AWS Foundation

DATABASE SERVICES

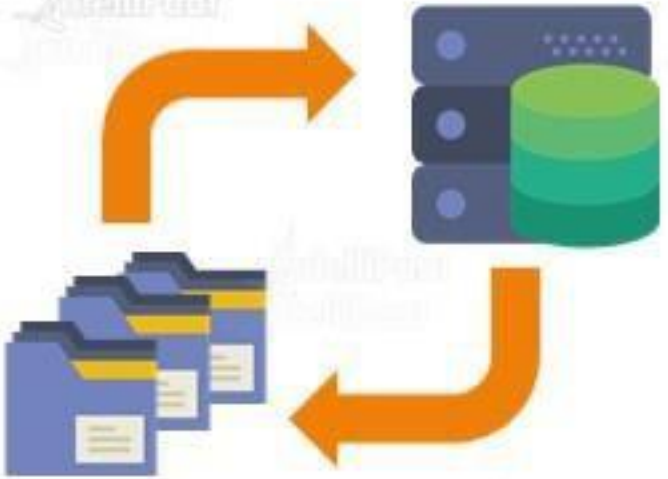
Agenda

1. Databases on AWS
2. Introduction to RDS
3. Multi-AZ Deployments
4. Features of RDS
5. Reserved DB Instances
6. Aurora Introduction
7. Introduction to DynamoDB
8. What is Redshift
9. ElastiCache
10. AWS Athena

What is Database ?

What is a Database ?

- Database is a collection of information that is organized so that the data or the information can be easily accessed, maintained, and updated. The data in a database can be added, updated, expanded, and deleted
- A software system created to perform all these operations on a database is called a database management system or DBMS



Types of Databases

Types of Databases ?

There are several types of databases that are categorized based on various factors. Following is the list of the most used and popular types of databases that are categorized mainly on the basis of the different ways in which the data is stored in each of them

- RDBMS
- NoSQL Database
- Hierarchical Database
- Flat File Database

Types of Databases ?

- RDBMS
- NoSQL Database
- Hierarchical Database
- Flat File Database

In RDBMS, data is stored in a tabular form, and SQL (structured query language) is used to run queries for inserting, updating, deleting, and searching for the data or records

Types of Databases ?

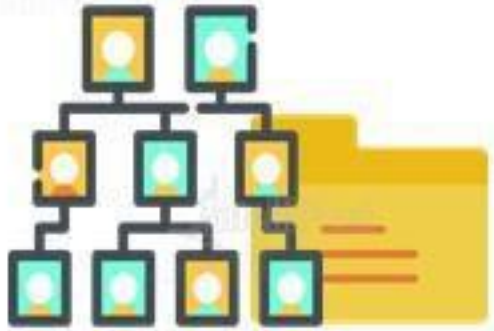
- RDBMS
- NoSQL Database
- Hierarchical Database
- Flat File Database

NoSQL databases don't follow the normal row/column or table approach of storing data like RDBMS. Thus, it is a non-relational database system. NoSQL databases store data in the JSON format. Data structures included in NoSQL are key/value, wide column, graph, or documentations

Types of Databases ?

- RDBMS
- NoSQL Database
- Hierarchical Database
- Flat File Database

Here, data is stored in parent–child relationship nodes. Hierarchical databases also contain the information of the respective groups of the data according to the parent–child relationships, along with the data



Types of Databases ?

- RDBMS
- NoSQL Database
- Hierarchical Database
- Flat File Database

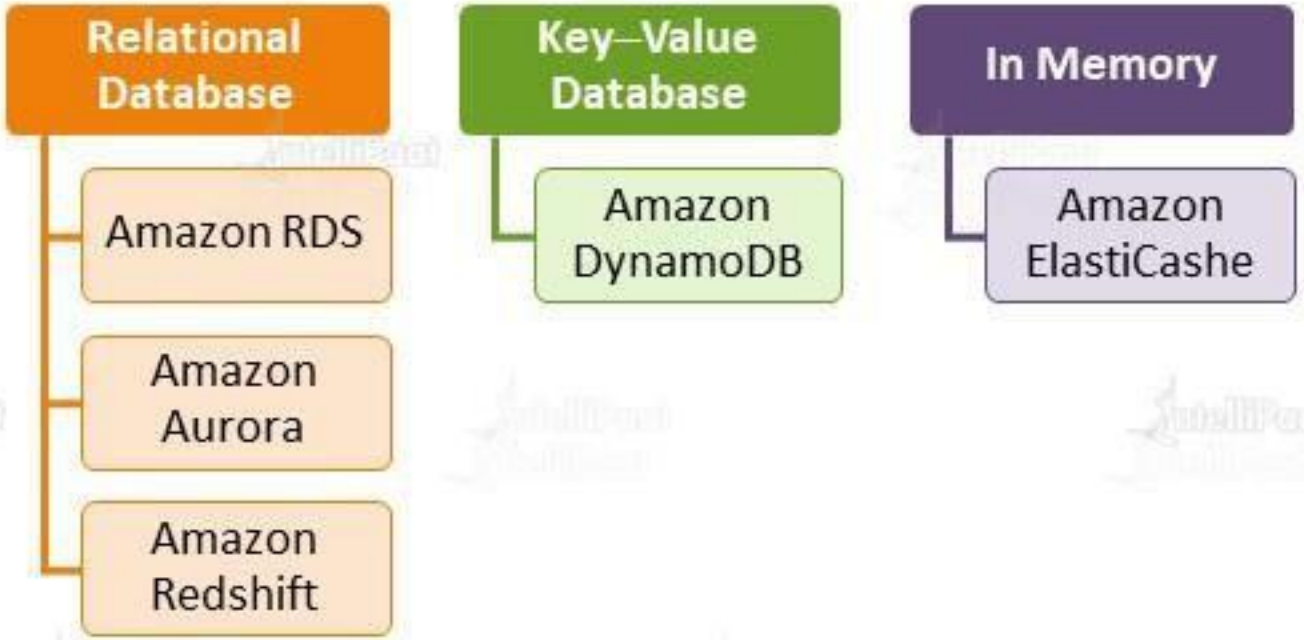
There are no particular structures for indexing and recognizing relationships among the data in flat file databases. All the records follow a uniform format and are saved in a file called flat file, which can be a plain text file or even a binary file

Databases on AWS

Databases on AWS

Amazon Web Services (AWS) provides fully managed and purpose-built database services to support relational as well as non-relational database requirements of their clients

Following are the database services provided by AWS



Introduction to RDS

Introduction to RDS

- Amazon RDS (Relational Database Service) is used to set up, manage, and scale a relational database instance in the cloud
- RDS is a fully managed RDBMS service
- Amazon RDS manages backups, software patching, failure detection, and many more tasks
- With RDS, CPU, memory, storage, and IOPS are all independent and hence can be scaled independently
- RDS offers mainly six database engines, namely, Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server

Multi-AZ Deployments

Multi-AZ

- Amazon RDS provisions and maintains a replica of DB instances synchronously. This replica is termed as standby replica, and the DB instance that gets synchronously replicated is termed as the primary DB instance
- The primary DB instance is replicated to a standby replica in different availability zones

The multi-availability zones deployment provides:

- Data redundancy
- Elimination of I/O freezes
- Less latency spikes during system backups

Features of Amazon RDS

Feature of Amazon RDS

- **Performance and Scalability**
- High Availability
- Security
- Backup and Restore
- Maintenance and Upgrades

- Various storage types in Amazon RDS, such as general-purpose storage and provisioned IOPS storage, are designed to deliver consistent, fast, and predictable I/O performance
- Compute and memory resources can be scaled up or down as per the deployment requirements. Scaling up can be done up to a maximum of 32 vCPUs and 244 GiB of RAM

Feature of Amazon RDS

- Performance and Scalability
- **High Availability**
- Security
- Backup and Restore
- Maintenance and Upgrades

Amazon RDS provides enhanced availability and failover support for database instances as it uses Multi- AZ deployments

Feature of Amazon RDS

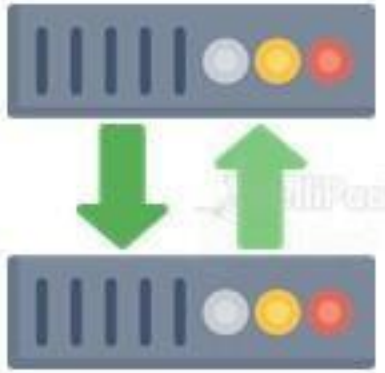
- Performance and Scalability
- High Availability
- Security
- Backup and Restore
- Maintenance and Upgrades

Amazon RDS allows to encrypt the database using the keys that are managed only by users. The automated backups, snapshots, or read replicas of the encrypted data will also be encrypted

Feature of Amazon RDS

- Performance and Scalability
- High Availability
- Security
- **Backup and Restore**
- Maintenance and Upgrades

- The feature of automated backups in Amazon RDS offers point-in-time recovery of the data in the database instance
- User-initiated backups are also an option for backing up in Amazon RDS. These are called database snapshots



Feature of Amazon RDS

- Performance and Scalability
- High Availability
- Security
- Backup and Restore
- Maintenance and Upgrades

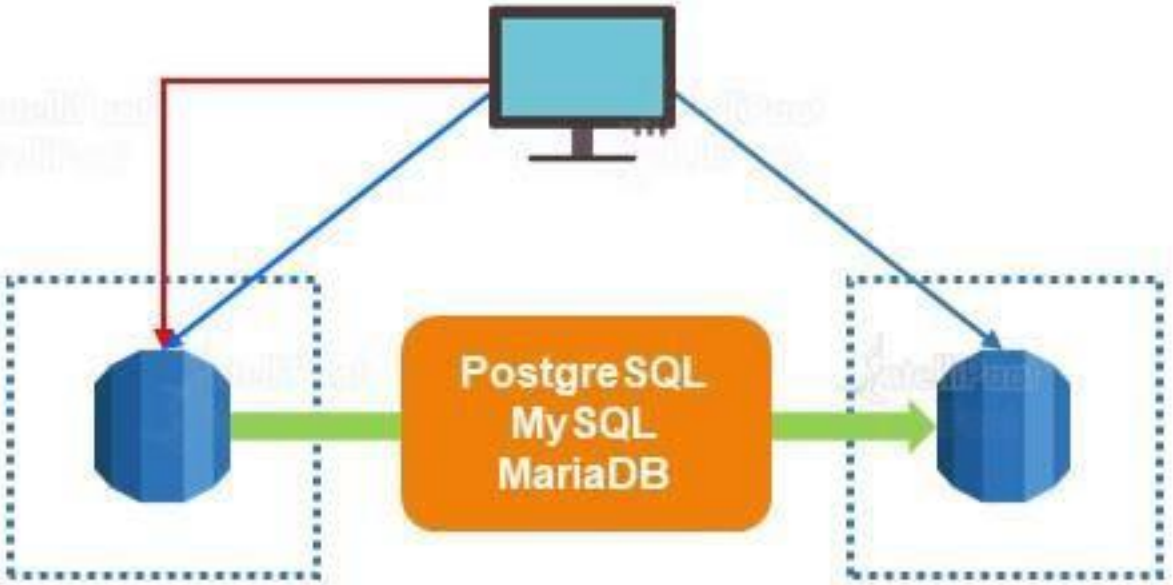
- RDS Management Console can be used to view key operational metrics, such as computer, memory, storage capacity utilization, I/O activity, and more
- Major engine version upgrades (changes might not be compatible with the existing applications) and minor version upgrades (backward-compatibility with the existing applications) can be performed

IntelliPaat

Read Replicas

Read Replicas

- Read replica is an exact replica of the primary DB instance
- RDS uses DB engine's built-in replication functionality to create these read replicas
- Replication happens asynchronously
- Read replicas can be created in the same or a different region. Automatic backup has to be enabled



Read Replicas

Amazon RDS does not support the circular replication of DB instances. Creating a read replica is different for all Amazon RDS DB engines; e.g., a read replica can be created from another read replica for MySQL and PostgreSQL, whereas the same is not possible with Oracle DB engines

Significant Differences Between the Read Replicas of Various RDS DB Engines

PostgreSQL	MySQL and MariaDB	Oracle
Physical replication	Logical replication	Physical replication
Read replicas cannot be made writeable	Read replicas can become writeable	Read replicas cannot be made writeable
Automated backups cannot be performed on PostgreSQL read replicas, but manual snapshots can be created	Automated backups can be performed	Manual snapshots cannot be created in Oracle read replicas. Automatic backups cannot be enabled as well

Read Replicas

Source Storage Type	Replication Storage Type	Storage Size
PIOPS	PIOPS or GP2 or Standard	100 GB to 3 TB
GP2	PIOPS or GP2 or Standard	100 GB to 3 TB
GP2	GP2 or Standard	Less than 100 GB
Standard	PIOPS or GP2 or Standard	100 GB to 3 TB
Standard	GP2 or Standard	Less than 100 GB

Reserved DB Instances

Reserved DB Instances

Amazon RDS offers an option to reserve a DB instance for a 1 or 3-year term for which a user gets a significant discount as compared to the on-demand instant pricing for DB instances

Features of Reserved DB Instances

- There are three types of payment options available for reserved instances, namely, All Upfront, Partial Upfront, and No Upfront
- Reserved instances can save up to an average of 69 percent over the on-demand instances
- Reserved instances are available in all of the AWS regions and for all supported DB engines

Pricing and Designs

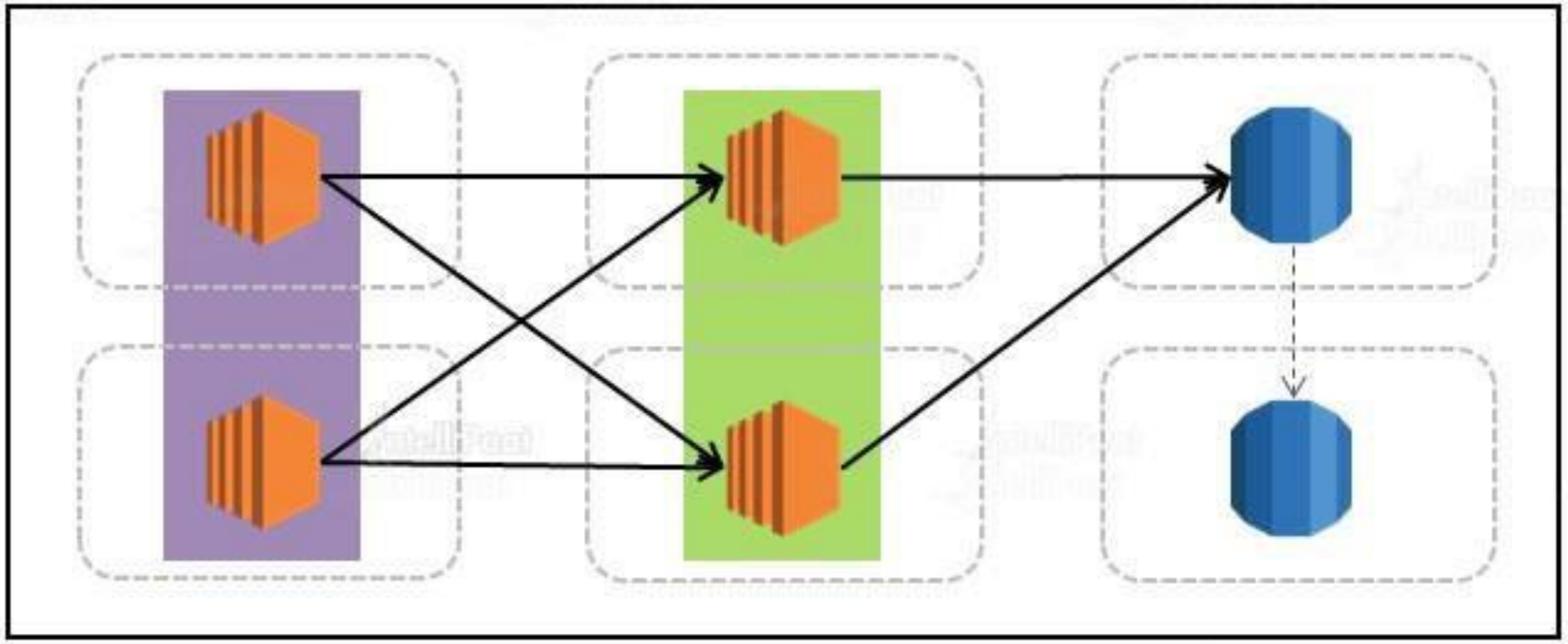
Pricing & Designs

- **On-demand instances:**
  - Single-AZ deployments
  - Multi-AZ deployments
- **Reserved instances:**
  - db.m4.large

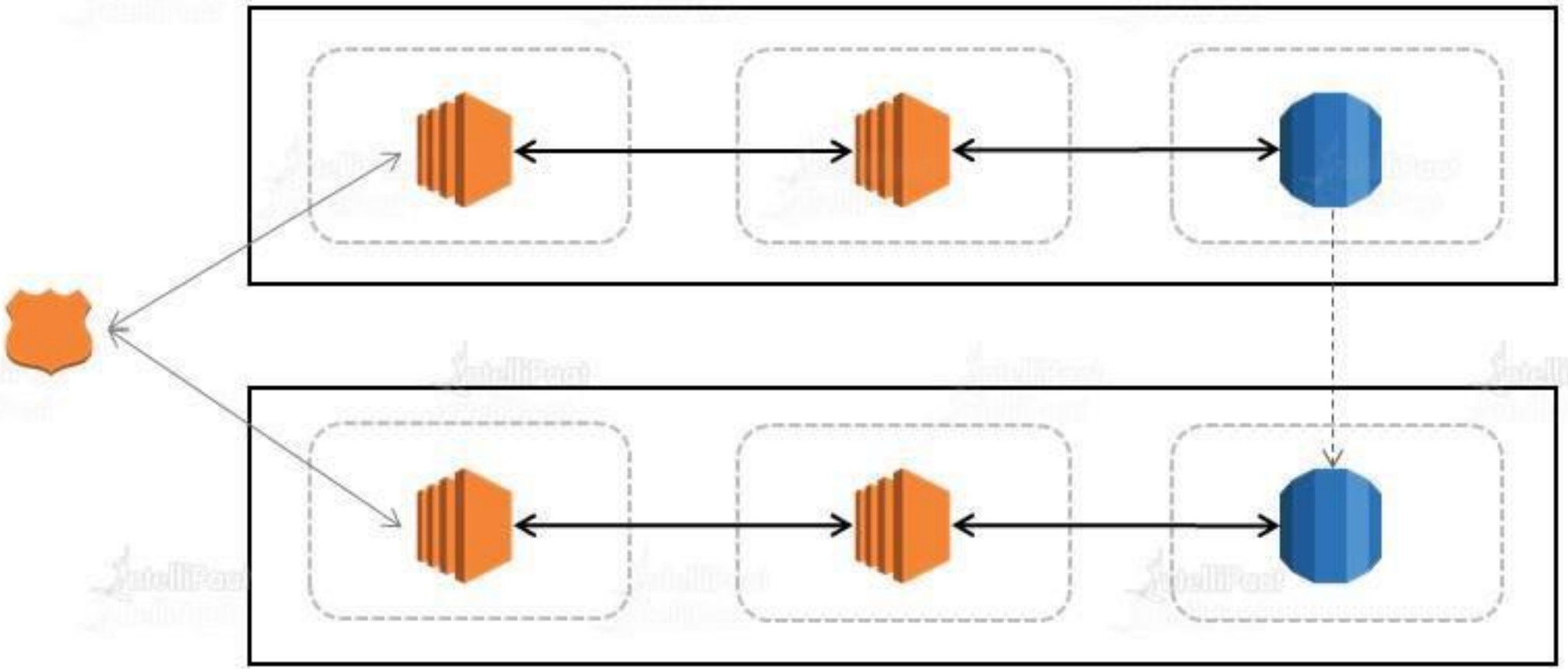
STANDARD 3-YEAR TERM					
Payment Option	**Upfront**	**Monthly***	**Effective Hourly****	**Savings over On-Demand**	**On-Demand Hourly**
Partial Upfront	$657.00	$32.85	$0.07	60%	$0.175
All Upfront	$1,717.00	$0.00	$0.06533	63%	

- **Storage:**
  - SSD (SAZ): US$0.115 per GB/month
  - SSD (MAZ): US$0.23 per GB/month
  - PIOPS (SAZ): Storage - US$0.125 per GB/month; PIOPS Rate - US$0.10 per IOPS/month
  - Magnetic (SAZ): Storage - US$0.10 per GB/month; I/O Rate - US$0.10 per 1 million requests
- **Backup:**
  - Free up to the size of active databases in a region. Additional storage costs US$0.095 per GB/month
  - If there's an active MySQL DB instance with 500 GB/month of provisioned database storage and an active PostgreSQL DB instance with 200 GB/month of provisioned database storage, up to 700 GB/month of backup storage will not be chargeable
- **Data transfer charges:** IN is free, but OUT depends on where the data is moving out

Pricing & Designs



RDS Design Patterns



Introduction to Amazon Aurora

Introduction to Amazon Aurora

- Amazon Aurora is a relational database engine built for the cloud, and it is completely compatible with MySQL and PostgreSQL
- Amazon Aurora is fully managed by Amazon RDS, which in turn automates administrative tasks, such as database setup, patching, backups, and more

Benefits of Amazon Aurora

Benefits of Amazon Aurora

High Performance and Scalability

Provides 5x throughput of standard MySQL and 3x throughput of standard PostgreSQL

High Security

Provides multiple levels of security, including encryption using user-generated keys and network isolation using Amazon VPC

High Availability and Durability

Offers greater than 99.99% availability and an option to backtrack to a previous point in time

Fully Managed

Fully managed by RDS and automates time-consuming tasks, such as hardware provisioning, software patching, setup, or backups

Pricing

Pricing

- DB instance: Pay-as-you-go pricing; the price varies from one instance class to another
- Pricing for various database engines in N. Virginia region (DB instance class: db.r3.xlarge, Multi-AZ deployment)
- Reserved DB instance
- For details, visit https://aws.amazon.com/rds/pricing/

Database Engine	Price Per Hour	Storage Cost	I/O Cost
Amazon Aurora	US$0.580	US$0.100/GB per month	US$0.200 per 1 million requests
MySQL	US$0.480	US$0.230/GB per month	US$0.20 per IOPS/Month
MariaDB	US$0.950	US$0.230/GB per month	US$0.20 per IOPS/Month
Oracle (SE2)	US$1.957	US$0.230/GB per month	US$0.20 per IOPS/Month
PostgreSQL	US$1.000	US$0.230/GB per month	US$0.20 per IOPS/Month

Introduction to DynamoDB

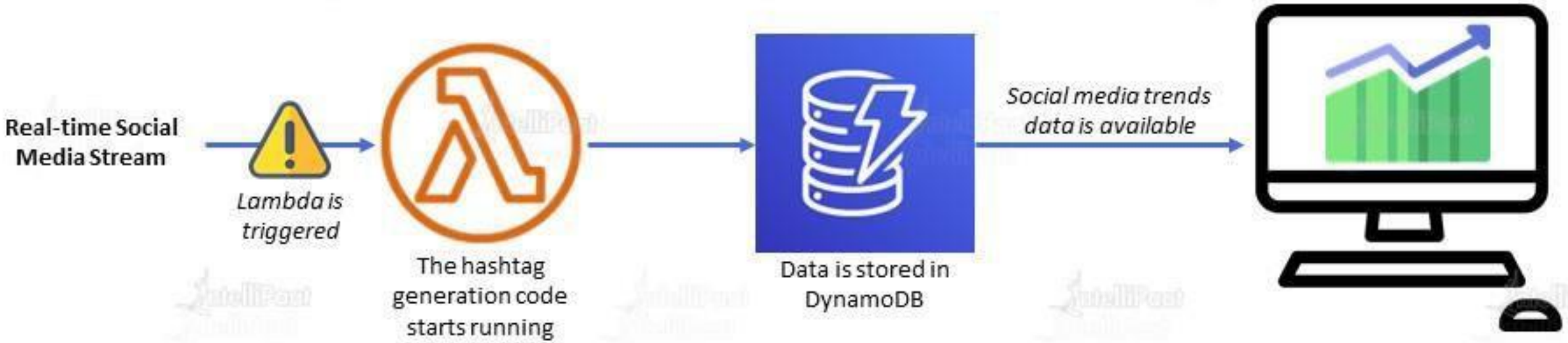
Introduction to DynamoDB

Amazon DynamoDB is a NoSQL database, which is fully managed by AWS to provide fast and predictable performance. It creates database tables that can store and retrieve any amount of data

- Key/Value and document data models
- Automated global replication
- Read/Write capacity modes
- Point-in-time recovery

Introduction to DynamoDB

Sample Use Case for a DynamoDB Table

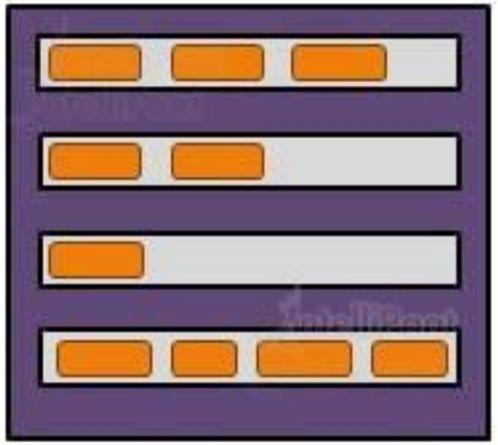


Core Components in DynamoDB

Core Components of DynamoDB

- **Tables, Items, and Attributes**
- **Primary Key**
- **Secondary Indexes**
- **DynamoDB Streams**

In DynamoDB, the collection of items is known as a table. A table in DynamoDB is not a structured table with fixed number of cells or columns

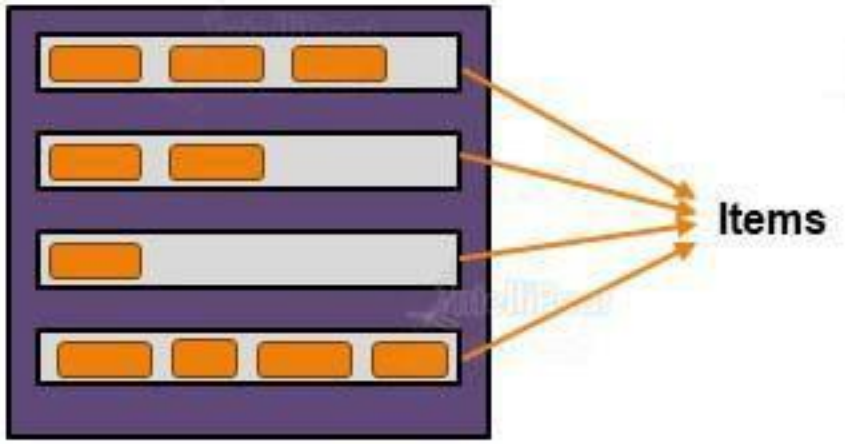


Table

Core Components of DynamoDB

- **Tables, Items, and Attributes**
- **Primary Key**
- **Secondary Indexes**
- **DynamoDB Streams**

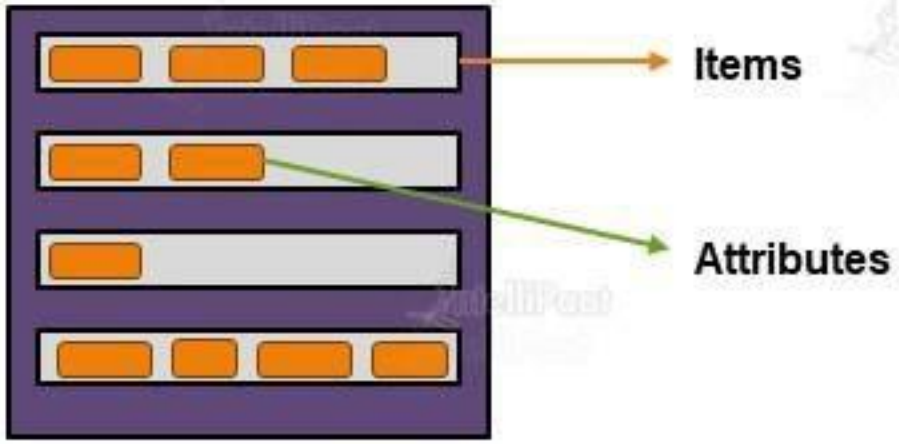
Each table contains zero or more items. An item is a group of attributes uniquely identifiable among all of the other items



Core Components of DynamoDB

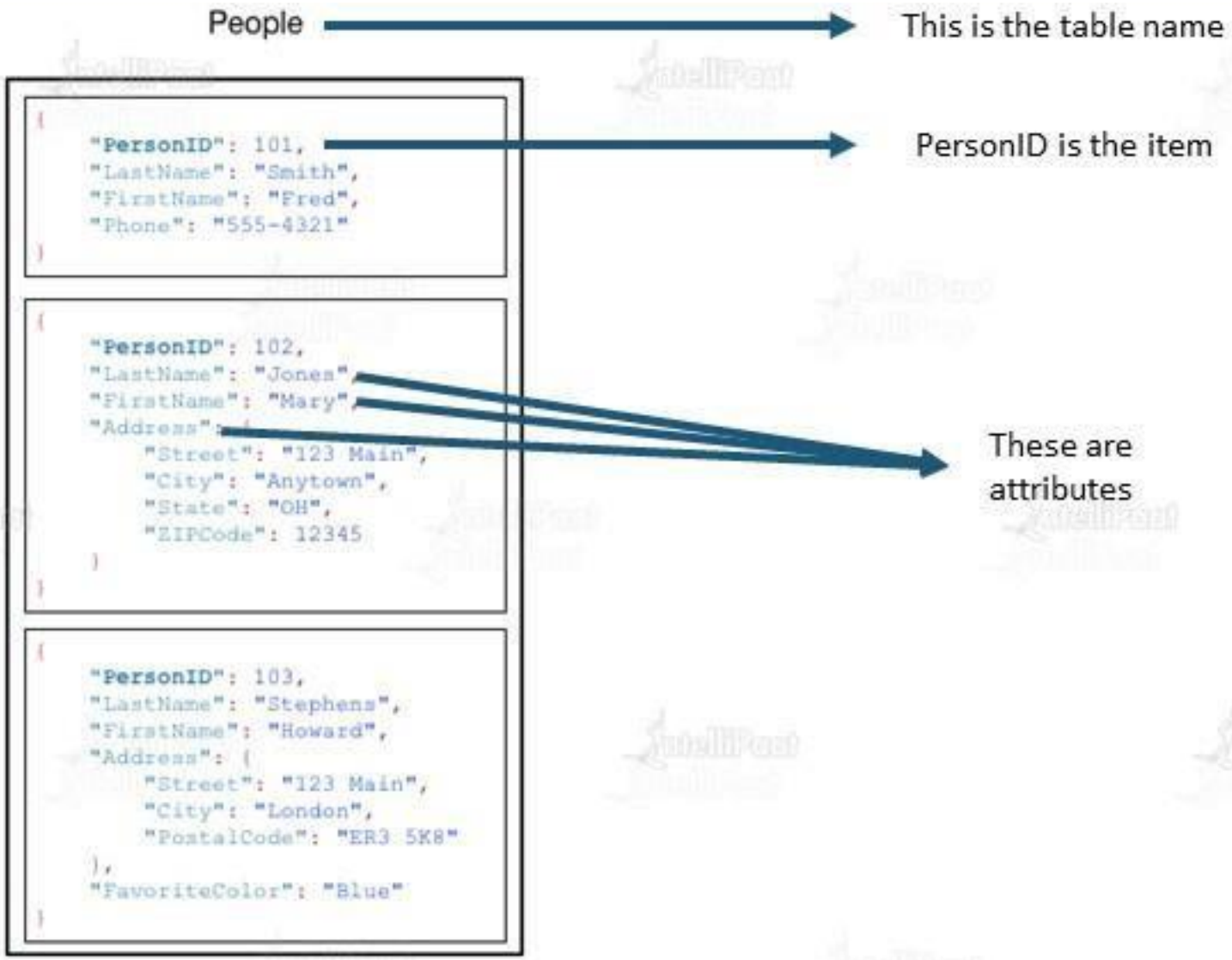
- **Tables, Items, and Attributes**
- **Primary Key**
- **Secondary Indexes**
- **DynamoDB Streams**

Each item is composed of one or more attributes. An attribute is a fundamental data element, which does not need to be broken down any further



Core Components of DynamoDB

- Tables, Items, and Attributes
- Primary Key
- Secondary Indexes
- DynamoDB Streams



Core Components of DynamoDB

- Tables, Items, and Attributes
- Primary Key
- Secondary Indexes
- DynamoDB Streams

When we create a table, in addition to the table name, we must specify the primary key of the table

Partition Key

A simple primary key comprises one attribute known as the partition key. In the previous example, 'PersonID' is the partition key

Partition Key and Sort Key

It is possible for two items to have the same partition key value. However, those two items must have different sort key values

Core Components of DynamoDB

- Tables, Items, and Attributes
- Primary Key
- Secondary Indexes
- DynamoDB Streams

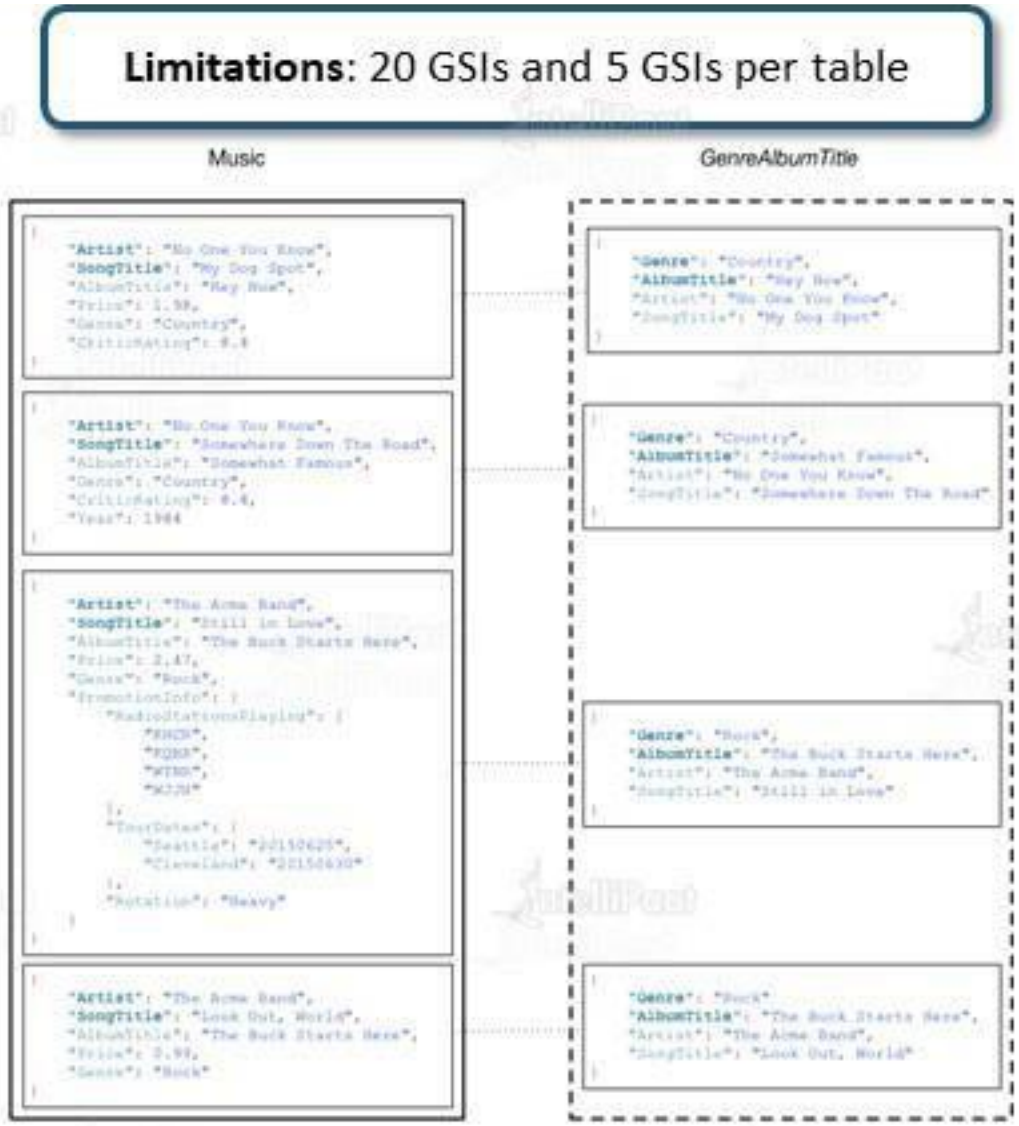
We can create one or more secondary indexes on a table. A secondary index lets us query the data in the table using an alternate key

Types of Secondary Indexes

1. Global secondary index: An index with a partition key and a sort key that can be different from those on the table
2. Local secondary index: An index that has the same partition key as the table, but a different sort key

Core Components of DynamoDB

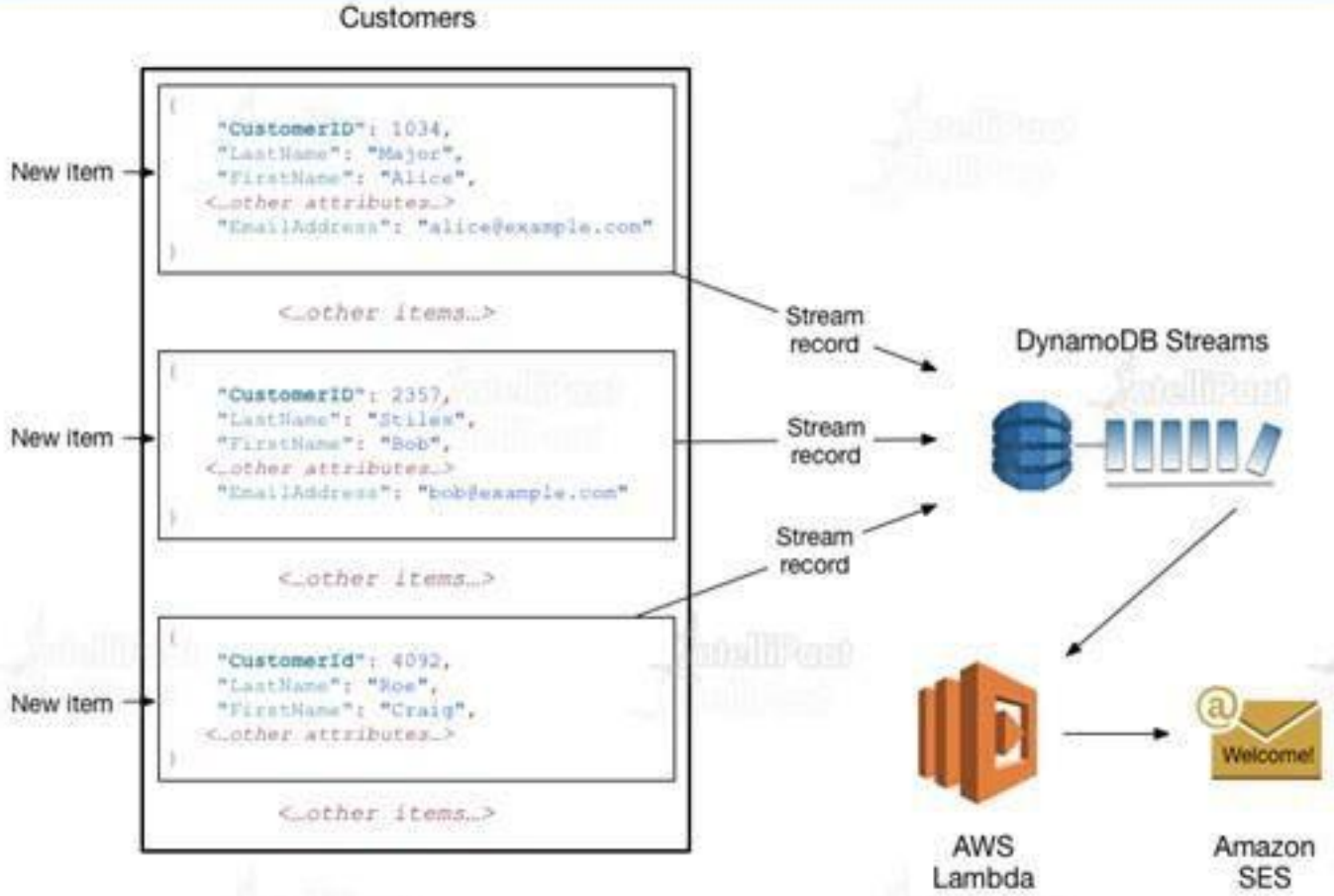
- Tables, Items, and Attributes
- Primary Key
- Secondary Indexes
- DynamoDB Streams



Core Components of DynamoDB

- Tables, Items, and Attributes
- Primary Key
- Secondary Indexes
- DynamoDB Streams

It is a feature that captures data modification events in DynamoDB tables, and it is optional



DynamoDB APIs

DynamoDB operations has to be done by below-mentioned APIs:

- CreateTable: Creates a new table. It can be used to create indexes as well
- DescribeTable: Returns information about tables
- ListTables: Returns all the tables
- UpdateTable: Modifies the settings of a table or its indexes, creates or removes new indexes on a table, or modifies DynamoDB Streams settings for a table
- DeleteTable: Removes a table and its dependent objects
- PutItem: Writes a single item to a table. The primary key must be specified here
- BatchWriteItem: Writes up to 25 items to a table
- GetItem: Retrieves a single item from a table with the primary key
- BatchGetItem: Retrieves up to 100 items from a table
- Query: Retrieves all items that have a specific partition key
- Scan: Retrieves all items from the specified table or index
- UpdateItem: Modifies one or more attributes in an item when the primary key is provided
- DeleteItem: Deletes a single item with a specific primary key

Concepts in DynamoDB

Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

Names should be meaningful and concise—for example, names such as 'Products', 'Books', and 'Authors' are self-explanatory

We can use following characters:

- a–z
- A–Z
- 0–9
- _ (underscore)
- - (dash)
- . (dot)

- Attribute names must be between 1 and 255 characters long

Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

- **Scalar types**: They can represent exactly one value. The scalar types are number, string, binary, Boolean, and null
- **Document types**: They can represent a complex structure with nested attributes as we would find in a JSON document
- **Set types**: They can represent multiple scalar values. The set types include string set, number set, and binary set

Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

Scalar Types

- **Number**: Numbers can be positive, negative, or zero
- **String**: It is a unicode with UTF-8 binary encoding, and the length should be greater than 0
- **Binary**: It can store any binary data, such as compressed text, encrypted data, or images
- **Boolean**: It can store either true or false
- **Null**: It is an attribute with an unknown or undefined state

Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

Document Types

- **List**: A list type attribute can store an ordered collection of values. Lists are enclosed in square brackets as [...]
- **Map**: A map type attribute can store an unordered collection of name–value pairs. Maps are enclosed in curly braces as { ... }

Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

Document Types

List

```
FavoriteThings: ["Cookies", "Coffee", 3.14159]
```

Map

```
{
    Day: "Monday",
    UnreadEmails: 42,
    ItemsOnMyDesk: [
        "Coffee Cup",
        "Telephone",
        {
            Pens: { Quantity : 3},
            Pencils: { Quantity : 2},
            Erasers: { Quantity : 1}
        }
    ]
}
```

Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

Set Types

DynamoDB supports the types that represent sets of number, string, or binary values. All elements within a set must be of the same type

```
["Black", "Green", "Red"]

[42.2, -19, 7.5, 3.14]

["U3Vubnk=", "UmFpbnk=", "U25vd3k="]
```

For example, an attribute of the type 'number set' can only contain numbers; a 'string set' can only contain strings, and so on

Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

DynamoDB supports **eventually consistent** reads and **strongly consistent** reads

Eventually Consistent

When reading data from DynamoDB, this may give us results quicker because of low latency, but the data might not be up to date

Strongly Consistent

In DynamoDB, this offers updated data all the time, but it has a very high latency

Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

Amazon DynamoDB has two read/write capacity modes for processing reads and writes on our tables:

- **On-demand**
- **Provisioned (default, free-tier eligible)**

Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

Amazon DynamoDB on-demand mode is a flexible billing option capable of serving thousands of requests per second without capacity planning

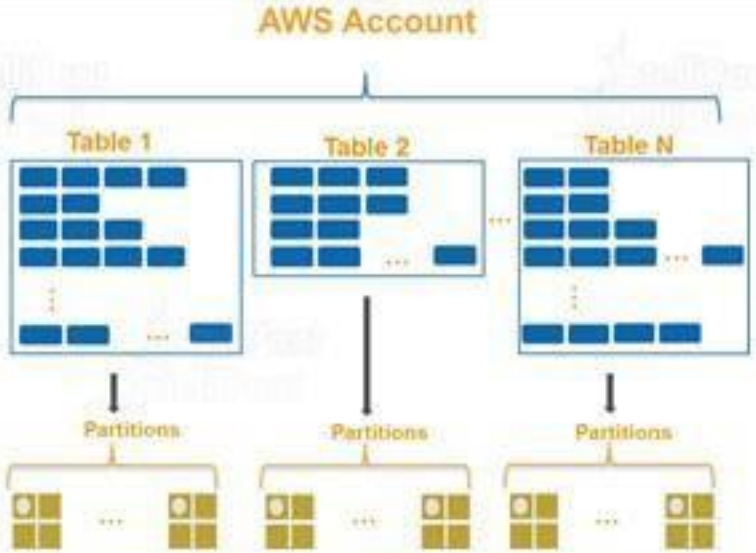
We use on-demand mode only under these conditions:

- When we create new tables with unknown workloads
- When we have unpredictable application traffic
- When we prefer the ease of paying for only what we use

Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

A **partition** is the allocation of storage for a table, backed by solid-state drives (SSDs) and automatically replicated across multiple availability zones within an AWS region



Partition management is handled entirely by DynamoDB—we never have to manage partitions ourselves

Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

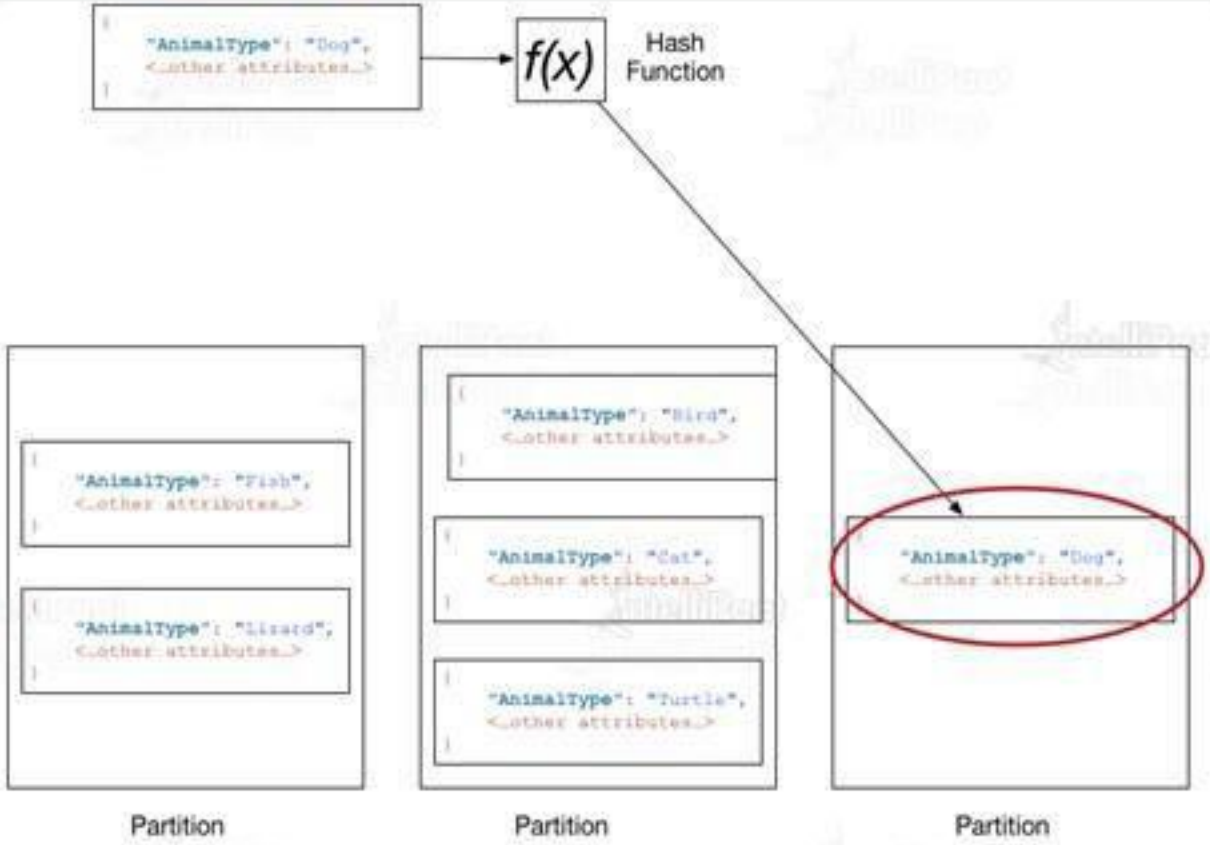
DynamoDB allocates additional partitions to a table in the following situations:

- If we increase the table's provisioned throughput settings beyond what the existing partitions can support
- If an existing partition fills up to the capacity and if more storage space is required

Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

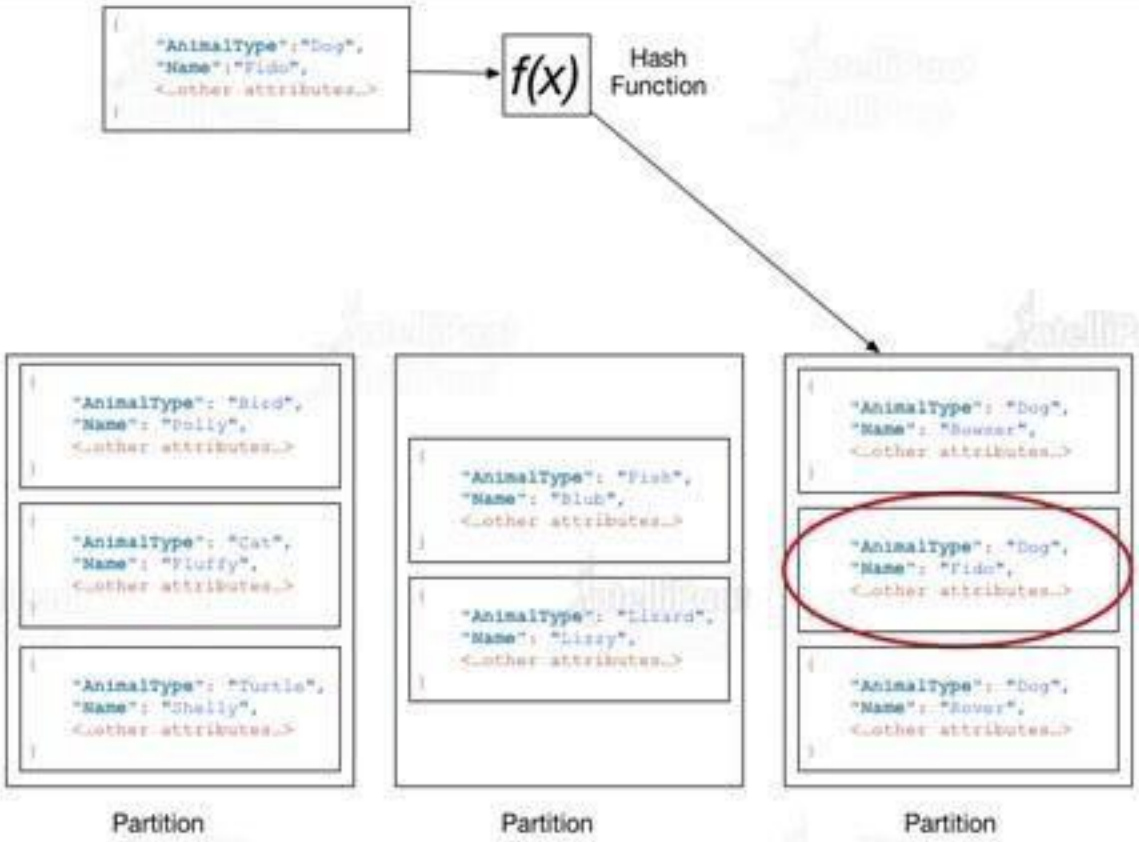
If our table has a simple primary key (partition key only), DynamoDB stores and retrieves each item based on its partition key value



Concepts in DynamoDB

- Naming Rules
- DynamoDB Data Types
- Read Consistency
- Read/Write Capacity Modes
- Partitions & Data Distribution

If the table has a composite primary key, DynamoDB calculates the hash value of the partition key the same way as for a simple primary key



Pricing and Designs

DynamoDB Pricing

- Free tier per month: 25 RCU, 25 WCU, 25 GB data storage, 2.5 million read requests from DynamoDB Streams. The free tier does not expire
- RCU: US$0.09 per RCU/month
- WCU: US$0.47 per WCU/month
- Data storage: US$0.25 per GB/month

Application requirement:

- 10 million eventual consistent reads per day
- 10 million writes per day
- 200 GB of data
- The maximum item size is 1 KB

Write per second = 10M/(24*60*60) = 115.7
WCU needed = 116
Write cost = 116 X 0.47 = US$54.52

Reads per second = 115.7
RCU needed = 116/2 = 58
RCU cost = 58 X 0.09 = US$5.22

Storage cost = 200 X 0.25 = US$50

Total cost = 54.52 + 5.22 + 50 = US$109.74

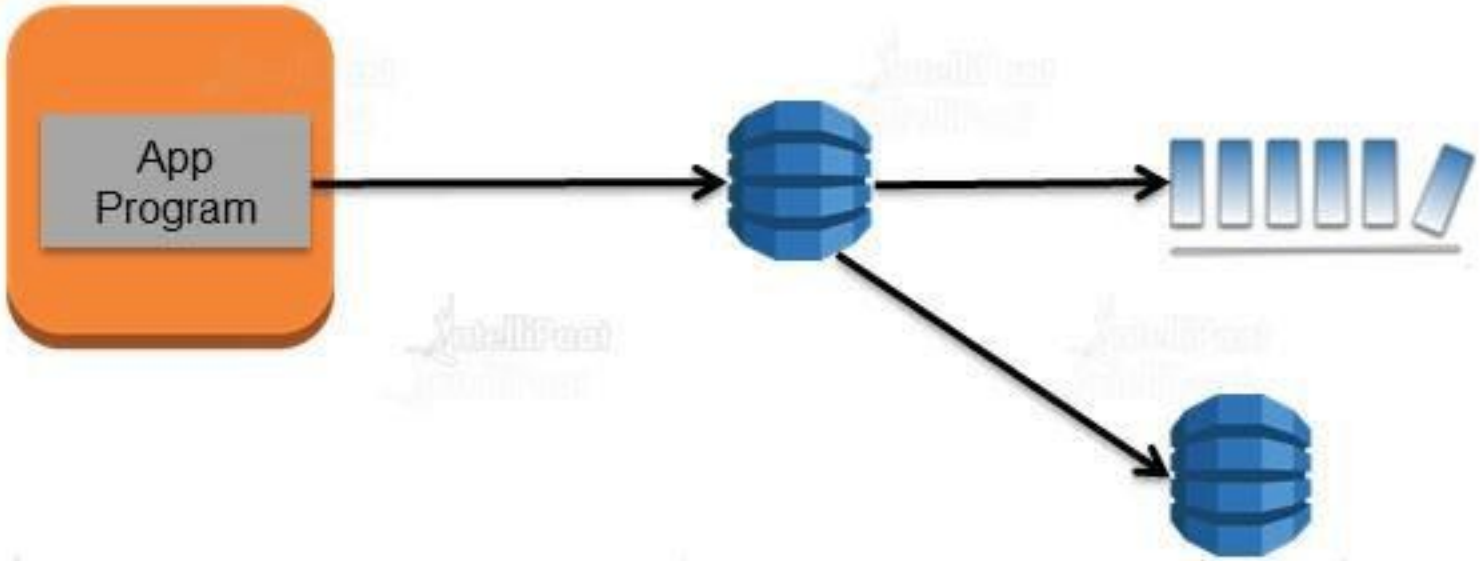
If the free tier is not consumed = 42.77 + 2.97 + 43.75 = US$89.49

DynamoDB Pricing – Reserved

- A minimum of 100 RCUs and WCUs
- The upfront fee has to be paid
- Maximum for 1 year
- Data transfer IN: Free
- Data transfer OUT (within the same region): Free
- Data transfer OUT (across other regions) Chargeable

Monthly Commitment	Upfront	Hourly
100 Write Capacity Units	$150.00	$0.0128 per Hour
100 Read Capacity Units	$30.00	$0.0025 per Hour

Design Patterns



What is Amazon Redshift ?

What is Redshift ?

- Amazon Redshift is a fully managed data warehouse service in the cloud. The data in the Amazon Redshift data warehouse can be scaled up to petabytes or more
- To create a database, a set of nodes called as Amazon Redshift cluster has to be launched

Advantages of Amazon Redshift

Advantage of Redshift

- Faster Performance
- Easy to Setup, Deploy, and Manage
- Cost-effective
- Secure

- Even on large datasets, Amazon Redshift delivers fast query performance. It also uses ML to predict incoming query runtime and assigns an optimal queue for fast processing
- It also provides performance boost when it comes to executing repeated queries by using result caching

Advantage of Redshift

- Faster Performance
- Easy to Setup, Deploy, and Manage
- Cost-effective
- Secure

- AWS provisions the infrastructure of the data warehouse automatically, making it simple and easy to create a new data warehouse with just a few clicks on the AWS console
- It also backs up the data in the data warehouse to Amazon S3 automatically

Advantage of Redshift

- Faster Performance
- Easy to Setup, Deploy, and Manage
- Cost-effective
- Secure

- Amazon Redshift is the most cost-effective data warehouse where users only have to pay for the resources they use
- Redshift is the only cloud data warehouse offering on-demand pricing with no upfront costs

Advantage of Redshift

- Faster Performance
- Easy to Setup, Deploy, and Manage
- Cost-effective
- Secure

- Amazon Redshift provides end-to-end encryption with just a couple of parameter settings. It takes care of key management
- It also enables users to configure firewall rules, providing full control over the network access of the data warehouse cluster

Why AWS ElastiCache ?

What is ElastiCache ?

ElastiCache is a web service that provides high-performance, cost-effective and scalable caching solutions. With ElastiCache, it is easy to manage and scale a distributed in-memory data store or cache environment in the cloud itself

Why choose ElastiCache

Response Time

Using ElastiCashe improves the load and response time as it allows to retrieve data from a fast in-memory system and eliminates the dependence on slower disk-based databases

Scalability

AWS ElastiCache is designed to be able to automatically modify itself to scale out or scale in as per the fluctuating application requirements

Complete Management

AWS ElastiCache can also automate common administrative tasks, such as hardware provisioning, failure recovery, backups, software patching, and more

AWS Athena

AWS Athena

Amazon Athena is a query service that allows you to easily analyse data in Amazon S3 using standard SQL. Because Athena is serverless, there is no infrastructure to manage, and you only pay for the queries you run.

AWS Athena

Benefits of AWS Athena

- Begins querying immediately. It is Serverless, no ETL
- It is Standard, open, and powerful.
- It is Presto-based application that runs standard SQL.
- It is really fast
- Even for large datasets, interactive performance is excellent.
- Pay per search
- Only pay for data that has been scanned.

.

Pricing

Pricing

Amazon Athena charges you for the number of bytes scanned, rounded up to the nearest megabyte, with a 10MB minimum per query. There are no fees for Data Definition Language (DDL) statements such as CREATE/ALTER/DROP TABLE, partition management statements, or failed queries. Queries that are cancelled are charged based on the amount of data scanned.

Pricing

Consider a table with three columns of equal size that is stored on Amazon S3 as an uncompressed text file with a total size of 3 TB. Because text formats cannot be split, running a query to get data from a single column of the table requires Amazon Athena to scan the entire file.

This search would cost $15. (The price for scanning 3 TB is 3 * $5/TB = $15.)

India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

support@intellipaat.com

24/7 Chat with Our Course Advisor