

Module 7: Databases

Tuesday, September 16, 2025

3:52 PM

A. Topics and Module Objectives (MOs)

- Topics:
 - + Amazon Relational Database Service (Amazon RDS)
 - + Amazon DynamoDB
 - + Amazon RedShift
 - + Amazon Aurora
- Module Objectives (MOs):
 - + Explain Amazon Relational Database Service (Amazon RDS)
 - + Identify the functionality in Amazon RDS
 - + Explain Amazon DynamoDB
 - + Identify the functionality in Amazon DB
 - + Explain Amazon Redshift
 - + Explore Amazon Aurora
 - + Platform tasks in an RDS database, such as launching, configuring, and interacting.

B. Amazon Relational Database Service (Amazon EDS)

- Unmanaged services versus Managed services:
 - + Unmanaged: Scaling, fault tolerance, and availability are managed by user
 - + Managed: Scaling, fault-tolerance, and availability are typically built in to the service
- Challenges of relational databases:
 - + Server maintenance and energy footprint
 - + Software installation and patches
 - + Database backups and high availability
 - + Limits on scalability
 - + Operating system (OS) installation and patches
- Amazon RDS: Managed service that sets up and operates a relational database in the cloud
Users ==> Application servers ==> Amazon RDS
- Managed services responsibilities:
 - + User Manage: Application optimization
 - + AWS Manages:
 - * OS installation and patches
 - * Database software installation and patches
 - * Database backups
 - * High availability
 - * Scaling
 - * Power and racking and stacking servers
 - * Server maintenance
- Amazon RDS DB main instance:
 - + DB Instance Class: CPU, Memory, Network Performance
 - + DB Instance Storage: Magnetic, General Purpose (solid state drive, or SSD, Provisioned IOPS
- Amazon RDS DB supports six database engines including MySQL, Amazon Aurora, Microsoft SQL Server, PostgreSQL, MariaDB, Oracle, and DB engines.
- Amazon RDS read replicas:
 - + Features:
 - * Offers asynchronous replication
 - * Can be promoted to master if needed
 - + Functionality:
 - * Use for read-heavy database workloads
 - * Offload read queries
- Use Cases

Web and mobile applications	<ul style="list-style-type: none">• High throughput• Massive storage scalability• High availability
Ecommerce Applications	<ul style="list-style-type: none">• Low-cost database• Data Security

	<ul style="list-style-type: none"> • Fully managed solution
Mobile and online games	<ul style="list-style-type: none"> • Rapidly grow capacity • Automatic Scaling • Database monitoring

- Use the Amazon RDS when application requires:
 - + Complex transactions or complex queries
 - + A medium to high query or write rate - Up to 30,000 IOPS (15,000 reads + 15,000 writes)
 - + No more than a single worker node or shard
 - + High durability
- Do not use when application requires:
 - + Massive read/write rates (for example, 150,000 write/second)
 - + Sharding due to high data size or throughput demands
 - + Simple GET or PUT requests and queries that a NoSQL database can handle
 - + Relational database management system (RDBMS) customization
- Amazon RDS: Clock-hour billing and database characteristics
 - + Clock-hour billing: Resources incur charges when running
 - + Database characteristics: Physical capacity of database:
 - * Engine
 - * Size
 - * Memory class
- Amazon RDS: DB purchase type and multiple DB instances:
 - + DB purchase type:
 - * On-Demand Instances: Compute capacity by the hour
 - * Reserved Instances: Low, one-time, upfront-payment for database instances that are reserved with 1-year or 3-year term.
 - + Number of DB Instances: Provision multiple DB instances to handle peak loads
- Amazon RDS: Storage
 - + Provisioned storage:
 - * No change: Backup storage of up to 100 percent of database storage for an active database
 - * Charge:(GB/month): Backup storage for terminated DB instances
 - + Additional storage:
 - * Charge (GB/month): Backup storage in addition to provisional storage
- Amazon RDS: Deployment type and data transfer
 - + Requests: The number of input and output requests that are made to the database
 - + Deployment type - Storage and I/O charges vary, depending on whether user deploy to:
 - * Single AZ
 - * Multiple AZs
 - + Data Transfer:
 - * No charge for inbound data transfer
 - * Tiered charges for outbound data transfer

C. Amazon Dynamo DB

- Relational versus non-relational databases

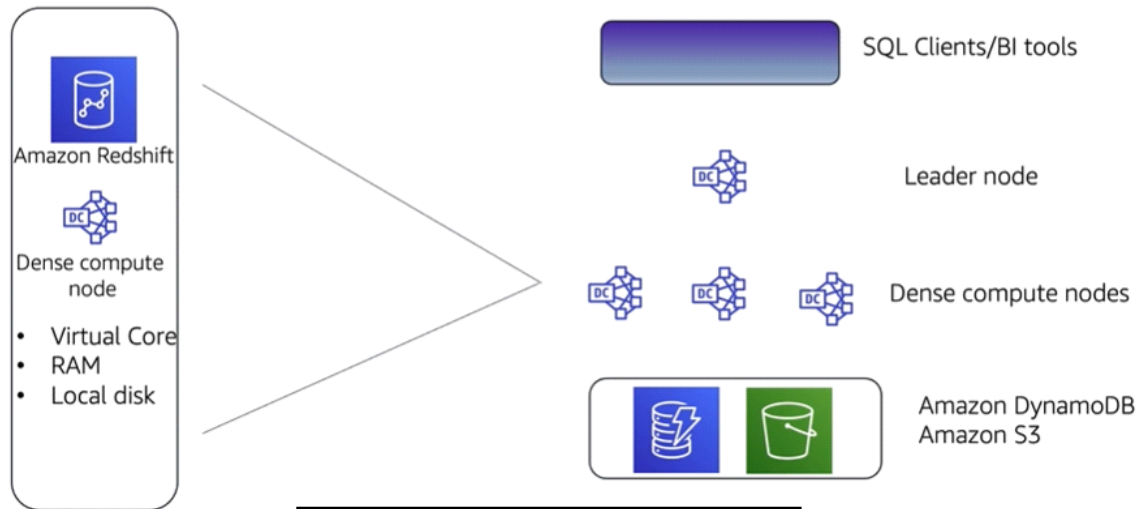
	<i>Relational (SQL)</i>	<i>Non-Relational</i>
<i>Data Storage</i>	Rows and columns	Key-value, document, graph
<i>Schemas</i>	Fixed	Dynamic
<i>Querying</i>	Uses SQL	Focuses on collection of documents
<i>Scalability</i>	Vertical	Horizontal

- Amazon DB: Fast and Flexible NoSQL database service for any scale
 - + NoSQL database tables
 - + Virtually unlimited storage
 - + Items can have differing attributes
 - + Low latency queries
 - + Scalable read/write throughput
- Amazon DynamoDB core components:
 - + Tables, items, and attributes are the core DynamoDB components
 - + DynamoDB supports two different kinds of primary keys: Partition key and partition and sort key

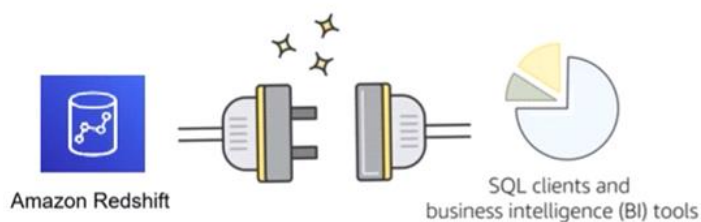
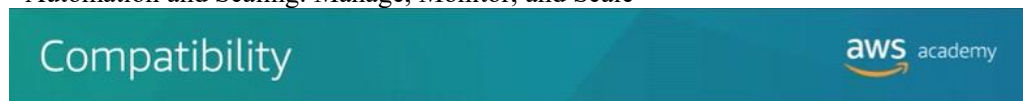
- The difference between the Compound key and Single key is that Compound Key has Sort Key while both of them have Partition key and Attributes

D. Amazon RedShift

- Amazon RedShift is a fast, fully managed data warehouse that makes it simple and cost effective to analyze all user's data by using standard SQL and existing business intelligence tools.



- Automation and Scaling: Manage, Monitor, and Scale



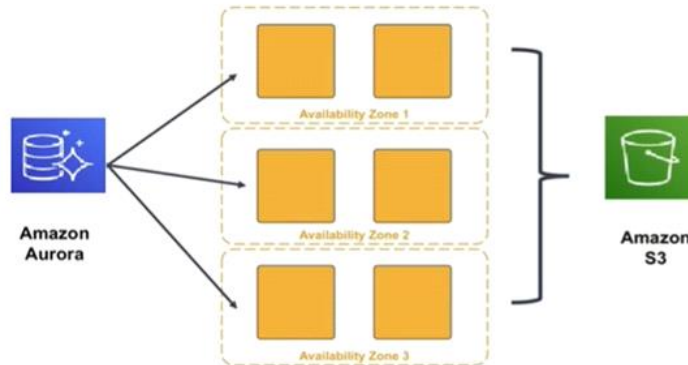
- Amazon Redshift use cases:
 - + Migrate at a pace that customers are comfortable with
 - + Experiment without large upfront cost or commitment
 - + Respond faster to business needs
- Big data:
 - + Low price point for small customers
 - + Managed service for ease of deployment and maintenance
 - + Focus more on data and less on database management
- SaaS:
 - + Scale the data warehouse capacity as demand grows
 - + Add analytic functionality to applications
 - + Reduce hardware and software costs.

E. Amazon Aurora

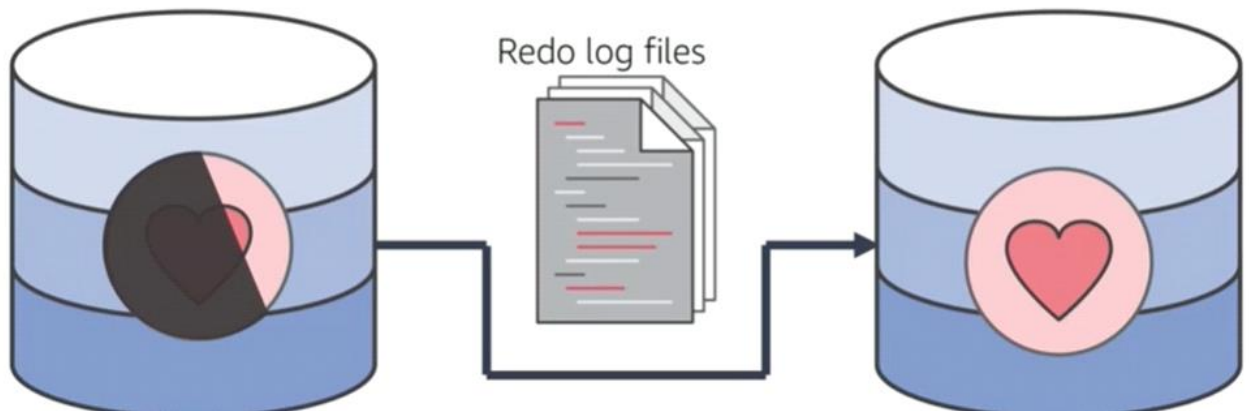
- Enterprise-class relational database
- Compatible with MySQL or PostgreSQL
- Automate time-consuming tasks (such as provisioning, patching, backup, recovery, failure detection, and repair)
- Aurora benefits:
 - + Fast and available

- + Managed service
- + Simple
- + Compatible
- + Pay-as-you-go

High availability



Resilient design



****NOTE: Amazon RDS, Amazon Aurora, and Amazon RedShift are fully managed SQL database service while Amazon DynamoDB is fully managed NoSQL***

1. NoSQL database excel at scaling to hundreds of thousands of requests with key/value access to user profile and session
2. To find an item in a DynamoDB table other than the team's primary key, user would use the scan operation
3. Amazon RedShift is bets suited for analyzing data
4. If user is developing an application that requires a database with extremely fast performance, fast scalability, and flexibility in the database schema, consider the DynamoDB