Cloud Databases

Cloud Computing

Brenden west

Contents

Learning Outcomes

- Overview of Common Datastore types
- Overview of AWS Cloud Database Offerings
- Choosing a cloud DB service
- Working with Amazon RDS
- Working with Amazon DynamoDB

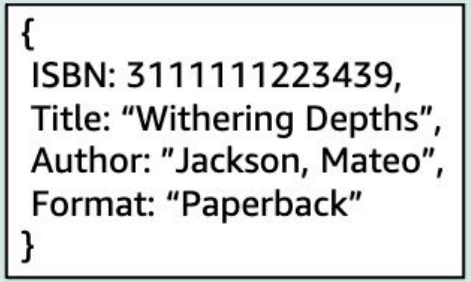
Reading

- AWS Cloud Foundations - Module 8 - Databases
- AWS Cloud Developing - Module 5 - NoSQL Solutions
- https://aws.amazon.com/free/database/

Types of Databases

- **Relational** - Traditional DB suitable for general-purpose workloads.
  - Data stored in rows & columns.. Tables related by keys
  - Useful when strict schema & data quality required, or for transactions.
  - Hard to scale horizontally
  - E.g. - Microsoft SQL Server, Oracle, MySQL, PostgreSQL
- **Document** (aka NoSQL or schema-less) - No predefined schema structure.
  - Stores JSON **documents** in **collections**.
  - Optimal for horizontal scaling.
  - E.g. MongoDB

Relational versus non-relational databases

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

Other Non-relational Databases

- **In-memory** - e.g. Redis. Key-value store optimal for fast, direct access
- **Search-optimized** - e.g. ElasticSearch - Special-purpose document DB where data is indexed for complex text searches.
- **Columnar** - e.g. Redshift, AlloyDB - Column-oriented data representation stores values in a single column together. Optimized for analytical queries.
- **Graph** - data stored as a network of entities and relationships.

AWS Database Offerings

- **Relational** -
  - **Relational Database Service (RDS)** - Configurable service offering different optimizations and ~6 proprietary & open-source database engines
  - **Aurora** - Fully managed DB service compatible with MySQL and PostgreSQL
- **Document** - DynamoDB
- Columnar - Redshift
- **OpenSearch** - Managed search db based on Lucene engine
- **In-memory** - MemoryDB, ElastiCache - redis-compatible services
- **Graph** - Neptune

Amazon RDS

- Amazon RDS is a **managed** service that sets up and operates a relational database in the cloud.
- RDS abstracts administrative DB tasks such as server maintenance, OS & software updates & patches, DB backups
- RDS automatically scales the DB as needed
- Based on a **database instance** - an isolated DB environment that can contain multiple user-created DBs
- Accessible via same tools as a standalone DB instance on EC2
- Pricing based on **clock hours** of running time, as well as db engine, size, & memory class

AWS RDS Instances

- Developer must choose DB engine & instance type (**on-demand** or **reserved**), backup storage, & data transfer
- Usually run on a private subnet & accessible only by authorized services
- Can be configured for **Multi-AZ deployment** with synchronous data replication
- Can be configured with a **read replica** optimized for read-heavy queries

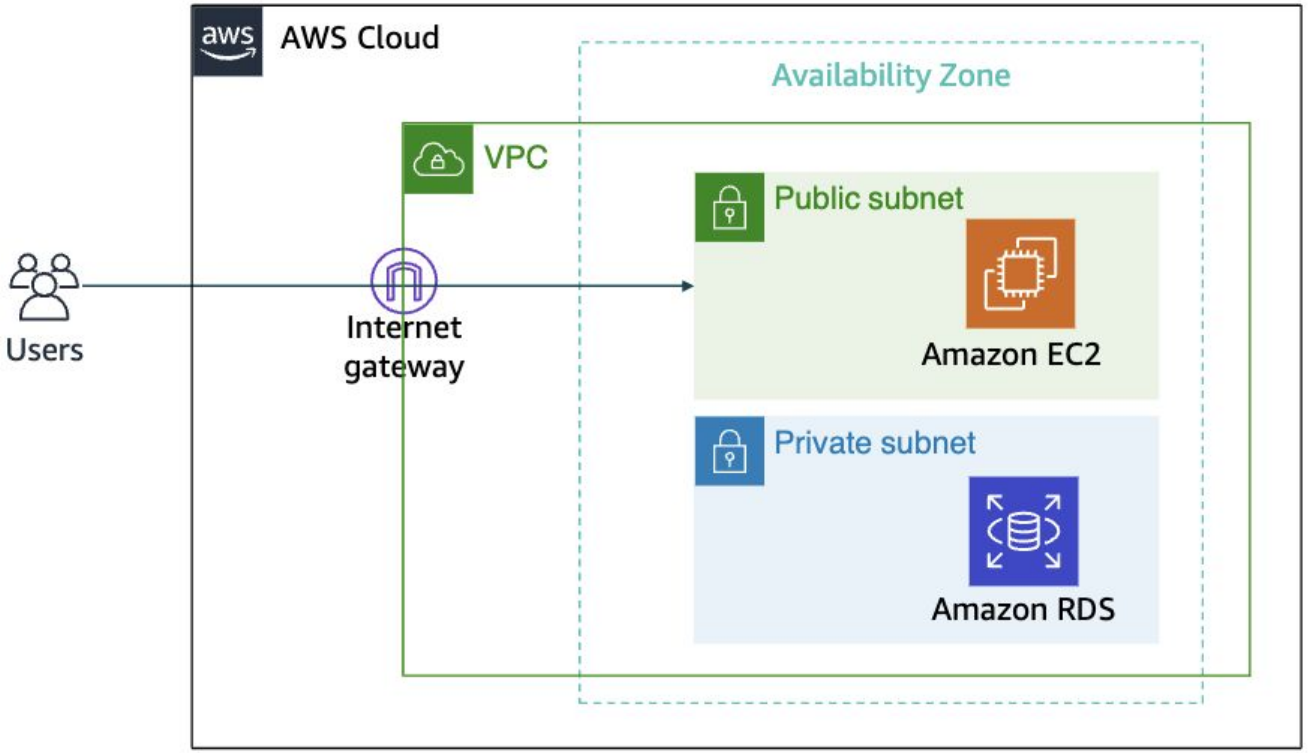
When to Use Amazon RDS

Use Amazon RDS when your 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 Amazon RDS when your 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

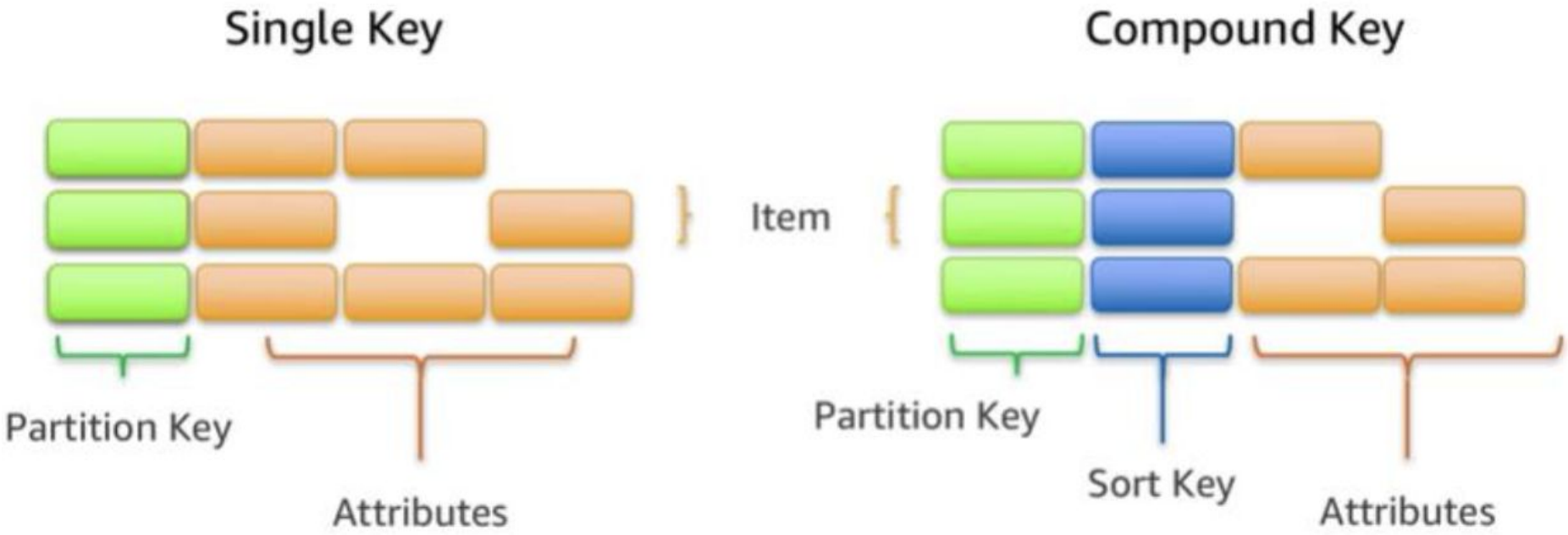
Amazon RDS in a virtual private cloud (VPC)



Amazon DynamoDB

- Fast, flexible non-relational DB service w/ virtually unlimited storage
- Supports **document** & **key-value** datastore models
- Can automatically replicate across regions for redundancy
- Supports encryption at rest and item Time-to-live (TTL)
- Data **items** (records) are stored in schema-less **tables** (collections)
- Supports two types of **primary key** on tables:
  - **Partition key** - single attribute **sort** key
  - **Composite primary key** - Based on two attributes
- Primary key must be a unique identifier
- As data grows, table is automatically partitioned by primary key

Items in a table must have a key



Amazon Redshift

- Fully managed data warehouse
- Supports standard SQL, JDBC & ODBC connectors, and common business intelligence (BI) tools
- Supports complex analytical queries using parallel processing on clustered compute nodes

Amazon Aurora

- Fully managed, pay-as-you-go relational database service
- Compatible with open-source MySQL and PostgreSQL engines
- Stores multiple copies of data across multiple Availability Zones with continuous backups to AWS S3
- Designed for instant crash recovery

Other Cloud Databases

- https://firebase.google.com/products/realtime-database
- https://supabase.com/database
- https://www.mongodb.com/atlas/database