



BITS Pilani

Cloud Computing

Session 9

IaaS (Databases – RDS, DynamoDB, RedShift, Aurora)

PaaS

Topics

- Amazon Relational Database Service (Amazon RDS)
- Amazon DynamoDB
- Amazon Redshift
- Amazon Aurora

Amazon Relational Database Service (Amazon RDS)

Unmanaged versus managed services



Unmanaged: Scaling, fault tolerance, and availability are managed by you



Managed: Scaling, fault tolerance, and availability are typically built into the service.



Challenges of relational databases

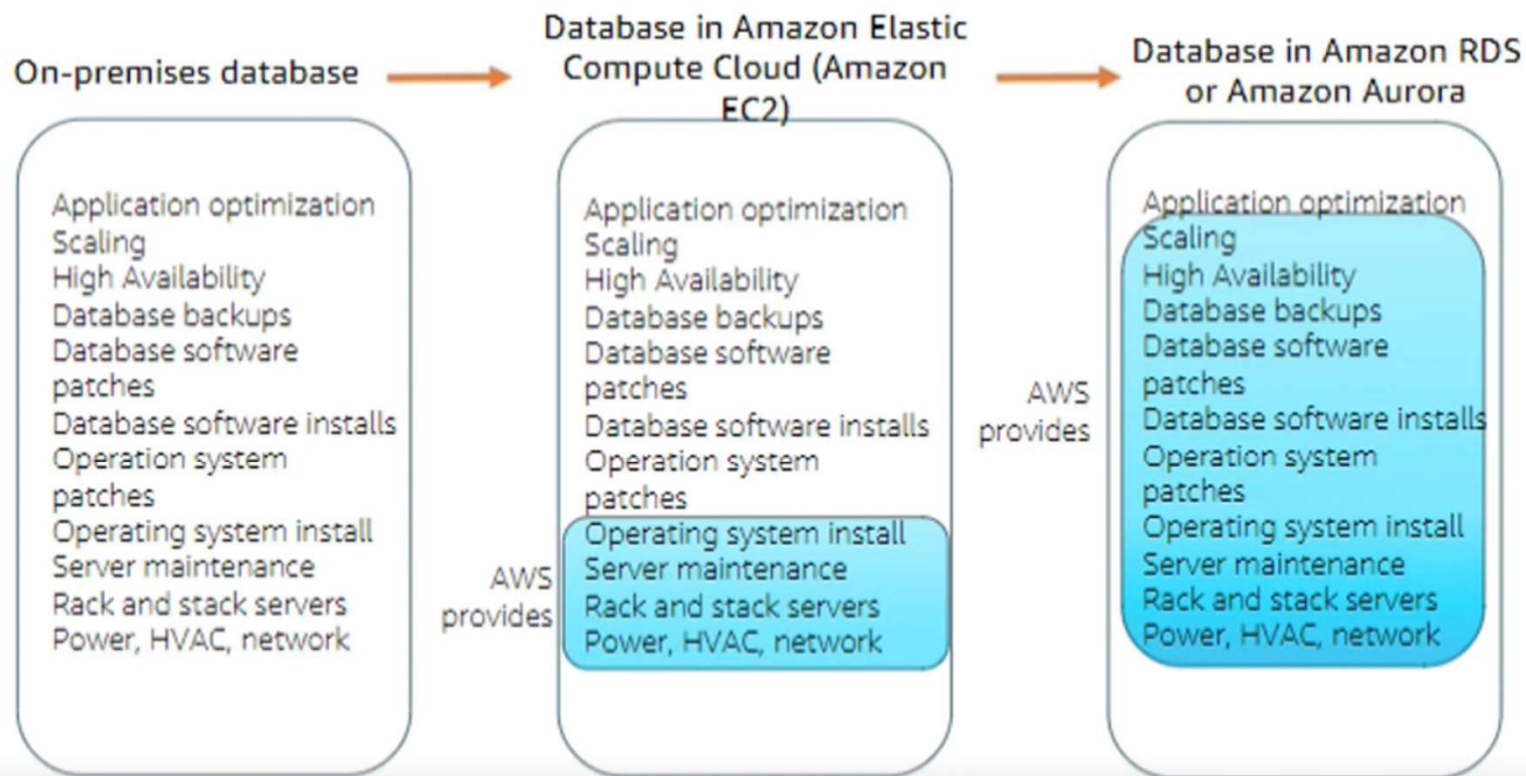
- Server maintenance and energy footprint
- Software installation and patches
- Database backups and high-availability
- Limits on scalability
- Data security
- Operating system(OS) installation and patches

Amazon RDS

Managed service that sets up and operates a relational database in the cloud.



From on-premises databases to Amazon RDS



Managed services responsibilities

You 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 instances

Amazon RDS



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

MySQL

Amazon Aurora

Microsoft SQL Server

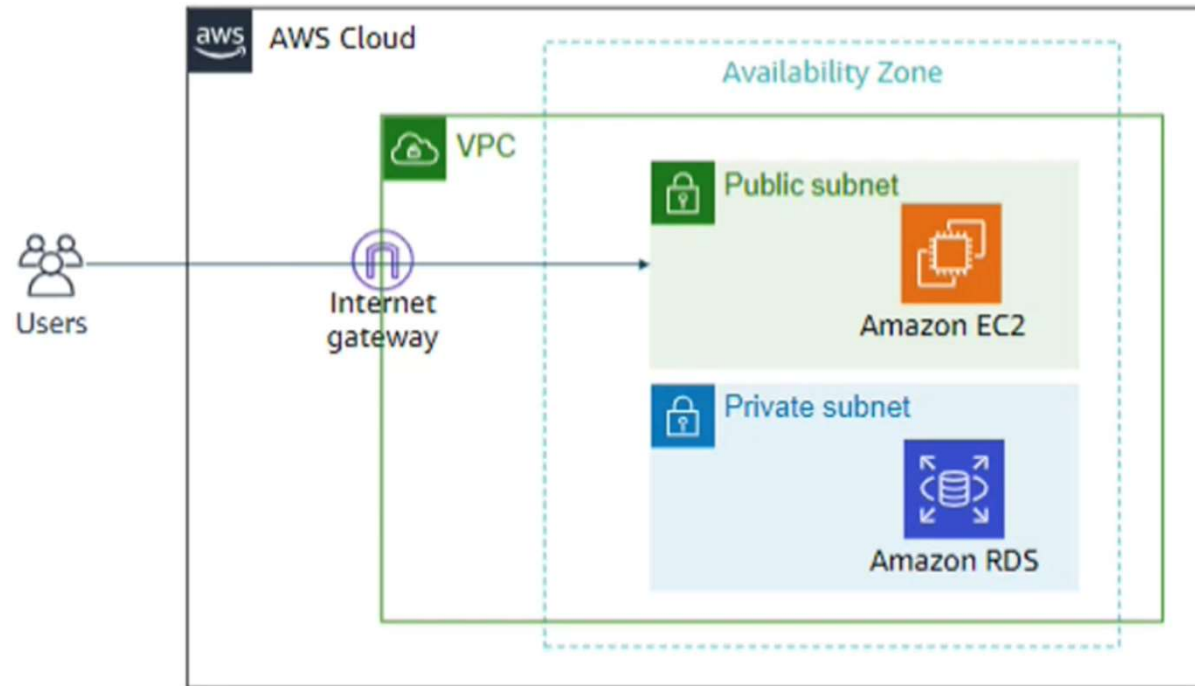
PostgreSQL

MariaDB

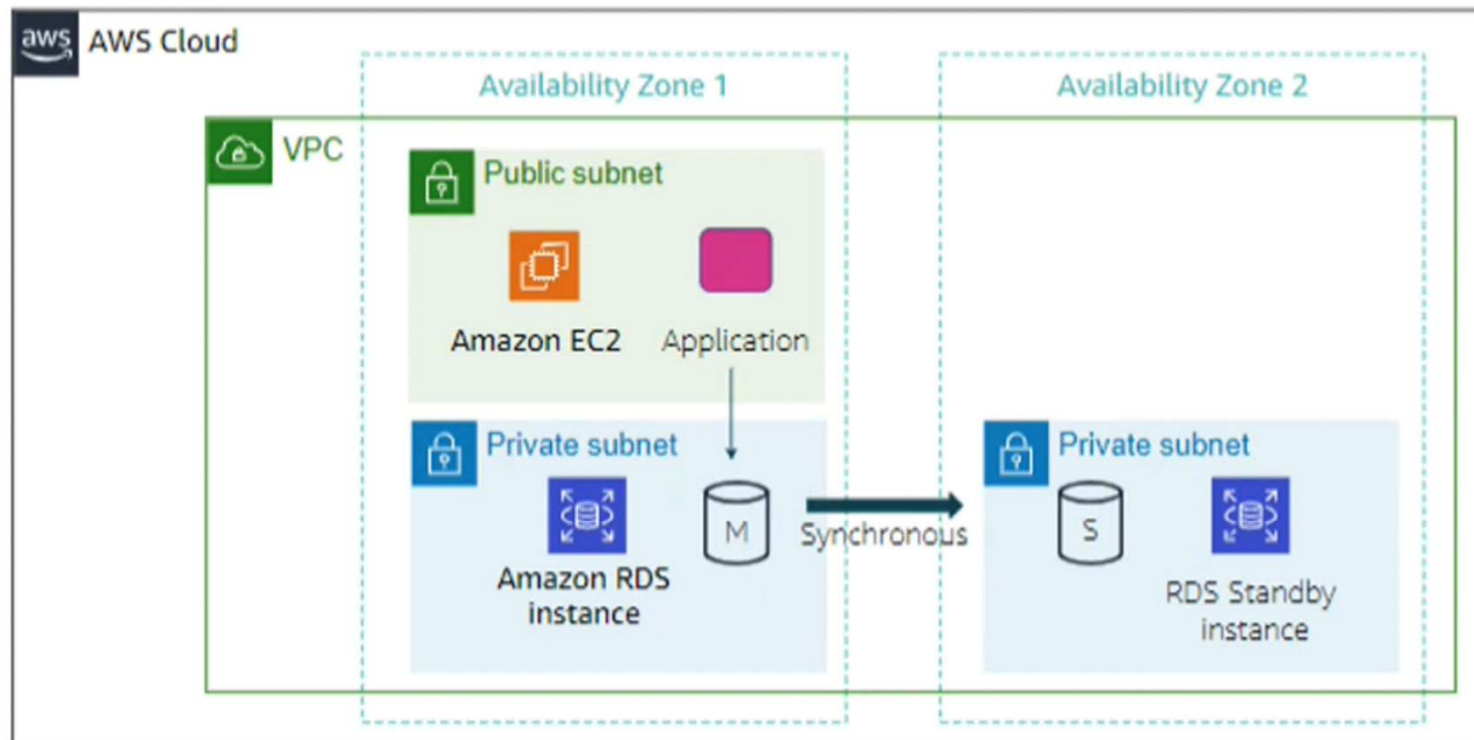
Oracle

DB engines

Amazon RDS in a virtual private cloud (VPC)



High availability with Multi-AZ deployment(1/2)



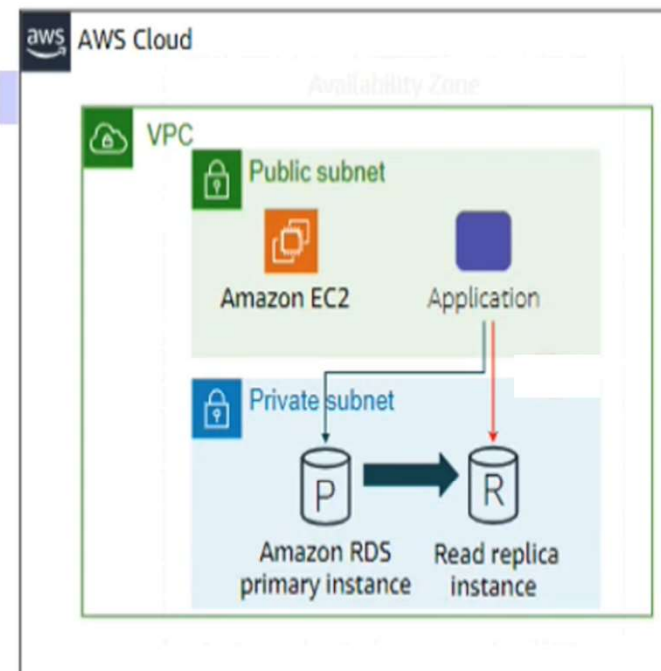
Amazon RDS read replicas

Features

- Offers asynchronous replication
- Can be promoted to primary if needed

Functionality

- Use for read-heavy database workloads
- Offload read queries



Use cases

Web and mobile applications	✓High throughput ✓Massive storage scalability ✓High availability
Ecommerce applications	✓Low-cost database ✓Data security ✓Fully managed solution
Mobile and online games	✓Rapidly grow capacity ✓Automatic scaling ✓Database monitoring

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: 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: Storage

Provisioned storage -

- No charge
 - 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 provisioned 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 you deploy to -

- Single Availability Zone
- Multiple Availability Zones

Data transfer -

- No charge for inbound data transfer
 - Tiered charges for outbound data transfer
-

Amazon DynamoDB

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	<table><tr><th>ISBN</th><th>Title</th><th>Author</th><th>Format</th></tr><tr><td>3111111223439</td><td>Withering Depths</td><td>Jackson, Mateo</td><td>Paperback</td></tr><tr><td>312222223439</td><td>Wily Willy</td><td>Wang, Xiulan</td><td>Ebook</td></tr></table>	ISBN	Title	Author	Format	3111111223439	Withering Depths	Jackson, Mateo	Paperback	312222223439	Wily Willy	Wang, Xiulan	Ebook	<div><pre>{ ISBN: 3111111223439, Title: "Withering Depths", Author: "Jackson, Mateo", Format: "Paperback" }</pre></div>
ISBN	Title	Author	Format											
3111111223439	Withering Depths	Jackson, Mateo	Paperback											
312222223439	Wily Willy	Wang, Xiulan	Ebook											

What is Amazon DynamoDB?

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

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
 - Sort key

Amazon DynamoDB

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	<table><tr><th>ISBN</th><th>Title</th><th>Author</th><th>Format</th></tr><tr><td>3111111223439</td><td>Withering Depths</td><td>Jackson, Mateo</td><td>Paperback</td></tr><tr><td>312222223439</td><td>Wily Willy</td><td>Wang, Xiulan</td><td>Ebook</td></tr></table>	ISBN	Title	Author	Format	3111111223439	Withering Depths	Jackson, Mateo	Paperback	312222223439	Wily Willy	Wang, Xiulan	Ebook	<div><pre>{ ISBN: 3111111223439, Title: "Withering Depths", Author: "Jackson, Mateo", Format: "Paperback" }</pre></div>
ISBN	Title	Author	Format											
3111111223439	Withering Depths	Jackson, Mateo	Paperback											
312222223439	Wily Willy	Wang, Xiulan	Ebook											

Amazon DynamoDB core components

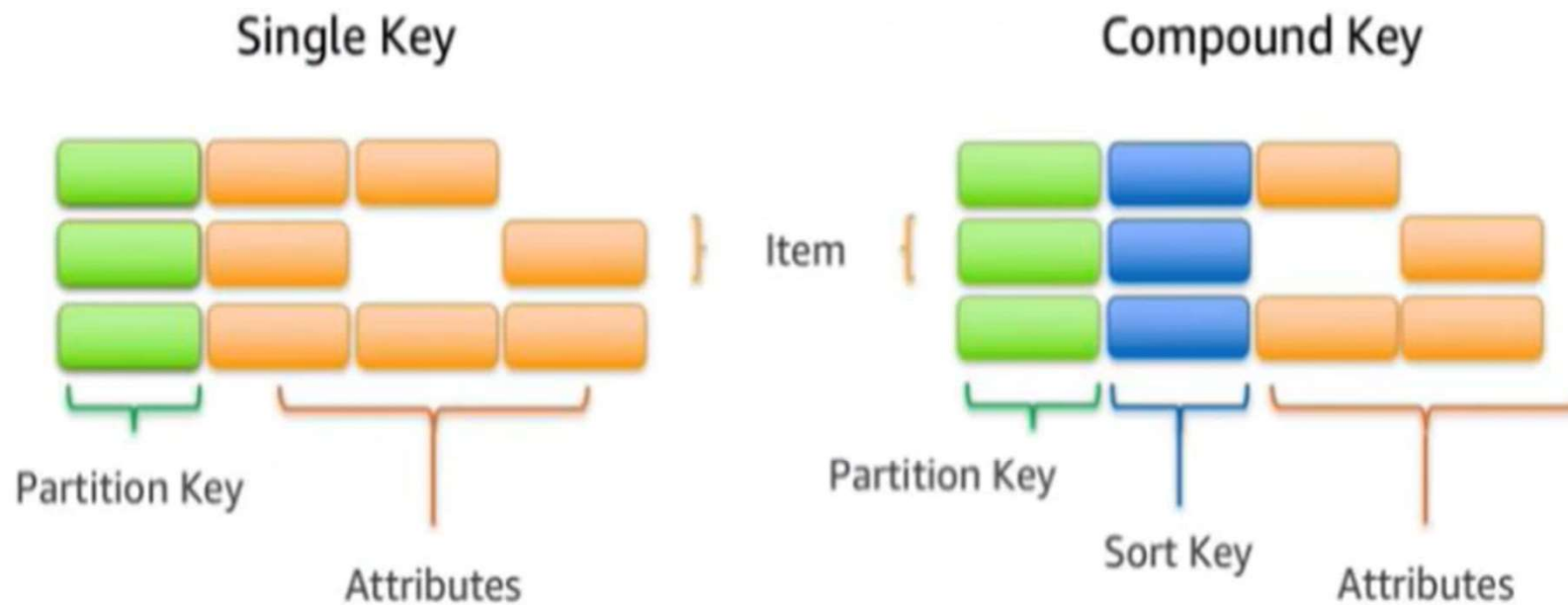
- Tables, items, and attributes are the core DynamoDB components
- DynamoDB supports two different kinds of primary keys:
 - Partition key and partition
 - Sort key

Partitioning

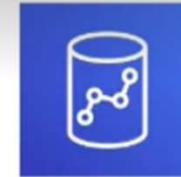
As data grows, table partitioned by key

QUERY by Key to find items efficiently
SCAN to find items by any attribute

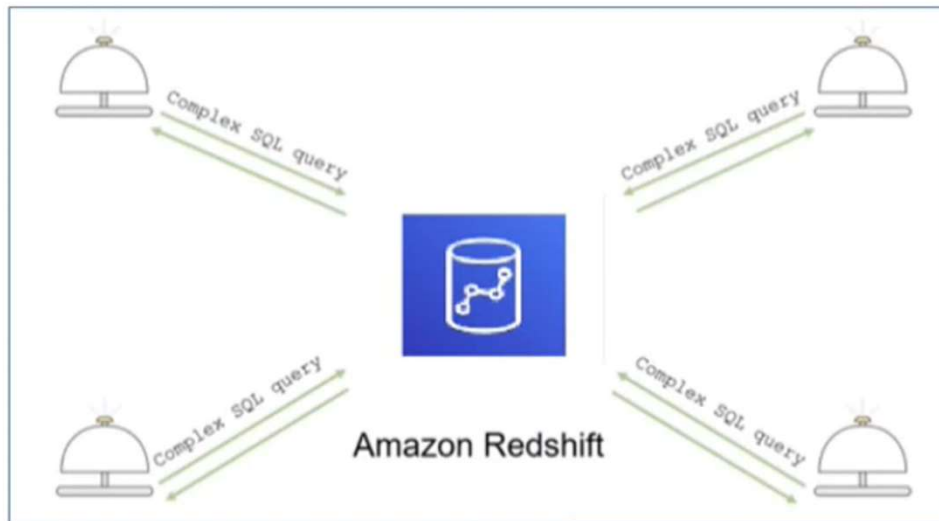
Items in a table must have a key



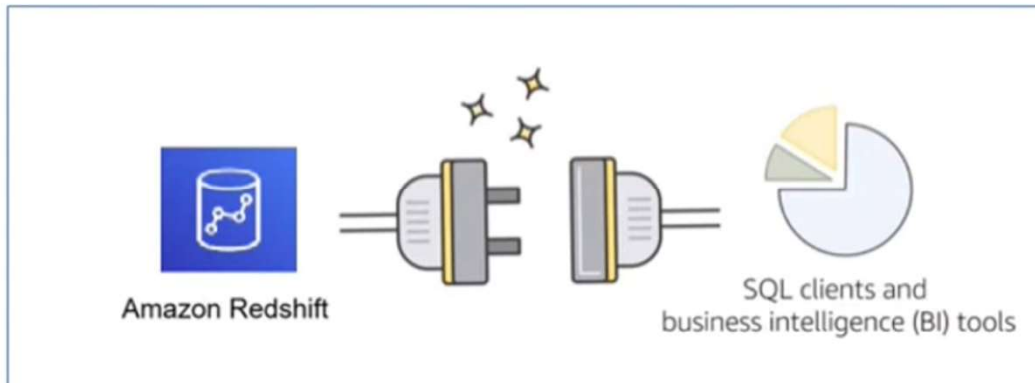
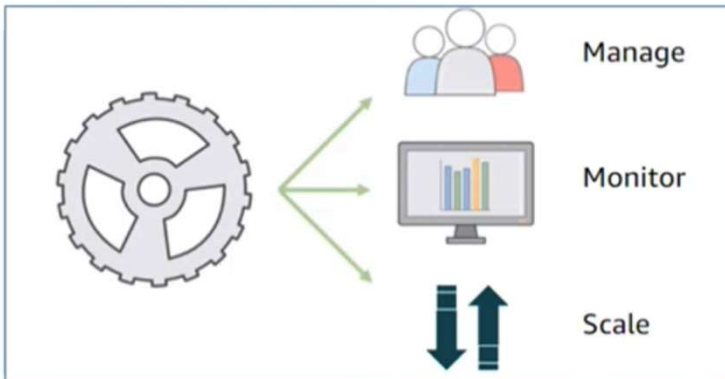
Amazon Redshift



Amazon Redshift



Features



Amazon Redshift use cases

Enterprise dataware house(EDW)

- Migrate at a pace that customers are comfortable with
- Experiment without large up front 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

Software as a service(SaaS)

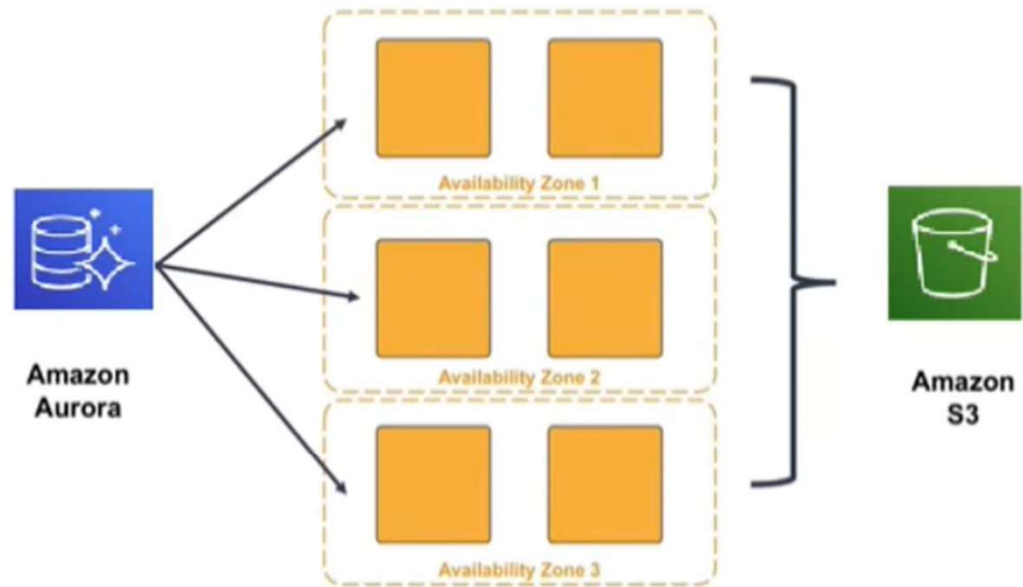
- Scale the dataware house capacity as demand grows
- Add analytic functionality to applications
- Reduce hardware and software costs

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).
-

High availability



Resilient design



The right tool for the right job

What are my requirements?

Enterprise-class relational database - **Amazon RDS**

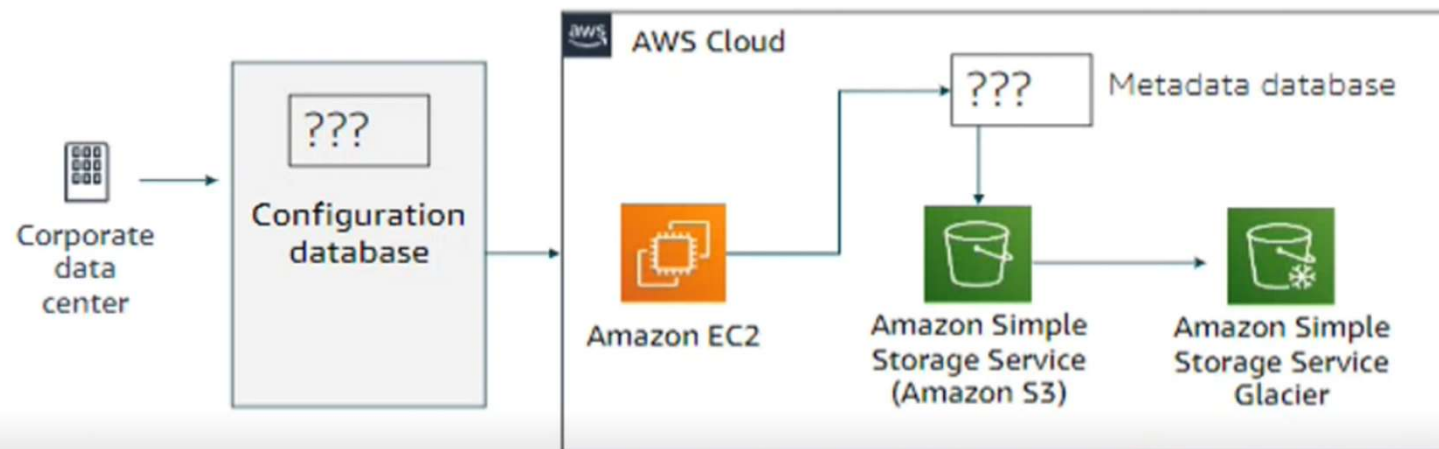
Fast and flexible NoSQL database service for any scale - **Amazon Dynamo DB**

Operating system accessor application features that are not supported by AWS database services - **Databases on Amazon EC2**

Specific case-driven requirements (machine learning, data warehouse, graphs) - **AWS purpose-built database services**

Database case study

Case 1: A data protection and management company that provides services to enterprises. They must provide database services for over 55 petabytes of data. They have two types of data that require a database solution. First, they need a relational database store for configuration data. Second, they need a store for unstructured metadata to support a de-duplication service. After the data is de-duplicated, it is stored in Amazon S3 for quick retrieval, and eventually moved to Amazon S3 Glacier for long-term storage. The following diagram illustrates their architecture



Thanks.