AWS Solutions Architect

Databases

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
  cout << "Hello, world!" << endl;
  return 0;
}
```

A Day in the Life of Cloud Architect

You are working as a Cloud Architect at a company, and you have been asked to design a database solution for a product with their specifications and store them in DynamoDB.

The product specification must include product catalogs, sales areas, and customer information, all of which must be stored in MYSQL.

To achieve all the above, along with some additional concepts, we would be learning a few concepts in this lesson that will help you find a solution for the above scenario.

Learning Objectives

By the end of this lesson, you will be able to:

- Create DynamoDB database
- Create DynamoDB global table Replica
- Create a Redshift database and query data from the database
- Create an MYSQL database

TECHNOLOGY

Introduction to Databases

Databases Overview

AWS provides the broadest selection of purpose-built databases allowing to save, grow, and innovate faster.

Features of AWS databases

Easy Migration

High performance scalability

Highly versatile

Time and cost efficient

Types of AWS Databases

In AWS, databases can be categorized into the following types:

Relational

Traditional applications, ERP, CRM, and e-commerce

Key-value

High-traffic web applications, e-commerce, and gaming applications

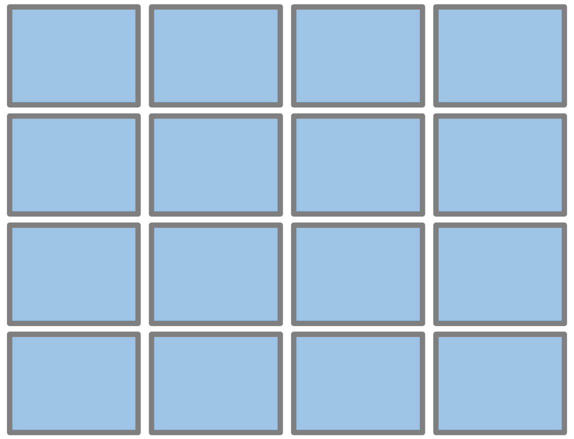
In-memory

Caching, session management, and geospatial applications

Relational Databases

The most common form of database is relational databases or SQL databases.

Structured data

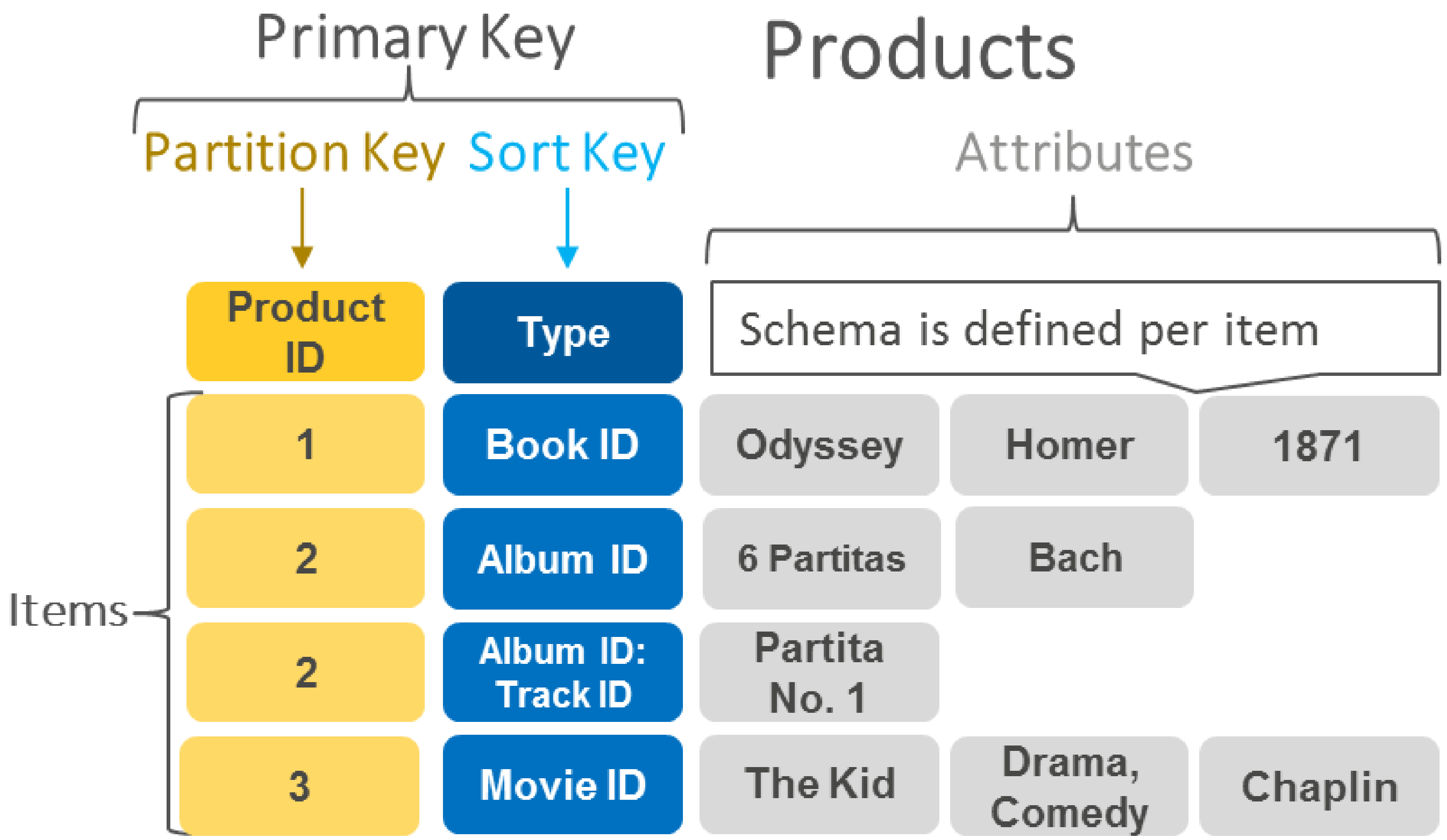


Relational database

- A relational database is a collection of data items organized as a set of formally-described tables which is used to store structured data.
- It is also known as the relational model.

Key-Value Database

A key-value database is a type of non-relational database that uses a simple key-value method to store data.

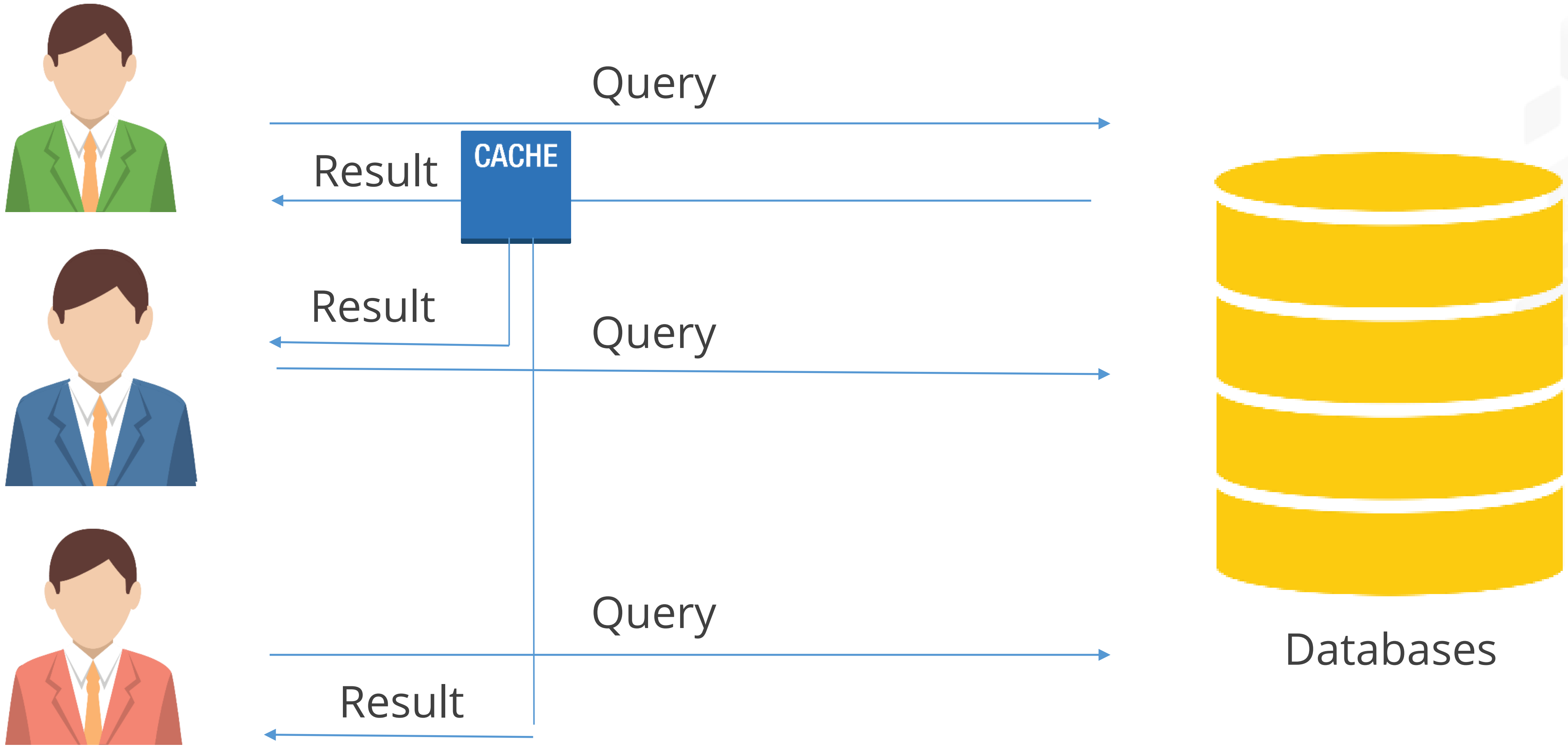


Product ID	Type	Schema is defined per item		
1	Book ID	Odyssey	Homer	1871
2	Album ID	6 Partitas	Bach	
2	Album ID: Track ID	Partita No. 1		
3	Movie ID	The Kid	Drama, Comedy	Chaplin

Source: https://aws.amazon.

In-Memory Databases

In-memory databases are cache-based databases that store results in memory to reduce the load on your database infrastructure and to improve user response time.



TECHNOLOGY

Amazon Relational Database Service (RDS)

Amazon RDS

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easy to set up, operate, and scale a relational database in the AWS cloud.

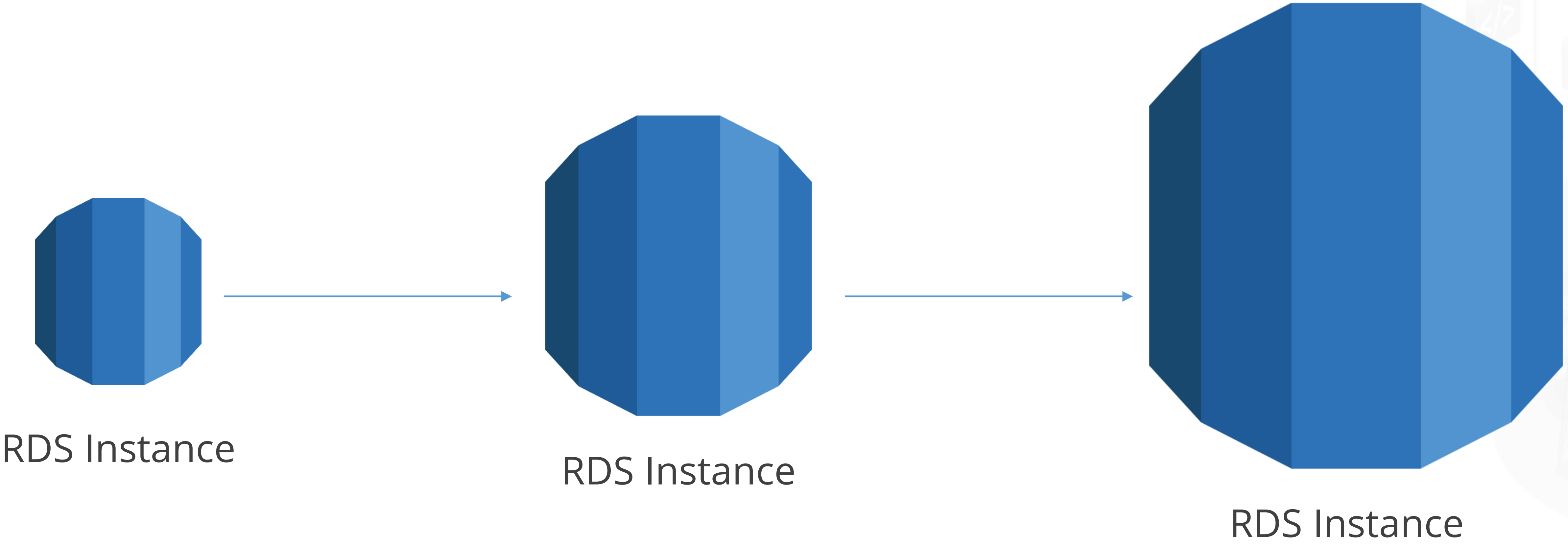
Here are some of the benefits of Amazon RDS:

- Easy to administer
- Highly scalable
- Available and durable
- Secure
- Cost-effective
- Fast performance

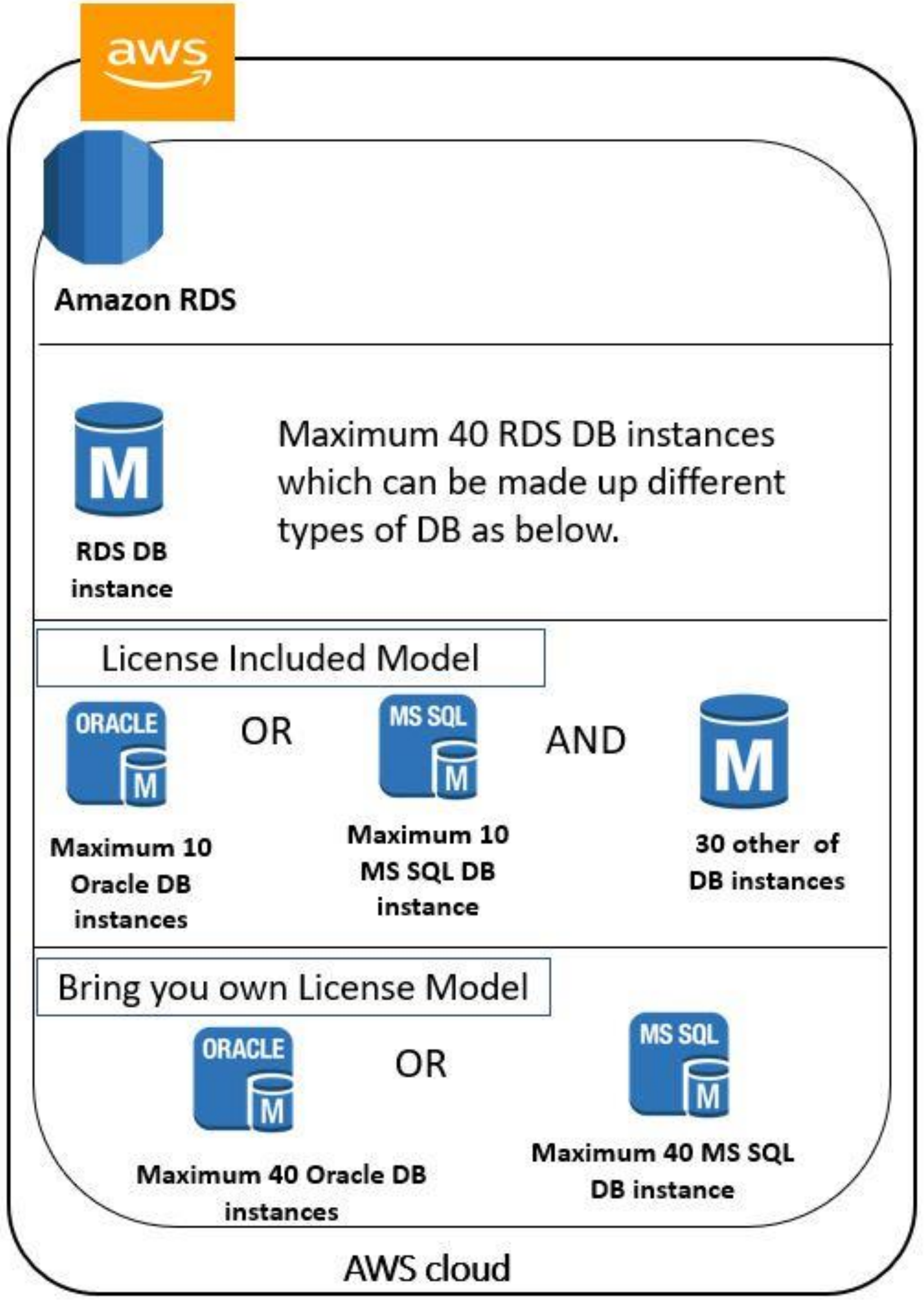
Scalability

The users can scale the compute and memory resources powering their deployment up or down, up to a maximum of 32 vCPUs and 244 GiB of RAM.

The compute scaling operations are usually completed in a few minutes.



DB Instance



Amazon RDS

RDS DB instance

Maximum 40 RDS DB instances which can be made up different types of DB as below.

License Included Model

ORACLE M — Maximum 10 Oracle DB instances

OR

MS SQL M — Maximum 10 MS SQL DB instance

AND

M — 30 other of DB instances

Bring you own License Model

ORACLE M — Maximum 40 Oracle DB instances

OR

MS SQL M — Maximum 40 MS SQL DB instance

AWS cloud

Source: https://www.tutorialspoint.com/amazonrds/amazonrds_db_instances.htm

- A DB instance is a standalone database environment that runs in the cloud. It serves as Amazon RDS's fundamental building piece.
- It can contain multiple user-created databases and can be accessed using the same client tools and applications.

RDS classes

The DB instance class determines the computation and memory capacity of an Amazon RDS DB instance.

They are of three types:

- General purpose
- Memory optimized
- Burstable performance

Amazon RDS Database Engines

Some of the databases available in Amazon RDS are:

Amazon Aurora

PostgreSQL

ORACLE®

MySQL®

Microsoft® SQL Server®

MariaDB

Limitations of Amazon RDS for Microsoft SQL

The following server-level roles of Microsoft SQL are not currently available in Amazon RDS:

- bulkadmin
- dbcreator
- diskadmin
- securityadmin
- serveradmin
- sysadmin

Limitation of Amazon RDS for MySQL

The following features of MySQL are not currently available in Amazon RDS:

- Authentication plugin
- Error logging to the system log
- Group replication plugin
- InnoDB tablespace encryption
- Password strength plugin
- Persisted system variables
- Replication filters
- Semisynchronous replication
- Transportable tablespace
- X Plugin

Limitations of Amazon RDS for Oracle

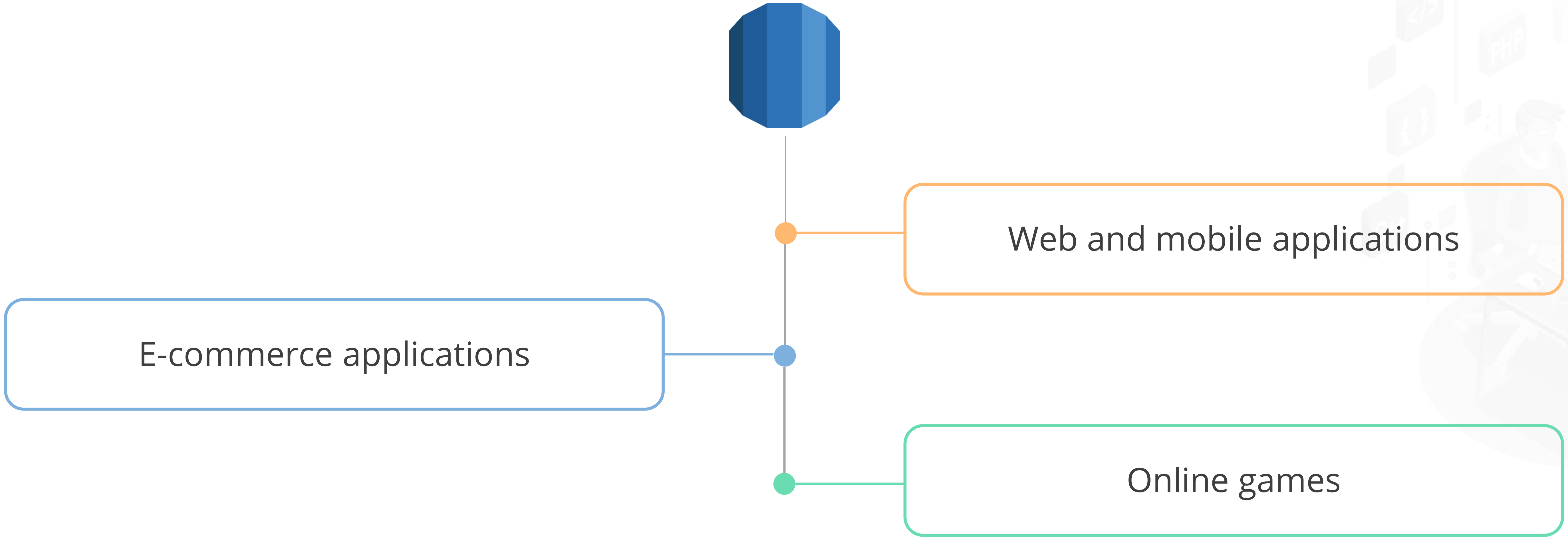
The following privileges of Oracle are not currently available in Amazon RDS:

ORACLE®

- Alter database
- Alter system
- Create any directory
- Drop any directory
- Grant any privilege
- Grant any role

Use Cases of Amazon RDS

Some of the use cases in Amazon RDS are as follows:



Web and mobile applications

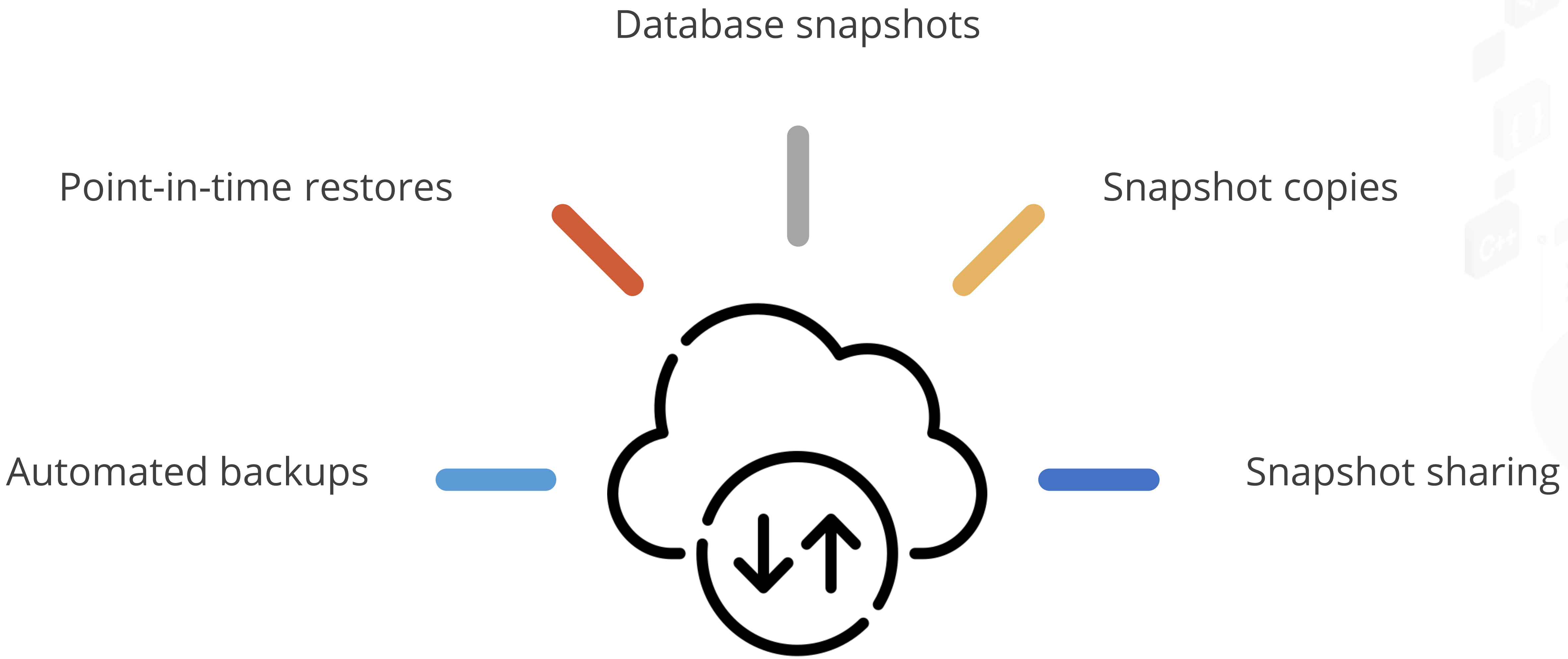
E-commerce applications

Online games

Amazon RDS Backups

By default, Amazon RDS creates and saves automated backups of DB instance securely in Amazon S3 for a user-specified retention period.

The types of Amazon RDS backups are:



Database snapshots

Point-in-time restores

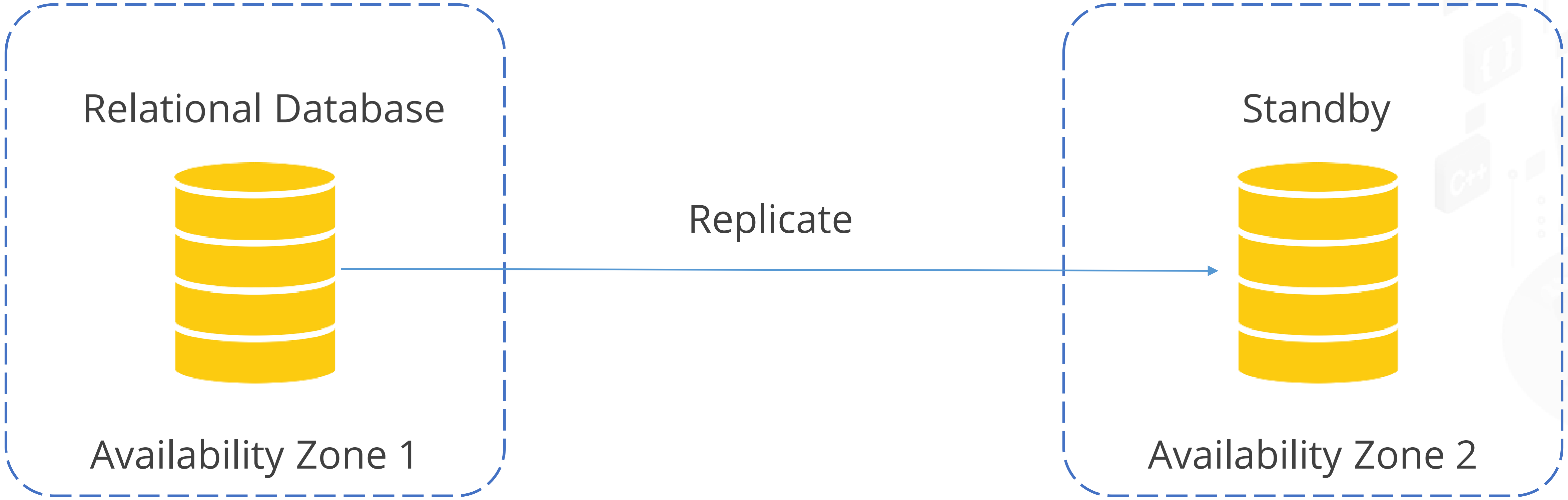
Snapshot copies

Automated backups

Snapshot sharing

Multi Availability Zone Deployments

Multi Availability Zone deployments synchronously replicate the data to a standby instance in a different Availability Zone.

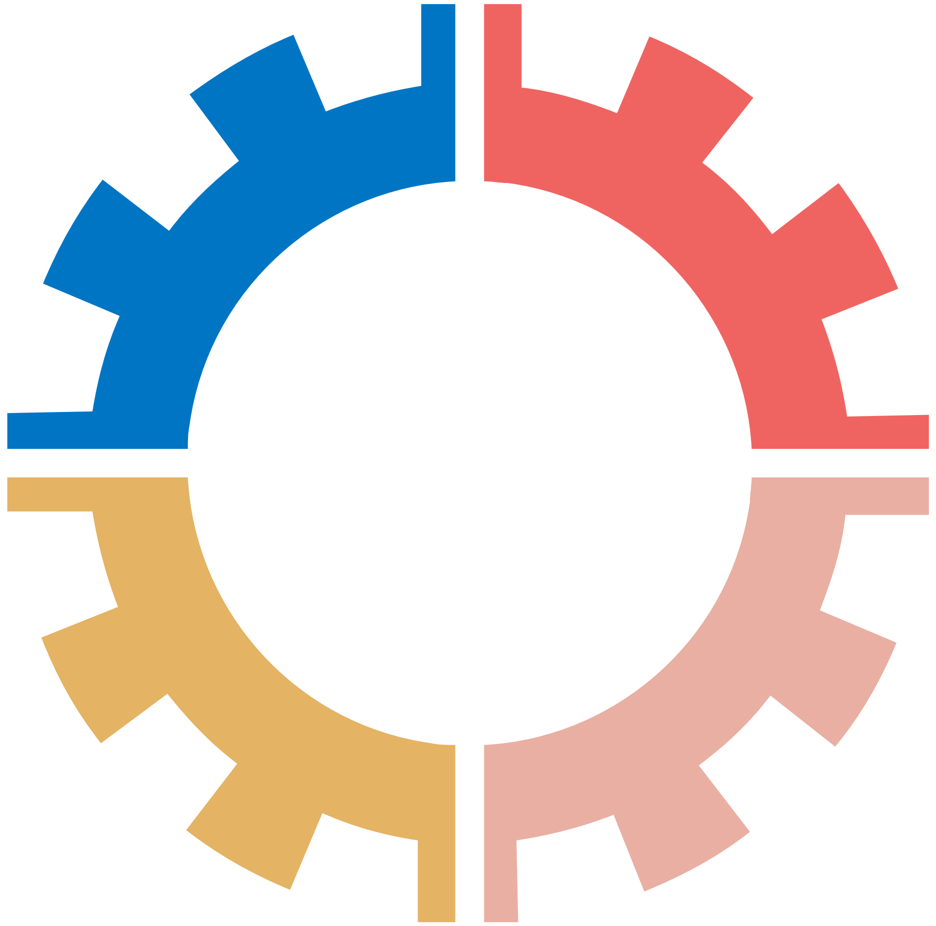


Multi Availability Zone Deployments

Here are some benefits of Multi Availability Zone:

Enhanced durability: Replica of the data will be up-to-date with the primary database

Increased availability: In case of Availability Zone failure, the backup will be ready less than 30 seconds

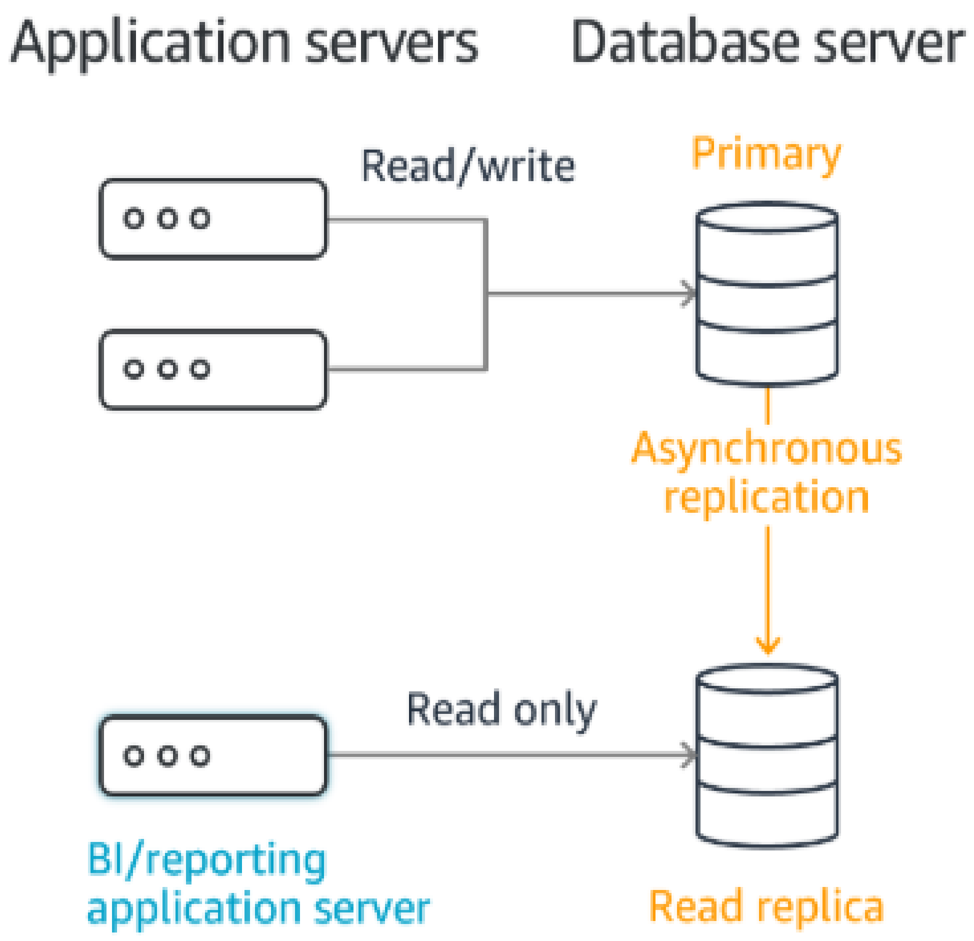


Stable Performance: I/O activity is not suspended on your primary during backup for Multi-AZ deployments

Automatic failover: Amazon RDS automatically initiates a failover to the up-to-date standby

Read Replicas

Amazon RDS Read Replicas provide enhanced performance and durability for RDS database (DB) instances.



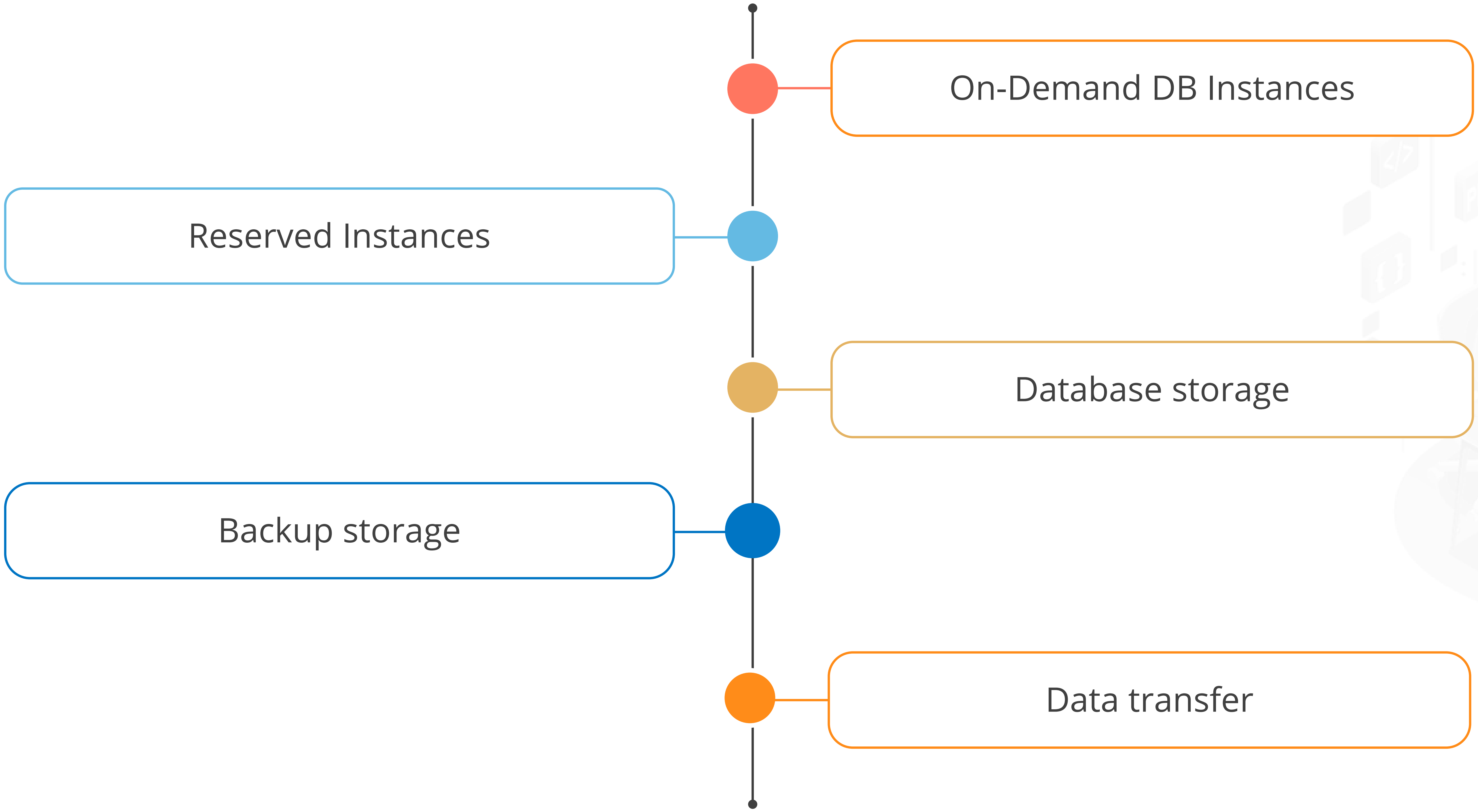
Here are some benefits of Read Replicas:

- Enhanced performance
- Increased availability
- Designed for security

Source: https://aws.amazon.com/rds/features/read-replicas/

Amazon RDS Costs

The following are the costs associated with Amazon RDS for Microsoft SQL Server:



On-Demand DB Instances

Reserved Instances

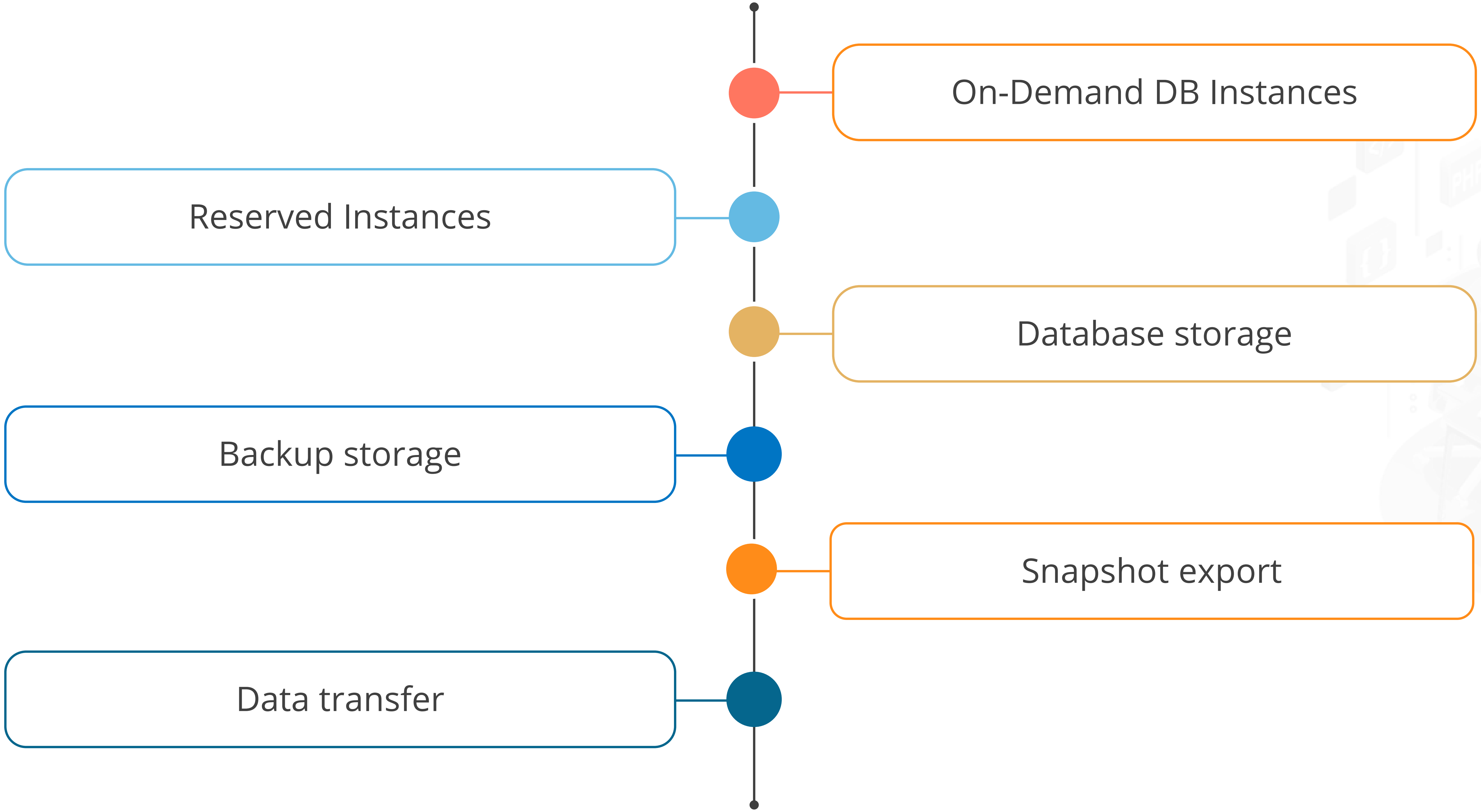
Database storage

Backup storage

Data transfer

Amazon RDS Costs

The following are the costs associated with Amazon RDS for MySQL and PostgreSQL:



On-Demand DB Instances

Reserved Instances

Database storage

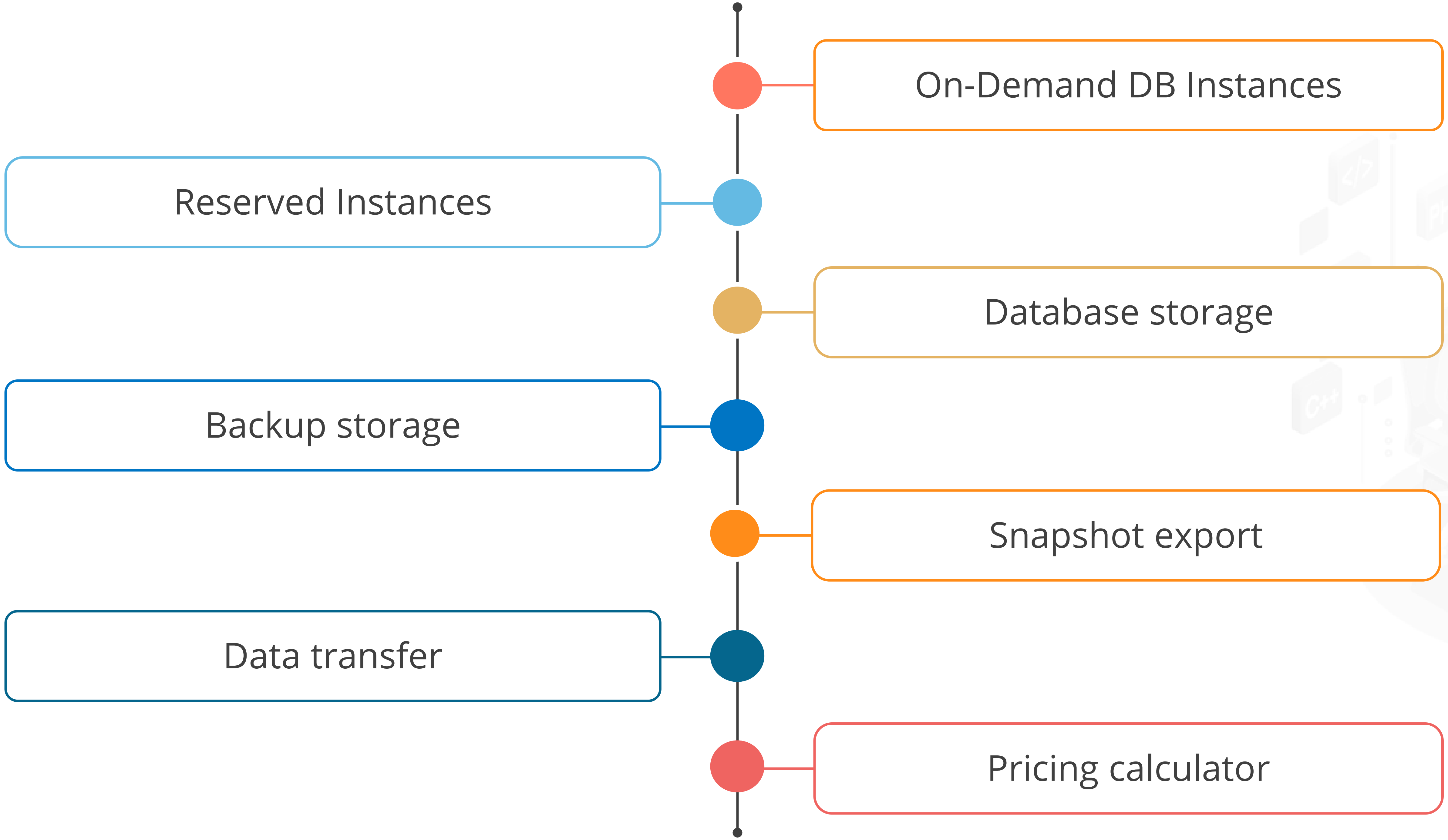
Backup storage

Snapshot export

Data transfer

Amazon RDS Costs

The following are the costs associated with Amazon RDS for MariaDB:



On-Demand DB Instances

Reserved Instances

Database storage

Backup storage

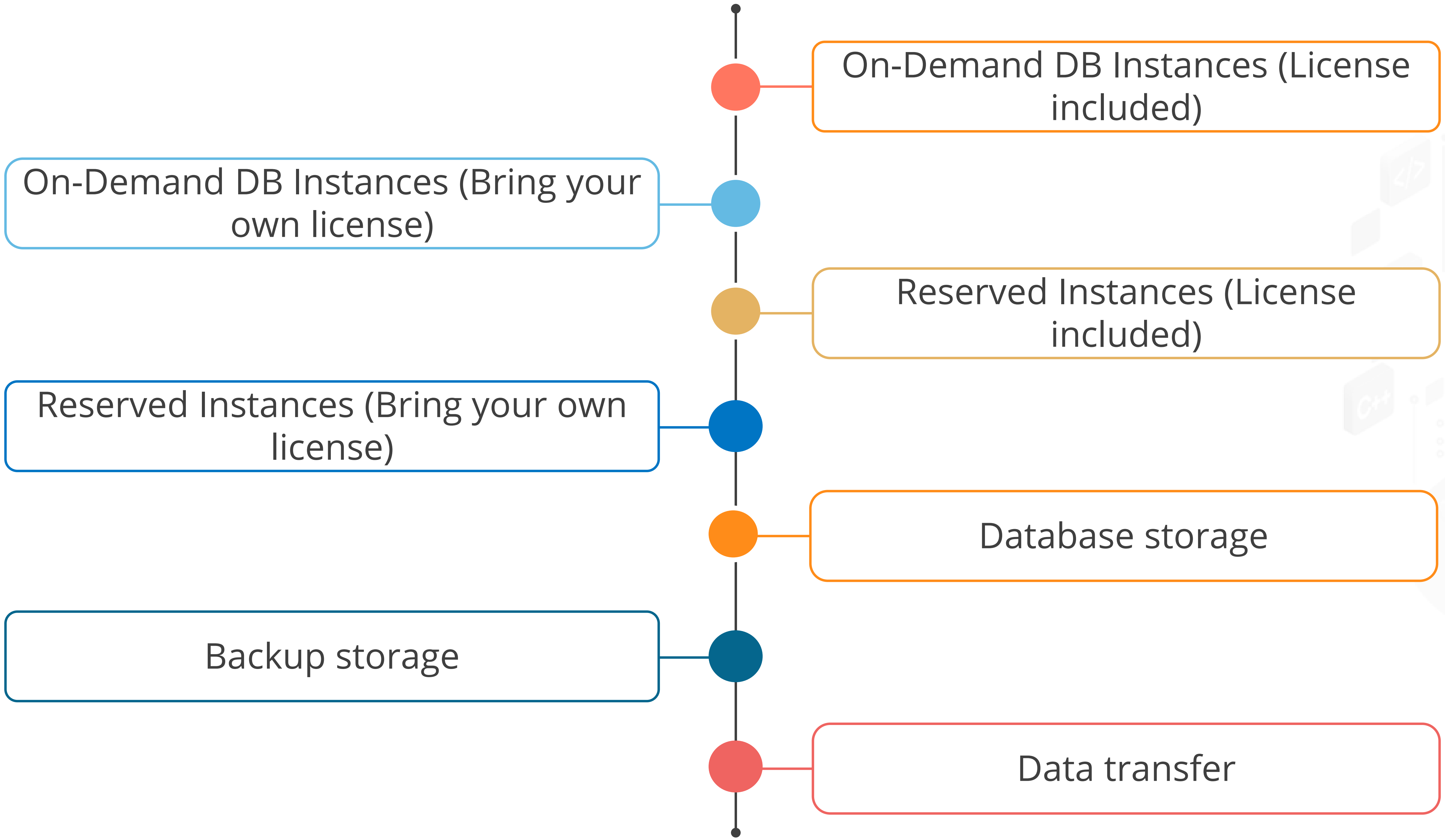
Snapshot export

Data transfer

Pricing calculator

Amazon RDS Costs

The following are the costs associated with Amazon RDS for Oracle:



On-Demand DB Instances (License included)

On-Demand DB Instances (Bring your own license)

Reserved Instances (License included)

Reserved Instances (Bring your own license)

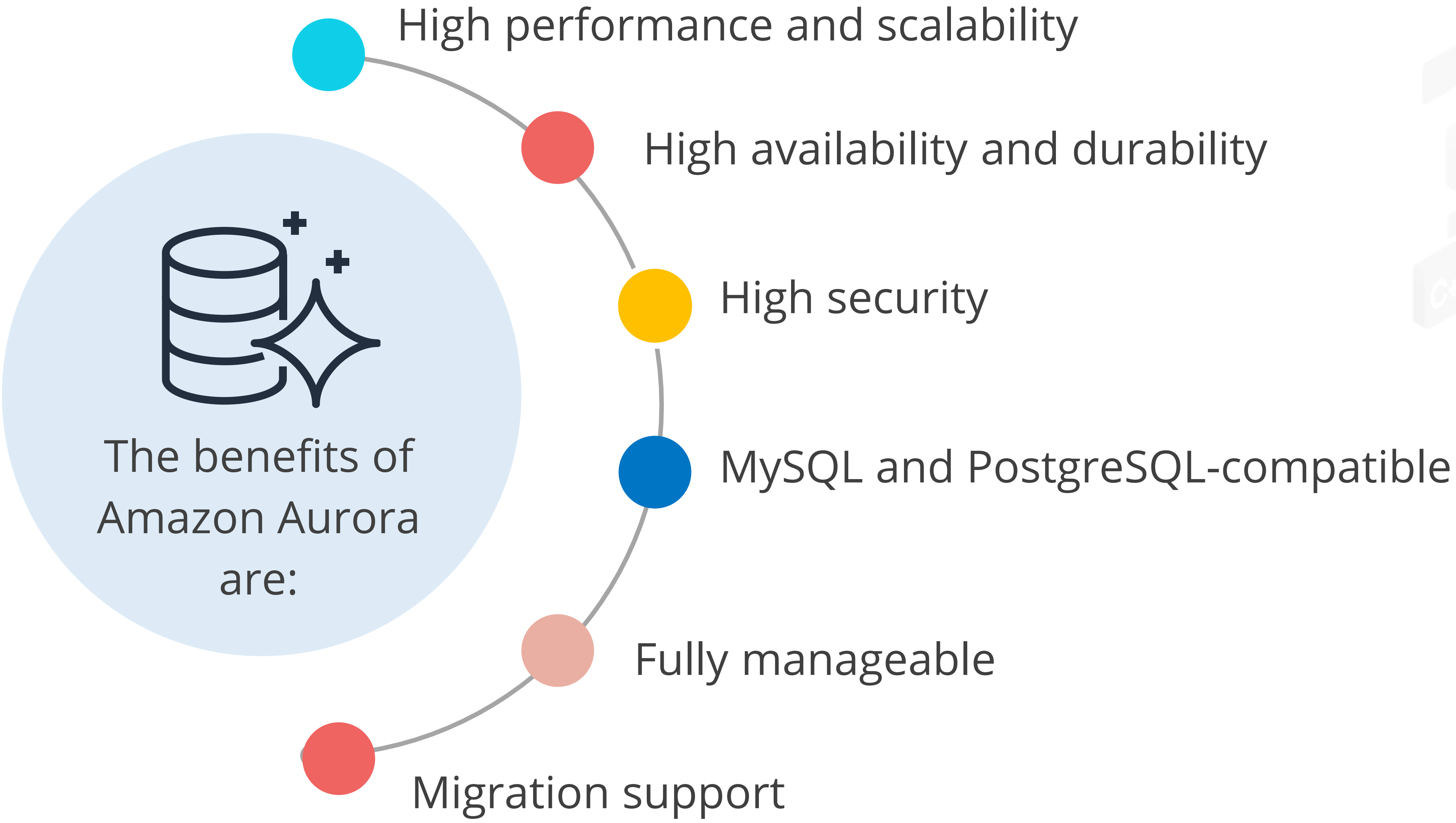
Database storage

Backup storage

Data transfer

Amazon Aurora

Amazon Aurora is a relational database that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases.



The benefits of Amazon Aurora are:

- High performance and scalability
- High availability and durability
- High security
- MySQL and PostgreSQL-compatible
- Fully manageable
- Migration support

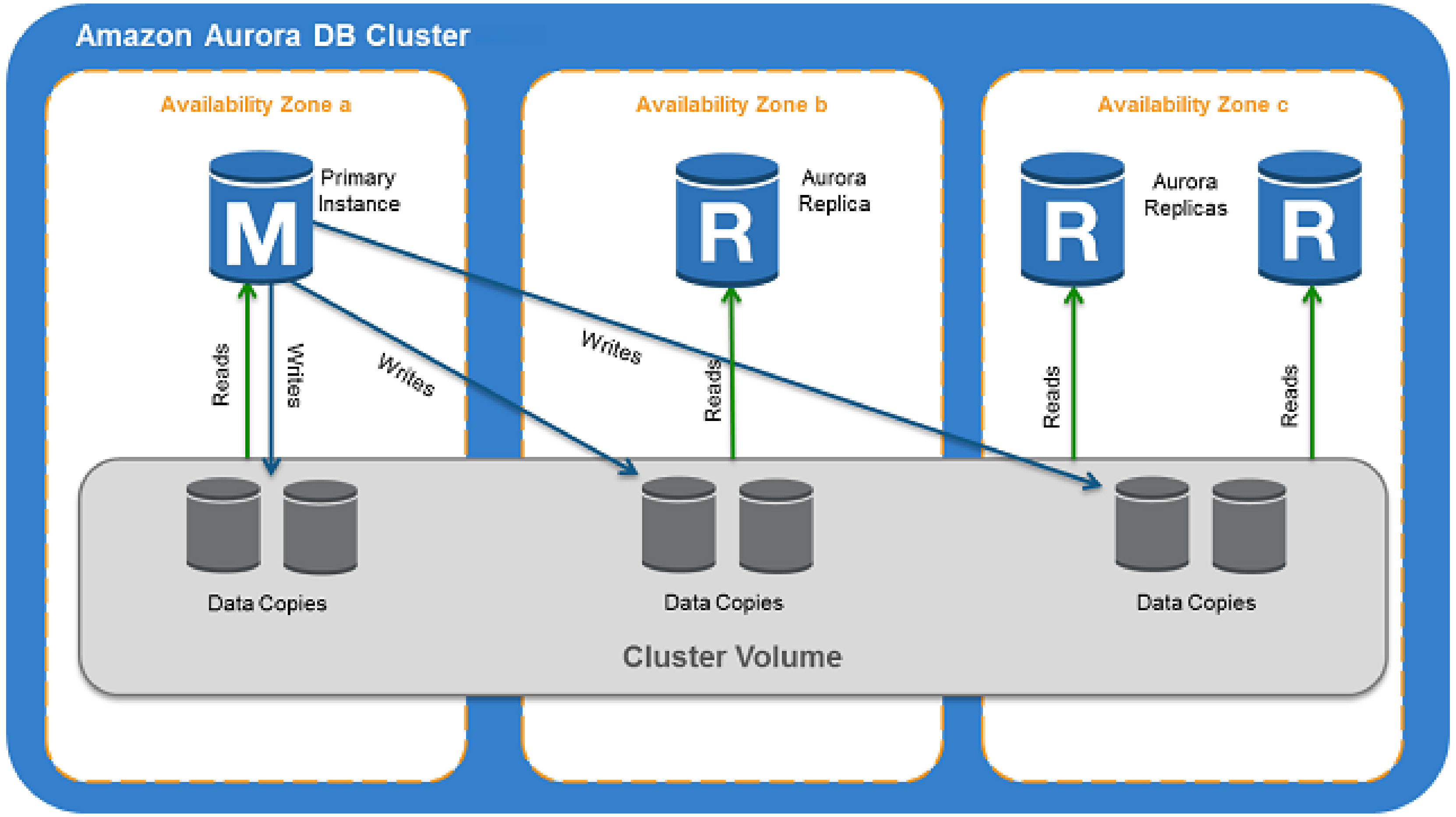
Amazon Aurora Relational Database

- Amazon Aurora is a MySQL and PostgreSQL-compatible relational database
- It combines the speed and availability of high-end commercial databases
- Cost-effectiveness is like that of open-source databases.

Amazon Aurora DB Cluster

An Amazon Aurora DB cluster consists of one or more DB Instances and a cluster volume that manages the data for those DB Instances.

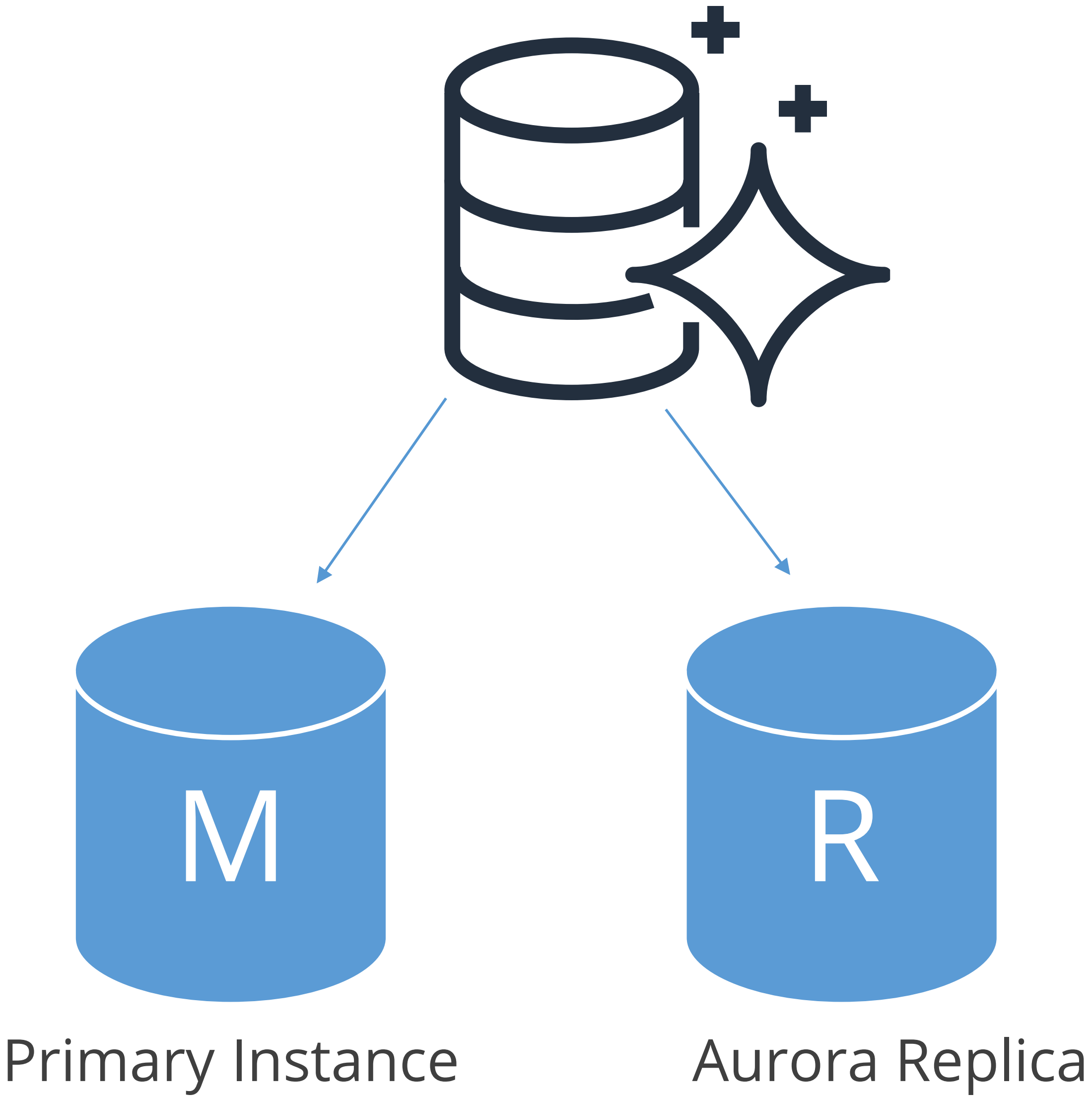
Relationship between a cluster volume, a primary DB Instance, and Aurora Replicas in an Aurora DB cluster:



Source: https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.html

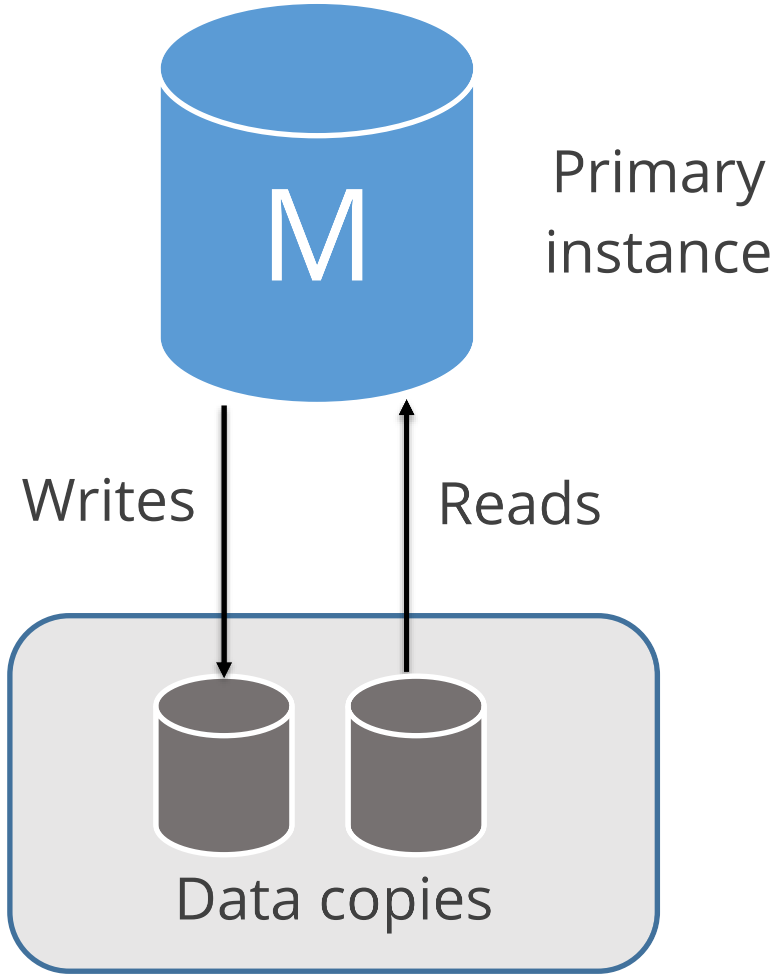
Amazon Aurora DB Cluster

There are two types of Aurora DB Cluster as follows:



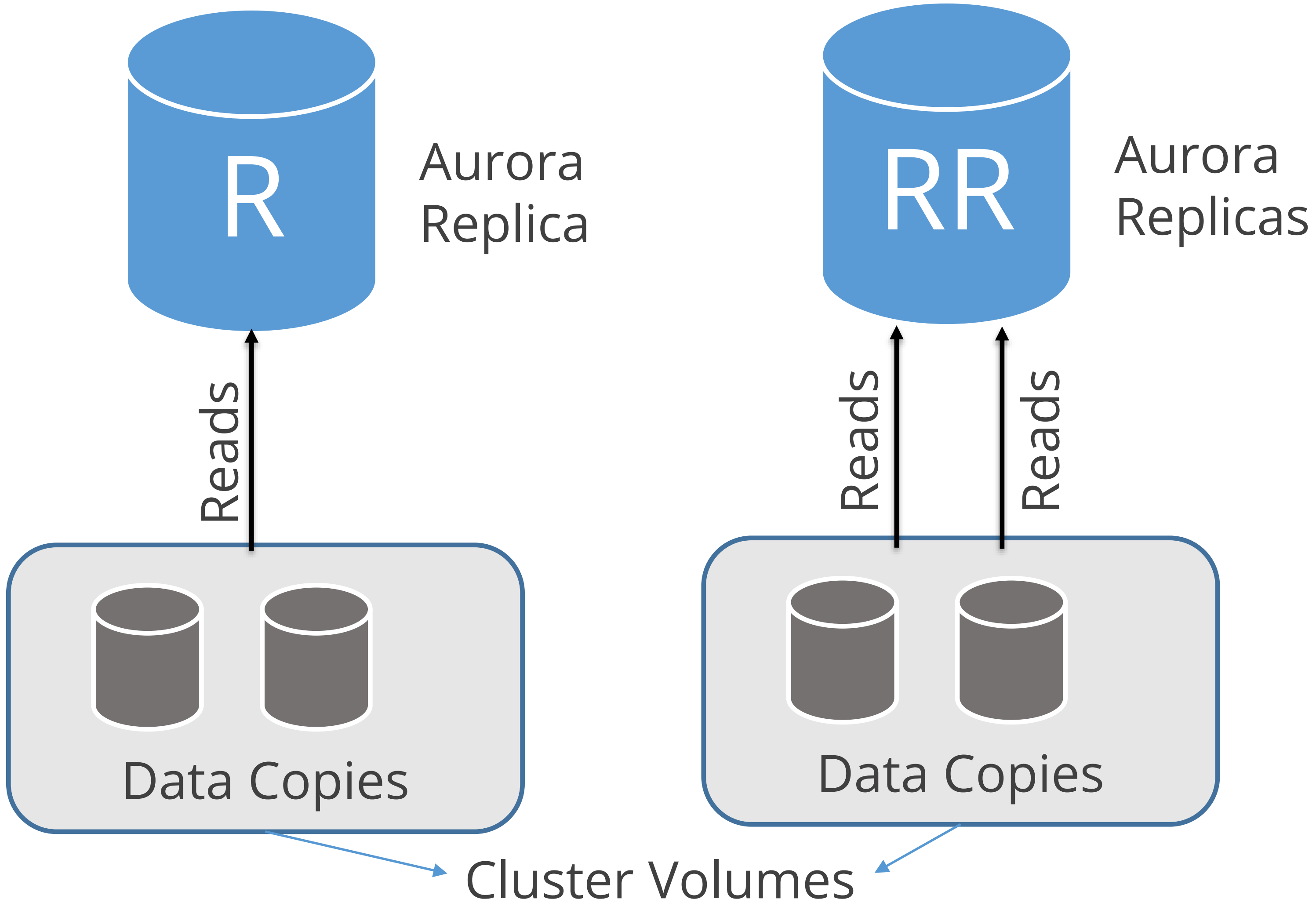
Amazon Aurora DB Cluster

The **primary instance** supports read and write operations and updates the cluster volume's data in all its data alterations. One primary DB instance exists in each Aurora DB cluster.



Amazon Aurora DB Cluster

Aurora replica supports only read operations and connects to the same storage disc as the primary DB instance.



In addition to the primary DB instance, each cluster of an Aurora DB can contain up to 15 Aurora Replicas.

Amazon Aurora Serverless

Amazon Aurora Serverless is an auto-scaling configuration, where the database automatically starts up, shuts down, and scales capacity up or down based on the application's needs.

There are two types of Amazon Aurora serverless:

- Amazon Aurora Serverless v1
- Amazon Aurora Serverless v2

Amazon Aurora Serverless v1

Aurora Serverless v1 offers a simple, cost-effective choice for infrequent, inconsistent, or unpredictable workloads.

- The computational capacity of an Aurora Serverless v1 DB cluster can be scaled up or down in response to the demands of the application.
- DB clusters created by Aurora are handled manually in terms of capacity.

Amazon Aurora Serverless v1

The use cases of Amazon Aurora Serverless v1 are as follows:

Infrequently-used applications

Development and test databases

Amazon Aurora Serverless v1

Here are some benefits of Aurora Serverless v1:

Amazon Aurora Serverless v2

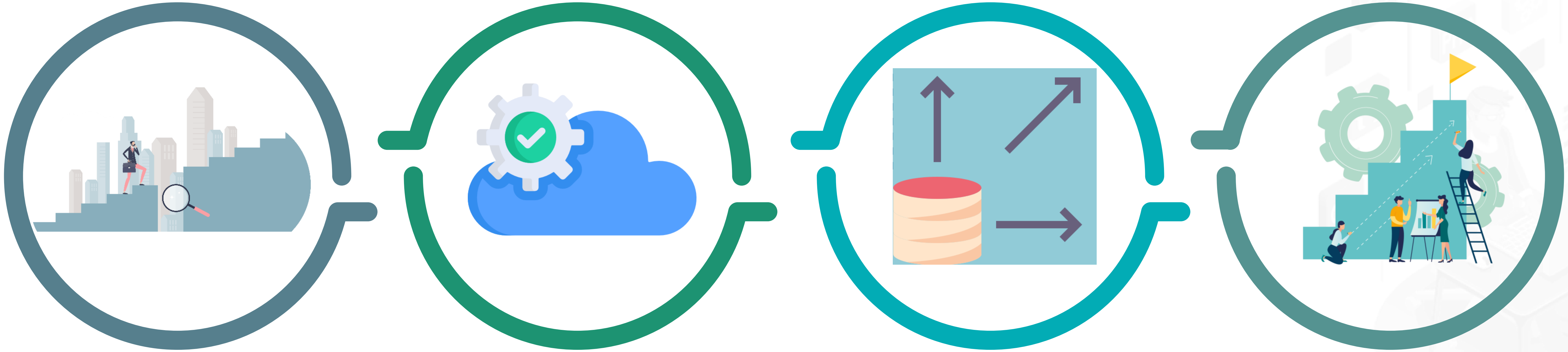
Aurora Serverless v2 helps the users to automate the procedures for workload monitoring and database capacity adjustment.

Here are some benefits of Aurora Serverless v2:

- Highly scalable
- Simple
- Highly Available
- Transparent
- Cost-effective
- Durable

Amazon Aurora Serverless v2

The use cases for Amazon Aurora Serverless v2 are as follows:



Enterprise database fleet management

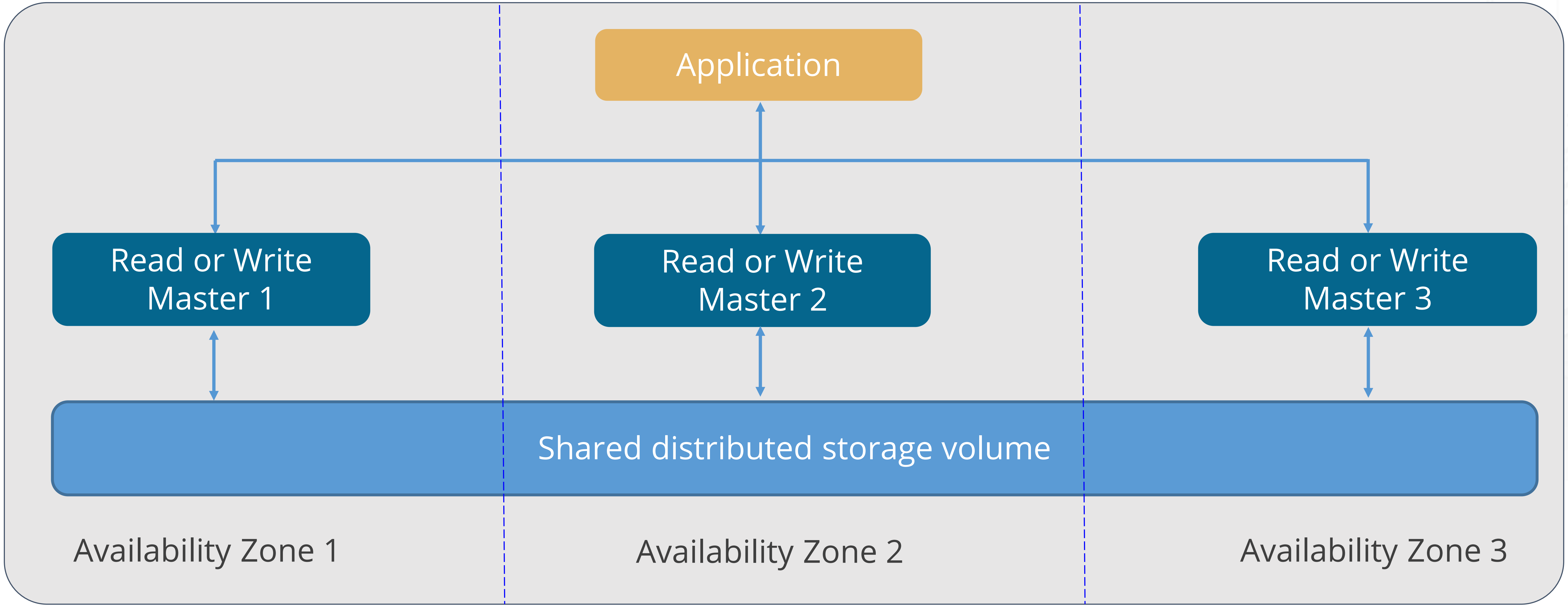
Software as a service applications

Scaled-out databases split across multiple servers

Unpredictable workloads

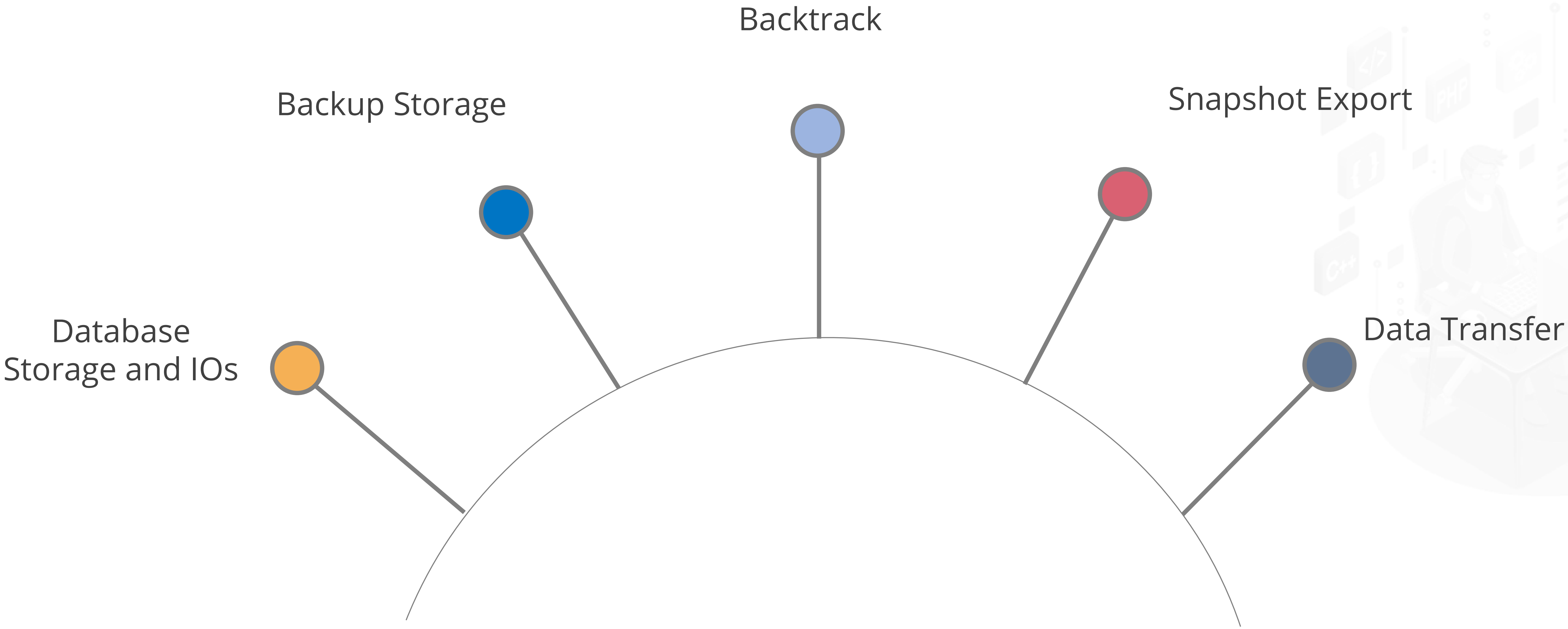
Aurora Multi-Master

Aurora Multi-Master allows the users to create multiple read or write master instances across multiple Availability Zones.



Amazon Aurora Costs

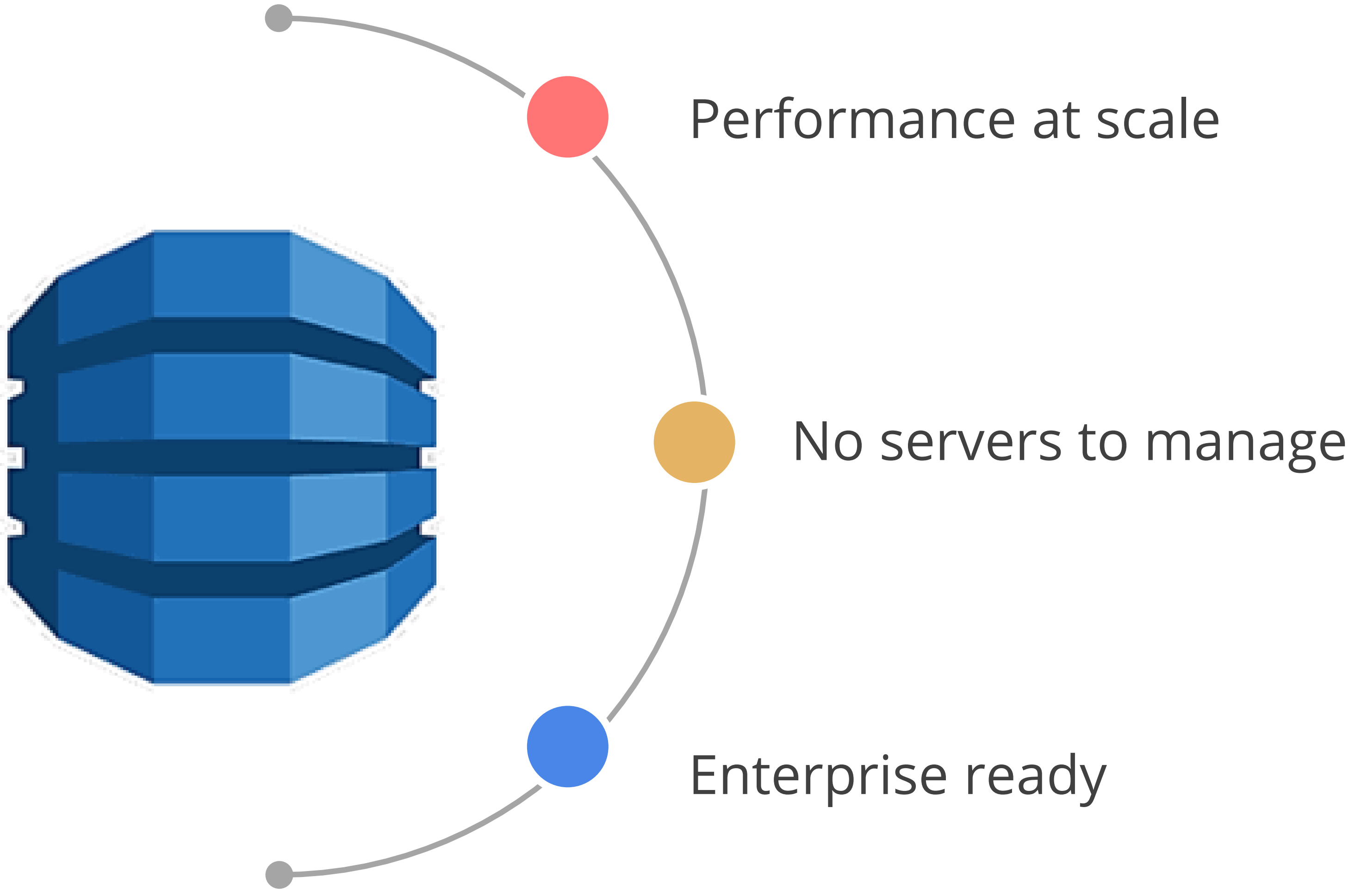
The following are the costs associated with Amazon Aurora:



DynamoDB

Amazon DynamoDB

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. The benefits of Amazon DynamoDB are as follows:



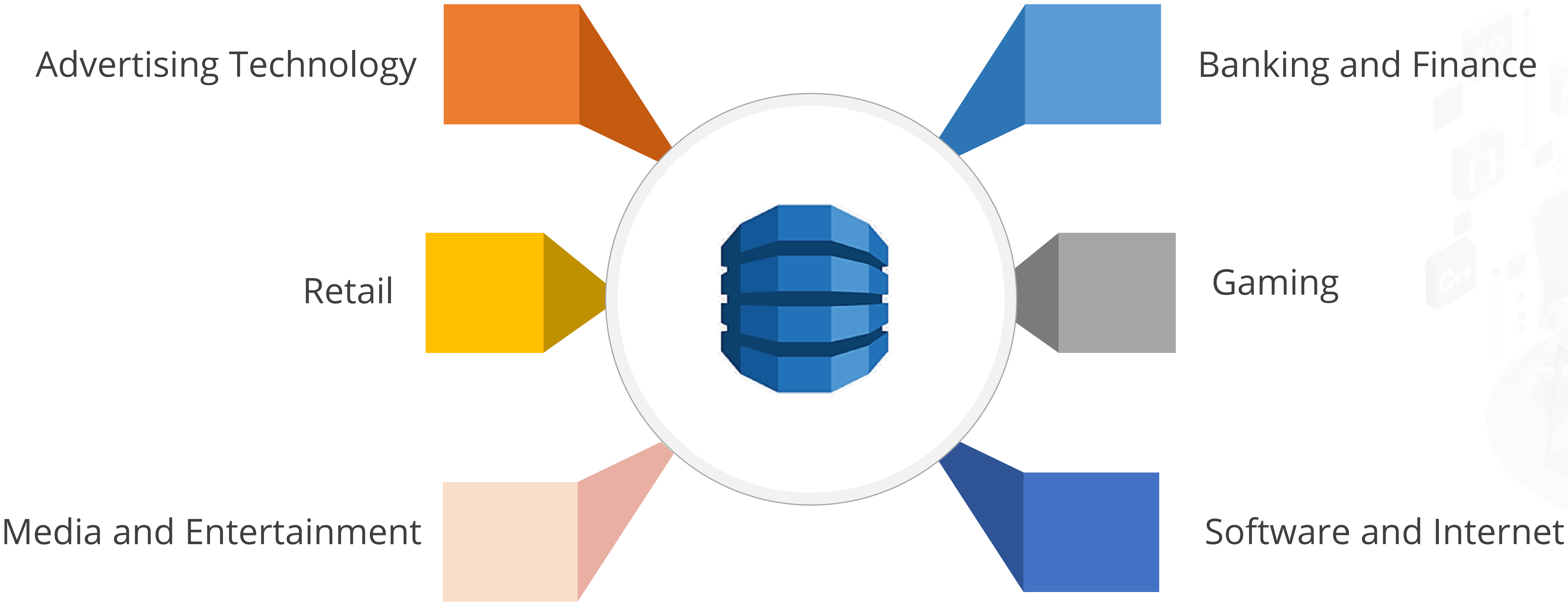
Performance at scale

No servers to manage

Enterprise ready

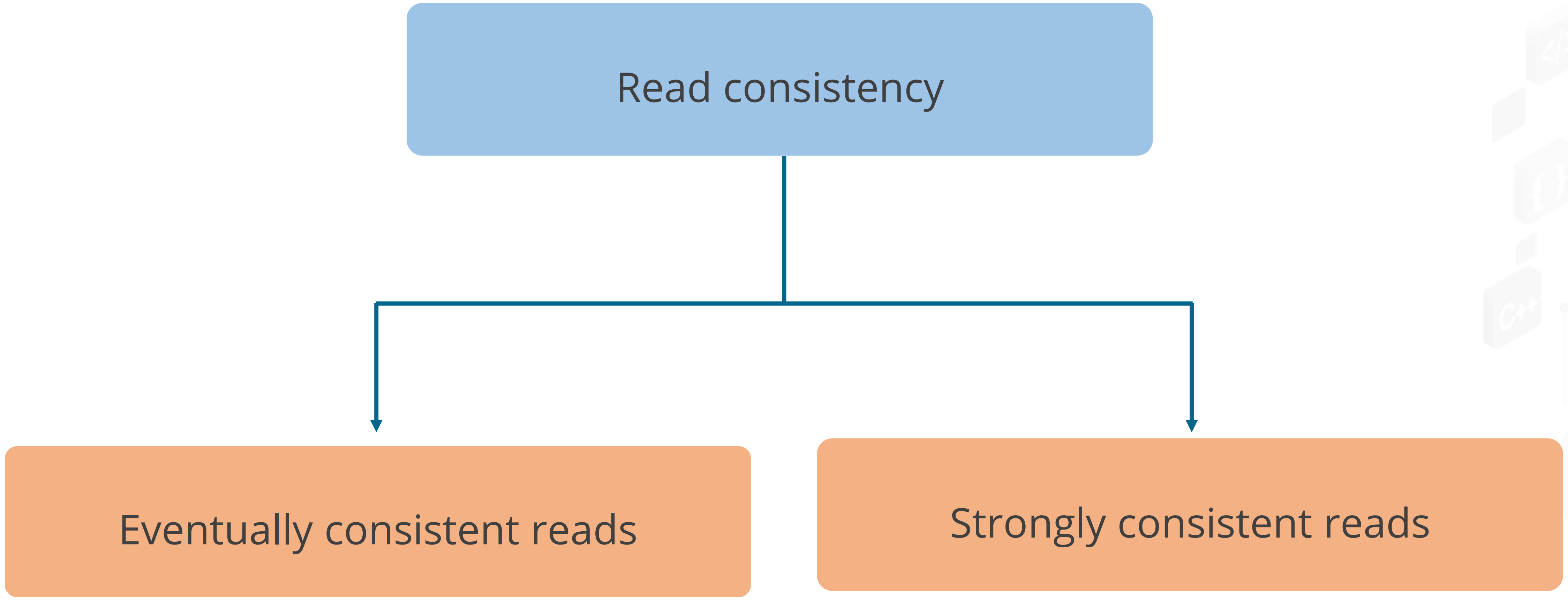
Amazon DynamoDB

The use cases of Amazon DynamoDB are as follows



Amazon DynamoDB: Read Consistency

Amazon DynamoDB offers two types of read consistency:



Read consistency

Eventually consistent reads

Strongly consistent reads

Amazon DynamoDB Global Tables

Global Dispersed users

Global App

Replica (Europe)

Replica (Asia)

- Global tables can provide a fully managed, multi-region, and multi-master database
- Amazon delivers fast, local, read, and write performance.
- Global tables replicate the DynamoDB tables automatically across the users’ choice of AWS Regions.

Amazon DynamoDB

To read an item from a table in DynamoDB use the **get-item** command.
A simple example to show the read operation:

```
aws dynamodb get-item \
   --table-name Catalog \
  --key '{"Id":{"N":"1"}}'
```

Amazon DynamoDB

To write an item in a table in DynamoDB there are three commands:

PutItem

DeleteItem

UpdateItem

This function produces a brand-new item. The new item replaces any existing items with the same key in the table

This function deletes the item associated with the key is deleted using the **deleteItem** command

This function adds a new item to the list if the key supplied does not already exist. It alters the properties of an existing object if not

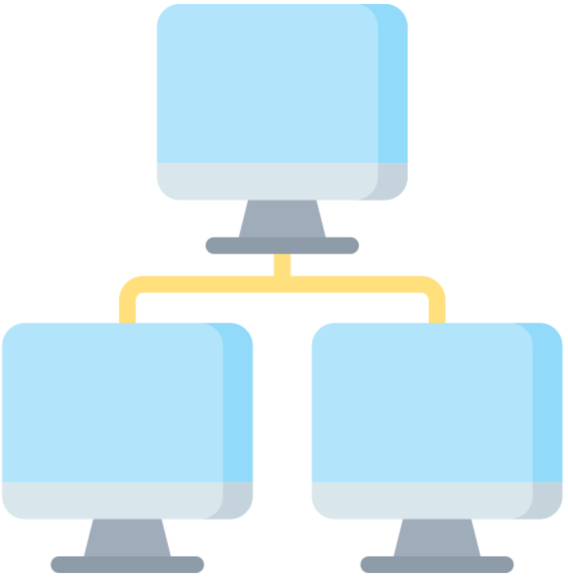
Amazon DynamoDB

A global secondary index is one that has a partition key and a sort key that are distinct from the main key of the base database.

Queries on the index can access the data from many base table partitions, it is considered as being global.

Amazon DynamoDB

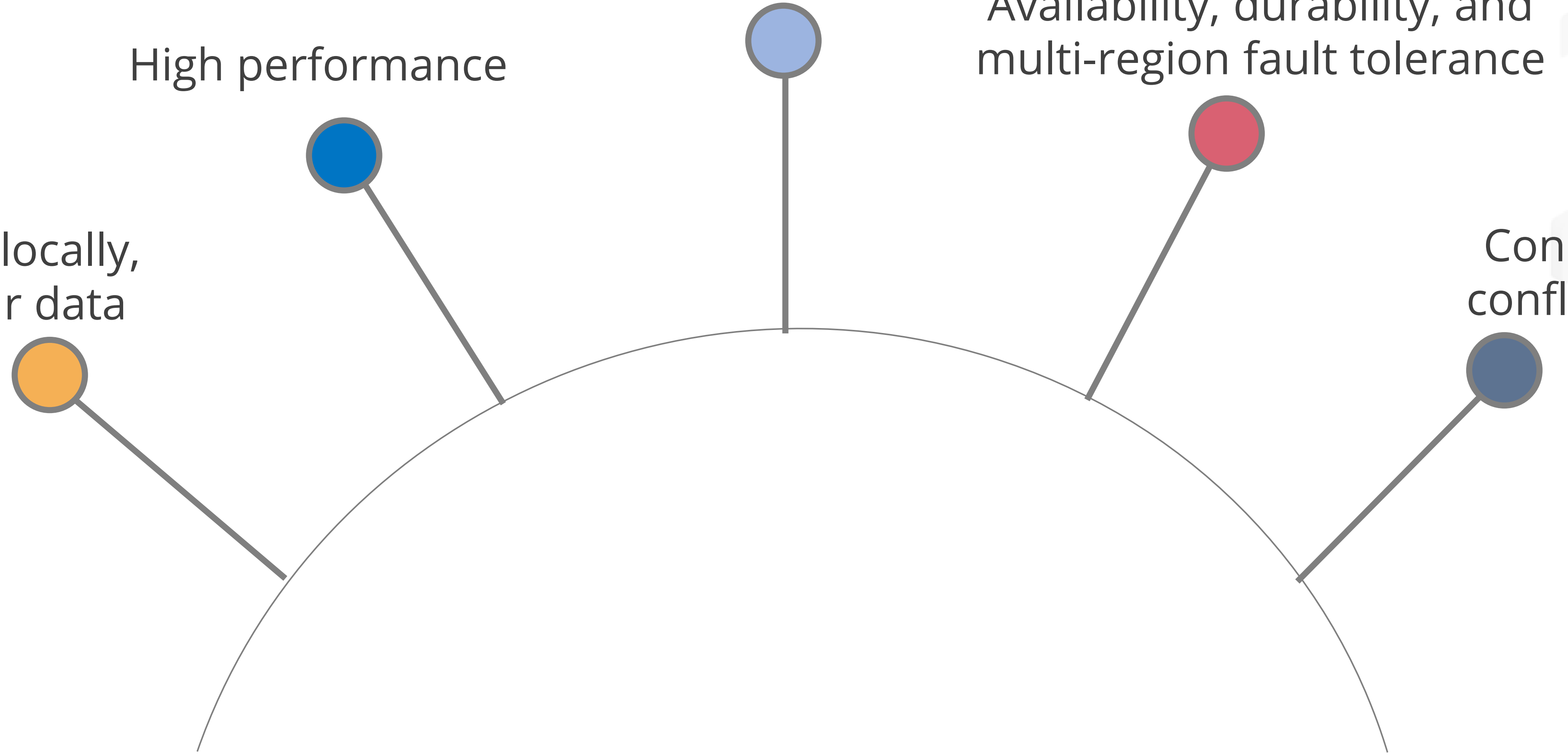
Local Secondary Index: This index requires the same partition key as the main table but a different sort of key.



Each partition of a local secondary index is constrained by the same partition key value of the base table, it is referred to as being local.

DynamoDB Global Tables

The benefits of Amazon DynamoDB global tables are as follows:



Read and write locally, and access your data globally

High performance

Easy to set up and operate

Availability, durability, and multi-region fault tolerance

Consistency and conflict resolution

Case Study: Duolingo

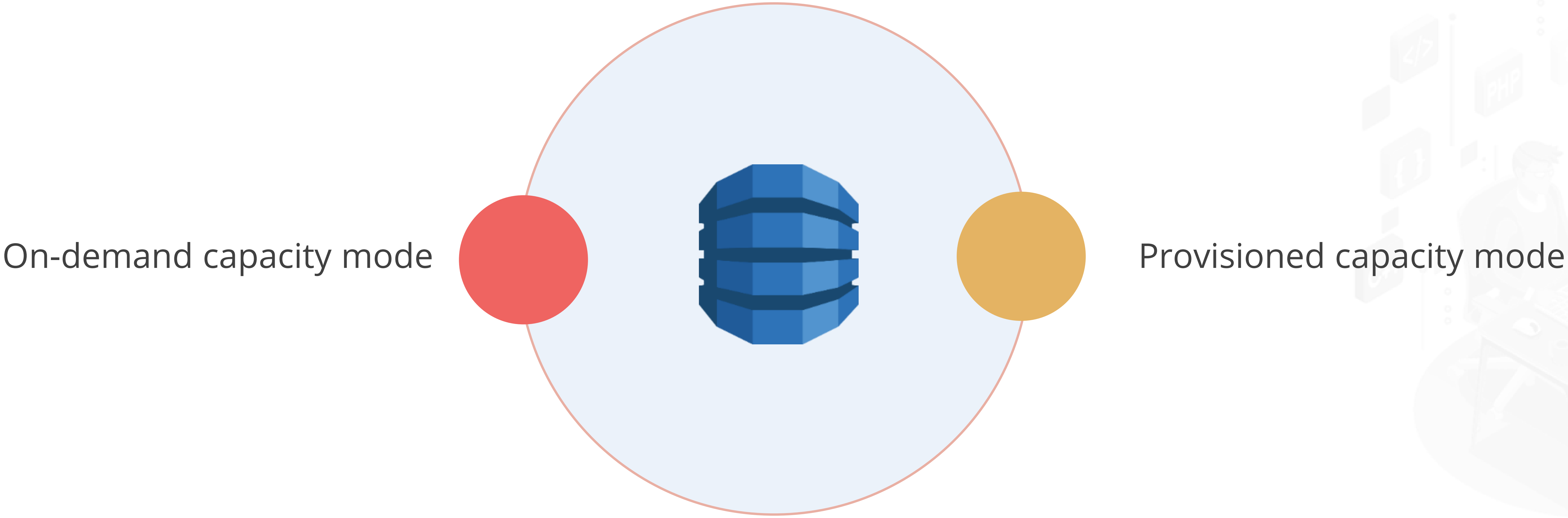
Duolingo uses Amazon DynamoDB to store 31 billion items for its online learning site that delivers lessons in 80 languages.

duolingo
Amazon DynamoDB

- 31 billion items
- 80 different languages
- 24,000 read units per second and 3300 write units per second

Amazon DynamoDB Costs

The following are the costs associated with Amazon DynamoDB:



On-demand capacity mode

Provisioned capacity mode

Amazon DynamoDB Features

Strong consistency

- DynamoDB supports eventually consistent and strongly consistent read
- Reading data from a DynamoDB table, may not reflect the results of a recently completed write operation
- Maximizes the read throughput.

Amazon DynamoDB Features

Capacity Management

- DynamoDB manages throughput capacity in read and write operations.
- It can scale up and down to cope with variable read/write demand.
- The developer can choose which capacity mode fits better with the application's needs.

Amazon DynamoDB Features

Best practices

- Using burst capacity effectively
- Understanding DynamoDB adaptive capacity
- Designing partition keys to distribute your workload evenly
- Sharding using random suffixes
- Using sort keys for version control.

Amazon DynamoDB Features

A few of the best practices for designing and using partition keys effectively are:

- Using burst capacity effectively
- Understanding DynamoDB adaptive capacity
- Designing partition keys to distribute your workload evenly
- Distributing write activity efficiently during data upload.

Amazon DynamoDB Features

A few of the best practices for querying and scanning data are:

- Performance considerations for scans
- Avoiding sudden spikes in read activity
- Taking advantage of parallel scans
- Choosing total segments.

Amazon Keyspaces for Apache Cassandra

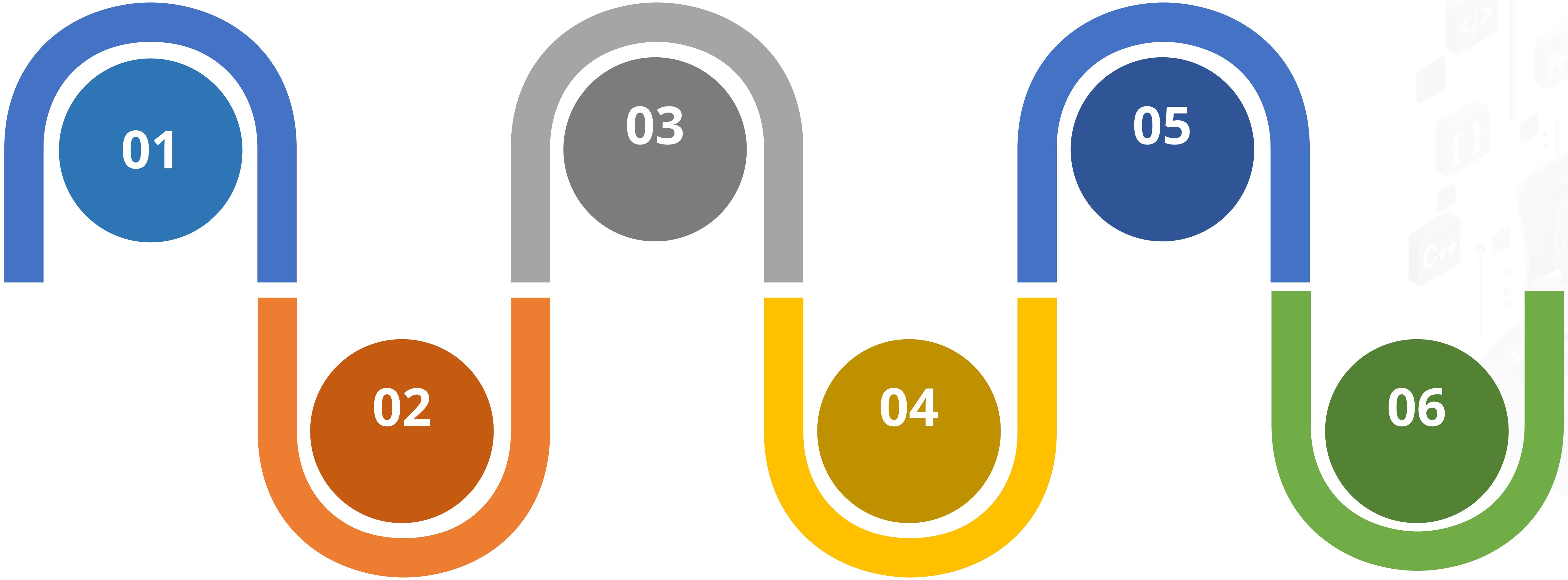
Cassandra workloads can be run on AWS using the same Cassandra application code and developer tools.

Apache Cassandra is scalable and highly available.

Apache Cassandra–compatible database service, now available in preview in 18 AWS regions.

Amazon Keyspaces for Apache Cassandra

Here are some benefits of Apache Cassandra:



High performance and scalability

High availability

High security

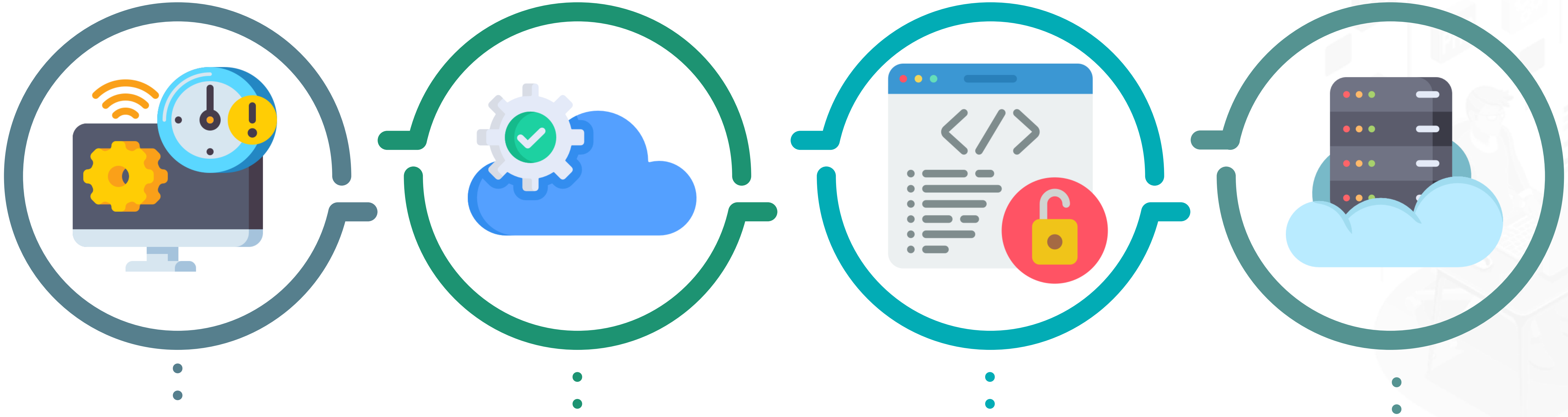
Compatible with Apache Cassandra

No servers to manage

Performance at scale

Amazon Keyspaces for Apache Cassandra

The use cases for Apache Cassandra are as follows:



Building low latency applications

Move your Cassandra workloads to the cloud

Build applications using open-source technologies

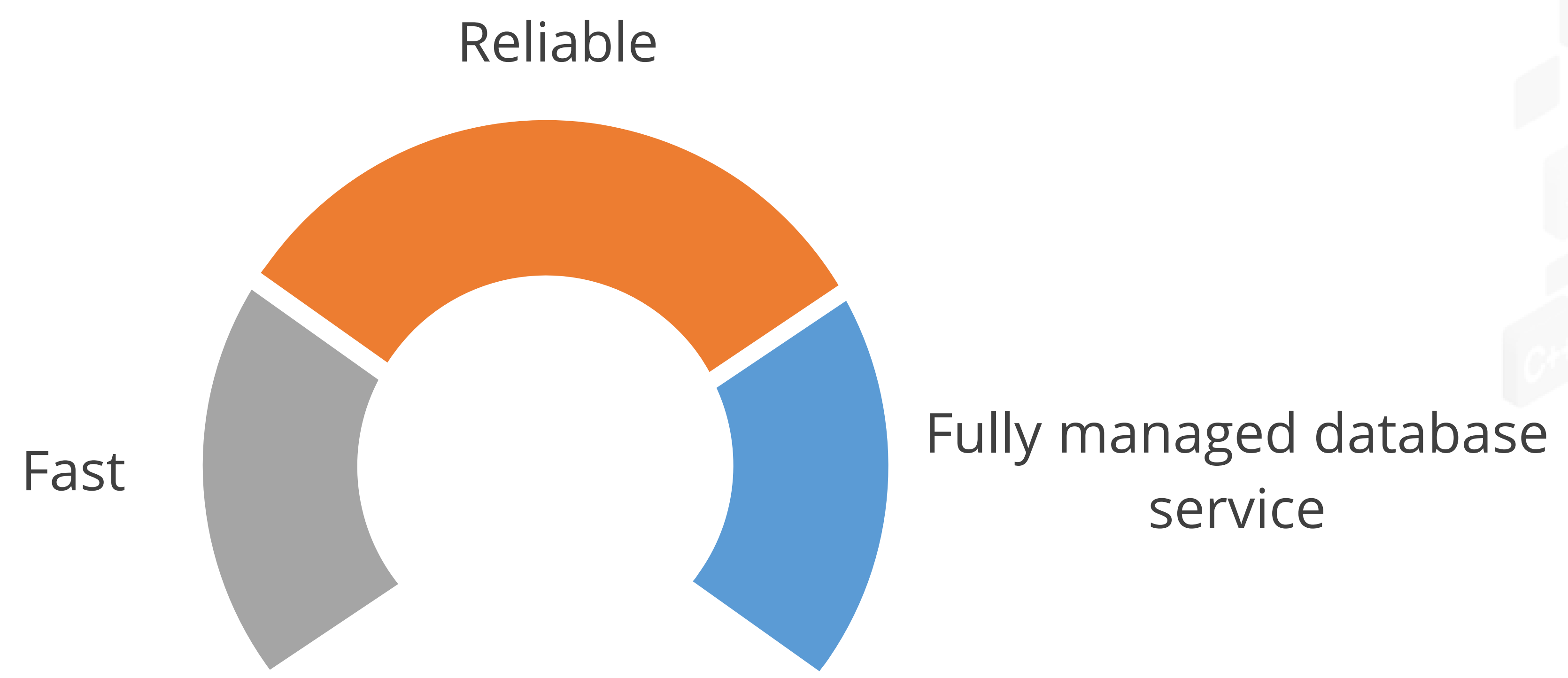
Data store for applications

Amazon DocumentDB

- The data is represented as a JSON document.
- It is more intuitive for developers to think of their data model as a document.
- The persist data in a database by using the same document model format that you use in your application code.
- It provides powerful and intuitive APIs for flexible and agile development.

Amazon DocumentDB

The benefits of Amazon DocumentDB (with MongoDB compatibility) are:



Amazon DocumentDB

Few use cases of Amazon DocumentDB are:

User Profiles

Real Time Big Data

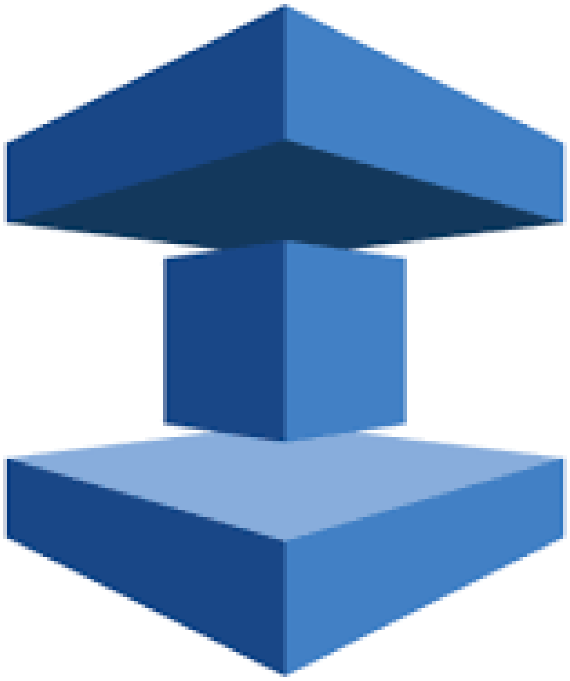
Content Management

TECHNOLOGY

Amazon ElastiCache

Amazon ElastiCache

Amazon ElastiCache is a web service that helps in deploying, operating, and scaling an in-memory data store or cache in the cloud.

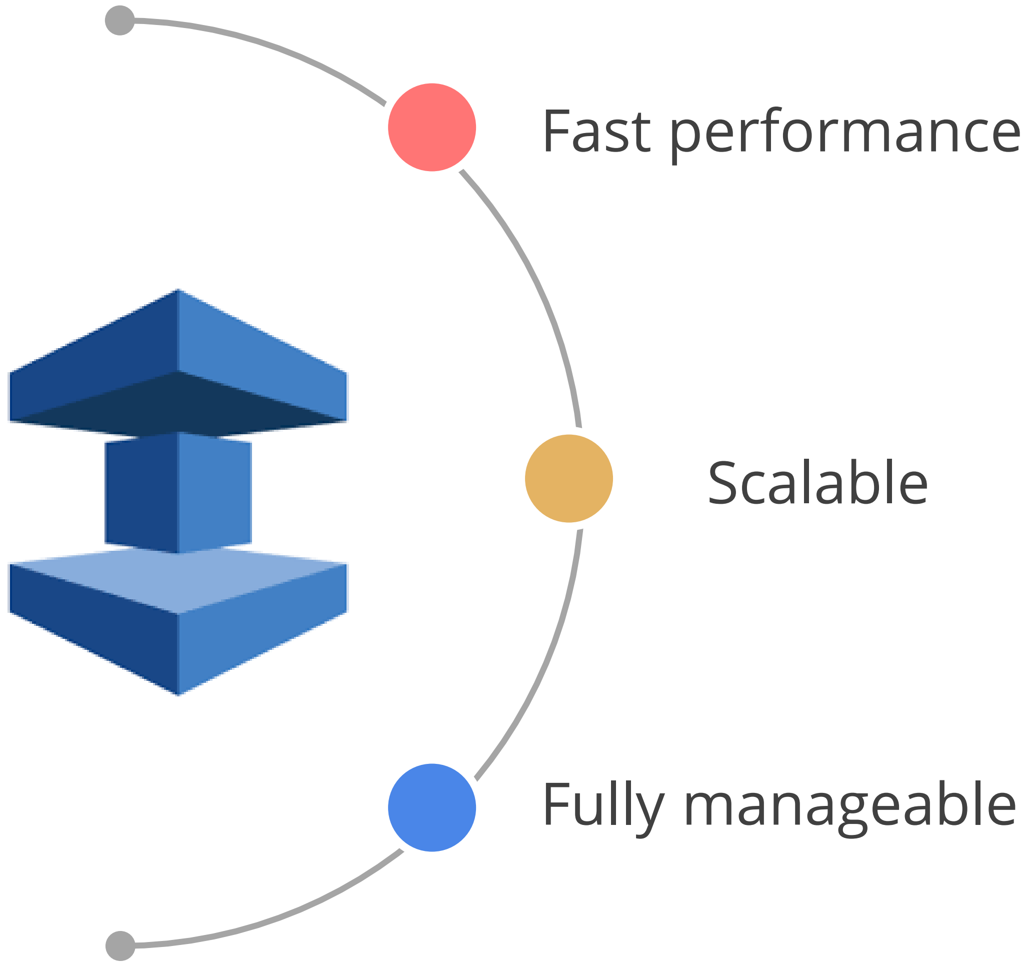


Amazon ElastiCache provides:

- Ease of management via AWS management console
- Compatibility with the specific engine protocol
- Detailed monitoring statistics for the engine nodes at no extra cost via Amazon CloudWatch
- Pay-per-use model for resource consumption.

Amazon ElastiCache

Here are some benefits of Amazon ElastiCache:



Fast performance

Scalable

Fully manageable

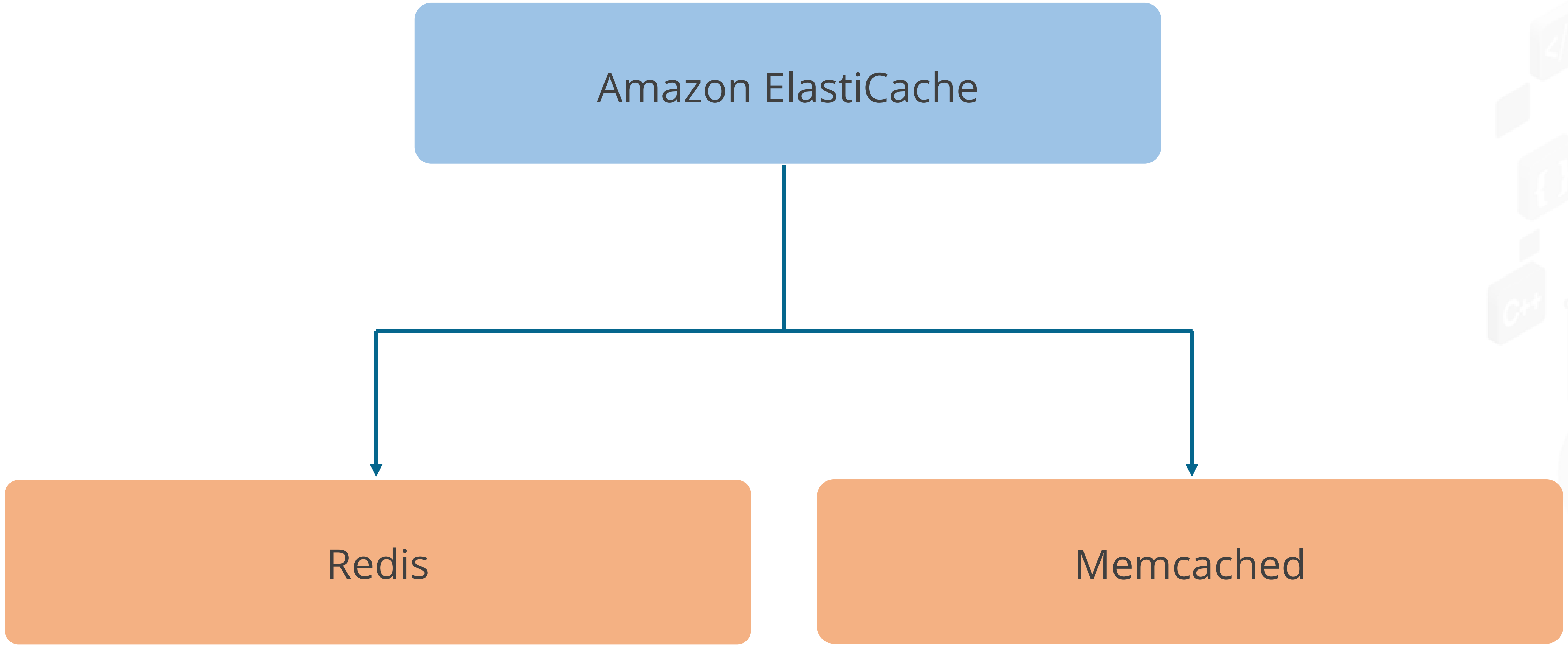
Amazon ElastiCache

Some of the use cases of Amazon ElastiCache:

- Caching
- Geospatial
- Media Streaming

Amazon ElastiCache: Data Engines

Amazon ElastiCache supports two different types of data engines:



Amazon ElastiCache

Redis

Memcached

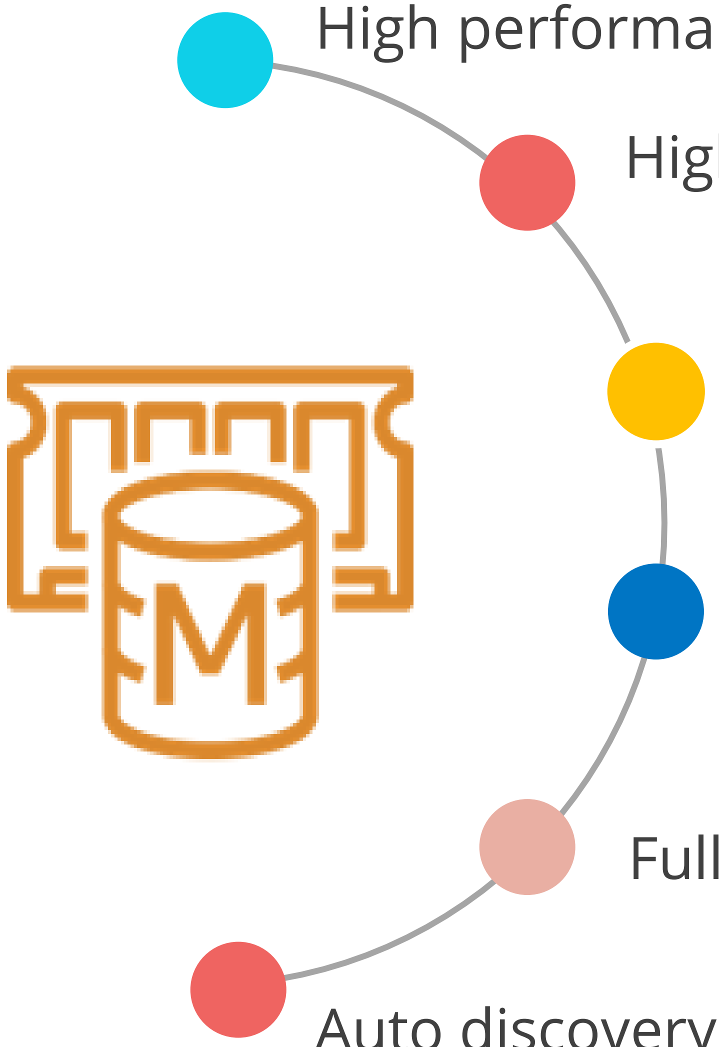
Amazon ElastiCache for Redis

Amazon ElastiCache is a Redis-compatible in-memory data store service and can support high performance since it is fully managed, scalable, and secure. The benefits are as follows:

- High performance
- Fully manageable
- Redis-compatible
- Highly available and reliable
- Easily scalable
- Secure and compliant

Amazon ElastiCache for Memcached

Amazon ElastiCache is a Memcached-compatible in-memory key-value store service that can be used as a cache or a data store. The benefits of Amazon ElastiCache for Memcached are as follows:

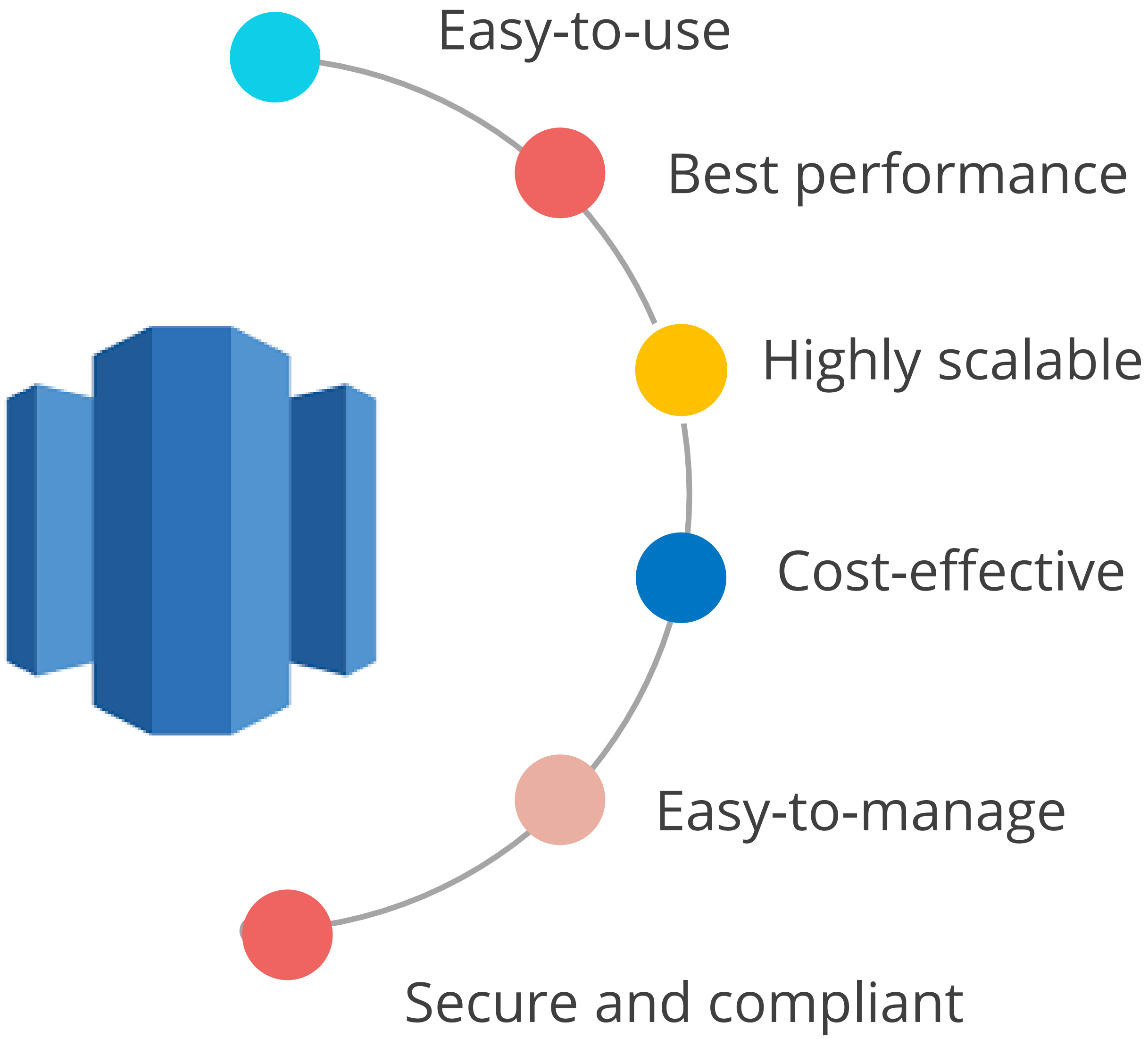


- High performance
- High security
- Memcached-compatible
- Easily scalable
- Fully manageable
- Auto discovery

Amazon Redshift

Amazon Redshift

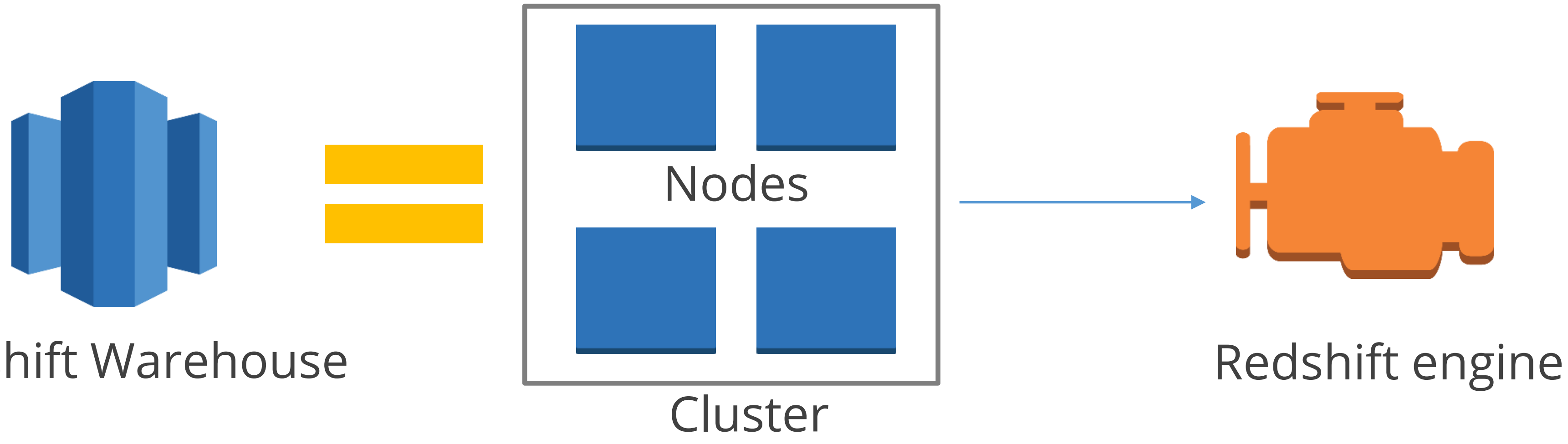
Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the cloud. The benefits of Amazon Redshift are as follows:



- Easy-to-use
- Best performance
- Highly scalable
- Cost-effective
- Easy-to-manage
- Secure and compliant

Amazon Redshift Clusters

An Amazon Redshift data warehouse is a collection of computing resources called nodes. Nodes are organized into a group called a cluster. Each cluster runs an Amazon Redshift engine and contains one or more databases.

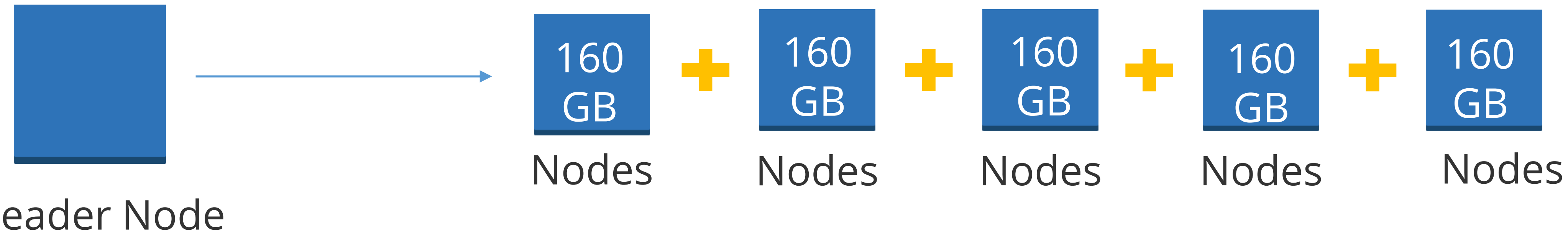


Redshift Warehouse

Nodes

Cluster

Redshift engine



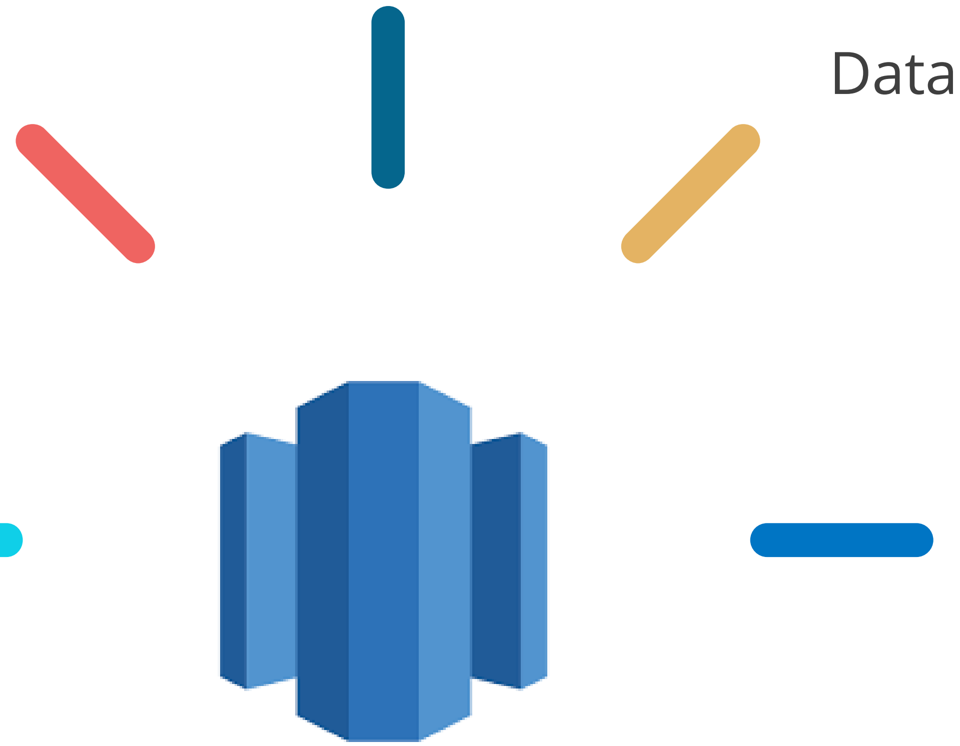
160 GB Nodes + 160 GB Nodes + 160 GB Nodes + 160 GB Nodes + 160 GB Nodes

Leader Node

Massive Parallel Processing

Amazon Redshift Costs

The following are the costs associated with Amazon Redshift:



Backup storage

Managed storage

Data transfer

Compute node hours

Data scan

Amazon Redshift

The use cases of Amazon Redshift:

- Data analytics as a service
- Session Store
- Streaming data analytics
- Real-time Analytics

Create Table, Store and Read Items in DynamoDB

Duration: 10 mins

Problem statement:

You have been assigned a task to create a DynamoDB using the AWS console.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Setting up the prerequisites for configuring the AWS console
2. Creating a table using DynamoDB

Create a DynamoDB global table

Duration: 8 mins

Problem statement:

You have been assigned a task to create a replica of the table.

Assisted Practice: Guidelines

Steps to be followed:

1. Configuring the global tables
2. Creating a replica using DynamoDB

Create a Cluster in Redshift, and Query Data Using Query Editor

Duration: 13 mins

Problem statement:

You have been assigned a task to create a Redshift cluster.

Assisted Practice: Guidelines

Steps to be followed:

1. Setting up the prerequisites for configuring the Redshift cluster
2. Creating a cluster, and querying data using query editor

Create an RDS MySQL database

Duration: 13 mins

Problem statement:

You have been assigned a task to create an RDS MYSQL database.

Assisted Practice: Guidelines

Steps to be followed:

1. Setting up the prerequisites for configuring MySQL
2. Creating a MySQL database

Key Takeaways

- Amazon RDS contains Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL, and MariaDB as database options.
- Amazon DynamoDB is a NoSQL database service that provides fast and predictable performance with seamless scalability.
- Amazon ElastiCache is a web service that makes it easy to deploy, operate, and scale an in-memory data store.
- Amazon DocumentDB is for the JSON format management scale.

Deploy MySQL RDS Using AWS

Duration: 30 mins

LESSON-END PROJECT

Project agenda: To launch MySQL Relational Database Services

Description:

You have been asked to design a solution to deploy MySQL using the AWS Console. The operations on MySQL instance need additional resources like virtual machine instances (EC2).

Perform the following:

1. Launch MySQL RDS
2. Launch a virtual machine
3. Set up the security groups for both the components to achieve networking between them

TECHNOLOGY

Thank You