**AWS RELATIONAL DATABASE SERVICES RDS**

**OVERVIEW IN DATABASE IN GENERAL**

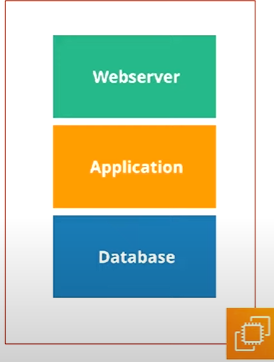
Database

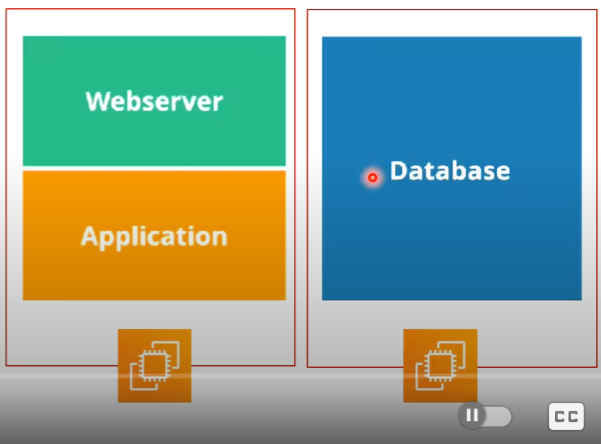
* System that stores and manage data
* They are divided in to two types

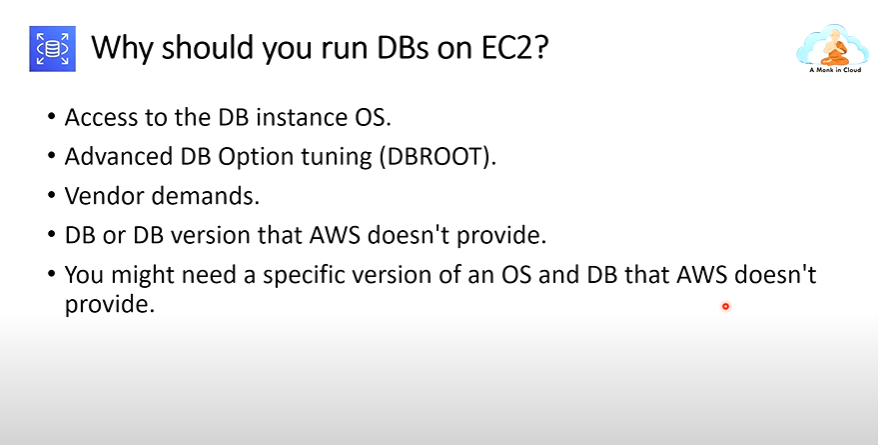
1. Relation (SQL) is a structured query language
2. Non-relational (NoSQL) non-structured query language

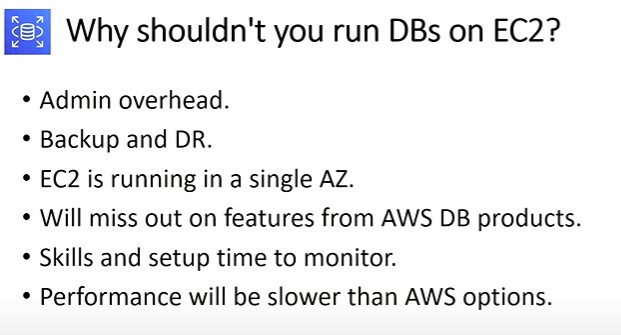
**Databases on ec2 instances**

Data connection in the same instances

****

Database connection in two different instances with different available zone 





* Select the standard create
* Select the free tire account

Settings

* Database name
* Master user name and master password (manual)

Instances configuration

* T2.micro

Storage

* Gp2
* Allocated Storage

enable storage auto-scaling can

Connectivity

* connect to ec2 computer resources
* Ec2 instances
* DB subnet group = default
* Public access
* Yes = if select yes then we can access the with public access
* No= if we select no the we can not access the data base

VPC SECURITY GROUP

* Security Group = default one
* Available zone
* 3306 is the port number for my sequel database

Database authentication

* Password authentication

**Monitoring**

* How to check the aws is working

**Additional configuration**

* Initial database name
* Db parameter group
* Option group

**Backup**

* Enable automated backups (it is project we are not select this )
* It will store the backup 7 storage and store in the snapshot storage

**Encryption**

* It is charge amount

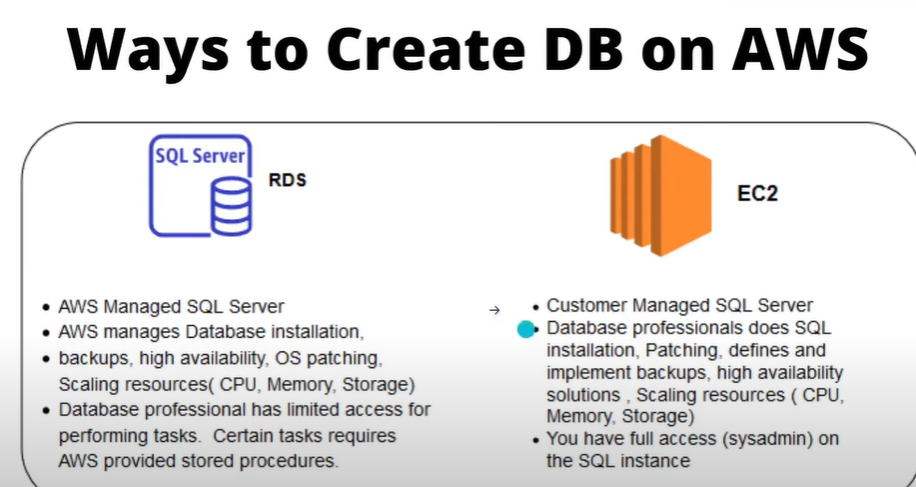
**Log exports**

* Is data stored in the cloud watch

**Maintenance windows**

* NO Preferences

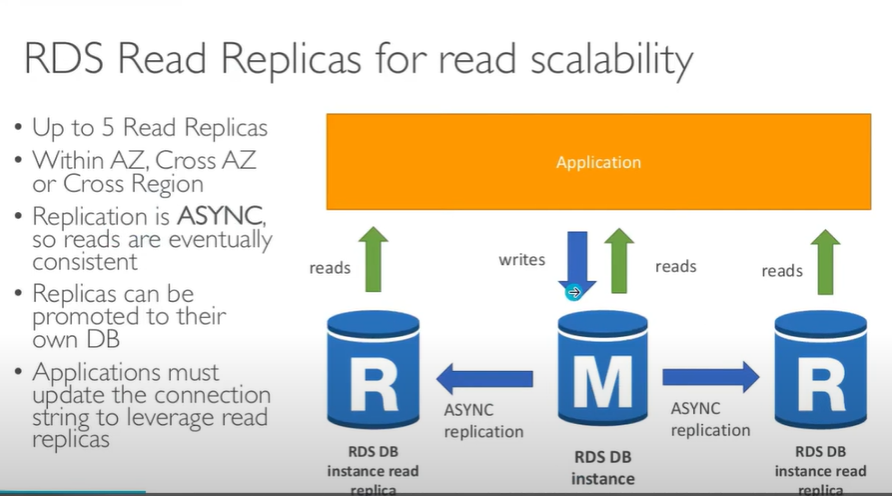
**Deletion protection**

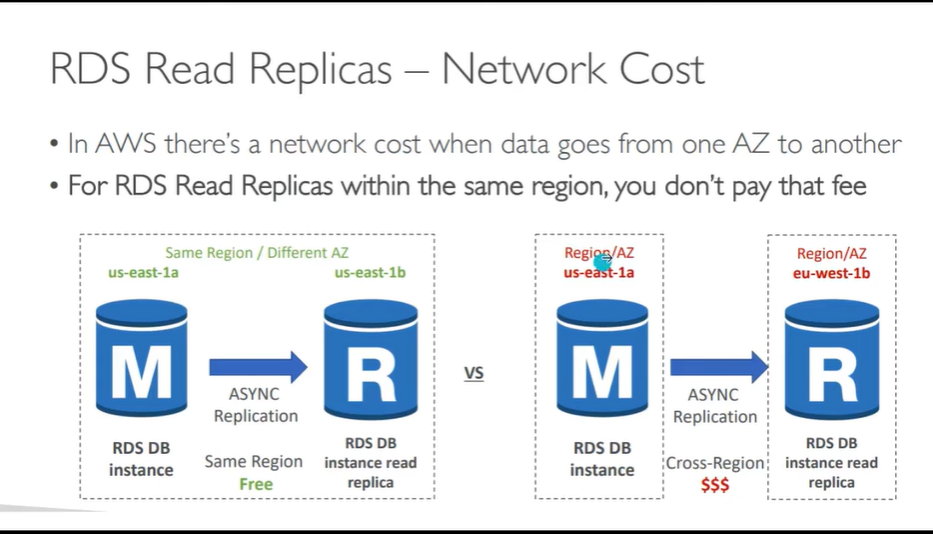
****

in ec2 instances, we can connect them with the database and access the data and go in to the server but in the RDS we connect with SSH

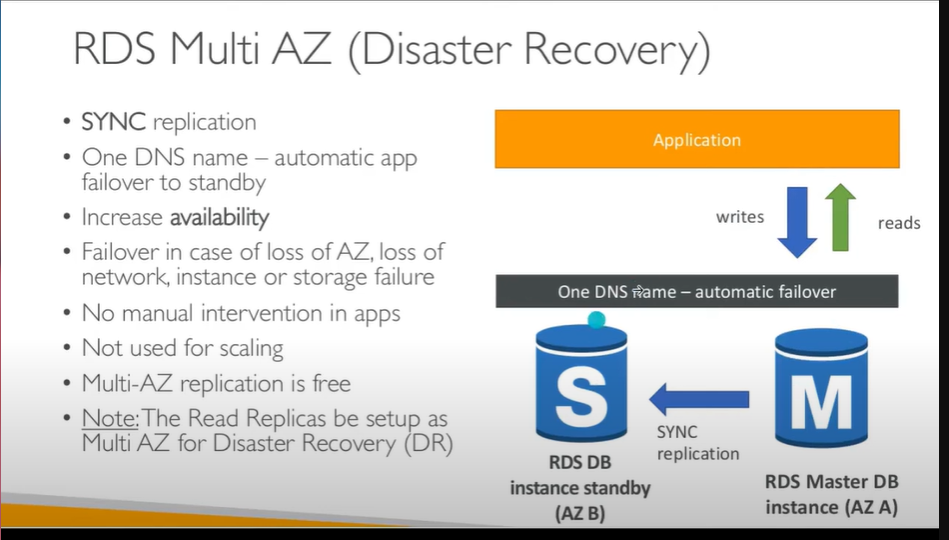
**RDS Read Replicas for read scalability**

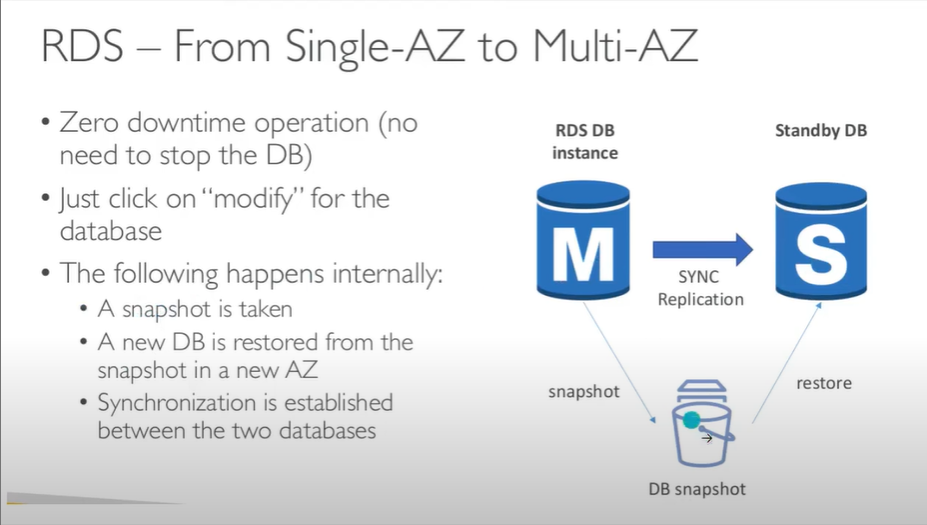
* It used to increase the performances

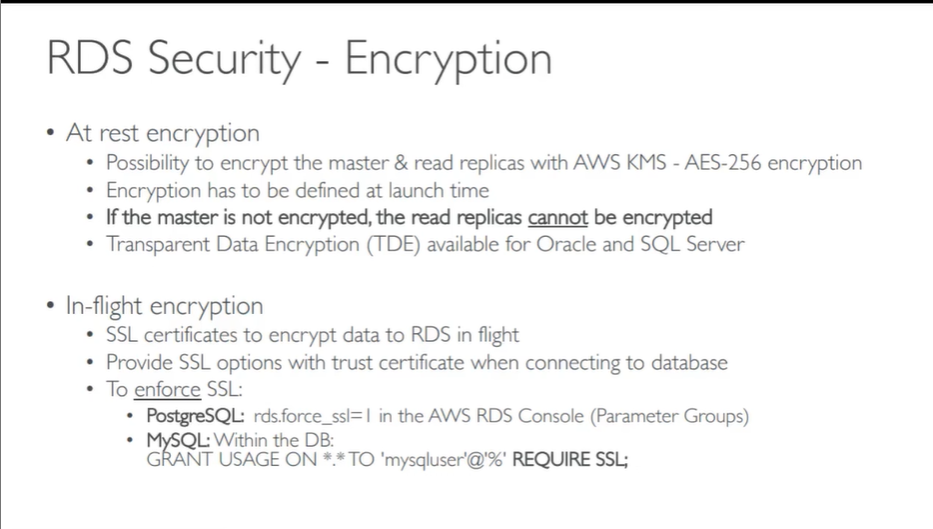




**RDS multi-AZ ( disaster recovery )**







**Amazon Aurora: A Powerful Database Service**

* **High Performance:** Aurora is significantly faster than traditional databases, especially for read-heavy workloads.
* **Scalability:** It can easily scale up or down to handle changing workloads.
* **Reliability:** It automatically replicates data across multiple Availability Zones (AZs) for high availability.
* **Security:** It offers robust security features to protect your data.
* **Cost-Effective:** It's a cost-efficient option for various workloads, especially when considering its performance and scalability

**DynamoDB: A Powerful NoSQL Database**

* Fully Managed Highly available with replication across 3 AZ
* NoSQL database - not a relational database
* Scales to massive workloads, distributed "serverless" database
* Millions of requests per second, trillions of rows, 100s of TB of storage
* Single-digit millisecond latency - low latency retrieval
* Integrated with IAM for security, authorization and administration
* Low cost and auto scaling capabilities

DynamoDM ACCELERATOR -DAX

* MANAGED IN **MEMORY CACHE**
* 10XPROFORMANCES improvement
* DAX IS ONLY USED FOR AND IS INTEGRATED WITH Dynamodb

**Redshift overview**

It is not used for OLTP

Online analytical processing

Load data once every house

Columnar storage of data

Massively parallel query execution

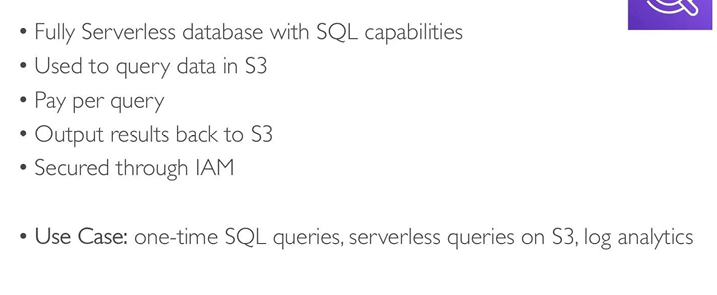
Bi too;s such as AWS quick sight

**Amazon EMR**

* Elastic map reduce
* Hadoop cluter (big date )
* The cluster can be mand of hunders of ec2 instances
* User

1. **Data processing**
2. **Maching learning**

**Athena overview**



**Amazon Quicksight**