

Computing (Database Services)

DATABASE SERVICES · SQL VS NOSQL · RDS · AURORA · DYNAMODB · DMS & SCT

STUDY & REVIEW

01 MUST KNOW

if you remember five things, remember these

- 01 Purpose-built, fully managed:** AWS offers a family of managed databases — **RDS** and **Aurora** (relational/SQL, transactional), **Redshift** (data warehouse/analytics), **DynamoDB** (NoSQL), **Neptune** (graph), and **ElastiCache** (in-memory cache). Match the workload — data shape, size, access patterns, performance — to the engine. Managed means AWS handles provisioning, patching, backups, and scaling.
- 02 SQL vs NoSQL:** **SQL** (relational) stores **rows and columns** with a **fixed** schema, is queried with **SQL**, and scales **vertically** (bigger hardware). **NoSQL** stores **key-value / documents / graphs** with a **dynamic** schema, queries **collections of documents**, and scales **horizontally** (more servers). *NoSQL* today means *not only SQL* — it can still support SQL-style queries.
- 03 RDS availability vs scaling:** managed relational service with **six engines** (Aurora, MySQL, MariaDB, Oracle, SQL Server, PostgreSQL). **Multi-AZ** = a **synchronous** standby in a second AZ that **auto-fails-over** with **no data loss** (availability). **Read replicas** = **asynchronous** read-only copies that scale **read-heavy** workloads. *Instance class* = compute; *storage type* = storage/IOPS.
- 04 Aurora cluster architecture:** MySQL- and PostgreSQL-compatible engine, up to **5x** MySQL / **3x** PostgreSQL throughput. Stores **six copies of data across three AZs** and allows up to **15 read replicas**. A **DB cluster** = one primary (read/write) + replicas (read-only) over a shared **cluster volume**. A **Serverless** option auto-starts, stops, and scales.
- 05 DMS + SCT migration:** **AWS DMS** migrates databases into AWS with minimal/zero downtime — **homogeneous** (Oracle→Oracle) and **heterogeneous** (Oracle→MySQL). **At least one endpoint must be on an AWS service** (on-prem→on-prem is unsupported). Two steps: **SCT converts the schema/code**, then **DMS loads and replicates the data**.

02 KEY TERMS

TERM	DEFINITION
DB instance	The basic building block of Amazon RDS — an isolated database environment that can hold multiple user-created databases and uses the same tools as a standalone database
Instance class	Determines the CPU, memory, and network capacity of an RDS instance; types are Standard, Memory Optimized, and Burstable Performance
Storage type	Determines the type, performance, and size of the RDS EBS volume — General Purpose SSD (gp2), Provisioned IOPS SSD (io1), or magnetic
Multi-AZ deployment	An RDS high-availability configuration with a synchronously replicated standby in a second AZ that automatically becomes primary on failover — no data loss
Read replica	A read-only RDS copy kept current by asynchronous replication; scales read-heavy workloads beyond a single instance
DB snapshot	A storage-volume backup of an entire RDS instance; the first snapshot is full, subsequent snapshots are incremental (only changed data)
Aurora DB cluster	One or more DB instances — one primary plus up to 15 replicas — plus a cluster volume that manages the data for them
Cluster volume	Aurora's virtual storage volume that spans AZs, holding six copies of the data across three Availability Zones (10 GB protection groups)
Partition key	The DynamoDB attribute (the hash attribute) whose value places an item in a partition; part of every primary key

TERM	DEFINITION
Sort key	An optional DynamoDB attribute that stores items sharing a partition key together and orders them, enabling range queries
AWS DMS	AWS Database Migration Service — migrates databases into AWS with minimal downtime; one endpoint must be on an AWS service
AWS SCT	AWS Schema Conversion Tool — automates conversion of database and data-warehouse schema and application SQL between engines

03 SQL VS NOSQL

match the model to the workload

SQL (RELATIONAL) <i>one row per entry; columns are attributes</i> — Data stored in rows and columns — Fixed schema — columns locked before entry — Queried with SQL (complex queries) — Scales vertically (bigger hardware) — RDS · Aurora · Redshift	NOSQL (NON-RELATIONAL) <i>not only SQL — key-value, document, graph</i> — Key-value / document / graph storage — Dynamic schema — rows need not match — Queried over collections of documents — Scales horizontally (more servers) — DynamoDB · Neptune · ElastiCache
---	---

04 AWS DATABASES AT A GLANCE

one purpose-built engine per use case

SERVICE	TYPE	BEST USED FOR
Amazon RDS	Relational	Transactional apps — ERP, CRM, ecommerce; a choice of six engines
Amazon Aurora	Relational	MySQL/PostgreSQL-compatible; high performance, low cost, open source
Amazon Redshift	Relational	Data warehouse — reporting and TB-to-EB-scale analytics
Amazon DynamoDB	NoSQL	Internet-scale apps; document + key-value; structured & unstructured
Amazon Neptune	NoSQL graph	Highly connected data — social feeds, recommendations, fraud detection
Amazon ElastiCache	In-memory	Submillisecond latency — leaderboards, chat, streaming, IoT (Redis/Memcached)

05 MULTI-AZ STANDBY VS READ REPLICA

availability is not the same as read scaling

MULTI-AZ STANDBY <i>availability — serves no reads</i> — Standby copy in a second AZ (same VPC) — Synchronous replication → no data loss — Automatic failover promotes standby — Apps reconnect via the RDS DNS endpoint	READ REPLICA <i>read scaling — not for failover</i> — Read-only copy for read-heavy load — Asynchronous replication — Scales reads beyond one instance — Primary still handles all read/write
--	---

06 TWO-STEP MIGRATION INTO AWS

SCT first, then DMS

01 AWS SCT converts source schema & code to the target engine → **02** Build the target schema on AWS → **03 AWS DMS** loads and replicates the data → **04** Switch applications to the target database

- × “The Multi-AZ standby serves read traffic to lighten the primary.” — No. The standby exists for **availability**; it accepts no reads and only takes over on failover. Offload reads with **read replicas**.
- × “Read replicas give you automatic failover.” — Read replicas use **asynchronous** replication to scale **reads**; automatic failover with no data loss is the job of a **Multi-AZ** deployment.
- × “Scaling RDS storage and changing the instance class both take downtime.” — Increasing **allocated storage** happens with no downtime; **changing the instance class requires downtime**.
- × “After a Multi-AZ failover you must repoint the application.” — Apps connect by the **RDS DNS endpoint**, so failover needs no code change, and synchronous replication means **no data loss**.
- × “Aurora replicas are just an RDS Multi-AZ standby.” — Aurora stores **six copies across three AZs** and allows up to **15 read-only replicas** over a shared cluster volume — its own architecture, distinct from RDS Multi-AZ.
- × “AWS DMS can move an on-premises database straight to another on-premises database.” — Not supported: **at least one endpoint must be on an AWS service**.
- × “AWS DMS converts your schema and stored code.” — DMS creates only the **minimal objects** needed to move data; converting schema, code, and application SQL is the job of **AWS SCT**.
- × “NoSQL can't be queried with SQL and forces every row to share the same columns.” — NoSQL now means **not only SQL** (SQL-style queries are possible), and its **dynamic schema** lets each item carry different attributes.

08 SELF-QUIZ

answers are upside-down below

- Q1** Which AWS database service is purpose-built for graph workloads that navigate highly connected data?
- Q2** How many database engines can Amazon RDS run? Name any three.
- Q3** In an RDS Multi-AZ deployment, what replication keeps the standby current, and how much data is lost on failover?
- Q4** RDS read replicas scale which kind of workload, and what replication type do they use?
- Q5** Which RDS change requires downtime — increasing allocated storage or changing the instance class?
- Q6** How many copies of your data does Amazon Aurora store, and across how many Availability Zones?
- Q7** How many Aurora replicas can a DB cluster have in addition to the primary instance?
- Q8** What two attributes make up a DynamoDB composite (partition-and-sort) primary key?
- Q9** What is the one hard requirement for the endpoints of any AWS DMS migration?
- Q10** In the two-step migration, which tool converts the schema and which loads and replicates the data?

ANSWER KEY (rotate the page)

Q1 Amazon Neptune **Q2** Six — Aurora, MySQL, MariaDB, Oracle, SQL Server, PostgreSQL **Q3** Synchronous replication; no data loss **Q4** Read-heavy workloads; asynchronous replication **Q5** Changing the instance class **Q6** Six copies across three Availability Zones **Q7** Up to 15 **Q8** The partition key and the sort key **Q9** At least one endpoint must be on an AWS service (on-prem to on-prem is unsupported) **Q10** AWS SCT converts the schema; AWS DMS loads and replicates the data