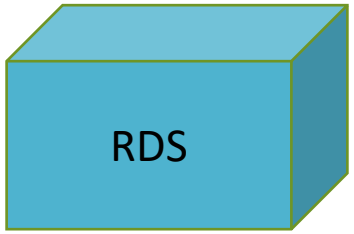
AWS Database

Analytics Tensor
Mahesh KC
mahesh.kc@analyticstensor.com
https://analyticstensor.com

Overview of AWS Database

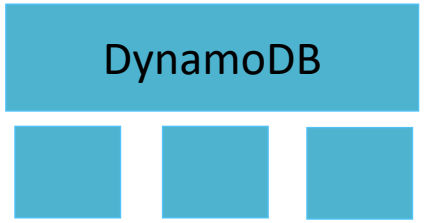


Amazon Relational Database Service

- MySQL
- SQL Server
- PostgreSQL
- MariaDB
- Oracle
- Amazon Aurora

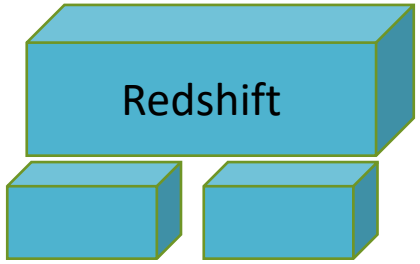
Amazon ElastiCache

- Memcached
- Redis
- Clustered
- Automated maintenance



Amazon DynamoDB

- NoSQL
- Highly-scalable
- Fault-tolerant
- Replicated
- Event-drives

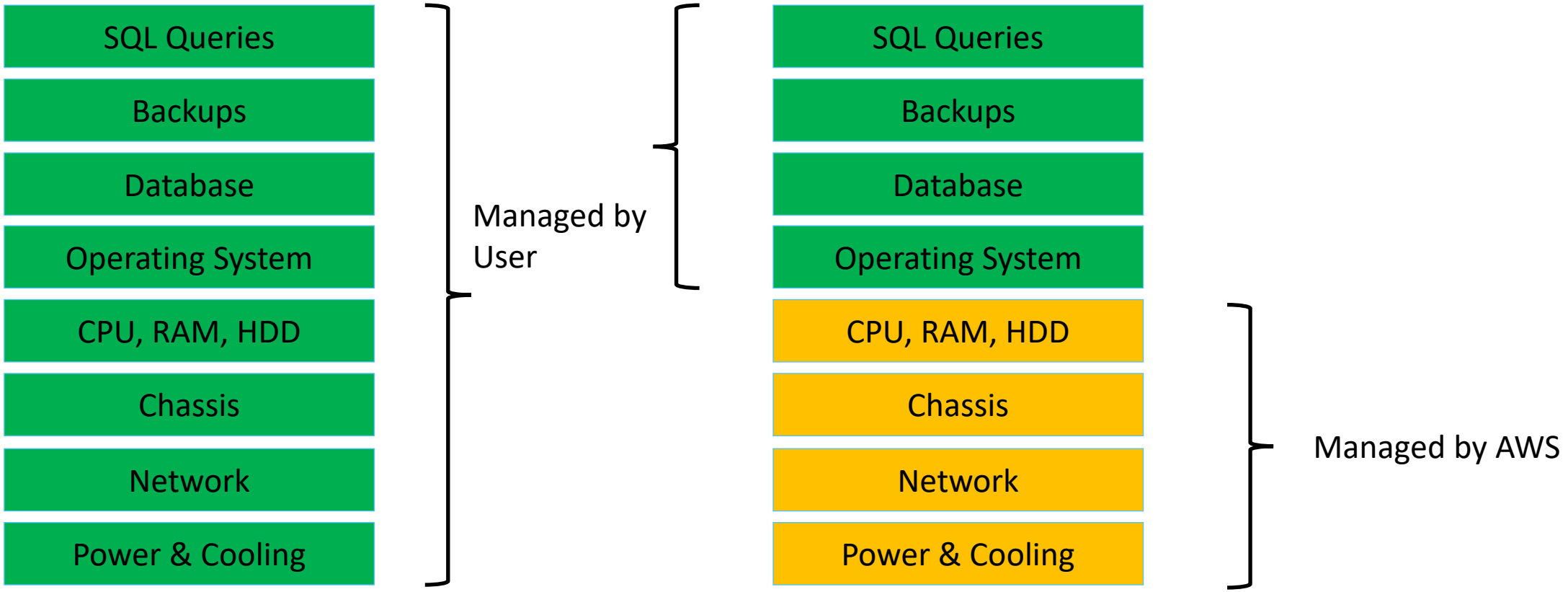


Amazon Redshift

- SQL Compliant
- Petabyte-scale
- Fault-tolerant
- Replicate
- Fork of PostgreSQL
- Encryption
- Used for Datawarehouse

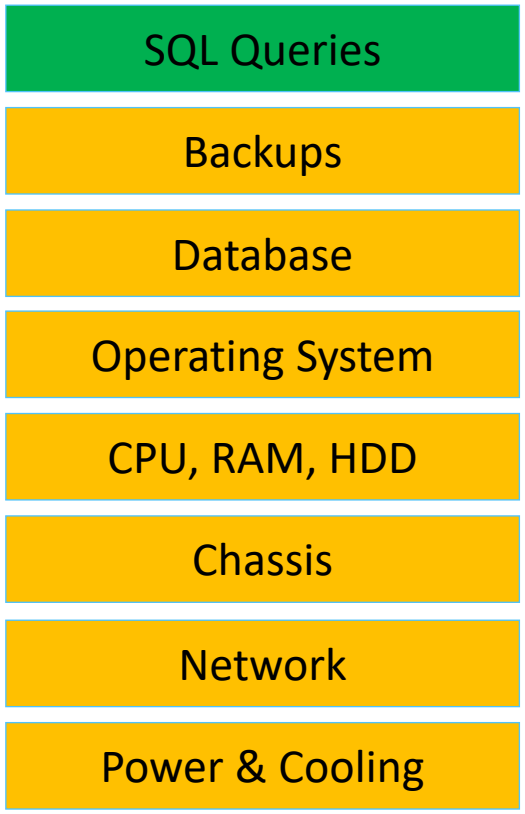
Analytics Tensor

Managing a Relational Database



Analytics Tensor

Managing a Relational Database (cont.)

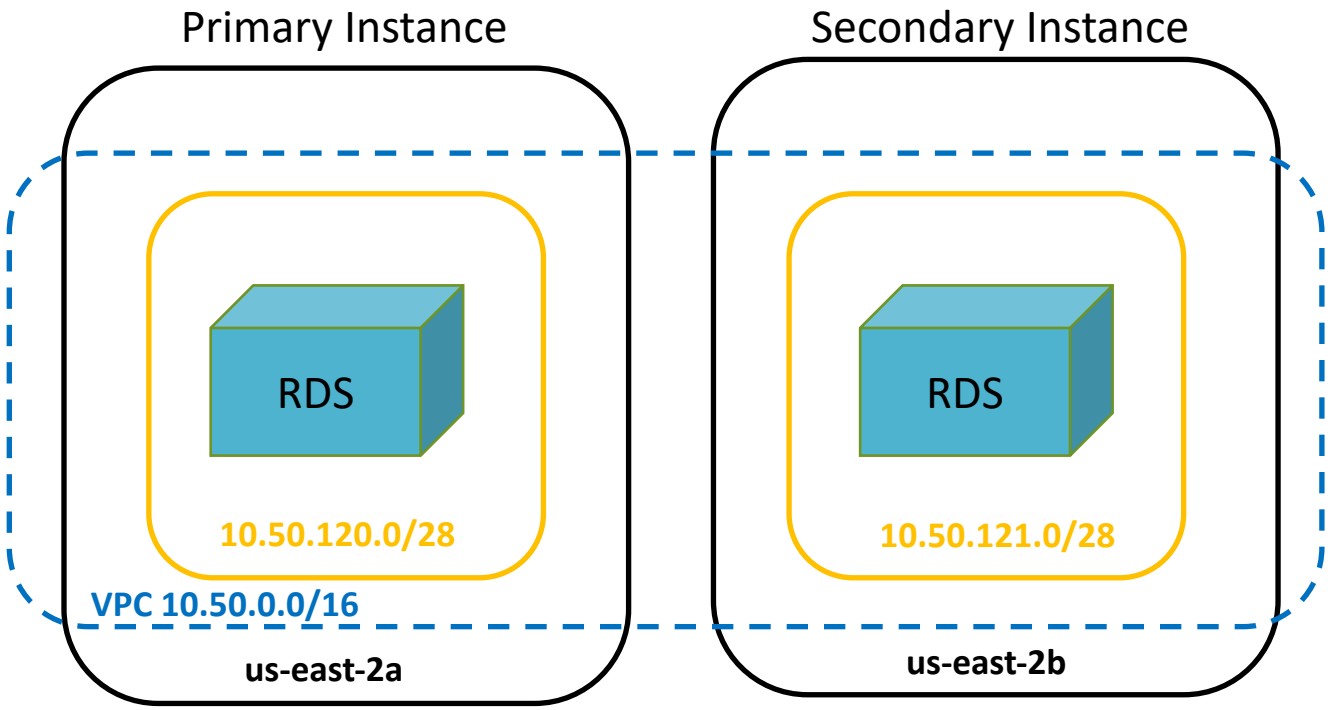


When using on RDS

- Reduced operational burden
- Focus on application
- Data durability is built-in
- Easy geographical diversity
- Choice of Databases:-
  - MySQL
  - Amazon Aurora (MySQL compatible)
  - SQL Server
  - PostgreSQL
  - MariaDB
- Read Replicas
- Automated:-
  - Backups
  - Patches

Analytics Tensor

Data Durability with Amazon RDS



Choosing Multi-AZ Deployments provides:

- High Availability
- Lower RTO/RPO in minutes
- Synchronous replication
- Physically distinct
- Automatic Failover
  - Loss of Network
  - Compute Failure
  - Storage Failure
- It is best practice for production environment
- We can set-up automated backup of RDS.

While launching RDS instance we can optionally choose Multi-AZ Deployments. By doing this, Relational Database Service (RDS) will place one primary instance in one-subnet (i.e. in 1 AZ) and other secondary instance (standby) in other subnet (i.e. other AZ). This gives resilient not only to the loss of devices but also the loss of entire Data Center. If primary fails either by network issue, hardware, AZ is unreachable etc. then we can failover to the secondary standby in a minutes. We can do this by DNS entry. The RDS can update the ip address for the DNS entry quickly.

AT Analytics Tensor

Automated Backups of RDS

- We can setup automated backup of RDS on certain window period either daily or hourly.
- The snapshot will be stored in S3.
- We can easily restore.
- We can do point-in-time restore from binary logs.
- We can easily backup to other geographical region and re-create the database, using that RDS instance from backups.
- All the ability is included in the cost of machine.

Demo: Launch Amazon RDS Instance

- From Console, Choose RDS.

Prerequisites: Create subnet group. (https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/SubnetGroups.html). For this tutorial, we can choose default VPC.

  - Choose Subnet Groups-> Create subnet group. Choose VPC ID. Choose AZ. Choose Subnet ID. Name subnet group as “RDS-Test.”
- Choose Create Database.
- Choose Standard Create.
- Choose Amazon Aurora. Choose MySQL compatibility.
- Choose Regional
- Choose Dev/Test
- Type “Yourname-Aurora-db”. Must be unique.
- User: admin, Password: MyAuroraDB202004!
- DB instance: Choose db.r4.large
- Don’t create an Multi-AZ
- Choose Create Database.
- Default port is 3306
- Once the instance is created, We can see instance information. It will give cluster endpoint, which we need use to connect to Aurora RDS instance.

AT Analytics Tensor

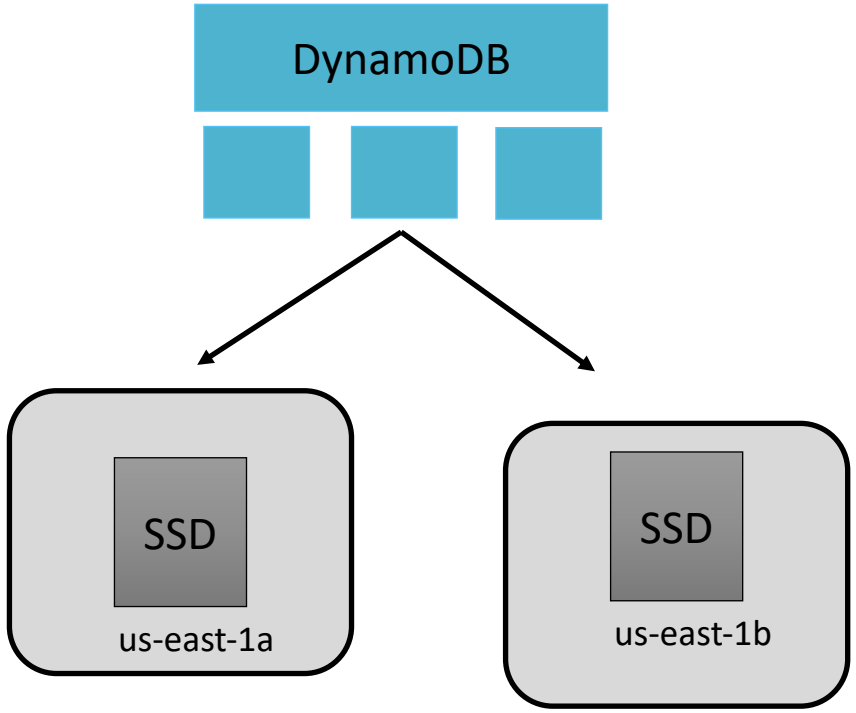
Connect to Amazon RDS Instance

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ConnectToInstance.html
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Connecting.html

https://datascience-enthusiast.com/R/AWS_RDS_R_Python.html

Amazon Dynamo DB

- It is NoSQL Data Store
- It is more than fully managed like RDS which is highly available, fault tolerant and durable to very high degree.
- All the data will be stored in SSD volume.
- Data will be replicated over multiple AZs in a particular region.
- It is designed to receive single digit millisecond response time.
- We can provide both read and write throughput ***anytime*** when we provisioned Amazon DynamoDB.

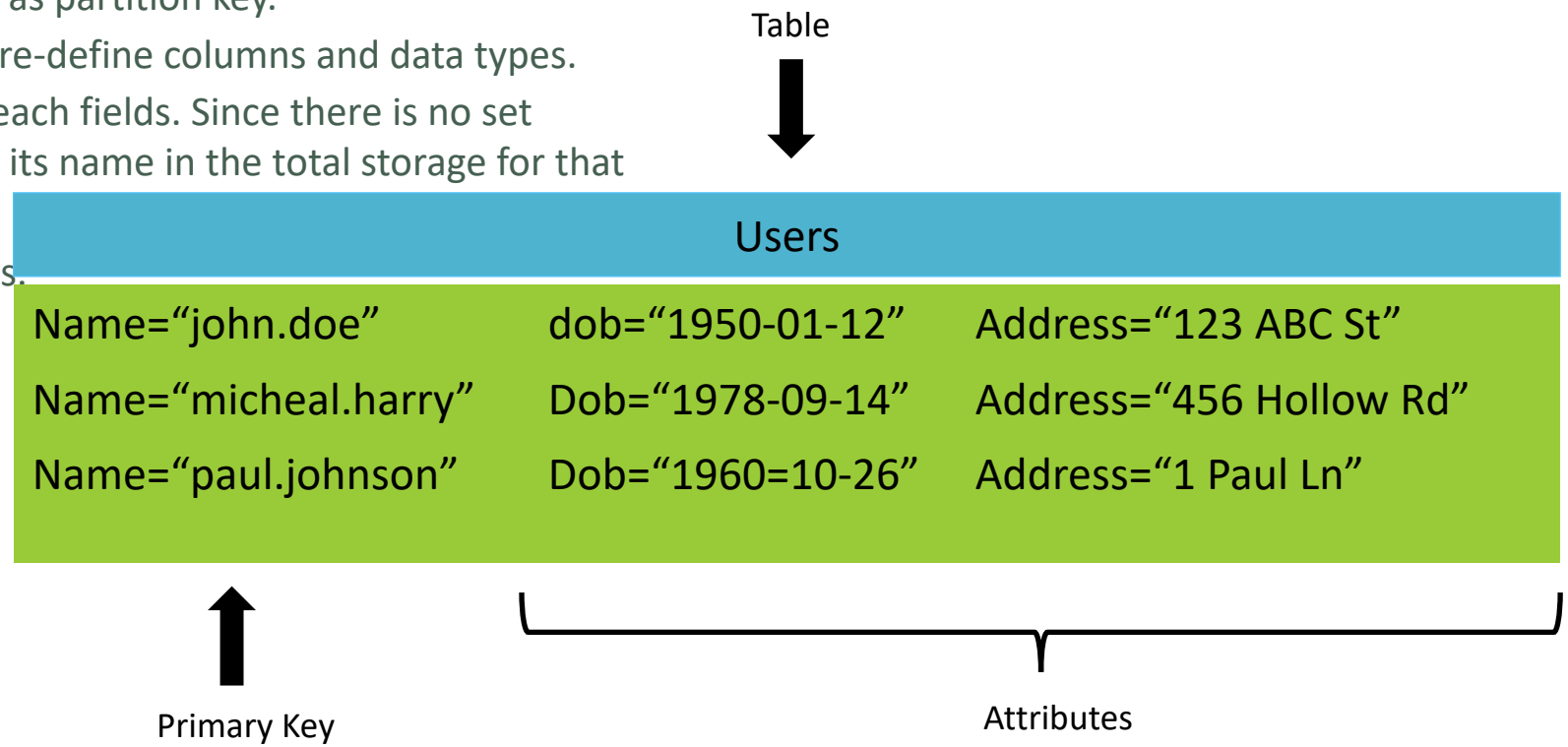


Analytics Tensor

Amazon DynamoDB Data Models

- There is no joins/relationships.
- **Tables** is made up of items. An **items** are made of up an **attributes**. Each items has primary key which is unique. The other columns are called attributes. Primary key is also known as partition key.
- It is Schema-less. We don't have to pre-define columns and data types.
- The column name is included inside each fields. Since there is no set schema, each attributes will includes its name in the total storage for that particular item.
- We can also create secondary indexes.
- Table size doesn't have limit.
- It can be GB, TB, PB etc.
- There is limit in particular item.
- 1 items can't be more than 400 KB

For e.g name shouldn't exceed 400KB.



Amazon DynamoDB Primary Keys

Users		
Name="john.doe"	dob="1950-01-12"	Address="123 ABC St"
Name="micheal.harry"	Dob="1978-09-14"	Address="456 Hollow Rd"
Name="paul.johnson"	Dob="1960=10-26"	Address="1 Paul Ln"

Primary Key as partition key

John Doe has two relationship

Relationship		
Name="john.doe"	Friend="bob.harry"	Relation="Friend"
Name="jane.doe"	Friend="jim.doe"	Relation="Brother"
Name="paul.johnson"	Friend="joe.patt"	Relation="Friend"

Primary key as "composite" partition + sort key

AT Analytics Tensor

Demo: Create table in Amazon DynamoDB

- Choose DynamoDB from Services.
- Choose Create Table.
- Table Name: employees. Primary key: id; Sort key: email
- Provisioned Capacity: Choose 5 for both read and write
- Choose Create.

Amazon ElastiCache

- It is a fully managed service similar to RDS.
- It has automated minor patch updates.
- It also supports clustering.
- It supports two choices:
  1. Redis: It is an in-memory databases that supports multiple data types. It is also backed to disk, so that we have ability to stop and recover if there is any failure.
  2. Memcache: It is an in-memory cache, simple key/value store. It is not backed to disk. We can't recover.

Setup Redis Cluster: https://aws.amazon.com/getting-started/hands-on/setting-up-a-redis-cluster-with-amazon-elasticache/

Redis Vs Memcache: https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/SelectEngine.html

Videos: https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Tutorials.html#WhatIs.Videos.Beginning

Amazon Redshift

- It is a fully managed service similar to RDS.
- It is a petabyte scale data warehouse.
- It is based on fork of Postgres 8.0.2.
- It is SQL compliant.
- We can connect with JDBC/ODBC drivers.
- Redshift is a clustered services, so it has multiple node. All the queries run in parallel.
- It is ideal for OLAP and BI applications.
- It is developed by Amazon.

Analytics Tensor

Use Case: Databases in E-commerce