

Q1)

**A developer is implementing an AWS Lambda function that will read data from an Amazon DynamoDB table.**

**What is the most secure method of providing the Lambda function access permissions to the DynamoDB table?**

- ☐ Create IAM policy granting necessary access permissions to the DynamoDB table. Assign the IAM policy to the Lambda Function policy.
- ☐ Create an IAM user and access key. Create IAM policy granting necessary access permissions to the DynamoDB table. Assign the IAM policy to the IAM user. Provide the access key ID and the secret access key to the developer.
- ☐ Create a DynamoDB username and password. Provide the username and password to the developer.
- ☒ Create an IAM role. Create IAM policy granting necessary access permissions to the DynamoDB table. Assign the IAM policy to the IAM role. Assign the IAM role to the Lambda Execution Role.

**Explanation:-**security best practice for access control to DynamoDB tables is to use IAM policies and IAM roles. The role then needs to be assigned to the Lambda Execution Role. An AWS Lambda function's execution role grants the function permission to access other AWS services and resources.

- ☐ Create an IAM role. Create IAM policy granting necessary access permissions from AWS Lambda. Assign the IAM policy to the IAM role. Assign the IAM role to the DynamoDB table.

Q2)

**An IoT application has a requirement to collect data from a large number of devices. The data needs to be ingested quickly with very low latency.**

**What would be the recommended option for the database in this scenario?**

- ☒ Amazon DynamoDB

**Explanation:-**Amazon DynamoDB is a key-value and document database suitable for unstructured data and optimized for low-latency (sub-millisecond) performance.

- ☐ Amazon RedShift
- ☐ Amazon RDS for PostgreSQL with PostGIS
- ☐ Amazon RDS for Microsoft SQL

Q3)

**A company is developing a business-critical application. Their RTO and RPO requirements call for a database with data storage in three AWS regions deployed in an active-active configuration. The application requires data synchronization across regions.**

**What is the optimal solution for these requirements?**

- ☐ Amazon Athena with Amazon S3 cross-region replication
- ☒ Amazon DynamoDB with global tables

**Explanation:-**Amazon DynamoDB global tables provide a fully managed solution for deploying a multiregion, multi-master database. When a DynamoDB global table is created, it consists of multiple replica tables (one per Region) that DynamoDB treats as a single unit. When an application writes data to a replica table in one Region, DynamoDB propagates the write to the other replica tables in the other AWS Regions automatically.

- ☐ Amazon Aurora Global Database
- ☐ Amazon RDS with Multi-AZ

Q4)

**A retail organization is developing a data lake solution utilizing Amazon S3 to store large amount of data. The solution must be accessible via SQL queries. The organization wants to minimize infrastructure costs.**

**What AWS service should be part of their solution?**

- ☒ Amazon Athena

**Explanation:-**Amazon Athena can be used to query data in S3 directly using SQL query syntax. It is also a serverless service requiring no infrastructure.

- ☐ Amazon DynamoDB
- ☐ Amazon Redshift Spectrum
- ☐ Amazon Aurora

Q5)

**A retail organization is developing a data lake solution utilizing Amazon S3 to store a large amount of data. They would like to be able to perform data exploration and discovery activities by running SQL queries on the data. Based on the output of those activities, they would like to produce complex reports accessible to a large number of users via BI applications.**

**What AWS services should be part of their solution (SELECT TWO)?**

- ☐ Amazon QuickSight for the data discovery activities
- ☒ Amazon Athena for the data discovery activities

**Explanation:-**Amazon Athena can be used to perform ad-hoc queries on data in S3 directly using SQL syntax. Option E is incorrect because Amazon QuickSight is a business analytics service used to build visualizations and business insights reports. It is not used for data exploration activities.

- ☐ AWS Glue for the data discovery activities
- ☐ Amazon Lambda for the complex reporting
- ☒ Amazon RedShift Spectrum for the complex reporting

**Explanation:-**Amazon Redshift Spectrum can be used to query data from files in Amazon S3 without having to load the data into Amazon Redshift tables. Redshift Spectrum compute-intensive queries employ massive parallelism to execute very fast against large datasets.

---

**Q6)**

**An application uses Amazon RDS for MySQL deployed in Multi-AZ configuration as the backend database. Users have started raising complaints regarding the performance of the application. It has been identified that the performance issues are caused by increased read activity on the database.**

**What is the most performance optimized resolution to this problem?**

- ☒ Deploy an Amazon ElastiCache cluster in front of the RDS DB instance

**Explanation:-**Amazon ElastiCache provides sub-millisecond response for read queries.

- ☐ Deploy an RDS Read Replica in a different AZ to the master DB instance
- ☐ Configure the application to read from the Multi-AZ standby instance
- ☐ Deploy an RDS Read Replica in the same AZ as the master DB instance

---

**Q7)**

**In order to enforce compliance and auditing requirements, the TTL feature is enabled on a DynamoDB table.**

**What approach can be used to ensure unauthorized updates to the TTL attribute are prevented?**

- ☐ When configuring DynamoDB table TTL, specify authorized users ARNs.
- ☐ TTL is a DynamoDB compliance and audit feature and cannot be altered once enabled.
- ☐ Create an inline resource-based policy that allows dynamodb:ConfigureTimeToLive and denies other update actions. Attach the policy to the DynamoDB table.

- ☒ Use IAM policies to deny update actions to the TTL attribute or feature configuration. Create an IAM role policy that allows dynamodb:UpdateTimeToLive. Assign the role policy to the authorized users.

**Explanation:-**role based IAM policies can be used to deny specific update actions to DynamoDB tables. Allowing dynamodb:UpdateTimeToLive is required to grant modification of TTL on a DynamoDB table. A role policy allowing this action should be assigned to authorized users.

---

**Q8)**

**A DynamoDB table has Time to Live (TTL) enabled. A solution for processing items deleted by TTL needs to be implemented.**

**What is the optimal solution to this requirement?**

- ☐ Use CloudTrail to collect TTL delete events. Create a Lambda function to process the event. Create an SNS Topic and trigger a Lambda function to process the event.
- ☐ Use CloudWatch Events to collect TTL delete events. Create a Lambda function to process the event. Configure the Lambda function as the target for CloudWatch Events.
- ☐ Enable continuous backups with point-in-time recovery for the table.
- ☒ Enable DynamoDB Streams on the table.

**Explanation:-**DynamoDB Streams can be used to collect and retain items deleted by TTL.

---

**Q9)**

**An organization uses an IAM group DatabaseAdmin to manage administrative access to their database services.**

**What is the best way to prevent accidental deletion of a DynamoDB table?**

- ☒ Create an IAM policy with explicit deny on dynamodb:DeleteTable action. Attach the policy to DatabaseAdmin group.

**Explanation:-**using role-based access control with an explicit deny effect on the IAM policy attached to the DatabaseAdmin group is the optimal solution to the requirement.

- ☐ Enable deletion protection on the DynamoDB table.
- ☐ Create a resource based inline policy with explicit deny on the dynamodb:DeleteTable action. Attach the policy to the DynamoDB table.
- ☐ Create an inline policy with explicit deny on the dynamodbDeleteTable action. Attach the policy to the users of the DatabaseAdmin group.

---

**Q10)**

**An organization uses an IAM group DatabaseAdmin to manage administrative access to their database services.**

**What is the best way to prevent accidental deletion of a Amazon RDS Database?**

- ☐ Create an IAM policy with explicit deny on rds:DeleteDBInstance action. Attach the policy to DatabaseAdmin group.
- ☐ Create an inline policy with explicit deny on the rds:DeleteDBInstance action. Attach the policy to the users of the DatabaseAdmin group.
- ☐ Create a resource based inline policy with explicit deny on the rds:DeleteDBInstance action. Attach the policy to the RDS database.
- ☒ Enable deletion protection on the RDS database.

**Explanation:-**AWS RDS natively supports delete protection. It can be turned on for databases that have this requirement.

---

**Q11)**

**Testing team created Aurora Read Replica for an RDS MySQL database to evaluate the performance behavior of an Aurora instance under production loads. After completing the evaluation, the team is unable to delete this read replica instance.**

**What is the cause and solution to this problem?**

- ☒ It is impossible to delete the last instance of a read replica DB cluster. It must be promoted to a standalone DB cluster.

**Explanation:-**the read replica cluster must be promoted to a standalone database cluster before it can be deleted.

- It is impossible to delete an active read replica instance. Turn off the read replica instance in order to delete it.
- Read Replicas must be deleted using CLI with `--enable-delete` flag.
- Deletion protection is enabled on the RDS MySQL master database. Disable deletion protection.

---

#### Q12)

**When deleting an RDS instance using AWS CLI, following error is encountered: An error occurred (InvalidParameterCombination) when calling the DeleteDBInstance operation: FinalDBSnapshotIdentifier cannot be specified when deleting a cluster instance**

**How can you resolve it?**

✔ A cluster-level snapshot wasn't taken before deleting the database. Use the `--skip-final-snapshot` flag in the CLI delete command to skip the final snapshot.

**Explanation:-**the error warns that a cluster-level snapshot wasn't taken before the delete command. One way to resolve the issue is to use the `--skip-final-snapshot` flag.

- Deletion protection is enabled. Use `aws rds modify-db-instance` command with `--no-deletion-protection` flag.
- In the CLI delete command, Specify the S3 bucket ARN where the final database snapshot must be stored.
- There is not enough allocated storage for the final database snapshot. Use the `aws rds modify-db-instance` command to allocate additional storage for the snapshot.

---

#### Q13)

**An application uses DynamoDB to store data. Compliance audit requirements state that all data must be retained for 90 days.**

**What is the optimal solution for these requirements?**

- Create a Lambda function to initiate an on-demand snapshot of the DynamoDB table. Trigger the Lambda function using CloudWatch Events.
- ✔ Enable DynamoDB Streams on the DynamoDB table. Create a Firehose stream to load the data into an S3 bucket. Create a Lambda function to poll the DynamoDB stream and deliver batch records from streams to Firehose.

**Explanation:-**S3 offers the most cost-effective long-term storage solution. Using native integration capabilities such as DynamoDB streams and Kinesis Firehose streams is the simplest and most cost-effective solution to this requirement.

- Enable and configure point-in-time recovery (PITR) on the DynamoDB table.
- Enable and configure DynamoDB backups.

---

#### Q14)

**An organization has a hybrid cloud architecture consisting of their on-premise datacenter and AWS cloud connected using AWS Direct Connect. The organization is looking to migrate from an on-premise MySQL Database to Amazon Aurora MySQL Database cluster.**

**What is the optimal solution for performing data migration in this scenario?**

**The organization is particularly concerned about performing the migration as fast as possible.**

- Use the AWS Database Migration Service (DMS) to migrate the data.
- Use the MySQL `mysqldump` utility to copy the data.
- Create an Aurora Read Replica of the source database. After the migration is complete, promote the Aurora Read Replica to a stand-alone DB cluster.
- ✔ Use Percona XtraBackup utility to create backup files from the source database to an Amazon S3 bucket. Restore the Aurora MySQL DB cluster from those files.

**Explanation:-**Percona XtraBackup utility can be used to create backup files from the source database to an Amazon S3 bucket. XtraBackup utility can copy source database backup files directly to S3. These files can then be directly imported into an Aurora MySQL cluster using the "Restore from S3" option. As `mysqldump` utility uses DDL and DML statements to recreate the database schemas and load the data, it is a very time consuming process. Therefore, using XtraBackup utility backups is a preferred method when time is a high priority consideration.

---

#### Q15)

**An organization is looking to migrate from a very large on-premise MySQL Database to Amazon RDS MySQL Database cluster. The organization is particularly concerned about minimizing the impact on application availability. Since the migration data set is very large, the organization would also like to minimize any data transfer costs.**

**What is the optimal solution for performing data migration in this scenario?**

- Use Percona XtraBackup utility to create backup files from the source database to an Amazon S3 bucket. Restore the Aurora MySQL DB cluster from those files
- Create a read replica of the source database. After the migration is complete, promote the read replica to a stand-alone database.
- Use the AWS Database Migration Service (DMS) to migrate the data.
- ✔ Use the MySQL `mysqldump` utility to copy the data.

**Explanation:-**`mysqldump` utility can be used to create a backup of the existing on-premise database. The backup can then be transferred in compressed form to AWS cloud. Finally, `mysqldump` utility can be used to restore the data into an RDS instance.

---

#### Q16)

**An IoT application collects data from a large number of devices and uses DynamoDB to store the data. Due to variable and unpredictable workloads, short bursts of traffic spikes occur resulting in throttling of requests on the DynamoDB table.**

**What is the optimal solution to this problem?**

- ✔ Switch the DynamoDB table to on-demand capacity mode.

**Explanation:-**Amazon on-demand capacity mode is the best solution for applications whose database workload is complex to forecast, with large

spikes of short duration, or average table utilization well below the peak.

- Increase provisioned capacity on the DynamoDB table
- Configure DynamoDB Auto-Scaling
- Configure CloudWatch alarm on DynamoDB utilization. Use the CloudWatch alarm to trigger a Lambda function to scale-up the DynamoDB capacity.

---

#### Q17)

**A company is developing a business-critical application. Their RTO and RPO requirements call for a database with data storage in three AWS regions. The company wants to minimize the latency of all database operations.**

**What is the optimal solution for these requirements?**

- Amazon Aurora Multi-Master Cluster
- ✓ Amazon DynamoDB with global tables

**Explanation:-**DynamoDB global tables automatically replicates data across two or more AWS Regions, with full support for multi-master writes.

- Amazon Aurora Global Database
- Amazon RDS with Multi-AZ

---

#### Q18)

**A company is developing a business-critical application. Their RTO and RPO requirements call for a relational database in active-active configuration, with zero down-time for all database operations.**

**What is the optimal database choice for these requirements?**

- Amazon DynamoDB with global tables
- ✓ Amazon Aurora Multi-Master Cluster

**Explanation:-**Amazon Aurora Multi-Master cluster offers write capability to multiple instances. In applications where zero downtime is required for database write operations, a multi-master cluster can be used to avoid an outage when a writer instance becomes unavailable.

- Amazon Aurora Global Database
- Amazon RDS with Multi-AZ

---

#### Q19)

**In order to scale and off-load read traffic of an Amazon RDS database, multiple Amazon RDS Read Replicas have been configured.**

**What is the optimal solution to distribute the read traffic to the Amazon read replicas?**

- Create a Network Load Balancer. Register each read replica endpoint with the load balancer target group.
- Create an Application Load Balancer. Register each read replica endpoint with the load balancer target group.
- Amazon RDS has a cluster reader endpoint `xxx.cluster-ro-xxxxxx.us-east1.rds.amazonaws.com` that provides load-balancing for read traffic.
- ✓ Add each read replica endpoint to a Route 53 record set. Use Weighted routing policy to distribute the traffic.

**Explanation:-**it is possible to create a Route 53 hosted zone and record set to store the DNS endpoints of the RDS read replica instances. Each endpoint should have the same value for the weighted routing configuration to ensure equal distribution of traffic among the RDS read replicas.

---

#### Q20)

**Company compliance requirements specify that database events such as connections, disconnections, tables queried, or types of queries issued (DML, DDL, or DCL) on an Aurora MySQL DB cluster must be retained for audit purposes.**

**What is the optimal solution to accomplish this?**

- Update CloudTrail audit log trail for the RDS database to send the audit data to CloudWatch log group.
  - ✓ Create a custom DB cluster parameter group. Enable and configure Advanced Auditing parameters. Associate the custom parameter group with the Aurora RDS cluster. Modify the log export configuration of the RDS cluster to publish logs to CloudWatch.
- Explanation:-**Amazon Aurora RDS provides native capability to write audit logs to CloudWatch log groups. This functionality can be enabled and configured using a custom parameter group.
- Use `exec rdsadmin.manage_tracefiles.set_tracefile_table_location` command to configure the `tracefile_table` view to point to the audit log trace file. Create a Lambda function to query the view and write the audit data to CloudWatch log group. Trigger the Lambda function using CloudWatch events.
  - Create a Lambda function to download the database audit log file using `DownloadDBLogFilePortion` API and write the audit data to CloudWatch log group. Trigger the Lambda function using CloudWatch events.

---

#### Q21)

**A data analytics team wishes to perform daily data export from DynamoDB table and an import into an S3 bucket in parquet file format for further data analytics processing.**

**What is the operationally optimal solution?**

- Enable DynamoDB Streams. Create Lambda function to poll DynamoDB stream and push items to Kinesis Data Firehose. Use Data Firehose to perform data conversion and store the data in S3.
- Enable DynamoDB Streams. Create a Lambda function to poll the DynamoDB stream, perform data conversion, and deliver batch records to S3
- Use AWS Data Pipeline service to manage the Amazon EMR jobs for data export, conversion, and data import.
- ✓ Use AWS Step functions to manage the data export, conversion, and data import workflow. Use AWS Glue job to perform the data export, conversion, and data import.

**Explanation:-**relatively simple AWS Glue job can be created to perform ETL operations. Glue has native capability to connect to DynamoDB and S3 for read/write operations. It also contains native support for the parquet data format. Further, this solution is completely serverless, thus requiring

**Q22)**

**A data analytics team wishes to develop a solution for real-time data export from DynamoDB table and an import into an S3 bucket in parquet file format for further data analytics processing.**

**Which solution implements these requirements?**

✔ Enable DynamoDB Streams. Create Lambda function to poll DynamoDB stream and push items to Kinesis Data Firehose. Use Data Firehose to perform data conversion and store the data in S3.

**Explanation:**-Lambda function can be used to read data from stream and write data to Kinesis Data Firehose. Kinesis data firehose can perform data conversion to Parquet data format and support native data write to S3.

- Use AWS Data Pipeline service to manage the Amazon EMR jobs for data export, conversion, and data import.
- Use AWS Step functions to manage the data export, conversion, and data import workflow. Use AWS Glue job to perform the data export, conversion, and data import
- Enable DynamoDB Streams. Create a Lambda function to poll the DynamoDB stream, perform data conversion, and deliver batch records to S3.

**Q23)**

**A sales application stores inventory of available items for purchase. Each item is uniquely identified by its Part Number, and contains additional attributes: Part Type, Manufacture Date, Manufacturer Name, Country of Origin, and Cost. The application requires to retrieve item information during each sales transaction. Further, it produces reports for list of sales for each country.**

**What would be the optimal DynamoDB data model for this application?**

- Table Partition Key=Part Number; Table Sort Key=Part Type; GSI Partition Key= Manufacture Date; GSI Sort Key=Country of Origin;
  - Table Partition Key=Part Number; Table Sort Key=Part Type; GSI Partition Key=Country of Origin; GSI Sort Key=Manufacturer Name;
  - ✔ Table Partition Key=Part Number; Table Sort Key=Part Type; GSI Partition Key=Random Prefix; GSI Sort Key=Country of Origin;
- Explanation:**-Part Number is unique for each item thus making it a good choice for the table partition key. Using a random prefix for the GSI partition key and Country of Origin as the sort key enables to have high cardinality for the partition key (thus avoiding any hot partitions) while still allowing for fast querying based on Country of Origin.
- Table Partition Key=Manufacture Date; Table Sort Key=Part Number; GSI Partition Key= Manufacturer Name; GSI Sort Key=Country of Origin;

**Q24)**

**A DBA wishes to limit the number of simultaneous connections that a user account can make to 10 on a specific MySQL RDS Database.**

**What method should the DBA use to accomplish this task?**

- Modify the default parameter group of the MySQL RDS Database by setting the max\_user\_connections parameter to 10.
  - Modify the default option group of the MySQL RDS Database by setting the max\_user\_connections parameter to 10.
  - ✔ Create a new DB parameter group. Modify the max\_user\_connections parameter to 10. Update the RDS MySQL Database to use the new DB parameter group.
- Explanation:**-DB parameter groups need to be used to modify RDS DB parameters. To set a particular parameter, you must create a new DB parameter group, and update the specific parameter (max\_user\_connections in this case). Then you must modify the RDS database instance to use the new DB Parameter group.
- Execute command SET max\_user\_connections=10;.

**Q25)**

**An application uses AWS RDS SQL Server instance as the database backend. A DBA wishes to take a SQL Server native backup of a database.**

**What action must he perform to enable this requirement?**

- Create a new DB Parameter Group and configure the SQLSERVER\_BACKUP\_RESTORE system parameter. Associate the DB Parameter Group with the DB instance.
  - Use the AWS CLI create-db-snapshot command to create a backup of the RDS instance.
  - ✔ Create a new Option Group and configure SQLSERVER\_BACKUP\_RESTORE option. Associate the option group with the DB instance.
- Explanation:**-performing SQL Server native backup using .bak files is a SQL Server specific functionality. To enable this capability, SQLSERVER\_BACK\_RESTORE option must be configured. To add this functionality to an RDS SQL Server instance, you must create a new custom Option Group and configure the SQLSERVER\_BACKUP\_RESTORE setting, and then associate the new Option Group with the RDS instance.
- Ensure that RDS automated backup feature is configured on the RDS instance.