

CS 4037

# Introduction to Cloud Computing

## Lecture 28

**Danyal Farhat**  
**FAST School of Computing**  
**NUCES Lahore**

# AWS Databases

# Lecture's Agenda

- **Amazon Relational Database Service**
- Amazon DynamoDB
- Amazon Redshift
- Amazon Aurora



# Unmanaged vs. Managed Services

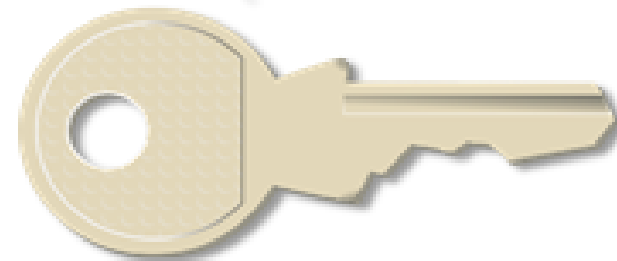
## Unmanaged:

*Scaling, fault tolerance, and availability are managed by customer.*



## Managed:

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

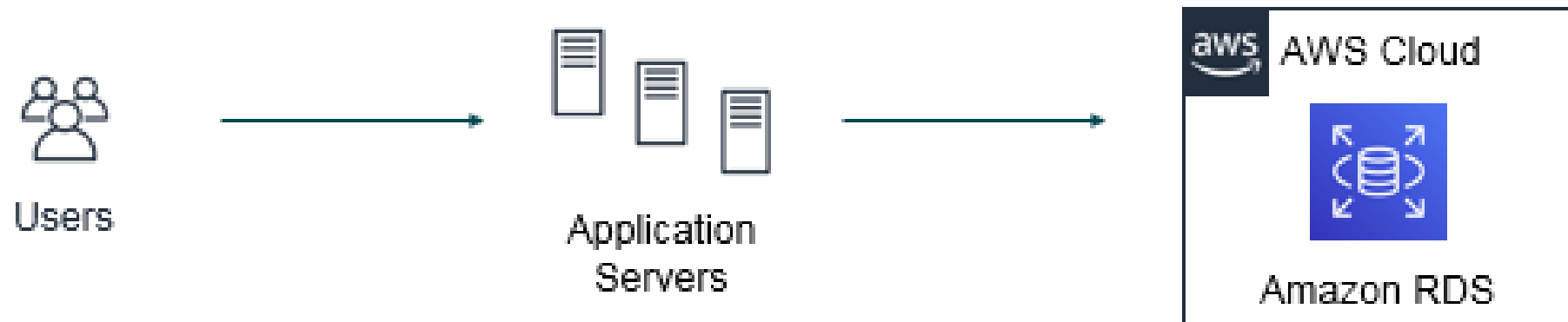


# Challenges of Unmanaged Relational Databases

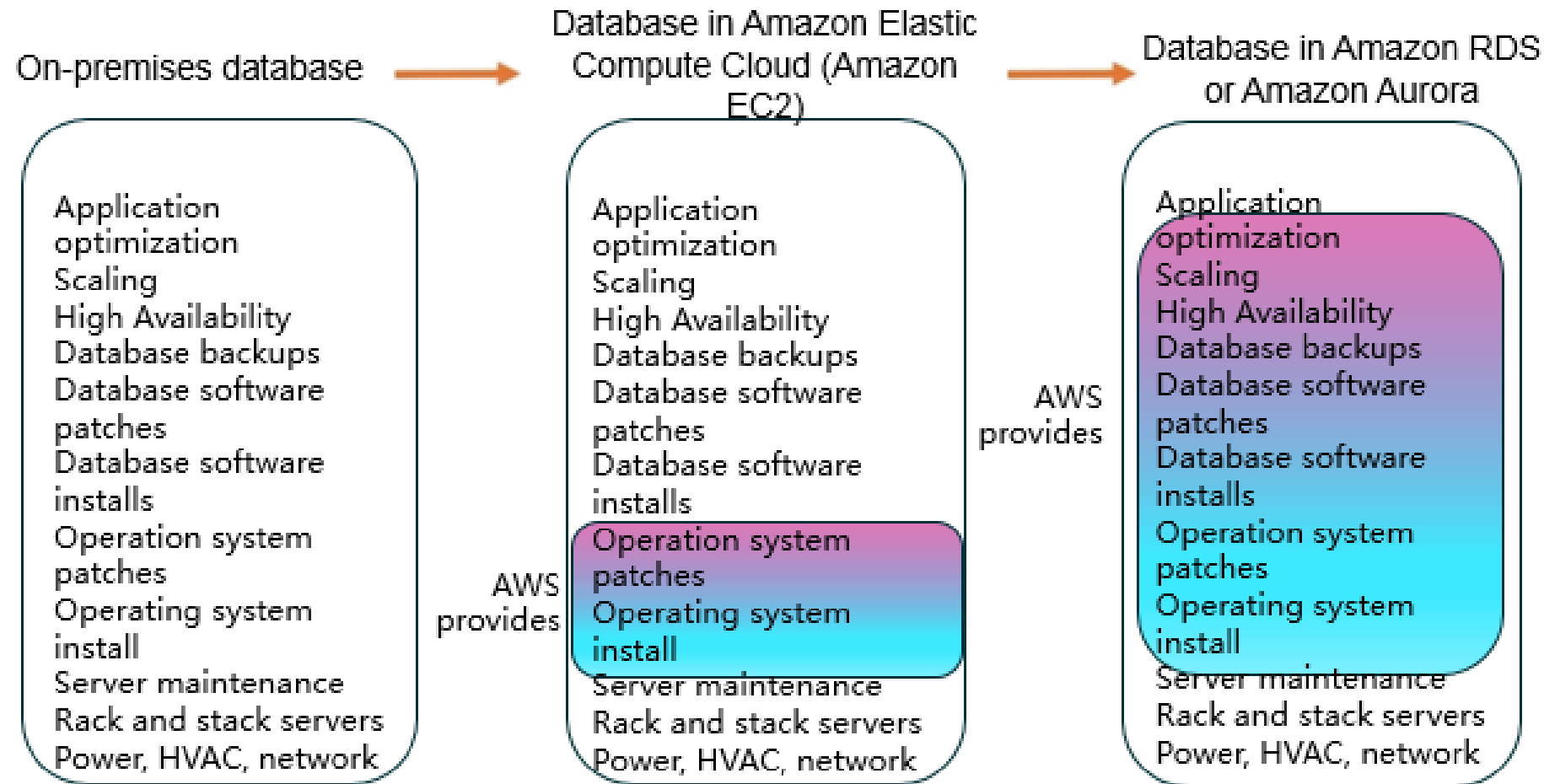
- **Server maintenance and energy footprint**
- **OS installation and patches**
- **Software installation and patches**
- **Database backups and high availability**
- **Limits on scalability**

# Amazon RDS

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



# From On-Premises Databases to Amazon RDS



# 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

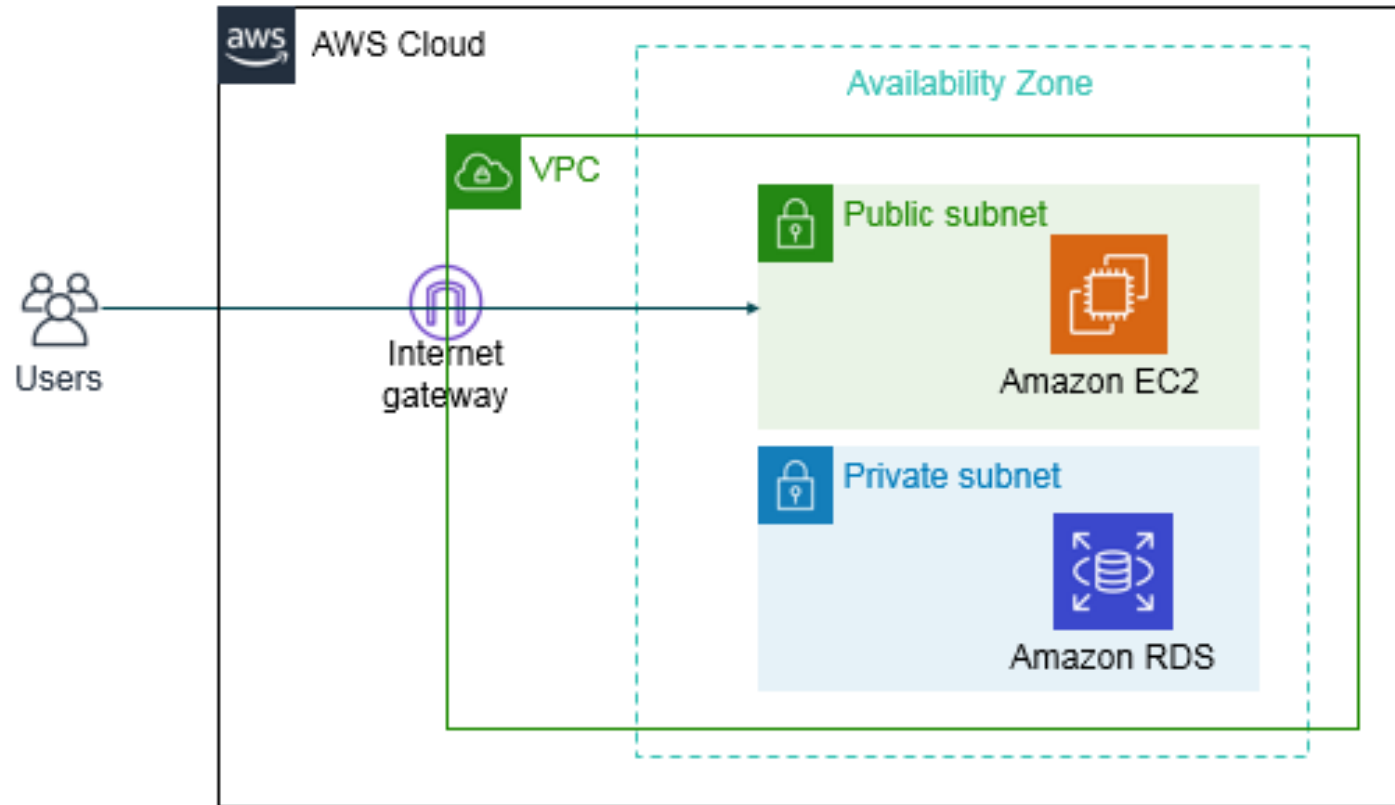
PostgreSQL

MariaDB

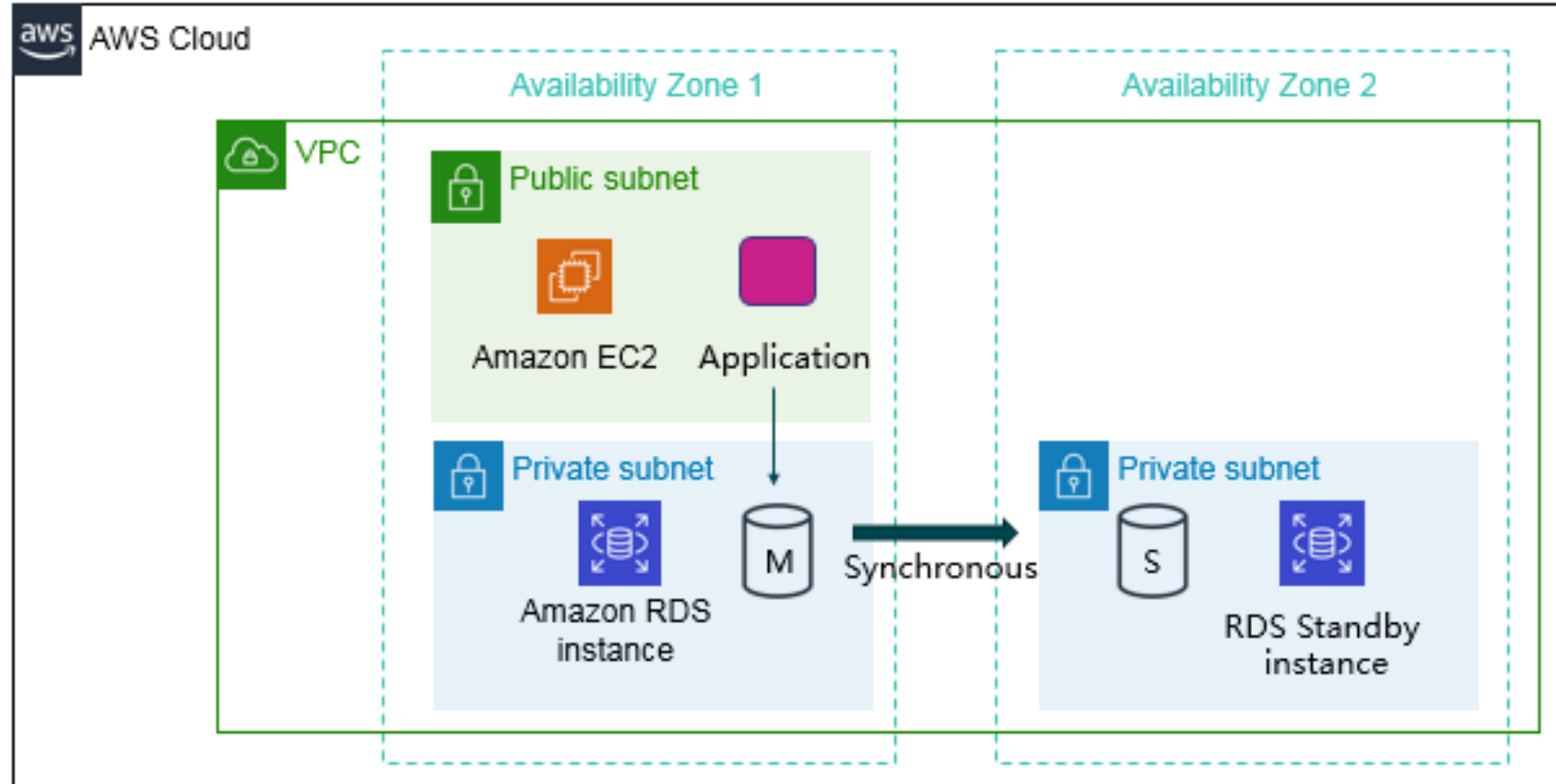
Oracle

DB engines

# Amazon RDS in a VPC

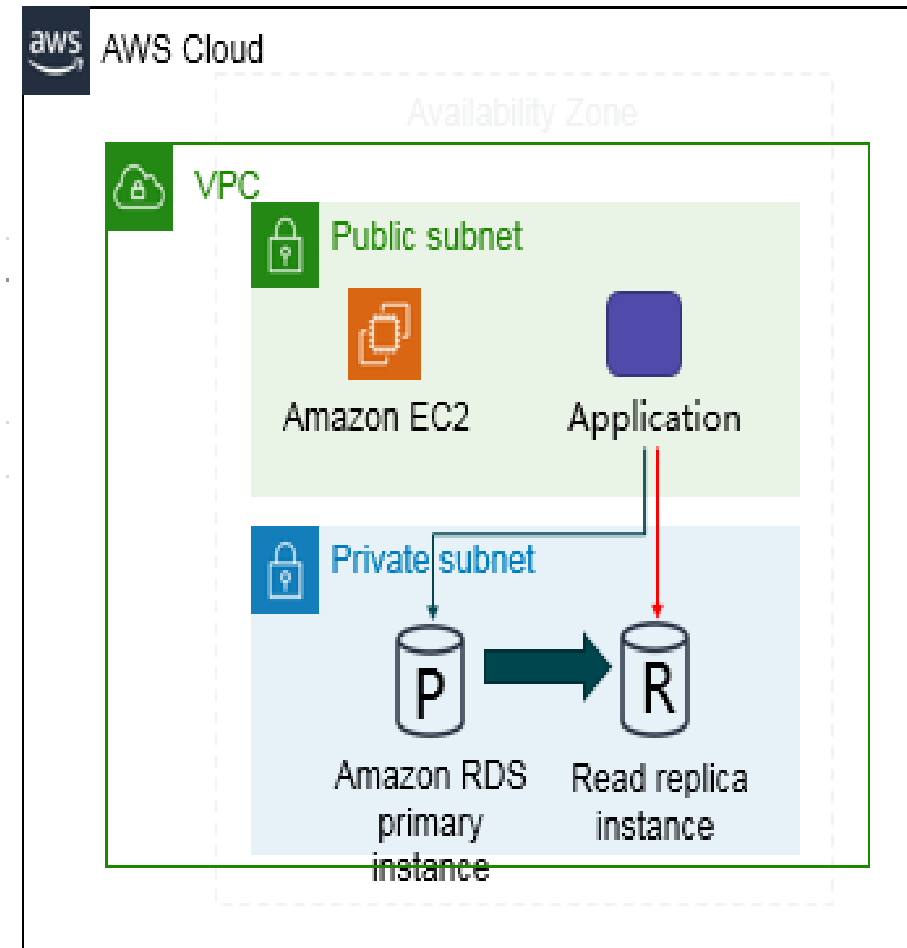


# High Availability with Multi-AZ Deployment



# Amazon RDS Read Replica

- Offers **asynchronous** replication
  - MySQL, MariaDB, PostgreSQL, and Aurora
- Updates that are made to the source database instance are asynchronously copied to the **read replica** instance
- Customer can reduce load on source database instance by **routing read queries** from applications to the read replica
- Can be **promoted to primary** if needed
- Can be created in a **different Region** than the primary database
- Use for **read-heavy database** workloads



# When to Use Amazon RDS

**Use Amazon RDS** when your application requires:

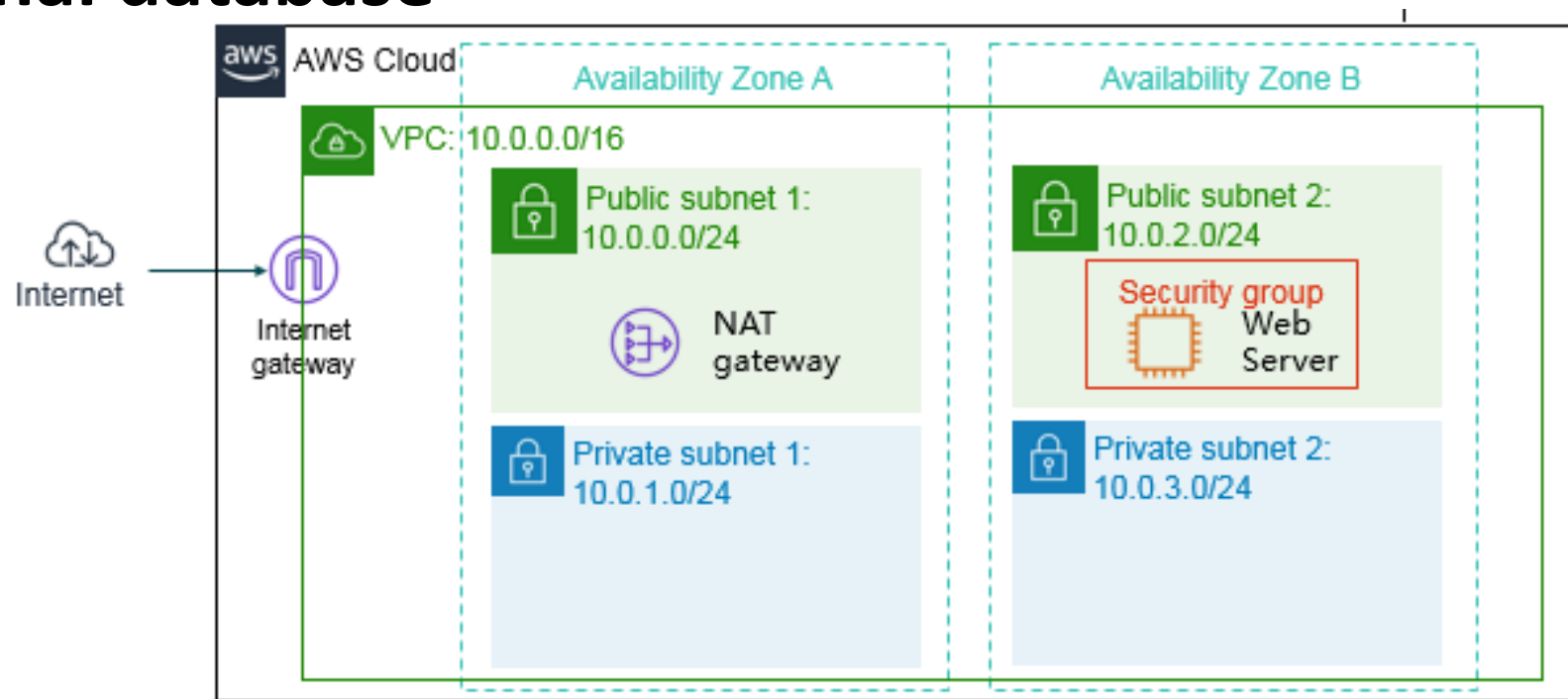
- Complex transactions or **complex queries**
- A medium to high query or write rate – **Up to 30,000 IOPS** (15,000 reads + 15,000 writes)
- No more than a **single worker node**
- High **durability**

**Do not use Amazon RDS** when your application requires:

- **Massive read/write rates** (for example, 150,000 write/second)
- High data size or **throughput demands**
- Simple **GET or PUT requests** and queries that a database can handle NoSQL
- Relational database management system (RDBMS) **customization**

# Lab 5: Build a Database Server

**Lab Scenario:** This lab is designed to show you how to use an AWS managed database instance to solve a need for a relational database



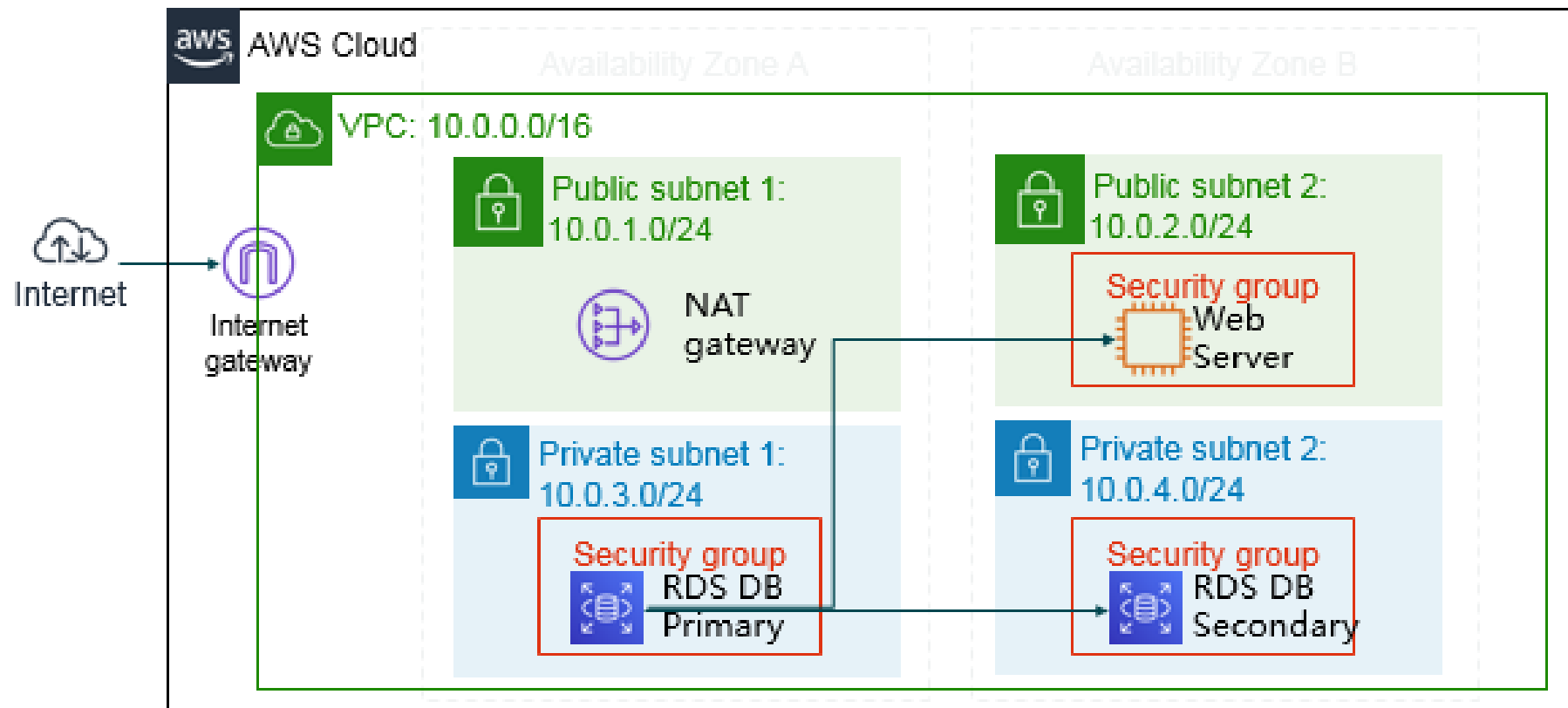
# Lab 5: Build a Database Server (Cont.)

## Lab Tasks:

- **Task 1 – Create a VPC security group**
- **Task 2 – Create a DB subnet group**
- **Task 3 – Create an Amazon RDS instance**
- **Task 4 – Interact with the Amazon RDS instance**

# Lab 5: Build a Database Server

## Final Product:



# Lecture's Agenda

- Amazon Relational Database Service
- **Amazon DynamoDB**
- Amazon Redshift
- Amazon Aurora



# Relational (Str.) vs. Non-relational (Semi-Str.) 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>3122222223439</td><td>Wily Willy</td><td>Wang, Xiulan</td><td><u>Ebook</u></td></tr></table>	ISBN	Title	Author	Format	3111111223439	Withering Depths	Jackson , Mateo	Paperback	3122222223439	Wily Willy	Wang, Xiulan	<u>Ebook</u>	<div><pre>{   ISBN: 3111111223439,   Title: "Withering Depths",   Author: "Jackson, Mateo",   Format: "Paperback" }</pre></div>
ISBN	Title	Author	Format											
3111111223439	Withering Depths	Jackson , Mateo	Paperback											
3122222223439	Wily Willy	Wang, Xiulan	<u>Ebook</u>											

# Amazon DynamoDB

- Fast and flexible **NoSQL database** service for any scale
- NoSQL database **tables**
- Virtually **unlimited** storage
- Items can have **differing attributes**
- Runs **exclusively** on SSDs
- Supports document and **key-value store** models
- **Replicates tables** across user's choice of AWS Regions



amazon DynamoDB

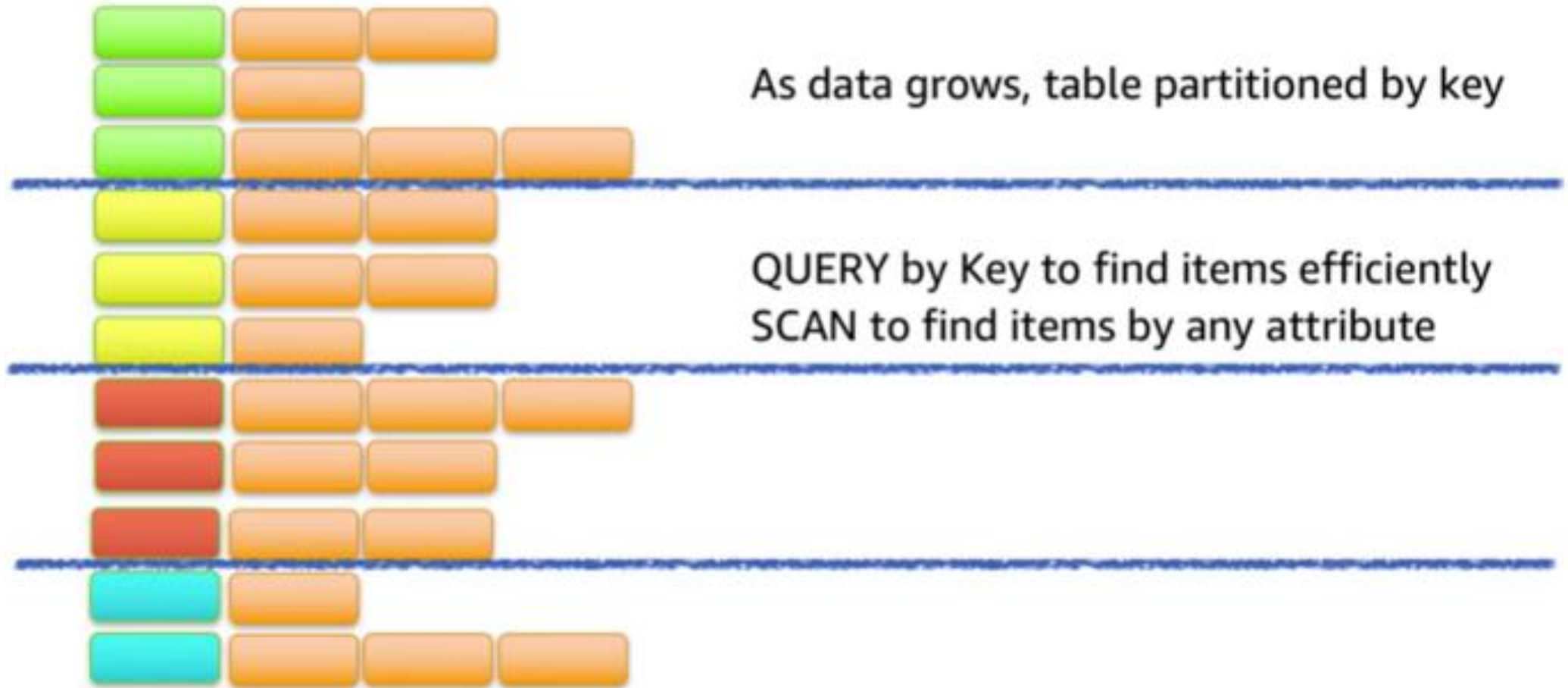
# Amazon DynamoDB (Cont.)

- Has **no limits** on table size or throughput
- **Accessible** via the console, AWS CLI, and API calls
- Provides **consistent, single-digit** millisecond latency
- Works well for **mobile, web, gaming, adtech, and IoT** applications

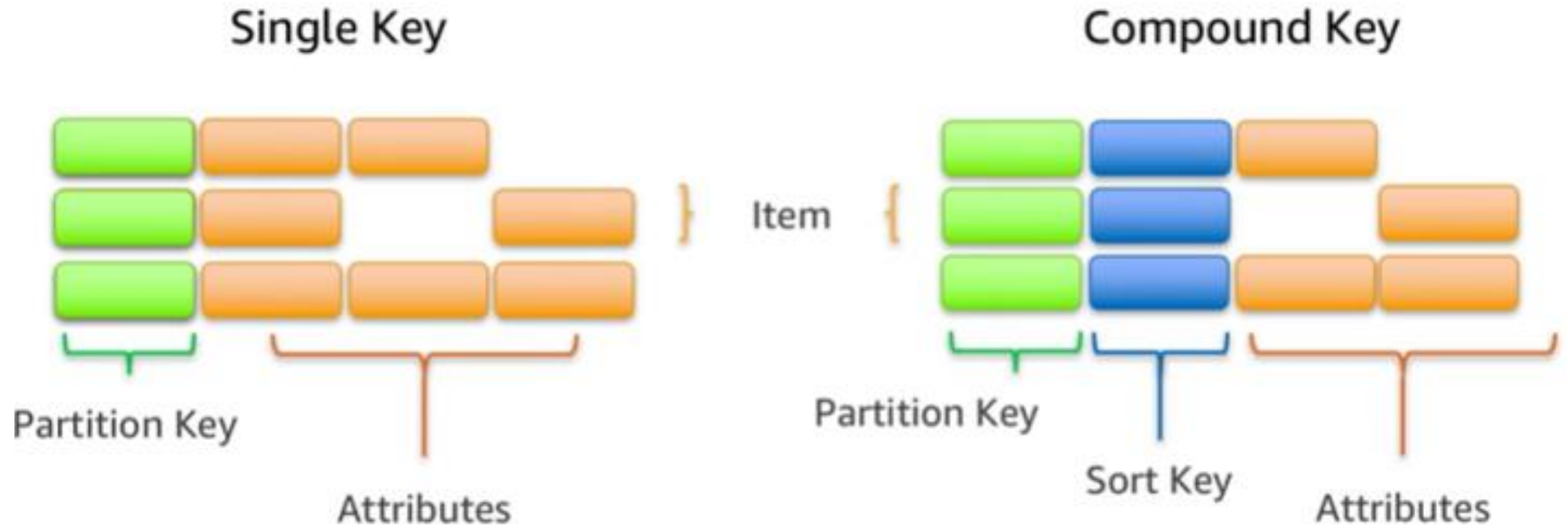
# Amazon DynamoDB Core Components

- Tables, items, and attributes are the **core** DynamoDB components
  - A **table** is a collection of data
  - **Items** are a group of attributes that is uniquely identifiable among all the other items
  - Attributes are a **fundamental data element**, something that does not need to be broken down any further.
    - ✓ Friends table, Items [Ali, Umar], Attribute [Name, CNIC, Contact, Address]
- DynamoDB supports **two different kinds** of primary keys
  - Partition key and partition and sort key

# Partitioning



# Items in a table must have a key



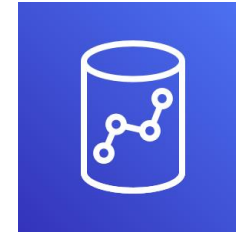
# Lecture's Agenda

- Amazon Relational Database Service
- Amazon DynamoDB
- **Amazon Redshift**
- Amazon Aurora



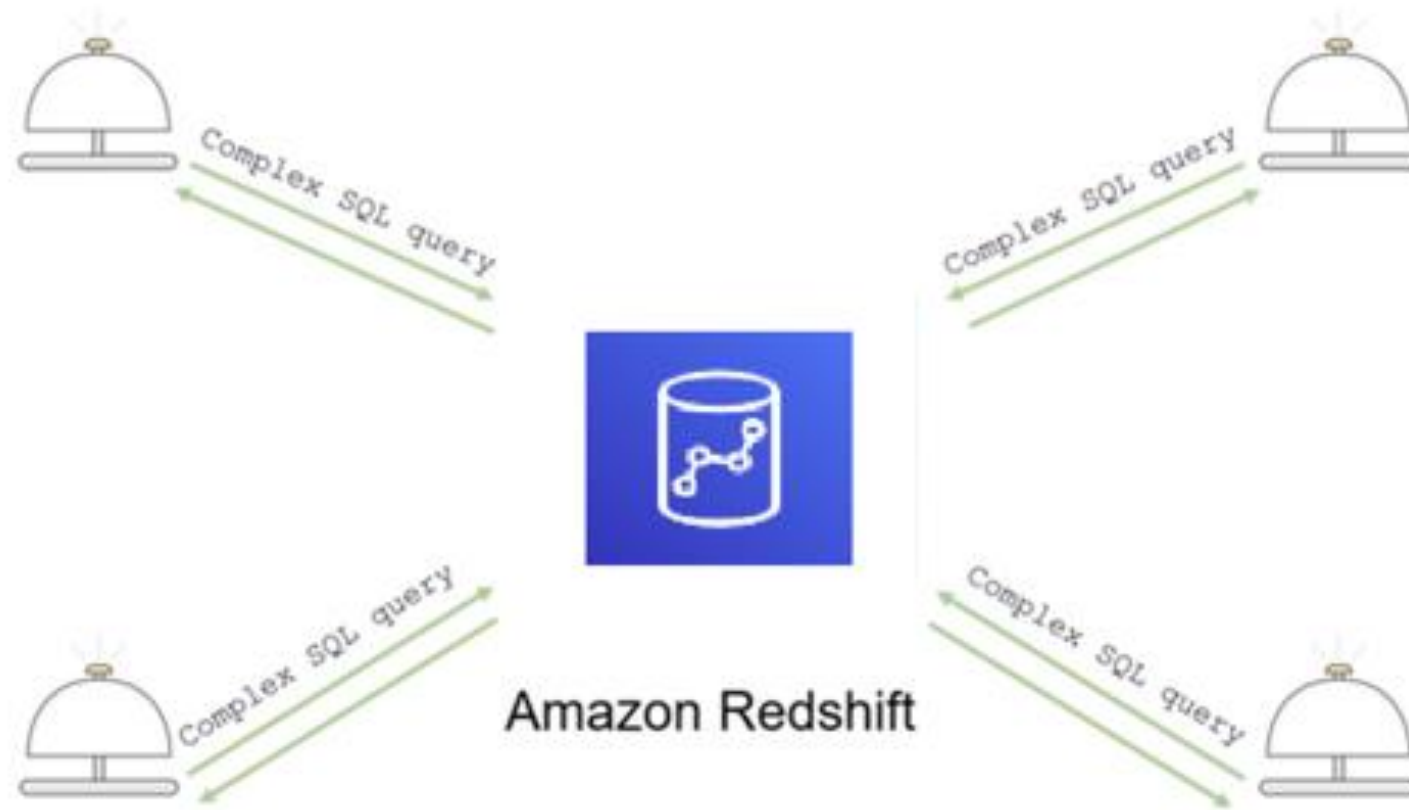
# Amazon Redshift

- Fast, fully managed data warehouse
- Simple and cost-effective DWH option
- Analyze petabytes of data by using standard SQL and existing business intelligence (BI) tools
- Example of Data Warehouse Tools: Teradata, Oracle DB, Apache Hive

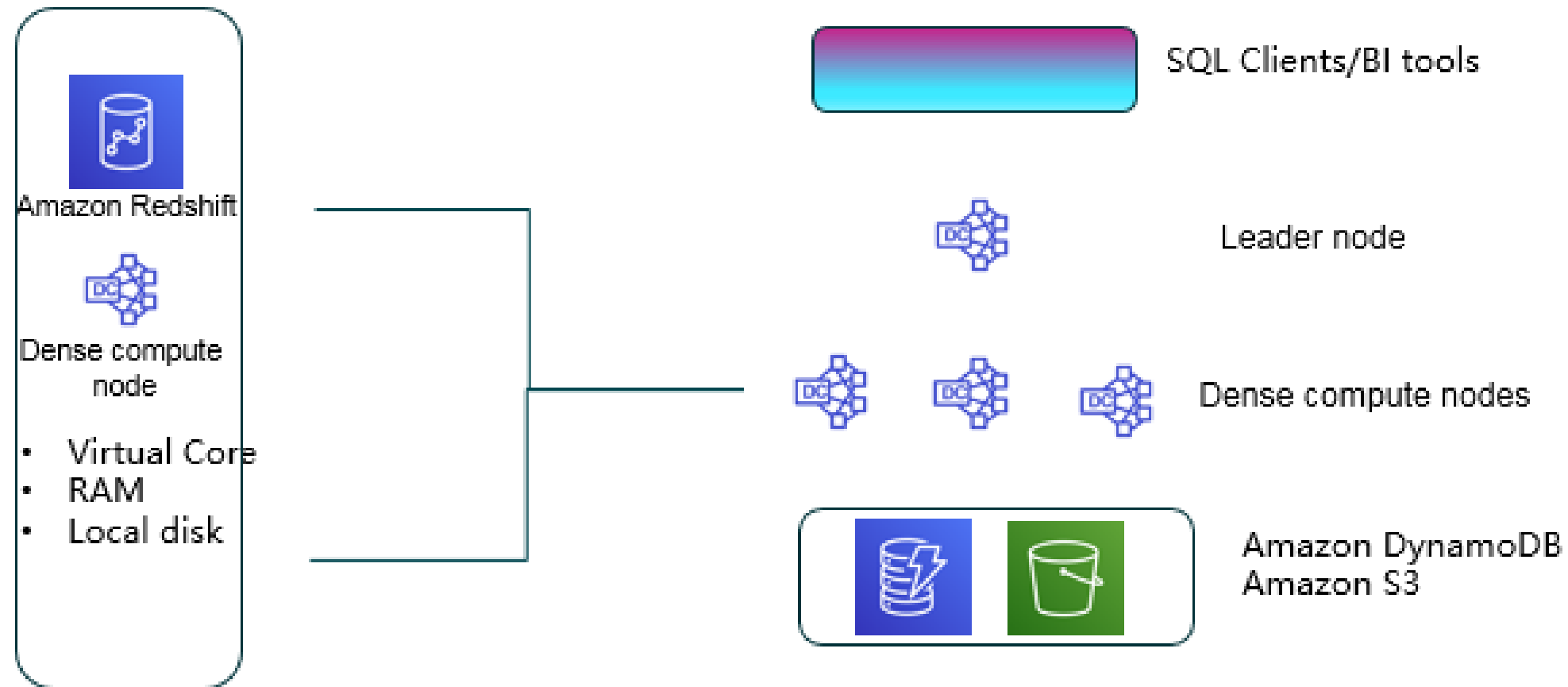


Amazon Redshift

# Introduction to Amazon Redshift

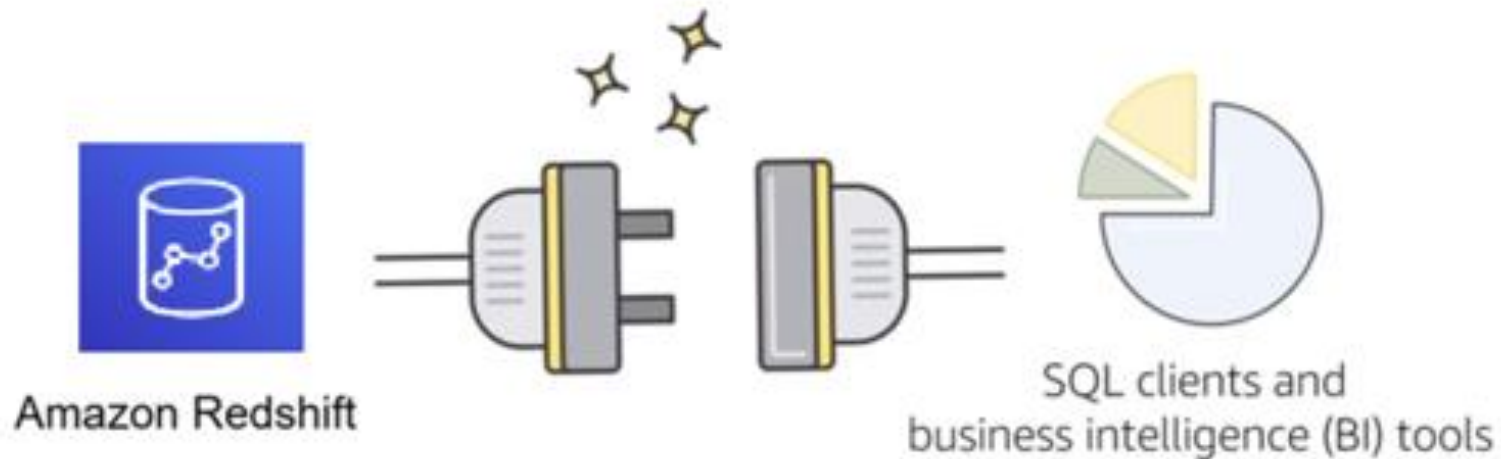


# Parallel Processing Architecture



# Redshift Compatibility

- Redshift is **compatible** with SQL, JDBC and ODBC connectors, which enables to use SQL clients and BI tools



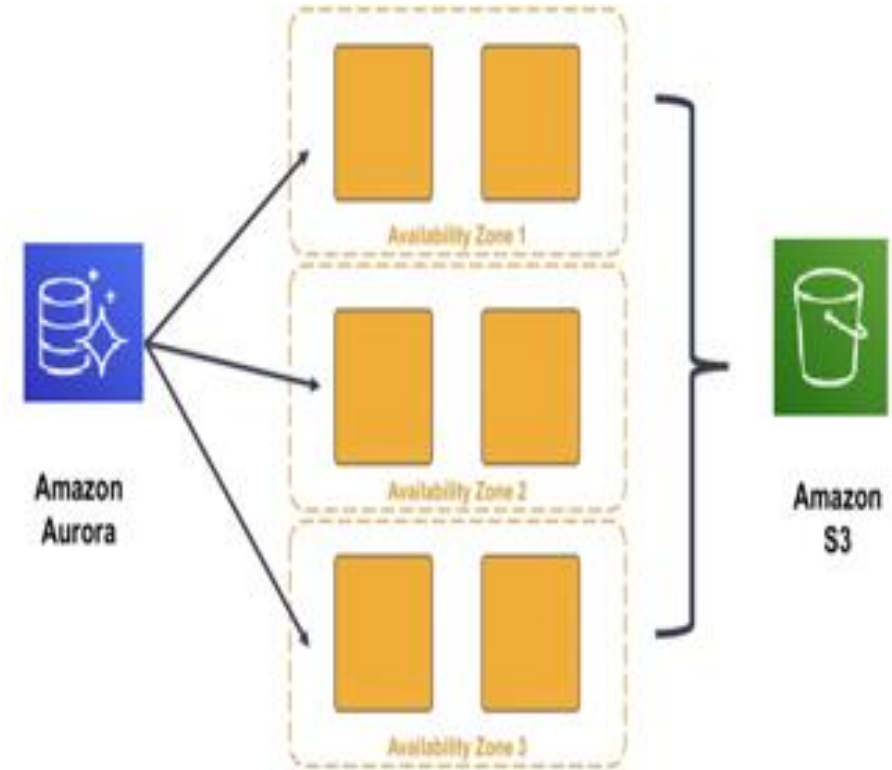
# Lecture's Agenda

- Amazon Relational Database Service
- Amazon DynamoDB
- Amazon Redshift
- **Amazon Aurora**



# Amazon Aurora

- Enterprise-class relational DB
  - Compatible with **MySQL** or **PostgreSQL**
- Stores **copies of data** across multiple Availability Zones
  - With continuous **backups** to S3
- Can use **up to 15** read replicas
- **Instant crash recovery** if primary DB becomes unhealthy



# The right tool for the right job

What are my requirements?	
Enterprise-class relational database	Amazon RDS / Amazon Aurora
Fast and flexible NoSQL database service for any scale	Amazon DynamoDB
Operating system access or 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 e.g. Redshift

# Additional Resources

- **AWS Database Page**
  - <https://aws.amazon.com/products/databases/>
- **Amazon RDS Page**
  - <https://aws.amazon.com/rds/>
- **Overview of Amazon Database Services**
  - <https://docs.aws.amazon.com/whitepapers/latest/aws-overview/database.html>
- **Getting started with AWS Databases**
  - <https://aws.amazon.com/products/databases/learn/>

Questions?