amazon S3 Standard storage**
   - Amazon S3 Standard storage is much costlier than Amazon S3 Glacier Deep Archive, given the relaxed retrieval times specified.
2. **Amazon S3 Glacier**
   - Amazon S3 Glacier is more expensive compared to Amazon S3 Glacier Deep Archive. Deep Archive provides up to 75% cost savings over Glacier, making it a less cost-effective option for long-term storage.
3. **Third party tape storage**
   - The company specifies the use of AWS storage services, ruling out third-party tape storage. Additionally, tape storage does not offer the same level of accessibility and durability as AWS storage services.

Conclusion / Points to memorize:

1. **Amazon S3 Glacier Deep Archive** is the most cost-effective storage solution for long-term retention of infrequently accessed data.
2. **Durability** and **security** of 99.999999999% ensure compliance with stringent regulatory requirements.
3. **Retrieval times** of up to 48 hours can be met using Bulk retrieval from Amazon S3 Glacier Deep Archive.
4. **Cost savings** of up to 75% over Amazon S3 Glacier make Deep Archive the preferred choice for petabyte-scale storage.
5. **Amazon S3 Standard storage** and **third-party tape storage** are not cost-effective or compliant with the specified AWS service requirement.

Question 6 - (correct)

Question: A financial services company is moving its IT infrastructure to AWS Cloud and wants to enforce adequate data protection mechanisms on Amazon Simple Storage Service (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?

Correct option/s:

- Amazon S3 can encrypt object metadata by using Server-Side Encryption

Explanation behind correct option/s:

1. **Amazon S3 can encrypt object metadata by using Server-Side Encryption**
   - Amazon S3 is a key-value store where each object consists of key, value, version ID, metadata, subresources, and access control information.
   - Metadata, which is a set of name-value pairs stored with the object, is **not** encrypted by Server-Side Encryption (SSE). Therefore, sensitive information should not be included in Amazon S3 metadata.
   - This makes it the incorrect option as object metadata is not encrypted using SSE.

Incorrect options:

1. **Amazon S3 can protect data at rest using Server-Side Encryption**
   - This is correct. AWS provides three ways to perform server-side encryption:
     - SSE-S3: Server-side encryption with Amazon S3-managed keys.
     - SSE-KMS: Server-side encryption with customer master keys stored in AWS Key Management Service (KMS).
     - SSE-C: Server-side encryption with customer-provided keys.
2. **Amazon S3 can protect data at rest using Client-Side Encryption**
   - This is correct. Client-side encryption involves encrypting data on the client side before uploading it to Amazon S3. The client manages the encryption process and the keys.
3. **Amazon S3 can encrypt data in transit using HTTPS (TLS)**
   - This is correct. Using HTTPS (TLS) helps prevent potential attackers from eavesdropping on or manipulating network traffic, ensuring data in transit is encrypted.

Conclusion / Points to memorize:

1. **Amazon S3 metadata** is not encrypted using Server-Side Encryption; avoid storing sensitive information in metadata.
2. **Server-Side Encryption (SSE)** has three variations: SSE-S3, SSE-KMS, and SSE-C.
3. **Client-Side Encryption** allows encryption of data before uploading to Amazon S3.
4. **Data in transit** can be encrypted using HTTPS (TLS) to ensure secure communication.
5. Always verify the encryption capabilities and limitations when designing secure solutions on AWS.

Question 7 - (correct)

Question: 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 Amazon EC2 instances running under an Application Load Balancer with AWS Web Application Firewall (AWS WAF). As a solutions architect, which of the following solutions can be combined to address the given use-case? (Select two)

Correct option/s:

- Use AWS WAF geo match statement listing the countries that you want to block
- Use AWS WAF IP set statement that specifies the IP addresses that you want to allow through

Explanation behind correct option/s:

1. **Use AWS WAF geo match statement listing the countries that you want to block**
   - AWS WAF allows you to create security rules that include geographic match conditions. You can block requests originating from specific countries by listing them in a geo match statement.
2. **Use AWS WAF IP set statement that specifies the IP addresses that you want to allow through**
   - AWS WAF provides IP set statements to define specific IP addresses that should be allowed. This allows you to grant access to the remote development team by specifying their IP addresses in the IP set statement.

Incorrect options:

1. **Create a deny rule for the blocked countries in the network access control list (network ACL) associated with each of the Amazon EC2 instances**
   - Network ACLs operate at the subnet level within a VPC and do not support geographic match conditions, making them unsuitable for blocking traffic based on country.
2. **Use Application Load Balancer geo match statement listing the countries that you want to block**
   - Application Load Balancers operate at the request level (layer 7) and do not support blocking traffic based on geographic match conditions.
3. **Use Application Load Balancer IP set statement that specifies the IP addresses that you want to allow through**
   - Similarly, Application Load Balancers do not support allowing traffic based on specific IP addresses.

Conclusion / Points to memorize:

1. **AWS WAF** can be used to block or allow traffic based on geographic location and specific IP addresses.
2. **Geo match statements** in AWS WAF allow you to block requests from specific countries.
3. **IP set statements** in AWS WAF allow you to specify IP addresses that should be allowed through, providing granular control over access.
4. **Network ACLs** and **Application Load Balancers** do not support geographic match conditions or IP-based conditions for this use-case.

Question 8 - (incorrect)

Question: The engineering team at a weather tracking company wants to enhance the performance of its relational database and is looking for a caching solution that supports geospatial data. As a solutions architect, which of the following solutions will you suggest?

Correct option/s:

- Use Amazon ElastiCache for Redis

Explanation behind correct option/s:

1. **Use Amazon ElastiCache for Redis**
   - Amazon ElastiCache for Redis is a fast, open-source, in-memory key-value data store that supports a rich set of features, including geospatial data operations.
   - Redis can deliver sub-millisecond response times, supporting millions of requests per second, making it suitable for real-time applications.
   - Redis has purpose-built commands for working with real-time geospatial data at scale, such as finding the distance between elements and locating all elements within a given distance from a point.

Incorrect options:

1. **Use Amazon ElastiCache for Memcached**
   - Memcached is a high-performance, distributed memory cache service designed for simplicity and does not support geospatial data.
   - While both Redis and Memcached are in-memory, open-source data stores, Memcached lacks the advanced data structures and features, such as geospatial support, provided by Redis.
2. **Use Amazon DynamoDB Accelerator (DAX)**
   - Amazon DynamoDB Accelerator (DAX) is an in-memory cache for DynamoDB, designed to accelerate read operations.
   - DAX does not support relational databases or geospatial data, making it unsuitable for this use case.
3. **Use AWS Global Accelerator**
   - AWS Global Accelerator is a networking service that improves the availability and performance of applications for global users.
   - This option is unrelated to database caching or geospatial data support and has been included as a distractor.

Conclusion / Points to memorize:

1. **Amazon ElastiCache for Redis** supports advanced data structures, including geospatial data operations, making it the ideal choice for real-time applications requiring geospatial support.
2. **Memcached** is suitable for simpler caching requirements but does not support geospatial data.

3. **DynamoDB Accelerator (DAX)** is designed for DynamoDB and does not support relational databases or geospatial data.
4. **AWS Global Accelerator** is a networking service, not a caching solution, and is irrelevant for this specific use case.

Question 9 - (incorrect)

Question: An application with global users across AWS Regions had suffered an issue when the Elastic Load Balancing (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?

Correct option/s:

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

Explanation behind correct option/s:

1. **Set up AWS Global Accelerator and add endpoints to cater to users in different geographic locations**
   - AWS Global Accelerator provides two static IPs that are anycast from globally distributed edge locations, giving a single entry point to the application, regardless of how many AWS Regions it's deployed in.
   - It allows automatic failover by redirecting traffic to a healthy endpoint within seconds if an application endpoint has a failure or availability issue.
   - AWS Global Accelerator helps reduce internet latency by using the AWS global network for routing traffic, which provides lower latency compared to the public internet.
   - It offers features like traffic dials to control the proportion of traffic directed to each endpoint, easy movement of endpoints between Availability Zones or Regions, and static IP addresses that do not require DNS updates.

Incorrect options:

1. **Set up AWS Direct Connect as the backbone for each of the AWS Regions where the application is deployed**
   - AWS Direct Connect is used to connect on-premises systems to AWS Cloud for extremely low latency use cases but cannot be used to serve users directly.
2. **Create Amazon S3 buckets in different AWS Regions and configure Amazon CloudFront to pick the nearest edge location to the user**
   - While CloudFront can improve performance for static content, it does not cover the failover needs for ELBs and EC2 instances. The architecture includes dynamic content and compute resources that need automatic failover.
3. **Set up an Amazon Route 53 geoproximity routing policy to route traffic**
   - Route 53 geoproximity routing directs traffic based on geographic location but managing and routing to different instances and ELBs would become complex and operationally intensive as resource count increases.
   - AWS Global Accelerator provides built-in features like static anycast IP addresses, fault tolerance, and performance-based routing, which are more suitable for multi-region, low latency use cases.

Conclusion / Points to memorize:

1. **AWS Global Accelerator** provides automatic failover and reduces latency by using the AWS global network and static anycast IP addresses.
2. **Direct Connect** is for connecting on-premises systems to AWS and not for direct user traffic.
3. **Amazon CloudFront** is suitable for static content delivery but not for dynamic content or compute resource failover.
4. **Route 53 geoproximity routing** can handle geographic routing but lacks the built-in automatic failover and simplicity of management provided by AWS Global Accelerator.

Question 10 - (correct)

Question: You have just terminated an instance in the us-west-1a Availability Zone (AZ). The attached Amazon EBS volume is now available for attachment to other instances. An intern launches a new Linux Amazon EC2 instance in the us-west-1b Availability Zone (AZ) and is attempting to attach the Amazon 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?

Correct option/s:

- Amazon EBS volumes are Availability Zone (AZ) locked

Explanation behind correct option/s:

1. **Amazon EBS volumes are Availability Zone (AZ) locked**
   - Amazon EBS volumes are designed to be durable, block-level storage devices that can be attached to Amazon EC2 instances within the same Availability Zone.
   - When an EBS volume is created, it is automatically replicated within its AZ to prevent data loss due to hardware failure.
   - This means that an EBS volume can only be attached to instances that are in the same AZ where the volume was created. The volume created in us-west-1a cannot be attached to an instance in us-west-1b.

Incorrect options:

1. **The required IAM permissions are missing**
   - While missing IAM permissions can prevent attaching an EBS volume, this is not the issue in this case. The problem is due to the volume being AZ locked.
2. **The Amazon EBS volume is encrypted**
   - Encryption does not affect the ability to attach an EBS volume to an instance. Encrypted volumes can be attached to instances as long as they are in the same AZ.
3. **Amazon EBS volumes are region locked**
   - EBS volumes are not locked by region but by Availability Zone. They can be used within the same region but must remain in their originating AZ.

Conclusion / Points to memorize:

1. **Amazon EBS volumes** are AZ-locked and can only be attached to instances within the same Availability Zone.
2. **Replication** within an AZ ensures durability and data protection against hardware failure.
3. **IAM permissions** and **encryption** do not impact the ability to attach volumes within the same AZ.
4. **Understanding AZ constraints** is crucial when managing and attaching EBS volumes to EC2 instances.

Question 11 - (correct)

Question: A silicon valley based healthcare startup uses AWS Cloud for its IT infrastructure. The startup stores patient health records on Amazon Simple Storage Service (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?

Correct option/s:

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

Explanation behind correct option/s:

1. **Use Amazon 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 cloud storage class for data archiving and long-term backup, providing 99.999999999% durability.
   - An Amazon S3 Glacier vault is a container for storing archives. Vaults can be configured with vault lock policies to enforce compliance controls.
   - Amazon S3 Glacier Vault Lock allows you to deploy and enforce compliance controls such as “write once read many” (WORM), ensuring that data is not altered after being written.
   - Once a vault lock policy is locked, it cannot be changed, thus meeting stringent regulatory requirements.

Incorrect options:

1. **Use Amazon S3 Glacier to store the sensitive archived data and then use an Amazon S3 lifecycle policy to enforce compliance controls**
   - Amazon S3 lifecycle policies can manage the transition and expiration of objects but cannot enforce compliance controls. Lifecycle policies are used to transition objects between storage classes or delete them after a specified period.
2. **Use Amazon S3 Glacier vault to store the sensitive archived data and then use an Amazon S3 Access Control List to enforce compliance controls**
   - Amazon S3 Access Control Lists (ACLs) manage access permissions to buckets and objects but are not designed to enforce compliance controls like “write once read many” (WORM).
3. **Use Amazon S3 Glacier to store the sensitive archived data and then use an Amazon S3 Access Control List to enforce compliance controls**
   - Similar to the previous option, ACLs are for managing access permissions and cannot enforce compliance controls required for regulatory needs.

Conclusion / Points to memorize:

1. **Amazon S3 Glacier Vault Lock** is essential for enforcing compliance controls such as “write once read many” (WORM).
2. **Lifecycle policies** are useful for managing the lifecycle of objects but not for compliance enforcement.
3. **Access Control Lists (ACLs)** are designed for managing permissions, not for enforcing compliance policies.
4. **Vault lock policies** provide a method to enforce regulatory and compliance controls by locking policies to prevent future edits.

Question 12 - (correct)

Question: 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?

Correct option/s:

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

Explanation behind correct option/s:

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

- Amazon S3 is ideal for hosting static websites. It allows you to configure a bucket for website hosting and upload your website content. You must enable website hosting, set permissions, and create an index document.
- Amazon S3 alone does not support HTTPS access for website endpoints. To ensure in-transit data encryption, you can use Amazon CloudFront.
- Amazon CloudFront is a global content delivery network (CDN) that caches your website files (such as HTML, images, and videos) at edge locations around the world, improving performance by serving content from the nearest edge location to the user.
- CloudFront also supports HTTPS, ensuring that data in transit is encrypted.

Incorrect options:

1. **Host the application on Amazon EC2 instance with instance store volume for high performance and low latency access to users**
   - Amazon EC2 is not a serverless solution and involves managing and maintaining instances, which increases complexity and cost. The use case requires a serverless and cost-effective solution.
2. **Host the application on AWS Fargate and front it with Elastic Load Balancing for an improved performance**
   - AWS Fargate is a serverless compute engine for containers but adds unnecessary complexity for a static single-page application. Elastic Load Balancing is also not needed for this simple use case.
3. **Configure Amazon S3 to store the static data and use AWS Fargate for hosting the application**
   - Using AWS Fargate to host a static single-page application is overkill. Amazon S3 with CloudFront is a simpler and more cost-effective solution.

Conclusion / Points to memorize:

1. **Amazon S3** is suitable for hosting static websites and provides an easy setup for website hosting.
2. **Amazon CloudFront** enhances performance by caching content at edge locations and supports HTTPS for in-transit data encryption.
3. **Serverless and cost-effective solutions** are preferred for simple use cases like static single-page applications.
4. **Avoid unnecessary complexity** by choosing the simplest combination of services that meet the requirements.

Question 13 - (correct)

Question: A financial services company runs its flagship web application on AWS. The application serves thousands of users during peak hours. The company needs a scalable near-real-time solution to share hundreds of thousands of financial transactions with multiple internal applications. The solution should also remove sensitive details from the transactions before storing the cleansed transactions in a document database for low-latency retrieval. As an AWS Certified Solutions Architect Associate, which of the following would you recommend?

Correct option/s:

- Feed the streaming transactions into Amazon Kinesis Data Streams. Leverage AWS Lambda integration to remove sensitive data from every transaction and then store the cleansed transactions in Amazon DynamoDB. The internal applications can consume the raw transactions off the Amazon Kinesis Data Stream

Explanation behind correct option/s:

1. **Feed the streaming transactions into Amazon Kinesis Data Streams. Leverage AWS Lambda integration to remove sensitive data from every transaction and then store the cleansed transactions in Amazon DynamoDB. The internal applications can consume the raw transactions off the Amazon Kinesis Data Stream**
   - Amazon Kinesis Data Streams allows you to build custom applications that process or analyze streaming data in real-time. It manages the necessary infrastructure, storage, and networking, and scales to accommodate your data throughput.
   - AWS Lambda can be used as a consumer to process data from Kinesis Data Streams. In this case, Lambda would remove sensitive data from each transaction before storing the cleansed data in Amazon DynamoDB.
   - This setup provides a near-real-time, scalable solution where internal applications can also consume the raw transactions from the Kinesis Data Stream.

Incorrect options:

1. **Batch process the raw transactions data into Amazon S3 flat files. Use S3 events to trigger an AWS Lambda function to remove sensitive data from the raw transactions in the flat file and then store the cleansed transactions in Amazon DynamoDB. Leverage DynamoDB Streams to share the transactions data with the internal applications**
   - The use case requires a near-real-time solution, and batch processing would not meet this requirement.
2. **Feed the streaming transactions into Amazon Kinesis Data Firehose. Leverage AWS Lambda integration to remove sensitive data from every transaction and then store the cleansed transactions in Amazon DynamoDB. The internal applications can consume the raw transactions off the Amazon Kinesis Data Firehose**
   - Amazon Kinesis Data Firehose is an ETL service for delivering streaming data to data lakes and analytics services. It does not support multiple consumers as required by the use case.

3. **Persist the raw transactions into Amazon DynamoDB. Configure a rule in Amazon DynamoDB to update the transaction by removing sensitive data whenever any new raw transaction is written. Leverage Amazon DynamoDB Streams to share the transactions data with the internal applications**
    - There is no such rule in DynamoDB to automatically update data upon writing. You would need to use DynamoDB triggers to invoke Lambda functions for cleansing, which introduces inefficiency as the same item would be written and then updated.

Conclusion / Points to memorize:

1. **Amazon Kinesis Data Streams** provides a scalable, real-time solution for processing streaming data.
2. **AWS Lambda** can be used to process and cleanse data in real-time before storing it in Amazon DynamoDB.
3. **Batch processing** is not suitable for near-real-time requirements.
4. **Kinesis Data Firehose** is for ETL and does not support multiple consumers.
5. **DynamoDB Streams** can trigger external processing, but adding rules for in-place data cleansing introduces inefficiencies.

Question 14 - (incorrect)

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

Correct option/s:

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

Explanation behind correct option/s:

1. **By default, cross-zone load balancing is enabled for Application Load Balancer and disabled for Network Load Balancer**
    - Cross-zone load balancing is enabled by default for Application Load Balancers, meaning each load balancer node distributes traffic across all registered targets in all enabled Availability Zones.
    - For Network Load Balancers, cross-zone load balancing is disabled by default. Each load balancer node distributes traffic only to the registered targets within its own Availability Zone unless cross-zone load balancing is manually enabled.

Incorrect options:

1. **By default, cross-zone load balancing is disabled for both Application Load Balancer and Network Load Balancer**
    - This is incorrect because cross-zone load balancing is enabled by default for Application Load Balancers.
2. **By default, cross-zone load balancing is enabled for both Application Load Balancer and Network Load Balancer**
    - This is incorrect because cross-zone load balancing is disabled by default for Network Load Balancers.
3. **By default, cross-zone load balancing is disabled for Application Load Balancer and enabled for Network Load Balancer**
    - This is incorrect because cross-zone load balancing is enabled by default for Application Load Balancers and disabled by default for Network Load Balancers.

Conclusion / Points to memorize:

1. **Application Load Balancer (ALB)**
    - Cross-zone load balancing is enabled by default.
    - Ensures traffic is distributed across all registered targets in all enabled Availability Zones.
2. **Network Load Balancer (NLB)**
    - Cross-zone load balancing is disabled by default.
    - Traffic is distributed only to the registered targets within the same Availability Zone unless manually enabled.
3. **Cross-zone load balancing** distributes traffic more evenly across targets in multiple Availability Zones, improving overall load distribution and resilience.

Question 15 - (correct)

Question: 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 Application Load Balancer in another Region in case of failure of the primary Application Load Balancer. As a Solutions Architect, what will you recommend to automate the failover process?

Correct option/s:

- Enable an Amazon Route 53 health check

Explanation behind correct option/s:

1. **Enable an Amazon Route 53 health check**
    - Amazon Route 53 DNS Failover integrates with ELB behind the scenes, managing health checks for individual ELB nodes.
    - Route 53 evaluates the health of both the load balancer and the application running on the EC2 instances behind it by combining results from ELB health checks and its own health checks.

- In case of a failure, Route 53 detects the issue and routes traffic away from the failed endpoint to a healthy one.
- This automation ensures that end-users are routed to the closest healthy region, minimizing downtime and improving reliability.

Incorrect options:

1. **Enable an ALB health check**
   - ALB health checks verify that a specified TCP port on an instance is accepting connections or that a specified page returns a status code of 200.
   - This option does not handle the failover process in the case of an ALB failure in the primary region.
2. **Enable an Amazon EC2 instance health check**
   - Instance status checks monitor the software and network configuration of your instance.
   - These checks do not verify if the application on the instance is working correctly and do not handle ALB-specific failover scenarios.
3. **Configure AWS Trusted Advisor to check on unhealthy instances**
   - AWS Trusted Advisor provides recommendations based on best practices and configuration checks.
   - It does not automate failover processes or handle the complexity of cross-region failover.

Conclusion / Points to memorize:

1. **Amazon Route 53 health checks** automate the failover process by monitoring the health of ELBs and the applications behind them.
2. **Route 53 DNS Failover** ensures minimal downtime by routing traffic to the nearest healthy endpoint in case of a failure.
3. **ALB health checks** and **EC2 instance health checks** do not provide the necessary automation and cross-region failover capabilities.
4. **AWS Trusted Advisor** provides best practice recommendations but does not handle automated failover processes.

Question 16 - (incorrect)

Question: A media company is evaluating the possibility of moving its IT infrastructure to the AWS Cloud. The company needs at least 10 terabytes 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 terabytes of very durable storage for storing media content and almost double of it, i.e. 900 terabytes for archival of legacy data. As a Solutions Architect, which set of services will you recommend to meet these requirements?

Correct option/s:

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

Explanation behind correct option/s:

1. **Amazon EC2 instance store for maximum performance**
   - Instance store provides temporary block-level storage for your instance, ideal for applications requiring high random I/O performance and very low latency.
   - Instance stores use NVMe or SATA-based SSDs, making them suitable for processing large files like videos.
   - Note: Instance store is ephemeral and data is lost when the instance is stopped or terminated, so it's ideal for temporary storage needs.
2. **Amazon S3 for durable data storage**
   - Amazon S3 Standard offers high durability (99.999999999%) and availability, making it suitable for storing frequently accessed media content.
   - S3 provides low latency and high throughput, appropriate for cloud applications, content distribution, and big data analytics.
3. **Amazon S3 Glacier for archival storage**
   - Amazon S3 Glacier is designed for secure, durable, and low-cost data archiving.
   - It provides different retrieval options, from minutes to hours, suitable for long-term data retention.
   - This option is ideal for the archival of 900 terabytes of legacy data.

Incorrect options:

1. **Amazon S3 standard storage for maximum performance, Amazon S3 Intelligent-Tiering for intelligent, durable storage, and Amazon S3 Glacier Deep Archive for archival storage**
   - EC2 instance store volumes provide better I/O performance and lower latency compared to Amazon S3 for temporary storage.
   - S3 Intelligent-Tiering is designed to optimize costs by automatically moving data to the most cost-effective access tier, but it does not provide the maximum I/O performance needed for processing large video files.
   - S3 Glacier Deep Archive is for long-term retention and digital preservation, not suitable for data that needs more frequent access.
2. **Amazon EBS for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage**
   - While Amazon EBS provides block-level storage volumes for EC2 instances, for high I/O performance, instance store volumes are a better option.
   - EBS is well-suited for primary storage for file systems and databases but does not match the high I/O performance of instance stores.
3. **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 suitable for hybrid cloud storage needs, providing on-premises access to cloud storage.
- In this case, the requirement is to move the infrastructure to AWS, not to retain on-premises storage.

Conclusion / Points to memorize:

1. **Amazon EC2 instance store** provides high I/O performance and low latency, suitable for temporary storage needs and processing large files.
2. **Amazon S3** offers high durability and availability for frequently accessed data.
3. **Amazon S3 Glacier** is ideal for low-cost, long-term archival storage.
4. **S3 Intelligent-Tiering** and **EBS** are not optimal for scenarios requiring the highest I/O performance for large video files.

Question 17 - (correct)

Question: An e-commerce company uses a two-tier architecture with application servers in the public subnet and an Amazon RDS MySQL DB in a private subnet. The development team can use a bastion host in the public subnet to access the MySQL database and run queries from the bastion host. However, end-users are reporting application errors. Upon inspecting application logs, the team notices several "could not connect to server: connection timed out" error messages. Which of the following options represent the root cause for this issue?

Correct option/s:

- The security group configuration for the database instance does not have the correct rules to allow inbound connections from the application servers

Explanation behind correct option/s:

1. **The security group configuration for the database instance does not have the correct rules to allow inbound connections from the application servers**
   - Security groups control the inbound and outbound traffic for your resources.
   - For the database instance in a private subnet, you need to ensure that its security group allows inbound traffic from the security group of the application servers.
   - If the security group for the RDS instance does not have the correct rules to allow inbound traffic from the application servers, the application servers will not be able to establish a connection, leading to "connection timed out" errors.

Incorrect options:

1. **The security group configuration for the application servers does not have the correct rules to allow inbound connections from the database instance**
   - Application servers do not require inbound connections from the database instance; rather, the database instance needs to allow inbound connections from the application servers.
2. **The database user credentials (username and password) configured for the application do not have the required privilege for the given database**
   - This issue would result in an "access denied" error, not a "connection timed out" error.
3. **The database user credentials (username and password) configured for the application are incorrect**
   - Incorrect credentials would also result in an "access denied" error, not a "connection timed out" error.

Conclusion / Points to memorize:

1. **Security groups** must be correctly configured to allow inbound traffic from the application servers to the RDS instance.
2. **Inbound rules** in the security group of the database instance should specify the security group of the application servers as the source.
3. **Connection timed out** errors often indicate a network connectivity issue, typically related to security group configurations.
4. **Database user credentials** issues result in "access denied" errors, not "connection timed out" errors.

Question 18 - (incorrect)

Question: 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 Amazon EC2 instance with the optimal storage type on the attached Amazon EBS volume. As a solutions architect, which of the following configurations would you suggest to the engineering team?

Correct option/s:

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

Explanation behind correct option/s:

1. **Amazon EC2 with Amazon EBS volume of Provisioned IOPS SSD (io1) type**
   - Amazon EBS Provisioned IOPS SSD (io1) volumes are designed for I/O-intensive applications and databases that require sustained IOPS performance or more than 16,000 IOPS per volume.
   - These volumes offer consistent performance with low latency, making them ideal for large database workloads such as PostgreSQL, MongoDB, Cassandra, Microsoft SQL Server, MySQL, and Oracle.
   - The io1 volumes support up to 64,000 IOPS per volume on Nitro-based instances, providing the necessary performance for the given use case.

Incorrect options:

1. **Amazon EC2 with Amazon EBS volume of General Purpose SSD (gp2) type**
   - General Purpose SSD (gp2) volumes are suitable for a wide variety of workloads and provide a balance between price and performance.
   - However, gp2 volumes do not offer the high IOPS required for massive databases like PostgreSQL that demand consistent performance with high IOPS.
2. **Amazon EC2 with Amazon EBS volume of Throughput Optimized HDD (st1) type**
   - Throughput Optimized HDD (st1) volumes are designed for frequently accessed, throughput-intensive workloads such as big data, data warehouses, and log processing.
   - They are not optimized for the I/O performance needed for large database workloads.
3. **Amazon EC2 with Amazon EBS volume of cold HDD (sc1) type**
   - Cold HDD (sc1) volumes are the lowest-cost HDD volumes designed for less frequently accessed workloads.
   - These volumes are not suitable for databases requiring high IOPS and low latency.

Conclusion / Points to memorize:

1. **Provisioned IOPS SSD (io1)** volumes are optimized for I/O-intensive applications and large database workloads requiring high IOPS and low latency.
2. **General Purpose SSD (gp2)** volumes provide a balance of price and performance for a variety of workloads but are not suitable for high IOPS requirements.
3. **Throughput Optimized HDD (st1)** volumes are designed for throughput-intensive workloads and not for I/O-intensive database workloads.
4. **Cold HDD (sc1)** volumes are the lowest-cost option for infrequently accessed data and are not suitable for high-performance database applications.

Question 19 - (incorrect)

Question: A Customer relationship management (CRM) application is facing user experience issues with users reporting frequent sign-in requests from the application. The application is currently hosted on multiple Amazon 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?

Correct option/s:

- Use Amazon Elasticache for distributed in-memory cache based session management

Explanation behind correct option/s:

1. **Use Amazon Elasticache for distributed in-memory cache based session management**
   - Amazon ElastiCache can be used as a distributed in-memory cache for session management.
   - It allows seamless setup, running, and scaling of popular open-source compatible in-memory data stores in the cloud.
   - ElastiCache supports both Memcached and Redis for session stores, making it ideal 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.
   - This solution ensures that session data is not lost when an individual server becomes unhealthy, improving user experience.

Incorrect options:

1. **Use Application Load Balancer sticky sessions**
   - Sticky sessions (also known as session affinity) enable each user to interact with one server only.
   - However, if the server becomes unhealthy, all the session data is lost, leading to the same user experience issues that are being faced.
2. **Use Amazon 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.
   - It is a NoSQL database and not designed for in-memory caching, making it unsuitable for this use case.
3. **Use Amazon RDS for distributed in-memory cache based session management**
   - Amazon RDS is a relational database service that 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, hence it is not suitable for session management.

Conclusion / Points to memorize:

1. **Amazon ElastiCache** is suitable for distributed in-memory cache-based session management, supporting both Redis and Memcached.
2. **Sticky sessions** tie users to individual servers, which can cause issues if those servers become unhealthy.
3. **Amazon DynamoDB** is a NoSQL database and not appropriate for in-memory caching needs.
4. **Amazon RDS** is a relational database service and not intended for use as an in-memory cache.

Question 20 - (correct)

Question: 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 as well as managed internally. The startup is now migrating to AWS Cloud and would like the data to be encrypted on AWS. The startup wants to continue using their existing encryption key generation as well as key management mechanism. What do you recommend?

Correct option/s:

- SSE-C

Explanation behind correct option/s:

1. **SSE-C (Server-Side Encryption with Customer-Provided Keys)**
   - With SSE-C, you manage the encryption keys while Amazon S3 manages the encryption as it writes to disks and decryption when you access your objects.
   - This allows the startup to continue generating and managing their encryption keys internally while leveraging AWS for the actual encryption and decryption processes.
   - This method ensures compliance with the requirement to use internally generated and managed encryption keys while benefiting from AWS's robust encryption infrastructure.

Incorrect options:

1. **SSE-KMS (Server-Side Encryption with AWS Key Management Service)**
   - AWS KMS provides a key management system scaled for the cloud, combining secure, highly available hardware and software.
   - While you can specify a customer-managed CMK (Customer Master Key) with SSE-KMS, you never get access to the actual encryption key itself, which does not align with the startup's requirement to manage the encryption keys internally.
2. **SSE-S3 (Server-Side Encryption with Amazon S3-Managed Keys)**
   - With SSE-S3, each object is encrypted with a unique key managed entirely by Amazon S3.
   - This option does not provide the capability to use internally generated and managed keys, nor does it allow for auditing the usage of encryption keys.
3. **Client-Side Encryption**
   - Client-side encryption involves encrypting data before sending it to Amazon S3.
   - Options for client-side encryption include using an AWS KMS key stored in AWS Key Management Service or a master key stored within the application.
   - Since the startup wants to leverage AWS's encryption facilities while managing their keys internally, client-side encryption is not the appropriate choice.

Conclusion / Points to memorize:

1. **SSE-C (Server-Side Encryption with Customer-Provided Keys)** allows you to manage your own encryption keys while AWS manages the encryption and decryption processes.
2. **SSE-KMS** and **SSE-S3** involve AWS managing the keys, which does not meet the requirement for internally managed encryption keys.
3. **Client-Side Encryption** requires encrypting data before sending it to AWS, which does not leverage AWS's encryption facilities as desired.
4. For scenarios requiring internal key management with AWS handling encryption and decryption, **SSE-C** is the optimal choice.

Question 21 - (correct)

Question: 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?

Correct option/s:

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

Explanation behind correct option/s:

1. **Use AWS Site-to-Site VPN to establish encrypted network connectivity between the on-premises data center and AWS Cloud**
   - AWS Site-to-Site VPN enables secure connectivity between your on-premises network or branch office site and your Amazon Virtual Private Cloud (VPC).
   - The VPN connection uses IPSec to establish encrypted network connectivity over the Internet, ensuring that data in transit is secure.
   - This solution can be set up quickly and provides the necessary encryption to meet the company's requirements.

Incorrect options:

1. **Use AWS Direct Connect to establish encrypted network connectivity between the on-premises data center and AWS Cloud**
   - AWS Direct Connect provides a dedicated network connection between your network and an AWS Direct Connect location.

- However, AWS Direct Connect does not encrypt traffic in transit by default. To encrypt the data, additional encryption methods must be implemented.
- Given the need for encryption in transit, AWS Direct Connect alone is not sufficient.

2. **Use AWS DataSync to establish encrypted network connectivity between the on-premises data center and AWS Cloud**
   - AWS DataSync is designed to move large amounts of data online between on-premises storage and AWS.
   - It handles data transfer tasks but does not establish network connectivity between an on-premises data center and AWS Cloud.
   - Therefore, AWS DataSync is not suitable for this requirement.
3. **Use AWS Secrets Manager to establish encrypted network connectivity between the on-premises data center and AWS Cloud**
   - AWS Secrets Manager helps manage and retrieve secrets such as database credentials and API keys.
   - It is not designed to establish network connectivity or handle data encryption in transit.
   - Thus, AWS Secrets Manager is not relevant to this use case.

Conclusion / Points to memorize:

1. **AWS Site-to-Site VPN** is the best choice for quickly establishing encrypted network connectivity between an on-premises data center and AWS Cloud.
2. **AWS Direct Connect** provides a dedicated connection but does not encrypt traffic by default.
3. **AWS DataSync** is for data transfer and not network connectivity.
4. **AWS Secrets Manager** is for managing secrets, not for establishing network connectivity or encryption.

Question 22 - (correct)

Question: A retail company maintains an AWS Direct Connect connection to AWS and has recently migrated its data warehouse to AWS. The data analysts at the company query the data warehouse using a visualization tool. The average size of a query returned by the data warehouse is 60 megabytes and the query responses returned by the data warehouse are not cached in the visualization tool. Each webpage returned by the visualization tool is approximately 600 kilobytes. Which of the following options offers the LOWEST data transfer egress cost for the company?

Correct option/s:

- Deploy the visualization tool in the same AWS region as the data warehouse. Access the visualization tool over a Direct Connect connection at a location in the same region

Explanation behind correct option/s:

1. **Deploy the visualization tool in the same AWS region as the data warehouse. Access the visualization tool over a Direct Connect connection at a location in the same region**
   - AWS Direct Connect is a networking service that provides a dedicated connection from your on-premises data center to AWS.
   - Data Transfer Out (DTO) charges apply to data transferred out of AWS to your on-premises data center.
   - By deploying the visualization tool in the same AWS region as the data warehouse and accessing it over Direct Connect, you minimize DTO charges because only the smaller data (600 KB for the visualization tool webpage) is transferred out via Direct Connect.
   - This option ensures the lowest egress costs since the larger data (60 MB query responses) does not incur DTO charges.

Incorrect options:

1. **Deploy the visualization tool in the same AWS region as the data warehouse. Access the visualization tool over the internet at a location in the same region**
   - Data transfer pricing over AWS Direct Connect is lower than data transfer pricing over the internet.
   - Therefore, accessing the tool over the internet would result in higher data transfer costs compared to using Direct Connect.
2. **Deploy the visualization tool on-premises. Query the data warehouse over the internet at a location in the same AWS region**
   - Similar to the previous point, data transfer over the internet is more expensive than Direct Connect.
   - Additionally, transferring 60 MB query responses over the internet would incur higher costs.
3. **Deploy the visualization tool on-premises. Query the data warehouse directly over an AWS Direct Connect connection at a location in the same AWS region**
   - This option incurs DTO charges for the 60 MB query responses from the data warehouse to the on-premises visualization tool.
   - As transferring 60 MB for each query is more expensive than transferring 600 KB for each webpage, this option is less cost-effective.

Conclusion / Points to memorize:

1. **AWS Direct Connect** offers lower data transfer egress costs compared to the internet.
2. Deploying services within the same AWS region can minimize DTO charges.
3. **Minimizing the amount of data transferred out of AWS** (e.g., by keeping larger data transfers within AWS) reduces costs.
4. Understanding data transfer costs is crucial for optimizing AWS usage and reducing expenses.

Question 23 - (incorrect)

Question: 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)?

Correct option/s:

- Configure an Amazon S3 bucket to enable AWS Multi-Factor Authentication (AWS MFA) delete
- Close the company's AWS account

Explanation behind correct option/s:

1. **Configure an Amazon S3 bucket to enable AWS Multi-Factor Authentication (AWS MFA) delete**
   - Only the root account user can enable or disable MFA delete on an Amazon S3 bucket. This is a security measure to prevent unauthorized deletions.
   - Even with full administrative privileges, an IAM user cannot perform this task.
2. **Close the company's AWS account**
   - Closing an AWS account is a sensitive operation that can only be performed by the root account user.
   - Full administrative privileges do not grant the ability to close the AWS account.

Incorrect options:

1. **Change the password for his own IAM user account**
   - An IAM user with full administrative privileges can change the password for their own IAM user account.
2. **Delete the IAM user for his manager**
   - With full administrative privileges, an IAM user can delete other IAM users, including those with higher roles or their manager.
3. **Delete an Amazon S3 bucket from the production environment**
   - An IAM user with full administrative privileges can delete any Amazon S3 bucket, regardless of the environment.

Conclusion / Points to memorize:

1. **Certain tasks require root user credentials:** Even with full administrative privileges, some tasks in AWS are reserved for the root account user for security reasons.
2. **Examples of root-only tasks:** These include enabling MFA delete on S3 buckets, closing the AWS account, changing the root password, and modifying root account settings.
3. **Administrator privileges:** While powerful, administrator privileges do not grant complete control over all AWS account settings and configurations.

Question 24 - (correct)

Question: 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)

Correct option/s:

- The route table in the instance's subnet should have a route to an Internet Gateway
- The network access control list (network ACL) associated with the subnet must have rules to allow inbound and outbound traffic

Explanation behind correct option/s:

1. **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.
   - For an EC2 instance to connect to the internet, its subnet's route table must have a route that directs internet-bound traffic to the Internet Gateway (IGW).
2. **The network access control list (network ACL) associated with the subnet must have rules to allow inbound and outbound traffic**
   - Network ACLs act as a firewall for controlling traffic in and out of subnets.
   - The network ACL associated with the subnet must have rules that allow inbound and outbound traffic on the required ports (e.g., port 80 for HTTP and port 443 for HTTPS) for internet connectivity.

Incorrect options:

1. **The instance's subnet is not associated with any route table**
   - This is incorrect because a subnet is always associated with a route table. If no specific route table is associated with a subnet, it uses the main route table by default.
2. **The instance's subnet is associated with multiple route tables with conflicting configurations**

- This is incorrect because a subnet can only be associated with one route table at a time. Therefore, there can be no conflicting route table configurations for a single subnet.

3. **The subnet has been configured to be public and has no access to the internet**
   - This is incorrect because by definition, a public subnet has a route to the Internet Gateway, allowing internet access. If it does not have internet access, it might not be correctly configured as a public subnet.

Conclusion / Points to memorize:

1. **Internet Gateway connectivity requirements:**
   - Ensure that the subnet's route table has a route to the Internet Gateway.
   - Ensure that the network ACL allows inbound and outbound traffic.
2. **Subnets and route tables:**
   - A subnet is always associated with a route table, either explicitly or by default (main route table).
   - A subnet can only have one route table associated with it at any time.
3. **Public subnet configurations:**
   - A public subnet must have a route to the Internet Gateway for internet access.
   - Properly configure network ACLs to allow necessary traffic for connectivity.

Question 25 - (correct)

Question: 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 via ad-hoc SQL queries. As a Solutions Architect, which of the following solutions would you recommend?

Correct option/s:

- Use Amazon Transcribe to convert audio files to text and Amazon Athena to understand the underlying customer sentiments

Explanation behind correct option/s:

1. **Use Amazon Transcribe to convert audio files to text and Amazon Athena to understand the underlying customer sentiments**
   - **Amazon Transcribe:** It is an automatic speech recognition (ASR) service that converts audio to text. It supports features like speaker identification to label individual speakers in multi-speaker audio files, making it ideal for analyzing call center recordings.
   - **Amazon Athena:** It is an interactive query service that allows you to analyze data in Amazon S3 using standard SQL. It is serverless, which means there's no infrastructure to manage, and you only pay for the queries you run. Athena can quickly query the transcribed text files to perform sentiment analysis.

By using these services together, the company can automate the transcription of audio files and run SQL queries to analyze customer sentiments.

Incorrect options:

1. **Use Amazon 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 Data Streams is designed for real-time data streaming and does not directly handle audio file processing. While it can be used to stream data, it cannot read audio files directly and convert them into text. AWS Transcribe is required for this purpose.
2. **Use Amazon Kinesis Data Streams to read the audio files and Amazon Alexa to convert them into text. Amazon Kinesis Data Analytics can be used to analyze these files and Amazon Quicksight can be used to visualize and display the output**
   - Amazon Kinesis Data Streams cannot read audio files, and Amazon Alexa is not designed as an ASR service for this purpose. Additionally, Amazon Kinesis Data Analytics is for real-time data analytics, and Amazon Quicksight is for data visualization, not SQL-based analysis.
3. **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 used for data visualization and dashboarding, not for SQL-based querying and analysis. Amazon Athena is more suitable for running SQL queries on the transcribed text files for sentiment analysis.

Conclusion / Points to memorize:

1. **Amazon Transcribe:** Best suited for converting audio files to text with features like speaker identification.
2. **Amazon Athena:** Ideal for running ad-hoc SQL queries on data stored in Amazon S3, suitable for sentiment analysis of transcribed text.
3. **Amazon Quicksight:** Used for data visualization and dashboards, not for SQL query-based analysis.
4. **Amazon Kinesis Data Streams:** Suitable for real-time data streaming, not directly for processing audio files.

Question 26 - (correct)

Question: A medium-sized business has a taxi dispatch application deployed on an Amazon 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?

Correct option/s:

- Set up an Amazon 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.

Explanation behind correct option/s:

1. **Set up an Amazon 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.**
   - **Amazon CloudWatch Alarm Actions:** CloudWatch alarms can automatically perform actions such as stopping, terminating, rebooting, or recovering EC2 instances. The reboot alarm action is ideal for handling Instance Health Check failures and is resource-efficient as it directly reboots the instance without additional steps.
   - **Instance Health Check:** This checks the health of the EC2 instance and can trigger the alarm to reboot the instance if it detects a failure.

Using the CloudWatch alarm's native ability to reboot the instance is the most straightforward and resource-efficient solution, ensuring the instance is automatically rebooted when it freezes.

Incorrect options:

1. **Set up an Amazon CloudWatch alarm to monitor the health status of the instance. In case of an Instance Health Check failure, Amazon CloudWatch Alarm can publish to an Amazon Simple Notification Service (Amazon SNS) event which can then trigger an AWS Lambda function. The AWS Lambda function can use Amazon EC2 API to reboot the instance.**
   - This approach introduces unnecessary complexity by involving SNS and Lambda functions when the CloudWatch alarm itself can directly reboot the instance.
2. **Use Amazon EventBridge events to trigger an AWS Lambda function to check the instance status every 5 minutes. In the case of Instance Health Check failure, the AWS Lambda function can use Amazon EC2 API to reboot the instance.**
   - Similar to the previous incorrect option, this approach is more complex and resource-intensive than necessary, as EventBridge and Lambda functions are not needed for a straightforward reboot.
3. **Use Amazon EventBridge events to trigger an AWS Lambda function to reboot the instance status every 5 minutes.**
   - This approach is inefficient because it involves periodically invoking a Lambda function to reboot the instance without checking its health status, leading to unnecessary reboots and resource consumption.

Conclusion / Points to memorize:

1. **Amazon CloudWatch Alarm Actions:** These can be used to automatically reboot an EC2 instance upon detecting health check failures, making it a cost-efficient and resource-efficient solution.
2. **Avoid Unnecessary Complexity:** Directly using CloudWatch alarm actions to reboot instances is simpler and more effective than involving SNS, EventBridge, and Lambda functions for the same purpose.
3. **Instance Health Check:** This mechanism checks the instance's health and can trigger appropriate actions like reboots to maintain application availability.

Question 27 - (correct)

Question: A global media company uses a fleet of Amazon 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 an Amazon 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?

Correct option/s:

- Use AWS Web Application Firewall (AWS WAF) with Amazon CloudFront distribution

Explanation behind correct option/s:

1. **Use AWS Web Application Firewall (AWS WAF) with Amazon CloudFront distribution**
   - **AWS WAF:** 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.

- **Integration with CloudFront:** 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 Amazon 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 AWS WAF with Amazon CloudFront can effectively prevent SQL injection and cross-site scripting attacks.

Incorrect options:

1. **Use Amazon Route 53 with Amazon CloudFront distribution**
   - **Amazon Route 53:** Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It does not provide capabilities to prevent SQL injection and cross-site scripting attacks. Therefore, this option is incorrect.
2. **Use AWS Security Hub with Amazon CloudFront distribution**
   - **AWS Security Hub:** AWS Security Hub gives you a comprehensive view of your high-priority security alerts and security posture across your AWS accounts. While it aggregates, organizes, and prioritizes security alerts from multiple AWS services, it does not directly prevent SQL injection and cross-site scripting attacks. Therefore, this option is incorrect.
3. **Use AWS Firewall Manager with CloudFront distribution**
   - **AWS Firewall Manager:** 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. While it helps in managing firewall rules, it is not directly used to prevent SQL injection and cross-site scripting attacks. Therefore, this option is incorrect.

Conclusion / Points to memorize:

1. **AWS WAF:** AWS WAF is specifically designed to protect web applications by allowing you to create security rules that block common attack patterns such as SQL injection and cross-site scripting.
2. **Integration with CloudFront:** Integrating AWS WAF with Amazon CloudFront provides a robust solution for filtering and blocking malicious traffic before it reaches your application.
3. **Other AWS Services:** While AWS Route 53, AWS Security Hub, and AWS Firewall Manager offer various security and management capabilities, they do not directly address the specific need to prevent SQL injection and cross-site scripting attacks as AWS WAF does.

Question 28 - (incorrect)

Question: A company wants to publish an event into an Amazon Simple Queue Service (Amazon SQS) queue whenever a new object is uploaded on Amazon S3. Which of the following statements are true regarding this functionality?

Correct option/s:

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

Explanation behind correct option/s:

1. **Only Standard Amazon SQS queue is allowed as an Amazon S3 event notification destination, whereas FIFO SQS queue is not allowed**
   - **Amazon S3 Notification:** 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.
   - **Supported Destinations:** 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
   - **Standard SQS Only:** Currently, only the Standard Amazon SQS queue is allowed as an Amazon S3 event notification destination, whereas the FIFO SQS queue is not allowed. This restriction is important to ensure proper functioning of event notifications without the need for the strict ordering guarantees provided by FIFO queues.

Incorrect options:

1. **Both Standard Amazon SQS queue and FIFO SQS queue are allowed as an Amazon S3 event notification destination**
   - This option is incorrect because only the Standard Amazon SQS queue is allowed as an Amazon S3 event notification destination, not the FIFO SQS queue.
2. **Neither Standard Amazon SQS queue nor FIFO SQS queue are allowed as an Amazon S3 event notification destination**
   - This option is incorrect because the Standard Amazon SQS queue is allowed as an Amazon S3 event notification destination.
3. **Only FIFO Amazon SQS queue is allowed as an Amazon S3 event notification destination, whereas Standard SQS queue is not allowed**
   - This option is incorrect because only the Standard Amazon SQS queue is allowed, not the FIFO SQS queue.

Conclusion / Points to memorize:

1. **Amazon S3 Event Notification:** Amazon S3 can send notifications for specific events in your bucket to various destinations.
2. **Supported Destinations:** Amazon S3 supports notifications to Amazon SNS topics, Amazon SQS queues, and AWS Lambda functions.
3. **Standard SQS Queue Only:** Only the Standard Amazon SQS queue is allowed as an Amazon S3 event notification destination. FIFO SQS queues are not supported for this purpose.

Question 29 - (incorrect)

Question: The engineering team at an online fashion retailer uses AWS Cloud to manage its technology infrastructure. The Amazon 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?

Correct option/s:

- **Auto Scaling group scheduled action**

Explanation behind correct option/s:

1. **Auto Scaling group scheduled action:**
   - **Scheduled Action:** The engineering team can create a scheduled action for the Auto Scaling group to preemptively provision additional instances for the sale duration. This ensures that adequate instances are ready before the sale goes live.
   - **Functionality:** 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.
   - **Benefits:** This proactive approach ensures that the necessary infrastructure is in place before the anticipated traffic spike, thus maintaining the performance and availability of the application during high-demand periods.

Incorrect options:

1. **Auto Scaling group target tracking scaling policy:**
   - **Functionality:** With target tracking scaling policies, you choose a scaling metric and set a target value. Application Auto Scaling creates and manages the Amazon CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value.
   - **Limitation:** Target tracking scaling policies are reactive in nature and provision instances only when the underlying Amazon CloudWatch alarms go off. This entails a lag and is not suitable for preemptively handling anticipated traffic spikes.
2. **Auto Scaling group step scaling policy:**
   - **Functionality:** With step scaling, you choose scaling metrics and threshold values for the Amazon 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.
   - **Limitation:** Like target tracking, step scaling policies are also reactive and involve a delay as instances are provisioned only when CloudWatch alarms are triggered. This is not ideal for preemptively addressing known traffic surges.
3. **Auto Scaling group lifecycle hook:**
   - **Functionality:** 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.
   - **Limitation:** Lifecycle hooks are used for managing custom actions during instance launch or termination but cannot be used to preemptively provision additional instances for a specific period such as the sale duration.

Conclusion / Points to memorize:

1. **Scheduled Action:** Use Auto Scaling group scheduled actions to proactively provision instances for anticipated traffic spikes.
2. **Reactive Policies:** Target tracking and step scaling policies are reactive and trigger scaling based on CloudWatch alarms, which is not ideal for preemptive scaling.
3. **Lifecycle Hooks:** Lifecycle hooks are for custom actions during instance lifecycle events and do not address preemptive scaling needs.

Question 30 - Incorrect

The engineering team at an e-commerce company uses an AWS Lambda function to write the order data into a single DB instance Amazon Aurora cluster. The team has noticed that many order-writes to its Aurora cluster are getting missed during peak load times. The diagnostics data has revealed that the

database is experiencing high CPU and memory consumption during traffic spikes. The team also wants to enhance the availability of the Aurora DB. Which of the following steps would you combine to address the given scenario? (Select two)

Correct options:

1. Handle all read operations for your application by connecting to the reader endpoint of the Amazon Aurora cluster so that Aurora can spread the load for read-only connections across the Aurora replica.
2. Create a replica Aurora instance in another Availability Zone to improve the availability as the replica can serve as a failover target.

Explanation behind correct options:

1. **Handle all read operations by connecting to the reader endpoint:**
   - Spreads the load for read-only connections across Aurora replicas.
   - Reduces load on the primary instance, addressing high CPU and memory consumption.
2. **Create a replica Aurora instance in another Availability Zone:**
   - Enhances availability as the replica can serve as a failover target.
   - Aurora automatically promotes a replica to the new primary instance in case of failure.

Incorrect options:

1. **Increase the concurrency of the AWS Lambda function:**
   - Does not address the bottleneck at the database layer.
   - The issue is high CPU and memory consumption in Aurora, not Lambda concurrency.
2. **Use Amazon EC2 instances behind an Application Load Balancer:**
   - Does not solve the database layer bottleneck.
   - High CPU and memory consumption in Aurora needs to be addressed.
3. **Create a standby Aurora instance in another Availability Zone:**
   - Aurora does not support creating standby instances.
   - Uses replicas for failover instead.

Conclusion / Points to memorize:

1. Use Aurora replicas to handle read operations and reduce load on the primary instance.
2. Create Aurora replicas in different Availability Zones to enhance availability and serve as failover targets.
3. Increasing Lambda concurrency or using EC2 with an ALB does not address database bottlenecks.
4. Aurora does not support standby instances; it uses replicas for high availability.

Question 31 - Correct

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?

Correct options:

1. Amazon Kinesis Data Streams

Explanation behind correct options:

1. **Amazon Kinesis Data Streams:**
   - Massively scalable and durable real-time data streaming service.
   - Can capture gigabytes of data per second from hundreds of thousands of sources.
   - Supports high throughput messages with ordered data based on the sensor's key.
   - Provides partition key to guarantee ordered messages for a specific sensor, even if the stream is sharded.

Incorrect options:

1. **Amazon Simple Queue Service (Amazon SQS):**
   - Fully managed message queuing service for decoupling and scaling microservices.
   - Not designed for real-time data streaming.
2. **Amazon Simple Notification Service (Amazon SNS):**
   - Highly available, durable, secure pub/sub messaging service.
   - Not suitable for data streaming, more suited for push-based, many-to-many messaging.
3. **AWS Lambda:**
   - Serverless compute service to run code without provisioning or managing servers.

- Not designed for production-grade serverless log analytics or data streaming.

Conclusion / Points to memorize:

1. Amazon Kinesis Data Streams is ideal for high throughput, ordered data streaming.
2. SQS and SNS are not suitable for real-time data streaming scenarios.
3. AWS Lambda is not intended for data retention or real-time streaming analytics.
4. For streaming systems requiring ordered data and high throughput, Kinesis Data Streams is the best choice.

Question 32 - Incorrect

You are a cloud architect at an IT company. The company has multiple enterprise customers that manage their own mobile applications 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?

Correct options:

1. Use batch messages

Explanation behind correct options:

1. **Use batch messages:**
   - Amazon Kinesis Data Streams (KDS) is designed to handle real-time data streaming at high throughput.
   - Batching records and implementing parallel HTTP requests increase efficiency and ensure optimal use of shards.
   - Batching reduces the overhead and increases the throughput compared to sending one message at a time.

Incorrect options:

1. **Use Exponential Backoff:**
   - Helps temporarily by reducing the rate of requests during retries.
   - Not a long-term solution as high request rates will still cause ProvisionedThroughputExceededException.
2. **Increase the number of shards:**
   - Provides more capacity but significantly increases costs.
   - Not a cost-effective solution compared to batching messages.
3. **Decrease the Stream retention duration:**
   - Does not address the issue of throughput limits.
   - May result in data loss and does not help with exceptions related to high message rates.

Conclusion / Points to memorize:

1. Batch messages to increase efficiency and optimize shard usage in Kinesis Data Streams.
2. Exponential backoff helps with temporary throughput issues but is not a permanent fix.
3. Increasing shards increases capacity but is costly.
4. Stream retention duration does not impact throughput and can lead to data loss.
5. For high throughput messaging, batching is the most cost-effective and efficient solution.

Question 33 - Correct

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?

Correct option:

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

Explanation behind correct options:

1. **Strong Read-After-Write Consistency:**
   - Amazon S3 provides strong read-after-write consistency for PUT and DELETE requests of objects.
   - After a successful write or overwrite of an object, subsequent read requests will immediately reflect the latest version of the object.
   - Strong consistency applies to all operations, including GET, PUT, and LIST, ensuring no data discrepancies.

Incorrect options:

1. **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:**

- This is incorrect as S3 ensures strong read-after-write consistency, so it will not return outdated data.

2. **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:**
   - This is incorrect as S3 will return the latest version of the data immediately after an overwrite.
3. **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:**
   - This is incorrect because S3 provides immediate consistency, so the latest data is always returned.

Conclusion / Points to memorize:

1. **Amazon S3 provides strong read-after-write consistency for all object operations.**
2. **Subsequent reads after a write or overwrite will always reflect the latest data.**
3. **S3 consistency ensures no data discrepancies for parallel reads and writes.**
4. **All S3 operations, including GET, PUT, and LIST, are strongly consistent.**
5. **S3 handles overwrites seamlessly, ensuring immediate access to updated data.**

Question 34 - Incorrect

A mobile chat application uses Amazon 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 Amazon DynamoDB which have been tweaked for certain technical requirements. AWS CloudTrail service has been enabled on all the resources used for the project. Yet, Amazon DynamoDB encryption details are nowhere to be found. Which of the following options can explain the root cause for the given issue?

Correct option:

- By default, all Amazon DynamoDB tables are encrypted using AWS owned keys, which do not write to AWS CloudTrail logs.

Explanation behind correct option:

1. **AWS Owned Keys:**
   - AWS owned keys are part of a collection of KMS keys that AWS owns and manages for use in multiple AWS accounts.
   - AWS owned keys are not stored in your AWS account and you cannot view, manage, or use them.
   - They do not write to AWS CloudTrail logs, which explains why the encryption details are not found in CloudTrail.
   - All DynamoDB tables are encrypted by default under an AWS owned key.

Incorrect options:

1. **By default, all Amazon DynamoDB tables are encrypted under AWS managed Keys, which do not write to AWS CloudTrail logs:**
   - AWS managed keys are different from AWS owned keys and their use can be tracked in CloudTrail logs.
2. **By default, all Amazon DynamoDB tables are encrypted under Customer managed keys, which do not write to AWS CloudTrail logs:**
   - Customer managed keys are user-controlled and their usage is tracked in CloudTrail logs.
3. **By default, all Amazon DynamoDB tables are encrypted using Data keys, which do not write to AWS CloudTrail logs:**
   - Data keys are generated by AWS KMS to encrypt data, but their management and usage are tracked in CloudTrail logs.

Conclusion / Points to memorize:

1. **DynamoDB Encryption Defaults:**
   - By default, all Amazon DynamoDB tables are encrypted using AWS owned keys.
2. **AWS Owned Keys:**
   - AWS owned keys are not stored in the user's account, cannot be viewed, managed, or audited by the user, and do not write to CloudTrail logs.
3. **Tracking Encryption in DynamoDB:**
   - To have encryption details available in AWS CloudTrail, use AWS managed keys or customer managed keys for encryption.
4. **Encryption Flexibility:**
   - Users can choose to encrypt DynamoDB tables with customer managed keys or AWS managed keys if needed, for more control and auditing capabilities.

Question 35 - Correct

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 a highly available as well as HIPAA compliant in-memory database that supports caching results of SQL queries. As a solutions architect, which of the following AWS services would you recommend for this task?

Correct option:

- Amazon ElastiCache for Redis/Memcached

Explanation behind correct option:

- **Amazon ElastiCache Overview:**

- **Redis:**
  - Blazing fast in-memory data store providing sub-millisecond latency.
  - Supports real-time transactional and analytical processing, caching, chat/messaging, gaming leaderboards, geospatial, machine learning, media streaming, queues, real-time analytics, and session store.
  - Supports replication, high availability, and cluster sharding.
- **Memcached:**
  - Memcached-compatible in-memory key-value store.
  - Ideal for implementing an in-memory cache to decrease latency, increase throughput, and reduce load on databases.
  - Easy to create session stores.
- **HIPAA Compliance:**
  - Both Redis and Memcached under Amazon ElastiCache are HIPAA Eligible.

Incorrect options:

1. **Amazon DynamoDB Accelerator (DAX):**
   - DAX is a caching service for DynamoDB but does not support SQL query caching.
2. **Amazon DynamoDB:**
   - Key-value and document database, not an in-memory database.
   - Not designed for caching SQL query results.
3. **Amazon DocumentDB:**
   - Document database for MongoDB workloads, not an in-memory database.
   - Unsuitable for SQL query caching.

Conclusion / Points to memorize:

1. Amazon ElastiCache for Redis/Memcached supports in-memory data storage and is HIPAA compliant.
2. Redis offers sub-millisecond latency and supports real-time use cases.
3. Memcached is suitable for in-memory caching to reduce latency and load on databases.
4. DAX, DynamoDB, and DocumentDB do not meet the in-memory SQL query caching requirement.

Question 36 - Correct

A cyber security company is running a mission-critical application using a single Spread placement group of Amazon EC2 instances. The company needs 15 Amazon EC2 instances for optimal performance. How many Availability Zones (AZs) will the company need to deploy these Amazon EC2 instances per the given use-case?

Correct option:

- 3

Explanation behind correct option:

- **Spread Placement Group:**
  - A group of instances that are each placed on distinct racks with their own network and power source.
  - Recommended for applications with a small number of critical instances that should be kept separate to reduce the risk of simultaneous failures.
- **Spread Placement Group Capacity:**
  - Can span multiple Availability Zones in the same Region.
  - Maximum of 7 running instances per Availability Zone per group.
  - To deploy 15 instances, 3 Availability Zones are required (15 instances / 7 instances per AZ = 2.14, rounded up to 3).

Incorrect options:

1. **7:**
   - Incorrect because you only need 3 Availability Zones based on the 7 instances per AZ limit.
2. **14:**
   - Incorrect because this would imply spreading the instances across more AZs than necessary.
3. **15:**
   - Incorrect because it implies one instance per AZ, which is not necessary or efficient.

Conclusion / Points to memorize:

1. **Spread Placement Group** allows spreading instances across multiple Availability Zones.
2. **Maximum 7 instances per AZ** in a Spread Placement Group.
3. For 15 instances, **3 Availability Zones** are required (15 / 7 = 3).

Question 37 - Correct

The engineering team at a retail company manages 3 Amazon EC2 instances that make read-heavy database requests to the Amazon RDS for the PostgreSQL database instance. As an AWS Certified Solutions Architect - Associate, you have been tasked to make the database instance resilient from a disaster recovery perspective. Which of the following features will help you in disaster recovery of the database? (Select two)

Correct options:

- Use cross-Region Read Replicas
- Enable the automated backup feature of Amazon RDS in a multi-AZ deployment that creates backups across multiple Regions

Explanation behind correct options:

1. **Use cross-Region Read Replicas:**
   - Read Replicas help reduce the load on the source database and can be used for disaster recovery.
   - Read Replicas can be promoted to standalone source servers if the primary DB fails.
   - Cross-Region Read Replicas provide resilience against regional availability issues.
2. **Enable the automated backup feature of Amazon RDS in a multi-AZ deployment that creates backups across multiple Regions:**
   - Automated backups enable point-in-time recovery for your database instance.
   - Backups occur on standby in a Multi-AZ configuration to reduce I/O impact on the primary.
   - Cross-Region Automated Backups ensure data is backed up in different regions, enhancing disaster recovery.

Incorrect options:

1. **Enable the automated backup feature of Amazon RDS in a multi-AZ deployment that creates backups in a single AWS Region:**
   - This limits the backups to a single region, not providing regional disaster recovery.
2. **Use Amazon RDS Provisioned IOPS (SSD) Storage in place of General Purpose (SSD) Storage:**
   - Provisioned IOPS enhances performance but does not contribute to disaster recovery.
3. **Use the database cloning feature of the Amazon RDS Database cluster:**
   - Database cloning is only available for Amazon Aurora, not Amazon RDS.

Conclusion / Points to memorize:

1. **Cross-Region Read Replicas** are essential for regional disaster recovery.
2. **Automated backups** in a multi-AZ deployment across multiple regions enhance disaster recovery.
3. **Multi-AZ deployments** ensure high availability and failover support within a region.
4. **Provisioned IOPS** improves performance but does not help in disaster recovery.
5. **Database cloning** is not applicable for Amazon RDS, only for Amazon Aurora.

Question 38 - Correct

You have built an application that is deployed with Elastic Load Balancing and an Auto Scaling Group. As a Solutions Architect, you have configured aggressive Amazon 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 Auto Scaling Group. The log files are saved on the Amazon EC2 instance. How will you resolve the issue and make sure it doesn't happen again?

Correct options:

- Install an Amazon CloudWatch Logs agent on the Amazon EC2 instances to send logs to Amazon CloudWatch

Explanation behind correct options:

1. **Install an Amazon CloudWatch Logs agent on the Amazon EC2 instances to send logs to Amazon CloudWatch:**
   - The Amazon CloudWatch Logs agent allows automatic sending of log files from the EC2 instance to CloudWatch.
   - Logs can be analyzed even after the instance is terminated, ensuring no loss of crucial debugging information.
   - This solution is automated, cost-effective, and integrates seamlessly with existing AWS services.

Incorrect options:

1. **Disable the Termination from the Auto Scaling Group any time a user reports an issue:**
   - Disabling termination affects the elasticity and cost-efficiency of the Auto Scaling Group.
   - This is not a sustainable solution for managing production issues.
2. **Make a snapshot of the Amazon EC2 instance just before it gets terminated:**
   - This approach is tedious, not elastic, and expensive.
   - Snapshots are meant for data backup rather than for log analysis purposes.

3. **Use AWS Lambda to regularly SSH into the Amazon EC2 instances and copy the log files to Amazon S3:**
    - Using AWS Lambda for SSH operations is complex and not recommended for log analytics.
    - This solution is cumbersome and not scalable for production-grade log management.

Conclusion / Points to memorize:

1. **Amazon CloudWatch Logs agent** is the best solution for automatic log management and retrieval.
2. **Auto Scaling Groups** should maintain elasticity and cost-efficiency, avoiding manual intervention.
3. **Snapshots** are for data backup and not ideal for log analysis.
4. **AWS Lambda** is not suited for server-based operations like SSH for log collection.

Question 39 - Correct

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?

Correct options:

- Amazon API Gateway exposing AWS Lambda Functionality

Explanation behind correct options:

1. **Amazon API Gateway exposing AWS Lambda Functionality:**
    - Amazon API Gateway is a fully managed service that allows 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 backend services.
    - AWS Lambda lets you run code without provisioning or managing servers, and you pay only for the compute time you consume.
    - Amazon API Gateway can expose AWS Lambda functionality through RESTful APIs, making both services the right choice for a serverless architecture.

Incorrect options:

1. **AWS Fargate with AWS Lambda at the front:**
    - AWS Lambda cannot directly handle RESTful API requests.
    - Invoking an AWS Lambda function over HTTPS requires defining a custom RESTful API using Amazon API Gateway.
    - Combining AWS Fargate with AWS Lambda is not a suitable solution for handling RESTful API requests directly.
2. **Public-facing Application Load Balancer with Amazon Elastic Container Service (Amazon ECS) on Amazon EC2:**
    - Amazon ECS on Amazon EC2 does not fall under serverless architecture.
    - This option involves managing servers, which contradicts the goal of moving to a serverless architecture.
3. **Amazon Route 53 with Amazon EC2 as backend:**
    - Amazon EC2 is not a serverless service.
    - Using Amazon EC2 contradicts the requirement of implementing a serverless architecture.

Conclusion / Points to memorize:

1. **Amazon API Gateway** and **AWS Lambda** are core components of serverless architecture for creating RESTful APIs.
2. **Amazon API Gateway** exposes AWS Lambda functionality through RESTful APIs, making it a perfect fit for serverless solutions.
3. **Serverless architecture** means no need to manage servers, relying on fully managed services like **API Gateway** and **Lambda**.
4. **Incorrect options** involve using non-serverless components such as Amazon EC2, which does not align with the serverless architecture goal.

Question 40 - Correct

A medical devices company uses Amazon 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 Amazon 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 Amazon S3 while keeping the IT team's involvement to a minimum?

Correct options:

- Use Amazon S3 Intelligent-Tiering storage class to optimize the Amazon S3 storage costs

Explanation behind correct options:

1. Amazon S3 Intelligent-Tiering:
    - Automatically moves data to the most cost-effective access tier based on access patterns.
    - Stores objects in two access tiers: one optimized for frequent access and another for infrequent access.
    - Moves objects that haven't been accessed for 30 consecutive days to the infrequent access tier.
    - Moves objects back to the frequent access tier if accessed again.
    - No retrieval fees and no additional tiering fees when objects are moved between access tiers.
    - Ideal for long-lived data with unpredictable access patterns.
    - Helps reduce costs while minimizing IT team involvement.

Incorrect options:

1. Configure Amazon EFS:

- Amazon EFS is a scalable NFS file system for use with AWS Cloud services and on-premises resources.
- Costlier than Amazon S3 and requires Amazon EC2 instances or AWS Direct Connect.
- Not the best fit for reducing Amazon S3 storage costs.

2. Use Amazon S3 One Zone-Infrequent Access:
- Stores data in a single AZ and costs 20% less than Amazon S3 Standard-IA.
- Suitable for less frequently accessed data that doesn't require high availability.
- Not ideal for business-critical data due to single AZ storage risk.

3. Use Amazon S3 Outposts:
- Delivers object storage to on-premises AWS Outposts environments.
- Used with AWS Outposts and not relevant to the current use case of optimizing S3 storage costs.

Conclusion / Points to memorize:

1. Amazon S3 Intelligent-Tiering is the optimal solution for automatically reducing S3 storage costs with minimal IT involvement.
2. Intelligent-Tiering automatically moves data between frequent and infrequent access tiers based on usage patterns.
3. Amazon S3 One Zone-IA is cost-effective but not suitable for critical data due to its single AZ storage.
4. Amazon EFS and S3 Outposts are not suitable for optimizing S3 storage costs in this scenario.

Question 41 - Correct

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 Amazon EC2 instances, and an Amazon 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 Amazon EC2 instances?

Correct options:

- Add a rule authorizing the Amazon EC2 security group

Explanation behind correct options:

1. **Amazon EC2 Security Group:**
   - Security groups act as virtual firewalls that control traffic for one or more instances.
   - To ensure only the Amazon EC2 instances can access the Aurora database, add a rule to the Aurora security group that authorizes traffic from the Amazon EC2 security group.
   - This ensures that only traffic from instances within the specified security group is allowed to reach the Aurora database.

Incorrect options:

1. **Add a rule authorizing the Amazon Aurora security group:**
   - This option is incorrect as it does not control traffic from the Amazon EC2 instances to the Aurora database.
2. **Add a rule authorizing the Auto Scaling group subnets CIDR:**
   - Authorizing the entire CIDR of the Auto Scaling group's subnets is over-permissive and could allow access from non-Auto Scaling Group instances within the same CIDR range.
3. **Add a rule authorizing the Elastic Load Balancing security group:**
   - Adding a rule authorizing the ELB security group would allow traffic from the load balancer, not specifically from the Amazon EC2 instances managed by the Auto Scaling group. This is not secure for the Aurora database layer which should only be accessed by the backend instances.

Conclusion / Points to memorize:

1. **Security groups** act as virtual firewalls that control traffic to instances.
2. **Authorize specific security groups** in your rules to limit access to only necessary instances.
3. **Avoid over-permissive rules** such as authorizing entire CIDR ranges or unrelated security groups.
4. For a secure architecture, **ensure that only the intended instances** (e.g., Amazon EC2 instances in the Auto Scaling group) can access your database layer by configuring security group rules appropriately.

Question 42 - Incorrect

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 Amazon 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?

Correct option:

- Use Amazon Athena to run SQL based analytics against Amazon S3 data

Explanation behind correct options:

1. **Amazon Athena:**
   - **Serverless and No ETL:** Athena is serverless, eliminating the need to set up and manage servers or data warehouses.
   - **Cost-Effective:** You only pay for the queries you run, making it a cost-efficient solution for ad-hoc queries.

- **Built on Presto:** Supports standard SQL and integrates seamlessly with data stored in Amazon S3.
- **Fast and Scalable:** Can handle large datasets with interactive performance, making it suitable for data sanity checks on clickstream data.

Incorrect options:

1. **Load the incremental raw zone data into Amazon Redshift on an hourly basis and run the SQL based sanity checks:**
   - **High Maintenance:** Requires maintaining and monitoring the Redshift cluster, which involves significant development time and effort.
   - **Cost Implications:** Running an hourly job to load data into Redshift can be costly and resource-intensive.
2. **Load the incremental raw zone data into an Amazon EMR based Spark Cluster on an hourly basis and use SparkSQL to run the SQL based sanity checks:**
   - **Infrastructure Management:** Requires managing the underlying infrastructure, which contradicts the requirement for a low-maintenance solution.
   - **Development Effort:** Setting up and maintaining an EMR cluster adds complexity and development overhead.
3. **Load the incremental raw zone data into Amazon RDS on an hourly basis and run the SQL based sanity checks:**
   - **Data Migration Jobs:** Involves writing data migration jobs, increasing the development effort.
   - **Ongoing Maintenance:** Maintaining RDS instances and ensuring data consistency can be resource-intensive.

Conclusion / Points to memorize:

1. **Amazon Athena** is ideal for running SQL queries on data stored in Amazon S3, offering a serverless, cost-effective, and easy-to-maintain solution.
2. **Avoid solutions** that require heavy infrastructure management or significant development effort, such as Redshift, EMR, or RDS.
3. **Opt for serverless** and pay-per-query services like Athena to minimize costs and operational overhead while ensuring scalability and performance.

Question 43 - Incorrect

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 Amazon 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?

Correct option:

- NAT Gateways deployed in your public subnet

Explanation behind correct option:

1. **NAT Gateways:**
   - **Managed by AWS:** NAT Gateways are fully managed by AWS, reducing maintenance overhead.
   - **High Availability:** Deployed in each Availability Zone to ensure redundancy.
   - **Scalability:** Can scale up to 45 Gbps of bandwidth automatically.
   - **Security:** Use network ACLs to control traffic; security groups cannot be associated.
   - **Supports IPv4:** Enables instances in private subnets to connect to the internet over IPv4.
   - **Cost:** Charged based on usage and data processed.

Incorrect options:

1. **NAT Instances deployed in your public subnet:**
   - **Manual Management:** Requires you to manage the instances, including scaling and updates.
   - **Scalability Issues:** Not as scalable as NAT Gateways, which can automatically handle more traffic.
2. **Internet Gateways deployed in your private subnet:**
   - **Wrong Placement:** Internet Gateways must be deployed in public subnets, not private ones.
   - **Functionality:** Primarily used to allow communication between instances in VPC and the internet, not suitable for providing internet access to private subnets.
3. **Egress-Only Internet Gateways deployed in your private subnet:**
   - **IPv6 Only:** Designed for outbound IPv6 traffic and does not support IPv4, which is a requirement in this scenario.

Conclusion / Points to memorize:

1. **NAT Gateways** are the preferred solution for enabling internet access for instances in private subnets, offering scalability, high availability, and reduced maintenance.
2. **NAT Instances** require manual management and do not scale as effectively as NAT Gateways.
3. **Internet Gateways** must be placed in public subnets and are not suitable for providing internet access to private subnets.
4. **Egress-Only Internet Gateways** support only IPv6 and are not applicable for IPv4 use cases.

Question 44 - Correct

A company wants to store business-critical data on Amazon Elastic Block Store (Amazon EBS) volumes which provide persistent storage independent of Amazon EC2 instances. During a test run, the development team found that on terminating an Amazon EC2 instance, the attached Amazon EBS volume was also lost, which was contrary to their assumptions. As a solutions architect, could you explain this issue?

Correct option:

- The Amazon 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.

Explanation behind correct option:

1. **Amazon EBS Overview:**
   - Amazon Elastic Block Store (EBS) provides persistent block storage volumes for use with Amazon EC2 instances.
   - EBS volumes are independent of the lifecycle of an EC2 instance and can persist after the instance is terminated if configured correctly.
2. **Default Behavior:**
   - By default, the root volume of an EC2 instance (the volume containing the OS) is deleted upon instance termination.
   - This default behavior is intended to ensure that temporary data stored on the root volume does not persist beyond the lifecycle of the instance.
3. **Customization:**
   - Users can modify the default behavior to retain the root volume after instance termination by changing the "Delete on Termination" attribute.
   - Non-root (secondary) EBS volumes are not deleted by default and remain available even after instance termination unless explicitly configured to be deleted.

Incorrect options:

1. **The Amazon EBS volumes were not backed up on Amazon S3 storage, resulting in the loss of volume:**
   - Amazon EBS volumes do not require backups on Amazon S3 to persist beyond instance termination.
   - This option is incorrect as the issue is related to the default termination behavior of root volumes, not the lack of backups.
2. **The Amazon EBS volumes were not backed up on Amazon EFS file system storage, resulting in the loss of volume:**
   - Similar to S3, EBS volumes do not need to be backed up on Amazon EFS for persistence.
   - The problem is related to the default termination setting of root volumes, making this option incorrect.
3. **On termination of an Amazon EC2 instance, all the attached Amazon EBS volumes are always terminated:**
   - This statement is incorrect. Only the root volume is terminated by default, while secondary EBS volumes remain available unless explicitly configured to be deleted.

Conclusion / Points to memorize:

1. **Root Volume Termination:**
   - By default, the root EBS volume of an EC2 instance is deleted upon instance termination.
2. **Retention Configuration:**
   - Users can modify the "Delete on Termination" attribute to retain the root volume after instance termination if required.
3. **Secondary Volumes:**
   - Non-root EBS volumes remain available by default after instance termination unless configured otherwise.
4. **EBS Independence:**
   - EBS volumes are designed to be independent of the lifecycle of EC2 instances, providing persistent storage.

Question 45 (Correct)

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?

Correct option: Use AWS transit gateway to interconnect the VPCs

Explanation behind correct option:

- **AWS Transit Gateway**:
  - Acts as a network transit hub to interconnect multiple VPCs and on-premises networks.
  - Supports dynamic and static routing, simplifying the network architecture.
  - Provides a scalable and efficient solution to interconnect VPCs.

- Allows centralized management and routing for connected VPCs and on-premises networks.
- Handles up to thousands of VPC connections, making it ideal for large-scale environments.

Incorrect options:

- **Establish VPC peering connections between all VPCs**:
  - Requires creating a complex mesh network with many peering connections.
  - Not scalable for a large number of VPCs.
  - Each VPC needs a direct peering connection, which does not support transitive routing.
- **Use an internet gateway to interconnect the VPCs**:
  - Internet gateways are used for enabling internet access from VPCs, not for interconnecting VPCs internally.
  - Not suitable for VPC-to-VPC communication within a private network.
- **Use a VPC endpoint to interconnect the VPCs**:
  - VPC endpoints are used to privately connect VPCs to supported AWS services without using the internet.
  - Not intended for interconnecting multiple VPCs.

Conclusion / Points to memorize:

- **AWS Transit Gateway**:
  - Ideal for interconnecting multiple VPCs and on-premises networks.
  - Supports high scalability and efficient routing.
  - Simplifies network management and reduces the complexity of VPC interconnections.
- **VPC Peering**:
  - Limited to direct connections between VPCs.
  - Does not support transitive routing.
- **Internet Gateway and VPC Endpoints**:
  - Not suitable for VPC interconnection.
    - Used for internet access and private connections to AWS services, respectively.

Question 46 (Incorrect)

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 5 petabytes 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?

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

Explanation behind correct option:

- **AWS Snowball Edge Storage Optimized**:
  - Ideal for securely and quickly transferring large amounts of data (dozens of terabytes to petabytes) to AWS.
  - Provides up to 80 TB of usable HDD storage, 40 vCPUs, 1 TB of SATA SSD storage, and up to 40 Gb network connectivity.
  - After transferring data to Amazon S3, you can create a lifecycle policy to transition the data into Amazon S3 Glacier for long-term, cost-effective storage.
  - Directly copying data from AWS Snowball Edge to Amazon S3 Glacier is not possible, so this method ensures data reaches Glacier via S3.

Incorrect options:

- **Transfer the on-premises data into multiple AWS Snowball Edge Storage Optimized devices. Copy the AWS Snowball Edge data into Amazon S3 Glacier**:
  - Direct transfer from AWS Snowball Edge to Amazon S3 Glacier is not supported. Data must first be copied into Amazon S3, then transitioned to Glacier.
- **Setup AWS Direct Connect between the on-premises data center and AWS Cloud. Use this connection to transfer the data into Amazon S3 Glacier**:
  - AWS Direct Connect provides a dedicated network connection to AWS but involves significant monetary investment and setup time (more than a month).
  - Not suitable for a one-time large-scale data transfer.
- **Setup AWS Site-to-Site VPN connection between the on-premises data center and AWS Cloud. Use this connection to transfer the data into Amazon S3 Glacier**:

- AWS Site-to-Site VPN is suitable for low to modest bandwidth requirements.
- Given the high data volume (5 petabytes), VPN connections would not be efficient for this use case.

Conclusion / Points to memorize:

- **AWS Snowball Edge**:
  - Ideal for large-scale, secure, and quick data transfer to AWS.
  - Use it to transfer data to Amazon S3 and then transition to Amazon S3 Glacier for cost-effective long-term storage.
- **AWS Direct Connect and Site-to-Site VPN**:
  - Direct Connect is costly and time-consuming to set up, making it unsuitable for one-time transfers.
  - Site-to-Site VPN is not efficient for very large data volumes due to bandwidth limitations.
- **Data Transfer Flow**:
  - Transfer data from Snowball Edge to S3 first.
    - Use lifecycle policies to transition data from S3 to Glacier for long-term storage.

Question 47 (Incorrect)

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?

Correct option: Create an Application Load Balancer

Explanation behind correct option:

- **Application Load Balancer (ALB)**:
  - Supports host-based routing, allowing routing based on the Host field of the HTTP header (e.g., checkout.mycorp.com).
  - Supports path-based routing, allowing routing based on the URL path of the HTTP header (e.g., mycorp.com/profile).
  - Enables multiple routing mechanisms such as HTTP header-based, HTTP method-based, query string parameter-based, and source IP address CIDR-based routing.
  - Designed to handle complex routing rules with minimal configuration effort.
  - Ideal for microservices architecture where different services need to be routed based on specific URL patterns.

Incorrect options:

- **Create an NGINX based load balancer on an Amazon EC2 instance to have advanced routing capabilities**:
  - Although technically possible, this option requires significant configuration effort and management.
  - Involves manual setup, scaling, and maintenance, making it less efficient compared to using ALB.
- **Create a Network Load Balancer**:
  - Best suited for low latency and high throughput workloads operating at the connection level (Layer 4).
  - Does not support advanced routing features like host-based and path-based routing required for this use case.
- **Create a Classic Load Balancer**:
  - Provides basic load balancing across multiple Amazon EC2 instances.
  - Operates at both the request level and connection level but lacks the advanced routing features of ALB.
  - Intended for older applications built within the EC2-Classic network.

Conclusion / Points to memorize:

- **Application Load Balancer (ALB)**:
  - Supports advanced routing mechanisms such as host-based and path-based routing.
  - Ideal for microservices architecture and complex routing requirements.
  - Requires minimal configuration and management effort compared to custom solutions.
- **Network Load Balancer and Classic Load Balancer**:
  - Not suitable for scenarios requiring advanced routing capabilities.
  - Better suited for simpler load balancing needs without complex routing rules.
- **NGINX-based Load Balancer**:
  - Possible but requires extensive setup and maintenance.
    - Not recommended for minimizing configuration and development effort.

Question 48 (Incorrect)

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)

Correct options:

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

Explanation behind correct options:

- **http://bucket-name.s3-website.Region.amazonaws.com**:
  - This format is used for Amazon S3 website endpoints where "bucket-name" is the name of your S3 bucket, "Region" is the AWS region where the bucket is hosted.
  - Example: If your bucket is named "mybucket" and it's in the "us-east-1" region, the URL would be http://mybucket.s3-website.us-east-1.amazonaws.com.
- **http://bucket-name.s3-website-Region.amazonaws.com**:
  - This format is another valid way for Amazon S3 website endpoints where "bucket-name" is the name of your S3 bucket and "Region" is the AWS region.
  - Example: If your bucket is named "mybucket" and it's in the "eu-west-1" region, the URL would be http://mybucket.s3-website-eu-west-1.amazonaws.com.

Incorrect options along with brief explanation, why they are incorrect:

- **http://s3-website-Region.bucket-name.amazonaws.com**:
  - This format places the region before the bucket name, which is incorrect for S3 website endpoints.
- **http://s3-website.Region.bucket-name.amazonaws.com**:
  - Similar to the previous option, this format incorrectly places the region and bucket name in the URL structure.
- **http://bucket-name.Region.s3-website.amazonaws.com**:
  - This format incorrectly places the region and S3 website service name in the URL structure.

Conclusion / points to memorize:

- For Amazon S3 static website endpoints, the correct formats are:
  - http://bucket-name.s3-website.Region.amazonaws.com
  - http://bucket-name.s3-website-Region.amazonaws.com
- Ensure the bucket name and region are correctly placed in the URL.
- Incorrect URL formats include:
  - http://s3-website-Region.bucket-name.amazonaws.com
  - http://s3-website.Region.bucket-name.amazonaws.com
  - http://bucket-name.Region.s3-website.amazonaws.com

Question 49 (Incorrect)

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?

Correct option:

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

Explanation behind correct option:

- **http://169.254.169.254/latest/meta-data/public-ipv4**:
  - This URL path is used to retrieve the public IP of an EC2 instance from the instance metadata.
  - Instance metadata is the data about your instance that you can use to configure or manage the running instance.
  - Example: If you need to get the public IP address, you would use curl http://169.254.169.254/latest/meta-data/public-ipv4 from within the EC2 instance.

Incorrect options along with brief explanation, why they are incorrect:

- **http://169.254.169.254/latest/user-data/public-ipv4**:
  - This URL path is incorrect because user data is data that you specified in the form of a configuration script while launching your instance, not for retrieving metadata like the public IP.
- **http://254.169.254.169/latest/meta-data/public-ipv4**:

- This URL path is incorrect because the IP address is wrong. The correct IP address for accessing instance metadata is 169.254.169.254, not 254.169.254.169.

- **http://254.169.254.169/latest/user-data/public-ipv4**:
  - This URL path is incorrect for the same reasons as above: incorrect IP address and user data is not used for retrieving the public IP.

Conclusion / points to memorize:

- To retrieve instance metadata, including the public IP, use the base URL http://169.254.169.254/latest/meta-data/.
- The specific URL path for retrieving the public IP is http://169.254.169.254/latest/meta-data/public-ipv4.
- User data is accessed using the URL http://169.254.169.254/latest/user-data/.
  - Ensure the correct IP address (169.254.169.254) and the correct path are used when querying instance metadata.

Question 50 (Correct)

A company wants to ensure high availability for its Amazon 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?

Correct option:

- The CNAME record will be updated to point to the standby database

Explanation behind correct option:

- **The CNAME record will be updated to point to the standby database**:
  - Amazon RDS Multi-AZ deployments ensure high availability by maintaining a synchronous standby replica in a different Availability Zone.
  - During a failure of the primary DB instance, Amazon RDS automatically handles the failover process without manual intervention.
  - The failover involves flipping the canonical name record (CNAME) for the DB instance to point to the standby instance, which is then promoted to become the new primary.
  - This process ensures minimal downtime and seamless transition, with the URL to access the database remaining unchanged.

Incorrect options along with brief explanation, why they are incorrect:

- **The URL to access the database will change to the standby database**:
  - Incorrect because the URL remains the same. The CNAME update ensures that the same URL points to the new primary instance.
- **An email will be sent to the System Administrator asking for manual intervention**:
  - Incorrect as the failover process in Amazon RDS Multi-AZ deployments is automatic and does not require manual intervention.
- **The application will be down until the primary database has recovered itself**:
  - Incorrect because the standby instance takes over immediately, minimizing downtime and ensuring high availability.

Conclusion / points to memorize:

- Amazon RDS Multi-AZ deployments provide high availability by automatically failing over to a standby instance in a different Availability Zone.
- The failover process involves updating the CNAME to point to the new primary instance, ensuring the database URL remains unchanged.
- The failover process is automatic, requiring no manual intervention.
  - Amazon RDS ensures minimal downtime and seamless transition during primary instance failures.

Question 51 (Correct)

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 AWS Multi-Factor Authentication (AWS MFA) to have maximum security. Ideally, the solution should be fully managed by AWS. Which technology do you recommend for managing the users' accounts?

Correct option:

- Amazon Cognito

Explanation behind correct option:

- **Amazon Cognito**:
  - Amazon Cognito enables user sign-up, sign-in, and access control to web and mobile apps quickly and easily.
  - It scales to millions of users and supports sign-in with social identity providers, such as Google, Facebook, and Amazon, and enterprise identity providers via SAML 2.0.
  - Amazon Cognito also provides advanced security features like Multi-Factor Authentication (MFA) and adaptive authentication to protect user accounts.

Incorrect options along with brief explanation, why they are incorrect:

- **Write an AWS Lambda function with Auth0 3rd party integration**:

- Using AWS Lambda and Auth0 would require additional code maintenance and management for user authentication and is not a fully managed AWS service.

- **AWS Identity and Access Management (AWS IAM)**:
  - IAM is used to manage access to AWS services and resources securely, but it is not designed to manage mobile app user accounts and authentication.
- **Enable the AWS Google Login Service**:
  - There is no such service as "AWS Google Login Service." This option is a distractor and does not exist within AWS services.

Conclusion / points to memorize:

- Amazon Cognito is the recommended AWS service for managing user accounts with features like social identity provider support and MFA.
- IAM is used for managing access to AWS resources, not for mobile app user accounts.
  - Always prefer fully managed AWS services to reduce the maintenance overhead and leverage built-in security features.

Question 52 (Correct)

Computer vision researchers at a university are trying to optimize the I/O bound processes for a proprietary algorithm running on Amazon 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?

Correct Option:

- Use Amazon EC2 instances with Instance Store as the storage option

Explanation behind correct option:

- **Instance Store:**
  - Provides temporary block-level storage for EC2 instances.
  - Located on disks physically attached to the host computer.
  - Ideal for temporary storage of information that changes frequently.
  - High random I/O performance using NVMe or SATA-based SSDs.
  - Included as part of the instance's usage cost, making it cost-optimal.

Incorrect options along with brief explanation:

- **Use Amazon EC2 instances with Amazon EBS General Purpose SSD (gp2) as the storage option:**
  - Offers cost-effective storage with single-digit millisecond latencies and ability to burst to 3,000 IOPS.
  - Persistent storage and costlier than Instance Store as it adds additional storage volume costs to the EC2 instance.
- **Use Amazon EC2 instances with Amazon EBS Provisioned IOPS SSD (io1) as the storage option:**
  - Designed for I/O-intensive workloads with consistent IOPS rate.
  - Persistent storage and more expensive than Instance Store due to additional storage volume costs.
- **Use Amazon EC2 instances with Amazon EBS Throughput Optimized HDD (st1) as the storage option:**
  - Low-cost HDD volumes designed for throughput-intensive workloads.
  - Persistent storage and costlier than Instance Store due to additional storage volume costs.

Conclusion / Points to memorize:

- **Instance Store** is the best option for high-performance, temporary storage needs.
- Instance Store is included in the instance's usage cost, making it more cost-effective.
  - EBS options (gp2, io1, st1) are persistent and add extra costs, hence not the most cost-optimal for temporary storage.

Question 53 - Correct

A big data analytics company is using Amazon 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?

Correct answer

Use Enhanced Fanout feature of Amazon Kinesis Data Streams

Explanation behind correct options:

- **Enhanced Fanout feature of Amazon Kinesis Data Streams:**
  - KDS is designed for real-time data streaming, capable of handling gigabytes of data per second.
  - By default, the 2MB/second/shard output is shared between all applications consuming data from the stream.

- Enhanced fan-out allows multiple consumers to retrieve data in parallel, each receiving their own 2MB/second pipe of read throughput per shard.
- This scales automatically with the number of shards, improving data delivery speed without contention among consumers.

Incorrect options along with brief explanation, why they are incorrect:

- **Swap out Amazon Kinesis Data Streams with Amazon Kinesis Data Firehose:**
  - Amazon Kinesis Data Firehose is used to load streaming data into data lakes, data stores, and analytics tools, not for direct consumption by multiple applications.
  - It only writes to Amazon S3, Amazon Redshift, Amazon Elasticsearch, or Splunk, which doesn't fit the need for multiple applications consuming data streams directly.
- **Swap out Amazon Kinesis Data Streams with Amazon SQS Standard queues:**
  - Amazon SQS is a message queuing service designed for decoupling and scaling microservices, distributed systems, and serverless applications.
  - SQS does not support multiple applications consuming the same stream concurrently as efficiently as KDS with enhanced fan-out.
- **Swap out Amazon Kinesis Data Streams with Amazon SQS FIFO queues:**
  - Amazon SQS FIFO queues ensure messages are processed exactly once and in the exact order sent, which is not necessary for the given use-case.
  - SQS FIFO queues do not provide the same level of concurrent consumption performance as KDS with enhanced fan-out.

Conclusion / points to memorize:

- **Enhanced fan-out in KDS:** Allows multiple applications to consume the same stream concurrently, each with dedicated throughput.
- **Use cases for KDS:** Ideal for real-time data streaming and multiple consumers.
- **Amazon Kinesis Data Firehose:** Designed for data ingestion into data lakes and analytics tools, not for multiple consumer applications.
  - **Amazon SQS:** Best for message queuing and microservices decoupling, not for concurrent data stream consumption.

Question 54 - Incorrect

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?

- **Correct answer**
  - 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

Explanation behind correct option:

- **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**:
    - AWS DataSync is designed for automating and accelerating the process of copying large amounts of data to and from AWS storage services.
    - It can handle up to 10 times faster data transfer compared to traditional command-line tools.
    - It supports seamless and secure integration with Amazon S3, Amazon EFS, Amazon FSx for Windows File Server.
    - It uses a purpose-built network protocol and scale-out architecture for efficient data transfer.
    - It performs data integrity verification during and after the transfer.
  - **File Gateway**:
    - File Gateway provides on-premises applications with access to data stored in Amazon S3 through standard file protocols (SMB or NFS).
    - It offers local caching to provide low-latency access to frequently accessed data.
    - File Gateway allows ongoing updates from on-premises applications to be synchronized with Amazon S3.

Incorrect options along with brief explanation:

- **Use AWS DataSync to migrate existing data to Amazon S3 as well as access the Amazon S3 data for ongoing updates**
  - AWS DataSync is primarily designed for data transfer and not for ongoing updates or accessing data. File Gateway is more suitable for ongoing updates as it provides local caching and standard file protocol access.
- **Use File Gateway configuration of AWS Storage Gateway to migrate data to Amazon S3 and then use Amazon S3 Transfer Acceleration (Amazon S3TA) for ongoing updates from the on-premises applications**
  - File Gateway is not optimized for high-volume data migration. AWS DataSync is better suited for initial data migration due to its high transfer speeds. Amazon S3 Transfer Acceleration is designed for fast upload but not for facilitating ongoing updates.

- **Use Amazon S3 Transfer Acceleration (Amazon S3TA) to migrate existing data to Amazon S3 and then use AWS DataSync for ongoing updates from the on-premises applications**
    - Amazon S3 Transfer Acceleration is suitable for increasing upload speed to Amazon S3 but does not facilitate ongoing updates. AWS DataSync is not designed for continuous updates after initial migration; File Gateway is better for ongoing updates.

Conclusion / points to memorize:

- **AWS DataSync** is optimal for migrating large datasets to AWS storage services, providing fast and secure data transfer.
- **File Gateway** is suitable for providing on-premises applications with ongoing access to data stored in Amazon S3, with local caching for low-latency access.
- **Amazon S3 Transfer Acceleration** increases upload speed but is not suitable for facilitating ongoing updates.
    - For large-scale data migration and ongoing updates, a combination of **AWS DataSync** for migration and **File Gateway** for updates is recommended.

Question 55 - Incorrect

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.

- **Your answer is incorrect**
    - Amazon Relational Database Service (Amazon RDS)
- **Correct answer**
    - Amazon DynamoDB

Explanation behind correct option:

- **Amazon DynamoDB**
    - **Scalability**: DynamoDB is horizontally scalable and can handle more than 10 trillion requests per day and support peaks of more than 20 million requests per second.
    - **Auto Scaling**: DynamoDB provides automatic scaling of throughput capacity using Read Capacity Units (RCU) and Write Capacity Units (WCU).
    - **High Availability**: It is a multi-Region, multi-master database with built-in durability and security.
    - **Flexible Schema**: DynamoDB can handle changes in data attributes over time, making it suitable for IoT use cases where data attributes may change frequently.
    - **DynamoDB Streams**: This feature captures a continuous stream of item-level changes in a DynamoDB table and provides an ordered flow of information about changes to items.

Incorrect options along with brief explanation:

- **Amazon Relational Database Service (Amazon RDS)**
    - RDS provides managed relational databases but is less suited for scenarios where the schema changes frequently, as relational databases require a more rigid schema.
    - While RDS supports multi-AZ deployments for high availability, it does not offer the same level of scalability or flexibility for changing data attributes as DynamoDB.
- **Amazon Aurora**
    - Aurora is a MySQL and PostgreSQL-compatible relational database built for the cloud.
    - It features high availability and scalability, but like RDS, it is less suited for frequently changing schema requirements.
    - Aurora is not an in-memory database, and schema changes in relational databases can be complex and costly to maintain.
- **Amazon Redshift**
    - Redshift is a fully managed petabyte-scale data warehouse designed for large-scale data storage and analysis.
    - It is not intended for use as a general-purpose database or for handling real-time data from IoT devices.
    - Redshift does not provide the same level of flexibility or support for auto-scaling and schema changes as DynamoDB.

Conclusion / points to memorize:

- **Amazon DynamoDB** is the best choice for IoT applications needing a flexible, scalable, and highly available database solution.
    - It supports auto-scaling, high throughput, and continuous streams for real-time data changes.
- **Relational databases** like RDS and Aurora are less suited for use cases with frequent schema changes and may involve more complex management.
- **Amazon Redshift** is designed for data warehousing and is not appropriate for real-time IoT data processing.
    - **DynamoDB Streams** provide continuous data change capture, essential for applications requiring real-time data updates and monitoring.

Question 56 - Correct

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.

- **Your answer is correct**
  - AWS Trusted Advisor

Correct option:

- **AWS Trusted Advisor**
  - **Description**: 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. It inspects your AWS environment and makes recommendations for saving money, improving system performance, or closing security gaps.
  - **Features**:
    - **Cost Optimization**: Provides recommendations to reduce costs by identifying idle and underutilized resources.
    - **Performance**: Suggests ways to improve the performance of your AWS resources.
    - **Security**: Identifies security settings that could make your AWS solution less secure.
    - **Fault Tolerance**: Highlights best practices to increase the resilience of your AWS resources.
    - **Service Limits**: Monitors your usage against the service limits and provides recommendations to avoid exceeding them.
  - **Use Case**: The service is ideal for optimizing AWS resources across multiple countries and regions by providing best practices for cost optimization, performance, and security.

Incorrect options along with brief explanation:

- **AWS Config**
  - **Description**: AWS Config enables you to assess, audit, and evaluate the configurations of your AWS resources. It provides detailed resource configuration histories and determines compliance against configurations specified in your internal guidelines.
  - **Reason for Incorrectness**: It does not offer feedback on architectural best practices or optimizations.
- **AWS Management Console**
  - **Description**: The AWS Management Console is a web application that comprises and refers to a broad collection of service consoles for managing Amazon Web Services.
  - **Reason for Incorrectness**: It does not offer feedback or recommendations about architectural best practices or optimizations.
- **AWS Systems Manager**
  - **Description**: AWS Systems Manager is an AWS service that you can use to view and control your infrastructure on AWS. It allows you to view operational data from multiple AWS services and automate operational tasks.
  - **Reason for Incorrectness**: While it provides operational control and management, it does not offer specific feedback on architectural best practices for cost optimization, performance, or security.

Conclusion / points to memorize:

- **AWS Trusted Advisor** is the go-to service for recommendations on cost optimization, performance, security, fault tolerance, and service limits.
- **AWS Config** is useful for compliance and configuration management but not for optimization recommendations.
- **AWS Management Console** provides access to AWS services but does not offer best practices advice.
- **AWS Systems Manager** helps in operational management but does not provide optimization recommendations.

By using AWS Trusted Advisor, companies can ensure they are following AWS best practices to optimize their resource usage, enhance performance, improve security, and manage costs effectively across their AWS environment.

Question 57 - Correct

An e-commerce company uses Amazon Simple Queue Service (Amazon SQS) queues to decouple their application architecture. The engineering team has observed message processing failures for some customer orders.

- **Your answer is correct**
  - Use a dead-letter queue to handle message processing failures

Correct option:

- **Use a dead-letter queue to handle message processing failures**
  - **Description**: Dead-letter queues can be used by other queues (source queues) as a target for messages that can't be processed (consumed) successfully. They are useful for debugging your application or messaging system because they let you isolate problematic messages to determine why their processing doesn't succeed.

- **How it works**:
  - **Redrive Policy**: Specifies the source queue, the dead-letter queue, and the conditions under which Amazon SQS moves messages from the source to the dead-letter queue.
  - **ReceiveCount**: When the ReceiveCount for a message exceeds the maxReceiveCount for a queue, Amazon SQS moves the message to a dead-letter queue.
  - **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.
- **Use Case**: Suitable for handling message processing failures by isolating and debugging failed messages.

Incorrect options along with brief explanation:

- **Use a temporary queue to handle message processing failures**
  - **Description**: Temporary queues are commonly used for request-response messaging patterns (e.g., processing a login request) where a requester creates a temporary queue for receiving each response message.
  - **Reason for Incorrectness**: Temporary queues are not designed for handling message processing failures.
- **Use short polling to handle message processing failures**
  - **Description**: Short polling sends the response right away, even if no messages are found.
  - **Reason for Incorrectness**: Short polling cannot handle message processing failures; it only affects how quickly messages are received.
- **Use long polling to handle message processing failures**
  - **Description**: Long polling sends a response after collecting at least one available message, up to the maximum number of messages specified in the request.
  - **Reason for Incorrectness**: Long polling cannot handle message processing failures; it only affects how quickly messages are received.

Conclusion / points to memorize:

- **Dead-letter queues** are used to handle message processing failures by isolating problematic messages.
- **Redrive policy** specifies the conditions under which messages are moved to a dead-letter queue.
- **ReceiveCount and maxReceiveCount**: When ReceiveCount exceeds maxReceiveCount, the message is moved to the dead-letter queue.
- **Temporary queues** are for request-response messaging, not for handling failures.
- **Short polling** sends immediate responses, while **long polling** waits for messages but neither can handle processing failures.

By using dead-letter queues, e-commerce companies can effectively handle and debug message processing failures, ensuring smoother operation of their application architecture.

Question 58 - Incorrect

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 Amazon 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.

- **Correct answer**
  - Use Amazon Redshift Spectrum to create Amazon Redshift cluster tables pointing to the underlying historical data in Amazon S3. The analytics team can then query this historical data to cross-reference with the daily reports from Redshift
- **Your answer is incorrect**
  - Use the Amazon Redshift COPY command to load the Amazon S3 based historical data into Amazon Redshift. Once the ad-hoc queries are run for the historic data, it can be removed from Amazon Redshift

Correct option:

- **Use Amazon Redshift Spectrum to create Amazon Redshift cluster tables pointing to the underlying historical data in Amazon S3. The analytics team can then query this historical data to cross-reference with the daily reports from Redshift**
  - **Description**: Amazon Redshift Spectrum allows you to run queries against exabytes of data in S3 without having to load the data into Redshift. Redshift Spectrum extends the capabilities of Redshift by allowing it to query data directly in S3, thereby enabling the ability to cross-reference historical data stored in S3 with the daily reports stored in Redshift.
  - **Benefits**:
    - No need to move large volumes of historical data into Redshift, saving on storage costs.
    - Enables seamless querying across both recent and historical data.
    - Utilizes the Redshift cluster's compute resources efficiently by pushing down compute-intensive tasks to Redshift Spectrum.

Incorrect options along with brief explanation:

- **Setup access to the historical data via Amazon Athena. The analytics team can run historical data queries on Amazon Athena and continue the daily reporting on Amazon 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**
  - **Reason for Incorrectness**: This option is cumbersome and inefficient because it requires exporting and merging data from two different systems (Redshift and Athena). This manual process adds complexity and increases the potential for errors.
- **Use the Amazon Redshift COPY command to load the Amazon S3 based historical data into Amazon Redshift. Once the ad-hoc queries are run for the historic data, it can be removed from Amazon Redshift**
  - **Reason for Incorrectness**: Loading and unloading large volumes of historical data repeatedly is cost-inefficient and time-consuming. This approach also negates the benefits of offloading historical data to S3.
- **Use AWS Glue ETL job to load the Amazon S3 based historical data into Redshift. Once the ad-hoc queries are run for the historic data, it can be removed from Redshift**
  - **Reason for Incorrectness**: Similar to the COPY command, using AWS Glue ETL jobs to load and then remove historical data is not cost-effective for one-time ad-hoc processes. The process is also more complex compared to using Redshift Spectrum.

Conclusion / points to memorize:

- **Amazon Redshift Spectrum**:
  - Enables querying data stored directly in Amazon S3.
  - Ideal for cost-effective access to historical data without the need to load it into Redshift.
  - Allows seamless integration of current and historical data for comprehensive analysis.
- **Amazon Athena**:
  - Suitable for ad-hoc querying directly in S3 but not ideal for seamless integration with Redshift data.
- **COPY command / AWS Glue ETL**:
  - Both methods are more costly and complex for handling large volumes of historical data compared to using Redshift Spectrum.

By using Amazon Redshift Spectrum, companies can efficiently and cost-effectively manage and analyze both current and historical data without the need for extensive data movement or integration processes.

Question 59 - Correct

A gaming company is doing pre-launch testing for its new product. The company runs its production database on an Aurora MySQL DB cluster and the performance testing team wants access to multiple test databases that must be re-created from production data. The company has hired you as an AWS Certified Solutions Architect - Associate to deploy a solution to create these test databases quickly with the LEAST required effort. What would you suggest to address this use case?

Correct option:

- **Use database cloning to create multiple clones of the production database and use each clone as a test database**

Explanation behind correct option:

1. **Quick Cloning**:
   - Database cloning allows the creation of clones much faster than manual snapshots.
   - Uses a copy-on-write protocol, where data is copied only when it changes, making cloning efficient.
2. **Isolated Testing**:
   - Cloning the DB cluster allows the performance testing team to have isolated access to the production data.
   - Multiple clones can be created quickly for different test scenarios.
3. **Efficient Use of Resources**:
   - Clones must be created in the same region, ensuring low latency and high performance.
   - Allows up to 15 clones per source database copy, providing ample resources for testing.

Incorrect options and explanations:

1. **Enable database Backtracking on the production database and let the testing team use the production database**:
   - Backtracking can quickly rewind the DB cluster to a previous state.
   - However, it is not suitable for isolated testing as it uses the production database directly, which can impact live operations.
2. **Take a backup of the Aurora MySQL database instance using the mysqldump utility, create multiple new test database instances and restore each test database from the backup**:
   - This process is manual and time-consuming.
   - Requires several steps, including taking a backup, creating new instances, and restoring data, which does not meet the requirement for the least effort.

3. **Set up binlog replication in the Aurora MySQL database instance to create multiple new test database instances**:
    - Binlog replication involves multiple steps such as creating snapshots, loading snapshots into replicas, and managing replication configurations.
    - This approach is complex and not suitable for quickly creating multiple test databases.

Conclusion / Points to memorize:

- **Database Cloning**: Ideal for quickly creating multiple test databases from a production database with minimal effort.
- **Isolated Testing**: Ensures that testing does not impact the production environment.
- **Efficient Resource Utilization**: Clones are created within the same region and use resources efficiently.
    - **Copy-on-Write Protocol**: Only changes to data are copied, making the cloning process efficient and fast.

Question 60 - Correct

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 Amazon EC2 server farm. As a solutions architect, which type of Elastic Load Balancing would you recommend as part of the solution stack?

Correct option:

- **Network Load Balancer**

Explanation behind correct option:

1. **High Throughput**:
    - Network Load Balancer (NLB) is designed to handle millions of requests per second.
    - It operates at the connection level (Layer 4), making it suitable for high throughput workloads.
2. **Low Latency**:
    - NLB is optimized for low latency, ensuring minimal delay in handling requests.
3. **Scalability**:
    - Automatically scales to accommodate varying traffic loads, providing the necessary elasticity for a video streaming provider.
4. **TCP/UDP Traffic**:
    - Ideal for applications that require stable and consistent performance, as it can handle TCP and UDP traffic efficiently.
5. **Health Checks**:
    - Performs health checks on targets to ensure that traffic is routed only to healthy instances.

Incorrect options and explanations:

1. **Application Load Balancer**:
    - Operates at the request level (Layer 7), routing traffic based on the content of the request.
    - Best suited for HTTP and HTTPS traffic with advanced routing features, but not ideal for handling millions of requests per second in a low latency, high throughput scenario.
2. **Classic Load Balancer**:
    - Provides basic load balancing across multiple Amazon EC2 instances and operates at both the request and connection level.
    - Suitable for applications built within the EC2-Classic network, but not optimal for the high throughput and low latency requirements of the given use case.
3. **Infrastructure Load Balancer**:
    - There is no such service as an Infrastructure Load Balancer in AWS. This option is a distractor and does not apply to any AWS load balancing solutions.

Conclusion / Points to memorize:

- **Network Load Balancer**:
    - Best for high throughput and low latency requirements.
    - Operates at Layer 4, handling millions of requests per second.
    - Suitable for TCP and UDP traffic, ensuring stable and consistent performance.
- **Application Load Balancer**:
    - Operates at Layer 7, ideal for HTTP/HTTPS traffic with advanced routing.
    - Not suitable for extremely high throughput and low latency scenarios.
- **Classic Load Balancer**:
    - Basic load balancing for EC2-Classic network applications.
    - Not designed for modern high-performance requirements.
- **Infrastructure Load Balancer**:

- Non-existent, used as a distractor in the question.

Question 61 - Correct

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?

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.**

Explanation behind correct option:

1. **Immutable Launch Configurations**:
   - Launch configurations in Auto Scaling cannot be modified once created. If you need to change any configuration, you must create a new launch configuration.
2. **Creating a New Launch Configuration**:
   - To fix the issue with the incorrect instance type, you need to create a new launch configuration with the correct instance type that is optimized for the application workflow.
3. **Updating the Auto Scaling Group**:
   - Modify the existing Auto Scaling group to use the new launch configuration. This ensures that any new instances launched by the Auto Scaling group will use the correct instance type.
4. **Cleanup**:
   - Once the Auto Scaling group is updated to use the new launch configuration, the old launch configuration can be deleted as it is no longer needed.

Incorrect options and explanations:

1. **Modify the launch configuration to use the correct instance type and continue to use the existing Auto Scaling group**:
   - Launch configurations cannot be modified after they are created. Therefore, this option is not feasible.
2. **No need to modify the launch configuration. Just modify the Auto Scaling group to use the correct instance type**:
   - The instance type is specified in the launch configuration, not directly in the Auto Scaling group. Therefore, this option is not feasible.
3. **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**:
   - Increasing the number of instances does not address the root cause of the performance issue, which is the incorrect instance type. This is not an effective long-term solution.

Conclusion / Points to memorize:

- **Launch Configuration Immutability**:
  - Once created, a launch configuration cannot be modified.
  - To change instance configurations, you must create a new launch configuration.
- **Auto Scaling Group Update**:
  - Update the Auto Scaling group to use the new launch configuration for any new instances.
- **Proper Cleanup**:
  - After updating the Auto Scaling group, delete the old launch configuration to avoid confusion and ensure proper resource management.
- **Performance Optimization**:
  - Ensure the correct instance type is selected in the launch configuration to match the application workload requirements.

Question 62 - Correct

A company's cloud architect has set up a solution that uses Amazon 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?

Correct option:

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

Explanation behind correct option:

1. **Active-Passive Failover Configuration**:

- Use active-passive failover configuration when a primary resource should be available most of the time, with a secondary resource on standby in case of failure.
- Route 53 will route traffic to the primary resource if it is healthy. If the primary resource becomes unhealthy, Route 53 will switch to the secondary resource (static error page on S3 in this case).

2. **Health Checks**:
   - Route 53 health checks can monitor the ALB endpoint. If the ALB is determined to be unhealthy, the routing policy will failover to the secondary resource.
3. **Minimal Changes**:
   - This setup requires minimal changes, just configuring a new failover routing policy and an S3 bucket to host the static error page.

Incorrect options and explanations:

1. **Set up Amazon Route 53 active-active type of failover routing policy. If Amazon Route 53 health check determines the Application Load Balancer endpoint as unhealthy, the traffic will be diverted to a static error page, hosted on Amazon S3 bucket**:
   - There is no such thing as an active-active failover routing policy in Route 53. Active-active failover is configured using other routing policies combined, and it is not suitable for this use case.
2. **Use Amazon Route 53 Latency-based routing. Create a latency record to point to the Amazon S3 bucket that holds the error page to be displayed**:
   - Latency-based routing directs traffic based on the lowest latency and is useful for multi-region setups. It does not provide failover capabilities required for this use case.
3. **Use Amazon 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 allows distribution of traffic based on weights assigned to resources, useful for load balancing and testing. It does not automatically route traffic to a secondary resource based on health checks.

Conclusion / Points to memorize:

- **Active-Passive Failover**:
  - Route 53's active-passive failover routing policy is ideal for ensuring high availability by diverting traffic to a secondary resource (like a static S3 error page) in case the primary resource (like an ALB) becomes unhealthy.
- **Health Checks**:
  - Utilize Route 53 health checks to monitor primary resources and trigger failover to the secondary resource when necessary.
- **Failover Routing Policy**:
  - Ensure to use failover routing policy for scenarios requiring automatic traffic redirection in case of resource failure.
- **Minimal Configuration**:
  - Aim for minimal configuration changes to achieve effective failover setups.

Question 63 - Incorrect

A streaming solutions company is building a video streaming product by using an Application Load Balancer (ALB) that routes the requests to the underlying Amazon EC2 instances. The engineering team has noticed a peculiar pattern. The Application Load Balancer removes an instance from its pool of healthy instances 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?

Correct option:

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

Explanation behind correct option:

1. **Health Check Types**:
   - Auto Scaling groups (ASG) can use either EC2 status checks or ELB health checks to determine the health of instances.
   - Application Load Balancers (ALB) use their own built-in health checks to determine the health of instances.
2. **Mismatch in Health Checks**:
   - If the ASG uses EC2 status checks, it only verifies if the instance is running and reachable from the AWS infrastructure perspective.
   - If the ALB uses its own health checks, it performs application-level health checks (e.g., HTTP status code, response content) to verify the instance's ability to serve traffic correctly.
3. **Scenario Analysis**:
   - When there is a mismatch, where the ASG relies on EC2 status checks and the ALB uses its own health checks, an instance could be considered healthy by the ASG but unhealthy by the ALB.
   - The ALB removes the instance from its routing pool because its health checks fail.

- However, the ASG does not replace the instance since its EC2 status checks are still passing, leading to the anomaly where unhealthy instances are not being replaced.

Incorrect options and explanations:

1. **The Auto Scaling group is using ALB based health check and the Application Load Balancer is using Amazon EC2 based health check**:
   - Application Load Balancer cannot use EC2-based health checks. This option is incorrect.
2. **Both the Auto Scaling group and Application Load Balancer are using ALB based health check**:
   - If both the ASG and ALB use ALB health checks, they will have a consistent view of instance health, and this situation will not occur.
3. **Both the Auto Scaling group and Application Load Balancer are using Amazon EC2 based health check**:
   - Application Load Balancer does not use EC2-based health checks. This option is incorrect.

Conclusion / Points to Memorize:

- **Health Check Consistency**:
  - Ensure consistency in health check mechanisms between Auto Scaling groups and Load Balancers.
  - Prefer using ALB-based health checks for both the ASG and ALB for better consistency and reliability.
- **ALB and ASG Integration**:
  - Understand how ALB health checks work and how they integrate with ASG.
  - Set up health checks that reflect the actual health and readiness of the application to handle traffic.
- **Common Mismatches**:
  - Be aware that mismatches between EC2 and ALB health checks can lead to issues where instances are not correctly replaced by ASG.
    - Always verify health check settings to avoid anomalies in scaling and instance management.

Question 64 - Incorrect

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 (AWS WAF) to handle this requirement. Can you identify the correct solution leveraging the capabilities of AWS WAF?

Correct option:

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

Explanation behind correct option:

1. **AWS WAF Integration**:
   - AWS WAF can be deployed on Amazon CloudFront, Application Load Balancer (ALB), and Amazon API Gateway. It cannot be configured directly on EC2 instances.
2. **Using Amazon CloudFront**:
   - When you use AWS WAF with Amazon CloudFront, you can protect your applications running on any HTTP web server, whether it's a web server that's running in Amazon Elastic Compute Cloud (Amazon EC2) or a web server that you manage privately.
   - CloudFront distributes the content to edge locations around the world, ensuring low latency and high performance. Using AWS WAF with CloudFront provides security at the edge locations, blocking malicious traffic before it reaches your application.
3. **Performance and Security**:
   - AWS WAF is tightly integrated with Amazon CloudFront, ensuring that your rules run in all AWS Edge Locations close to your end-users. This ensures that security measures do not compromise performance.
   - Blocked requests are stopped at the edge before they reach your web servers.

Incorrect options and explanations:

1. **Configure an Application Load Balancer (ALB) to balance the workload for all the Amazon EC2 instances. Configure Amazon CloudFront to distribute from an Application Load Balancer since AWS WAF cannot be directly configured on ALB. This configuration not only provides necessary safety but is scalable too**:
   - This statement is incorrect because AWS WAF can be directly configured on an ALB. It is not necessary to use CloudFront to distribute from an ALB to use AWS WAF.
2. **AWS WAF can be directly configured on Amazon EC2 instances for ensuring the security of the underlying application data**:
   - AWS WAF cannot be directly configured on Amazon EC2 instances. It can be configured on Amazon CloudFront, ALB, or API Gateway.
3. **AWS WAF can be directly configured only on an Application Load Balancer 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 partially correct but incomplete. AWS WAF can also be deployed on Amazon CloudFront, providing an additional option for security at the edge locations.

Conclusion / Points to Memorize:

- **AWS WAF Deployment**:
  - AWS WAF can be deployed on Amazon CloudFront, Application Load Balancer (ALB), and Amazon API Gateway.
  - AWS WAF cannot be directly configured on Amazon EC2 instances.
- **Best Practices**:
  - Use Amazon CloudFront with AWS WAF to protect applications and enhance performance by blocking malicious traffic at the edge locations.
  - Ensure the security architecture is scalable and provides low latency for end-users.
- **Integration**:
  - Understand the integration of AWS WAF with different AWS services and choose the appropriate configuration based on the specific use case and performance requirements.

Question 65 – Correct

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 Amazon 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?

Correct option/s:

The Auto Scaling group should be configured with the minimum capacity set to 4, with 2 instances each in two different Availability Zones. The maximum capacity of the Auto Scaling group should be set to 6.

Explanation behind correct option/s:

- Configure Auto Scaling group with minimum capacity to ensure redundancy and reliability.
- Distribute instances across multiple Availability Zones (AZs) for high availability.
- Amazon EC2 Auto Scaling enables geographic redundancy by spanning Auto Scaling groups across multiple AZs within a Region.
- Auto Scaling launches new instances in unaffected AZs when one becomes unhealthy, ensuring uninterrupted service.
- Auto Scaling attempts to distribute instances evenly between enabled AZs, hence requiring a minimum capacity of at least 2 instances per AZ.

Incorrect options:

- The Auto Scaling group should be configured with the minimum capacity set to 2, with 1 instance each in two different Availability Zones. The maximum capacity of the Auto Scaling group should be set to 6.
- The Auto Scaling group should be configured with the minimum capacity set to 2 and the maximum capacity set to 6 in a single Availability Zone.

Explanation: Both options lack redundancy and fail to ensure high availability.

Conclusion / points to memorize:

1. Set minimum capacity in Auto Scaling groups for redundancy and reliability.
2. Distribute instances across multiple AZs for high availability.
3. Auto Scaling ensures uninterrupted service by launching instances in unaffected AZs.
4. Maintain a minimum capacity of at least 2 instances per AZ for even distribution.
5. Auto Scaling groups cannot span multiple AWS Regions.