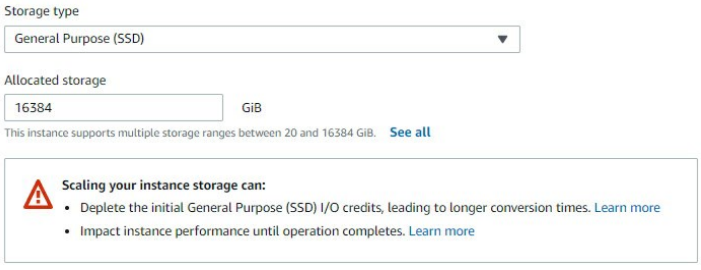
Answer Sheet

Q1) A retail company has an application that uses Amazon DocumentDB as its data store. A Database Specialist has been tasked to set up monitoring to ensure that all data definition language (DDL) statements performed, such as CREATE or ALTER commands, are visible to the Systems Administrator. The audit_logs parameter has been enabled in the cluster parameter group in order to comply with the company's audit policies.
What additional action must the Database Specialist do to automatically collect the database logs and share the data with the Administrator?

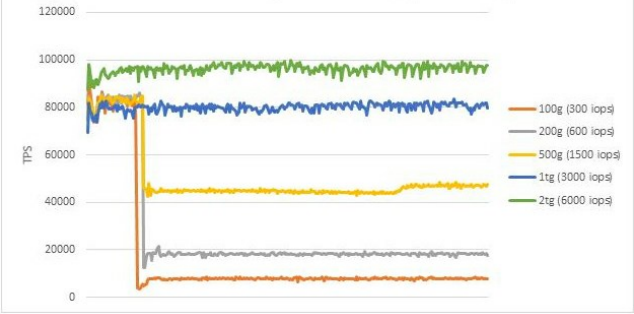
○ Enable Amazon DocumentDB to export logs to AWS CloudTrail. Instruct the Administrator to view the audit logs in the Event History page of the CloudTrail console.
Explanation:-This option is incorrect because You can't directly export the audit logs of your DocumentDB database to AWS CloudTrail. Take note that the auditing feature of DocumentDB is distinctly different from the service resource usage that is monitored with AWS CloudTrail. For example, CloudTrail is not capable of recording Data Definition Language (DDL) commands that were executed in your database.
○ Enable the Advanced Auditing configuration in Amazon DocumentDB and publish the audit logs to Amazon CloudWatch Logs.
Explanation:-This option is incorrect because Amazon DocumentDB doesn't have an Advanced Auditing feature. This is only available in Amazon Aurora.
✓ Enable Amazon DocumentDB Events to export logs to CloudWatch Logs. Instruct the Administrator to view the audit logs in CloudWatch.
Explanation:-With Amazon DocumentDB (with MongoDB compatibility), you can audit events that were performed in your cluster. Examples of logged events include successful and failed authentication attempts, dropping a collection in a database, or creating an index. By default, auditing is disabled on Amazon DocumentDB and requires that you opt-in to use this feature.
When auditing is enabled, Amazon DocumentDB records Data Definition Language (DDL), authentication, authorization, and user management events to Amazon CloudWatch Logs. When auditing is enabled, Amazon DocumentDB exports your cluster's auditing records (JSON documents) to Amazon CloudWatch Logs. You can use Amazon CloudWatch Logs to analyze, monitor, and archive your Amazon DocumentDB auditing events.
The Amazon DocumentDB auditing feature is distinctly different from the service resource usage that is monitored with AWS CloudTrail. CloudTrail records operations that are performed with the AWS Command Line Interface (AWS CLI) or AWS Management Console on resources like clusters, instances, parameter groups, and snapshots. Auditing of AWS resources with CloudTrail is on by default and cannot be disabled. The Amazon DocumentDB auditing feature is an opt-in feature. It records operations that take place within your cluster on objects, such as databases, collections, indexes, and users.
References:
https://docs.aws.amazon.com/documentdb/latest/developerguide/event-auditing.html
https://docs.aws.amazon.com/documentdb/latest/developerguide/managing-events.html
○ Create a shell script that runs the aws docdb download-db-log-file-portion AWS CLI command every hour. Configure the script to regularly send the export logs to the Administrator via email.
Explanation:-This option is incorrect because the download-db-log-file-portion CLI command is not available in Amazon DocumentDB. This command is only available in Amazon RDS. Furthermore, manually creating a scheduled job using a shell script is not an effective way to automate the notification process. A better solution is to simply configure Amazon DocumentsDB Events to automatically export the audit logs to CloudWatch, which the Administrator can easily monitor.

Q2) A Database Specialist manages an EBS-Optimized Amazon RDS for MariaDB instance with General Purpose SSD storage. The users recently raised a latency issue with the database performance, specifically during their new nightly batch job package that runs for three hours. The Specialist discovered that the database consistently utilizes the maximum available IOPS configured for the EBS volumes in the middle of the job. Furthermore, the Specialist confirms that both the IO throughput and free available storage space are at optimal levels.
Which of these actions can the Database Specialist do to increase the maximum IOPS?

○ Disable EBS optimization on the MariaDB DB instance.
Explanation:-This option is is incorrect becuase EBS optimization is designed to provide dedicated bandwidth. If EBS optimization was disabled, it could potentially degrade the database performance.
○ Modify the DB instance and enable Storage autoscaling.
Explanation:-This option is incorrect becuase Storage autoscaling does not consider IOPS or IO credits maximized when scaling automatically.
○ Change the underlying EBS storage type of the instance to Magnetic.
Explanation:-This option is incorrect becuase Magnetic EBS volumes are primarily used for infrequent data access and only provide a low IOPS. An alternative solution for the scenario is to use a Provisioned IOPS storage, not a Magnetic storage type.
✓ Increase the size allocated to the RDS DB instance storage.
Explanation:-
Amazon RDS volumes are built using Amazon EBS volumes, except for Amazon Aurora, which uses an SSD-backed virtualized storage layer purpose-built for database workloads. RDS currently supports both magnetic and SSD-based storage volume types. There are two supported Amazon EBS SSD-based storage types, Provisioned IOPS (called io1) and General Purpose (called gp2).
With io1, it's quite simple to predict IOPS because this is the value you provide when the volume is created. The gp2 storage type also has a base IOPS that is set when the volume is created. However, you do not provide a value for the IOPS directly—instead, IOPS is a function of the volume's size. The IOPS for a gp2 volume is the size of the volume in GiB x 3, with a minimum of 100 IOPS and a maximum of 10K IOPS. The gp2 volumes have a characteristic called burst mode. Baseline I/O performance for General Purpose SSD storage is 3 IOPS for each GiB, with a minimum of 100 IOPS. This relationship means that larger volumes have better performance. When your storage requires more than the base performance I/O level, it uses I/O credits in the I/O credit balance to burst to the level of performance needed. Such a burst goes to a maximum of 3,000 IOPS.



Suppose that your storage uses all of its I/O credit balance. If so, its maximum performance remains at the base performance level until I/O demand drops below the base level, and unused I/O credits are added to the I/O credit balance. You might notice that your storage performance is frequently limited to the base level due to an empty I/O credit balance. If so, consider allocating more General Purpose SSD storage with a higher base performance level. Alternatively, you can switch to Provisioned IOPS storage for workloads that require sustained IOPS performance.

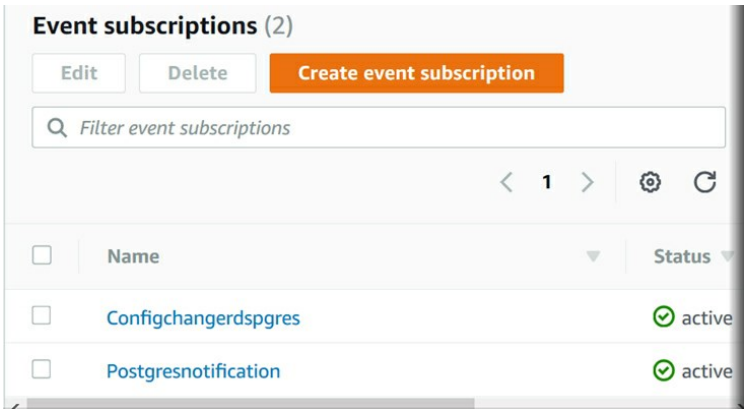


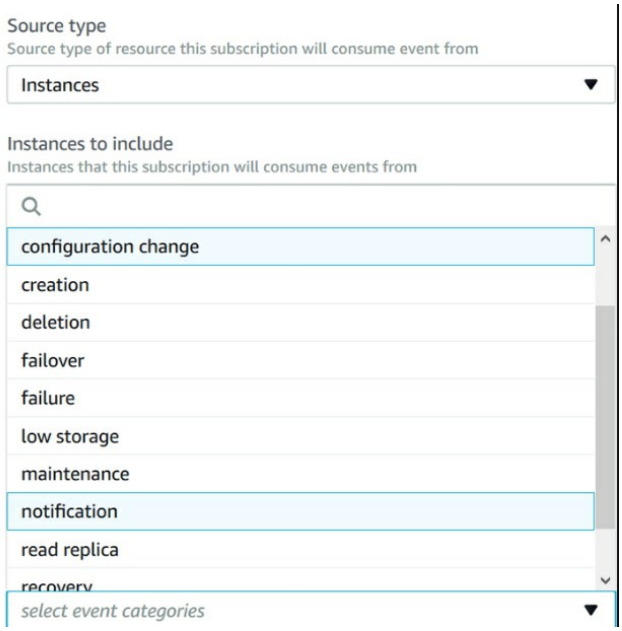
References:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Storage.html
https://aws.amazon.com/blogs/database/understanding-burst-vs-baseline-performance-with-amazon-rds-and-gp2/
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-optimized.htm

---

Q3) The Database Manager wants to keep track of DB instance-level operations in the Amazon RDS for Oracle infrastructure in the US-East region and get notified via a text message whenever the DB instances failover or restart.
Which solution will meet these requirements with minimal effort?

● Start a Database Activity Stream that will track any changes in the RDS database. Use Amazon SNS to send notifications.
Explanation:-This option is Incorrect because Database Activity Streams is only available for Amazon Aurora.
● Create an AWS Lambda function to trigger on AWS CloudTrail API calls. Filter on specific RDS API calls and use Amazon SNS to send the notifications.
Explanation:-This option is incorrect Although this is a possible solution, using Amazon RDS Event Notifications would require less effort.
✅ Use Amazon RDS Event Notifications and create proper event subscriptions that send notifications to the manager using the RDS Console.
Explanation:-
Amazon RDS uses the Amazon Simple Notification Service (Amazon SNS) to notify when an Amazon RDS event occurs. These notifications can be in any notification form supported by Amazon SNS for an AWS Region, such as an email, a text message, or a call to an HTTP endpoint.
Amazon RDS groups these events into categories that you can subscribe to so that you can be notified when an event in that category occurs. You can subscribe to an event category for a DB instance, DB snapshot, DB parameter group, or DB security group. Event notifications are sent to the addresses that you provide when you create the subscription. Amazon RDS uses the ARN of an Amazon SNS topic to identify each subscription. The Amazon RDS console creates the ARN for you when you create the subscription.





Reference:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_Events.html

● Create an Amazon Cloudwatch Events rule with the operations that need to be tracked on the DB instances. Write an AWS Lambda function to act on these rules and use Amazon SNS to send a text message.
Explanation:-This option is incorrect Although this is a possible solution, using Amazon RDS Event Notifications would require less effort.

Q4) An online tax registration system uses an Amazon Aurora PostgreSQL database to store financial data and generate tax return reports. The system has a write-intensive workload that processes and saves hundreds of user information every hour. The taxpayers who are lodging their data are complaining about the slow performance of the system. As per initial checking, there is a spike in the IO:Xactsync metric of the PostgreSQL database.
What should the Database Specialist do to resolve this issue?

- Decrease the fillfactor of the PostgreSQL tables.

Explanation:-This option is incorrect because this action will only alleviate the heavy write contention for a single page (hot page) caused by the frequent updates of the same piece of data by many sessions. This issue can be detected by checking the LWLock:buffer_content metric. Remember that in the scenario, it was explicitly mentioned that the spike is on the IO:XactSync metric. Decreasing the fillfactor of the PostgreSQL table is not warranted for this situation.

- ✅ Refactor the online tax registration system to commit transactions in batches.

Explanation:-You manage your Amazon Aurora DB cluster in the same way that you manage other Amazon RDS DB instances, by using parameters in a DB parameter group. Amazon Aurora differs from other DB engines in that you have a DB cluster that contains multiple DB instances. As a result, some of the parameters that you use to manage your Amazon Aurora DB cluster apply to the entire cluster, while other parameters apply only to a particular DB instance in the DB cluster.
Cluster-level parameters are managed in DB cluster parameter groups. Instance-level parameters are managed in DB parameter groups. Although each DB instance in an Aurora PostgreSQL DB cluster is compatible with the PostgreSQL database engine, some of the PostgreSQL database engine parameters must be applied at the cluster level, and are managed using DB cluster parameter groups. Cluster-level parameters are not found in the DB parameter group for a DB instance in an Aurora PostgreSQL DB cluster
In the IO:XactSync wait event, a session is issuing a COMMIT or ROLLBACK, requiring the current transaction's changes to be persisted. Aurora is waiting for Aurora storage to acknowledge persistence.
This wait event most often arises when there is a very high rate of commit activity on the system. You can sometimes alleviate this by modifying applications to commit transactions in batches. You might see this wait event at the same time as CPU waits in a case where the DB load exceeds the number of virtual CPUs (vCPUs) for the DB instance. In this case, the storage persistence might be competing for CPU with CPU-intensive database workloads. To alleviate this scenario, you can try reducing those workloads, or scaling up to a DB instance with more vCPUs.
References:
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraPostgreSQL.Reference.html
https://aws.amazon.com/blogs/database/analyzing-amazon-rds-database-workload-with-performance-insights/

- Decrease the concurrent queries on the rows that are being updated.

Explanation:-This option is incorrect because this action will only address the wait events associated by the sessions that lookup or manipulate the parent/child relationship between a transaction and a subtransaction. This issue can be detected by checking the LWLock:SubtransControlLock metric which is apparently different from the specific metric provided in the scenario.

- Enable the Amazon Aurora cluster cache management.

Explanation:-This option is incorrect because this feature is primarily used to ensure the fast recovery of the writer DB instance in your Aurora PostgreSQL clusters if there's a failover. This will not alleviate, let alone solve, the issue in this scenario.

Q5) An application hosted in an Auto Scaling group of EC2 instances is using Amazon RDS for SQL Server as its database and Amazon ElastiCache for caching. The Database Specialist needs to view the metrics of the underlying operating system (OS) that the DB instance runs on in real-time to analyze how the processes or threads use the CPU.
What should the Database Specialist do to view the required metrics?

- Turn on Detailed Monitoring

Explanation:-This option is incorrect because this feature is only available in Amazon EC2 and not in Amazon RDS.

- Use Performance Insights

Explanation:-This option is incorrect because it is not capable of providing you how different processes or threads on a DB instance use the CPU in real-time. It can only visualize the database load and filter the load by waits, SQL statements, hosts, or users.

- ✅ Enable Enhanced Monitoring

Explanation:-Amazon RDS provides metrics in real time for the operating system (OS) that your DB instance runs on. You can view the metrics for your DB instance using the console, or consume the Enhanced Monitoring JSON output from CloudWatch Logs in a monitoring system of your choice. By default, Enhanced Monitoring metrics are stored in the CloudWatch Logs for 30 days. To modify the amount of time the metrics are stored in the CloudWatch Logs, change the retention for the RDSOSMetrics log group in the CloudWatch console.
Take note that there are certain differences between CloudWatch and Enhanced Monitoring Metrics. CloudWatch gathers metrics about CPU utilization from the hypervisor for a DB instance, and Enhanced Monitoring gathers its metrics from an agent on the instance. As a result, you might find differences between the measurements, because the hypervisor layer performs a small amount of work.
The differences can be greater if your DB instances use smaller instance classes, because then there are likely more virtual machines (VMs) that are managed by the hypervisor layer on a single physical instance. Enhanced Monitoring metrics are useful when you want to see how different processes or threads on a DB instance use the CPU.
References:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_Monitoring.OS.html#USER_Monitoring.OS.CloudWatchLogs
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/MonitoringOverview.html#monitoring-cloudwatch

- View the metrics from the Amazon RDS Events dashboard.

Explanation:-This option is incorrect because the RDS Events feature is primarily used to track the events or changes of a given DB instance, DB snapshot, DB security group, or DB parameter group. It's doesn't track the CPU utilization of your DB instance.

Q6) A company has been running a production application that uses Amazon RDS for MySQL Multi-AZ database. The Database Specialist would frequently receive an urgent request from the application support team to restore a recent copy of the database. Upon investigation, it was found that some users commit data deletion changes on the tables of the production database, which need to be undone immediately upon system validation. They are considering a more sustainable solution.
Which solution should the Database Specialist implement to accelerate the provision of the desired production database whenever the error occurs?

- Configure the standby replica to delay replication. When an error occurs, force a manual failover to its standby replica.

Explanation:-This option is incorrect because a standby replica is designed for high availability and is configured for synchronous replication. Furthermore, it cannot be configured for delayed replication.

- ✅ Migrate to an Amazon Aurora with MySQL compatibility DB cluster. Enable and use the Backtrack feature.

Explanation:-With Amazon Aurora with MySQL compatibility, you can backtrack a DB cluster to a specific time, without restoring data from a backup. Backtracking allows you to rewind the DB cluster to the time you specify. When you specify a time for a backtrack, Aurora automatically chooses the nearest possible consistent time. Although backtracking is not a replacement for backing up your database, it provides the following advantages over traditional backup and restore:
1) You can easily undo mistakes. If you mistakenly perform a destructive action, such as a DELETE without a WHERE clause, you can backtrack the

DB cluster to a time before the destructive action with minimal interruption of service.
2) You can backtrack a DB cluster quickly. Restoring a DB cluster to a point in time launches a new DB cluster and restores it from backup data or a DB cluster snapshot, which can take hours. Backtracking a DB cluster doesn't require a new DB cluster and rewinds the DB cluster in minutes.
3) You can explore earlier data changes. You can repeatedly backtrack a DB cluster back and forth in time to help determine when a particular data change occurred. For example, you can backtrack a DB cluster for three hours and then backtrack forward in time for one hour. In this case, the backtrack time is two hours before the original time.

- Use Amazon RDS for MySQL Backtrack feature. Backtrack the database to the latest point in time before the error occurred.

Explanation:-This option is incorrect because Amazon RDS for MySQL deployments does not support the Backtrack feature.

- Create a MySQL Read Replica. When an error occurs, promote the replica to be the new source DB instance. Modify the application configuration file to point the users to connect to it.

Explanation:-This option is incorrect because a Read Replica is designed to enhance performance and durability for RDS database instances. It cannot revert to the desired point in time. Although it can be configured to delay replication, it cannot undo the committed transactions that occurred beyond what has already been replicated.

Q7) A Database Specialist manages an Amazon DocumentDB. An Application Manager requested to audit all events when an index or a collection was either created or dropped. The Manager wanted to push these records into a central dashboard. Hence, the existing custom parameter group was modified to enable the audit_logs parameter.
What else should the Database Specialist do to meet this requirement?

- Do nothing. The audit logs will automatically be viewed in Amazon CloudWatch Logs.

Explanation:-This option is incorrect. Since the requirement asks to publish the data in a central dashboard, the DocumentDB cluster needs to be modified to publish the logs to Amazon CloudWatch.

- ✅ Modify the DocumentDB cluster to export the audit logs to Amazon CloudWatch.

Explanation:-With Amazon DocumentDB (with MongoDB compatibility), you can audit events that were performed in your cluster. Examples of logged events include successful and failed authentication attempts, dropping a collection in a database, or creating an index. By default, auditing is disabled on Amazon DocumentDB and requires that you to opt-in use this feature.
When auditing is enabled, Amazon DocumentDB records Data Definition Language (DDL), authentication, authorization, and user management events to Amazon CloudWatch Logs. When auditing is enabled, Amazon DocumentDB exports your cluster's auditing records (JSON documents) to Amazon CloudWatch Logs. You can use Amazon CloudWatch Logs to analyze, monitor, and archive your Amazon DocumentDB auditing events.

- Modify the DocumentDB cluster to export DocumentDB Events to Amazon CloudWatch Logs.

Explanation:-This option is incorrect. DocumentDB Events relate to your clusters, instances, snapshots, security groups, and cluster parameter groups. This feature doesn't track audit logs by default and can't be used to directly export audit logs to CloudWatch.

- Enable the profiler feature of DocumentDB and publish the logs to Amazon CloudWatch.

Explanation:-This option is incorrect because the profiler is used for logging the execution time and details of operations that were performed on your cluster.

Q8) A Database Specialist is managing an Amazon DynamoDB table called tutorialsdojo-users. To maintain the size and capacity utilization, the specialist enabled the table's Time to Live (TTL) attribute. A user complained that queries, at certain times, still capture items that they expected to expire 24 hours ago. They confirmed that no updates were made on the expired items. The Database Specialist checked the TTL settings to find the root cause. The 24-hour backup streams setting is disabled. The CloudWatch metric "TTL deleted items" shows non-zero values in the graph.
What is most likely causing this issue?

- The TTL attribute's value is a Number data type.

Explanation:-This option is incorrect because this statement, if true, only confirms that the TTL setting is configured correctly. However, it does not explain why it deletes expired items inconsistently.

- Expired items are deleted manually through the console.

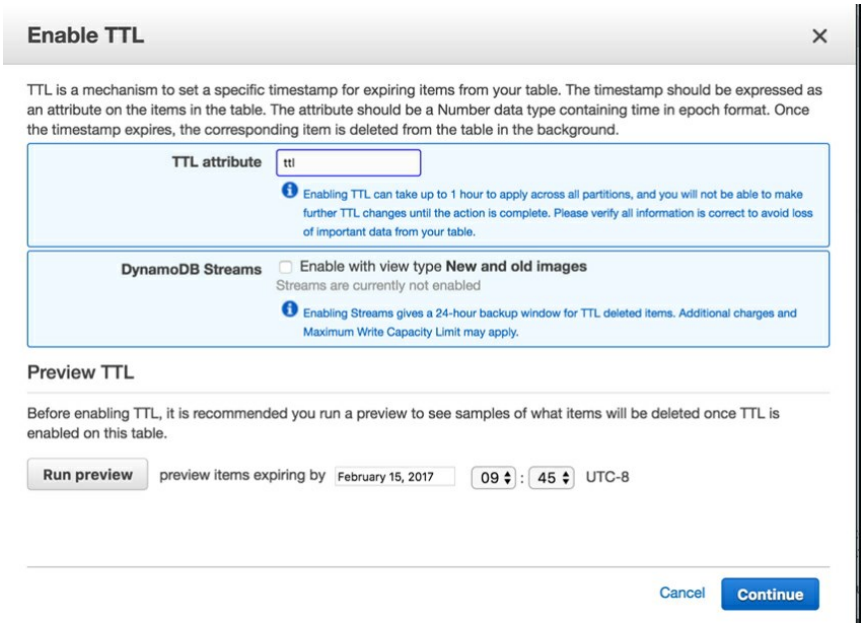
Explanation:-This option is incorrect because TTL is a background process that deletes expired items. No manual intervention is required.

- The TTL attribute's value is a String data type.

Explanation:-This option is incorrect because the investigation showed evidence through CloudWatch and the user's feedback that the expired items are deleted. The TTL attribute's value must be a Number data type to work correctly. Otherwise, it ignores the item.

- ✅ The TTL process has not deleted the expired item yet.

Explanation:-
DynamoDB deletes expired items on a best-effort basis to ensure there's enough throughput for other data operations. Depending on the size and activity level of a table, an expired item's actual delete operation can vary. Because TTL is meant to be a background process, the nature of the capacity used to expire and delete items via TTL is variable (but free of charge). TTL typically deletes expired items within 48 hours of expiration. Processing takes place automatically, in the background, and doesn't affect read or write traffic to the table.
To avoid expired items in queries, reads, or scans, you must filter them out. To do this, use a filter expression that returns only items where the Time to Live expiration value is higher than the current time in epoch format.



Q9) A Database Specialist manages an Amazon DocumentDB. An Application Manager escalated a performance issue and has requested to identify the slowest queries performed on the cluster as well as log the execution times and details of the operations.
What should the Database Specialist do to meet this requirement?

- Modify the DocumentDB cluster to enable auditing and export the logs to Amazon CloudWatch.

Explanation:-This option is incorrect. Audit logs will include all records, and thus, you cannot filter which queries will be flagged as slow.

- Modify the DocumentDB cluster to export DocumentDB Events to Amazon CloudWatch Logs.

Explanation:-This option is incorrect. DocumentDB Events relate to your clusters, instances, snapshots, security groups, and cluster parameter groups. This feature is not capable of identifying the slowest queries performed on the cluster nor logging the execution times.

- Use the Amazon DocumentDB metrics available in Cloudwatch.

Explanation:-This option is incorrect. Although these metrics will be beneficial, it does not meet the requirements of the Application Manager, who wanted more details on certain operations performed in the cluster.

- ✅ Enable the profiler feature of DocumentDB and publish the logs to Amazon CloudWatch.

Explanation:-You can use the profiler in Amazon DocumentDB (with MongoDB compatibility) to log the execution time and details of operations that were performed on your cluster. The profiler is useful for monitoring the slowest operations on your cluster to improve individual query performance and overall cluster performance.
By default, the profiler feature is disabled. When enabled, the profiler logs operations that are taking longer than a customer-defined threshold value (for example, 100 ms) to Amazon CloudWatch Logs. Recorded details include the profiled command, time, plan summary, and client metadata.
After the operations are logged to CloudWatch Logs, you can use CloudWatch Logs Insights to analyze, monitor, and archive your Amazon DocumentDB profiling data.
Enabling the profiler on a cluster is a three-step process. Ensure that all steps are completed, or profiling logs will not be sent to CloudWatch Logs. The profiler is set at the cluster level and is performed on all of the cluster's databases and instances.
References:
https://docs.aws.amazon.com/documentdb/latest/developerguide/profiling.html
https://docs.aws.amazon.com/documentdb/latest/developerguide/best_practices.html

---

Q10) A user escalated an error to the Database Administrator when running a table update statement against an Amazon Aurora for MySQL DB cluster with two Aurora Replicas:
Error Code: 1290. The MySQL server is running with the -–read-only option so it cannot execute this statement
The administrator confirmed that the test connection could write to the Aurora cluster, and no instance failover event occurred for the past month.
Which is the most likely reason for this issue?

- ✅ The user was connected to the Aurora cluster's reader endpoint when the error occurred.

Explanation:-Amazon Aurora typically involves a cluster of DB instances instead of a single instance. A specific DB instance handles each connection. When you connect to an Aurora cluster, the hostname and port that you specify points to an intermediate handler called an endpoint. Aurora uses the endpoint mechanism to abstract these connections. Thus, you don't have to hardcode all the hostnames or write your logic for load-balancing and rerouting connections when some DB instances aren't available. The primary instance handles all data definition language (DDL) and data manipulation language (DML) statements. Up to 15 Aurora Replicas handle read-only query traffic.
A reader endpoint for an Aurora DB cluster provides load-balancing support for read-only connections to the DB cluster. If the cluster contains one or more Aurora Replicas, the reader endpoint load-balances each connection request among the Aurora Replicas. In that case, you can only perform read-only statements such as SELECT in that session. If the cluster only contains a primary instance and no Aurora Replicas, the reader endpoint connects to the primary instance. In that case, you can perform write operations through the endpoint.
In this situation, a user was attempting to run a series of DML statements against an Aurora DB cluster running without failover for at least 30 days. Error code 1290 commonly occurs in an Amazon Aurora DB cluster environment whenever the database cluster is configured to be read-only or if the connection is attempting to commit write transactions to an endpoint that points to the Aurora replica.

- An automatic DB cluster snapshot was currently running.

Explanation:-This option is incorrect because an Aurora cluster will not set the instance to read-only when an automatic snapshot is running. No performance impact or interruption of database service occurs while the backup is running.

- The user had no write privileges on the database table.

Explanation:-This option is incorrect because the error least likely referred to the privileges of the database user.

- The Aurora cluster went over the maximum number of connections limit as defined by the max_connections parameter.

Explanation:-This option is incorrect because an Aurora cluster will not automatically failover to read-only mode when the cluster reaches the connections limit. If this statement is true, the user would not be able to successfully connect to the database and possibly receive an: Error 1040: Too many connection error.

Q11) A Database Engineer plans to migrate an Oracle database instance running in the Northern California (us-west-1) region to an Amazon RDS for MySQL DB instance located in the Northern Virginia (us-east-1) region using AWS DMS. The source database is configured to dynamically transform and manipulate data using transformation rule expressions.
How should the engineer set up the AWS DMS replication instance to achieve the MOST optimal performance?

- Launch the AWS DMS replication instance in the same AWS Region and VPC where the Amazon RDS for MySQL database instance is running.

Explanation:-This option is incorrect because should you place the DMS replication instance in the same VPC and Availability Zone (AZ) of the source database. The replication instance connects to your source data store then reads and formats the source data before sending it to the target database. A large amount of traffic is exchanged between the source and the replication instance which is why they have to be in the same VPC and AZ for optimal performance.

- Launch the AWS DMS replication instance in the same AWS Region where the Oracle database instance is running.

Explanation:-This option is is incorrect because the replication instance could be placed in a different VPC or Availability Zone where the database source is not running. This setup is not the most optimal configuration because the replication instance is not running on the exact same Availability Zone and VPC of the source database.

- ✅ Launch the AWS DMS replication instance in the same AWS Region, VPC, and Availability Zone where the Oracle database instance is running.

Explanation:-When you create an AWS DMS replication instance, AWS DMS creates the replication instance on an Amazon Elastic Compute Cloud (Amazon EC2) instance in a VPC based on the Amazon Virtual Private Cloud (Amazon VPC) service. You can use this replication instance to perform your database migration. The replication instance provides high availability and failover support using a Multi-AZ deployment when you select the Multi-AZ option.
AWS DMS uses a replication instance to connect to your source data store, read the source data, and format the data for consumption by the target data store. A replication instance also loads the data into the target datastore. Most of this processing happens in memory. However, large transactions

might require some buffering on disk. Cached transactions and log files are also written to disk.
You can set up the DMS replication instance in the same VPC, Availability Zone, and AWS Region of either the source or target database. However, if you are only migrating or replicating a subset of data using filters or transformations, it is recommended that you launch the replication instance on the same VPC and Availability Zone where the source database is, to optimize the processing. Most of the time, the amount of data transferred over the network to the target database is less compared with the source database.
To define content for new and existing columns, you can use an expression within a transformation rule. For example, using expressions you can add a column or replicate source table headers to a target. You can also use expressions to flag records on target tables as inserted, updated, or deleted at the source.
References:
https://docs.aws.amazon.com/dms/latest/userguide/CHAP_ReplicationInstance.html
https://aws.amazon.com/blogs/database/disaster-recovery-on-amazon-rds-for-oracle-using-aws-dms
https://docs.aws.amazon.com/dms/latest/userguide/CHAP_ReplicationInstance.VPC.html
● Launch the AWS DMS replication instance in the same AWS Account where the Amazon RDS for MySQL database instance is running.
Explanation:-This option is incorrect because running the replication instance in the same AWS account is not enough. The AWS DMS replication instance must be launched in the same AWS Region, VPC, and Availability Zone where the Oracle database instance (source database) is running for optimal network and migration performance.

Q12) A developer accidentally deleted an Amazon RDS MySQL database in the production environment that resulted in two hours of downtime. The database has been successfully restored using Point-in-Time Recovery. To prevent this error from happening again, the Database Specialist must implement a more stringent policy to safeguard the production resources.
Which of the following should be done in this scenario?

● Enable Enhanced Monitoring in the RDS database. Set up RDS Event Notification to get notified if a DB instance was deleted. Enable Performance Insights.
Explanation:-This option is incorrect. Enhanced Monitoring, RDS Event Notification, and RDS Performance Insights are meant to be used to monitor your RDS databases. These are not enough in preventing the developers from accidentally deleting a database in the production environment.
● Block all developers from accessing or deleting the database using the RDS Access Control List (ACL). Enable multi-factor authentication for sensitive operations.
Explanation:-This option is incorrect because RDS doesn't have an Access Control List(ACL). You have to use IAM policies to block an IAM user from accessing or deleting your RDS databases.
✅ Review the IAM policies associated with the IAM Users. Grant the least privilege to the developers and enable multi-factor authentication for sensitive operations.
Explanation:-Amazon RDS conforms to the AWS shared responsibility model, which includes regulations and guidelines for data protection. AWS is responsible for protecting the global infrastructure that runs all the AWS services. AWS maintains control over data hosted on this infrastructure, including the security configuration controls for handling customer content and personal data. AWS customers and APN partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.
For data protection, AWS recommends that you protect your AWS account credentials and set up principals with AWS Identity and Access Management (IAM). Doing this means that each user is given only the permissions necessary to fulfill their job duties. It is also recommended that you secure your data in the following ways:
- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
AWS strongly recommends that you never put sensitive identifying information, such as your customers' account numbers, into free-form fields such as a Name field. This recommendation includes when you work with Amazon RDS or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into these fields in Amazon RDS or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.
Use AWS Identity and Access Management (IAM) accounts to control access to Amazon RDS API operations, especially operations that create, modify, or delete Amazon RDS resources. Such resources include DB instances, security groups, and parameter groups. Also, use IAM to control actions that perform common administrative actions such as backing up and restoring DB instances.
References:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAM.html
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/DataDurability.html
● Use RDS Proxy to block any delete operations in Amazon RDS. Enable multi-factor authentication for sensitive operations.
Explanation:-This option is incorrect because RDS Proxy is primarily used to allow applications to pool and share connections established with the database. This feature can't be used to block delete operations in Amazon RDS.

Q13) An application has a global sales leaderboard that provides the list of agents filtered by department and sales volume. A Sorted Set data type in Amazon ElastiCache for Redis is used for the leaderboard to show and arrange the updated data in real-time. The Database Specialist needs to implement a fault-tolerant caching layer that enhances the data durability of the ElastiCache cluster. The data loss should not exceed 1 hour to comply with the RPO.
Which of the following are valid solutions that can be implemented to meet the above requirement? (Select TWO.)

✅ Set up ElactiCache Multi-AZ with Automatic Failover.
Explanation:-An ElastiCache Redis cluster provides varying levels of data durability, performance, and cost for implementing disaster recovery or fault tolerance of your cached data. You can choose the following options to improve the data durability of your ElastiCache cluster:
- Daily automatic backups
- Manual backups using Redis append-only file (AOF)
- Setting up a Multi-AZ with Automatic Failover
Enabling ElastiCache Multi-AZ on your Redis cluster (in the API and CLI, replication group) improves your fault tolerance. This is true particularly in cases where your cluster's read/write primary cluster becomes unreachable or fails for any reason. Multi-AZ is only supported on Redis clusters that have more than one node in each shard.
By default, the data in a Redis node on ElastiCache resides only in memory and is not persistent. If a node is rebooted, or if the underlying physical server experiences a hardware failure, the data in the cache is lost.
If you require data durability, you can enable the Redis append-only file feature (AOF). When this feature is enabled, the node writes all of the commands that change cache data to an append-only file. When a node is rebooted and the cache engine starts, the AOF is "replayed"; the result is a warm Redis cache with all of the data intact.
AOF is disabled by default. To enable AOF for a cluster running Redis, you must create a parameter group with the appendonly parameter set to yes, and then assign that parameter group to your cluster. You can also modify the appendfsync parameter to control how often Redis writes to the AOF file.

References:
https://aws.amazon.com/premiumsupport/knowledge-center/fault-tolerance-elasticache/
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/AutoFailover.html
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/RedisAOF.html

Set up daily automatic backups.

Explanation:-This option is incorrect because the data loss potential for daily scheduled backups is high (up to a day's worth of data). Remember that the scenario indicates that the data loss should not exceed 1 hour. Since you will only backup your ElastiCache once a day, there will be hours of gap in between the scheduled backup. If you scheduled the backup every 1 AM and the outage happened in the middle of the day (1 PM), then the data loss will be up to 12 hours.

Schedule manual backups using Redis append-only file (AOF).

Explanation:-An ElastiCache Redis cluster provides varying levels of data durability, performance, and cost for implementing disaster recovery or fault tolerance of your cached data. You can choose the following options to improve the data durability of your ElastiCache cluster:
- Daily automatic backups
- Manual backups using Redis append-only file (AOF)
- Setting up a Multi-AZ with Automatic Failover

Enabling ElastiCache Multi-AZ on your Redis cluster (in the API and CLI, replication group) improves your fault tolerance. This is true particularly in cases where your cluster's read/write primary cluster becomes unreachable or fails for any reason. Multi-AZ is only supported on Redis clusters that have more than one node in each shard.

By default, the data in a Redis node on ElastiCache resides only in memory and is not persistent. If a node is rebooted, or if the underlying physical server experiences a hardware failure, the data in the cache is lost.

If you require data durability, you can enable the Redis append-only file feature (AOF). When this feature is enabled, the node writes all of the commands that change cache data to an append-only file. When a node is rebooted and the cache engine starts, the AOF is "replayed"; the result is a warm Redis cache with all of the data intact.

AOF is disabled by default. To enable AOF for a cluster running Redis, you must create a parameter group with the appendonly parameter set to yes, and then assign that parameter group to your cluster. You can also modify the appendfsync parameter to control how often Redis writes to the AOF file.

References:
https://aws.amazon.com/premiumsupport/knowledge-center/fault-tolerance-elasticache/
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/AutoFailover.html
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/RedisAOF.html

Switch to Amazon ElastiCache for Memcached instead of Redis.

Explanation:-This option is incorrect because using Memcached doesn't automatically make your ElactiCache cluster more fault-tolerant. Moreover, Memcached doesn't support advanced data structures like the Sorted Set data type that is used by the global sales leaderboard application.

Use Redis AUTH on the ElastiCache for Redis cluster.

Explanation:-This option is incorrect because Redis Auth is just an authorization layer that refuses any query by unauthenticated clients.

Q14) A Database Specialist is managing an Amazon RDS for PostgreSQL DB instance with 32 TiB of storage. The database needs to be migrated to an Aurora PostgreSQL DB cluster to avail of the various features of Amazon Aurora. The Database Specialist must ensure that there is minimal downtime and no data loss when doing the migration.
Which of the following approach is the FASTEST way to migrate the database?

Generate an Aurora Read Replica from the RDS PostgreSQL DB Instance. In the database cut-over, promote the Aurora Read Replica to a standalone Aurora DB cluster.

Explanation:-To migrate from an RDS PostgreSQL DB instance to an Aurora PostgreSQL DB cluster, it is recommended to create an Aurora Read Replica of your source PostgreSQL DB instance. When the replica lag between the PostgreSQL DB instance and the Aurora PostgreSQL Read Replica is zero, you can stop replication. At this point, you can promote the Aurora Read Replica to be a standalone Aurora PostgreSQL DB cluster. This standalone DB cluster can then accept write loads.

Be prepared for migration to take a while, roughly several hours per tebibyte (TiB) of data. While the migration is in progress, your Amazon RDS PostgreSQL instance accumulates write-ahead log (WAL) segments. Make sure that your Amazon RDS instance has sufficient storage capacity for these segments.

When you create an Aurora Read Replica of a PostgreSQL DB instance, Amazon RDS creates a DB snapshot of your source PostgreSQL DB instance. This snapshot is private to Amazon RDS and incurs no charges. Amazon RDS then migrates the data from the DB snapshot to the Aurora Read Replica. After the DB snapshot data is migrated to the new Aurora PostgreSQL DB cluster, RDS starts replication between your PostgreSQL DB instance and the Aurora PostgreSQL DB cluster.

You can only have one Aurora Read Replica for a PostgreSQL DB instance. If you try to create an Aurora Read Replica for your Amazon RDS PostgreSQL instance and you already have a read replica, the request is rejected.

References:
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraPostgreSQL.Migrating.html#AuroraPostgreSQL.Migrating.RDSPostgreSQL.Replica
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.AuroraPostgreSQL.html
https://docs.aws.amazon.com/prescriptive-guidance/latest/database-migration-strategy/cut-over.html

Take a DB snapshot of the RDS PostgreSQL DB instance. Create an Aurora PostgreSQL DB cluster from the snapshot.

Explanation:-This option is incorrect. Although this process is valid, this approach will result in a data loss since the generated snapshot would not have the latest data. It also takes time to take a DB snapshot, migrate the data, and launch a new Aurora PostgreSQL DB cluster. A faster solution is to generate an Aurora Read Replica from the RDS database instance instead.

Create an RDS Read Replica from the RDS PostgreSQL DB Instance. In the database cut-over, promote the Read Replica to a standalone Aurora DB cluster.

Explanation:-This option is incorrect because you cannot directly promote an RDS Read Replica to a standalone Aurora DB cluster. You have to generate an Aurora Read Replica, not just a regular RDS replica.

Use AWS DMS to migrate the RDS PostgreSQL DB instance to a standalone Aurora DB Cluster. Launch a replication instance and use change data capture (CDC) by disabling binary logging.

Explanation:-This option is incorrect because you actually have to enable binary logging, and not disable it, to use the change data capture (CDC) replication.

Q15) A startup is developing a mobile app that will use a DynamoDB table named SessionData for user session management. The table has UserName, SessionId, CreationTime, and ExpirationTime attributes for tracking the session information. Each item only needs to be stored for 2 days and must be deleted afterward. The SessionId attribute is set to be the partition key in alphanumeric format.
How should the Database Specialist configure the table?

Enable DynamoDB Streams on the table that will track the SessionId attribute. Configure the Stream settings to automatically expire the items every 2 days.

Explanation:-This option is incorrect because DynamoDB Streams is primarily used for capturing and processing changes to the individual items on a

table. Using the DynamoDB Streams feature is not appropriate in this scenario. You have to enable TTL in the table to satisfy the given requirement.

✅ Configure the ExpirationTime attribute to be a Number data type containing time in Unix epoch format. Enable time to live (TTL) on the table and track the ExpirationTime attribute. Ensure that the app sets the expiration time 2 days ahead for each new item.

Explanation:-Amazon DynamoDB Time to Live (TTL) allows you to define a per-item timestamp to determine when an item is no longer needed. Shortly after the date and time of the specified timestamp, DynamoDB deletes the item from your table without consuming any write throughput. TTL is provided at no extra cost as a means to reduce stored data volumes by retaining only the items that remain current for your workload's needs.

TTL is useful if you store items that lose relevance after a specific time. The following are example TTL use cases:

Remove user or sensor data after one year of inactivity in an application.

Archive expired items to an Amazon S3 data lake via DynamoDB Streams and AWS Lambda.

Retain sensitive data for a certain amount of time according to contractual or regulatory obligations.

When enabling TTL on a DynamoDB table, you must identify a specific attribute name that the service will look for when determining if an item is eligible for expiration. After you enable TTL on a table, a per-partition scanner background process automatically and continuously evaluates the expiry status of items in the table.

The scanner background process compares the current time, in Unix epoch time format in seconds, to the value stored in the user-defined attribute of an item. If the attribute is a Number data type, the attribute's value is a timestamp in Unix epoch time format in seconds, and the timestamp value is older than the current time but not five years older or more (in order to avoid a possible accidental deletion due to a malformed TTL value), then the item is set to expired. A second background process scans for expired items and deletes them. Both processes take place automatically in the background, do not affect read or write traffic to the table, and do not have a monetary cost.

References:

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TTL.html

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/howitworks-ttl.html

⚪ Set up an extract, transform, and load (ETL) process using AWS Glue that will automatically truncate the table every 2 days.

Explanation:-This option is incorrect because using AWS Glue is not suitable for truncating an entire DynamoDB table. Deleting individual items in the table can easily be accomplished by using TTL, and not ETL.

⚪ Configure the cache behavior of the DynamoDB table to expire each item after 2 days by setting the Cache-Control header.

Explanation:-This option is incorrect because by default, DynamoDB doesn't have a cache behavior setting. This is only available if you are using DynamoDB Accelerator (DAX). The Cache-Control header is commonly used in Amazon CloudFront, not in DynamoDB.

Q16) A database team is investigating an application connection failure on an Amazon Aurora MySQL DB cluster with multiple Aurora Replicas in the same Region that had been running with no issues for the past 2 months. The connection failure lasted for 5 minutes and corrected itself after that. The Database Specialist reviewed the events from Amazon CloudWatch and verified that a failover event did occur during that time. It took the failover process around 15 seconds to complete.
Which of the following most likely caused the 5-minute connection outage?

⚪ During failover, Amazon Aurora requires time to reattach the disk volumes that make up the cluster volume from the primary instance to the newly-promoted Replica.

Explanation:-This option is incorrect because Amazon Aurora stores copies of the data in a DB cluster across multiple Availability Zones in a single AWS Region. Aurora stores these copies regardless of whether the instances in the DB cluster span multiple Availability Zones. When data is written to the primary DB instance, Aurora synchronously replicates the data across Availability Zones to six storage nodes associated with your cluster volume. Doing so provides data redundancy, eliminates I/O freezes, and minimizes latency spikes during system backups. The data in the cluster volume is represented as a single, logical volume to the primary instance and to Aurora Replicas in the DB cluster.

✅ The client-side application's time-to-live is set too high and is caching the old DNS data.

Explanation:-Unless you use a smart database driver, you depend on DNS record updates and DNS propagation for failovers, instance scaling, and load balancing across Aurora Replicas. Ensure that your network and client configurations don't further increase the DNS cache TTL.

If your client application is caching the Domain Name Service (DNS) data of your DB instances, set a time-to-live (TTL) value of fewer than 30 seconds. Because the underlying IP address of a DB instance can change after a failover, caching the DNS data for an extended time can lead to connection failures if your application tries to connect to an IP address that no longer is in service. Aurora DB clusters with multiple read replicas can also experience connection failures when connections use the reader endpoint, and one of the read replica instances is in maintenance or is deleted.

Here are some examples of issues that can occur if you don't follow DNS caching best practices:

- After a new primary instance is promoted during a failover, applications continue to send write traffic to the old instance. Data-modifying statements will fail because that instance is no longer the primary instance.
- After a DB instance is scaled up or down, applications are unable to connect to it. Due to DNS caching, applications continue to use the old IP address of that instance, which is no longer valid.
- Aurora Replicas can experience unequal utilization, for example, one DB instance receiving significantly more traffic than the others.

References:

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

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Concepts.AuroraHighAvailability.html

⚪ There were no active Aurora Replicas in the Aurora DB cluster.

Explanation:-This option is incorrect because an Amazon Aurora DB Cluster with multiple Aurora Replicas is designed to be fault-tolerant and highly available. It is not possible to manually stop an Aurora replica.

⚪ Before accepting client connections, Amazon Aurora needs to warm the buffer pool cache during a database restart.

Explanation:-This option is incorrect because Amazon Aurora uses the Survivable Cache Warming technique. The Aurora page cache is managed in a separate process from the database, which allows the page cache to survive independently of the database. In the unlikely event of a database failure, the page cache remains in memory, which ensures that the buffer pool is warmed with the most current state when the database restarts.

Q17) A company runs its mission-critical application in AWS, which uses an Amazon Aurora DB cluster to store financial transactions. The application has a reporting module that's heavily utilized by its users. The Aurora DB cluster consists of a medium-sized primary DB instance, two medium-sized replicas, and a large-sized read replica. For the promotion tier, all of the replicas are using the same default values.
What will happen if the primary instance experienced a failure?

⚪ The primary instance will be recreated. No failover will occur.

Explanation:-This option is incorrect because the Aurora DB cluster has several read replicas. Therefore, Amazon RDS will promote one of these read replicas to become the primary instance. A failover will only be skipped if there are no available read replicas.

⚪ Amazon RDS promotes the two medium-sized replicas as the primary DB instances.

Explanation:-This option is incorrect because Amazon RDS only promotes one replica to become the primary instance in the Aurora DB cluster.

✅ Amazon RDS promotes the read replica with the largest instance size.

Explanation:-An Aurora DB cluster is fault-tolerant by design. The cluster volume spans multiple Availability Zones in a single AWS Region, and each Availability Zone contains a copy of the cluster volume data. This functionality means that your DB cluster can tolerate a failure of an Availability Zone without any loss of data and only a brief interruption of service.

If the primary instance in a DB cluster using single-master replication fails, Aurora automatically fails over to a new primary instance in one of two ways:
By promoting an existing Aurora Replica to the new primary instance
By creating a new primary instance
If the DB cluster has one or more Aurora Replicas, then an Aurora Replica is promoted to the primary instance during a failure event. A failure event results in a brief interruption, during which read and write operations fail with an exception. However, service is typically restored in less than 120 seconds, and often less than 60 seconds. To increase the availability of your DB cluster, it is recommended that you create at least one or more Aurora Replicas in two or more different Availability Zones.
You can customize the order in which your Aurora Replicas are promoted to the primary instance after a failure by assigning each replica a priority. Priorities range from 0 for the first priority to 15 for the last priority. If the primary instance fails, Amazon RDS promotes the Aurora Replica with better priority to the new primary instance. You can modify the priority of an Aurora Replica at any time. Modifying the priority doesn't trigger a failover.
If the DB cluster doesn't contain any Aurora Replicas, then the primary instance is recreated during a failure event. A failure event results in an interruption during which read and write operations fail with an exception. Service is restored when the new primary instance is created, which typically takes less than 10 minutes. Promoting an Aurora Replica to the primary instance is much faster than creating a new primary instance.
More than one Aurora Replica can share the same priority, resulting in promotion tiers. If two or more Aurora Replicas share the same priority, then Amazon RDS promotes the replica that is largest in size. If two or more Aurora Replicas share the same priority and size, then Amazon RDS promotes an arbitrary replica in the same promotion tier.
References:
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Managing.Backups.html#Aurora.Managing.FaultTolerance
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Modifying.html

- Amazon RDS promotes an arbitrary replica.

Explanation:-This option is incorrect because this will only occur if there is no read replica available than is larger than the primary instance and if there are two or more Aurora Replicas that share the same priority and size. Take note that in this scenario, there is an existing, large-sized read replica which Amazon RDS will promote first.

Q18) A Database Specialist is managing an Amazon RDS for PostgreSQL database configured in Multi-AZ Deployments configuration. The PostgreSQL logs are published to CloudWatch Logs. Several web applications across the company heavily utilize the database to produce reports. The generated database logs affect the available storage space of the database.
What should the Database Specialist do to rectify this issue?

- Using the AWS CLI, connect to the RDS DB instance and delete all of the database logs from the $PGDATA/pg_logs/ directory.

Explanation:-This option is incorrect because the RDS database is managed by AWS. You can't manually connect to the underlying server that runs your RDS database.

- Increase the allocated storage of the RDS database by 9%.

Explanation:-This option is incorrect because an increase of less than 10% will result in an error. You must increase the allocated storage by at least 10%. Although increasing the allocated storage is valid, it can only temporarily solve the storage issue. After a few days, the database logs will still consume the allocated storage of the DB instance.

- Run the DBCC SQLPERF(LOGSPACE) TSQL query.

Explanation:-This option is incorrect because this is only applicable for the Microsoft SQL Server (MS SQL) database and not for PostgreSQL. Moreover, this query doesn't actually delete the database logs. It only provides transaction log space usage statistics to your MS SQL database.

- ✅ Edit the value of the rds.log_retention_period parameter to 1440 in the parameter group of the RDS database.

Explanation:-Amazon RDS for PostgreSQL generates query and error logs. RDS PostgreSQL writes autovacuum information and rds_admin actions to the error log. PostgreSQL also logs connections, disconnections, and checkpoints to the error log. To set logging parameters for a DB instance, set the parameters in a DB parameter group and associate that parameter group with the DB instance.
To set the retention period for system logs, use the rds.log_retention_period parameter. You can find rds.log_retention_period in the DB parameter group associated with your DB instance. The unit for this parameter is minutes. For example, a setting of 1,440 retains logs for one day. The default value is 4,320 (three days). The maximum value is 10,080 (seven days). Your instance must have enough allocated storage to contain the retained log files. To retain older logs, publish them to Amazon CloudWatch Logs.
To store your PostgreSQL log records in highly durable storage, you can use CloudWatch Logs. With CloudWatch Logs, you can also perform real-time analysis of log data and use CloudWatch to view metrics and create alarms. To work with CloudWatch Logs, configure your RDS for PostgreSQL DB instance to publish log data to a log group.
References:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_LogAccess.Concepts.PostgreSQL.html
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_LogAccess.html
https://aws.amazon.com/premiumsupport/knowledge-center/rds-out-of-storage/

Q19) As part of compliance, a Database Specialist needs to conduct fault testing to an Amazon RDS for MySQL running in Multi-AZ deployments configuration. The test will assess the resiliency of the database in the event of DB instance failure and availability zone disruption.
Which of the following actions should the Database Specialist do to simulate a failure event?

- Use AWS DMS.

Explanation:-This option is incorrect because this service is used for migrating your data to and from the most widely used commercial and open-source databases. It cannot be used for fault testing.

- Use Amazon RDS Proxy.

Explanation:-This option is incorrect because it is a database proxy feature of Amazon RDS used to help improve scalability by pooling and sharing database connections.

- ✅ Use the Reboot with failover option.

Explanation:-You might need to reboot your DB instance, usually for maintenance reasons. For example, if you make certain modifications, or if you change the DB parameter group associated with the DB instance, you must reboot the instance for the changes to take effect. Rebooting a DB instance restarts the database engine service. Rebooting a DB instance results in a momentary outage, during which the DB instance status is set to rebooting.
If the Amazon RDS instance is configured for Multi-AZ, you can perform the reboot with a failover. An Amazon RDS event is created when the reboot is completed. If your DB instance is a Multi-AZ deployment, you can force a failover from one Availability Zone (AZ) to another when you reboot. Rebooting with failover is beneficial when you want to simulate a failure of a DB instance for testing, or restore operations to the original AZ after a failover occurs.

- Use fault injection queries.

Explanation:-This option is incorrect because this option is only available in Amazon Aurora. You can test the fault tolerance of your Amazon Aurora DB cluster by using fault injection queries.

Q20) A financial services company is running an online banking portal that uses an Amazon Aurora PostgreSQL DB cluster. The DB cluster has several tables that contain sensitive customer data. The Database Specialist has been instructed to manage access privileges at the table level.

Which of the following actions must be implemented to meet the above requirements?

✅ Restrict access to the tables containing sensitive customer data using Data Control Language (DCL) commands such as GRANT and REVOKE.
Explanation:-In Amazon RDS, you can manage which users have privileges to connect to which databases. In other PostgreSQL environments, you sometimes perform this kind of management by modifying the pg_hba.conf file.
Take note that Amazon RDS is a fully-managed database service which means that AWS controls and manages the underlying servers that power your databases. Unlike a database hosted in Amazon EC2, you don't have the ability to directly connect to server and the configuration artifacts such as the pg_hba.conf or postgresql.conf files. You can use the GRANT or REVOKE commands to control table-level access.
New databases in PostgreSQL are always created with a default set of privileges. The default privileges allow PUBLIC (all users) to connect to the database and to create temporary tables while connected.
psql> revoke all on database from public;
psql> grant connect, temporary on database to ;
To control which users are allowed to connect to a given database in Amazon RDS, first revoke the default PUBLIC privileges. Then grant back the privileges on a more granular basis as shown in the example code above.
References:

https://aws.amazon.com/blogs/database/applying-best-practices-for-securing-sensitive-data-in-amazon-rds/

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.PostgreSQL.CommonDBATasks.html
● Restrict access to the tables containing sensitive customer data using IAM database authentication.
Explanation:-This option is incorrect because, with this type of authentication method, you just don't need to use a password when you connect to a DB instance. However, it doesn't provide fine-grained access control to your sensitive tables.
● Configure AWS Systems Manager Parameter Store to automatically rotate the secrets for the tables of the Aurora DB cluster containing sensitive customer data.
Explanation:-This option is incorrect because you can't use AWS Systems Manager Parameter Store to protect the individual tables of your Aurora database cluster. Moreover, this service is incapable of automatically rotating secrets. If you require secrets rotation in your architecture, you can use AWS Secret Manager instead.
● Control the access privileges to the tables of the Aurora DB cluster that contain sensitive customer data by modifying the pg_hba.conf file.
Explanation:-This option is incorrect because you don't have the ability to directly access, let alone modify, the pg_hba.conf configuration file in Amazon Aurora. You have to implement table-level restrictions using DCL commands instead.

Q21) A Database Specialist manages an Amazon RDS for PostgreSQL DB instance in a single-AZ deployment currently residing in the default Amazon VPC with access to the Internet. The company wants to secure the database and associate it with a different VPC. The Application Manager reminded the Database Specialist that the application server resides in the default VPC and should successfully connect to the database once the change is completed.
What is the FASTEST method to meet this requirement?

● Restore the RDS PostgreSQL instance snapshot in the new VPC. Use VPC Peering to allow the connection between the default VPC and the new VPC.
Explanation:-This option is incorrect. Although the solution works, it will not be the fastest.
● Restore the RDS PostgreSQL instance snapshot in the new VPC using the AWS account root user. It will automatically allow the connection between the default VPC and the new VPC.
Explanation:-This option is incorrect. You need to configure both VPCs to communicate with each other via VPC peering.
✅ Create a new DB subnet group associated with the new VPC. Modify the RDS PostgreSQL instance and associate it with the new DB Subnet Group. Use VPC Peering to allow the connection between the default VPC and the new VPC.
Explanation:-Before you move the RDS DB instance to a new network, configure the new VPC, including the security group inbound rules, the subnet group, and the route tables. When you change the VPC for a DB instance, the instance reboots when the instance moves from one network to another. Because the DB instance isn't accessible during the network modification, change the VPC during a scheduled maintenance window.
You can't change the VPC for a DB instance if:
- The DB instance is in multiple Availability Zones. Convert the DB instance to a single Availability Zone, and then convert it back to a Multi-AZ DB instance after moving to the new VPC.
Note: You can't change a DB subnet group to a Multi-AZ configuration. By default, the Amazon Aurora storage is Multi-AZ—even for a single instance—so you can't modify the VPC for Amazon Aurora.
- The DB instance is a read replica or has read replicas. Remove the read replicas, and then add read replicas after the DB instance is moved to the new VPC.
- The subnet group created in the target VPC must have subnets from the Availability Zone, where the source database is running. If the Availability Zones are different, the operation fails.
A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IP addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your VPCs, with a VPC in another AWS account, or with a VPC in a different AWS Region.
References:
https://aws.amazon.com/premiumsupport/knowledge-center/change-vpc-rds-db-instance/
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_VPC.Scenarios.html
● Modify the existing RDS PostgreSQL instance and directly change the VPC to the new VPC. Use VPC Peering to connect the default VPC to the new VPC.
Explanation:-This option is incorrect. You cannot change the VPC directly to move it to a new VPC.

Q22) A Database Specialist is creating an AWS CloudFormation template that will deploy various Amazon DynamoDB tables to the AWS environment. The template is expected to be updated regularly based on the testing results. Each table must be independently configurable so that the deployment system can dynamically assign its respective provisioned read capacity units (RCU) and write capacity units (WCU).
Which of the following should be implemented to satisfy the above requirement?

● In the CloudFormation template, add an Outputs section and include the RCU and WCU count values of the DynamoDB tables. The provisioned throughput will be automatically updated based on the CloudFormation outputs.
Explanation:-This option is incorrect because the Outputs section just declares the output values that you can import into other stacks (to create cross-stack references), return in response (to describe stack calls), or view on the AWS CloudFormation console. This is not a straightforward solution as you have to create another stack in order to assign the RCU and WCU values to the DynamoDB tables.
● Include a Transform section in the CloudFormation template and add the RCU and WCU count values of the DynamoDB tables. In the ProvisionedThroughput property, configure the ReadCapacityUnits and WriteCapacityUnits to use the Ref intrinsic function to retrieve RCU and WCU count values.

Explanation:-This option is incorrect because the Transform section simply specifies one or more macros that AWS CloudFormation uses to process your template. This is not a valid solution for this specific scenario.

● Under the Resources section of the CloudFormation template, declare two custom parameters: rcuCount and wcuCount for the DynamoDB tables, each with a specified value. The provisioned throughput will be automatically updated based on the CloudFormation outputs. In the ProvisionedThroughput property, configure the ReadCapacityUnits and WriteCapacityUnits to use the Ref intrinsic function to fetch the rcuCount and wcuCount values.

Explanation:-This option is incorrect because you have to declare the custom parameters in the Mappings section of the CloudFormation template and not in the Resources section.

✔ Add a Mappings section in the CloudFormation template and include the RCU and WCU count values of the DynamoDB tables. In the ProvisionedThroughput property, configure the ReadCapacityUnits and WriteCapacityUnits to use the Ref intrinsic function to retrieve RCU and WCU count values from the mapping.

Explanation:-When you create a new provisioned table in Amazon DynamoDB, you must specify its provisioned throughput capacity. This is the amount of read and write activity that the table can support. DynamoDB uses this information to reserve sufficient system resources to meet your throughput requirements.

You can optionally allow DynamoDB auto scaling to manage your table's throughput capacity. However, you still must provide initial settings for read and write capacity when you create the table. DynamoDB auto scaling uses these initial settings as a starting point, and then adjusts them dynamically in response to your application's requirements.

As your application data and access requirements change, you might need to adjust your table's throughput settings. If you're using DynamoDB auto scaling, the throughput settings are automatically adjusted in response to actual workloads. You can also use the UpdateTable operation to manually adjust your table's throughput capacity. You might decide to do this if you need to bulk-load data from an existing data store into your new DynamoDB table. You could create the table with a large write throughput setting and then reduce this setting after the bulk data load is complete.

You specify throughput requirements in terms of capacity units—the amount of data your application needs to read or write per second. You can modify these settings later, if needed, or enable DynamoDB auto scaling to modify them automatically.

In AWS CloudFormation, the ProvisionedThroughput property of the AWS::DynamoDB::Table resource specifies the throughput capacity for the specified table, which consists of values for ReadCapacityUnits and WriteCapacityUnits.

References:

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-dynamodb-table.html

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-dynamodb-provisionedthroughput.html

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/ProvisionedThroughput.html

---

Q23) A startup plans to deploy a mobile dating application in multiple AWS Regions. The app needs to save the user profiles and local matches in Amazon DynamoDB tables in each Region. A Database Specialist has been tasked to automate the deployment of the new tables with identical configurations for the additional Regions, or even AWS accounts, where the app will soon be launched. The solution must seamlessly automate the configuration changes across all Regions to lessen the configuration management overhead.
Which of the following is the most suitable option to implement to meet the above requirements?

✔ Set up a CloudFormation template and use a stack set to deploy the required resources to all the AWS Regions or accounts.

Explanation:-AWS CloudFormation StackSets extends the functionality of stacks by enabling you to create, update, or delete stacks across multiple accounts and regions with a single operation. Using an administrator account, you define and manage an AWS CloudFormation template, and use the template as the basis for provisioning stacks into selected target accounts across specified regions.

A stack set lets you create stacks in AWS accounts across regions by using a single AWS CloudFormation template. All the resources included in each stack are defined by the stack set's AWS CloudFormation template. As you create the stack set, you specify the template to use, as well as any parameters and capabilities that the template requires.

After you've defined a stack set, you can create, update, or delete stacks in the target accounts and Regions you specify. When you create, update, or delete stacks, you can also specify operation preferences, such as the order of regions in which you want the operation to be performed, the failure tolerance beyond which stack operations stop, and the number of accounts in which operations are performed on stacks concurrently. A stack set is a regional resource. If you create a stack set in one Region, you cannot see it or change it in other Regions.

● Use the AWS Management Console to deploy and manage the new DynamoDB tables. Enable Amazon DynamoDB Accelerator (DAX) to automate the deployment of the new tables with identical configurations for the additional Regions and accounts.

Explanation:-This option is incorrect because Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache for DynamoDB that significantly reduces the response times of eventually consistent read workloads from single-digit milliseconds to microseconds. Therefore, DAX is not an appropriate feature to be used in this scenario.

● Create a CloudFormation template and use a nested stack to deploy the required resources to all Regions or accounts.

Explanation:-This option is incorrect because you can't deploy a template to multiple AWS Regions and accounts using a nested stack. You have to use a stack set instead.

● Prepare a CloudFormation template with a Mappings section that defines the AWS Regions and accounts. Use the Fn::FindInMap intrinsic function to retrieve values in the specified map to deploy the tables to multiple Regions.

Explanation:-This option is incorrect because, in itself, a regular CloudFormation template is not capable of directly deploying stacks across multiple accounts or regions even if you used the Mappings section and the Fn::FindInMap intrinsic function. You can only accomplish this if you used a stack set to deploy your resources to multiple regions.

Q24) A company needs to migrate an on-premises MySQL database to Amazon RDS with minimal downtime to avoid production outages. The Database Specialist will use AWS DMS with a full load plus CDC task for continuous replication. The tables of the source database could contain multiple rows with the same value.
How should the Database Specialist ensure that there will be no duplicate records on the target tables during migration?

● Ensure that each of the target tables has a foreign key.

Explanation:-This option is incorrect because you have to ensure that the target table must have a primary key or unique index to avoid any duplicate data. Having a foreign key in the target table will not produce any effect.

✔ Ensure that each of the target tables has a primary key or unique index.

Explanation:-You can create an AWS DMS task that captures ongoing changes to the source data store. You can do this capture while you are migrating your data. You can also create a task that captures ongoing changes after you complete your initial (full-load) migration to a supported target data store. This process is called ongoing replication or change data capture (CDC). AWS DMS uses this process when replicating ongoing changes from a source data store. This process works by collecting changes to the database logs using the database engine's native API.

Each source engine has specific configuration requirements for exposing this change stream to a given user account. Most engines require some additional configuration to make it possible for the capture process to consume the change data in a meaningful way, without data loss. For example, Oracle requires the addition of supplemental logging, and MySQL requires row-level binary logging (bin logging). To read ongoing changes from the source database, AWS DMS uses engine-specific API actions to read changes from the source engine's transaction logs.

Running a full load and CDC task can create duplicate records on target tables that don't have a primary key or unique index. To avoid duplicating records on target tables during full load and CDC tasks, make sure that target tables have a primary key or unique index.

References:
https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Troubleshooting.html#CHAP_Troubleshooting.General.DuplicateRecords
https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Task.CDC.html
https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Introduction.html

○ Ensure that each of the target tables has a log sequence number (LSN) column. Configure the AWS DMS to read the source changes in the transaction log based on the log sequence number (LSN).

Explanation:-This option is incorrect because having an LSN column will not protect the target tables from having duplicate records. In addition, it is the Microsoft SQL Server database that reads the source changes in the transaction log based on the log sequence number (LSN). Take note that the scenario is using an on-premises MySQL database as the source.

○ Ensure that each of the target tables has a primary key created on a LOB column. Run the migration in FULL LOB mode.

Explanation:-This option is incorrect because AWS DMS doesn't support the replication of primary keys that are LOB data types. Furthermore, using FULL LOB mode is irrelevant in avoiding duplicate data during migration.

Q25) A company has been running a critical system using Amazon Aurora with PostgreSQL. The development team needs at least eight new development environments and intends to perform both read and write transactions. The manager asked whether it is possible to refresh the new development databases with the latest copy of the production environment upon request.
What is the most cost-effective solution to deploy their requirements in the shortest amount of time?

✓ Use the Amazon Aurora cloning feature to make multiple clones from the same DB cluster.

Explanation:-Using database cloning, you can quickly and cost-effectively create clones of all of the databases within an Aurora DB cluster. The clone databases require only minimal additional space when first created. Database cloning uses a copy-on-write protocol, in which data is copied at the time that data changes, either on the source databases or the clone databases. You can make multiple clones from the same DB cluster. You can also create additional clones from other clones. You can use database cloning in a variety of use cases, especially when you don't want to have an impact on your production environment.

Some examples are the following:
- Experiment with and assess the impact of changes, such as schema changes or parameter group changes.
- Perform workload-intensive operations, such as exporting data or running analytical queries.
- Create a copy of a production DB cluster in a non-production environment for development or testing.

Take note of the keywords "cost-effective" and "shortest amount of time" in the question.

○ Create the development environment with Amazon Aurora Serverless.

Explanation:-This option is incorrect because it cannot be refreshed with the latest copy of the production environment. You need to create a new cluster by restoring it from the latest snapshot of the source DB cluster.

○ Create new Aurora Replicas and provide each project initiative with a dedicated instance endpoint.

Explanation:-This option is incorrect because users can only perform read operations on a replica.

○ Create a CloudFormation template that allows you to create a new Aurora DB instance from the latest database snapshot.

Explanation:-This option is incorrect because it will cost more than cloning the cluster since cloning uses the copy-on-write protocol. It will also take more time to create the cluster.

Q26) A Database Specialist was assigned to migrate an Amazon RDS DB Instance for MySQL to Amazon Aurora. Unfortunately, the records system using the 500-GB database can only afford near-zero downtime to migrate all data to the target database. The manager confirmed that the application has a proper connection pooling mechanism.
What is the most cost-effective solution to meet this requirement?

○ Create a new Amazon Aurora DB Cluster. Use MySQL binary logging replication to migrate the data.

Explanation:-This option is incorrect. Although this solution works, it will require more effort than using the Aurora MySQL Read Replica method, which mainly performs the same steps.

✓ Create an Aurora MySQL Read Replica of the RDS DB instance and promote it once the replication lag drops to 0.

Explanation:-Without the option to take the database offline, an approach based on replication is generally the best solution. Aurora uses the MySQL DB engines' binary log replication functionality to create a special type of DB cluster called an Aurora Read Replica for a source MySQL DB instance. Updates made to the source MySQL DB instance are asynchronously replicated to the Aurora Read Replica.

Today, RDS allows you to create an Aurora Read Replica. The migration process begins by creating a DB snapshot of the existing DB Instance and then using it as the basis for a fresh Aurora Read Replica. After the replica has been set up, replication is used to bring it up to date with respect to the source. Once the replication lag drops to 0, the replication is complete. At this point, you can make the Aurora Read Replica a standalone Aurora DB cluster and point your client applications at it.

In this scenario, the question asks for a full data migration method: 1) the most cost-effective solution, and 2) near-zero downtime.

References:
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Migrating.RDSMySQL.Replica.html
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Migrating.html

○ Restore an Amazon Aurora DB Cluster from a DB snapshot.

Explanation:-This option is incorrect because it will not restore the latest data from the source DB instance to the new Aurora DB Cluster. The solution requires an additional replication solution (e.g., binlogging MySQL replication) to transfer all data to the new database. Take note that this could be the quickest method to migrate a snapshot of the source database to the target cluster.

○ Use the clone feature of RDS MySQL DB Instance to create an Aurora clone.

Q27) A company based in Manila intends to migrate a learning management system from an on-premises Oracle database to an Amazon Aurora PostgreSQL DB Cluster. The manager asks the Database Specialist to review the license requirements and hardware configurations of both source and target database instances. The report should also include an estimate of the effort it will take to convert the code.
What should the database specialist do to meet these

○ Use AWS Schema Conversion Tool (AWS SCT) data extraction agent. By default, it will generate a report that lists down any migration issue.

Explanation:-This option is incorrect because the data extraction agent helps perform data transformation for certain scenarios (e.g., Redshift, DynamoDB). It does not help in reviewing the components.

✓ Use AWS Schema Conversion Tool (AWS SCT) and create a database migration assessment report.

Explanation:-An important part of the AWS Schema Conversion Tool is the database migration assessment report that it generates to help you convert your schema. The report summarizes all of the schema conversion tasks and details the action items for schema that can't be converted to the DB engine of your target DB instance. You can view the report in the application. To do so, export it as a comma-separated value (CSV) or PDF file.

The migration assessment report includes:
1) License evaluation.
2) Cloud support, indicating any features in the source database not available on the target.
3) Current source hardware configuration

4) Recommendations, including conversion of server objects, backup suggestions, and linked server changes
The report also includes information about an Amazon RDS DB instance if you selected Amazon RDS as your target, including the following:
1) The currently used storage size and maximum storage size for the DB instance.
2) The current number of databases on the DB instance and the maximum number of databases allowed on the DB instance.
3) A list of database services and server objects that are not available on the DB instance.
4) A list of databases that are currently participating in replication. Amazon RDS doesn't support replication.
It includes a report that estimates of the amount of effort that it will take to write the equivalent code for your target DB instance that can't be converted automatically. If you use AWS SCT to migrate your existing schema to an Amazon RDS DB instance, the report can help you analyze requirements for moving to the AWS Cloud and changing your license type.
References:
https://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/CHAP_AssessmentReport.html
https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Validating.html
https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Tasks.AssessmentReport.html

- Enable AWS DMS data validation on the task to compare the source and target records and report any issues.

Explanation:-This option is incorrect because it does not meet the requirements. The AWS DMS data validation task compares the source and target records and reports any mismatches. In addition, for a CDC-enabled task, AWS DMS compares the incremental changes and reports any mismatches.

- Enable and start an AWS Database Migration Service (AWS DMS) premigration assessment.

Explanation:-This option is incorrect because a premigration assessment evaluates specified components of a database migration task to help identify any problems that might prevent a migration task from running as expected. This assessment gives you a chance to identify issues before you run a new or modified task. You can then fix problems before they occur while running the migration task itself. This can prevent delays in completing a given database migration needed to repair data and your database environment.

Q28) A startup is developing a ride-hailing mobile app that stores GPS coordinates of all the rides made by its users. Real-time statistics must also be provided with microsecond latency for reporting purposes. A Database Specialist needs to deploy a highly scalable and fault-tolerant database that can meet the app workload with minimal development effort.
Which solution can satisfy the above requirements?

- Launch an Amazon DynamoDB table with Auto Scaling enabled to provide microsecond latency.

Explanation:-This option is incorrect because DynamoDB Auto Scaling is primarily used to automate capacity management for your tables and global secondary indexes.

- Launch an Amazon Aurora database with cluster cache management. Set up synchronization between the writer DB instance's buffer cache and the designated reader's warm buffer cache.

Explanation:-This option is incorrect because the cluster cache management feature only allows fast recovery of the writer DB instance in your Aurora PostgreSQL clusters if there's a failover. It doesn't provide microsecond latency to your database.

- ✅ Launch an Amazon DynamoDB table with DAX to provide microsecond latency.

Explanation:-Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache that can reduce Amazon DynamoDB response times from milliseconds to microseconds, even at millions of requests per second.
DAX does all the heavy lifting required to add in-memory acceleration to your DynamoDB tables, without requiring developers to manage cache invalidation, data population, or cluster management.
With DAX, you can focus on building great applications for your customers without worrying about performance at scale. You do not need to modify application logic because DAX is compatible with existing DynamoDB API calls.

- Launch an Amazon Aurora database cluster with Database Activity Streams to provide microsecond response time.

Explanation:-This option is incorrect because the Database Activity Streams feature just provides a near real-time data stream of the database activity in your relational database. It's not capable of reducing the response times of your database to microseconds, unlike DynamoDB DAX.

Q29) A database specialist was assigned to migrate a learning management system from an on-premises Oracle database to an Amazon RDS MySQL database using the AWS Database Migration Service (DMS). Given a 10-hour maintenance window for the full load migration activity, the specialist discovered that half of the tables inside the database contained columns of LOB data types. The largest LOB column size found in a table is 300MB, while the rest had at most, LOB column sizes of 10MB. These tables do not have referential integrity constraints against each other, and all tables have primary keys.
How should the database specialist configure the AWS DMS task to improve the speed of LOB data migration?

- Configure an AWS DMS task to migrate the tables without LOBs. Meanwhile, use AWS Schema Conversion Tool to migrate LOB data to the target RDS instance.

Explanation:-This option is incorrect because AWS SCT is used to help convert schema for heterogeneous migration. It does not help improve the performance of LOB migration.

- Configure an AWS DMS task to migrate the tables without LOBs. Configure another AWS DMS task using full LOB mode to migrate LOB data to the target RDS instance.

Explanation:-This option is incorrect because it migrates LOB data slower than in limited LOB mode. In full LOB mode, AWS DMS migrates all LOBs from source to target regardless of size. In this configuration, AWS DMS has no information about the maximum size of LOBs to expect. Thus, LOBs are migrated one at a time, piece by piece. Full LOB mode can be quite slow. Use this option when the LOB sizes are not deterministic.

- ✅ Configure an AWS DMS task to migrate the tables without LOBs. Configure another AWS DMS task using limited LOB mode with a LobMaxSize setting of 300MB to migrate LOB data to the target RDS instance.

Explanation:-When you migrate data from one database to another, you might take the opportunity to rethink how your LOBs are stored, especially for heterogeneous migrations. In limited LOB mode, you set a maximum size LOB that AWS DMS should accept. Doing so allows AWS DMS to pre-allocate memory and load the LOB data in bulk. LOBs that exceed the maximum LOB size are truncated, and a warning is issued to the log file. In limited LOB mode, you get significant performance gains over full LOB mode. We recommend that you use limited LOB mode whenever possible, especially when the LOB size is deterministic. Note, however, that if the LOB sizes are significantly large (e.g., 3GB), performance could be better if you use a DMS task in full lob mode with inline lob mode settings.

- Configure an AWS DMS task to migrate the tables without LOBs. Meanwhile, use Oracle LogMiner to migrate LOB data to the target RDS instance.

Explanation:-This option is incorrect because Oracle LogMiner is used, in this situation, for change data capture processes. The question asks how to improve the speed for full load migration.

Q30) A Database Specialist migrated an on-premises MySQL database to an Amazon RDS for MySQL DB instance using AWS DMS. The RDS database needed to be encrypted at rest using a CMK in AWS KMS. Due to time constraints, the database encryption must be done immediately for the project to proceed.
What approach must the Database Specialist implement to meet this requirement?

- ✅ Create a snapshot of the unencrypted Amazon RDS MySQL DB instance and then create an encrypted copy of that snapshot. Launch a new RDS instance by restoring the encrypted snapshot copy then terminate the unencrypted RDS DB instance.

Explanation:-You can copy a snapshot that has been encrypted using an AWS KMS encryption key. If you copy an encrypted snapshot, the copy of

the snapshot must also be encrypted. If you copy an encrypted snapshot within the same AWS Region, you can encrypt the copy with the same KMS encryption key as the original snapshot, or you can specify a different KMS encryption key. If you copy an encrypted snapshot across Regions, you can't use the same KMS encryption key for the copy as used for the source snapshot, because KMS keys are Region-specific. Instead, you must specify a KMS key valid in the destination AWS Region. The source snapshot remains encrypted throughout the copy process.
You can also encrypt a copy of an unencrypted snapshot. This way, you can quickly add encryption to a previously unencrypted DB instance. That is, you can create a snapshot of your DB instance when you are ready to encrypt it, and then create a copy of that snapshot and specify a KMS encryption key to encrypt that snapshot copy. You can then restore an encrypted DB instance from the encrypted snapshot.
The following limitations exist for Amazon RDS encrypted DB instances:
- You can only enable encryption for an Amazon RDS DB instance when you create it, not after the DB instance is created. As a workaround, you can encrypt a copy of an unencrypted DB snapshot, then restore a DB instance from the encrypted snapshot.
- DB instances that are encrypted can't be modified to disable encryption.
- You can't have an encrypted read replica of an unencrypted DB instance or an unencrypted read replica of an encrypted DB instance.
- Encrypted read replicas must be encrypted with the same key as the source DB instance when both are in the same AWS Region.
- You can't restore an unencrypted backup or snapshot to an encrypted DB instance.
- To copy an encrypted snapshot from one AWS Region to another, you must specify the KMS key identifier of the destination AWS Region. This is because KMS encryption keys are specific to the AWS Region that they are created in.
For an Amazon RDS encrypted DB instance, all logs, backups, and snapshots are encrypted. A read replica of an Amazon RDS encrypted instance is also encrypted using the same key as the primary DB instance when both are in the same AWS Region. If the primary DB instance and read replica are in different AWS Regions, you encrypt using the encryption key for that AWS Region.
References:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.Encryption.html
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CopySnapshot.html

- Create an encrypted read replica of the Amazon RDS MySQL DB instance. Promote the read replica as an encrypted standalone DB instance then terminate the unencrypted RDS DB instance.

Explanation:-This option is incorrect because you can't have an encrypted read replica of an unencrypted DB instance.

- On the Amazon RDS console, select the unencrypted database and toggle the Enable encryption option to automatically encrypt the existing database.

Explanation:-This option is incorrect because you can only enable encryption for an Amazon RDS DB instance when you create it, not after the DB instance is created.

- Launch a new encrypted Amazon RDS MySQL DB instance. Use AWS DMS to migrate the data of the unencrypted Amazon RDS MySQL DB instance to the encrypted database.

Explanation:-This option is incorrect. Although this option is feasible, the entire migration process takes a lot of time. It is still faster to create a snapshot of the unencrypted DB instance then create an encrypted snapshot copy.

Q31) A Database Specialist manages an Amazon DynamoDB table for a mobile application. Given the Security Team's new requirements, the Specialist needed to capture the activity of the table and create an audit trail of the modifications made to the items. These records should be securely stored for at least 6 months.
Which action should the Database Specialist take to meet these requirements?

- Use AWS CloudTrail. Create a trail that captures the DynamoDB table modifications and store the logs in Amazon S3.

Explanation:-This option is incorrect. Although CloudTrail captures the activity when a DynamoDB table was updated, it does not capture the modifications made to the table items.

- Create a CloudWatch alarm that sends the necessary information whenever the DynamoDB table is modified.

Explanation:-This option is incorrect because CloudWatch alarms cannot track and store the table modification information. CloudWatch is used to monitor DynamoDB performance.

✅ Enable DynamoDB Streams on the table. Write a Lambda function that copies each stream record to an Amazon S3 bucket.

Explanation:-Many applications can benefit from the ability to capture changes to items stored in a DynamoDB table, at the point in time when such changes occur. DynamoDB Streams enables solutions such as these, and many others. DynamoDB Streams captures a time-ordered sequence of item-level modifications in any DynamoDB table and stores this information in a log for up to 24 hours. Applications can access this log and view the data items as they appeared before and after they were modified, in near-real-time.
If you enable DynamoDB Streams on a table, you can associate the stream Amazon Resource Name (ARN) with an AWS Lambda function that you write. Immediately after an item in the table is modified, a new record appears in the table's stream. AWS Lambda polls the stream and invokes your Lambda function synchronously when it detects new stream records. The Lambda function can perform any actions you specify, such as sending a notification or initiating a workflow. For example, you can write a Lambda function to simply copy each stream record to persistent storage, such as Amazon Simple Storage Service (Amazon S3), to create a permanent audit trail of write activity in your table.
References:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.Lambda.html
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html

- Enable DynamoDB Streams on the table. Use DynamoDB Streams Kinesis Adapter to store the records.

Explanation:-This option is incorrect. Although DynamoDB Streams Kinesis Adapter provides you with great features to process the stream records in real-time, it cannot store the records for 6 months by itself. It needs to transfer the records to an additional persistent store such as Amazon S3.

Q32) A Database Specialist plans to migrate an on-premises non-relational key-value database to a fully-managed AWS service that can take care of tasks such as scaling databases and hardware provisioning. The Application Manager also requires a cost-effective database that could manage unknown workloads and unpredictable user traffic.
Which solution should the Database Specialist consider first?

- Amazon ElastiCache Cluster Redis (Cluster mode enabled)

Explanation:-This option is incorrect because ElastiCache requires manual scaling management. It is also not designed for unpredictable workloads.

✅ Amazon DynamoDB table with on-demand capacity mode

Explanation:-Amazon DynamoDB is a fast and flexible nonrelational database service for any scale. DynamoDB enables customers to offload the administrative burdens of operating and scaling distributed databases to AWS so that they don't have to worry about hardware provisioning, setup, and configuration, throughput capacity planning, replication, software patching, or cluster scaling.
Amazon DynamoDB on-demand is a flexible billing option capable of serving thousands of requests per second without capacity planning. DynamoDB on-demand offers pay-per-request pricing for read and write requests so that you pay only for what you use. When you choose on-demand mode, DynamoDB instantly accommodates your workloads as they ramp up or down to any previously reached traffic level.
On-demand mode is a good option if any of the following are true:
- You create new tables with unknown workloads.
- You have unpredictable application traffic.
- You prefer the ease of paying for only what you use.
References:

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ReadWriteCapacityMode.html
https://aws.amazon.com/blogs/aws/amazon-dynamodb-on-demand-no-capacity-planning-and-pay-per-request-pricing/

- Amazon DynamoDB table with provisioned capacity mode

Explanation:-This option is incorrect becuase Provisioned capacity mode works well with predictable workloads.

- Amazon DocumentDB cluster

Explanation:-This option is incorrect becuase as per the requirement, the AWS database must take care of certain tasks such as scaling and storage provisioning. DocumentDB requires manual management to scale the database, unlike DynamoDB.

Q33) A Database Specialist is managing a CRM application that uses an Amazon Aurora for PostgreSQL DB cluster with a primary DB instance that was configured to have a higher instance class than its four Aurora Replicas. Unexpectedly, a failover event occurs and the application team manager raised concerns that the performance was poor for several minutes. The manager also mentioned that the application servers in all Availability Zones are healthy.
What should the Database Specialist do to eliminate the performance issue raised by the manager?

- Configure one Aurora Replica to have the same instance class as the primary DB instance. Implement Aurora PostgreSQL DB cluster cache management. Set the failover priority to tier-0 for the primary DB instance and tier-1 for all Aurora Replicas.

Explanation:-This option is incorrect because to enable fast recovery after failover, an Aurora Replica with the same instance class as the primary DB instance should be configured with a promotion tier priority of tier-0.

- Deploy an AWS Lambda function that identifies which instance has failed, and then promote one Aurora Replica to be the primary DB instance. Configure an Amazon RDS event subscription to send a notification to an Amazon SNS topic to which the Lambda function is subscribed and gets triggered whenever the database is unresponsive.

Explanation:-This option is incorrect. Although this can potentially automate the failover of a database instance, it is not required for Amazon Aurora which has an automatic failover feature. Furthermore, the question asks for a solution that can prevent the application performance issue.

- Configure two Aurora Replicas to the same instance class as the primary DB instance. Enable cache coherence on the DB cluster, set the primary DB instance failover priority to tier-0, and assign a failover priority of tier-1 to the Aurora Replicas.

Explanation:-This option is incorrect because to enable fast recovery after failover, the DB cluster parameter for cluster cache management has to be enabled, not cache coherence. Furthermore, one Aurora Replica needs a promotion tier priority of tier-0.

- ✅ Configure all Aurora Replicas to have the same instance class as the primary DB instance. Implement Aurora PostgreSQL DB cluster cache management. Set the failover priority to tier-0 for the primary DB instance and one replica with the same instance class. Set the failover priority to tier-1 for the other Aurora Replicas.

Explanation:-For fast recovery of the writer DB instance in your Aurora PostgreSQL clusters if there's a failover, use cluster cache management for Amazon Aurora PostgreSQL. Cluster cache management ensures that application performance is maintained if there's a failover.
In a typical failover situation, you might see a temporary but large performance degradation after failover. This degradation occurs because when the failover DB instance starts, the buffer cache is empty. An empty cache is also known as a cold cache. A cold cache degrades performance because the DB instance has to read from the slower disk, instead of taking advantage of values stored in the buffer cache.
With cluster cache management, you set a specific reader DB instance as the failover target. Cluster cache management ensures that the data in the designated reader's cache is kept synchronized with the data in the writer DB instance's cache. The designated reader's cache with prefilled values is known as a warm cache. If a failover occurs, the designated reader uses values in its warm cache immediately when it's promoted to the new writer DB instance. This approach provides your application with much better recovery performance.
To configure cluster cache management, perform the following steps:
1) Set apg_ccm_enabled cluster parameter to 1 for your Aurora PostgreSQL DB Cluster. Note that you cannot change parameters in a default parameter group. Note that a cluster parameter affects all instances in your cluster.
2) Modify the primary/writer DB instance and set its promotion priority to tier-0.
3) Set one reader DB instance with the same instance class as the writer db instance class.
4) Set that reader DB instance's promotion tier priority to tier-0.
(Allow at least 1 minute after completing these steps for cluster cache management to be fully operational.)
References:
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraPostgreSQL.cluster-cache-mgmt.html
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraPostgreSQL.Reference.html#AuroraPostgreSQL.Reference.Parameters.Cluster

---

Q34) A financial services company is planning to build a new fraud detection system. The Database Specialist was assigned to create a fully-managed data storage solution in AWS that can execute fast queries to show relationships among financial transactions and detect whenever a person is using the same identity-related information as past fraud cases.
Which database service would best meet these requirements?

- Amazon RDS

Explanation:-This option is incorrect because it would require a huge effort to design a schema that can achieve the same benefit.

- Amazon Timestream

Explanation:-This option is incorrect because this database service is designed to easily store and analyze trillions of events per day - data that measures how things change over time.

- Amazon DynamoDB

Explanation:-This option is incorrect because DynamoDB is ideal for key-value or document stores that leverage on both noSQL and SQL queries (using Athena and S3).

- ✅ Amazon Neptune

Explanation:-Graph databases are useful for connected, contextual, relationship-driven data. Graph databases have advantages over relational databases for certain use cases—including social networking, recommendation engines, and fraud detection—when you want to create relationships between data and quickly query these relationships. They can represent how entities relate by using actions, ownership, parentage, etc.
There are a number of challenges to building these types of applications using a relational database. It requires you to have multiple tables with multiple foreign keys. The SQL queries to navigate this data require nested queries and complex joins that quickly become unwieldy. And the queries don't perform well as your data size grows over time.
Amazon Neptune uses graph structures such as nodes (data entities), edges (relationships), and properties to represent and store data. Edges can describe parent-child relationships, actions, product recommendations, purchases, and so on. A relationship, or edge, is a connection between two vertices that always has a start node, end node, type, and direction.
Reference:
https://docs.aws.amazon.com/neptune/latest/userguide/graph-database.html

---

Q35) A serverless application uses an Amazon DynamoDB table to store the collected user data. The table needs to be configured with DynamoDB Streams to monitor any table changes. Kinesis Firehose will be used to load the stream data, which contains the original value of the overwritten item in the table into an Amazon S3 bucket. When an item is updated, DynamoDB should only send a copy of the item's previous value to S3 and then maintain the new value in the table.
What Stream View type should be used to meet the above requirements?

✅ OLD_IMAGE

Explanation:-DynamoDB Streams provides a time-ordered sequence of item-level changes in any DynamoDB table. The changes are de-duplicated and stored for 24 hours. Applications can access this log and view the data items as they appeared before and after they were modified, in near real-time.

Amazon DynamoDB is also integrated with AWS Lambda so that you can create triggers—pieces of code that automatically respond to events in DynamoDB Streams. With triggers, you can build applications that react to data modifications in DynamoDB tables.

When an item in the table is modified, StreamViewType determines what information are written to the stream for this table. Valid values for StreamViewType are:

KEYS_ONLY — Only the key attributes of the modified item.

NEW_IMAGE — The entire item, as it appears after it was modified.

OLD_IMAGE — The entire item, as it appeared before it was modified.

NEW_AND_OLD_IMAGES — Both the new and the old images of the item.

For the OLD_IMAGE type, the entire item, which has the previous value as it appeared before it was modified, is written to the stream.

References:

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html

https://docs.amazonaws.cn/en_us/amazondynamodb/latest/developerguide/Streams.Lambda.html

https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_StreamSpecification.html

⚪ KEYS_ONLY

Explanation:-This option is incorrect because it will only write the key attributes of the modified item to the stream. This is wrong since the question states that values should be copied as well.

⚪ NEW_IMAGE

Explanation:-This option is incorrect because it will configure the stream to write the entire item with its new value as it appears after it was modified. This option is wrong since the stream should capture the item's pre-modified values and not the new one.

⚪ NEW_AND_OLD_IMAGES

Explanation:-This option is incorrect. Although it writes the new values of the item in the stream, it also includes the old one as well. Since this type will send both the new and the old item images of the item to the stream, this option is wrong. Remember that the scenario indicates that DynamoDB should only send a copy of the item's previous value to the S3 bucket, and not the new value in the DynamoDB table. The most suitable one to use here is the OLD_IMAGE type.

Q36) A product owner consults with a database specialist about creating a new mobile application, which he wants to build inside the cloud. They intend to store the data inside JSON documents without the inconvenience of managing the scalability or security of the database. They also expect this database to handle a high volume of both read and write requests per minute from an application that will have users in Manila, Australia, and London.
Which database service is the most cost-effective solution that can achieve these requirements?

⚪ Amazon DynamoDB table

Explanation:-This option is incorrect because the service needs to serve multiregional requirements.

⚪ Amazon DocumentDB

Explanation:-This option is incorrect. Although it is designed to store JSON documents and manage high volumes of read and write requests, DocumentDB still requires management of the operations, including scaling and security.

✅ Amazon DynamoDB Global Table

Explanation:-Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-regional, multimaster, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. Since the application will have users coming from different regions, a DynamoDB global table can cater to the multiregional performance requirements of the application.

Amazon DynamoDB global tables provide a fully managed solution for deploying a multi-regional, multi-master database, without having to build and maintain your replication solution. With global tables, you can specify the AWS Regions where you want the table to be available. DynamoDB performs all of the necessary tasks to create identical tables in these Regions and propagate ongoing data changes to all of them.

In this scenario, the answer needs to fulfill five requirements:

1) Store JSON documents

2) No requirement to manage scaling services or security

3) Manage high volume both read and write requests

4) Multiregional

5) Most cost-effective solution

References:

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GlobalTables.html

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html

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

https://aws.amazon.com/documentdb/

⚪ Amazon Aurora Global Database

Explanation:-This option is incorrect. Although it meets the product owner's requirements, it is significantly less cost-effective than a DynamoDB global table. Amazon Aurora is designed for more traditional enterprise applications.

Q37) The Database team is managing an Amazon Aurora DB cluster with MySQL compatibility for an online learning platform. An application manager asked for a full copy of the cluster in a new environment, where they plan to develop and test new monthly features for the next two years. Probing further, the application team will only use it sparingly during office hours and does not expect to execute batch jobs that run overnight.
What is the most cost-effective solution that meets this requirement?

⚪ Restore a snapshot of the Aurora Cluster to a new Amazon RDS DB Instance for MySQL.

Explanation:-This option is incorrect. If the cost to manage an RDS is compared with the actual workload, it will most likely be less cost-effective than using an Aurora Serverless environment. For example, while Aurora Serverless charges the user only while it is running, an RDS DB instance continues charging for the storage even if it is stopped.

⚪ Use the Amazon Aurora cloning feature to create a new MySQL DB environment.

Explanation:-This option is incorrect. In this situation, the environment will be used to develop new features and thus, most likely to perform frequent write operations. A clone is just a simple and cost-effective option for a temporary development environment that is necessary for a short period.

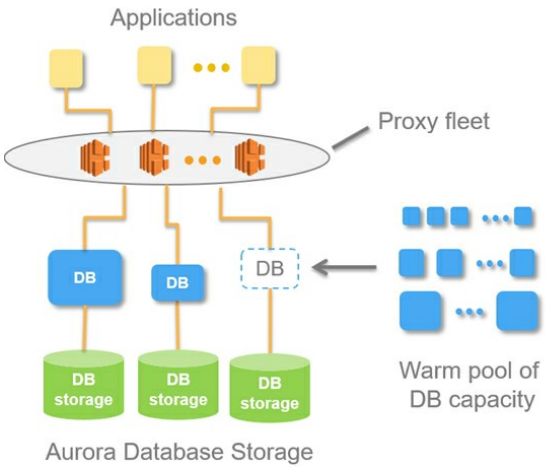
✅ Restore a snapshot of the Aurora Cluster to configure an Aurora Serverless DB Cluster.

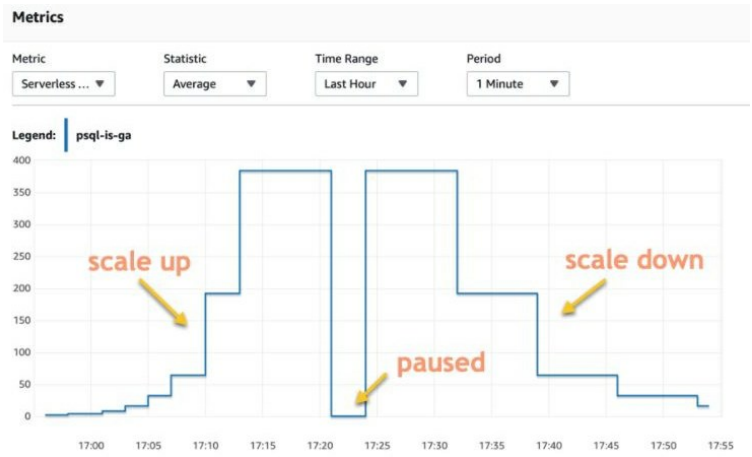
Explanation:-

Amazon Aurora Serverless is an on-demand, auto-scaling configuration for Amazon Aurora. An Aurora Serverless DB cluster is a DB cluster that automatically starts up, shuts down, and scales up, or down its compute capacity based on your application's needs.

Aurora Serverless provides a relatively simple, cost-effective option for intermittent or unpredictable workloads. It can offer this because it automatically

starts up, scales compute capacity to match your application's usage and shuts down when it's not in use. More importantly, Aurora Serverless is a cost-effective option for an application that is only used for a few minutes several times per day or week, such as a low-volume blog site. With Aurora Serverless, you pay for only the database resources that you consume on a per-second basis.





In this scenario, the application manager asks for what sounds like a development environment filled with data of the production environment. Besides, the application team does not plan to regularly access the database and run day-long workloads.
References:
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-serverless.html
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-multi-master.html

● Migrate to an Amazon Aurora Multi-Master Cluster and configure a WRITER custom endpoint for the application team to use the new writer instance.
Explanation:-This option is incorrect. The Multi-Master Cluster is designed to allow more than one writer instance accessing the same cluster volume. It is a good option for multitenant and sharded databases in a production environment but it will not meet the application manager's

---

Q38) A game development startup is building a mobile game app with Amazon DynamoDB as its data store. The Database Specialist must design the DynamoDB table to support various game features such as updating the player's score in real-time, storing the game session history, and fetching a player's score details for a particular game session. Each player has a unique player_id and there is also a unique game_session_id for each game.
Which of the following is the most suitable design for the DynamoDB table?

● Set up a composite primary key where the partition key is game_session_id and the sort key is: player_id.
Explanation:-This option is incorrect because this should be the other way around. The player_id must be set as the partition key and the game_session_id as the sort key.
● Set up a global secondary index where the partition key is player_id and the sort key is: game_session_id.
Explanation:-This option is incorrect because a global secondary index (GSI) is primarily used if your application requires a query that uses a variety of different attributes other than the provided partition and sort key of the table. The use cases in this scenario can be easily supported by using a composite primary key without any additional secondary indexes.
✅ Set up a composite primary key where the partition key is player_id and the sort key is: game_session_id.
Explanation:-When you create a table, in addition to the table name, you must specify the primary key of the table. The primary key uniquely identifies each item in the table, so that no two items can have the same key.
DynamoDB supports two different kinds of primary keys:
Partition key – A simple primary key, composed of one attribute known as the partition key.
Partition key and sort key – Referred to as a composite primary key, this type of key is composed of two attributes. The first attribute is the partition key, and the second attribute is the sort key.
DynamoDB uses the partition key value as input to an internal hash function. The output from the hash function determines the partition (physical storage internal to DynamoDB) where the item will be stored. All items with the same partition key value are stored together, in sorted order by sort key value.
In a table that has a partition key and a sort key, it's possible for two items to have the same partition key value. However, those two items must have different sort key values. A composite primary key gives you additional flexibility when querying data.
The partition key of an item is also known as its hash attribute. The term hash attribute derives from the use of an internal hash function in DynamoDB that evenly distributes data items across partitions, based on their partition key values.
The sort key of an item is also known as its range attribute. The term range attribute derives from the way DynamoDB stores items with the same partition key physically close together, in sorted order by the sort key value.
References:
https://aws.amazon.com/blogs/database/amazon-dynamodb-gaming-use-cases-and-design-patterns/
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.CoreComponents.html#HowItWorks.CoreComponents.PrimaryKey

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/SecondaryIndexes.html

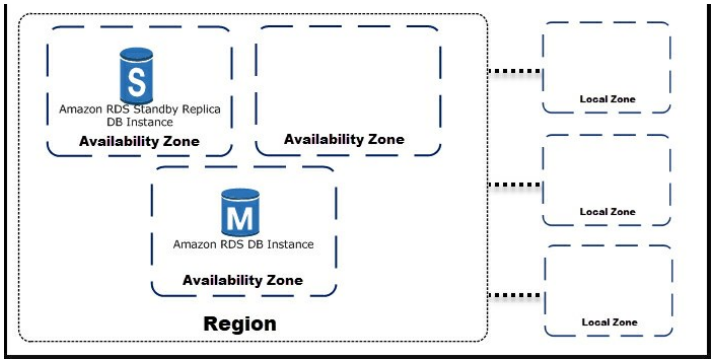
● Set up a local secondary index where the partition key is game_session_id and the sort key is: player_id. Ensure that it has the same partition and sort key as the table.

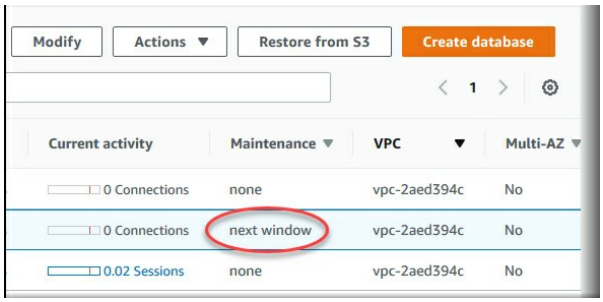
Explanation:-This option is incorrect because using a local secondary index (LSI) is not required in this scenario since an alternative sort key is not warranted to support the specified use cases. Moreover, the LSI must have a different sort key as the table.

Q39) A Database Specialist is planning to create a database for a critical enterprise application using Amazon RDS DB for MariaDB. The application owner wants to reduce the impact of maintenance events in the application and ensure that the database does not turn completely offline during operating system updates.
Which steps should the Database Specialist take to minimize downtime due to maintenance? (SELECT TWO.)

✅ Create the RDS DB instance with Multi-AZ deployment.

Explanation:-
Periodically, Amazon RDS performs maintenance on Amazon RDS resources. Some maintenance items require that Amazon RDS take your DB instance offline for a short time. Maintenance items that require a resource to go offline include the required operating system or database patching.
After OS maintenance is scheduled for the next maintenance window, maintenance can be postponed by adjusting your preferred maintenance window. Maintenance can also be deferred, if desired. To minimize downtime, modify the Amazon RDS DB instance to a Multi-AZ deployment. In a Multi-AZ deployment, Amazon RDS applies operating system updates by following these steps:
1) Perform maintenance on the standby.
2) Promote the standby to primary.
3) Perform maintenance on the old primary, which becomes the new standby.
Take note that this does not apply for database engine updates. If database engine modification occurs, both the primary and secondary DB instances are shut down at the same time.





References:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_UpgradeDBInstance.Maintenance.html
https://aws.amazon.com/premiumsupport/knowledge-center/rds-required-maintenance/

✅ Set the RDS maintenance window to perform maintenance items during a low activity period.

Explanation:-Periodically, Amazon RDS performs maintenance on Amazon RDS resources. Some maintenance items require that Amazon RDS take your DB instance offline for a short time. Maintenance items that require a resource to go offline include the required operating system or database patching.
After OS maintenance is scheduled for the next maintenance window, maintenance can be postponed by adjusting your preferred maintenance window. Maintenance can also be deferred, if desired. To minimize downtime, modify the Amazon RDS DB instance to a Multi-AZ deployment. In a Multi-AZ deployment, Amazon RDS applies operating system updates by following these steps:
1) Perform maintenance on the standby.
2) Promote the standby to primary.
3) Perform maintenance on the old primary, which becomes the new standby.
Take note that this does not apply for database engine updates. If database engine modification occurs, both the primary and secondary DB instances are shut down at the same time.
References:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_UpgradeDBInstance.Maintenance.html
https://aws.amazon.com/premiumsupport/knowledge-center/rds-required-maintenance/

● Instead of using automatic snapshots, use AWS Backup to schedule manual backups outside the maintenance window.

Explanation:-This option is incorrect because this does not reduce the impact of maintenance events in an application.

● Create a Read Replica in the Amazon RDS for MariaDB instance.

Explanation:-This option is incorrect because a Read Replica will not automatically failover the application to the replica while the primary DB instance maintenance activity is ongoing.

● Set up an event notification for Amazon RDS that sends a notification via email whenever maintenance is ongoing.

Explanation:-This option is incorrect. Although it is a good practice to create event notifications for such activities, it does not minimize the downtime of maintenance.

Q40) An online auction system has hundreds of users that concurrently update the bid price of an item stored in a DynamoDB table. There are incidents where another user overrode a particular change made by one user. The Database Specialist must ensure that all UpdateItem operation will only succeed if the item attributes meet one or more expected conditions. If a user modifies a specific item in the table, the solution must guarantee that the operation will not affect another user's attempt to change the same item.
What should the Specialist do to meet this requirement?

✅ Add condition expressions in the existing UpdateItem operation.

Explanation:-By default, the DynamoDB write operations (PutItem, UpdateItem, DeleteItem) are unconditional: each of these operations will overwrite an existing item that has the specified primary key.

DynamoDB optionally supports conditional writes for these operations. A conditional write will succeed only if the item attributes meet one or more expected conditions. Otherwise, it returns an error. Conditional writes are helpful in cases where multiple users attempt to modify the same item.

For example, by adding a conditional expression that checks if the current value of the item is still the same, you can be sure that your update will not affect the operations of other users:

```
aws dynamodb update-item \
--table-name ProductCatalog \
--key '{"Id":{"N":"1"}}' \
--update-expression "SET Price = :newval" \
--condition-expression "Price = :currval" \
--expression-attribute-values file://tutorialsdojo-attribute-values.json
```

The BatchWriteItem operation puts or deletes multiple items in one or more tables. A single call to BatchWriteItem can write up to 16 MB of data, which can comprise as many as 25 put or delete requests. Individual items to be written can be as large as 400 KB. Take note that the BatchWriteItem operation cannot update items.

⚪ Refactor the application to use the BatchWriteItem operation to update the items.

Explanation:-This option is incorrect because this particular operation is not capable of updating any item. You have to use the UpdateItem operation with condition expressions instead.

⚪ Use projection expressions in the existing UpdateItem operation.

Explanation:-This option is incorrect because this type of expression is just a string that identifies the attributes you want to retrieve during a GetItem, Query, or Scan operation. Take note that the scenario calls for a feature that can be used during an update operation hence, this option is irrelevant.

⚪ Include a ConsistentRead parameter in the existing UpdateItem operation.

Explanation:-This option is incorrect because the ConsistentRead parameter simply determines the read consistency model. This parameter is not useful for writing or updating items. If the parameter is set to true then the operation uses strongly consistent reads; otherwise, the operation uses eventually consistent reads.

Q41) A gaming company has successfully launched a mobile application that uses an Amazon DynamoDB and is becoming increasingly popular for the past year. The product owner plans to release a character change feature and asks the Database Specialist to prepare for the read/write activity surge and tune the database capacity.
What can the Database Specialist do to meet this requirement with the lowest cost?

⚪ Create and configure a DynamoDB Accelerator (DAX) cluster.

Explanation:-This option is incorrect. DAX is designed to speed up and offload reads from the table, potentially saving some costs. However, the mobile application is write-intensive and will not help the performance of the database.

✅ Set the selected DynamoDB table to provisioned capacity mode with auto-scaling enabled.

Explanation:-Amazon DynamoDB delivers low-latency performance at any scale and greatly simplifies database capacity management. After you provision a table, you can change its capacity on the fly. If you discover that your application has become wildly popular overnight, you can easily increase capacity. DynamoDB released auto-scaling to make it easier for you to manage capacity efficiently, and auto-scaling continues to help DynamoDB users lower the cost of workloads that have a predictable traffic pattern.

With on-demand, DynamoDB instantly allocates capacity as it is needed. On-demand is ideal for bursty, new, or unpredictable workloads whose traffic can spike in seconds or minutes, and when underprovisioned capacity would impact the user experience. On-demand is a perfect solution if your team is moving to a NoOps or serverless environment.

DynamoDB Accelerator (DAX) delivers fast response times for accessing eventually consistent data. It is not an ideal solution for applications that are write-intensive, or that do not perform much read activity.

Reference:
https://aws.amazon.com/blogs/database/amazon-dynamodb-auto-scaling-performance-and-cost-optimization-at-any-scale/

⚪ Set the selected DynamoDB table to on-demand mode.

Explanation:-This option is incorrect. Using On-demand Capacity Mode for DynamoDB tables is ideal for unpredictable application traffic, new tables with unknown workloads, and ease of management. However, the likelihood that the team can predict the user traffic after the release is very high for a mobile application that has been running for at least a year. The charges for a table with Provisioned Capacity mode and enabled Autoscaling should cost lower.

⚪ Migrate from DynamoDB to Amazon DocumentDB.

Explanation:-This option is incorrect because the reasons to move to another database service are weak. Furthermore, migration is not very cost-effective.

Q42) A Database Specialist is planning to create a new Amazon DynamoDB table that will have a SongName attribute as its partition key and a Genre attribute as its sort key. The table design must accommodate a feature that will query the partition key but will use the Artist attribute as a different sort key. It also requires a strong read consistency to fetch the latest data.
Which of the following must be done to satisfy the above requirements?

⚪ Launch the DynamoDB table with all the necessary attributes and then add a Global Secondary Index (GSI). Use the SongName attribute as the partition key of the GSI and the Artist attribute as its sort key.

Explanation:-This option is incorrect because the requirement says that the same partition key will be used, but with an alternate sort key which warrants the use of an LSI instead of a GSI. Strong read consistency is also required to fetch the latest data. Bear in mind that the queries on global secondary indexes support eventual consistency only.

✅ Add a Local Secondary Index (LSI) at the same time that the DynamoDB table is created. Configure the LSI to use the SongName attribute as the partition key and the Artist attribute as the sort key.

Explanation:-A local secondary index maintains an alternate sort key for a given partition key value. A local secondary index also contains a copy of some or all of the attributes from its base table; you specify which attributes are projected into the local secondary index when you create the table. The data in a local secondary index is organized by the same partition key as the base table, but with a different sort key. This lets you access data items efficiently across this different dimension. For greater query or scan flexibility, you can create up to five local secondary indexes per table.

Suppose that an application needs to find all of the threads that have been posted within the last three months. Without a local secondary index, the application would have to Scan the entire Thread table and discard any posts that were not within the specified time frame. With a local secondary index, a Query operation could use LastPostDateTime as a sort key and find the data quickly.

To create a Local Secondary Index, make sure that the primary key of the index is the same as the primary key/partition key of the table, just as shown below. Then you must select an alternative sort key which is different from the sort key of the table.

When you request a strongly consistent read, DynamoDB returns a response with the most up-to-date data, reflecting the updates from all prior write operations that were successful. A strongly consistent read might not be available if there is a network delay or outage. Strongly consistent reads are not supported on global secondary indexes.

The primary key of a local secondary index must be composite (partition key and sort key). A local secondary index lets you query over a single

partition, as specified by the partition key value in the query.
Local secondary indexes are created at the same time that you create a table. You cannot add a local secondary index to an existing table, nor can you delete any local secondary indexes that currently exist.
References:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/SecondaryIndexes.html
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/LSI.html

- Create the DynamoDB table first with all the necessary attributes. Once the table has been created, add a Local Secondary Index (LSI) with the SongName attribute as its partition key and the Artist attribute as the sort key.

Explanation:-This option is incorrect. Although using an LSI is right, the implementation is wrong as you cannot add a local secondary index to an already existing table

- Add a Global Secondary Index (GSI) at the same time that the DynamoDB table is created. Configure the SongName attribute as the partition key of the GSI and the Artist attribute as the sort key.

Explanation:-This option is incorrect as an LSI must be used in this scenario. Take note that GSIs can be created at the same time that the DynamoDB table is launched and to an already existing DynamoDB table as well. Therefore, the timing of creating the GSI is irrelevant. This constraint is only applicable to LSIs since these can only be created at the same time the DynamoDB table is launched.

Q43) A company has an application that uses an Amazon RDS database instance in the US East (Ohio) Region, which is encrypted at rest using the AWS Key Management Service (AWS KMS). The Database Specialist must launch a read replica of the RDS database in the US West (Northern California) Region to comply with the company's disaster recovery plan.
What should the Database Specialist do to satisfy the requirement?

- Transfer the existing customer master key (CMK) from US East to US West AWS Region. Create an encrypted, cross-region read replica by specifying the key identifier of the imported CMK in the US West Region.

Explanation:-This option is incorrect because you can't import a CMK from one AWS Region to another. You have to create a new CMK in the new Region instead.

- Launch an encrypted, cross-region read replica in the US West (Northern California) Region by specifying the key identifier of the existing CMK in the US East Region.

Explanation:-This option is incorrect because the keys generated by AWS KMS are only stored and used in the region in which they were created. You can't specify the key identifier of the current CMK in the source Region if you are creating a new read replica in another AWS Region.

✓ Use AWS KMS to create a new customer master key (CMK) in the US West (Northern California) region. Create an encrypted, cross-region read replica by specifying the key identifier of the recently created CMK in the US West Region.

Explanation:-Amazon RDS encrypted DB instances provide an additional layer of data protection by securing your data from unauthorized access to the underlying storage. You can use Amazon RDS encryption to increase data protection of your applications deployed in the cloud, and to fulfill compliance requirements for data-at-rest encryption.
Amazon RDS also supports encrypting an Oracle or SQL Server DB instance with Transparent Data Encryption (TDE). TDE can be used with encryption at rest, although using TDE and encryption at rest simultaneously might slightly affect the performance of your database. You must manage different keys for each encryption method.
To manage the keys used for encrypting and decrypting your Amazon RDS resources, you use the AWS Key Management Service (AWS KMS). AWS KMS combines secure, highly available hardware and software to provide a key management system scaled for the cloud. Using AWS KMS, you can create encryption keys and define the policies that control how these keys can be used. AWS KMS supports CloudTrail, so you can audit key usage to verify that keys are being used appropriately. You can use your AWS KMS keys with Amazon RDS and supported AWS services such as Amazon S3, Amazon EBS, and Amazon Redshift.
For an Amazon RDS encrypted DB instance, all logs, backups, and snapshots are encrypted. A read replica of an Amazon RDS encrypted instance is also encrypted using the same key as the master instance when both are in the same AWS Region. If the master and read replica are in different AWS Regions, you encrypt using the encryption key for that AWS Region.
Keys generated by AWS KMS are only stored and used in the region in which they were created. They cannot be transferred to another region. For example; keys created in the EU-Central (Frankfurt) region are only stored and used within the EU-Central (Frankfurt) region.
References:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.Encryption.html
https://docs.aws.amazon.com/kms/latest/developerguide/services-rds.html
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html

- Create a snapshot of the RDS database. Copy the encrypted snapshot to the US West (Northern California) Region. Launch a new read replica using the restored snapshot.

Explanation:-This option is incorrect because creating a cross-region read replica doesn't require creating and restoring a snapshot. You have to create a CMK in the target region first and specify that CMK when creating a read replica.

Q44) A startup is developing a distributed cache for session management using Amazon ElastiCache for Redis with cluster mode enabled. The cache layer will be used by various applications running in AWS. The cluster shall use the default port for Redis connection.
Which combination of steps should the Database Specialist do to protect the ElastiCache cluster from unauthorized access? (Select TWO.)

✓ Set the associated security group to allow inbound traffic on TCP port 6379 from trusted clients only.

Explanation:-To help keep your data secure, Amazon ElastiCache and Amazon EC2 provide mechanisms to guard against unauthorized access of your data on the server.
Amazon ElastiCache for Redis also provides optional encryption features for data on clusters running Redis versions 3.2.6, 4.0.10 or later:
- In-transit encryption encrypts your data whenever it is moving from one place to another, such as between nodes in your cluster or between your cluster and your application.
- At-rest encryption encrypts your on-disk data during sync and backup operations.
If you want to enable in-transit or at-rest encryption, you must meet the following conditions:
- Your cluster or replication group must be running Redis 3.2.6, 4.0.10 or later.
- Your cluster or replication group must be created in a VPC based on Amazon VPC.
- Optionally, you can also use AUTH and the AUTH token (password) needed to perform operations on this cluster or replication group.
Amazon ElastiCache in-transit encryption is an optional feature that allows you to increase the security of your data at its most vulnerable points—when it is in transit from one location to another. Because there is some processing needed to encrypt and decrypt the data at the endpoints, enabling in-transit encryption can have some performance impact. You should benchmark your data with and without in-transit encryption to determine the performance impact for your use cases.
At-rest encryption can be enabled on a replication group only when it is created. Because there is some processing needed to encrypt and decrypt the data, enabling at-rest encryption can have a performance impact during these operations. You should benchmark your data with and without at-rest encryption to determine the performance impact for your use cases.
Redis authentication tokens enable Redis to require a token (password) before allowing clients to run commands, thereby improving data security.

When you use Redis AUTH with your ElastiCache for Redis cluster, you should follow a strict token policy to enhance the security of your ElastiCache cluster. To connect to your cluster, you have to include the parameter --auth-token (API: AuthToken) with the correct token when you create your replication group or cluster. You must also include the token in all subsequent commands to the replication group or cluster.
The default TCP port for Redis is 6379. This port must be allowed in the associated security group of the cluster. For Memcached, the default port is 11211.
References:
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/in-transit-encryption.html
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/at-rest-encryption.html
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/auth.html

● Enable Transparent Data Encryption (TDE) in the ElastiCache cluster.

Explanation:-This option is incorrect because TDE is not an available encryption feature in ElastiCache. This is only applicable to the Microsoft SQL Server.

✔ Enable encryption in-transit and encryption at-rest on the ElastiCache cluster including Redis AUTH. Configure the clients to use the auth-token parameter when connecting to the Redis cluster.

Explanation:-To help keep your data secure, Amazon ElastiCache and Amazon EC2 provide mechanisms to guard against unauthorized access of your data on the server.
Amazon ElastiCache for Redis also provides optional encryption features for data on clusters running Redis versions 3.2.6, 4.0.10 or later:
- In-transit encryption encrypts your data whenever it is moving from one place to another, such as between nodes in your cluster or between your cluster and your application.
- At-rest encryption encrypts your on-disk data during sync and backup operations.
If you want to enable in-transit or at-rest encryption, you must meet the following conditions:
- Your cluster or replication group must be running Redis 3.2.6, 4.0.10 or later.
- Your cluster or replication group must be created in a VPC based on Amazon VPC.
- Optionally, you can also use AUTH and the AUTH token (password) needed to perform operations on this cluster or replication group.
Amazon ElastiCache in-transit encryption is an optional feature that allows you to increase the security of your data at its most vulnerable points—when it is in transit from one location to another. Because there is some processing needed to encrypt and decrypt the data at the endpoints, enabling in-transit encryption can have some performance impact. You should benchmark your data with and without in-transit encryption to determine the performance impact for your use cases.
At-rest encryption can be enabled on a replication group only when it is created. Because there is some processing needed to encrypt and decrypt the data, enabling at-rest encryption can have a performance impact during these operations. You should benchmark your data with and without at-rest encryption to determine the performance impact for your use cases.
Redis authentication tokens enable Redis to require a token (password) before allowing clients to run commands, thereby improving data security.
When you use Redis AUTH with your ElastiCache for Redis cluster, you should follow a strict token policy to enhance the security of your ElastiCache cluster. To connect to your cluster, you have to include the parameter --auth-token (API: AuthToken) with the correct token when you create your replication group or cluster. You must also include the token in all subsequent commands to the replication group or cluster.
The default TCP port for Redis is 6379. This port must be allowed in the associated security group of the cluster. For Memcached, the default port is 11211.
References:
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/in-transit-encryption.html
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/at-rest-encryption.html
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/auth.html

● Ensure that the associated security group allows inbound traffic on TCP port 11211 from trusted clients only.

Explanation:-This option is incorrect because this the default port for Memcached, and not for Redis.

● Integrate AWS Web Application Firewall (WAF) with the ElastiCache Redis cluster.

Explanation:-This option is incorrect because you can't directly integrate AWS WAF with ElastiCache. It is more appropriate to integrate AWS WAF to your CloudFront web distribution, Application Load Balancers, or Amazon API Gateway for your APIs.

Q45) A Database Specialist needs to secure an Amazon RDS for MySQL database that will be used by a web application. Instead of an alphanumeric password, the database must only be accessed via a short-lived authentication token for greater security. Which of the following will satisfy this requirement?

● Use AWS SSO to access the RDS database.

Explanation:-This option is incorrect because AWS SSO just enables you to centrally manage SSO access and user permissions for all of your AWS accounts managed through AWS Organizations.

✔ Configure the RDS database to use IAM DB Authentication. Generate the access token using the aws rds generate-db-auth-token AWS CLI command.

Explanation:-You can authenticate to your DB instance using AWS Identity and Access Management (IAM) database authentication. IAM database authentication works with MySQL and PostgreSQL. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token.
An authentication token is a unique string of characters that Amazon RDS generates on request. Authentication tokens are generated using AWS Signature Version 4. Each token has a lifetime of 15 minutes. You don't need to store user credentials in the database, because authentication is managed externally using IAM. You can also still use standard database authentication.
With IAM database authentication, you don't need to assign database passwords to the user accounts you create. If you remove an IAM user that is mapped to a database account, you should also remove the database account with the DROP USER statement.
With MySQL, authentication is handled by AWSAuthenticationPlugin—an AWS-provided plugin that works seamlessly with IAM to authenticate your IAM users. You can connect to the DB instance and issue the CREATE USER statement, as shown below:
CREATE USER jon_dojo IDENTIFIED WITH AWSAuthenticationPlugin AS 'RDS';
The IDENTIFIED WITH clause allows MySQL to use the AWSAuthenticationPlugin to authenticate the database account (jon_dojo). The AS 'RDS' clause refers to the authentication method, and the specified database account should have the same name as the IAM user or role.
IAM database authentication provides the following benefits:
Network traffic to and from the database is encrypted using Secure Sockets Layer (SSL).
You can use IAM to centrally manage access to your database resources, instead of managing access individually on each DB instance.
For applications running on Amazon EC2, you can use profile credentials specific to your EC2 instance to access your database instead of a password, for greater security
To generate the IAM authentication token, you can choose from the following procedures:
Via AWS CLI using the aws rds generate-db-auth-token command.
Via AWS SDK for Python (Boto3) using the generate-db-auth-token method.
Via AWS SDK for Java using RdsIamAuthTokenGenerator class and getAuthToken method.
References:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.html
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.DBAccounts.html

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.Connecting.AWSCLI.html#UsingWithRDS.IAMDBAuth.Connecting.AWSCLI.AuthToken

Enable IAM DB Authentication in the RDS database. Generate the access token from AWS Security Token Service (STS) via the GetSessionToken API. Use the AWSAuthenticationPlugin to authenticate the database account.

Explanation:-This option is incorrect. Although enabling IAM DB Authentication is right, this option is still wrong because IAM DB Authentication doesn't use AWS STS nor the GetSessionToken API to generate the access token.

Set up Multi-Factor Authentication (MFA) for Amazon RDS. Use the generated MFA token to access and connect to the database.

Explanation:-This option is incorrect because MFA simply adds extra security to your resources and IAM users. You can't use the MFA token to connect to your database. You have to set up IAM DB Authentication instead.

Q46) A financial services company that is running a web application on an Amazon RDS for MariaDB DB instance in a Multi-AZ deployment wants to achieve regulatory compliance. Currently, the RDS DB instance is associated with a DB subnet group with private subnets and without an Internet gateway. It is configured with security groups to allow traffic to and from the application servers only. The master key used to encrypt the database at rest is managed using AWS KMS.
Which step will provide new and additional security?

Create a new rule in the security group that denies port 22 traffic to the RDS DB instance.

Explanation:-This option is incorrect. Port 22 is used for SSH, and Amazon RDS, by design, inherently does not allow anyone to SSH to the DB instance. No additional security was enforced.

✅ Enable SSL in the RDS DB instance and encrypt data in transit.

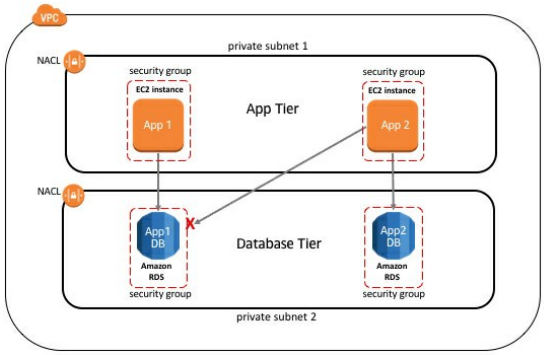
Explanation:-

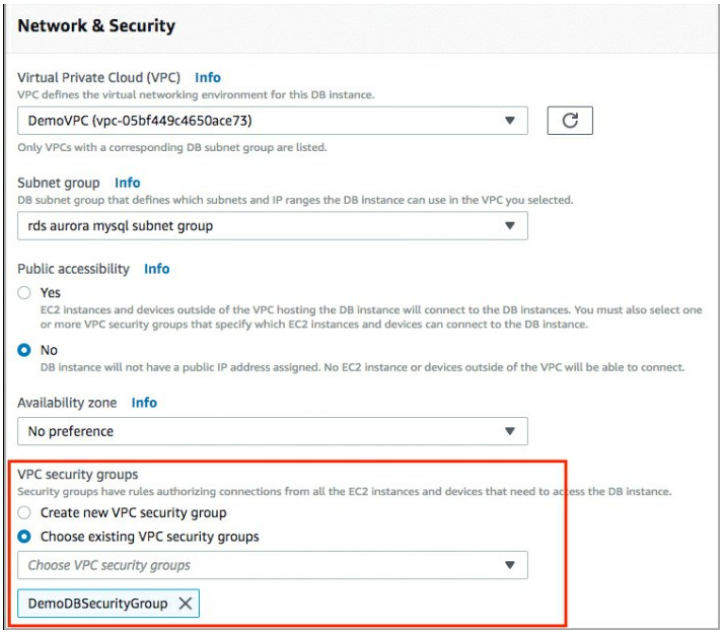
Managing a database requires enforcing security. AWS offers numerous features and tools that secure them.

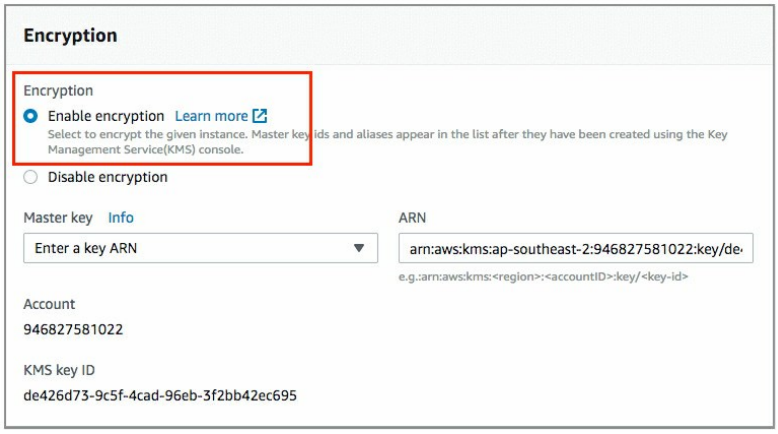
A critical security consideration for your sensitive data is the network isolation of the database. Network isolation makes your database accessible only on a private IP address range to only those components that require access to it. The fundamental design component that enables this security isolation is Amazon VPC. You associate a VPC with your database instance at the time that you create the database instance.

Also, the key to database security is encryption and tokenization. Enabling encryption at rest is essential. It ensures that the volumes underpinning the database and snapshots (if encrypted) can be read outside of the AWS account only with AWS KMS encryption key permissions explicitly granted.

Amazon RDS automatically creates a Secure Sockets Layer (SSL) certificate for each RDS database to enable client connections over SSL. Use RDS SSL certificates specific to your AWS Region or Regions to ensure that your applications connect to the RDS instance in the correct AWS Region.







Set up NACLs that allow the entire EC2 subnet to access the DB instance.

Explanation:-This option is incorrect because this action is not part of the best practice.

Turn off the Public Accessibility of the RDS DB instance.

Explanation:-This option is incorrect. The Public Accessibility option of RDS DB instance is inherently disabled when the scenario stated that the instance is associated with a DB subnet group with private subnets. No additional security was enforced.

Q47) A Database Specialist manages an unencrypted Amazon RDS for PostgreSQL DB instance in a Multi-AZ deployment. Due to the increase in user base, the Security team defined a new requirement that privileged IAM users should use their IAM credentials to access the DB instance and avoid other authentication methods that require database passwords. The Database Specialist noticed that the PostgreSQL DB instance is using a default parameter group with ssl set to 1.
What is the FASTEST method to meet this requirement?

Create a new RDS PostgreSQL DB instance from a snapshot of the existing DB instance using the same parameter group and set the authentication mode to Password and IAM Database Authentication. Assign appropriate IAM policies and database privileges for IAM users.

Explanation:-This option is incorrect. Although this works, it is not the fastest solution.

✅ Modify the existing RDS PostgreSQL DB instance and choose Enable IAM Database Authentication. Assign appropriate IAM policies and database privileges for IAM users.

Explanation:-You can authenticate to your DB instance using AWS Identity and Access Management (IAM) database authentication. IAM database authentication works with MySQL and PostgreSQL. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token.

By default, IAM database authentication is disabled on DB instances. You can enable IAM database authentication (or disable it again) using the AWS Management Console, AWS CLI, or the API. IAM authentication for PostgreSQL DB instances requires that the SSL value be 1. To allow an IAM user or role to connect to your DB instance, you must create an IAM policy. After that, you attach the policy to an IAM user or role. To use IAM authentication with PostgreSQL, connect to the DB instance, create database users, and then grant them the rds_iam role.

Kerberos provides a different authentication method than AWS Identity and Access Management (IAM). A database can use either Kerberos or IAM authentication, but not both.

Migrate a snapshot of the existing DB instance to an Amazon Aurora PostgreSQL DB cluster and set the authentication mode to Password and IAM Database Authentication. Assign appropriate IAM policies and database privileges for IAM users.

Explanation:-This option is incorrect. Although this works, it is also not the fastest solution. More importantly, RDS supports IAM Database Authentication, so it is unnecessary to migrate to Amazon Aurora.

Modify the existing RDS PostgreSQL DB instance and choose Enable IAM Database Authentication. Enable Kerberos Authentication by setting an Active Directory. Assign appropriate IAM policies and database privileges for the IAM users.

Explanation:-This option is incorrect. Kerberos authentication is not necessary to meet the requirements. It is also not possible for the database to use both IAM DB authentication and Kerberos authentication.

Q48) A large eCommerce company is developing an enterprise application that uses Amazon Aurora as the database. The application must be deployed to the production, staging, and development environments. A Database Specialist has been instructed to specify different MasterUsername and MasterUserPassword properties for each environment in the AWS CloudFormation templates as part of the automated deployment. All the AWS CloudFormation templates are stored and version-controlled in the company's AWS CodeCommit repository. The database master password in the production environment must be regularly rotated to comply with the internal IT security policies.
What is the MOST suitable and secure solution that the Database Specialist must implement?

Store the master passwords for all environments in the same AWS CodeCommit repository where the CloudFormation template is stored. Configure the template to retrieve the master password stored from the repository.

Explanation:-This option is incorrect because storing sensitive database credentials in your source code repository is a security risk since the passwords can easily be viewed in plaintext. A better solution is to use dynamic references in your CloudFormation template and storing the passwords in AWS Secrets Manager.

Store the master passwords for all environments in AWS Systems Manager Parameter Store. Configure the CloudFormation template to use the ssm dynamic reference to retrieve the master password stored in AWS Systems Manager Parameter Store per environment. Enable automatic rotation.

Explanation:-This option is incorrect. Although this solution could work, the AWS Systems Manager Parameter Store doesn't have the capability of rotating the passwords automatically. You have to use AWS Secrets Manager instead.

✅ Store the master passwords for all environments in AWS Secrets Manager. Configure the CloudFormation template to use the secretsmanager dynamic reference to retrieve the master password stored in AWS Secrets Manager per environment. Enable automatic rotation.

Explanation:-Dynamic references provide a compact, powerful way for you to specify external values that are stored and managed in other services, such as the Systems Manager Parameter Store, in your stack templates. When you use a dynamic reference, CloudFormation retrieves the value of the specified reference when necessary during stack and change set operations.

Rather than embedding sensitive information directly in your AWS CloudFormation templates, it is recommended that you use dynamic parameters in the stack template to reference sensitive information that is stored and managed outside of CloudFormation, such as in the AWS Systems Manager Parameter Store or AWS Secrets Manager.

CloudFormation currently supports the following dynamic reference patterns:

ssm - for plaintext values stored in AWS Systems Manager Parameter Store

ssm-secure - for secure strings stored in AWS Systems Manager Parameter Store

secretsmanager - for entire secrets or specific secret values that are stored in AWS Secrets Manager

Some considerations when using dynamic references:

You can include up to 60 dynamic references in a stack template.

For transforms, such as AWS::Include and AWS::Serverless, AWS CloudFormation does not resolve dynamic references prior to invoking any transforms. Rather, AWS CloudFormation passes the literal string of the dynamic reference to the transform. Dynamic references (including those inserted into the processed template as the result of a transform) are resolved when you execute the change set using the template.

Dynamic references for secure values, such as ssm-secure and secretsmanager, are not currently supported in custom resources.

You can use the secretsmanager dynamic reference to retrieve entire secrets or secret values that are stored in AWS Secrets Manager for use in your templates. Secrets can be database credentials, passwords, third-party API keys, and even arbitrary text. Using Secrets Manager, you can store and control access to these secrets centrally. Secrets Manager enables you to replace hardcoded credentials in your code (including passwords), with an API call to Secrets Manager to retrieve the secret programmatically.

The advantage of using Secrets Manager over Systems Manager Parameter Store is its ability to automatically rotate the secrets for a service or database.

Store the master passwords for all environments as CMKs in AWS Key Management Service. Configure the CloudFormation template to use the secretsmanager dynamic reference to retrieve the master password stored in AWS KMS key store per environment. Enable automatic rotation.

Explanation:-This option is incorrect because AWS KMS is a managed service that just enables you to easily create and control the keys used for cryptographic operations. AWS KMS is not a suitable service to store your database passwords.

Q49) The audit team required users to connect to a running Amazon Elasticache for Redis with cluster mode using AUTH before allowing them to execute commands. The engine version is 5.0.5. A Database Specialist tries to attempt to configure the new setting using the AWS console but fails.
Which among the options most likely caused this issue?

- The security group for the Elasticache cluster allows all inbound traffic from itself only.

Explanation:-This option is incorrect. Enabling AUTH does not require the security group to prescribe to a specific inbound traffic configuration.

- At-rest encryption is disabled for the Elasticache cluster.

Explanation:-This option is incorrect. Enabling AUTH does not require enabling at-rest encryption in the Elasticache cluster.

- ✅ In-transit encryption is disabled for the Elasticache cluster.

Explanation:-AUTH can only be enabled for ElastiCache for Redis clusters if in-transit encryption was enabled during creation. Furthermore, you can only enable in-transit encryption when you create an ElastiCache for Redis replication group using the AWS Management Console, the AWS CLI, or the ElastiCache API.
Reference:
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/auth.html

- Enabling AUTH can only be done using the AWS Command Line Interface.

Explanation:-This option is incorrect because using AUTH in the Elasticache cluster can be enabled using the console and the AWS Command Line Interface.

Q50) A government agency is storing confidential user data in an Amazon Aurora PostgreSQL DB cluster. The Database Specialist needs to ensure that all communications to the database are encrypted, and any non-SSL connection requests are blocked.
What must be done to satisfy this requirement properly?

- Add a new sslmode parameter in the DB parameter group of the Aurora cluster with a value of verify-full.

Explanation:-This option is incorrect because the sslmode is actually a connection parameter that you specify in your database client in order to connect to RDS via SSL. You have to set the rds.force_ssl parameter to 1 first and use the sslmode parameter in your connection string.

- Modify the rds.force_ssl parameter in the DB parameter group of the Aurora cluster to 0.

Explanation:-This option is incorrect because setting this parameter to 0 means that you are disabling the SSL support.

- ✅ Set the rds.force_ssl parameter in the DB parameter group of the Aurora cluster to 1.

Explanation:-Amazon RDS supports DB instances running several versions of PostgreSQL. You can create DB instances and DB snapshots, point-in-time restores, and backups. DB instances running PostgreSQL support Multi-AZ deployments, read replicas, Provisioned IOPS, and can be created inside a VPC. You can also use Secure Socket Layer (SSL) to connect to a DB instance running PostgreSQL.
Amazon RDS supports Secure Socket Layer (SSL) encryption for PostgreSQL DB instances. Using SSL, you can encrypt a PostgreSQL connection between your applications and your PostgreSQL DB instances. You can also force all connections to your PostgreSQL DB instance to use SSL.
SSL support is available in all AWS regions for PostgreSQL. Amazon RDS creates an SSL certificate for your PostgreSQL DB instance when the instance is created. If you enable SSL certificate verification, then the SSL certificate includes the DB instance endpoint as the Common Name (CN) for the SSL certificate to guard against spoofing attacks.
When you connect using SSL, your client can choose whether to verify the certificate chain. If your connection parameters specify sslmode=verify-ca or sslmode=verify-full, then your client requires the RDS CA certificates to be in their trust store or referenced in the connection URL. This requirement is to verify the certificate chain that signs your database certificate.
You can require that connections to your PostgreSQL DB instance use SSL by using the rds.force_ssl parameter. By default, the rds.force_ssl parameter is set to 0 (off). You can set the rds.force_ssl parameter to 1 (on) to require SSL for connections to your DB instance. Updating the rds.force_ssl parameter also sets the PostgreSQL ssl parameter to 1 (on) and modifies your DB instance's pg_hba.conf file to support the new SSL configuration.
You can set the rds.force_ssl parameter value by updating the parameter group for your DB instance. If the parameter group for your DB instance isn't the default one, and the ssl parameter is already set to 1 when you set rds.force_ssl to 1, you don't need to reboot your DB instance. Otherwise, you must reboot your DB instance for the change to take effect.
References:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_PostgreSQL.html
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithParamGroups.html
https://www.postgresql.org/docs/11/libpq-connect.html#LIBPQ-CONNECT-SSLMODE

- The Secure Socket Layer (SSL) encryption in an Amazon Aurora PostgreSQL DB cluster is enabled by default. No need to modify the rds.force_ssl parameter in the DB parameter group of the Aurora cluster.

Explanation:-This option is incorrect because, by default, the value of rds.force_ssl parameter is 0. You have to manually enable the SSL support by setting its value to 1.
