AWS DATABASE SERVICES OVERVIEW

Database

RDS

DynamoDB

ElastiCache

Amazon Redshift

SQL vs NoSQL Databases

	SQL	NoSQL
Data Storage	Rows and Columns	Key-Value
Schemas	Fixed	Dynamic
Querying	Using SQL	Focused on collection of documents
Scalability	Vertical	Horizontal

SQL

ISBN	Title	Author	Format
9182932465265	Cloud Computing Concepts	Wilson, Joe	Paperback
3142536475869	The Database Guru	Gomez, Maria	eBook

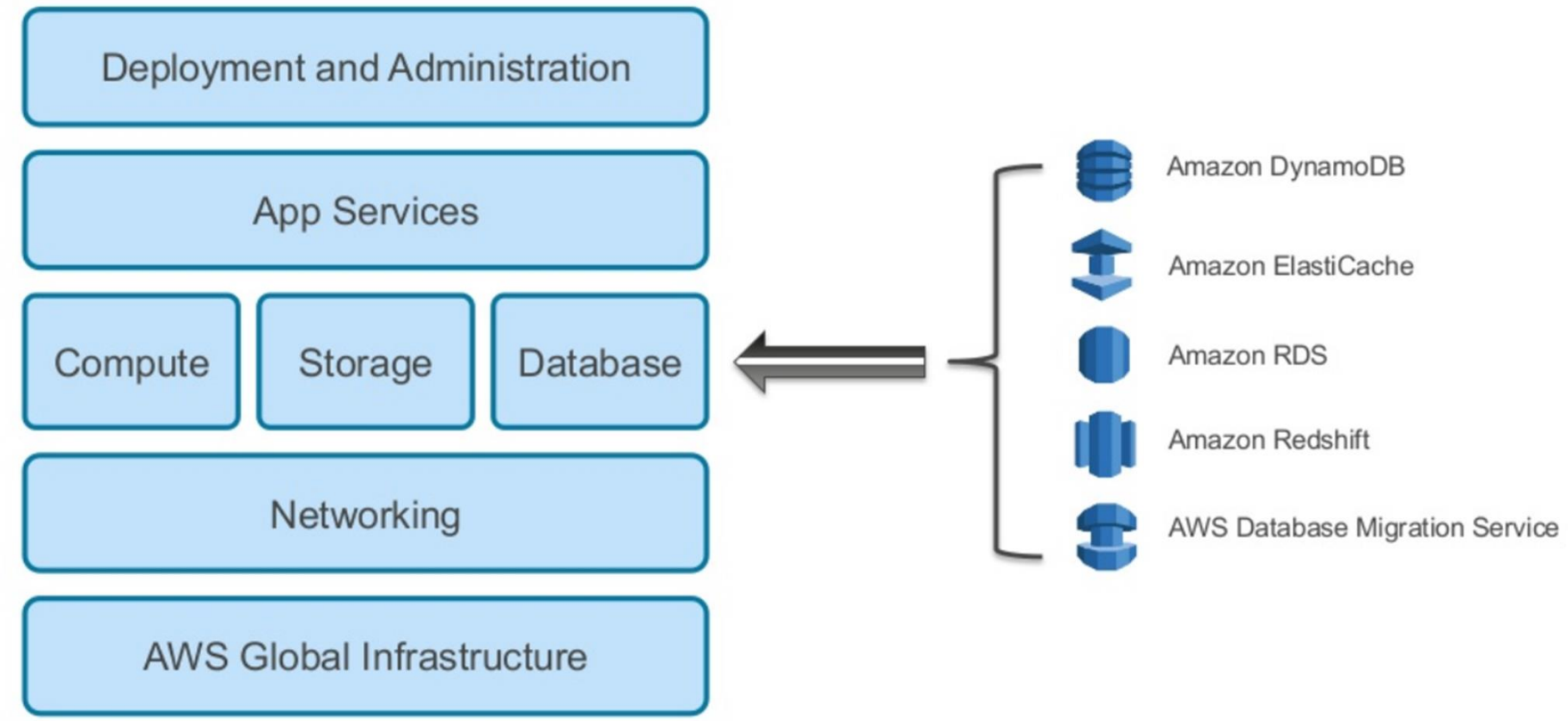
NoSQL

```
{
      ISBN: 9182932465265,
      Title: "Cloud Computing Concepts",
      Author: "Wilson, Joe",
      Format: "Paperback"
}
```

Data Storage Considerations

- No one size fits all.
- Analyze your data requirements by considering:
  - ✓ Data formats
  - ✓ Data size
  - ✓ Query frequency
  - ✓ Data access speed
  - ✓ Data retention period

AWS Managed Database Services



RDS

RDS (Relational Database Service)

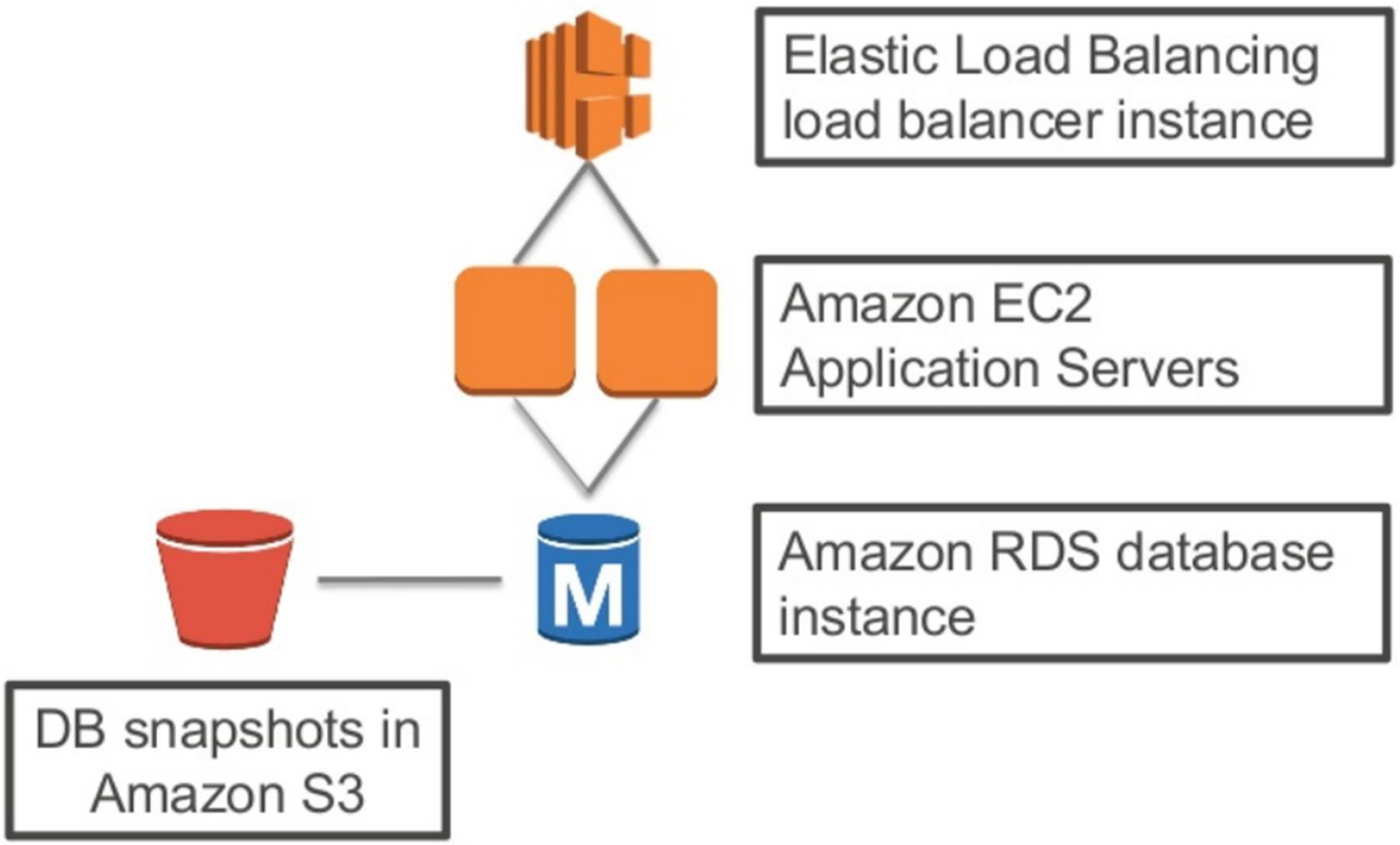
Amazon RDS

- Cost-efficient and **resizable capacity**
- Manages time-consuming **database administration** tasks
- Access to the full capabilities of **Amazon Aurora**, **MySQL**, **MariaDB**, **Microsoft SQL Server**, **Oracle**, and **PostgreSQL** databases

Amazon RDS

- Simple and **fast to deploy**
- Manages common database administrative tasks
- **Compatible** with your applications
- Fast, predictable performance
- Simple and **fast to scale**
- Secure
- Cost-effective

A Simple Application Architecture



How Amazon RDS Backups Work

Automatic Backups:

- Restore your database to a point in time.
- Are enabled by default.
- Let you choose a retention period up to 35 days.

Manual Snapshots:

- Let you build a new database instance from a snapshot.
- Are initiated by the user.
- Persist until the user deletes them.
- Are stored in Amazon S3.

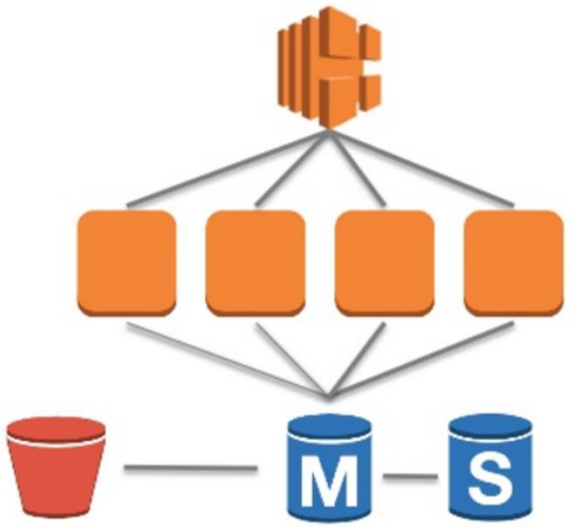
How Amazon RDS Backups Work

- Are a **copy** of a **database** snapshot stored in a **different AWS Region**.
- Provide a backup for disaster **recovery**.
- Can be used as a **base** for **migration** to a different region.

Multi AZ RDS Deployment

- With **Multi-AZ** operation, your database is **synchronously replicated to another Availability Zone** in the same AWS Region.
- **Failover** to the standby **automatically** occurs in case of master database failure.
- Planned maintenance is applied first to standby databases.

A Resilient, Durable Application Architecture



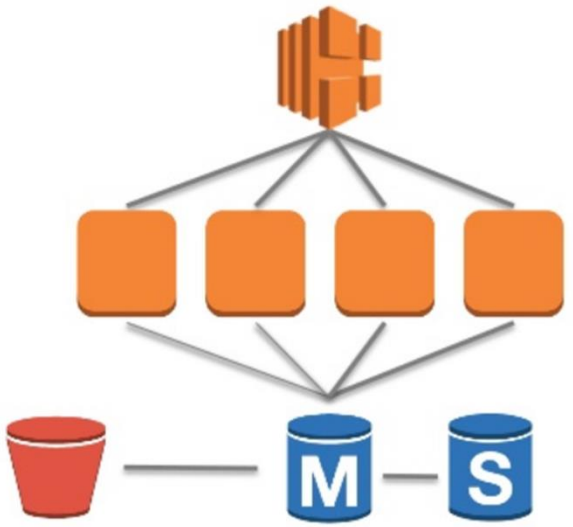
Elastic Load Balancing load balancer instance

Application, in Amazon EC2 instances

Amazon RDS database instances: Master and Multi-AZ standby

DB snapshots in Amazon S3

A Resilient, Durable Application Architecture



Elastic Load Balancing load balancer instance

Application, in Amazon EC2 instances

Amazon RDS database instances: Master and Multi-AZ standby

DB snapshots in Amazon S3

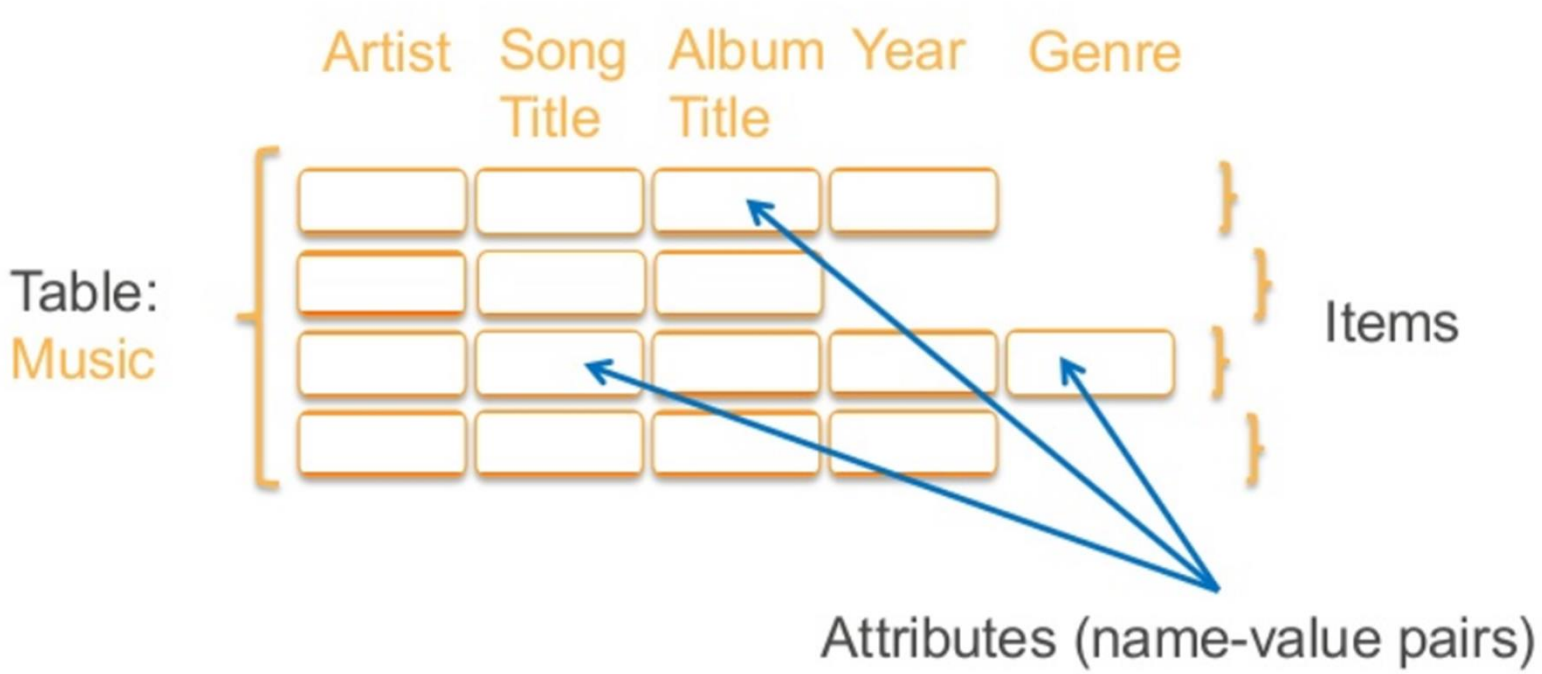
DynamoDB

Amazon DynamoDB

Amazon DynamoDB

- Allows you to store any amount of data with **no limits.**
- Provides fast, predictable performance using **SSDs.**
- Allows you to easily provision and change the **request capacity** needed for each table.
- Is a **fully managed**, **NoSQL** database service.

DynamoDB Data Model



DynamoDB Data Model

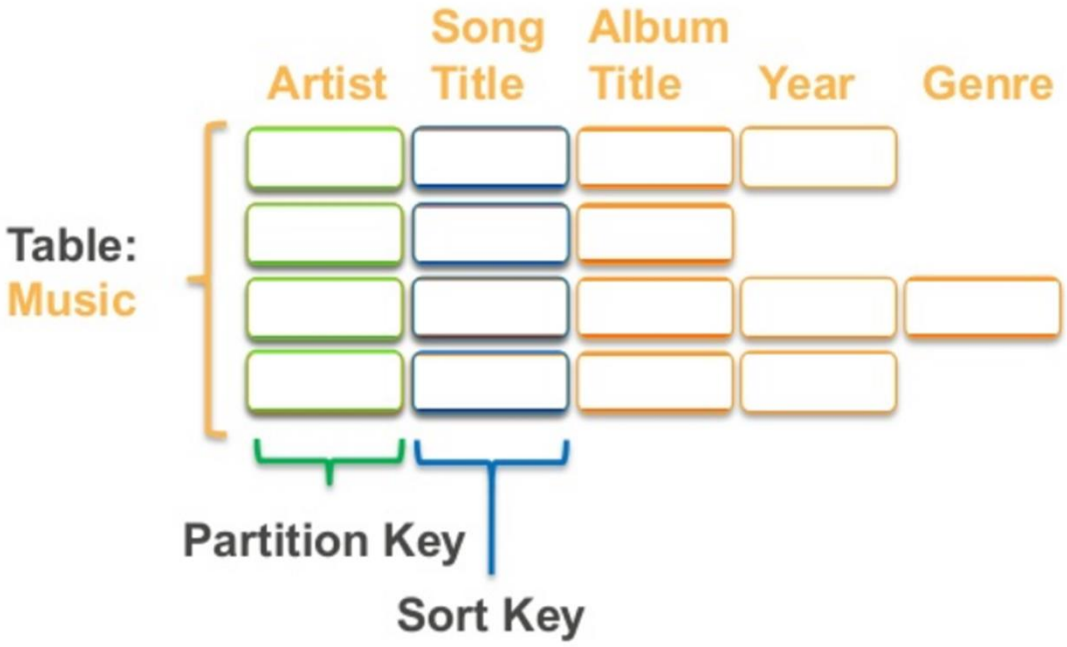
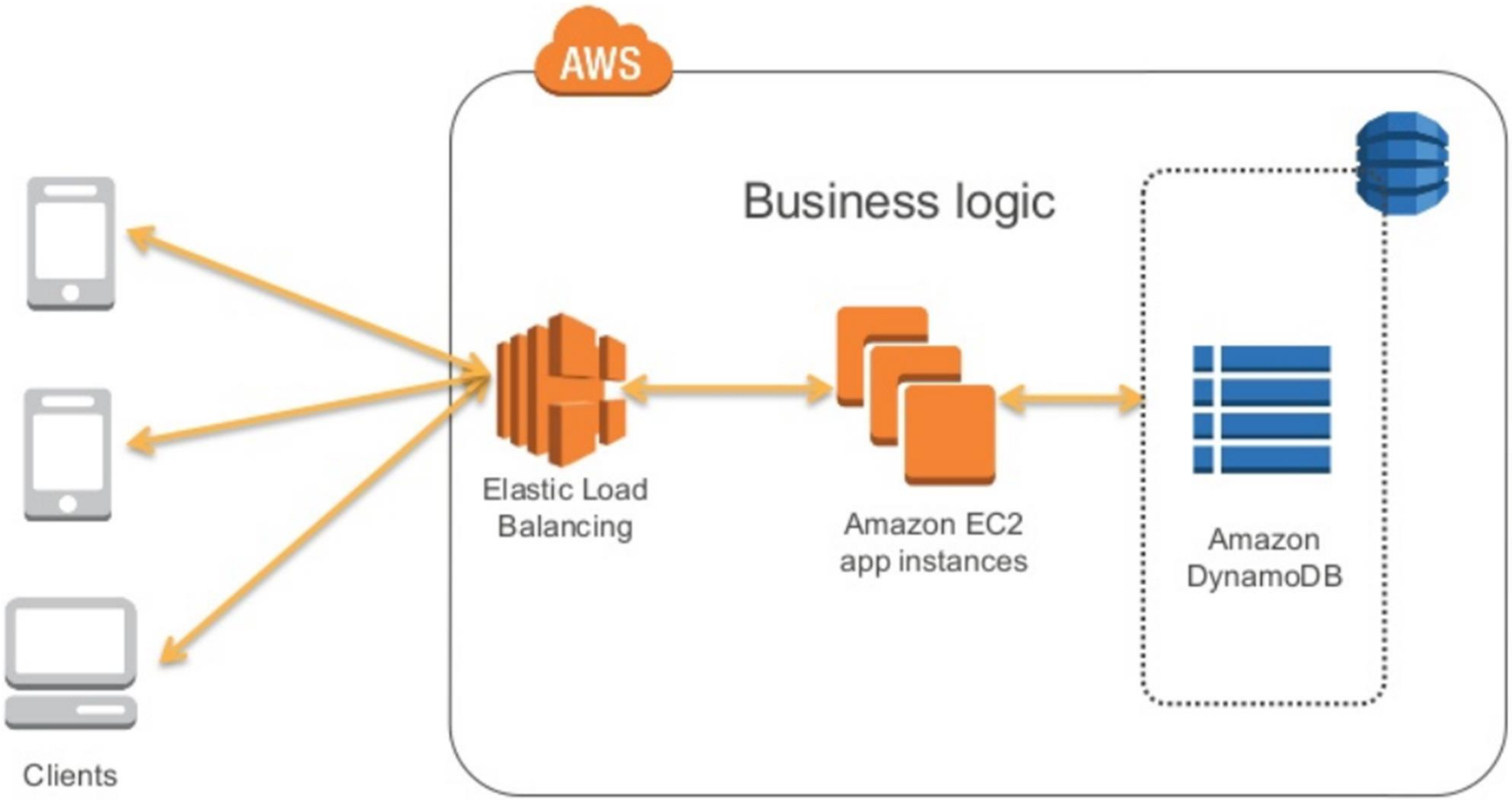


Table: Music
Partition Key: Artist
Sort Key: Song Title

(DynamoDB maintains a sorted index for both keys)

Simple Application Architecture



Supported Operations

- **Query**:
  - Query a table using the partition key and an optional sort key filter.
  - If the table has a secondary index, query using its key.
  - It is the **most efficient way to retrieve items** from a table or secondary index.
- **Scan**:
  - You can scan a table or secondary index.
  - Scan reads every item – **slower than querying**.
- You can use conditional expressions in both Query and Scan operations.

Amazon RDS vs Amazon DynamoDB

Factors	Relational (Amazon RDS)	NoSQL (Amazon DynamoDB)
Application Type	• Existing database apps • Business process–centric apps	• New web-scale applications • Large number of small writes and reads
Application Characteristics	• **Relational** data models, transactions • **Complex** queries, joins, and updates	• Simple data models, transactions • Range queries, simple updates
Scaling	Application or **DBA–architected** (clustering, partitions, sharding)	**Seamless, on-demand scaling** based on application requirements
QoS	• Performance–depends on data model, indexing, query, and storage optimization • Reliability and availability • Durability	• Performance–**Automatically optimized** by the system • Reliability and availability • Durability

Database Considerations

If You Need	Consider Using
A relational database service with minimal administration	**Amazon RDS** • Choice of Amazon Aurora, MySQL, MariaDB, Microsoft SQL Server, Oracle, or PostgreSQL database engines • Scale compute and storage • Multi-AZ availability 
A fast, highly scalable NoSQL database service	**Amazon DynamoDB** • Extremely fast performance • Seamless scalability and reliability • Low cost 
A database you can manage on your own	Your choice of **AMIs** on Amazon EC2 and Amazon EBS that provide scale compute and storage, complete control over instances, and more. 

Thank You!