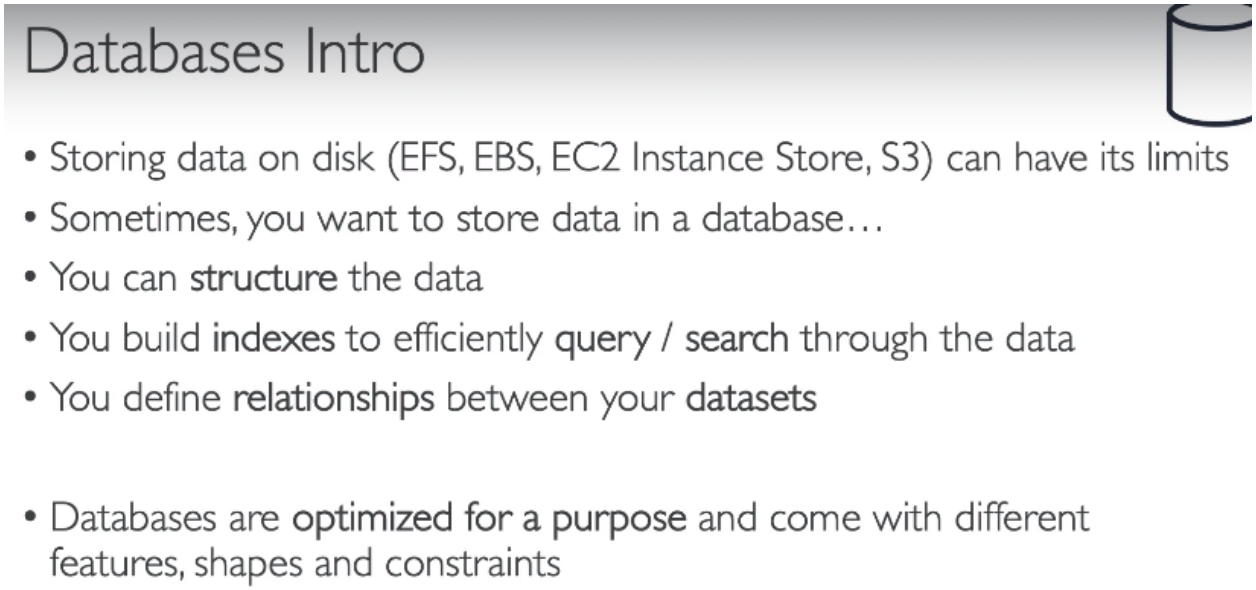
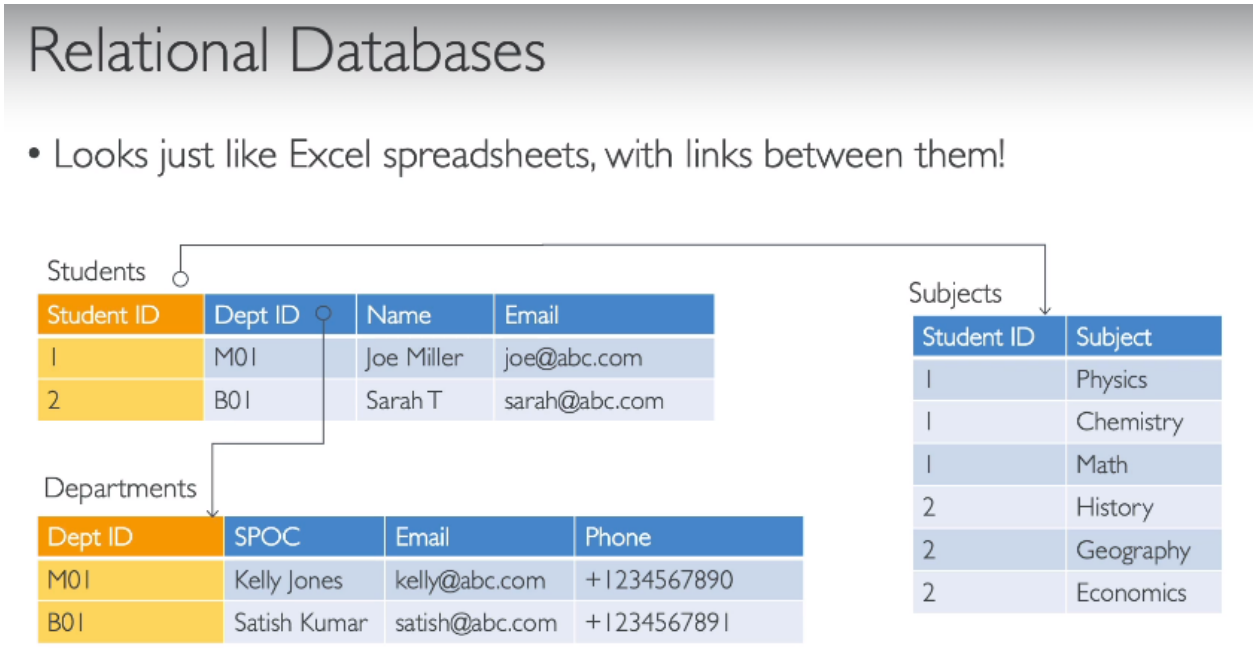
DataBases Intro

1. So when you are storing data on disk, would it be on an EBS drive, an EBS volume, an EC2 Instance Store, Amazon S3, you have your limits.
2. With EFS,EBS,EC2 and S3 we can only do per files operations..but with db..it will be more structured



3.
4. Relational DB



5.
6. In here we can define relationships between two tables..here we can see students table has relationship with departments and subjects

7. Relations can be defined by primary key,foreign key etc
8. • Can use the SQL language to perform queries / lookups
9. Whenever you see SQL..think of relational databases

NoSQL Databases

- NoSQL = non-SQL = non relational databases
- NoSQL databases are purpose built for specific data models and have flexible schemas for building modern applications.
- Benefits:
  - Flexibility: easy to evolve data model
  - Scalability: designed to scale-out by using distributed clusters
  - High-performance: optimized for a specific data model
  - Highly functional: types optimized for the data model
- Examples: Key-value, document, graph, in-memory, search databases

10.
11. For example in nosql..we can data in the format of JSON

NoSQL data example: JSON

- JSON = JavaScript Object Notation
- JSON is a common form of data that fits into a NoSQL model
- Data can be nested
- Fields can change over time
- Support for new types: arrays, etc…

```
{
    "name": "John",
    "age": 30,
    "cars": [
        "Ford",
        "BMW",
        "Fiat"
    ],
    "address": {
        "type": "house",
        "number": 23,
        "street": "Dream Road"
    }
}
```

12.
13. DB and Shared Responsibility on AWS

Databases & Shared Responsibility on AWS

- AWS offers use to **manage** different databases
- **Benefits** include:
  - Quick Provisioning, High Availability, Vertical and Horizontal Scaling
  - Automated Backup & Restore, Operations, Upgrades
  - Operating System Patching is handled by AWS
  - Monitoring, alerting

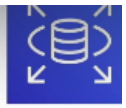
- Note: many databases technologies could be run on EC2, but you must handle yourself the resiliency, backup, patching, high availability, fault tolerance, scaling…

14.
15. If we use our own db on an ec2 instance…we are responsible for everything
16. If we use managed db provided by aws…it includes all the benefits..see pic

AWS RDS Overview

1. RDS stands for relational Database services
2. The databases can be different kinds like postgres…etc

AWS RDS Overview



- RDS stands for **Relational** Database Service
- It's a managed DB service for DB use **SQL** as a query language.
- It allows you to create databases in the cloud that are managed by AWS
  - Postgres
  - MySQL
  - MariaDB
  - Oracle
  - Microsoft SQL Server
  - Aurora (AWS Proprietary database)

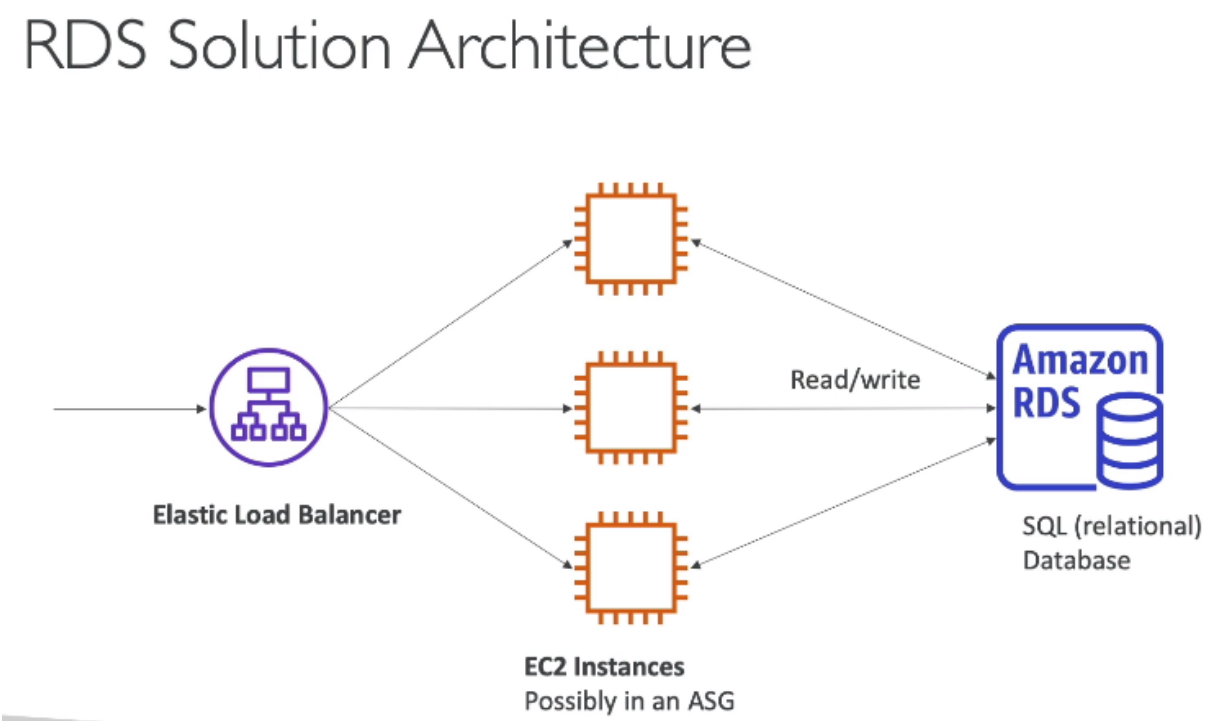
3.
4. Advantages

Advantage over using RDS versus deploying DB on EC2

- RDS is a managed service:
  - Automated provisioning, OS patching
  - Continuous backups and restore to specific timestamp (Point in Time Restore)!
  - Monitoring dashboards
  - Read replicas for improved read performance
  - Multi AZ setup for DR (Disaster Recovery)
  - Maintenance windows for upgrades
  - Scaling capability (vertical and horizontal)
  - Storage backed by EBS (gp2 or io1)
- BUT you can't SSH into your instances

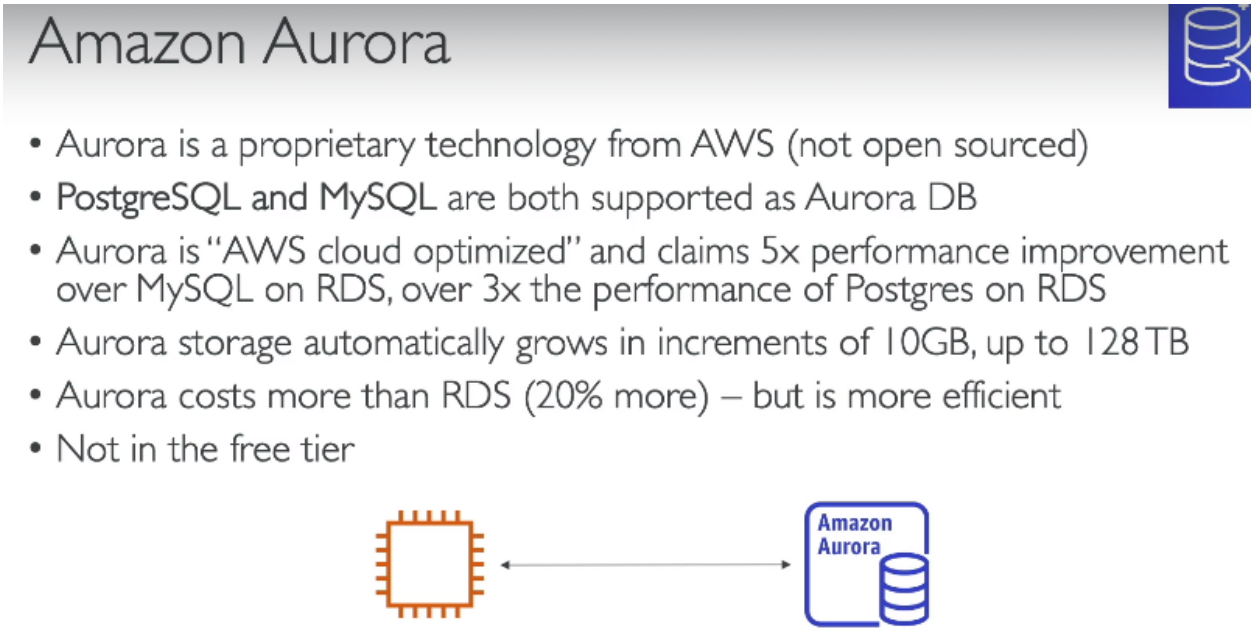
5.

6. System provisioning is the process of setting up infrastructure in the cloud, including establishing user, system, and service access to the applications, data, and cloud resources. The benefits of automatic system provisioning are: Reduced human errors during the deployment.
7. But we cannot SSH into our db instance…like we cannot see whats going on with our database
8. RDS Solution Architecture



9.

10. Here basically our ELB will load the requests into our ec2 instances and ec2 instances will be connected to database and doing read/write all at once
11. Amazon Aurora



12.

13. Amazon aurora in not in the free tier..wer as RDS is in the free tier

14. So from an exam perspective,
RDS and Aurora are going to be the two ways
for you to create relational databases on AWS.
They're both managed
and Aurora is going to be more cloud native
whereas the RDS is going to be running the technologies,
you know, directly as a managed service.

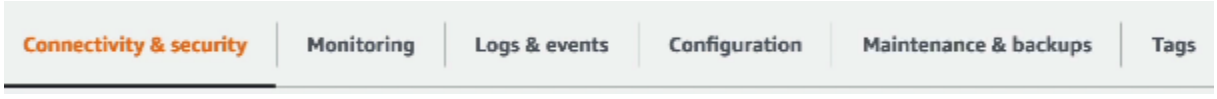
AWS RDS Hands On:

1. Here we are going to create our first database in the RDS
2. First go to RDS service and click on databases..then click on create DB
3. Here we choose standard and MySqL using RDS
4. In MySQL
   - Supports database size up to 64 TiB.
   - Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
   - Supports automated backup and point-in-time recovery.
   - Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.
5.

6. A read replica in AWS is a copy of a database instance that is used to improve performance and availability. Read replicas are asynchronous, which means that they do not reflect every change made to the primary instance immediately. However, they are typically updated within seconds or minutes.

   Read replicas can be used to improve the performance of database applications by offloading read queries from the primary instance to the read replicas. This can be especially helpful for applications that experience a high volume of read queries.

   Read replicas can also be used to improve the availability of database applications by providing a failover mechanism. If the primary instance fails, one of the read replicas can be promoted to become the new primary instance. This ensures that the database application remains available even in the event of a failure.
7. Here we will choose the default version of MySql and we choose free tier
8. We'll be identifying our db as Database-1
9. 

10. Then we have setup credentials settings by entering password
11. Next we have to choose an instance class..here burstable is used for free tier ..
12. Then we can see storage settings to allocate storage and to enable autoscaling
13. Next we move to connectivity settings ..here we need to choose a cloud for database
14. We need to give it public access…so to access outside of our vpc
15. Then we need to choose a security grp for vpc
16. Next we choose authentication for our DB…we choose simple password authentication
17. At last we can see estimated monthly costs..next we will create db..it will take few minutes to create
18. Next if we go to summary of our database..which check its settings such as
19. 

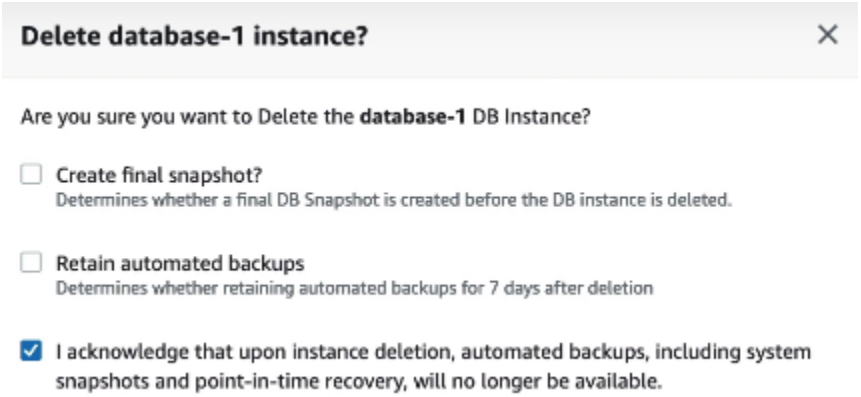
20. If we want to connect to our database …we have to use end point and the port of our db
21. If we go to security grps..we can see inbound rules and outbound rules
22. What are inbound and outbound rules in rds aws?

23. Sure. Inbound and outbound rules in AWS RDS control the traffic that is allowed to reach and leave your database instance. Inbound rules allow traffic to reach your database instance from specific sources, while outbound rules allow your database instance to connect to specific destinations.

    Inbound rules are used to control who can connect to your database instance. For example, you might create an inbound rule that allows only traffic from your company's internal network to reach your database instance. This would prevent unauthorized users from connecting to your database instance from the internet.

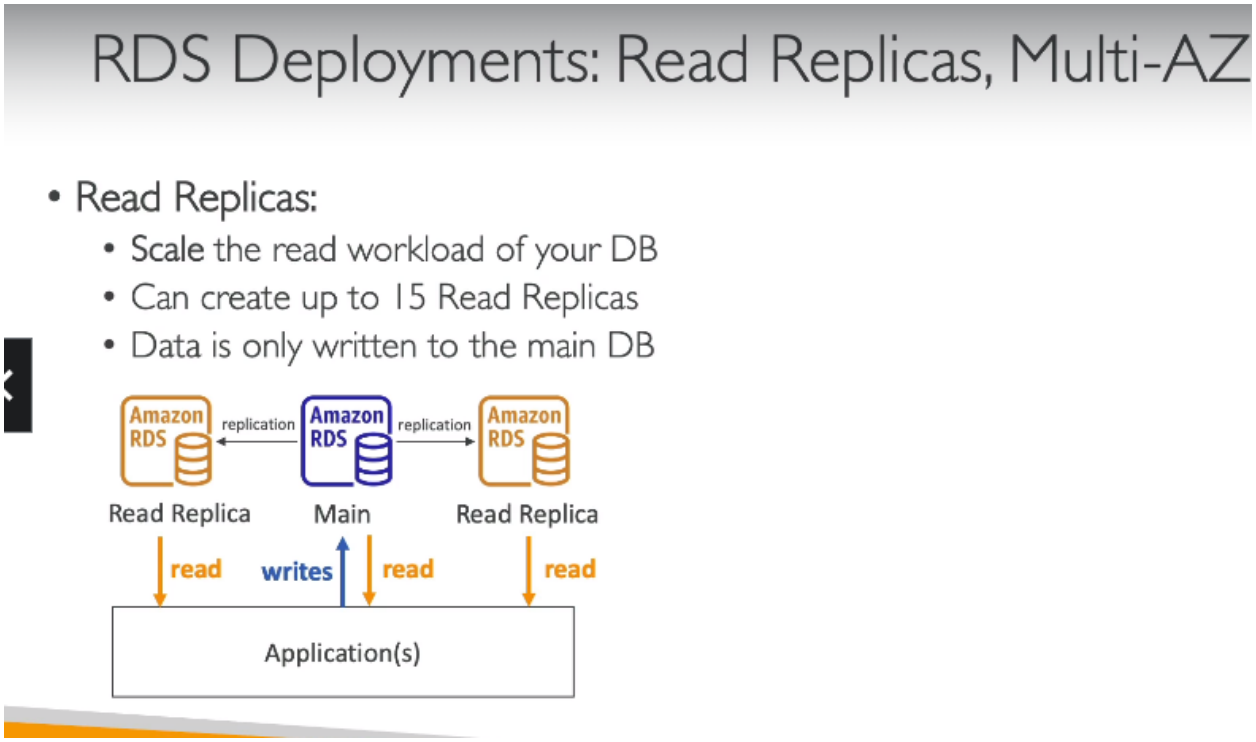
    Outbound rules are used to control where your database instance can connect to. For example, you might create an outbound rule that allows your database instance to connect to the internet to download software updates. However, you might also want to create an outbound rule that prevents your database instance from connecting to certain websites or IP addresses. This can help to protect your database instance from malicious attacks.
24. So, as we can see, having a database in the cloud is really helpful when it's managed because, well, we have all the nice monitoring, configuration, settings, and so on, associated with it,so this is a huge advantage.
25. If we go to actions we can also create a snapshot of our database
26. Difference bw read replicas and snapshots
27. Sure. Here are the key differences between read replicas and snapshots in AWS RDS:

    - **Purpose:** Read replicas are used to improve the performance and availability of database applications. Snapshots are used to create a backup of a database instance.
    - **Data consistency:** Read replicas are always up-to-date with the primary instance. Snapshots are point-in-time copies of a database instance.
    - **Replication:** Read replicas are asynchronously replicated from the primary instance. Snapshots are created manually or automatically.
    - **Cost:** Read replicas are charged for the amount of storage that they use. Snapshots are charged for the amount of storage that they use and the amount of time that they are retained.
28. After creating the snapshot..if we go to the actions..we have an option to restore snapshot

29. It helps us to create new db out of this snapshot And the reason why I would do so is that, for example,
30. you wanted to create a bigger database or create a copy of the database and so on, or create a different settings for your database.
31. We can copy the snapshot to different region
32. Other option is we can share our snapshot..so other users can create a db using our snapshot
33. We can see more options on snapshot action .
34. So you can see, Managed Services really have a good impact.on how you use the cloud, and they really speed it up for you to make sure that you don't manage too much infrastructure.
35. Upon deletion…
36. 

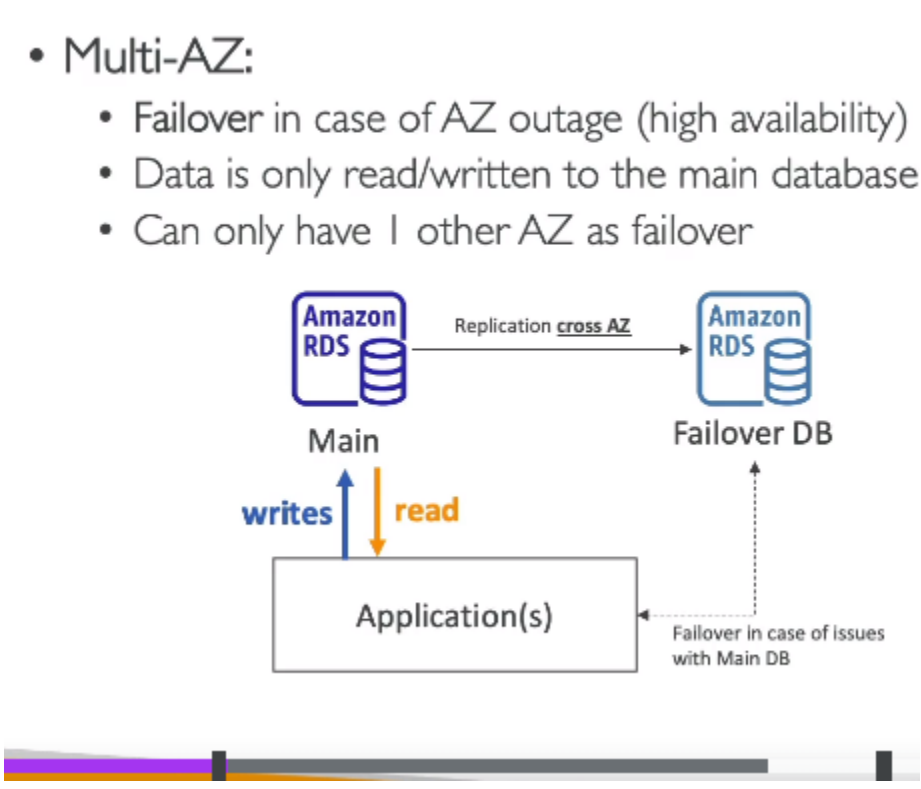
RDS Deployments: Read Replicas, Multi-AZ

1. So when you deploy RDS databases, you need to understand ..there are multiple architectural choices
2. Consider you have an application and it needs to read data from rds…consider having more applications trying to read more and more data from rds..the way we can handle this situation is creating Read Replica ..this allows our application to read data from replicas



3.

4. Multi-AZ



5.

6. Here we will create a failover DB in another region…

7. If by any reason our main db which is aws rds fails..then rds will trigger the failover db ..then use failover db
8. What is failover db?
9. In AWS, a failover AZ is an Availability Zone (AZ) that is configured to take over for another AZ in the event of a failure. This ensures that your workloads are always available, even if one AZ fails.

   Here are some of the benefits of using a failover AZ:

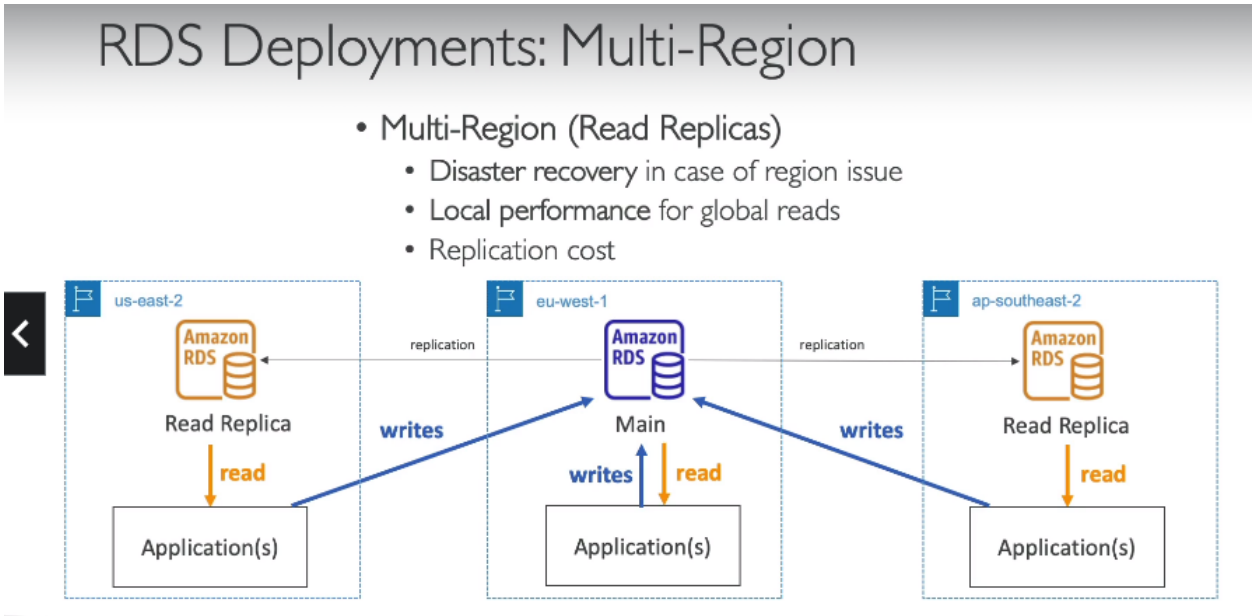
   - **Increased availability:** If one AZ fails, your workloads will automatically fail over to the failover AZ. This ensures that your workloads are always available, even if one AZ is unavailable.
   - **Reduced downtime:** The failover process is typically very fast, so your workloads will experience minimal downtime in the event of a failure.
   - **Simplified management:** You only need to manage one AZ, as the failover AZ is automatically configured. This can save you time and effort.

   If you are running workloads that are critical to your business, you should consider using a failover AZ. This will help to ensure that your workloads are always available, even if one AZ fails.

10. To create a failover AZ, you will need to create a database instance in a different AZ than your primary database instance. You can then configure the secondary database instance as a read replica of the primary database instance. When you do this, AWS will automatically replicate the data from the primary database instance to the secondary database instance. In the event of a failure, the secondary database instance will be promoted to become the new primary database instance.

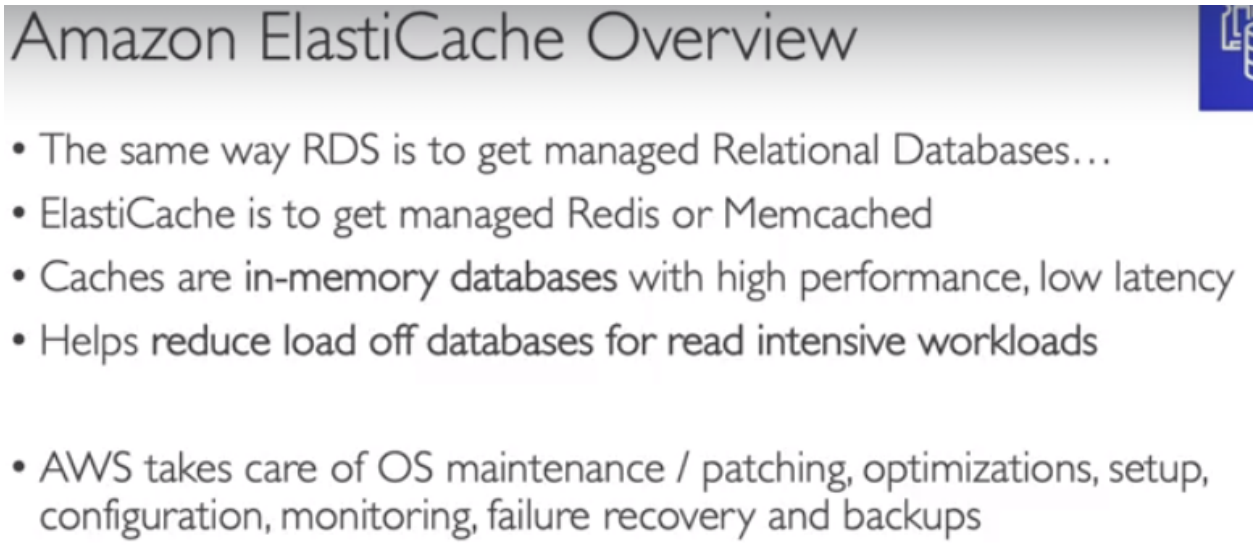
    Here are some of the things to keep in mind when creating a failover AZ:

    - The secondary database instance must be in a different AZ than the primary database instance.
    - The secondary database instance must be the same type and size as the primary database instance.
    - The secondary database instance must be configured as a read replica of the primary database instance.

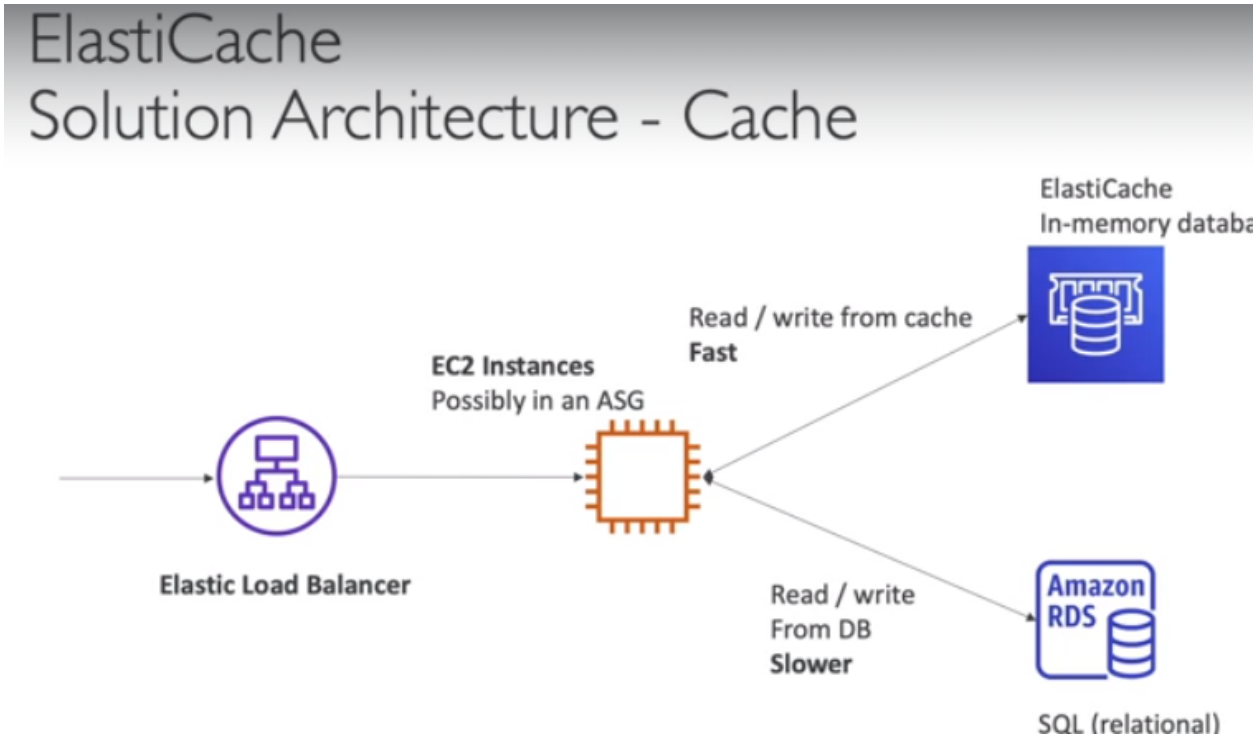
    Once you have created a failover AZ, you can test the failover process by simulating a failure of the primary database instance. You can do this by stopping the primary database instance or by deleting it. If the failover process works as expected, the secondary database instance will be promoted to become the new primary database instance.
11. RDS Deployments: Multi Region
12. 

13. Here we have our main rds db in eu west1..and replicas in the other region(us-east-2) and (ap-south-2)...
14. So the application's can read data from their own region from the replicas…but writing happens only in main rdb

Amazon ElasticCache Overview

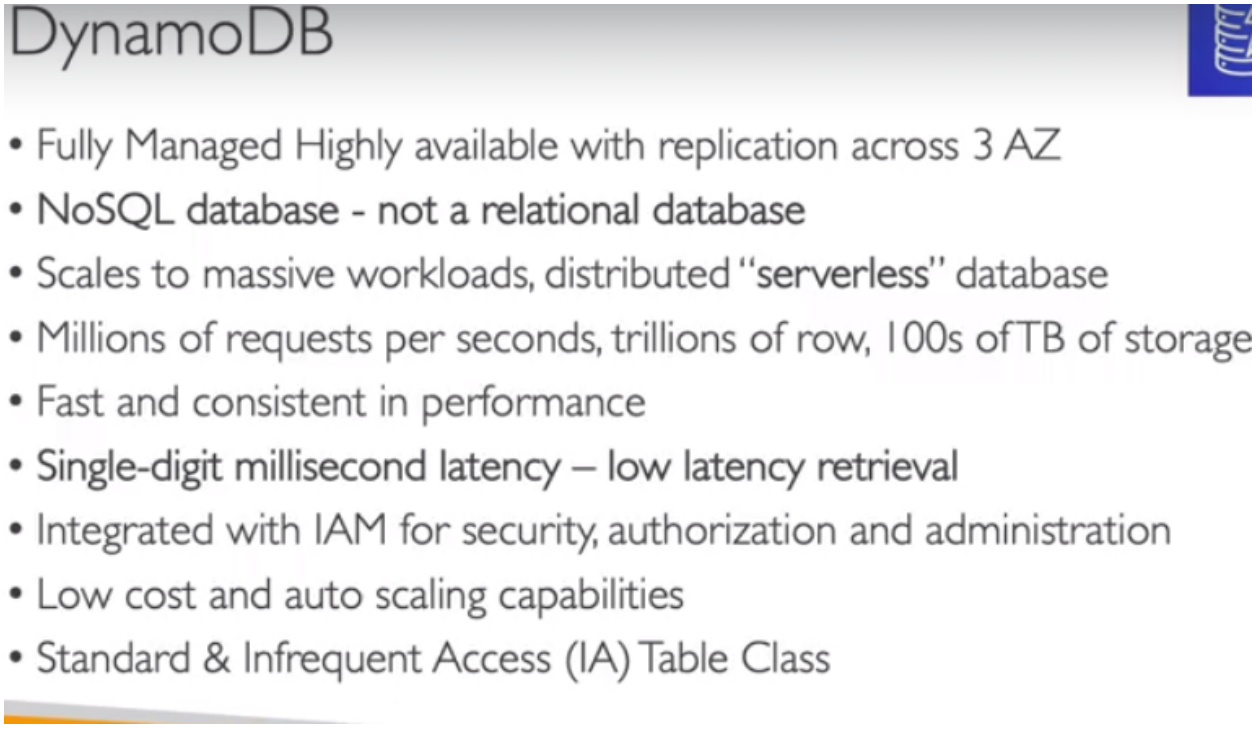


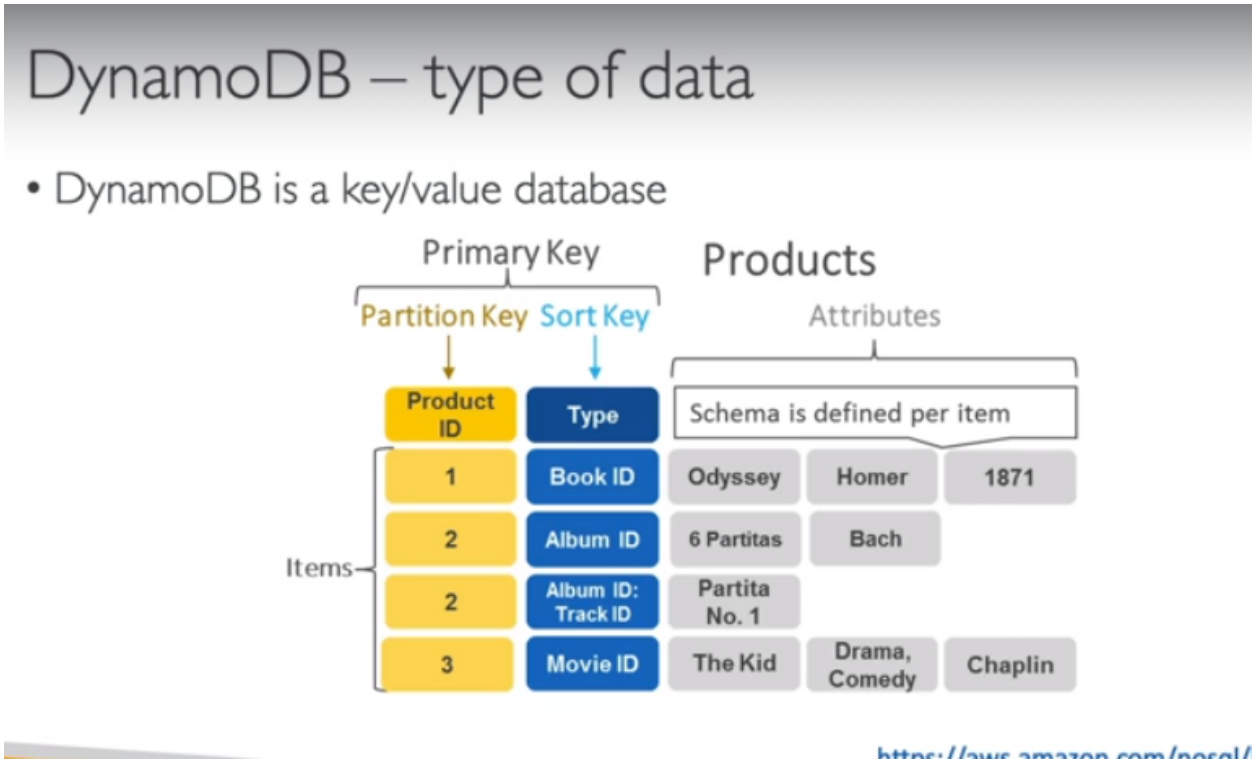
1.
2. Amazon Elastic Cache explained : https://chat.openai.com/share/01bbf599-0033-4ee1-b314-69bccd4a876d
3. Basically our load balancer send request to instance…and instance read /wrt from rds(which is slow)..And from elasticCache which is used to get frequently used data…

4. 

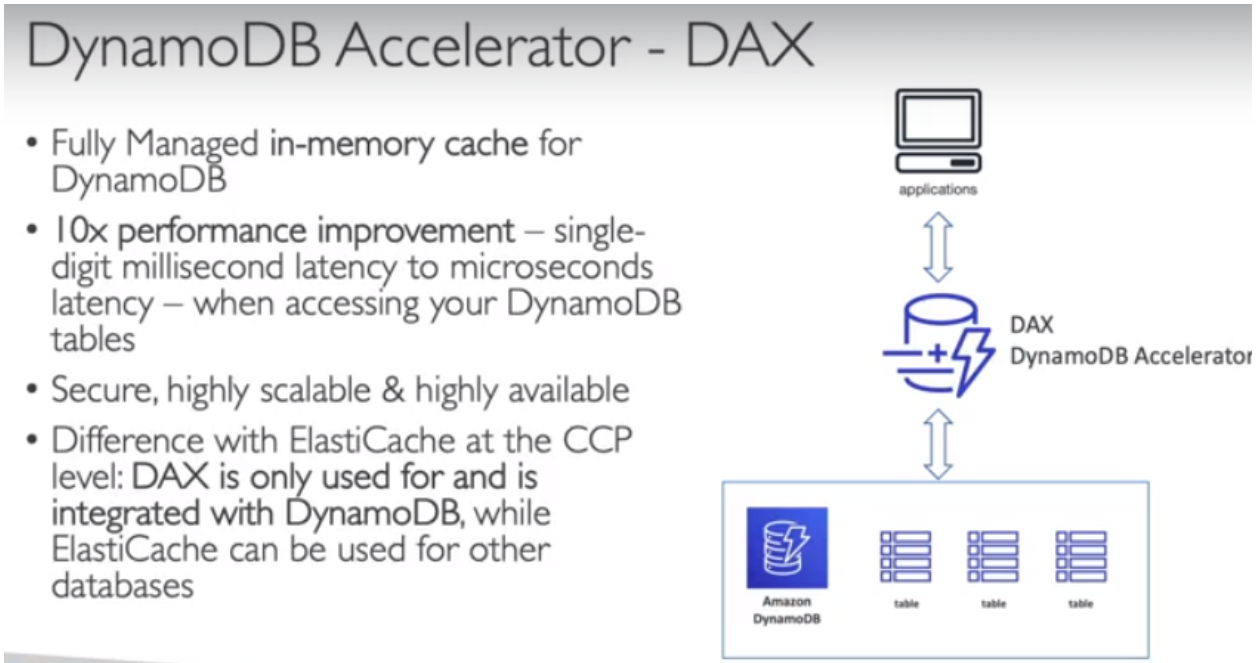
Dynamo DB

1. It is fully managed highly available with replication across 3 AZ
2. Its part of NoSql database ..and it is one of flagship products from aws
3. It scales to massive workload and it's distributed serverless database, that means that we don't provision any servers.
4. Like for rds and elasticCache we need to provide instance type..but in DynamoDB we wont

5. 

6. 

7. DynamoDB Accelerator - DAX
8. It is an Fully managed in memory cache for DYnamoDB..
9. If we have frequently accessed data..then we can store it in DAX

10. 

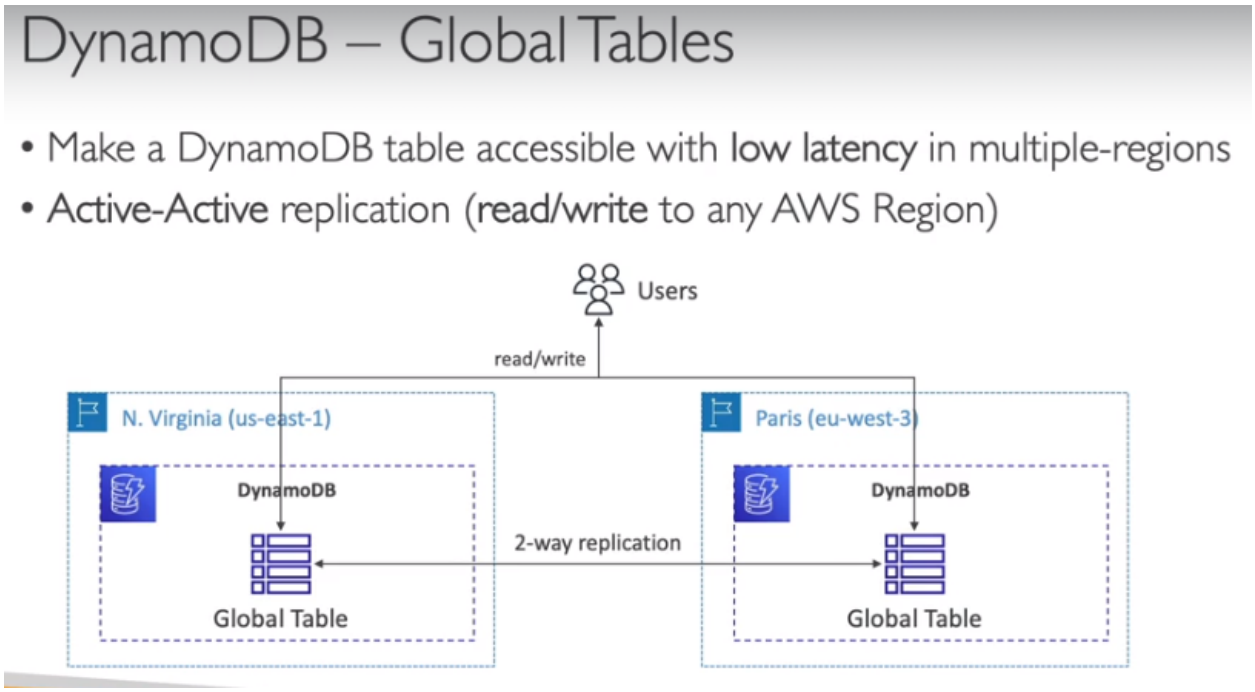
DynamoDB HandsOn:

1. First we go to dynamoDB and create a table..then we need to give a table name and partition key and click on create table
2. Here if u observe ..we don't have any database to store the table…
3. Thats the power of serverless services
4. Now we can add items into our table…first we have to give any user-id
5. Then click on add attributes and add few information
6. In DynamoDB as it is NoSql..we cannot join our table with different tables..as it has no relation

DynamoDB-Global Tables

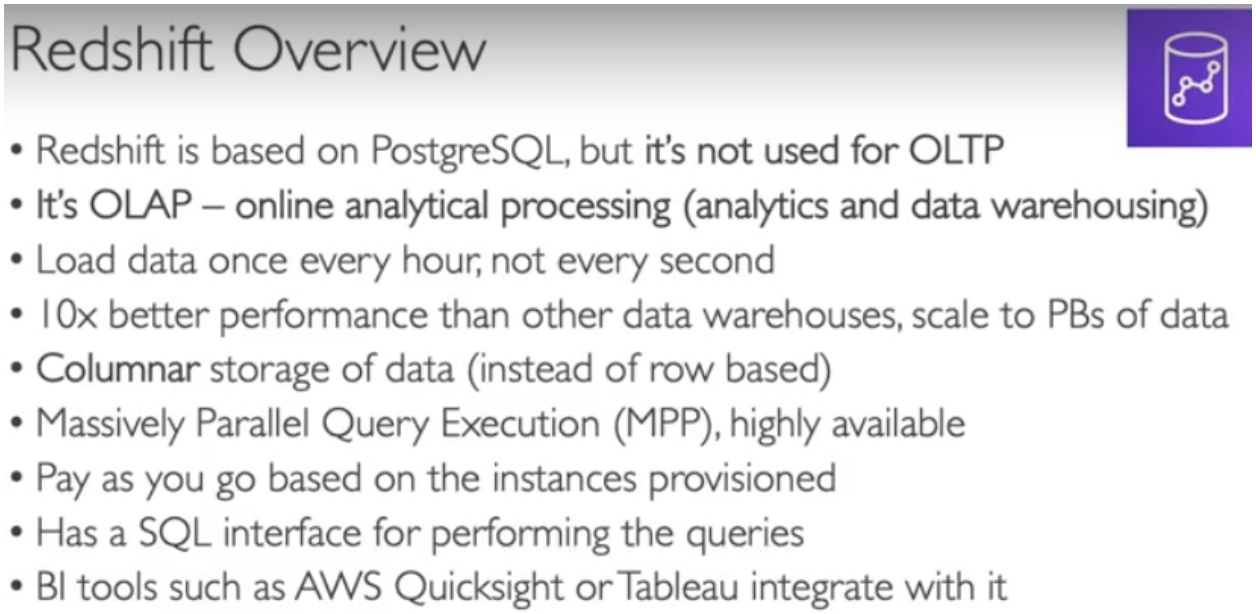
1. It is used to

2. Lets take an example..here we have dynamodb table in us-east1 and lets set it as global table..and users in this region can access the data in low latency

3. Now we can create this replication in another region..so here we have replicated this global in paris region..so the users who are close to paris ..can access the data very fast
4. Even if we actively write in one region..other region will also get replicated..
5. In this way we can create a global table in 10 regions
6. Like even if some users/employess write data in one region..all 10 regions will get that data
7. 

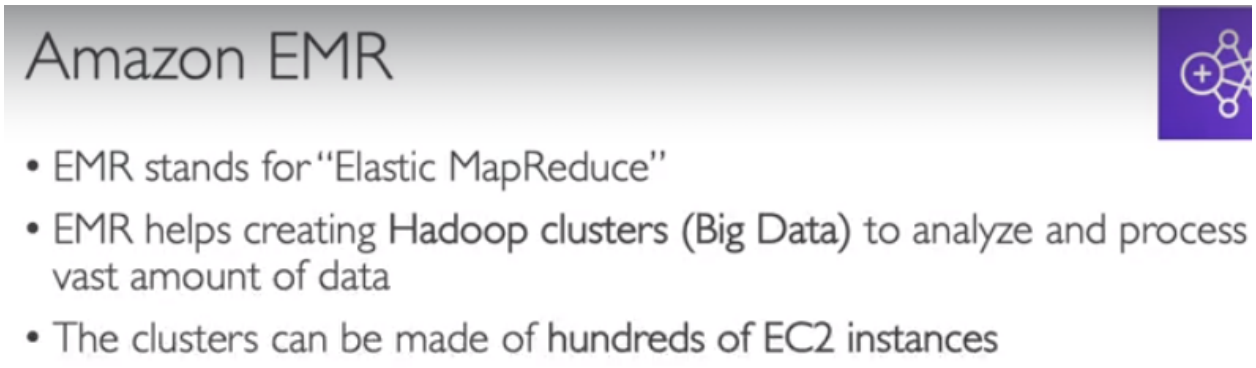
RedShift Overview

1. Next type of database we have is Redshift.
2. Redshift is a database that is based on PostgreSQL, but it is not used for OLTP. OLTP stands for Online Transaction Processing. That is what RDS was good for.
3. Redshift is good for OLAP
4. It is mainly used for data analytics
5. Refer bard/chatgpt for more info

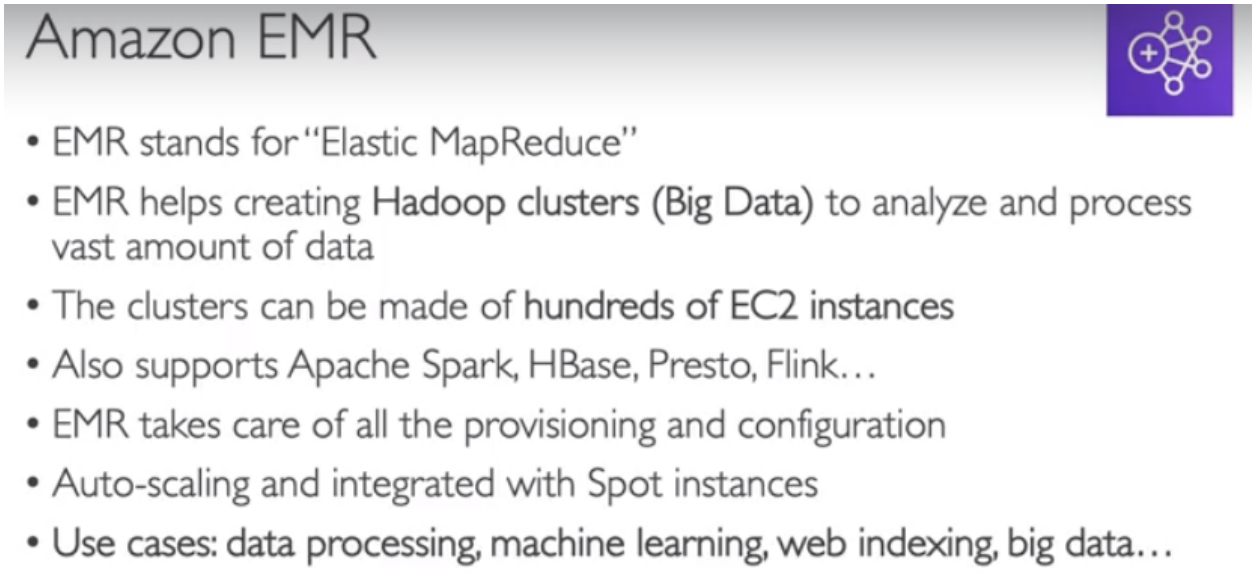


6.

Amazon EMR



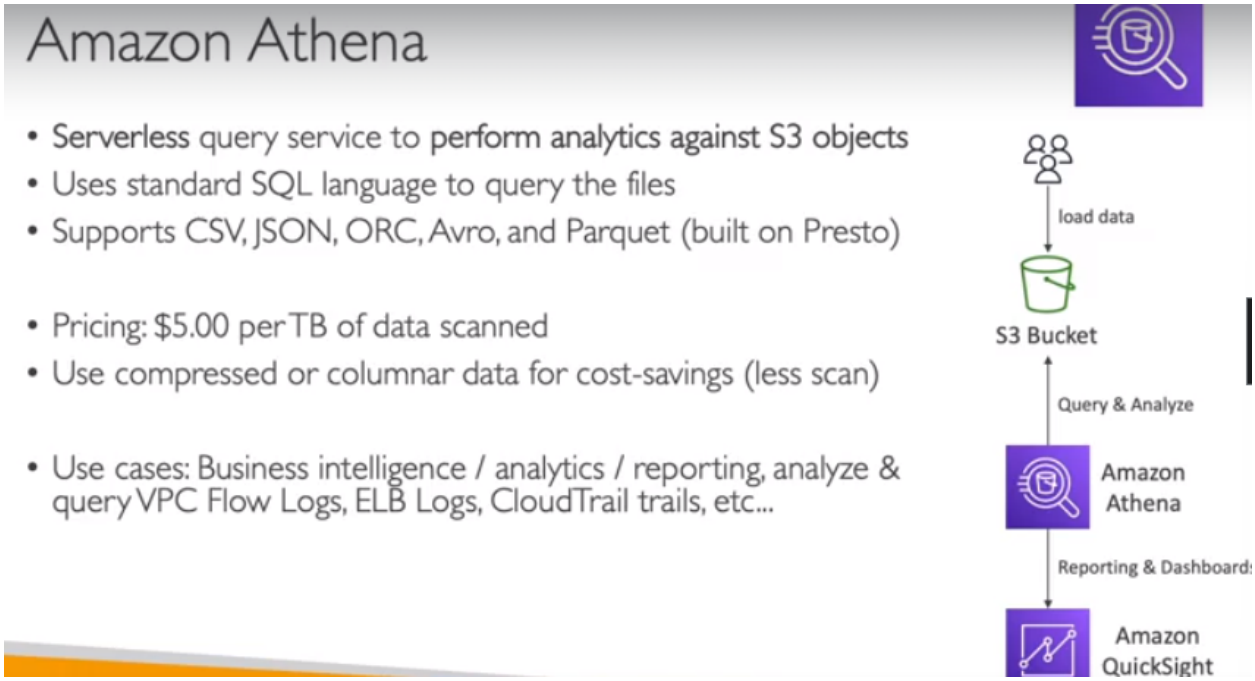
1.
2. So Hadoop is an open source technology, and they allow multiple servers that work in a cluster to analyze the data together, and so when you're using EMR,
3. you can create a cluster made of hundreds of EC2 instances that will be collaborating together to analyze your data.
4. So from an exam perspective, any time you see Hadoop cluster, think no more, it's going to be Amazon EMR.



5.

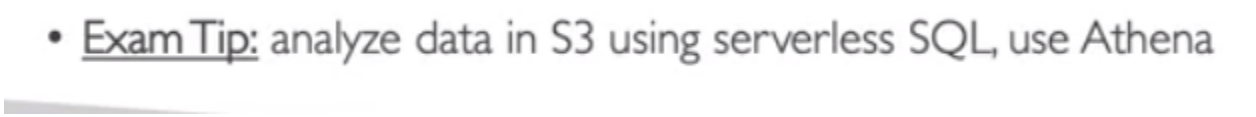
Amazon Athena

1. It is serverless query services to perform analysis on data
2. User loads the data in to s3 buckets..and athena can be used to query and analyze the data..
3. If we need dashboards and reports..we can use quicksight service in aws



4.

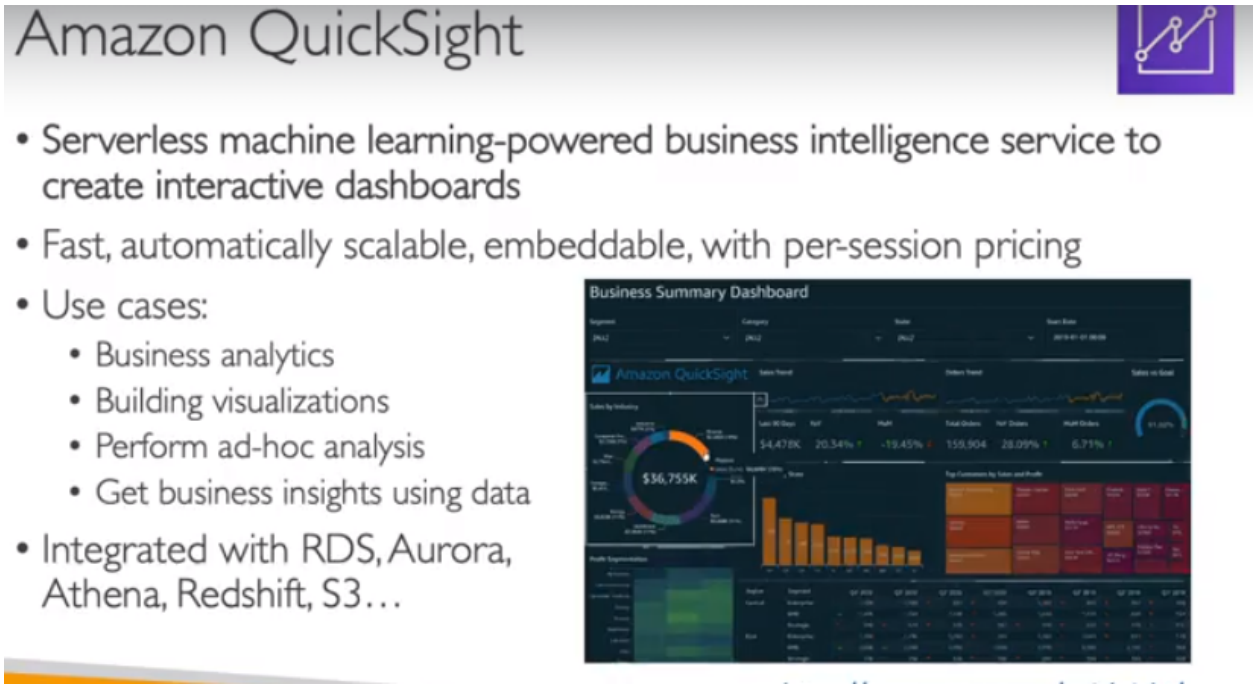
5.



QuickSight Overview

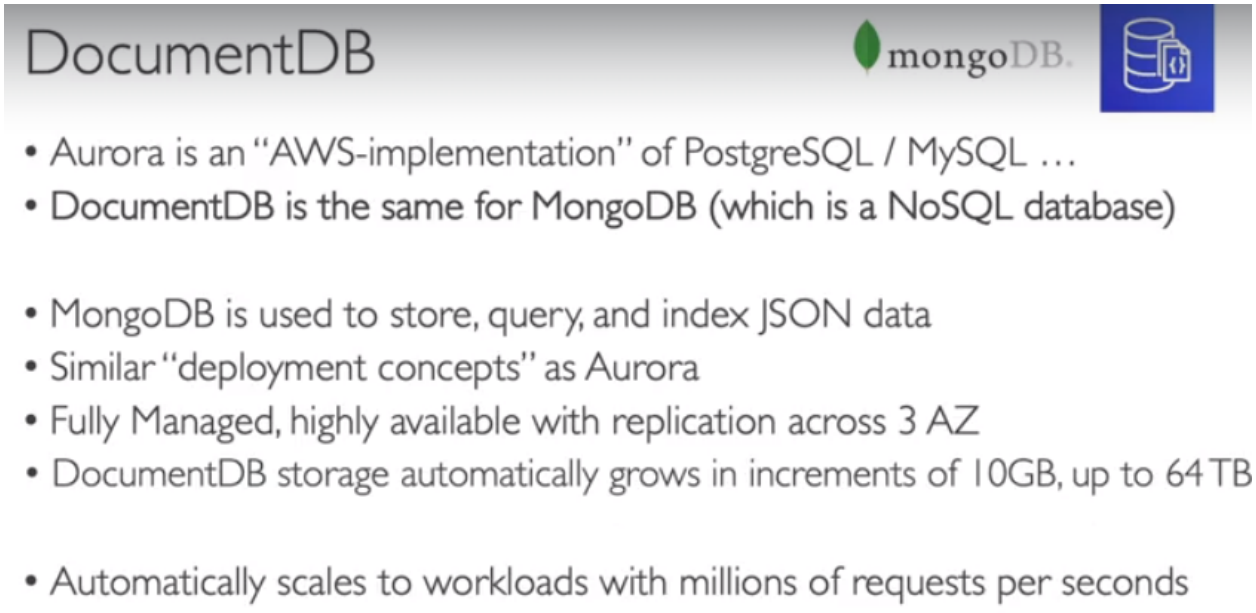
1. It is serverless

2. It create dashboard from our database …to showcase it to business stake holders



3.

DocumentDB



1.
2. Amazon Aurora and Amazon DocumentDB are both fully managed database services offered by Amazon Web Services (AWS). However, they have different strengths and weaknesses, and are suited for different types of applications.

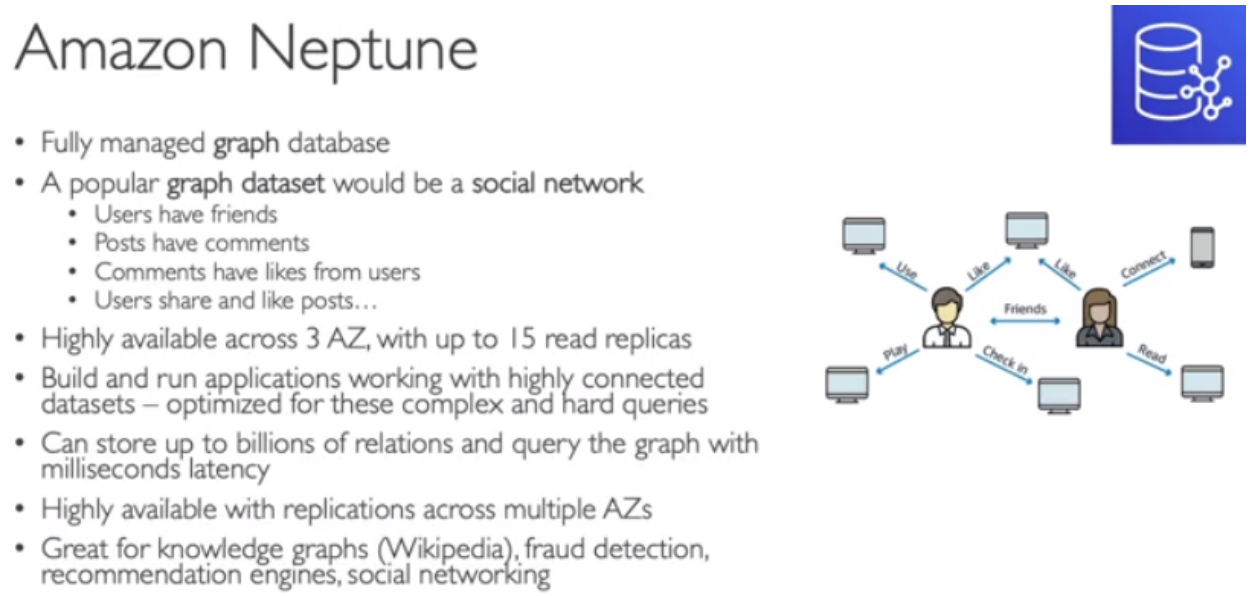
   Amazon Aurora is a relational database service that is compatible with MySQL and PostgreSQL. It is known for its high performance, scalability, and availability. Aurora is a good choice for applications that require high throughput and low latency, such as online transaction processing (OLTP) and data warehousing.

   Amazon DocumentDB is a document database service that is compatible with MongoDB. It is known for its high performance, scalability, and availability. DocumentDB is a good choice for applications that store and process large amounts of semi-structured data, such as JSON documents.
3. So at the exam, if you see anything related to MongoDB, think DocumentDB, or if you see anything related to NoSQL databases, think DocumentDB and also DynamoDB.

Amazon Neptune

1. It is a fully managed graph database..
2. It's used to build and run applications that are gonna be with highly connected datasets,

3. so like a social network, and because Neptune is optimized to run queries that are complex and hard on top of these graph datasets.
4. You can store up to billions of relations on the database and query the graph with milliseconds latency.

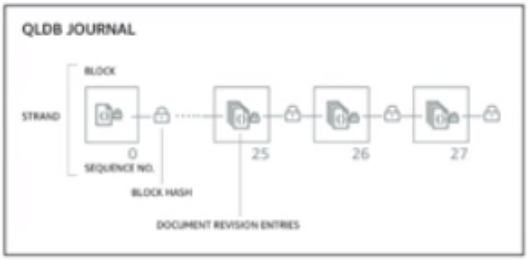


5.
6. For example, the Wikipedia database is a knowledge graph because all the Wikipedia articles are interconnected with each other, fraud detection, recommendations engine and social networking.
7. So, coming from an exam perspective, anytime you see anything related to graph databases, think no more than Neptune.

Amazon QLDB

Amazon QLDB

- QLDB stands for "Quantum Ledger Database"
- A ledger is a book recording financial transactions
- Fully Managed, Serverless, High available, Replication across 3 AZ
- Used to review history of all the changes made to your application data over time
- Immutable system: no entry can be removed or modified, cryptographically verifiable



1.
2. So how does it work?
3. Well, there is behind the scenes a journal, and so a journal has a sequence of modifications. And so anytime a modification is made, there is a cryptographic hash that is computed
4. which guarantees that nothing has been deleted or modified and so this can be verified by anyone using the database.
5. So this is extremely helpful for financial transactions because you wanna make sure that obviously no financial transaction is disappearing from your database which makes QLDB a great ledger database in the cloud.
6. What is cryptographically verifiable log?
7. A cryptographically verifiable log is a type of data structure that uses cryptography to ensure the authenticity and integrity of data. This means that it can be used to verify that data has not been tampered with, and that it has been created by the party that claims to have created it.

   Cryptographically verifiable logs are often used in applications where it is important to be able to trust the data, such as financial transactions, supply chain management, and healthcare records.

8. Here are some of the key features of cryptographically verifiable logs:

   - **Immutability:** The data in a cryptographically verifiable log cannot be changed or deleted once it has been added. This ensures that the log cannot be tampered with.
   - **Cryptographic verification:** The data in a cryptographically verifiable log can be verified using cryptography. This ensures that the data has not been tampered with and that it has been created by the party that claims to have created it.
   - **Auditability:** The data in a cryptographically verifiable log can be audited to track changes and to identify any unauthorized modifications.
   - **Scalability:** Cryptographically verifiable logs can be scaled to meet the needs of large-scale applications.

9. We can manipulate data using sql...like seeing

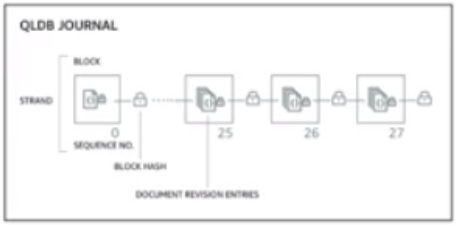
   Manipulate means handling the data efficiently.

   In the example, you have 5 rows and 10 columns and you want to see only the 4th row and 8th column.

10. You can get this data using SQL commands.

11. Amazon QLDB

    - QLDB stands for "Quantum Ledger Database"
    - A ledger is a book recording financial transactions
    - Fully Managed, Serverless, High available, Replication across 3 AZ
    - Used to review history of all the changes made to your application data over time
    - Immutable system: no entry can be removed or modified, cryptographically verifiable

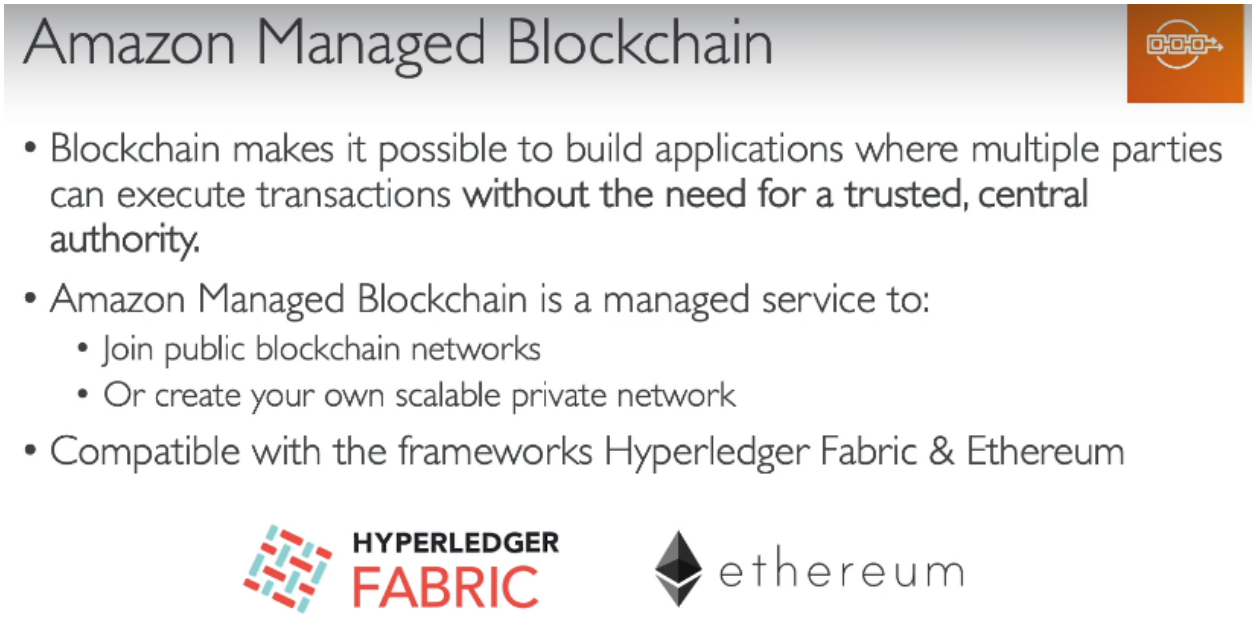
    

    - 2-3x better performance than common ledger blockchain frameworks, manipulate data using SQL
    - Difference with Amazon Managed Blockchain: no decentralization component, in accordance with financial regulation rules

12. But the difference between QLDB and Managed Blockchain is that with QLDB, there is no concept of decentralization. That means that there's just a central database owned by Amazon that allows you to write this journal.

13. So the difference between QLDB and Managed Blockchain is that QLDB has a central authority component and it's a ledger, whereas managed blockchain is going to have a de-centralization component as well. Okay. So that's it, anytime you see financial transactions

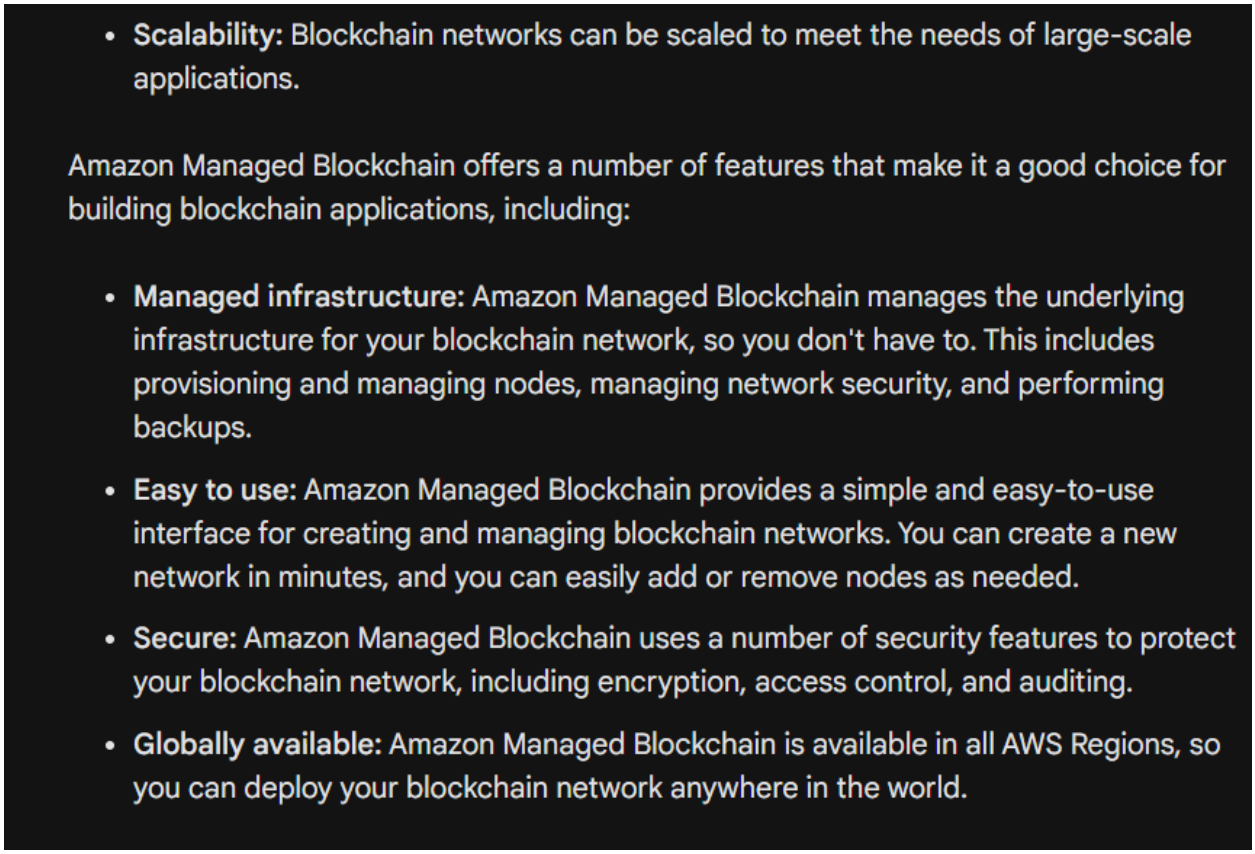
Amazon Managed Blockchain

1. 

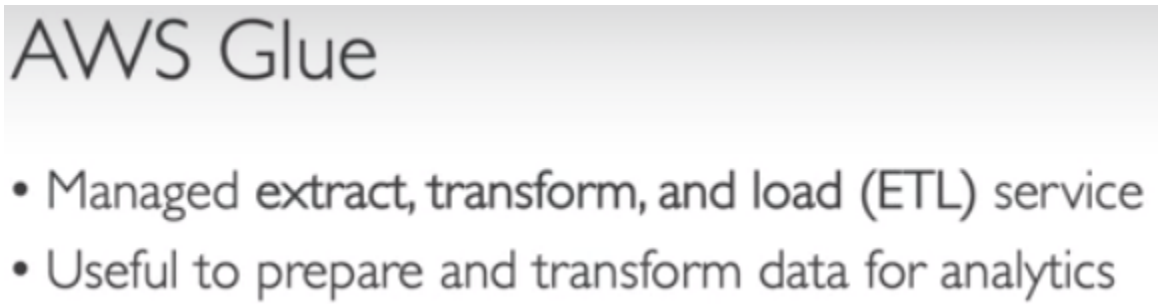
2. Amazon Managed Blockchain is a fully managed service that makes it easy to create and manage blockchain networks using the popular open source frameworks Hyperledger Fabric and Ethereum. It can be used to build applications where multiple parties can execute transactions without the need for a trusted, central authority.

   Amazon Managed Blockchain is a good choice for applications that require:

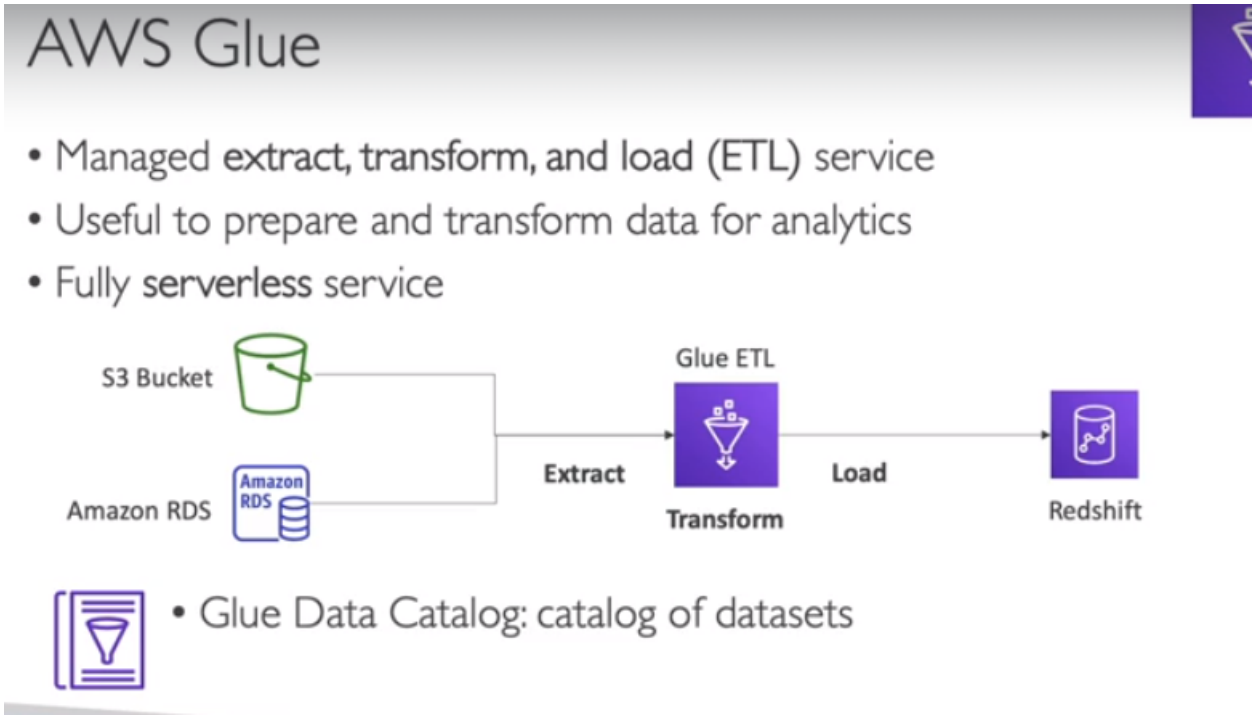
   - **Immutability:** The data in a blockchain network is immutable, meaning that it cannot be changed or deleted once it has been added. This makes blockchain networks ideal for applications that require a tamper-proof record of data changes, such as financial transactions, supply chain management, and healthcare records.
   - **Transparency:** All transactions in a blockchain network are transparent, meaning that they can be viewed by anyone. This makes blockchain networks ideal for applications that require accountability and traceability, such as voting systems and land registry systems.
   - **Security:** Blockchain networks are secure, meaning that it is difficult to tamper with the data or to hack into the network. This makes blockchain networks ideal for applications that require a high degree of security, such as financial transactions and intellectual property management.

3. 

AWS Glue

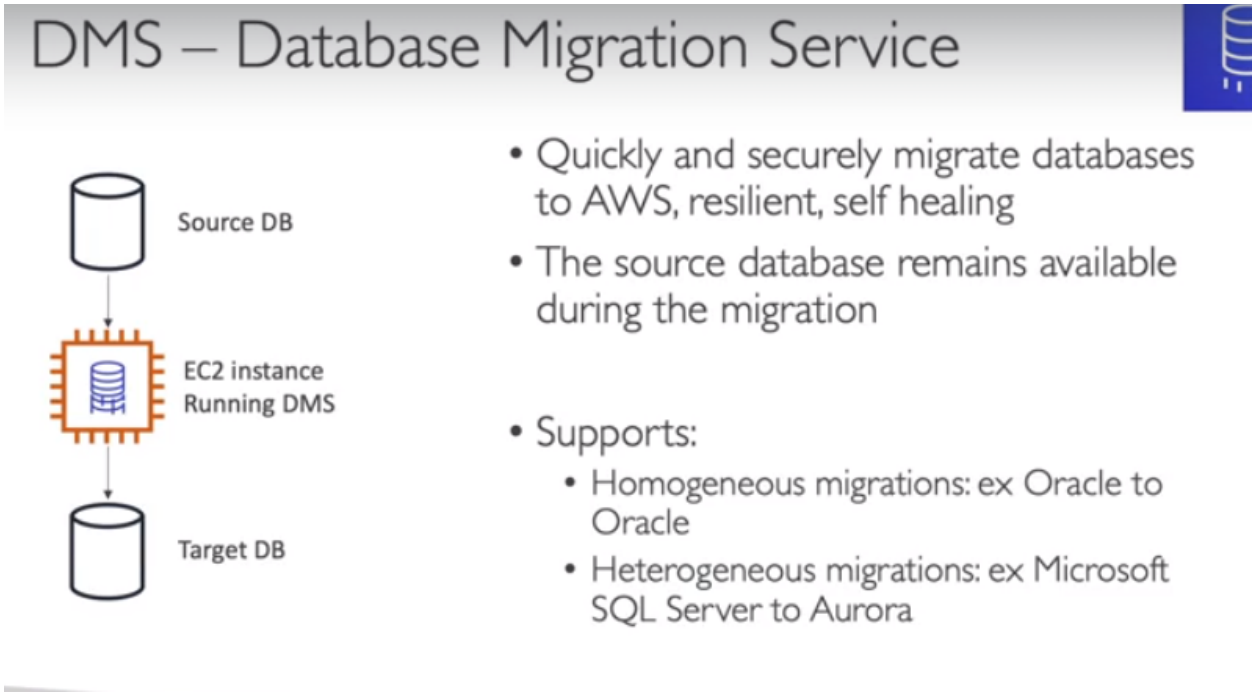
1. 

2. So, what is ETL?
3. Well, ETL is very helpful when you have some datasets but they're not exactly in the right form, or the right format that you need, to do your analytics on them. And so the idea is that you would use an ETL service to prepare and transform that data.
4. If we checj the diagram..here glue sits in the middle..
5. We use glue to extract data from s3 and rds...once the data is extracted it will be in glue service and for transforming data..we need to write some scripts

6. After transforming we will analyze the data and load into ..like for example to aws redshift
7. 

DMS-Database Migration Service

1. Here we extract the data from source DB …thru an ec2 instance which will be on DMS service
2. Then the DMS service will insert the data to the target DB

3. 

Database & Analytics Summary

1. First we have relational db..in this we have OLTP
2.