

AWS – Database & Data Storage Services (Basic to Intermediate)

1. What are the different types of database services provided by AWS?

Answer:

AWS offers both relational and non-relational database services:

- **Amazon RDS** – Relational DB (MySQL, PostgreSQL, Oracle, etc.)
- **Amazon Aurora** – High-performance MySQL/PostgreSQL-compatible
- **Amazon DynamoDB** – NoSQL, key-value/document store
- **Amazon Redshift** – Data warehousing (analytical DB)
- **Amazon ElastiCache** – In-memory cache (Redis, Memcached)
- **Amazon Neptune** – Graph database
- **AWS Timestream** – Time-series database
- **Amazon DocumentDB** – MongoDB-compatible document database

2. What is Amazon RDS?

Answer:

Amazon **RDS (Relational Database Service)** is a managed service that makes it easy to set up, operate, and scale relational databases in the cloud. It supports MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server.

3. What is the difference between Amazon RDS and Amazon Aurora?

Answer:

- **Amazon RDS** supports multiple database engines.
- **Amazon Aurora** is a high-performance, MySQL/PostgreSQL-compatible DB designed by AWS with up to **5x** throughput over standard MySQL.

4. What is DynamoDB and when should you use it?

Answer:

Amazon **DynamoDB** is a **NoSQL database** that offers key-value and document-based storage. It is fully managed, serverless, and suitable for high-performance, scalable applications like real-time gaming, IoT, and mobile apps.

5. What is the difference between SQL and NoSQL in AWS context?

Answer:

- **SQL (Relational):** Structured, fixed schema (e.g., RDS, Aurora)
- **NoSQL:** Schema-less, flexible, scalable (e.g., DynamoDB, DocumentDB)

6. How does AWS ensure high availability of its database services?

Answer:

- **Multi-AZ deployments** (automatic failover)
- **Read replicas** for performance and fault-tolerance
- **Backups and snapshots**
- **Automatic failover** for Aurora and RDS

7. What are read replicas in Amazon RDS?

Answer:

Read replicas provide read-only copies of your database. They:

- Improve performance by offloading read traffic
- Can be promoted to standalone DBs
- Support asynchronous replication

8. What is Amazon ElastiCache and why is it used?

Answer:

Amazon ElastiCache is a fully managed **in-memory data store** compatible with Redis and Memcached. It is used for:

- Caching
- Real-time analytics
- Session management
- Reducing latency

9. How is data secured in AWS databases?

Answer:

- **Encryption at rest** using AWS KMS

- **Encryption in transit** using SSL/TLS
- **IAM policies** and **VPC security groups**
- **Database authentication** using IAM and secrets manager

10. What is Amazon Redshift?

Answer:

Amazon Redshift is a **cloud data warehouse** service used for online analytical processing (OLAP). It is optimized for:

- Big data analytics
- Complex queries
- Integrates with AWS Glue, S3, Athena, QuickSight

11. What is Amazon Timestream?

Answer:

Timestream is a **fully managed time-series database** for IoT and operational applications. It is optimized for fast ingestion and querying of time-based data.

12. Can you explain backup and restore options in Amazon RDS?

Answer:

- **Automated backups:** Retained for a specified retention period (up to 35 days)
- **Manual snapshots:** Persist until deleted
- **Point-in-time recovery:** Restore DB to any second within retention window

13. What is Amazon DocumentDB?

Answer:

Amazon DocumentDB is a **fully managed document database** service designed to be compatible with **MongoDB**, optimized for JSON document storage and retrieval.

14. How does AWS Glue relate to databases?

Answer:

AWS Glue is a **serverless ETL** service that helps prepare and move data between databases, S3, Redshift, etc. It automatically discovers schema and builds data catalogs.

15. What is the purpose of Amazon Database Migration Service (DMS)?

Answer:

AWS DMS helps **migrate on-premises or cloud databases** to AWS quickly and securely. It supports:

- Homogeneous (e.g., MySQL to MySQL)
- Heterogeneous (e.g., Oracle to PostgreSQL) migrations