

# Database Services And Analytics

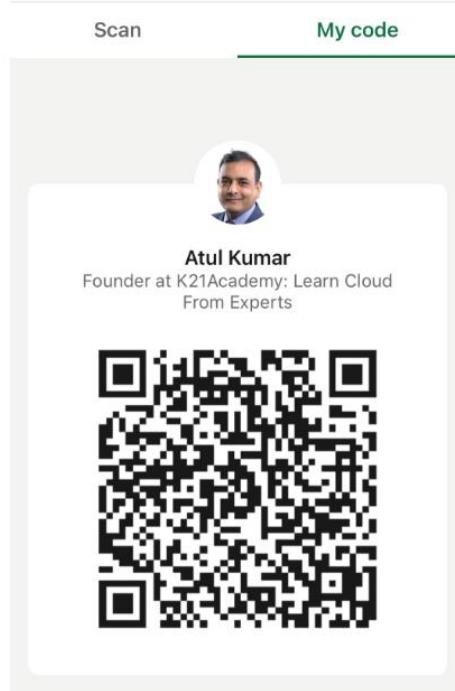


**Atul Kumar**  
Author & Cloud Expert



# Atul Kumar

LinkedIn QR code



- Author & Cloud Architect
- 21+ Years working in IT & Certified Cloud Architect
- Helped **6000+ individuals** to learn Cloud & Cloud Native



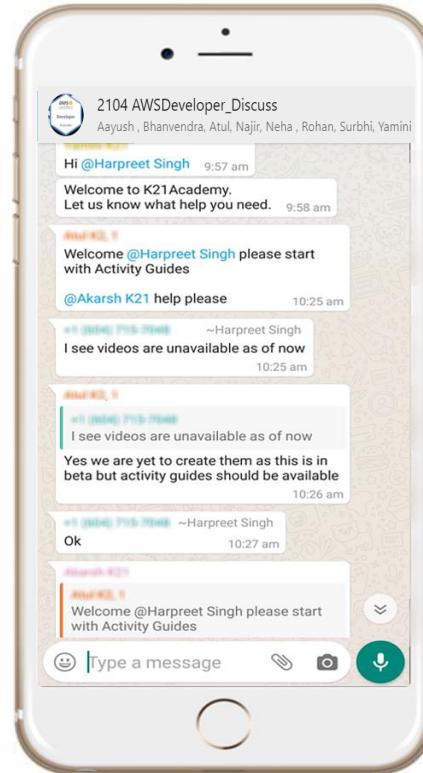
Oracle Identity and Access Management  
Administrator: Create Identity and Access Management instances, configuration, and day-to-day items

Atul Kumar

PACKT enterprise

# WhatsApp & Ticketing System

[support@k21academy.com](mailto:support@k21academy.com)





# Module Agenda

# Agenda: Module

- Database
- Types Of AWS Database Services
- Types Of AWS Database Services: Amazon RDS
- RDS DB Engines
- RDS DB Instances
- Amazon RDS : Read Replica
- DB Instance Regions and Availability Zones
- RDS Backup And Restore
- Types of AWS Database Services: Amazon Aurora
- Storage And I/O Traffic Aurora

# Agenda: Module

- Cash Recovery In AWS Aurora
- Types Of AWS Database Services: Amazon DynamoDB
- DynamoDB API
- Read Consistency In DynamoDB
- Amazon DynamoDB Accelerator (DAX)
- Types Of AWS Database Services: Amazon ElastiCache
- Amazon ElastiCache: Memcached
- Amazon ElastiCache: Redis
- Memcached V/S Redis
- Types Of AWS Database Services: Amazon Redshift
- Analytics- AWS Kinesis
- Analytics – Amazon QuickSight
- Analytics – AWS Glue



# Database

# Database

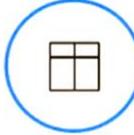
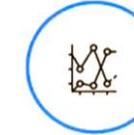
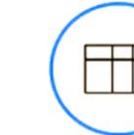
- A collection of individual data items which is stored in a highly structured manner is called a **Database**.
- It provides the ability to store **large amount** of information.
- Allows **quick Access** to information.
- It ensures **security** of data.
- Database is classified as **Relational and Non-Relational Database**.





# Types Of AWS Database Services

# Types of AWS Database Services

								
	Relational	Key-value	Document	In-memory	Graph	Time-series	Ledger	Wide Column
Referential integrity, ACID transactions, schema-on-write	High throughput, Low latency reads and writes, endless scale	Store documents and quickly access querying on any attribute	Query by key with microsecond latency	Quickly and easily create and navigate relationships between data	Collect, store, and process data sequenced by time	Complete, immutable, and verifiable history of all changes to application data	Scalable, highly available, and managed Apache Cassandra-compatible service	
AWS Service(s)	Aurora      RDS	DynamoDB	DocumentDB	ElastiCache	Neptune	Timestream	QLDB	Keyspaces Managed Cassandra
Lift and shift, ERP, CRM, finance	Real-time bidding, shopping cart, social, product catalog, customer preferences	Content management, personalization, mobile	Leaderboards real-time analytics, caching	Fraud detection, social networking, recommendation engine	IoT applications, event tracking	Systems of record, supply chain, health care, registrations, financial	Build low-latency applications, leverage open source, migrate Cassandra to the cloud	



# Types of AWS Database Services: Amazon RDS

# Amazon RDS

								
<b>AWS Service(s)</b>	<b>Relational</b>  Referential integrity, ACID transactions, schema-on-write	<b>Key-value</b>  High throughput, Low latency reads and writes, endless scale	<b>Document</b>  Store documents and quickly access querying on any attribute	<b>In-memory</b>  Query by key with microsecond latency	<b>Graph</b>  Quickly and easily create and navigate relationships between data	<b>Time-series</b>  Collect, store, and process data sequenced by time	<b>Ledger</b>  Complete, immutable, and verifiable history of all changes to application data	<b>Wide Column</b>  Scalable, highly available, and managed Apache Cassandra-compatible service
<b>Common Use Cases</b>	 Aurora  RDS	 DynamoDB	 DocumentDB	 ElastiCache	 Neptune	 Timestream	 QLDB	 * Keyspaces Managed Cassandra
	Lift and shift, ERP, CRM, finance	Real-time bidding, shopping cart, social, product catalog, customer preferences	Content management, personalization, mobile	Leaderboards, real-time analytics, caching	Fraud detection, social networking, recommendation engine	IoT applications, event tracking	Systems of record, supply chain, health care, registrations, financial	Build low-latency applications, leverage open source, migrate Cassandra to the cloud

# Why Managed Database Services?

When you choose on-premise database, you have to consider all these aspects

App Optimization

Scaling

High Availability

Database Backups

DB Software Patches

DB Software Installs

OS Patches

OS Installation

Server Maintenance

Rack and Stack

Power, HVAC, net

# Why Managed Database Services?

When you deploy Database on EC2 instance some portion of management is done by AWS



# Why Managed Database Services?

When you choose managed Database services 90% of management is done by AWS

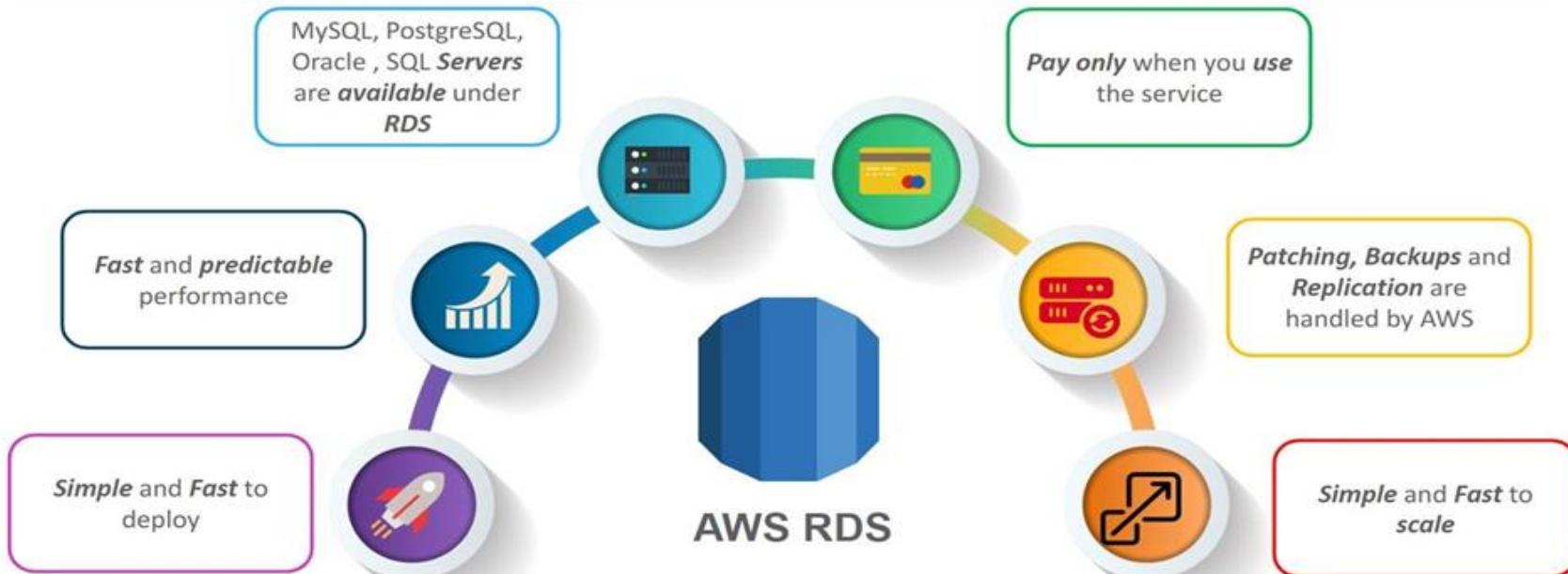


# Amazon RDS

- Amazon RDS is relational database management service which manages relational database for users.
- The role of RDS is to look after software patching, updates, backups, recovery and automatic failure detection.
- You can either create a backup via **Snapshot** or can have an automated backup performed.
- It is mainly used to manage data of **E-Commerce, Gaming, Apps, Websites** and many more.



# Amazon RDS Benefits

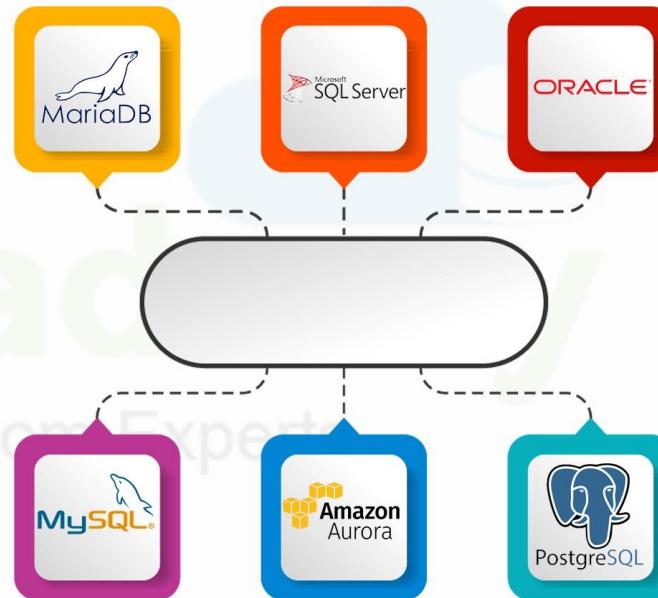




# RDS Database Engines

# Amazon RDS Database Engines

Database Engine is an underlying software component used by a database to create, read, update & delete data from it



# Database Schemas With DB Engines

Database Engine	Number of Schemas
Amazon Aurora	No limit imposed by software
MySQL	No limit imposed by software
MariaDB	No limit imposed by software
Oracle	1 database per instance no limit on number of schemas per database imposed by software
SQL Server	30 databases per instance
PostgreSQL	No limit imposed by software

# Database Engine Versions Supported

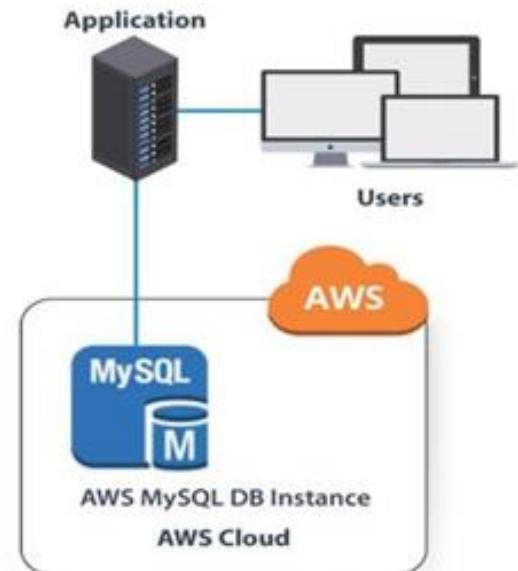
Database Engine	Version
Amazon Aurora	Compatible with MySQL and PostgreSQL database engines
MySQL	5.5, 5.6, 5.7
MariaDB	10.0, 10.1, 10.2
Oracle	11g, 12c
SQL Server	2008 R2 SP3 GDR 10.50.6560.0 2012 SP4 GDR 11.00.7462.6 2014 SP2 CU10 12.00.5571.0 2016 SP1 CU7 13.00.4466.4 2017 RTM CU3 14.00.3015.40
PostgreSQL	9.3, 9.4, 9.5, 9.6, 10.1, 10.3, 11 Beta 1



# RDS Database Instances

# Amazon RDS : DB Instance

- **Database Instance** is a set of memory structures that manages the database.
- It is the basic building block of **RDS** which runs on DB engine.
- Each DB Instance runs on a **DB engine**.
- By default, customer can have **40 RDS Instances**



# Amazon RDS : DB Instance Class

- The Computation and memory capacity of DB Instance is decided by the **DB Instance Class**.
- The necessity of DB Instance Class depends on users **processing power and memory requirement**.
- It supports mainly three types of resources : **Magnetic, General Purpose and Provisioned IOPS**.
- **Instance Class** offered by AWS RDS are:

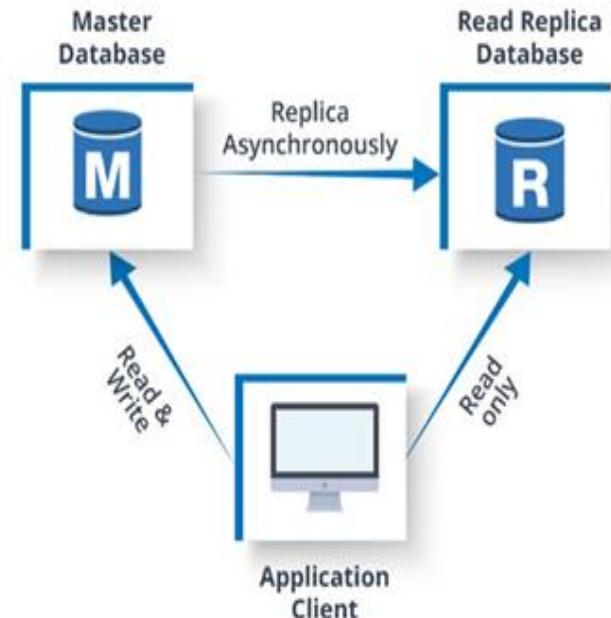
Standard	Memory Optimized	Burstable Performance
db.m4,db.m3,db.m1	db.r4 and db.r3	db.t2



# Amazon RDS: Read Replica

# Read Replica

- Read replica database is the copy of **master database** which reflects changes done to the master database system.
- Read Replica reduces the load on DB Instance.
- By taking the Snapshot of Source DB Instance, Amazon RDS creates **Read Only** Instance from the snapshots.
- The Read Replica allows **only read** connections.

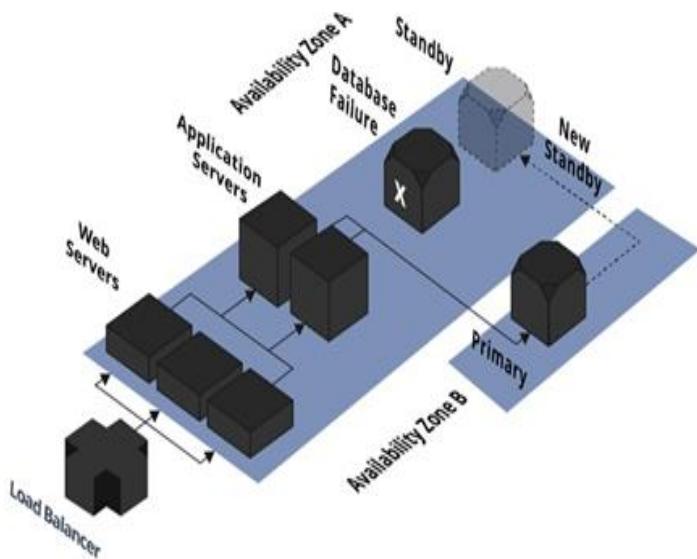




# DB Instance Regions And Availability Zones

# DB Instance Region & Availability Zones

- An Amazon RDS activity initiated runs only in your current default region.
- The default region can be changed in the console, by setting the EC2 region environment variable.
- Amazon RDS provides high availability and failover support for DB Instances by maintaining asynchronous standby replica in Multi availability zone deployments.



# DB Instance Failover Scenarios

**01**

If primary DB instance fails

**02**

An Availability Zone blackout

**03**

If OS of DB instance is undergoing software patching

**04**

A DB instance server type is changed

Learn Cloud from Experts



# RDS Backup And Restore

# RDS Backup And Restore

AWS RDS carries the ***automated backups*** of DB instances as per the specified ***backup retention period***

The ***backup retention period*** can be set between **1-35 days**

Even ***manually backups*** can be created via ***Snapshots***

When a DB instance is deleted the ***automated backups*** too get deleted with it but manual ***Snapshots*** remain the same

# Billing Of Amazon RDS

Parameters	Billing Procedure
DB instance hours	Based on the class, full hour will be considered if DB instance is consumed for partial hour
Storage (per GB per month)	Scaling your provisioned storage capacity within the month, will be billed pro-rated
I/O requests per month	Total number of storage I/O requests
Data transfer	Data transfer in and out of your DB instance on internet



# Types of AWS Database Services: Amazon Aurora

# Amazon Aurora

- Amazon Aurora is Relational Database which is managed by AWS

01

It is MySQL-compatible and delivers up to five times the performance of MySQL

02

It supports cross region region read replica

03

Offers simple monthly charge for each Amazon Aurora database instance usage

04

High performance and cost- effective



Amazon Aurora



# Storage and I/O Traffic In Aurora

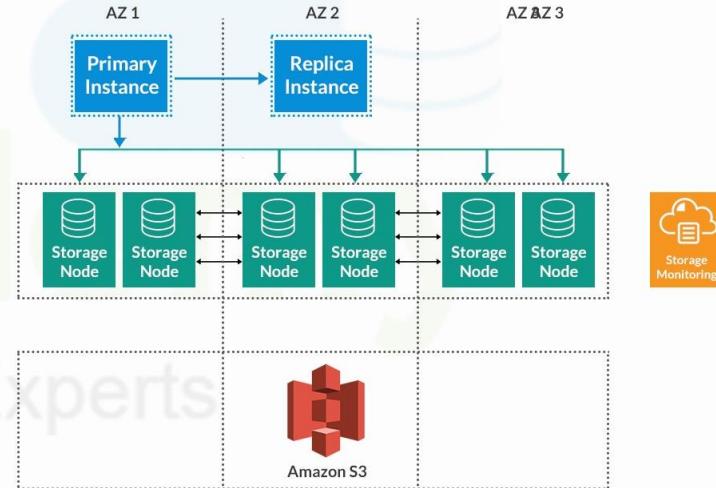
# Amazon Aurora Storage Engine

Amazon Aurora consists a storage volume of **10 GB** logical blocks  
(It can scale upto **64 TB** when required)

The data is replicated in **6 storage nodes** allocated in **3 Availability Zones** in parallel

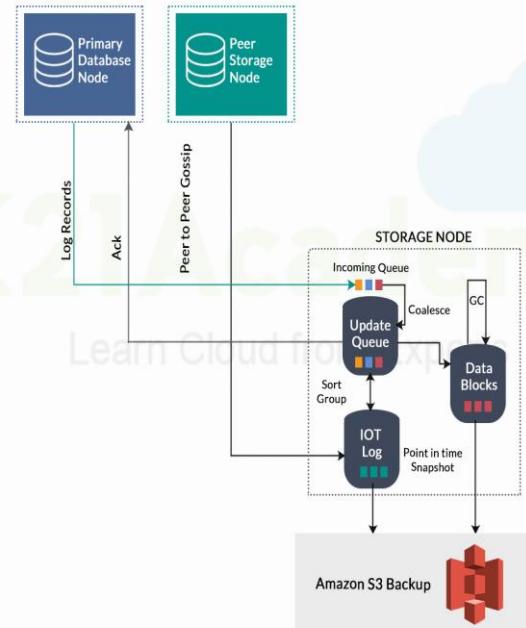
S3 manages the backups periodically

Continuous monitoring of these nodes is done to overcome repairs



# I/O Traffic In Aurora

- In each of the storage node the record first enters an **in-memory queue**.
- Persisted record is **carried to update the queue and primary Instance is acknowledged**.
- The records are acknowledged, and sort groups identifies gaps in logs.
- If Log sequence number or the storage node is lost, then they are retrieved from other nodes via **Gossip Protocol**.
- The log records are backed asynchronously to **Amazon S3**.
- Periodically garbage collects older versions and current records are stored in **Data Blocks**.





# Cache Recovery In AWS Aurora

# Cache Recovery : Traditional DB V/S Aurora

## Traditional Database

- Replay logs since the last checkpoint
- Generally takes 5 minutes between checkpoints
- MySQL works with Single-thread; number of disk accesses are very high

## AWS Aurora

- As part of disk read, redo of records is done on-demand
- Operations are Parallel, distributed, asynchronous
- Does not replay on Start-up of server

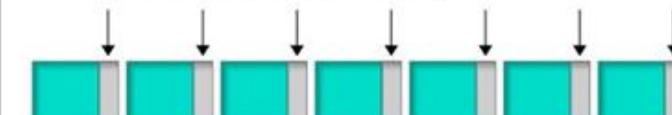
Crash at  $T_0$  requires  
a re-application of the  
SQL in the redo log since  
last checkpoint

Checkpointed Data

Redo Log

$T_0$

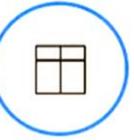
Crash at  $T_0$  will result in redo logs being applied to each segment  
on demand, in parallel, asynchronously





# Types Of AWS Database Service: Amazon DynamoDB

# DynamoDB

								
	Relational	Key-value	Document	In-memory	Graph	Time-series	Ledger	Wide Column
Referential integrity, ACID transactions, schema-on-write	High throughput, Low latency reads and writes, endless scale	Store documents and quickly access querying on any attribute	Query by key with microsecond latency	Quickly and easily create and navigate relationships between data	Collect, store, and process data sequenced by time	Complete, immutable, and verifiable history of all changes to application data	Scalable, highly available, and managed Apache Cassandra-compatible service	
AWS Service(s)	 Aurora  RDS	 DynamoDB	 DocumentDB	 ElastiCache	 Neptune	 Timestream	 QLDB	 Keyspaces Managed Cassandra
Common Use Cases	Lift and shift, ERP, CRM, finance	Real-time bidding, shopping cart, social, product catalog, customer preferences	Content management, personalization, mobile	Leaderboards, real-time analytics, caching	Fraud detection, social networking, recommendation engine	IoT applications, event tracking	Systems of record, supply chain, health care, registrations, financial	Build low-latency applications, leverage open source, migrate Cassandra to the cloud

# Difference Between SQL And NoSQL DB

Characteristics	SQL	NoSQL
Workloads	Adhoc queries, data warehousing, OLAP	Web-scale applications
Data Model	Well-defined schema where data is normalized into tables, rows and columns	Schema-less with a primary key and manages structured or semi-structured data
Data Access	SQL (Structured Query Language)	AWS Management Console or AWS CLI and perform adhoc tasks
Performance	Optimized for storage	Optimized for compute
Scaling	Vertical Scaling	Horizontal Scaling

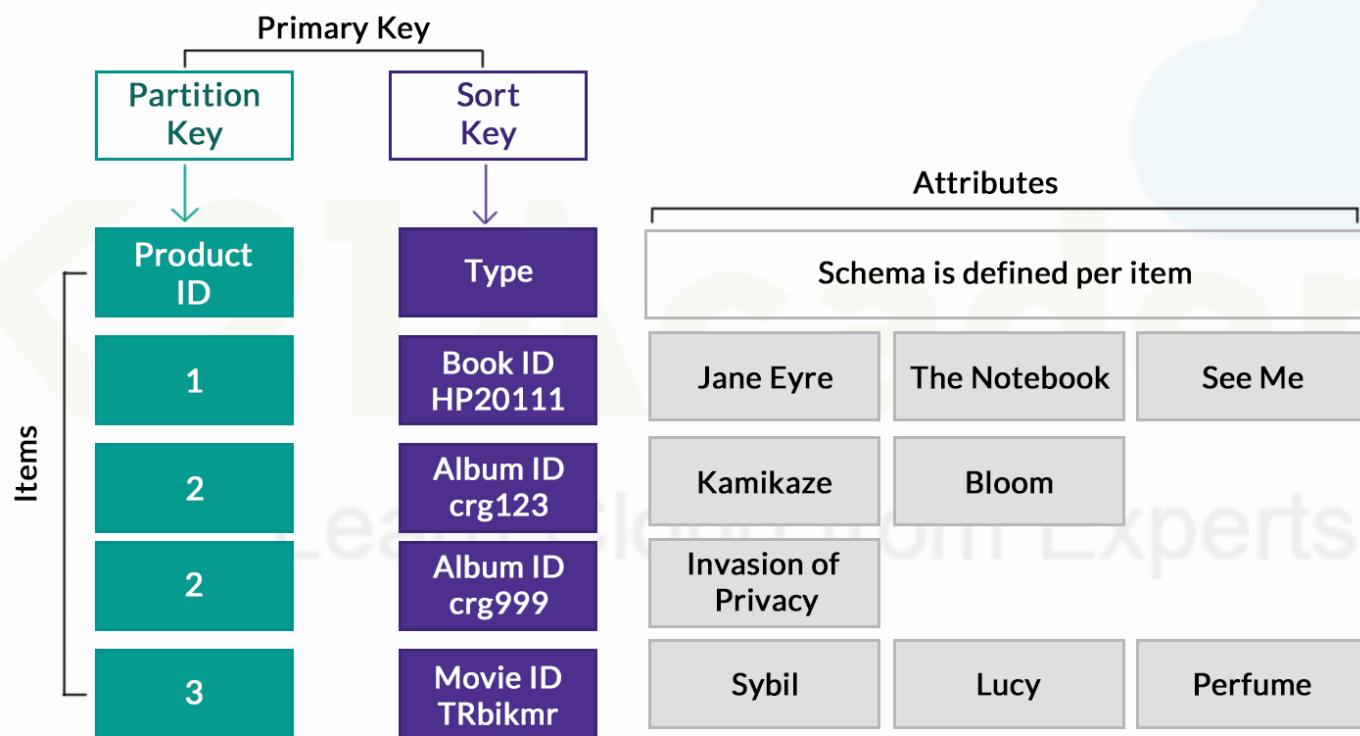
# Amazon DynamoDB

- **DynamoDB** is fully managed NoSQL database service offered by AWS.
- The record in every row is known as **Item**.
- Here you can set **TTL (Time To Leave)** to delete items automatically in the table once they expire.
- Operation like **create, insert, update, query, scan and delete** are performed in a table via appropriate API.
- For faster performance and data durability the table data is stored in **SSD link** and can be spread across servers in different availability zones.



DynamoDB

# Data Structure In DynamoDB





# DynamoDB API

# Control Plane

- **Control Plane** lets you create and manage DynamoDB Table.
- **Create Table:** Lets you create a table which includes table name, primary key, throughput settings.
- **Describe Table:** Used to view the details of the table.
- **Update Table:** Used to modify settings of table.
- **Delete Table:** Used to remove unused table.
- **List Table:** Used to return the names of DynamoDB table for the current AWS account and region.
- **Describe Limits:** It returns the current read and write capacity limits for the current AWS account and region.

# Data Plan

➤ Data Plan lets you perform CRUD action on data in a table.

## Creating Data

**PutItem**- To create an item

**BatchWriteItem**- To create up to 25 items in one or more tables

## Reading Data

**GetItem**- To read an item

**BatchGetItem**- To read up to 100 items from one or more tables

## Updating Data

**UpdateItem**- To update an item

**BatchGetItem**- To read up to 100 items from one or more tables

## Deleting Data

**DeleteItem**- To delete an item

**BatchWriteItem**- Delete up to 25 items in one or more tables

# DynamoDB Streams

- **DynamoDB streams** is used to replicate the data from a table to table in other region.

**API's used are:**

- **List Stream:** retrieves a list of stream descriptors for the current account and endpoint.
- **Describe Stream:** retrieves detailed information about a given stream.
- **Get Shard Iterator:** retrieves shard iterator, which describes a location within the shard.
- **Get Records:** retrieves the stream records within a given shard.



# Read Consistency In DynamoDB

# Read Consistency In DynamoDB

- DynamoDB supports both **Eventually Consistent Reads** and **Strongly Consistent Reads**.

## Eventually Consistent Read

It provides the Stale data instead of recently added data in the DynamoDB table (If you repeat your read request after a short time, the response should return the latest data)

## Strongly Consistent Read

Here, DynamoDB returns a response with the most up-to-date data, reflecting the updates from all prior write operations that were successful  
A strongly consistent read might not be available if there is a network delay or outage

# Throughput Capacity

On creating a table in DynamoDB you are supposed to specify the capacity requirements for read and write activity

A **read capacity unit** represents one strongly consistency read per second, or two eventually consistent reads per second, for an item up to 4 KB in size

Throughput capacity limits is specified in terms of read capacity unit and write capacity unit

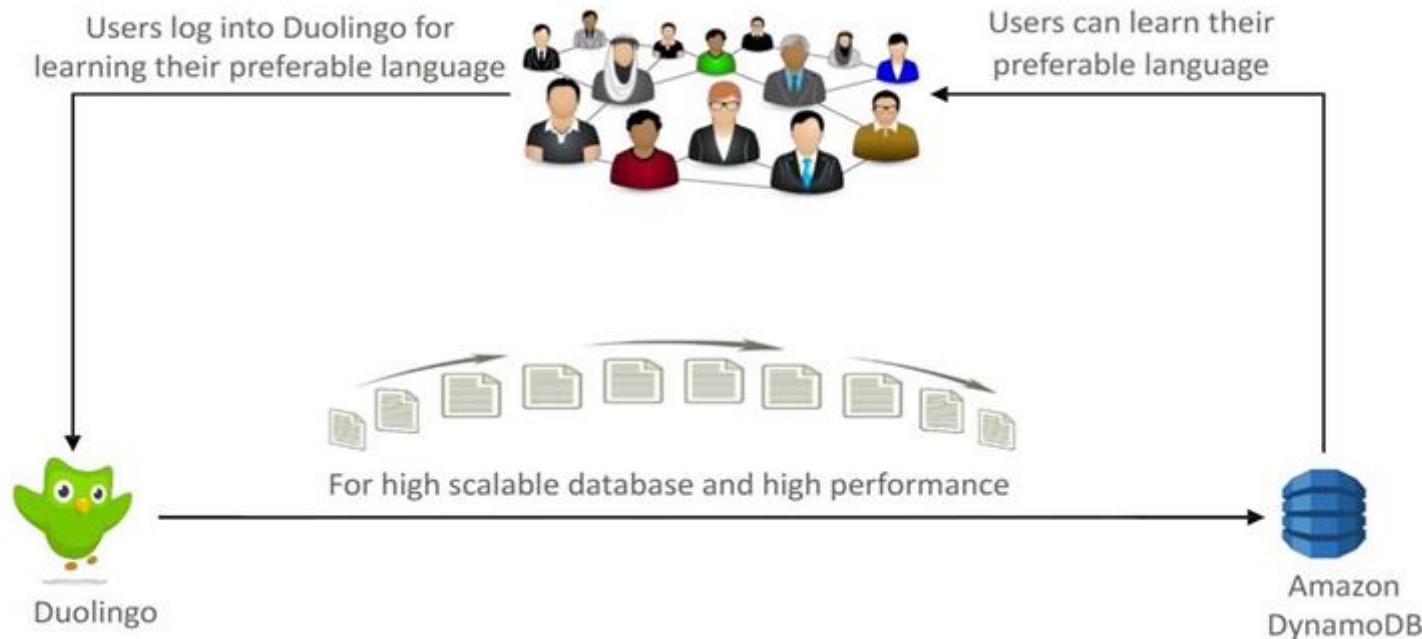
One **write capacity unit** represents one write per second for an item up to 1KB in size

# Amazon DynamoDB Benefits



Learn Cloud from Experts

# Amazon DynamoDB- Use Case



# How DynamoDB Works

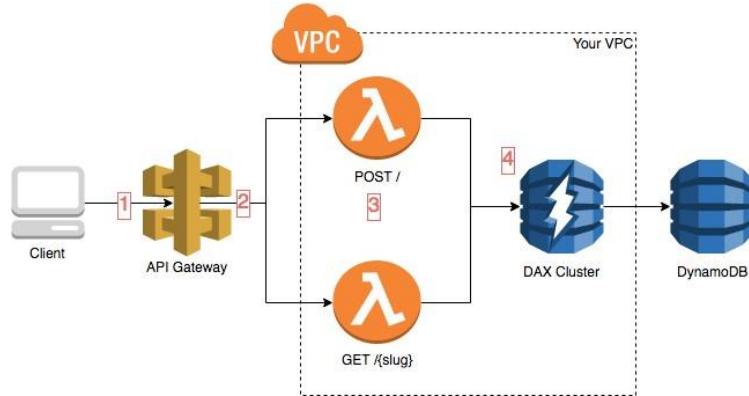




# Amazon DynamoDB Accelerator (DAX)

# Amazon DynamoDB Accelerator (DAX)

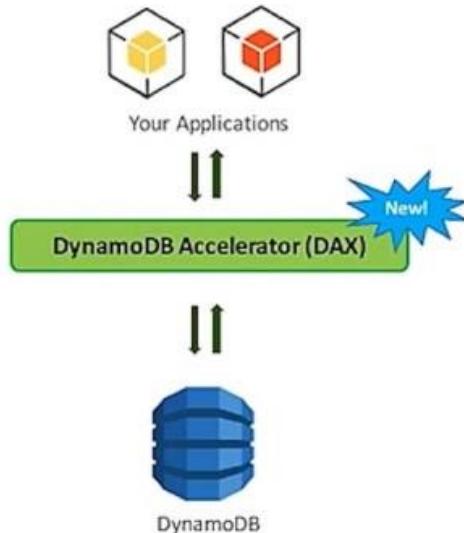
- Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache for Amazon DynamoDB that delivers up to a 10 times performance improvement—from milliseconds to microseconds—even at millions of requests per second.
- DAX does all the heavy lifting required to add in-memory acceleration to your DynamoDB tables, without requiring developers to manage cache invalidation, data population, or cluster management.



# Features of Amazon DynamoDB Accelerator (DAX)

## DynamoDB Accelerator (DAX)

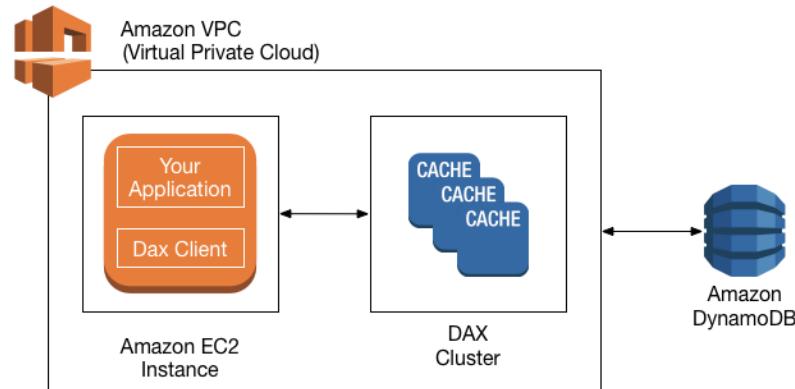
### Features



- **Fully managed:** handle all of the upgrades, patching, and software management
- **Flexible:** Configure DAX for one table or many
- **Highly available:** fault tolerant, replication across multi-AZs within a region
- **Scalable:** scales-out to any workload with up to 10 read replicas
- **Manageability:** fully integrated AWS service: Amazon CloudWatch, Tagging for DynamoDB, AWS Console
- **Security:** Amazon VPC, AWS IAM, AWS CloudTrail, AWS Organizations

# Working of Amazon DynamoDB Accelerator (DAX)

- DAX is designed to run within an Amazon Virtual Private Cloud (Amazon VPC) environment. Amazon VPC defines a virtual network that closely resembles a traditional data center. With a VPC, you have control over its IP address range, subnets, routing tables, network gateways, and security settings. You can launch a DAX cluster in your virtual network and control access to the cluster by using Amazon VPC security groups.
- The following diagram shows a high-level overview of DAX.





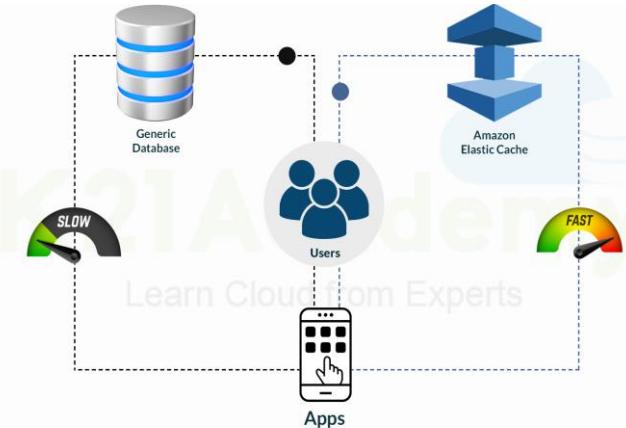
# Amazon Database Services: Amazon ElastiCache

# ElastiCache

								
AWS Service(s)	Relational  Referential integrity, ACID transactions, schema-on-write	Key-value  High throughput, Low latency reads and writes, endless scale	Document  Store documents and quickly access querying on any attribute	In-memory  <b>Query by key with microsecond latency</b>	Graph  Quickly and easily create and navigate relationships between data	Time-series  Collect, store, and process data sequenced by time	Ledger  Complete, immutable, and verifiable history of all changes to application data	Wide Column  Scalable, highly available, and managed Apache Cassandra-compatible service
Common Use Cases	  Aurora RDS	 DynamoDB	 DocumentDB	 <b>ElastiCache</b>	 Neptune	 Timestream	 QLDB	 Keyspaces Managed Cassandra
	Lift and shift, ERP, CRM, finance	Real-time bidding, shopping cart, social, product catalog, customer preferences	Content management, personalization, mobile	<b>Leaderboards, real-time analytics, caching</b>	Fraud detection, social networking, recommendation engine	IoT applications, event tracking	Systems of record, supply chain, health care, registrations, financial	Build low-latency applications, leverage open source, migrate Cassandra to the cloud

# ElastiCache

- ElastiCache is a cache-in environment used to cache results in order to reduce overhead and latency on database.
- It is a service which helps to improve the performance web applications by allowing user to retrieve information from fast, managed in-memory caches.
- It helps to set up, manage and scale distributed in-memory cache environment in the cloud.
- It supports two open source in-memory engines: Redis and Memcached.





# Amazon ElastiCache- Memcached

# Memcached Cache

- Memcached is general purpose distributed memory caching system used to speed up the dynamic data driven websites.
- Memcached is simple to use and multi thread.
- Memcached cluster can have a maximum 100 nodes in a region.
- It supports both horizontal and vertical scaling.
- It is instantly fast and well established.



**Memcached Cache**



# Amazon ElastiCache- Redis

# Amazon ElastiCache- Redis

- **Redis** is an In-memory data structure store, used as database, cache and message broker.
- It is single threaded, and its read replicas are synced **asynchronously**.
- Collection of 1 to 6 Redis nodes is called as **Shard**.
- Backups are stored in **S3** with a **backup period of 35 days**.



**Redis**



# Memcached V/S Redis

# Memcached V/S Redis

Characteristics	Memcached	Redis
Description	In-memory key-value store, originally intended for caching	In-memory data structure store, used as database, cache and message broker
Replication	Does not support replication	Supports master-slave replication
Storage type	Stores variables in memory & retrieves information directly from server instead of DB	Redis is like a database that resides in memory
Read/Write Speed	Good to handle high traffic websites	Neither can handle high traffic on read nor heavy writes
Key-Length	Memcached's key length has a maximum of 250 bytes	Redis has a maximum of 2GB
Ideal for	Caching relatively small and static data, such as HTML code fragments	Session Cache, Full Page Cache (FPC), Queues, Leaderboards /Counting and more

# Popular Use Cases of ElastiCache

## Ad Tech

- Ad Serving
- ID looking
- User profile management
- Session tracking
- Real time bidding

## Internet of Things

- Tracking State
- Metadata and reading from millions of devices
- Real time notification

## Gaming

- Recording game Details
- Leaderboards
- Session information
- Usage history
- Logs

## Mobile & Web

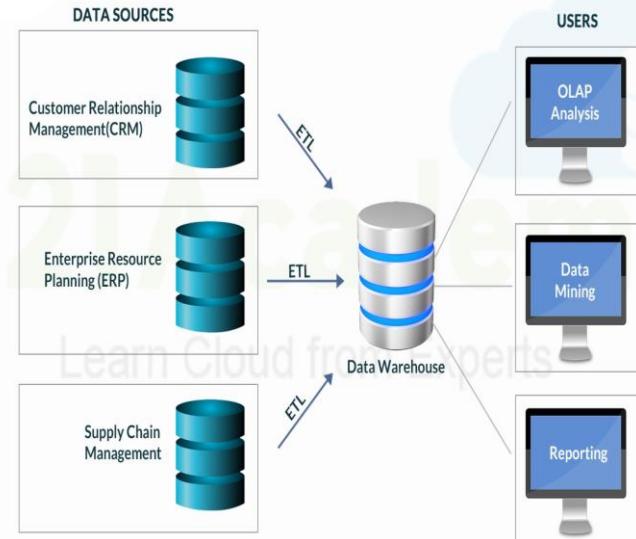
- Storing user profile
- Session details
- Personalization settings
- Entity-specific metadata



# Amazon Database Services: Amazon Redshift

# Data Warehouse

- **Data Warehouse** is a repository where data generated by different sources is **collected, stored, and transformed** for organization's decision-making process.
- It reduces total turnaround time required for analysis and reporting.
- It is also used to store historical data which helps users while analyzing data.
- It provides different time periods and trends to make future predictions.



# AWS Redshift

- Redshift is fully managed peta-byte scale Data Warehouse Service by AWS to do analysis on data.

01

Redshift is a combination of **nodes** and **cluster**

02

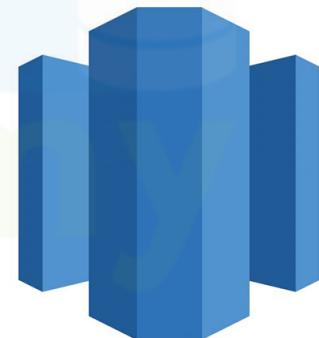
The total storage capacity of a cluster is determined via node size and number of nodes

03

Here, in the same Availability Zone of a region all cluster nodes are created

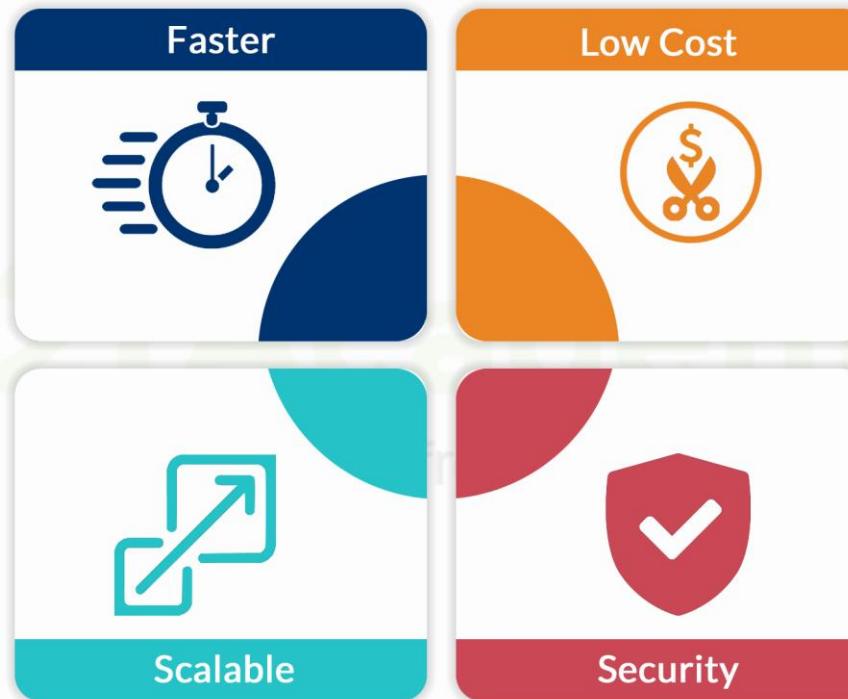
04

To execute queries faster Redshift makes use of massively parallel processing, columnar storage, data compression and zone mapping

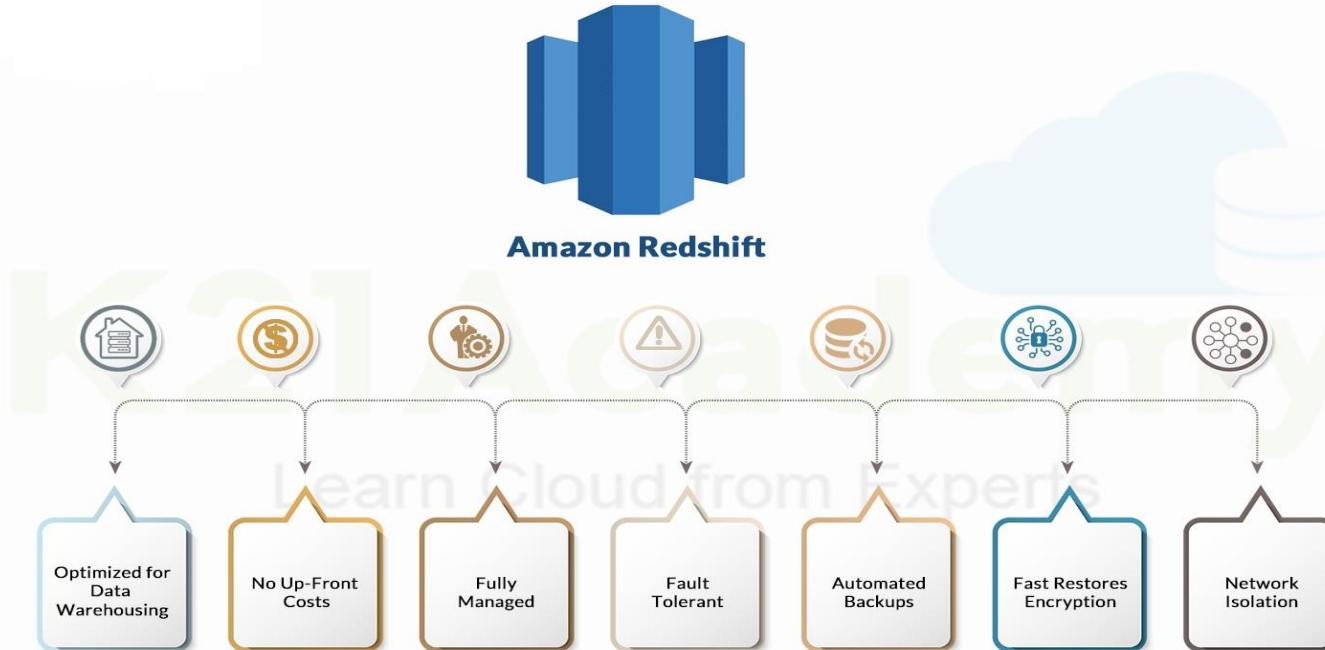


AWS Redshift

# Why Amazon Redshift

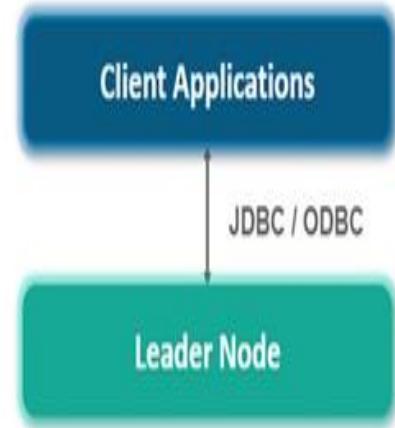


# Amazon Redshift Features



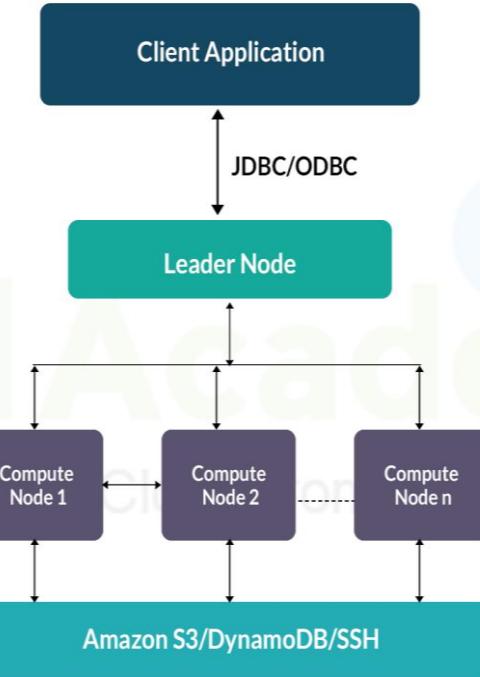
# Amazon Redshift Architecture

- AWS Redshift is a collection of computing resources known as **Nodes**, where nodes are organized to form a **Cluster**.
- Client Applications are the tools to work with Redshift e.g. SQL, Workbench etc.
- Client Applications connect to Redshift Cluster via **JDBC** or **ODBC** drivers.
- The Lead Node communicates with client application and compute Nodes.



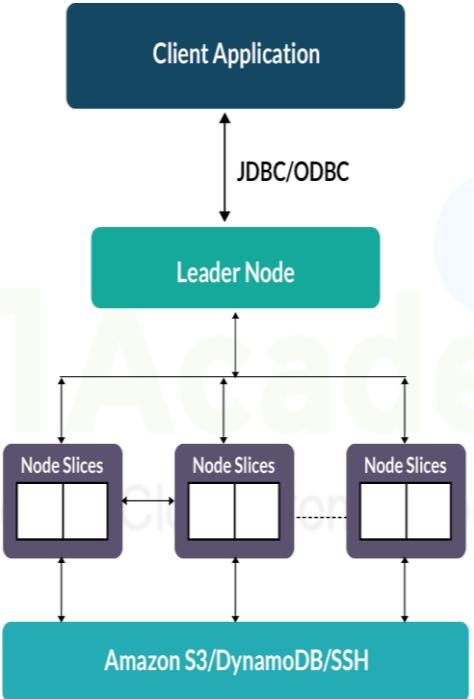
# Amazon Redshift Architecture

- Each compute node has its own CPU, memory and attached storage.
- There are two types of Nodes: **Dense Compute(SSD Based)** and **Dense Storage. (HDD Based)**
- All the user data gets stored in **compute node**.
- It performs the function as **load, backup and restore** via **Amazon S3, DynamoDB and SSH**.



# Amazon Redshift Architecture

- **Node Slice** is mainly used for the distribution of data within the node.
- Each slice is allocated a piece of **node memory** and **disk space**, where it processes the portion of workload assigned to its node.
- Once the leader node assigns the operations to the slices then these slices work in **parallel** to complete the operation.





# Analytics – AWS Kinesis

# Streaming Data

- **Streaming Data** is a data which is generated from thousand of sources.
- The Sources send data in small sizes i.e., Kilobytes.
- The data is processed **sequentially** and **incrementally** on record basis and used for wide range of analytics.
- Companies like **Flipkart, Uber** use Streaming Data.



# AWS Kinesis

- **Kinesis** is a data analysis service by Amazon which provides an easy way to collect, process and analyze real-time streaming data to get time insights so as to provide quick response to the information.
- **Kinesis Video Stream** is used to capture, process and analyze video stream for machine learning and Analytics.
- **Kinesis Data Firehose** is used to load data into AWS data Stores.
- **Kinesis Data Analytics** is an easy way to process data stream with SQL.



# Features of Kinesis



# Benefits of Kinesis Stream

- Easy to provision, deploy and manage.
- Elastically scalable and serverless.
- Reduce Latencies.
- Pay as you go, no upfront costs.
- Right services for your specific use cases



**AWS Kinesis**



# Analytics – Amazon QuickSight

# Amazon QuickSight

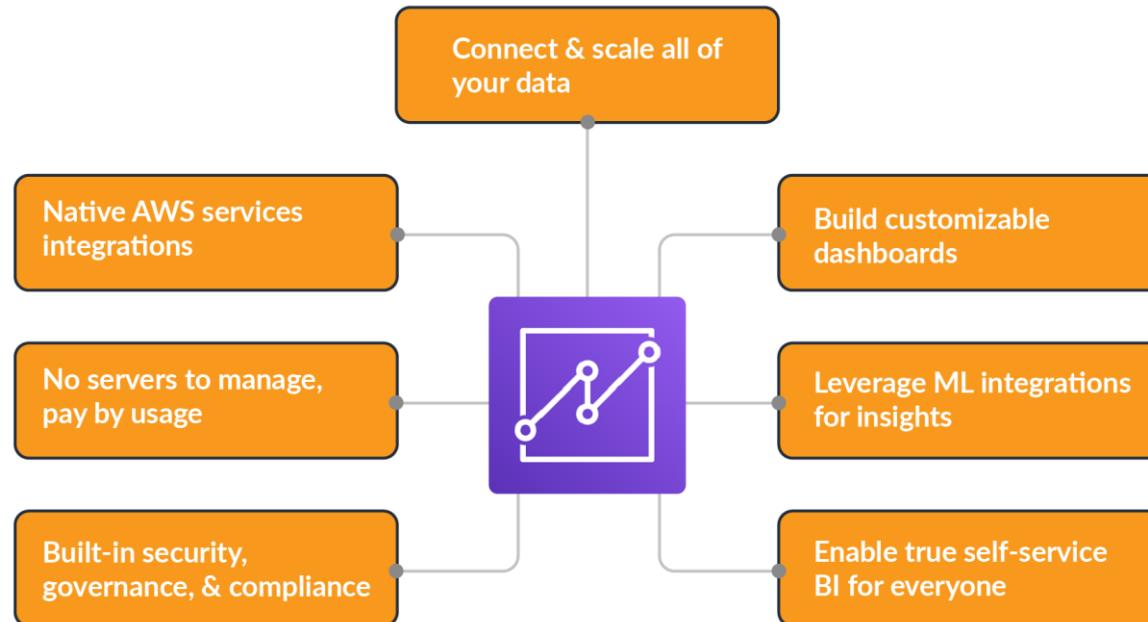
- Amazon QuickSight is a very fast, easy-to-use, cloud-powered business analytics service that makes it easy for all employees within an organization to build visualizations, perform ad-hoc analysis, and quickly get business insights from their data, anytime, on any device.
- Amazon QuickSight allows everyone in your organization to understand your data by asking questions in natural language, exploring through interactive dashboards, or automatically looking for patterns and outliers powered by machine learning.



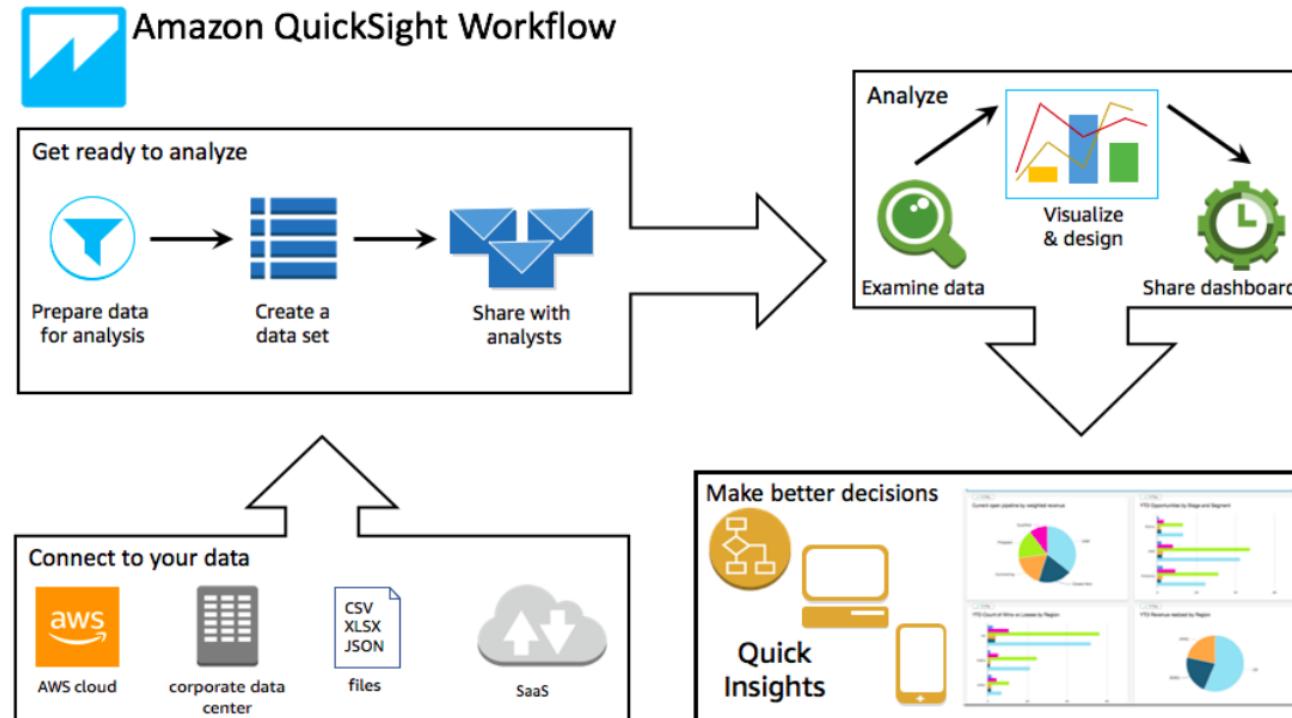
# Features of QuickSight

- **Enable BI for everyone with QuickSight Q** - Ask conversational questions of your data and use Q's ML-powered engine to receive relevant visualizations without the time consuming data preparation from authors and admins.
- **Perform advanced analytics with ML insights** - Discover hidden insights from your data, perform accurate forecasting and what-if analysis, or add easy-to-understand natural language narratives to dashboards by leveraging AWS' expertise in machine learning.
- **Embed analytics to differentiate your applications** - Easily embed interactive visualizations and dashboards, sophisticated dashboard authoring, or natural language query capabilities in your applications to differentiate user experience and unlock new monetization opportunities.

# Benefits of QuickSight



# Workflow of QuickSight

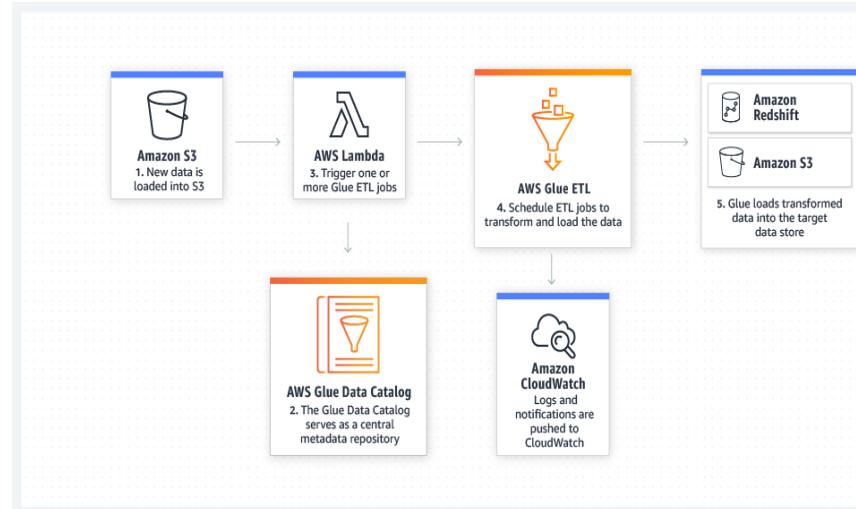




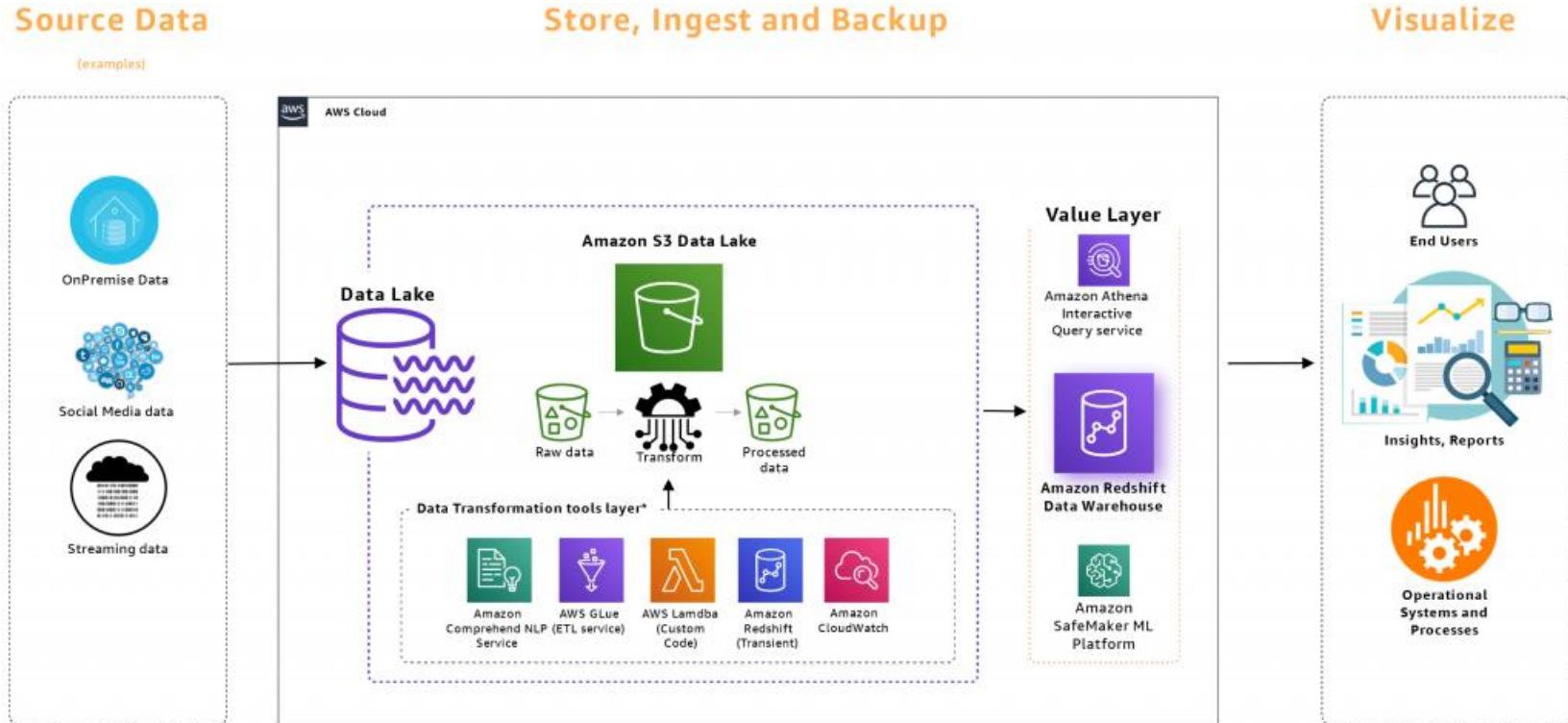
# Analytics – AWS Glue

# AWS Glue

- AWS Glue is a serverless data integration service that makes it easy to discover, prepare, and combine data for analytics, machine learning, and application development.
- AWS Glue provides all the capabilities needed for data integration so that you can start analyzing your data and putting it to use in minutes instead of months.



# AWS Glue Architecture



# Benefits of AWS Glue

- **Faster data integration** - Different groups across your organization can use AWS Glue to work together on data integration tasks, including extraction, cleaning, normalization, combining, loading, and running scalable ETL workflows.
- **Automate your data integration at scale** - AWS Glue automates much of the effort required for data integration. AWS Glue crawls your data sources, identifies data formats, and suggests schemas to store your data.
- **No servers to manage** - AWS Glue runs in a serverless environment. There is no infrastructure to manage, and AWS Glue provisions, configures, and scales the resources required to run your data integration jobs.

# Lab Exercises

## Activity Guide (Lab) : Creation Of Dynamo DB

- Create Dynamo DB
- CRUD Operation In Dynamo DB



### Table of Contents

1	Introduction .....	3
2	Documentation Links.....	4
3	Create a DynamoDB.....	5
3.1	CRUD Operation on DynamoDB.....	7
4	Summary.....	11

my

# Lab Exercises

## Activity Guide (Lab) :Configuring a MySQL DB Instance via Relational Database Service (RDS)

- Downloading MySQL Workbench
- Configuration of AWS RDS Instance
- Connecting to MySQL Database from Workbench
- Deleting The RDS Instance

### Contents

1	Introduction .....	3
2	Documentation Links.....	4
3	Pre-Requisites .....	5
3.1	Downloading MySQL Workbench .....	5
3.1.1	Downloading .Net Framework.....	7
3.1.2	Downloading Visual C++ Redistributable for Visual Studio 2019 .....	8
3.2	Installing MySQL Workbench .....	10
4	Configuration of AWS RDS Instance.....	14
5	Connecting to MYSQL Database from Workbench.....	19
6	Deleting the RDS Instance .....	26
7	Troubleshooting .....	29
7.1	Issue .....	29
7.2	Fix.....	29
8	Summary .....	32

# Lab Exercises

## Activity Guide (Lab) :Creating A Redis Cache And Connecting It To EC2 Instance

- Creating a Redis Cache and Connecting it with EC2 Instance
- Testing the Connection by running Redis Commands
- Deleting the Redis Cache



### Contents

1	Introduction .....	3
2	Documentation Links.....	4
3	Creating a Redis Cache And Connecting It With EC2 Instance .....	5
3.1	Creating a Linux EC2 Instance .....	5
3.2	Creating a Redis Cache .....	13
3.3	Connecting Redis With Linux EC2 Instance .....	20
3.4	Testing the Connection by Running Redis Commands .....	24
4	Delete/Cleanup .....	26
4.1	Terminating the Linux EC2 Instance .....	26
4.2	Deleting the Redis Cache .....	28
5	Summary.....	31

# Lab Exercises

## Activity Guide (Lab) :Kinesis Data Stream Creation

- Visualize Web Traffic Using Kinesis Data Stream
- Monitoring data by producer and consumer through AWS Kinesis Dashboard
- Deleting Resources

### Table of Contents

1	Introduction .....	3
2	Documentation Links.....	4
3	Visualize Web Traffic Using Kinesis Data Streams .....	5
4	Monitor the Data Generated by Producer and Consumer through AWS Kinesis Dashboard.....	11
5	Deletion of Resources .....	13
5.1	Deleting EC2 Instance.....	13
5.2	Deleting CloudFormation Stack.....	13
6	Summary.....	15

# Lab Exercises

## Activity Guide (Lab) :Amazon Athena

- Analyze CSV Data in S3 with Athena
- Deleting Resources (deleting Table)



### Table of Contents

1	Introduction.....	3
2	Documentation Links.....	4
3	Analyze CSV Data in S3 with Athena .....	5
4	Deleting The Resources.....	11
4.1	Deleting the Table.....	11
5	Summary.....	12

## Certification Sample Quiz : Database Server And Analytics

September 3, 2020 by [REDACTED] [\(Edit\)](#) [Edit with WPBakery Page Builder](#)

Q1. Which of the following is a caching engine?

- A. ElastiCache
- B. DynamoDB
- C. Memcached
- D. IAM

Check

# Quiz Question

Which of the following is true about RDS? (Choose two.)

- A. Reserved instances can be used for Multi-AZ deployments.
- B. Automated backups are turned off by default.
- C. Every database supported by RDS can also be installed directly on EC2 instances.
- D. All RDS databases support SQL as an interface.

**Correct Answer:** A, D

**Explanation:** RDS supports Multi-AZ deployments. Automated backups are turned *on* by default. Some RDS databases—notably Maria and Aurora—are only supported through a managed service like RDS. And all RDS databases provide a SQL interface.

# Quiz Question

You are working as a Solutions Architect for a technology company which is in the process Of migrating their applications to AWS. One Of their systems requires a database that can scale globally and can handle frequent schema changes. The application should not have any downtime or performance issues whenever there is a schema change in the database. It should also provide low-latency response to high-traffic queries.

Which is the most suitable database solution to use to achieve this requirement?

- A. An Amazon RDS instance in Multi-AZ Deployments configuration
- B. Amazon DynamoDB
- C. Redshift
- D. An Amazon Aurora database with Read Replicas

# Quiz Question

**Correct Answer: B**

**Explanation:**

For DynamoDB, it scales well due to these reasons:

- Its schema flexibility lets DynamoDB store complex hierarchical data within a single item. DynamoDB is not a totally schemaless database since the very definition of a schema is just the model or structure of your data.
- Composite key design lets it store related items close together on the same table.

# Find Us



<https://www.facebook.com/K21Academy>



<http://twitter.com/k21Academy>



<https://www.linkedin.com/company/k21academy>



<https://www.youtube.com/k21academy>



<https://www.instagram.com/k21academy>