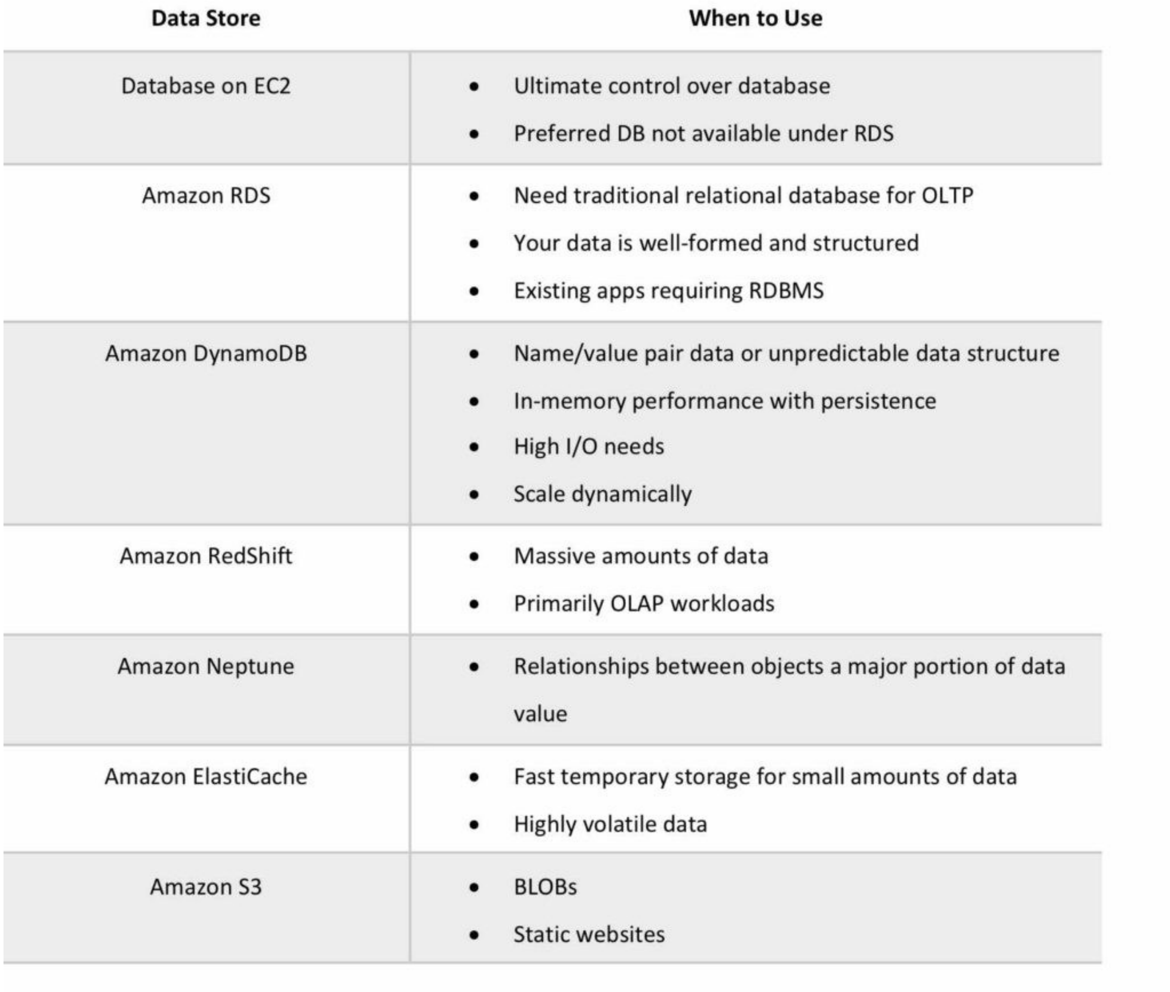
This article covers AWS Databases for the AWS Cloud Practitioner exam. This is one of the key technology areas covered in the exam blueprint.

Use Cases For Different Database Types

The table below provides guidance on the typical use cases for several AWS database/data store services:



Data Store	When to Use
Database on EC2	• Ultimate control over database • Preferred DB not available under RDS
Amazon RDS	• Need traditional relational database for OLTP • Your data is well-formed and structured • Existing apps requiring RDBMS
Amazon DynamoDB	• Name/value pair data or unpredictable data structure • In-memory performance with persistence • High I/O needs • Scale dynamically
Amazon RedShift	• Massive amounts of data • Primarily OLAP workloads
Amazon Neptune	• Relationships between objects a major portion of data value
Amazon ElastiCache	• Fast temporary storage for small amounts of data • Highly volatile data
Amazon S3	• BLOBs • Static websites

We’ll now cover several of these database types that may come up on the exam.

Amazon Relational Database Services (RDS)

Amazon Relational Database Service (Amazon RDS) is a managed service that makes it easy to set up, operate, and scale a relational database in the cloud.

Relational databases are known as Structured Query Language (SQL) databases.

Non-relational databases are known as NoSQL databases.

RDS is an Online Transaction Processing (OLTP) type of database.

RDS features and benefits:

- SQL type of database.
- Can be used to perform complex queries and joins.
- Easy to setup, highly available, fault tolerant, and scalable.
- Used when data is clearly defined.
- Common use cases include online stores and banking systems.

Amazon RDS supports the following database engines:

- SQL Server.
- Oracle.
- MySQL Server.
- PostgreSQL.
- Aurora.
- MariaDB.

Aurora is Amazon's proprietary database.

RDS is a fully managed service and you do not have access to the underlying EC2 instance (no root access).

The RDS service includes the following:

- Security and patching of the DB instances.
- Automated backup for the DB instances.
- Software updates for the DB engine.
- Easy scaling for storage and compute.
- Multi-AZ option with synchronous replication.
- Automatic failover for Multi-AZ option.
- Read replicas option for read heavy workloads.

A DB instance is a database environment in the cloud with the compute and storage resources you specify.

Encryption:

- You can encrypt your Amazon RDS instances and snapshots at rest by enabling the encryption option for your Amazon RDS DB instance.
- Encryption at rest is supported for all DB types and uses AWS KMS.
- You cannot encrypt an existing DB, you need to create a snapshot, copy it, encrypt the copy, then build an encrypted DB from the snapshot.

DB Subnet Groups:

- A DB subnet group is a collection of subnets (typically private) that you create in a VPC and that you then designate for your DB instances.
- Each DB subnet group should have subnets in at least two Availability Zones in a given region.
- It is recommended to configure a subnet group with subnets in each AZ (even for standalone instances).

AWS Charge for:

- DB instance hours (partial hours are charged as full hours).
- Storage GB/month.
- I/O requests/month – for magnetic storage.
- Provisioned IOPS/month – for RDS provisioned IOPS SSD.
- Egress data transfer.
- Backup storage (DB backups and manual snapshots).

Scalability:

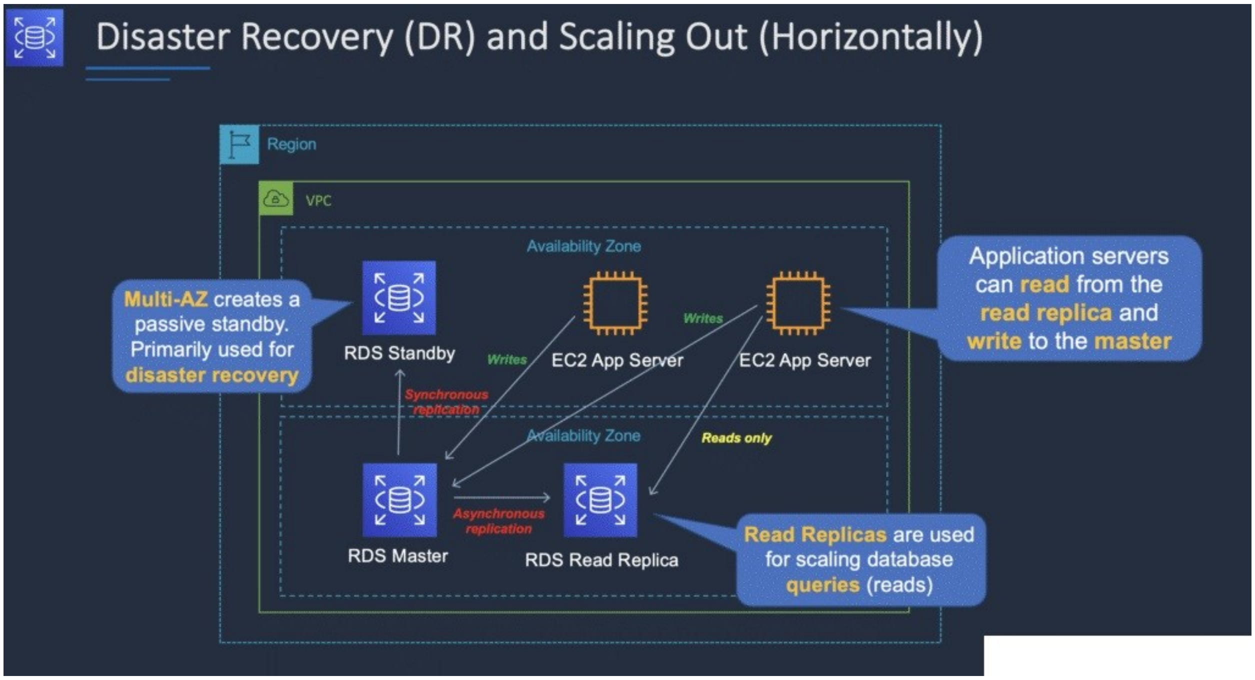
- You can only scale RDS up (compute and storage).
- You cannot decrease the allocated storage for an RDS instance.
- You can scale storage and change the storage type for all DB engines except MS SQL.

RDS provides multi-AZ for disaster recovery which provides fault tolerance across availability zones:

- Multi-AZ RDS creates a replica in another AZ and synchronously replicates to it (DR only).
- There is an option to choose multi-AZ during the launch wizard.
- AWS recommends the use of provisioned IOPS storage for multi-AZ RDS DB instances.
- Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable.
- You cannot choose which AZ in the region will be chosen to create the standby DB instance.

Read Replicas – provide improved performance for reads:

- Read replicas are used for read heavy DBs and replication is asynchronous.
- Read replicas are for workload sharing and offloading.
- Read replicas provide read-only DR.
- Read replicas are created from a snapshot of the master instance.
- Must have automated backups enabled on the primary (retention period > 0).



Disaster Recovery (DR) and Scaling Out (Horizontally)

Multi-AZ creates a passive standby. Primarily used for disaster recovery

Application servers can read from the read replica and write to the master

Read Replicas are used for scaling database queries (reads)

Amazon DynamoDB

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability.

Dynamo DB features and benefits:

- NoSQL type of database (non-relational).
- Fast, highly available, and fully managed.
- Used when data is fluid and can change.
- Common use cases include social networks and web analytics.

Push button scaling means that you can scale the DB at any time without incurring downtime.

SSD based and uses limited indexing on attributes for performance.

DynamoDB is a Web service that uses HTTP over SSL (HTTPS) as a transport and JSON as a message serialisation format.

Amazon DynamoDB stores three geographically distributed replicas of each table to enable high availability and data durability.

Data is synchronously replicated across 3 facilities (AZs) in a region.

Cross-region replication allows you to replicate across regions:

- Amazon DynamoDB global tables provides a fully managed solution for deploying a multi-region, multi-master database.
- When you create a global table, you 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.

Provides low read and write latency.

Scale storage and throughput up or down as needed without code changes or downtime.

DynamoDB is schema-less.

DynamoDB can be used for storing session state.

Provides two read models.

Eventually consistent reads (Default):

- The eventual consistency option maximises your read throughput (best read performance).
- An eventually consistent read might not reflect the results of a recently completed write.
- Consistency across all copies reached within 1 second.

Strongly consistent reads:

- A strongly consistent read returns a result that reflects all writes that received a successful response prior to the read (faster consistency).

Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache for DynamoDB that delivers up to a 10x performance improvement – from milliseconds to microseconds – even at millions of requests per second.

Amazon RedShift

Amazon Redshift is a fast, fully managed data warehouse that makes it simple and cost-effective to analyze all your data using standard SQL and existing Business Intelligence (BI) tools.

RedShift is a SQL based data warehouse used for **analytics** applications.

RedShift is a relational database that is used for Online Analytics Processing (OLAP) use cases.

RedShift is used for running complex analytic queries against petabytes of structured data, using sophisticated query optimization, columnar storage on high-performance local disks, and massively parallel query execution.

RedShift is ideal for **processing** large amounts of data for business intelligence.

RedShift is 10x faster than a traditional SQL DB.

RedShift uses columnar data storage:

- Data is stored sequentially in columns instead of rows.
- Columnar based DB is ideal for data warehousing and analytics.
- Requires fewer I/Os which greatly enhances performance.

RedShift provides advanced compression:

- Data is stored sequentially in columns which allows for much better performance and less storage space.
- RedShift automatically selects the compression scheme.

RedShift uses replication and continuous backups to enhance availability and improve durability and can automatically recover from component and node failures.

RedShift always keeps three copies of your data:

- The original.
- A replica on compute nodes (within the cluster).
- A backup copy on S3.

RedShift provides continuous/incremental backups:

- Multiple copies within a cluster.
- Continuous and incremental backups to S3.
- Continuous and incremental backups across regions.
- Streaming restore.

RedShift provides fault tolerance for the following failures:

- Disk failures.
- Nodes failures.
- Network failures.
- AZ/region level disasters.

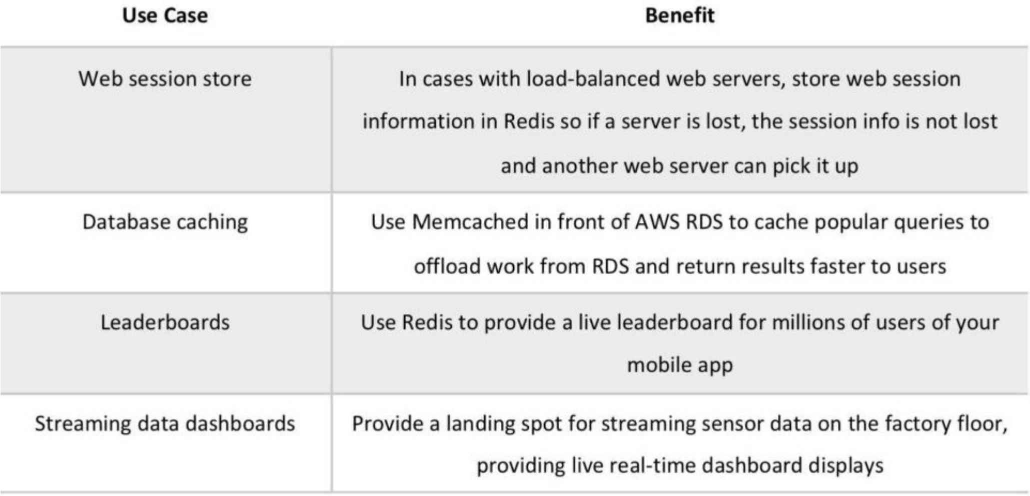
Amazon ElastiCache

ElastiCache is a web service that makes it easy to deploy and run Memcached or Redis protocol-compliant server nodes in the cloud.

The in-memory caching provided by ElastiCache can be used to significantly improve latency and throughput for many read-heavy application workloads or compute-intensive workloads.

Best for scenarios where the DB load is based on Online Analytics Processing (OLAP) transactions.

The following table describes a few typical use cases for ElastiCache:



Use Case	Benefit
Web session store	In cases with load-balanced web servers, store web session information in Redis so if a server is lost, the session info is not lost and another web server can pick it up
Database caching	Use Memcached in front of AWS RDS to cache popular queries to offload work from RDS and return results faster to users
Leaderboards	Use Redis to provide a live leaderboard for millions of users of your mobile app
Streaming data dashboards	Provide a landing spot for streaming sensor data on the factory floor, providing live real-time dashboard displays

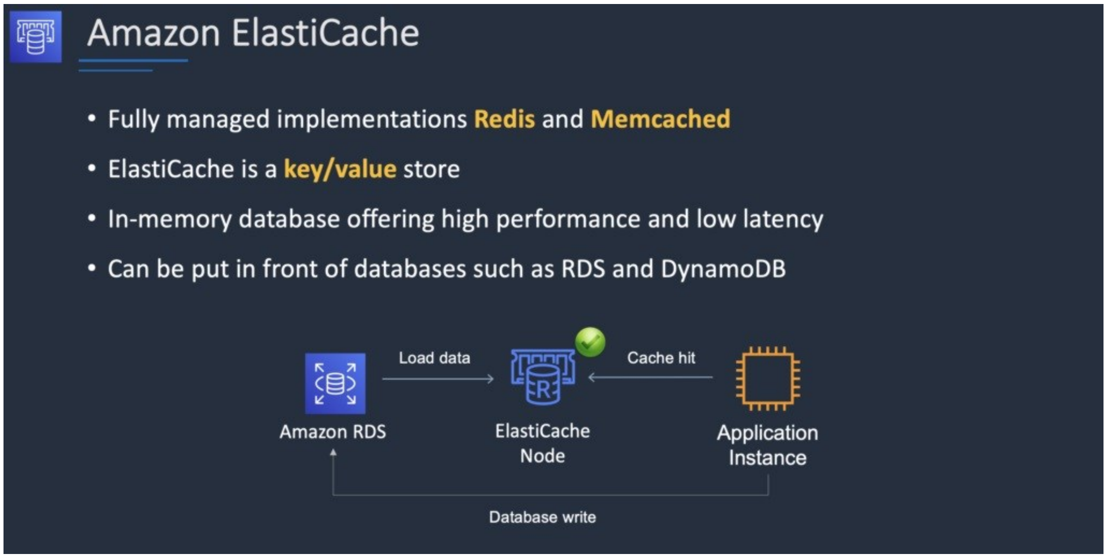
Elasticache EC2 nodes cannot be accessed from the Internet, nor can they be accessed by EC2 instances in other VPCs.

Can be on-demand or reserved instances too (but not Spot instances).

Elasticache can be used for storing session state.

There are two types of ElastiCache engine:

- Memcached – simplest model, can run large nodes with multiple cores/threads, can be scaled in and out, can cache objects such as DBs.
- Redis – complex model, supports encryption, master / slave replication, cross AZ (HA), automatic failover and backup/restore.



Amazon ElastiCache

- Fully managed implementations **Redis** and **Memcached**
- ElastiCache is a **key/value** store
- In-memory database offering high performance and low latency
- Can be put in front of databases such as RDS and DynamoDB

Amazon EMR

Amazon EMR is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data.

EMR utilizes a hosted Hadoop framework running on Amazon EC2 and Amazon S3.

Managed Hadoop framework for processing huge amounts of data.

Also support Apache Spark, HBase, Presto and Flink.

Most commonly used for log analysis, financial analysis, or extract, translate and loading (ETL) activities.