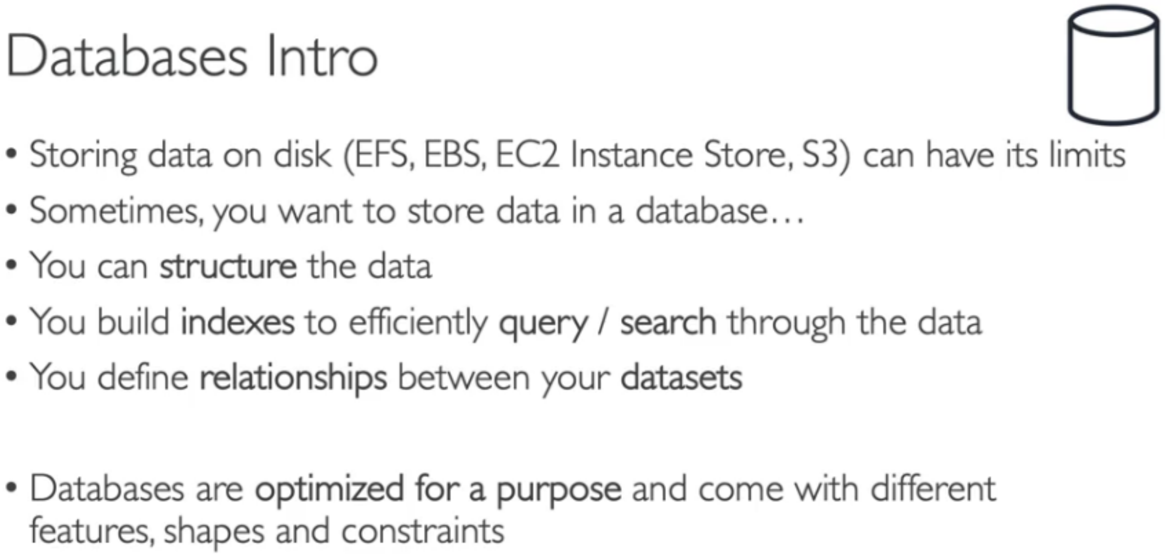
Databases & Analytics

Introduction

In this section, we have covered 3 types of databases:

- Relational
- Non relational
- In-memory



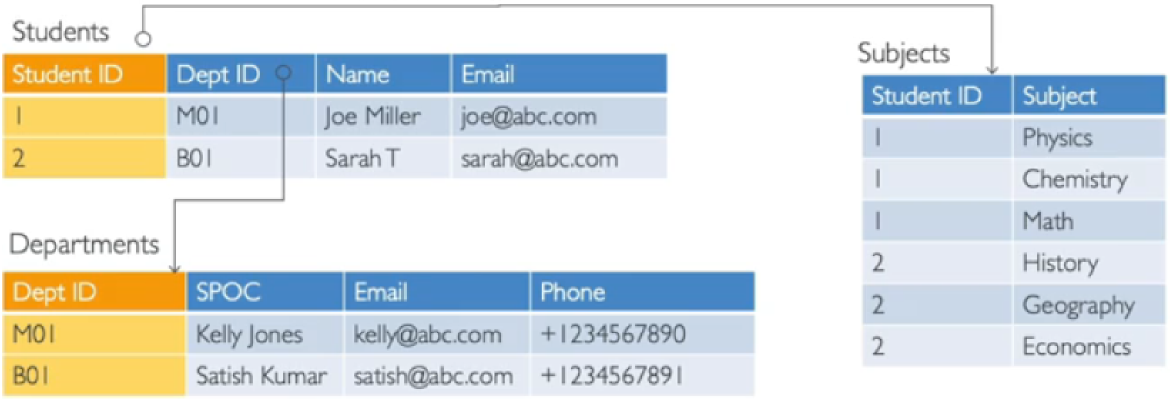
Databases Intro

- Storing data on disk (EFS, EBS, EC2 Instance Store, S3) can have its limits
- Sometimes, you want to store data in a database…
- You can **structure** the data
- You build **indexes** to efficiently **query** / **search** through the data
- You define **relationships** between your **datasets**

- Databases are **optimized for a purpose** and come with different features, shapes and constraints

Relational Databases

- Looks just like Excel spreadsheets, with links between them!
- Can use the SQL language to perform queries / lookups



Students

Student ID	Dept ID	Name	Email
1	M01	Joe Miller	joe@abc.com
2	B01	Sarah T	sarah@abc.com

Departments

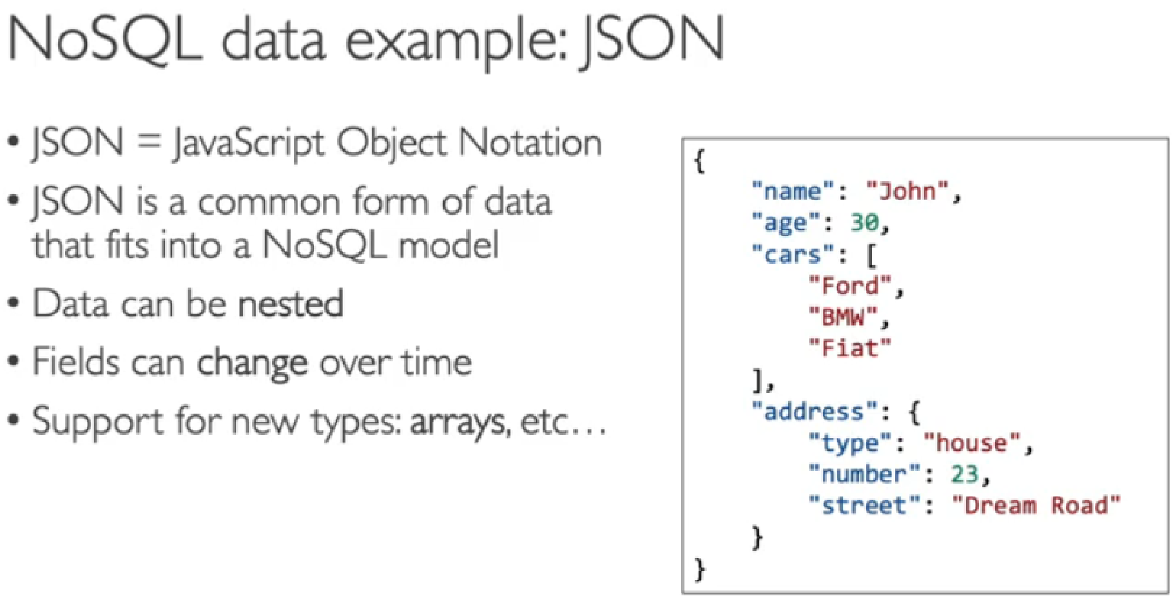
Dept ID	SPOC	Email	Phone
M01	Kelly Jones	kelly@abc.com	+1234567890
B01	Satish Kumar	satish@abc.com	+1234567891

Subjects

Student ID	Subject
1	Physics
1	Chemistry
1	Math
2	History
2	Geography
2	Economics

NoSQL Databases

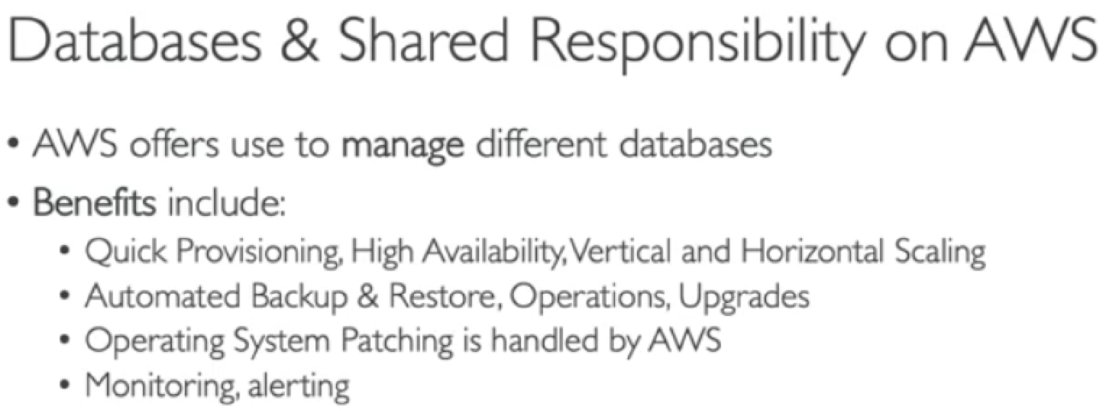
- NoSQL = non-SQL = non relational databases
- NoSQL databases are purpose built for specific data models and have flexible schemas for building modern applications.
- Benefits:
  - Flexibility: easy to evolve data model
  - Scalability: designed to scale-out by using distributed clusters
  - High-performance: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



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"
    }
}
```



Databases & Shared Responsibility on AWS

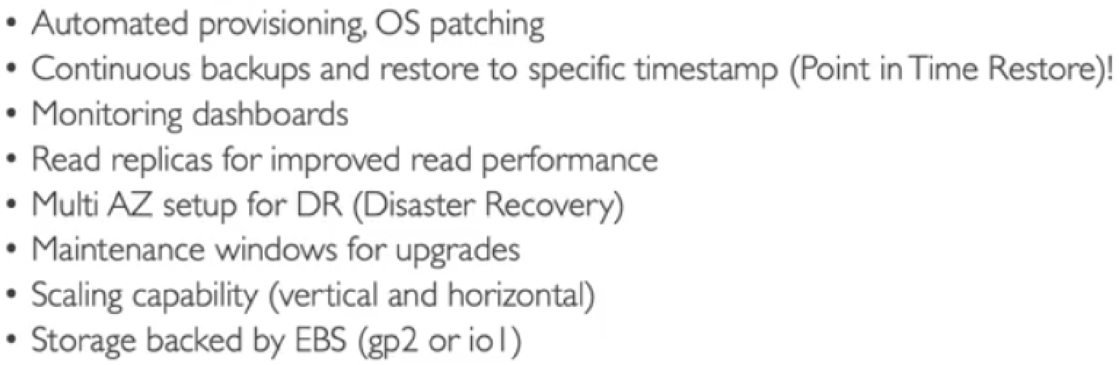
- AWS offers use to manage different databases
- Benefits include:
  - Quick Provisioning, High Availability, Vertical and Horizontal Scaling
  - Automated Backup & Restore, Operations, Upgrades
  - Operating System Patching is handled by AWS
  - Monitoring, alerting

AWS RDS

- RDS stands for relational database service
- It is managed by AWS and uses SQL as a query language.

- It allows you to create different kinds of databases in the cloud like postgres, mySQL, oracle, Aurora, etc.

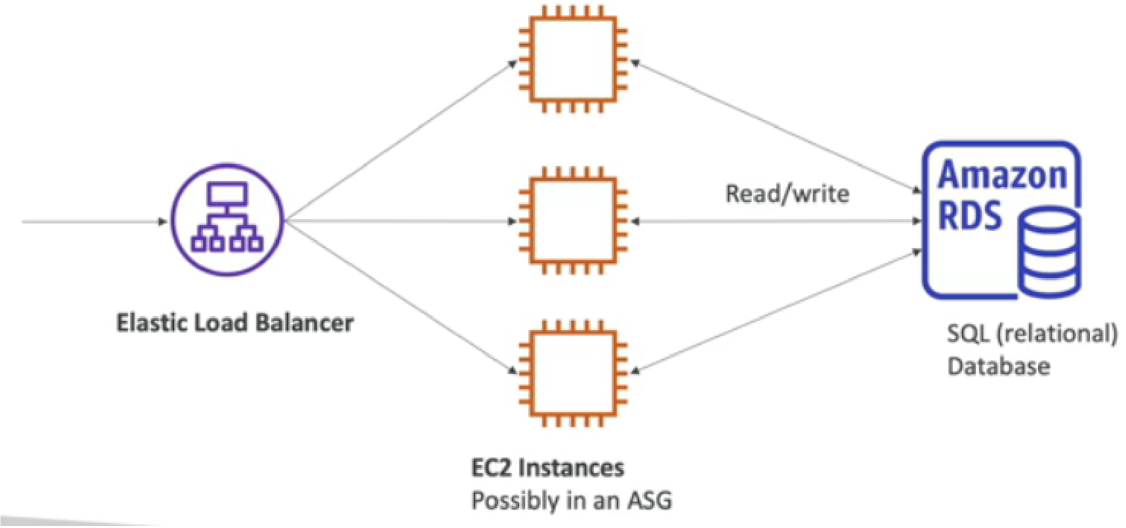
- Using RDS instead of simply deploying your database on EC2 has several advantages as RDS is a managed service, such as:



  - 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 database.

- **RDS Architecture**

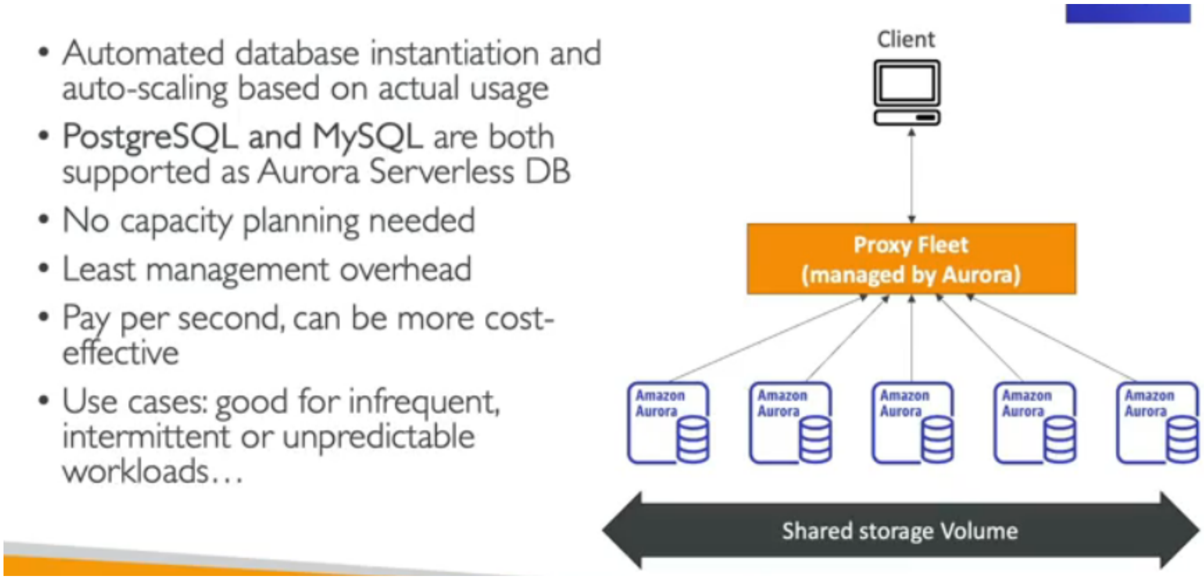


We have our load balancer which will be taking web requests and is fronting multiple EC2 instances (they are probably into an auto scaling group) and the EC2 instances need to store and share the data somewhere and because this is structured data, they will need a relational database. Hence, RDS comes to play.

Amazon Aurora

- Aurora is aws' database technology. It is not open source.
- Aurora supports PostgreSQL and MySQL.
- The idea behind aurora is simply performance optimization. Aurora is **cloud optimized or cloud native** and offers 5x better performance with MySQL and 3x better performance with PostgreSQL as compared to using them with RDS.
- Storage in aurora grows automatically with increments of 10GB (up to 128TB)
- Even though aurora is more expensive than RDs, it performs way better resulting in cost optimization and increased efficiency.
- RDS is free tier but aurora is not.

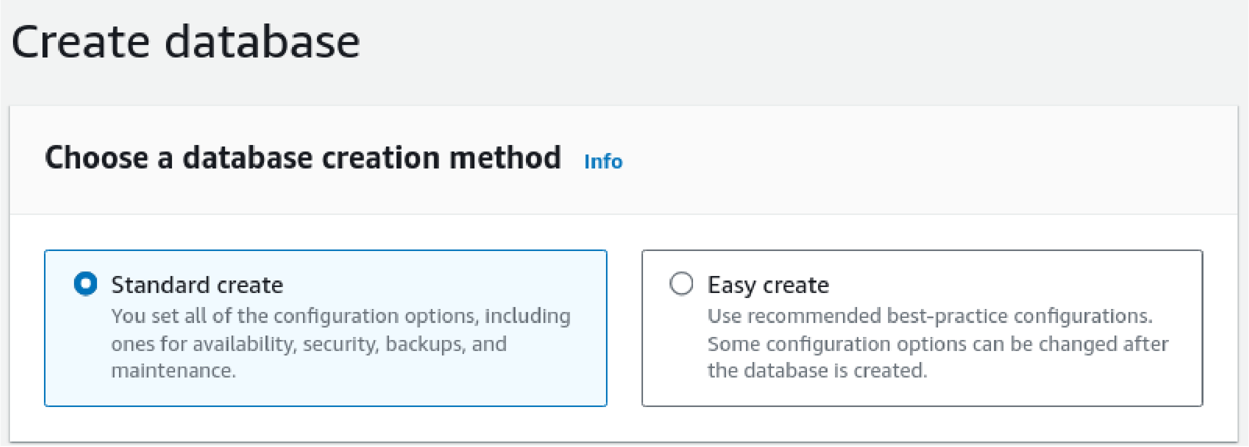
- Aurora offers serverless database



- Automated database instantiation and auto-scaling based on actual usage
- PostgreSQL and MySQL are both supported as Aurora Serverless DB
- No capacity planning needed
- Least management overhead
- Pay per second, can be more cost-effective
- Use cases: good for infrequent, intermittent or unpredictable workloads…

** from exam perspective, RDS and Aurora are two ways for you to create relational databases

RDS Hands-on

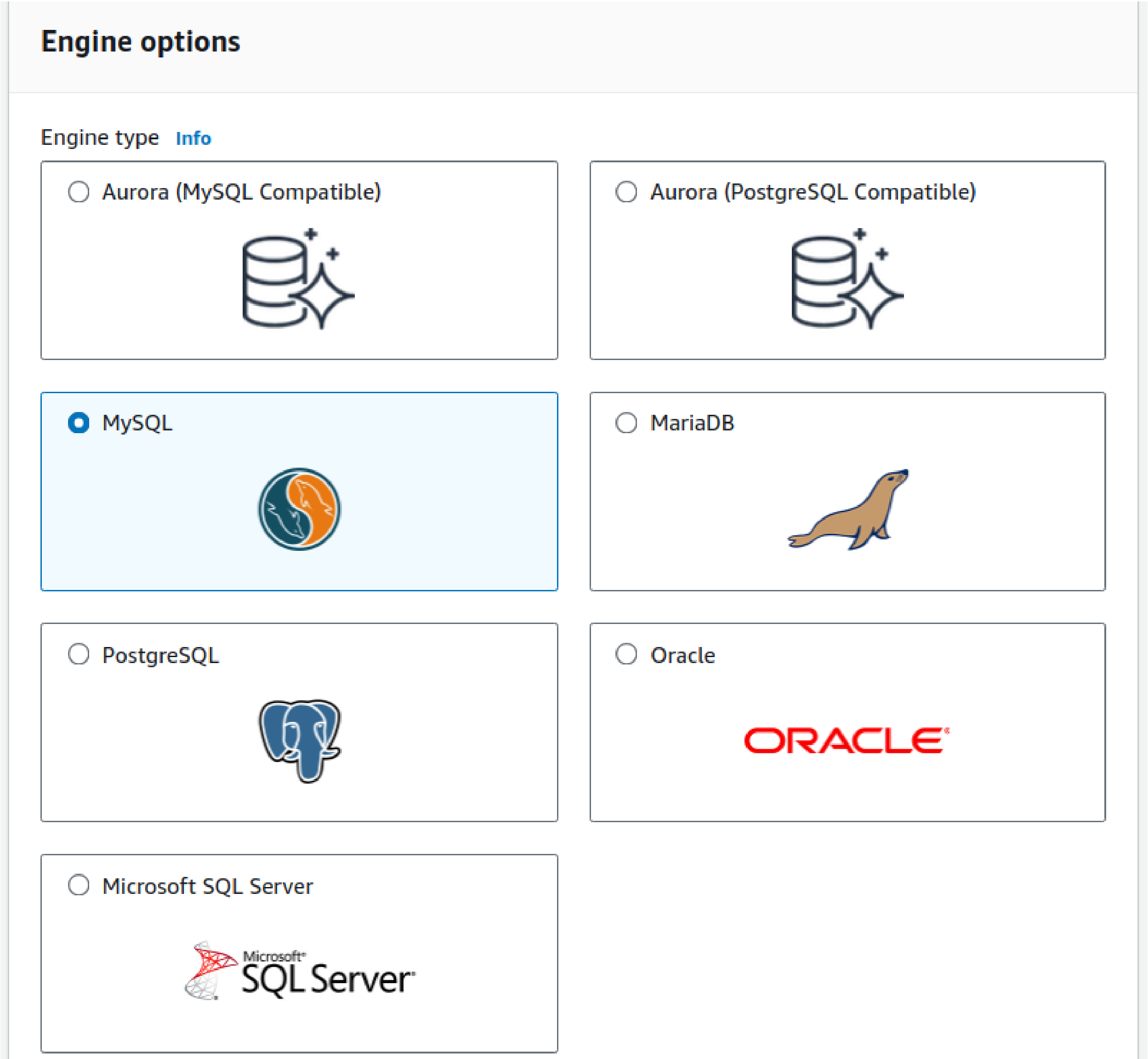


Create database

Choose a database creation method Info

● Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

○ Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.



Engine options

Engine type Info

○ Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

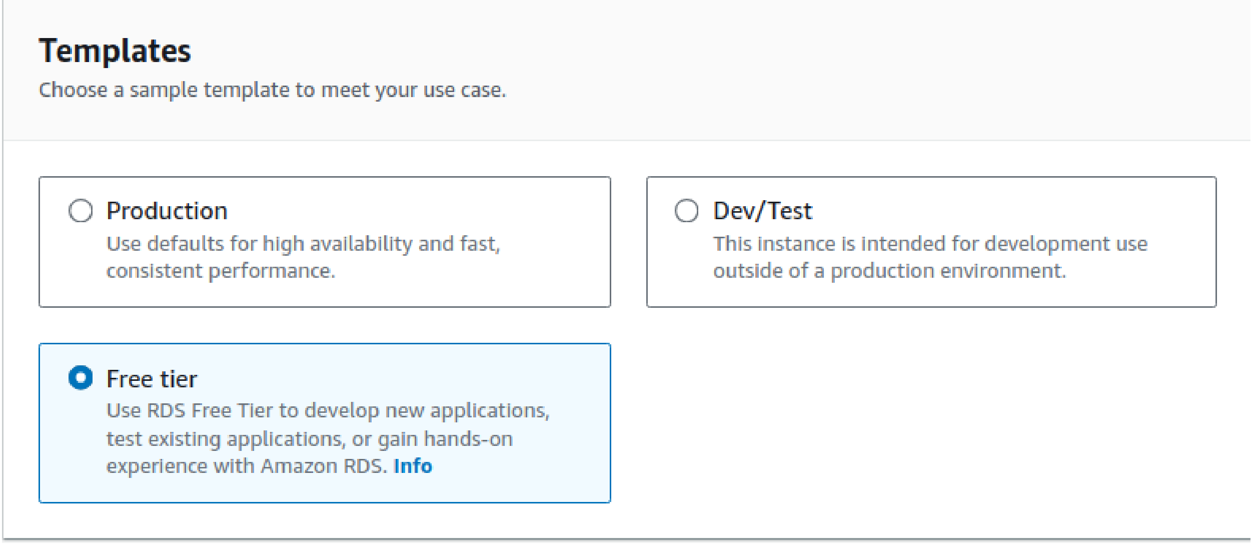
● MySQL

○ MariaDB

○ PostgreSQL

○ Oracle

○ Microsoft SQL Server



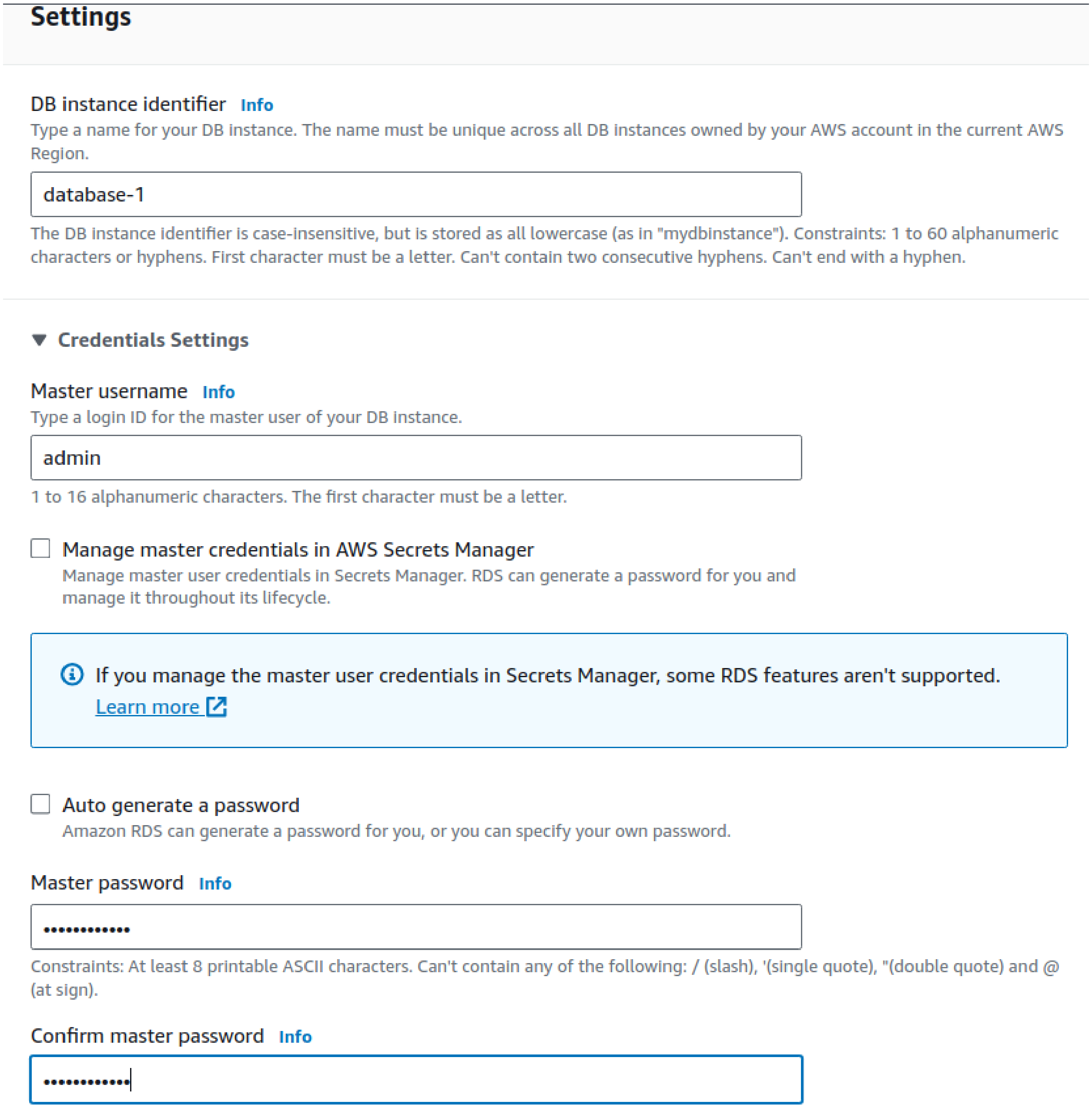
Templates

Choose a sample template to meet your use case.

○ Production
Use defaults for high availability and fast, consistent performance.

○ Dev/Test
This instance is intended for development use outside of a production environment.

● Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info



Settings

DB instance identifier Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username Info

Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. The first character must be a letter.

□ Manage master credentials in AWS Secrets Manager

Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

ⓘ If you manage the master user credentials in Secrets Manager, some RDS features aren't supported. Learn more

□ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password.

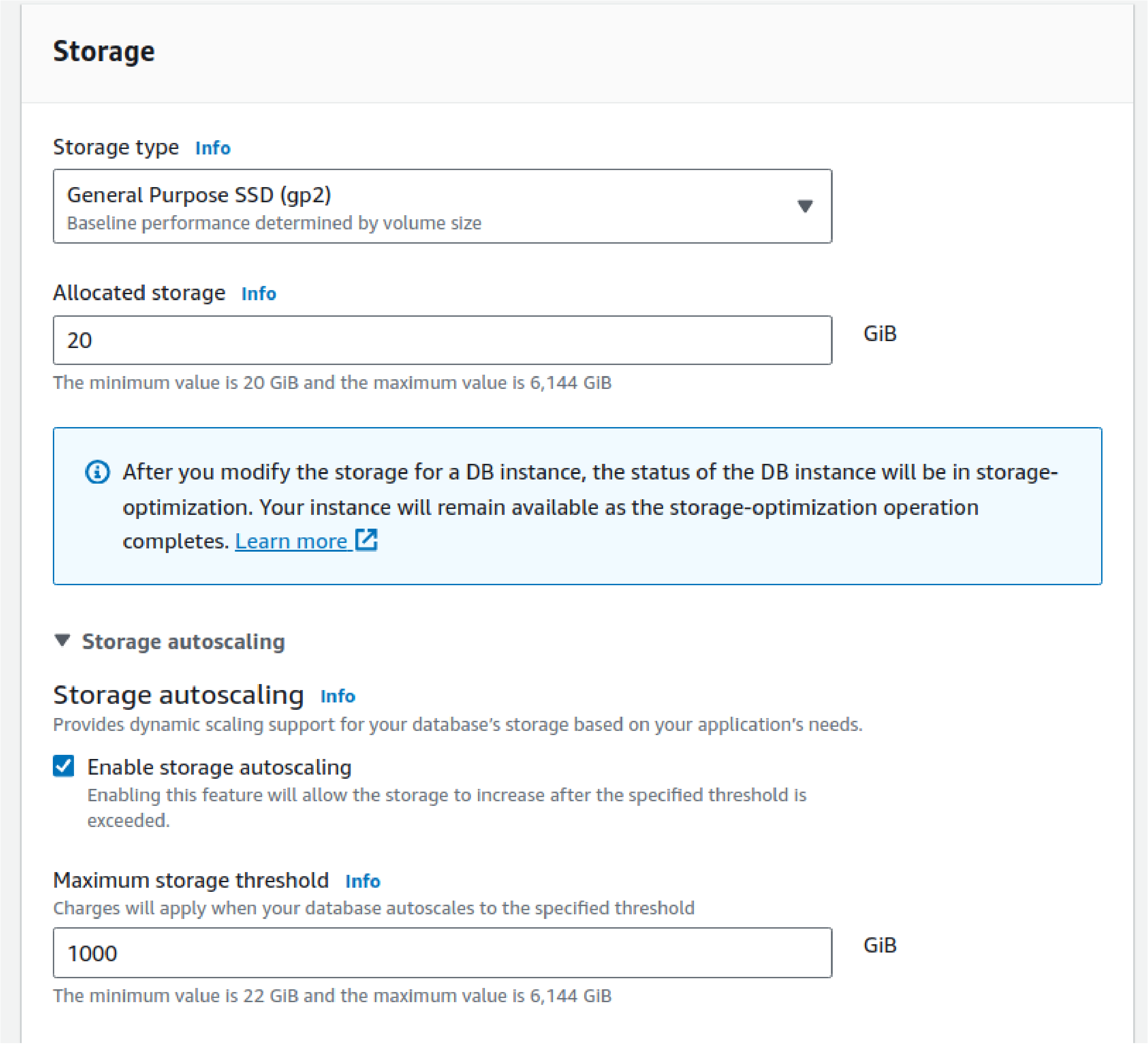
Master password Info

••••••••••••

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password Info

••••••••••••



Storage

Storage type Info

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage Info

20 GiB

The minimum value is 20 GiB and the maximum value is 6,144 GiB

After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes. Learn more

▼ Storage autoscaling

Storage autoscaling Info

Provides dynamic scaling support for your database's storage based on your application's needs.

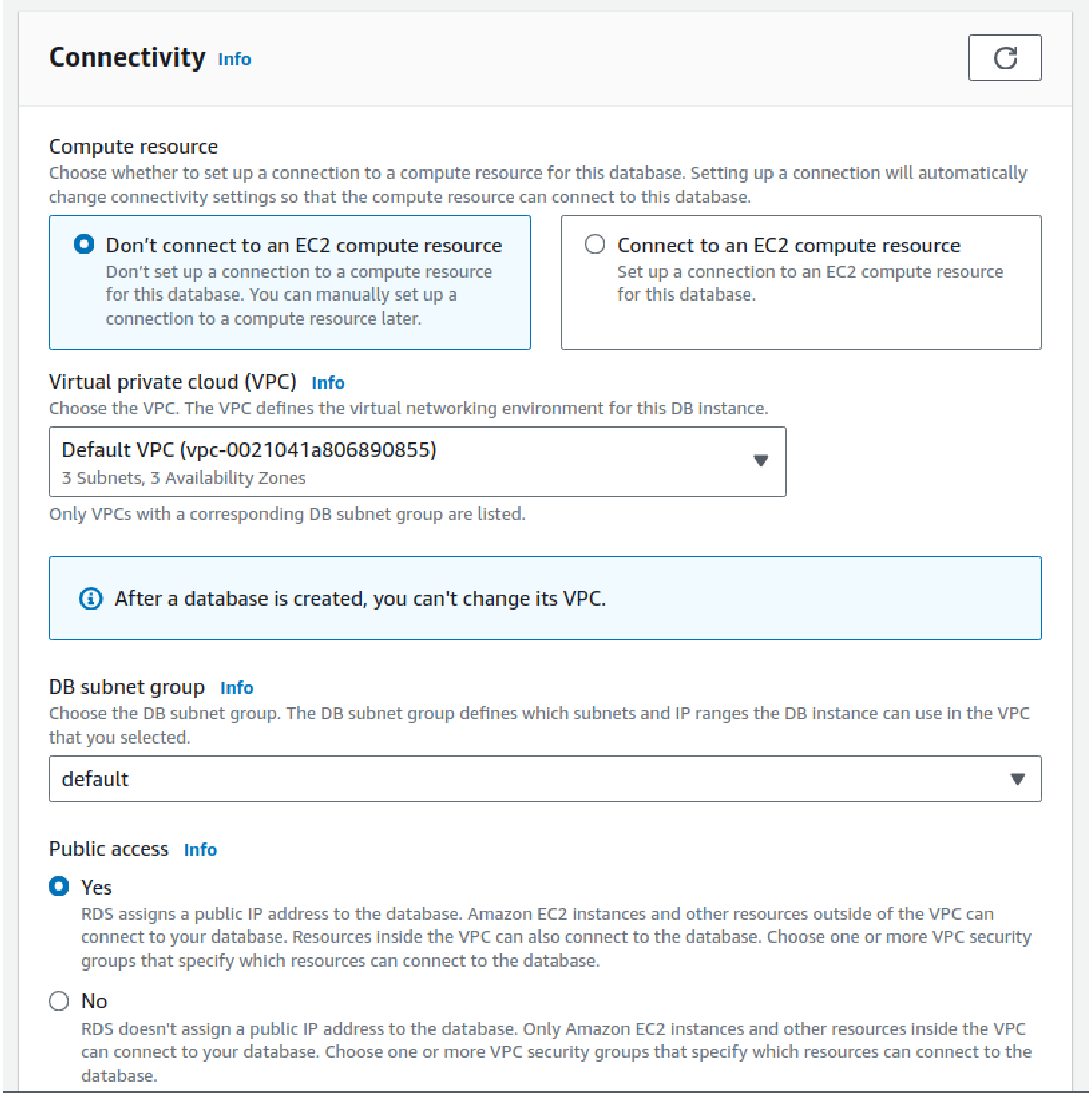
☑ Enable storage autoscaling
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

Maximum storage threshold Info

Charges will apply when your database autoscales to the specified threshold

1000 GiB

The minimum value is 22 GiB and the maximum value is 6,144 GiB



Connectivity

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

● Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

○ Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC)

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-0021041a806890855)
3 Subnets, 3 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

DB subnet group

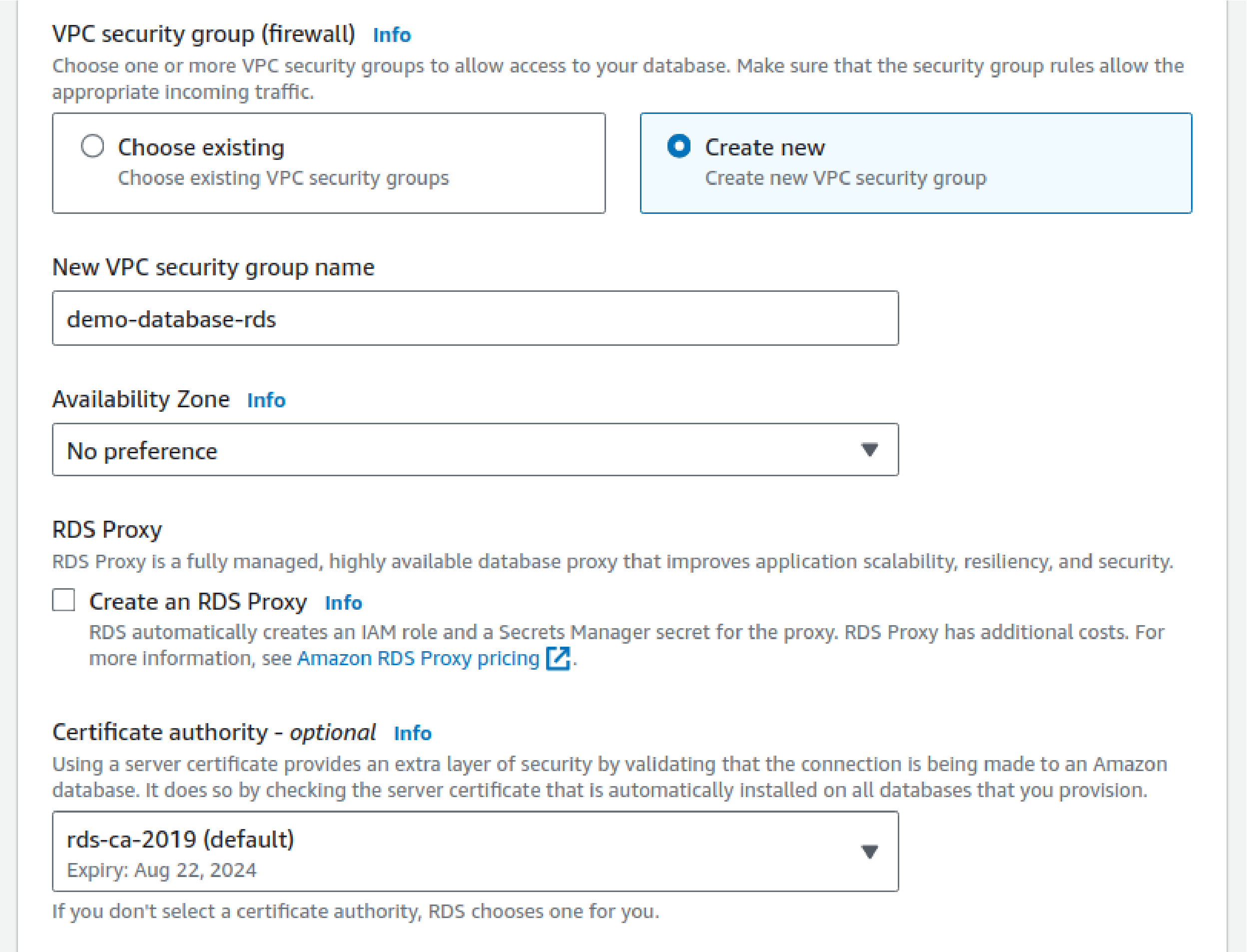
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

default

Public access

● Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

○ No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.



VPC security group (firewall) Info

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

○ Choose existing
Choose existing VPC security groups

● Create new
Create new VPC security group

New VPC security group name

demo-database-rds

Availability Zone Info

No preference ▼

RDS Proxy

RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

□ Create an RDS Proxy Info

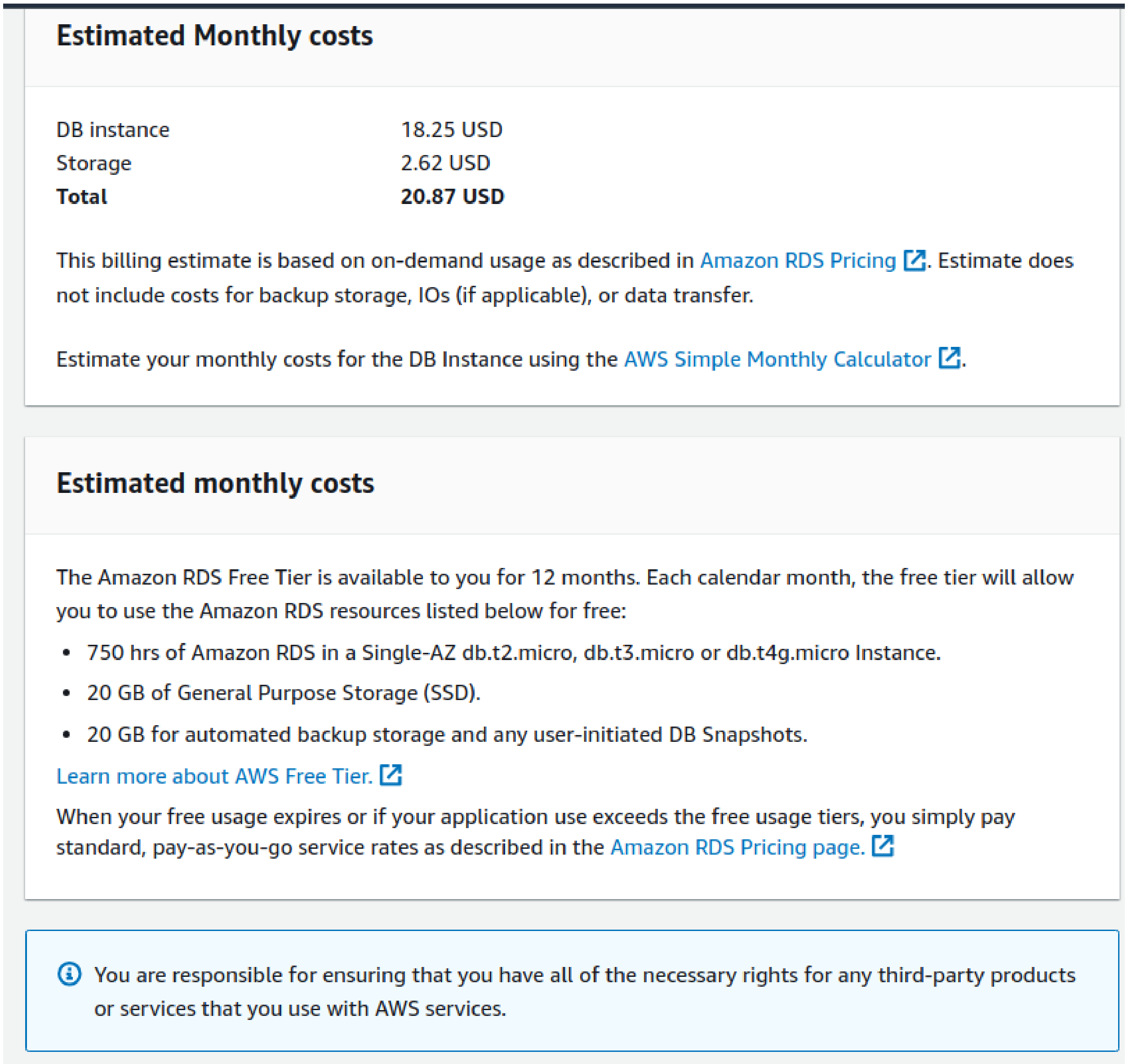
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see Amazon RDS Proxy pricing.

Certificate authority - *optional* Info

Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-2019 (default) ▼
Expiry: Aug 22, 2024

If you don't select a certificate authority, RDS chooses one for you.



Estimated Monthly costs

DB instance	18.25 USD
Storage	2.62 USD
Total	**20.87 USD**

This billing estimate is based on on-demand usage as described in Amazon RDS Pricing. Estimate does not include costs for backup storage, IOs (if applicable), or data transfer.

Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator.

Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

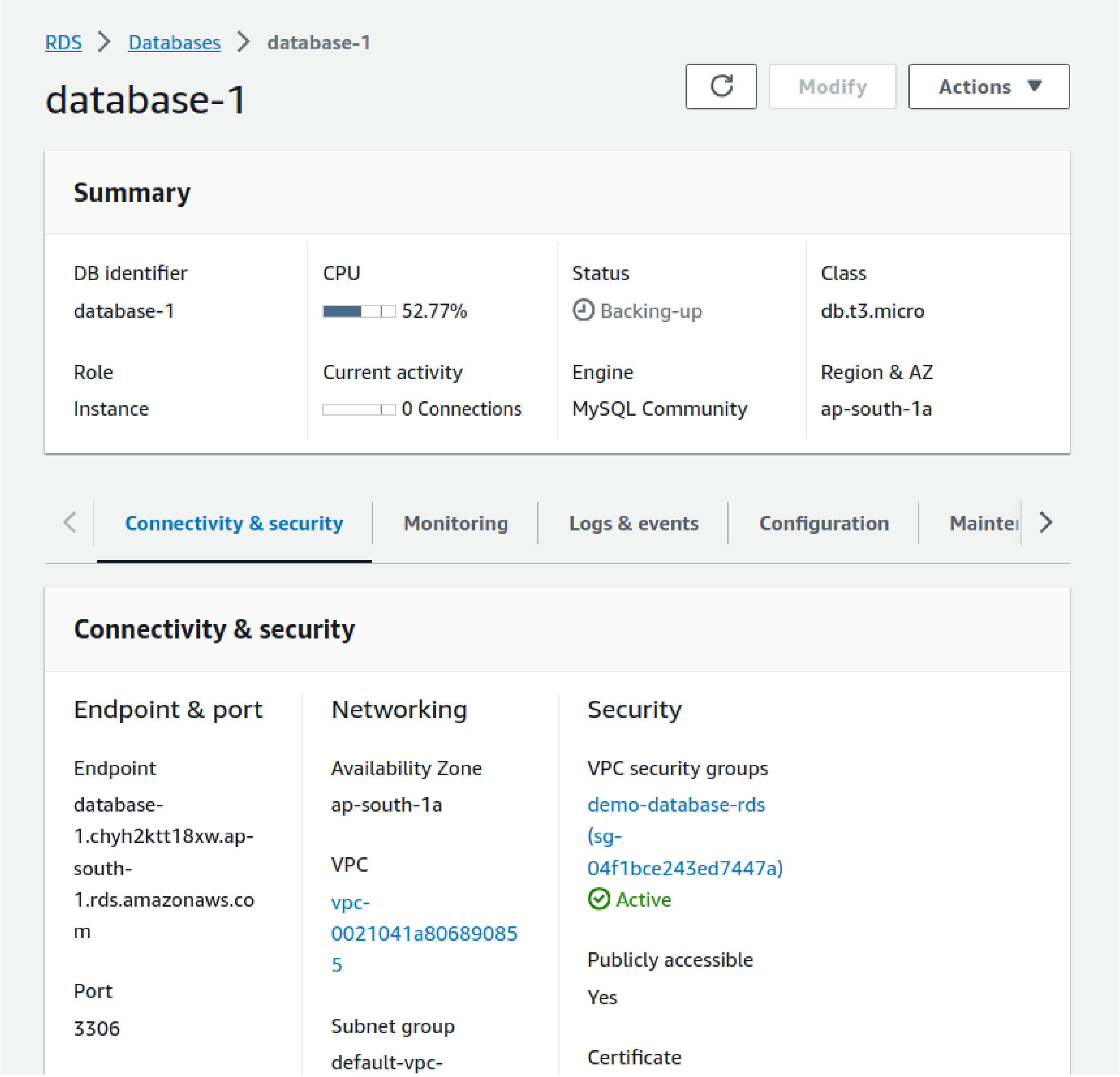
- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

Learn more about AWS Free Tier.

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the Amazon RDS Pricing page.

ⓘ You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

- Create the database. Once created:



database-1

Summary

DB identifier	CPU	Status	Class
database-1	52.77%	Backing-up	db.t3.micro
Role	**Current activity**	**Engine**	**Region & AZ**
Instance	0 Connections	MySQL Community	ap-south-1a

Connectivity & security

Endpoint & port

Endpoint
database-1.chyh2ktt18xw.ap-south-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
ap-south-1a

VPC
vpc-0021041a806890855

Subnet group
default-vpc-

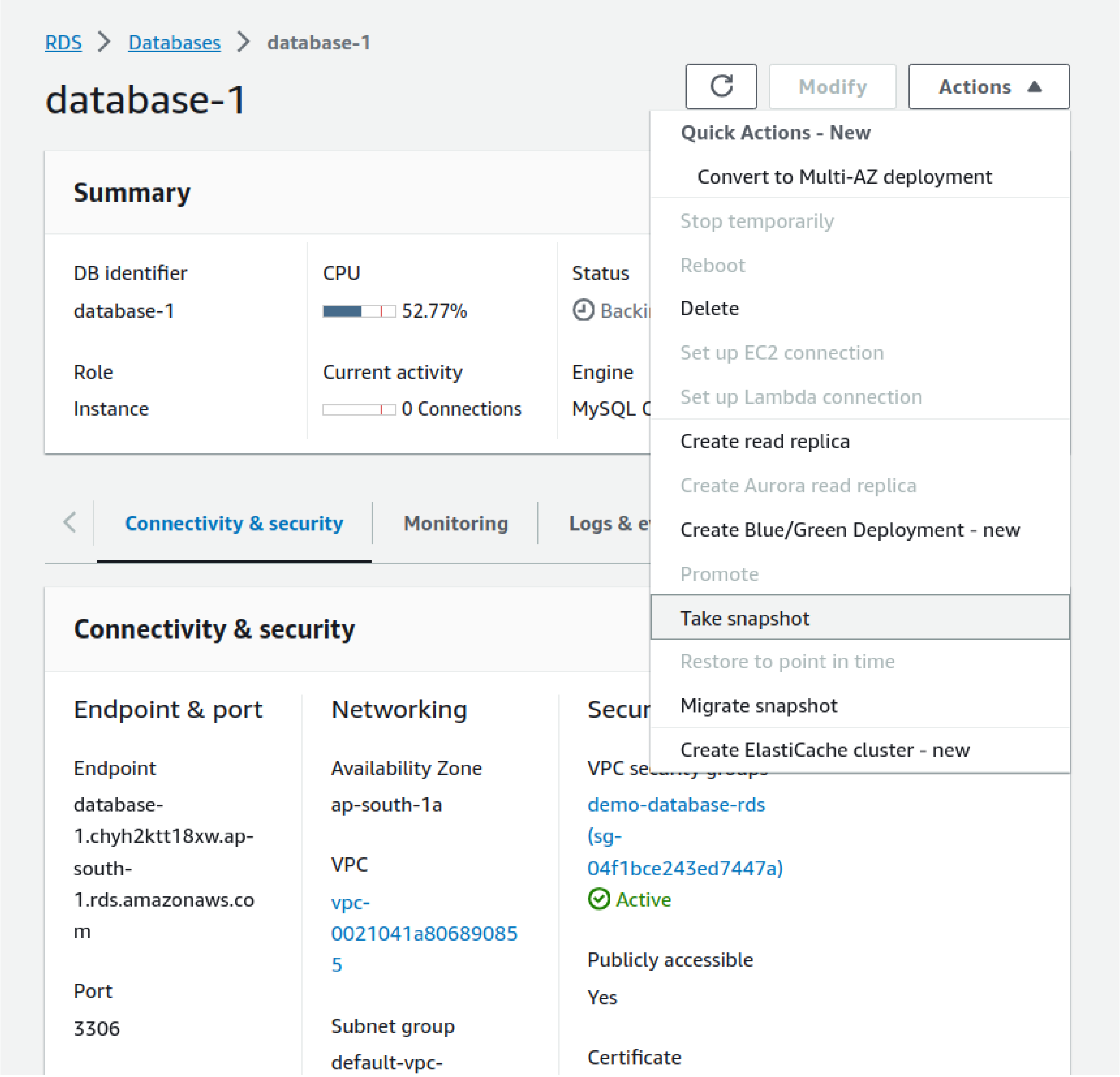
Security

VPC security groups
demo-database-rds (sg-04f1bce243ed7447a)
Active

Publicly accessible
Yes

Certificate

- You can also take a snapshot of the DB which will allow you to restore the snapshot into another DB.



database-1

Summary

DB identifier
database-1

CPU
52.77%

Status
Backi

Role
Instance

Current activity
0 Connections

Engine
MySQL C

Connectivity & security

Endpoint & port

Endpoint
database-1.chyh2ktt18xw.ap-south-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
ap-south-1a

VPC
vpc-00210941a806890855

Subnet group
default-vpc-

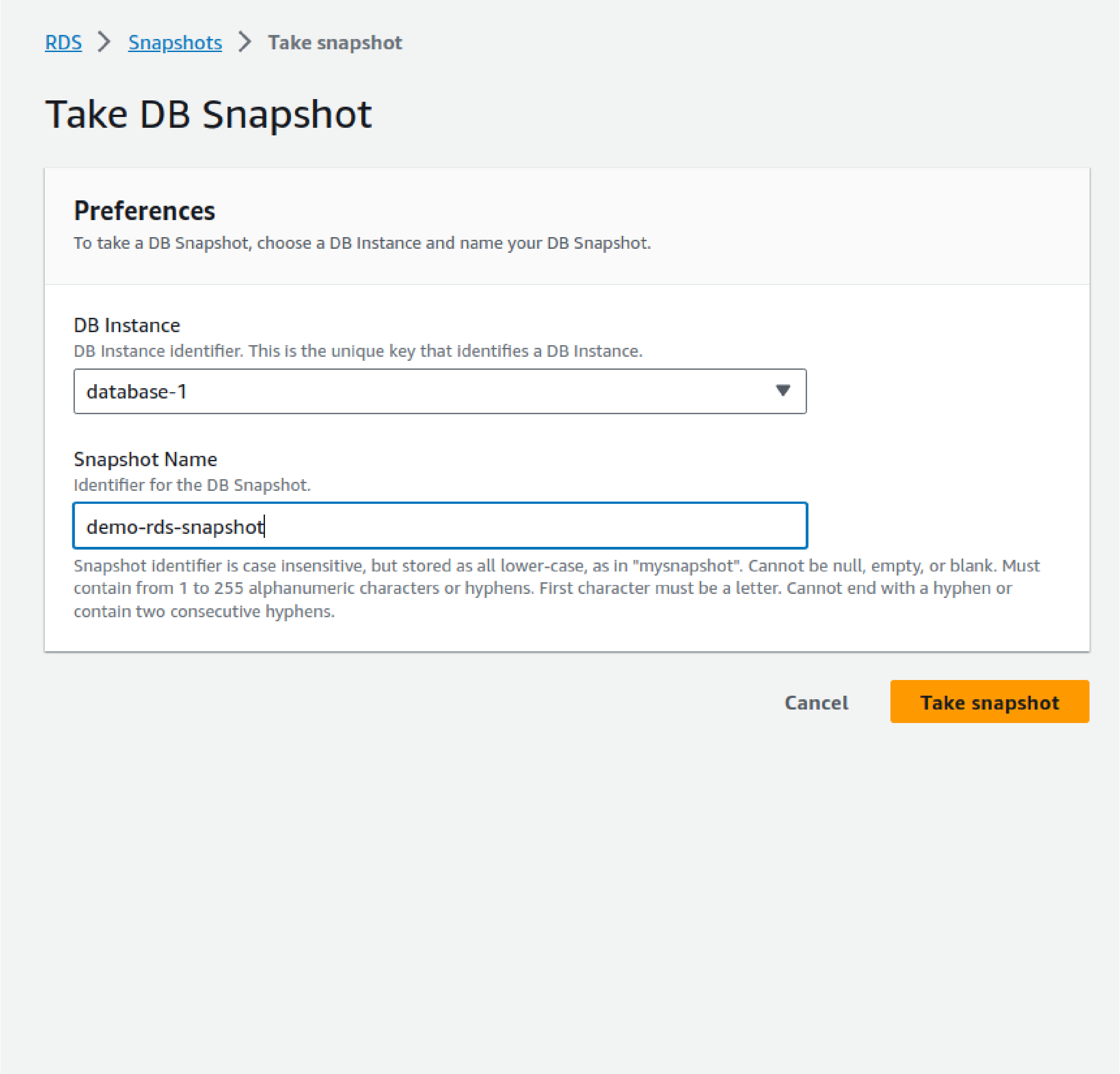
Secur

VPC se

demo-database-rds (sg-04f1bce243ed7447a)
Active

Publicly accessible
Yes

Certificate



Take DB Snapshot

Preferences

To take a DB Snapshot, choose a DB Instance and name your DB Snapshot.

DB Instance

DB Instance identifier. This is the unique key that identifies a DB Instance.

database-1

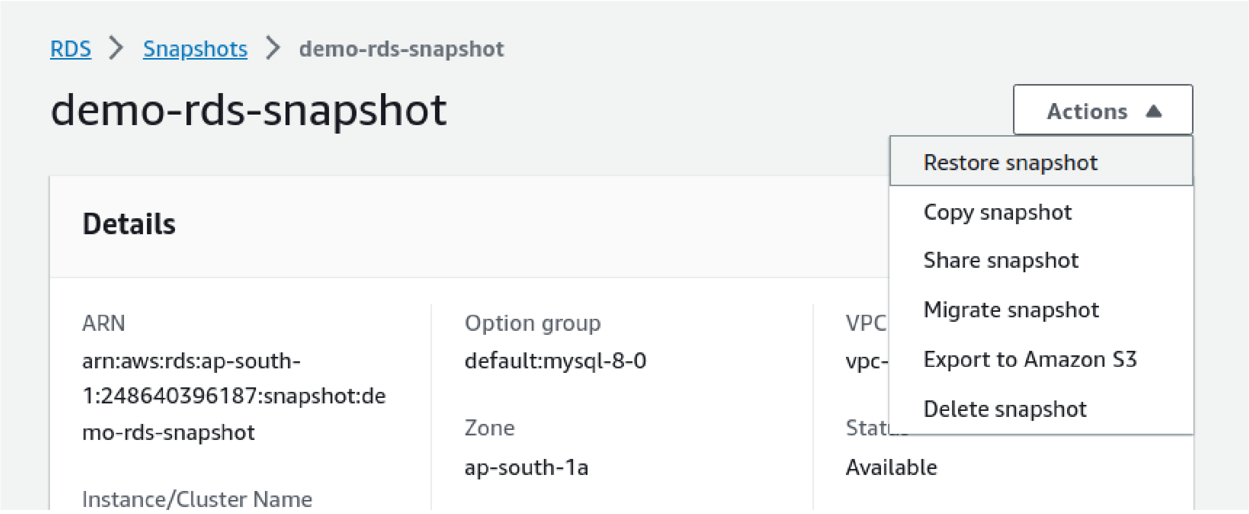
Snapshot Name

Identifier for the DB Snapshot.

demo-rds-snapshot

Snapshot identifier is case insensitive, but stored as all lower-case, as in "mysnapshot". Cannot be null, empty, or blank. Must contain from 1 to 255 alphanumeric characters or hyphens. First character must be a letter. Cannot end with a hyphen or contain two consecutive hyphens.

- Restoring snapshot option will let you create a new database out of the snapshot. This will enable you to recover data.



demo-rds-snapshot

Details

ARN
arn:aws:rds:ap-south-1:248640396187:snapshot:demo-rds-snapshot

Instance/Cluster Name

Option group
default:mysql-8-0

Zone
ap-south-1a

VPC
vpc-

Statu

Available

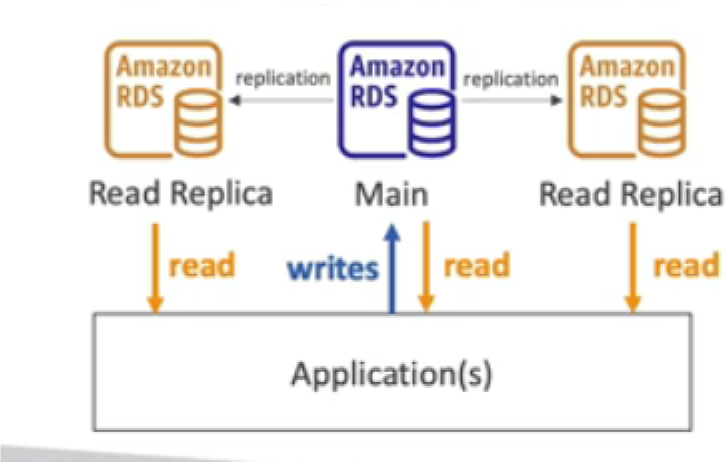
- Copy snapshot will allow you to restore the database into another region in AWS in case of disaster recovery.
- Share snapshot allows you to share your snapshot with others with which they can restore the database using your snapshot.

RDS Deployment Options

- While deploying RDS, you need to understand the multiple architectural choices you have which you can use.

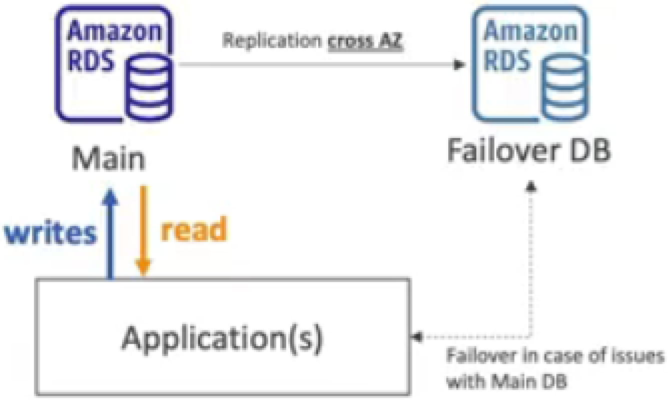
1. **RDS Read Replica:**

- Imagine your application has a lot of read operations to perform from the database, so, to scale your reads, you can create read replicas of your main amazon RDS.
- How this would work is, you can create **up to 15 read replicas** of your main database and then your applications will be able to read through any of those. This will allow load balancing and read scaling.
- Keep note that the application can still only write to the main database.



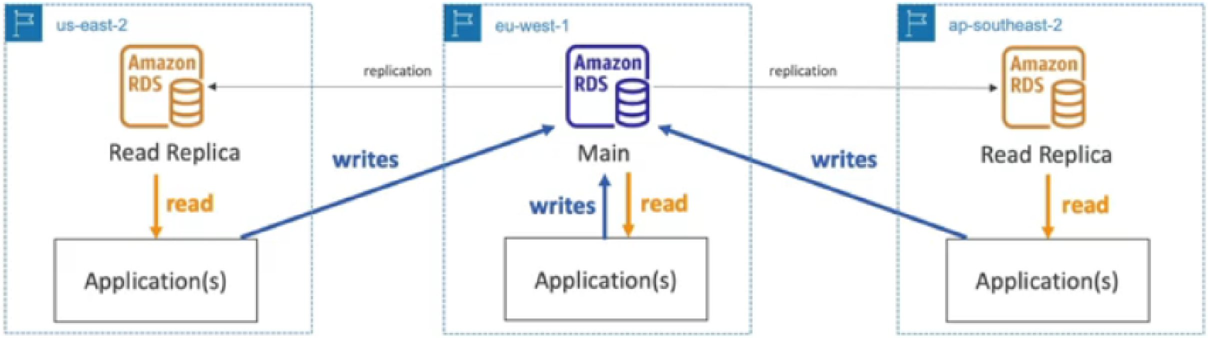
2. **Multi-AZ:**

- Multi AZ servers the application in case the main RDS fails.
- We have another replication of the main RDS set up in another AZ, which takes over in case of failure.
- The failover DB is passive until the main RDS is working fine. Once, the main RDS fails, the failover DB becomes the main RDS.
- The application in this case reads and writes to the main RDS only.
- You can only have the failover DB set up in one more AZ.



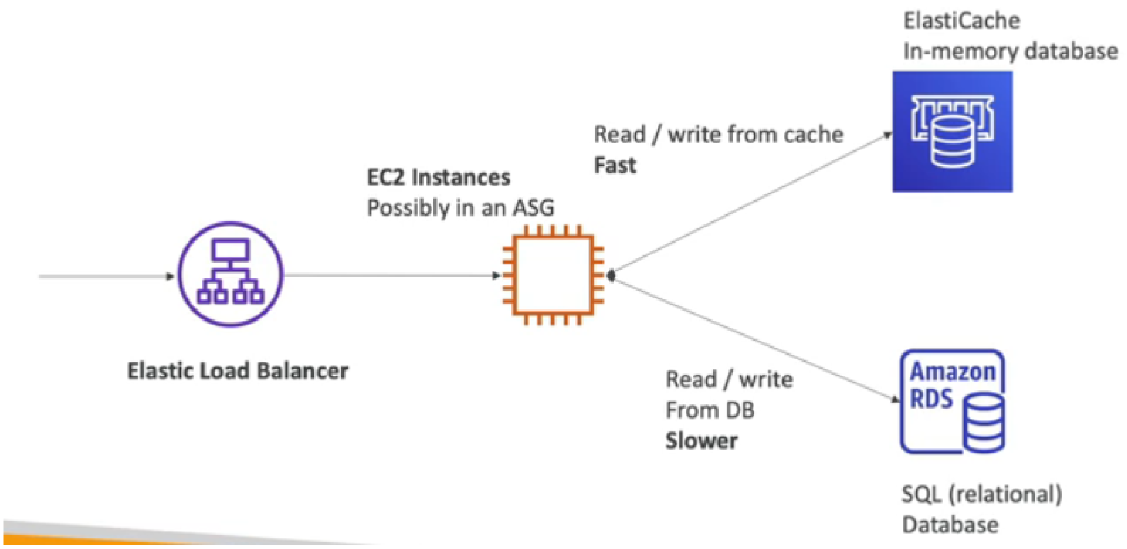
3. **Multi-region replicas**

- Replicas exists in multiple regions.
- This allows the applications to read from any (either main rds or the replicas) which enables load balancing BUT the applications can write only to the main RDS.
- This ensures disaster recovery in case of a regional crisis.
- Better read performance for global reads.



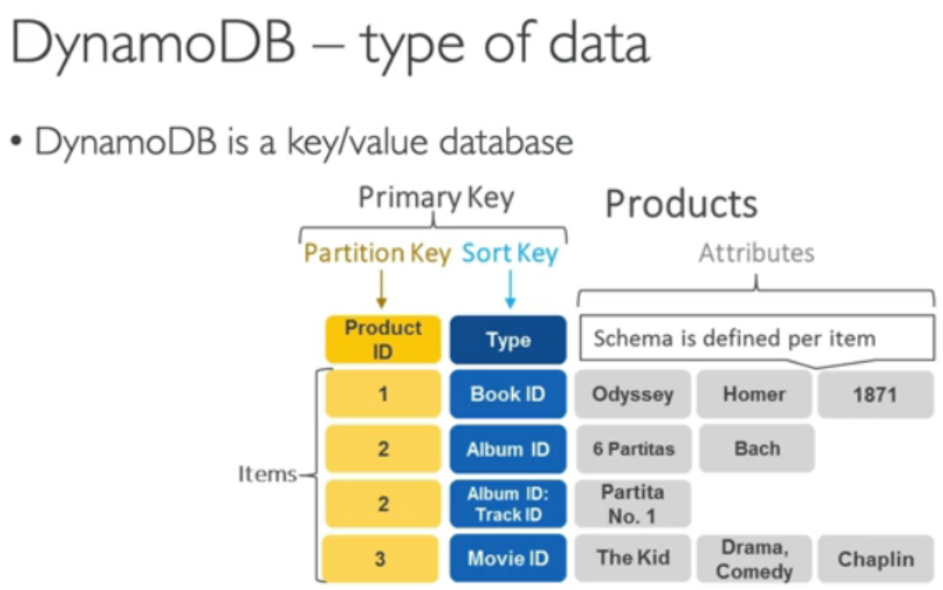
ElastiCache

- ElastiCache is to get managed Redis or Memcached databases.
- Redis and Memcached are both in-memory data stores, commonly used for caching and improving the performance of web applications by storing frequently accessed data in memory.
- It helps reduce load off databases (like RDS) for read intensive workloads by storing frequently accessed data in cache (or more such techniques) and so you don’t need to query the RDS for fetching the same data over and over again.
- AWS takes care of all the maintenance, setup, configuration, etc.



DynamoDB

- A fully managed highly available NoSQL (non relational) database with replication across 3 AZs.
- One of the flagship services of amazon.
- Serverless database.
- Scalable (auto scalable even) in case of massive workloads.
- Can process millions of requests per second and retrieve 100s of TB of storage.
- Low cost
- Integrated with IAM for security and administration.
- Low latency, serverless, high performance, NoSQL
- Has standard and IA storage table class so you can save costs.



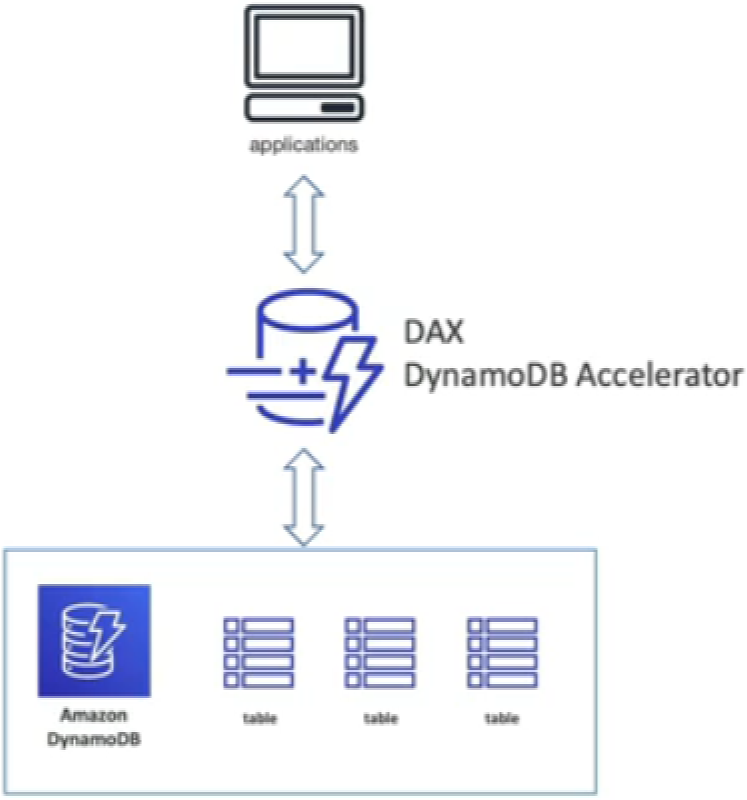
DynamoDB – type of data

- DynamoDB is a key/value database

Product ID	Type	Schema is defined per item		
1	Book ID	Odyssey	Homer	1871
2	Album ID	6 Partitas	Bach	
2	Album ID: Track ID	Partita No. 1		
3	Movie ID	The Kid	Drama, Comedy	Chaplin

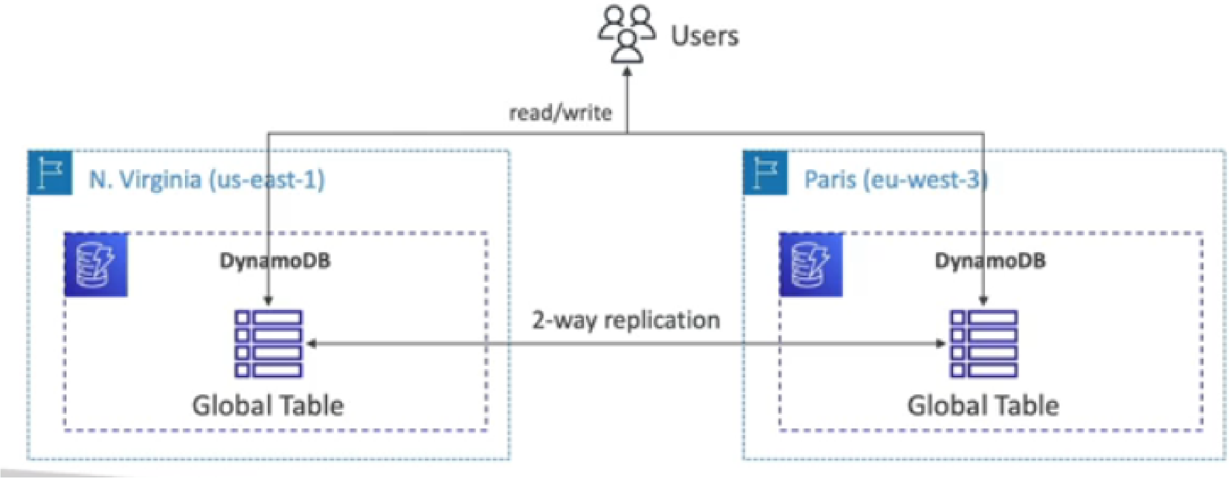
DynamoDB Accelerator - DAX

- DAX is a fully managed in-memory cache for DynamoDB.
- It is tailored to take loads off DynamoDB only. It cannot be used within any other aws service.
- DAX is basically ElastiCache but specialized for DynamoDB
- Improves DynamoDB performance by over 10x.
- Secure and highly scalable



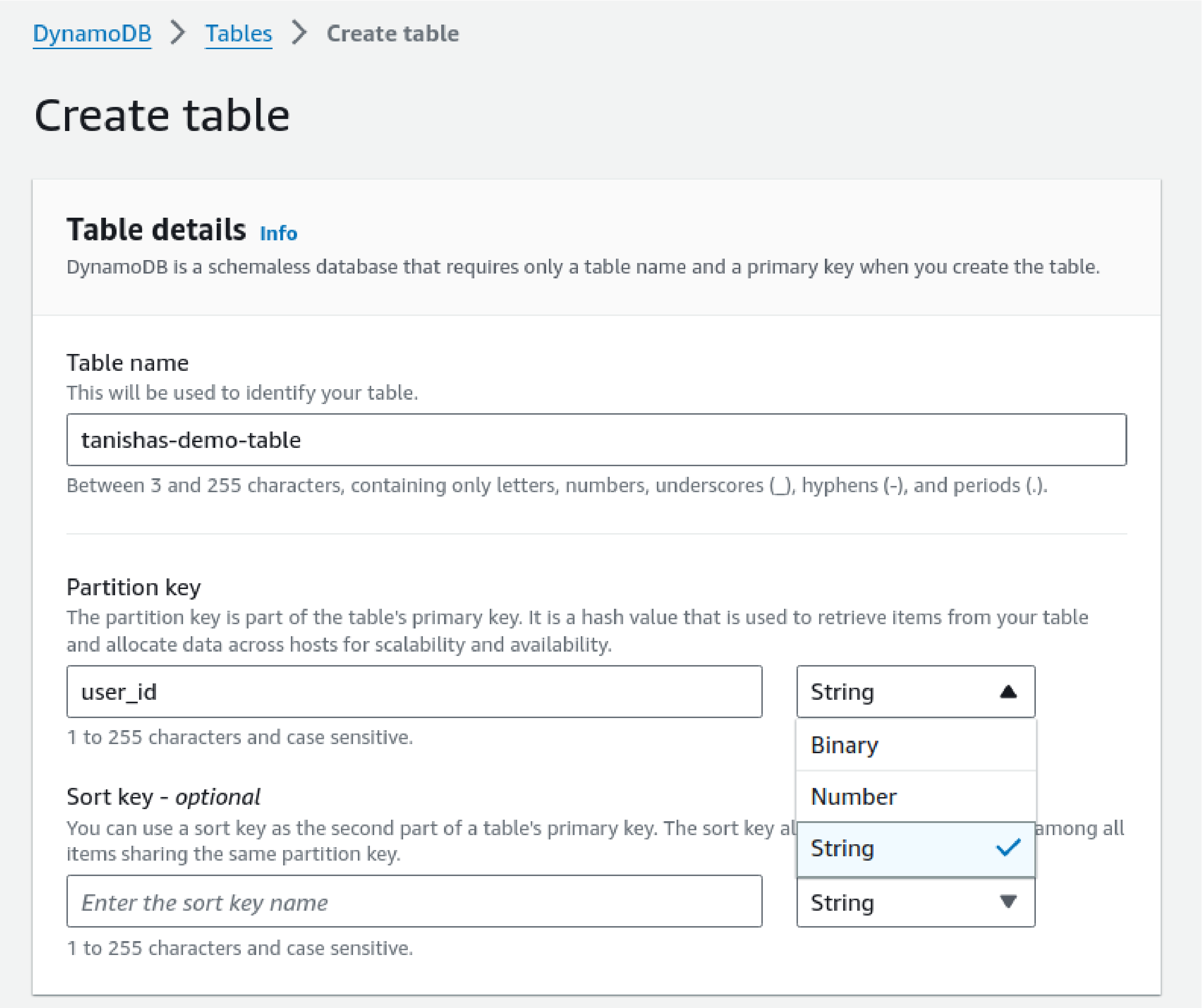
DynamoDB Global Tables

- A feature of DynamoDB that makes a table accessible in multiple regions that too with low latency.
- Enables 2-way replication in different regions.
- This allows the users to read and write in any region in real time. This makes it an active-active replication.
- MAKES DYNAMODB A TRULY GLOBAL DB.



DynamoDB Hands-on

- Create a table



DynamoDB > Tables > Create table

Create table

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

tanishas-demo-table

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

user_id | String

1 to 255 characters and case sensitive.

Sort key - *optional*

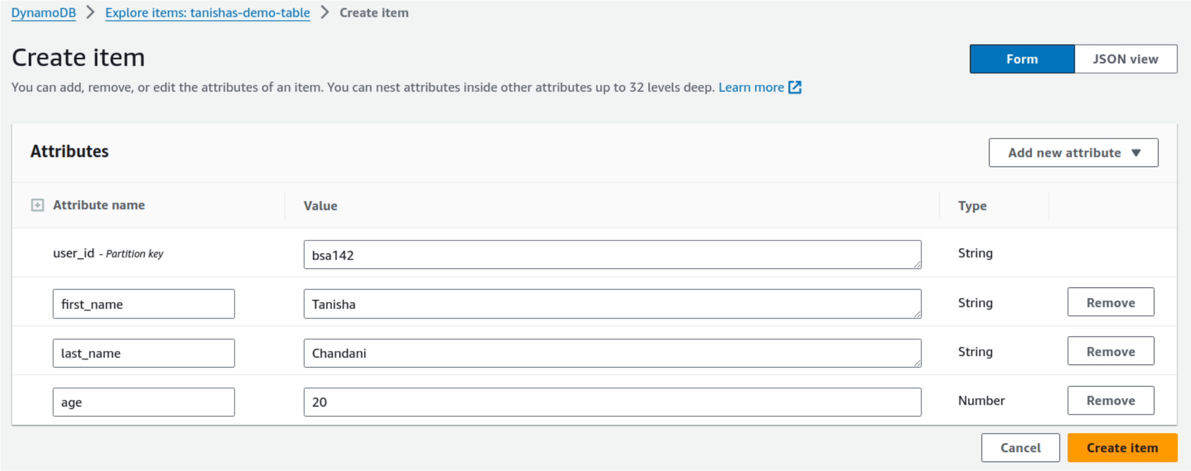
You can use a sort key as the second part of a table's primary key. The sort key al among all items sharing the same partition key.

Enter the sort key name | String

1 to 255 characters and case sensitive.

- One thing to note here is that we are creating a table without creating a database, this is because the database is already created (serverless) dynamoDB.
- It's like saying I want this table to be created and idc how it runs — dynamoDB. This is powerful.
- Inserting dummy data into the table

  DynamoDB tables >> open >> explore items>> create item (create 2-3 items)

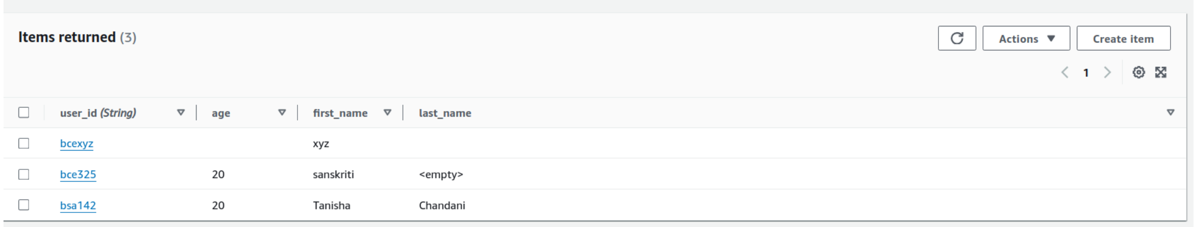


Create item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more

Attributes

Attribute name	Value	Type
user_id - *Partition key*	bsa142	String
first_name	Tanisha	String
last_name	Chandani	String
age	20	Number



Items returned (3)

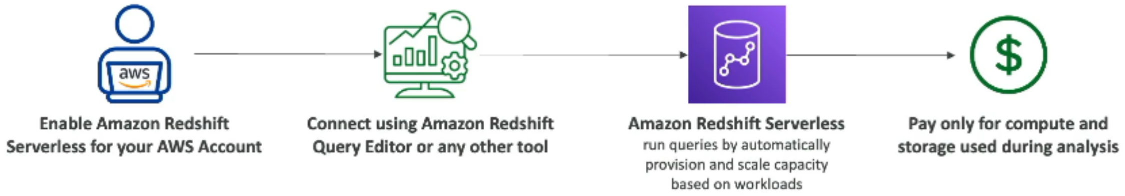
□	user_id *(String)*	age	first_name	last_name
□	bcexyz		xyz	
□	bce325	20	sanskriti	<empty>
□	bsa142	20	Tanisha	Chandani

- DynamoDB is super flexible with data values, if you don't put any info for age and last name for bcexyz user id, it still accepts it as valid.
- The KEY DIFFERENCE between dynamoDB and RDS is that dynamoDB accepts all data as values in a table and there is no way to link one table to another, whereas in RDS you can have relations between tables (because obviously).

Redshift

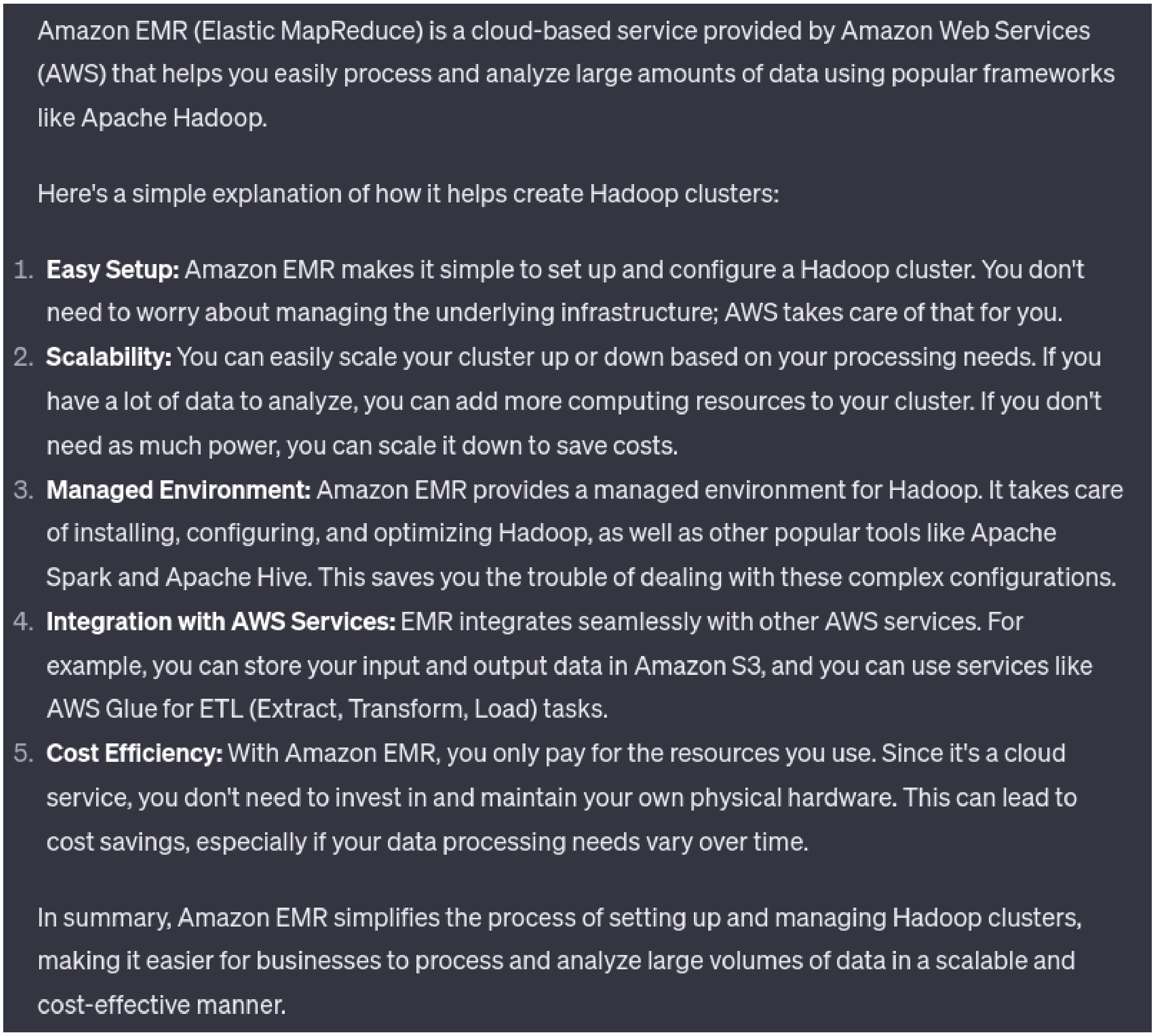
- Redshift is a database based on PostgreSQL.
- It is used for OLAP (online analytical processing) — data warehousing and analysis of the warehoused data, hence, it performs 10x better than any other data warehouse.
- Not used for OLTP (online transactions)
- It is a columnar storage (not row-based)

- It also has massively parallel query execution (MPP) and uses SQL for querying through the database
- Can be integrated with business intelligence tools
- Redshift Serverless database
  - no need to manually provision or scale data.
  - you can basically analyse all the warehoused data without having to worry about the infrastructure


aws
Enable Amazon Redshift Serverless for your AWS Account
Connect using Amazon Redshift Query Editor or any other tool
Amazon Redshift Serverless
run queries by automatically provision and scale capacity based on workloads
$
Pay only for compute and storage used during analysis

Amazon EMR (Elastic MapReduce)

- EMR is not a database itself but a tool that helps to create Hadoop clusters.



Amazon EMR (Elastic MapReduce) is a cloud-based service provided by Amazon Web Services (AWS) that helps you easily process and analyze large amounts of data using popular frameworks like Apache Hadoop.

Here's a simple explanation of how it helps create Hadoop clusters:

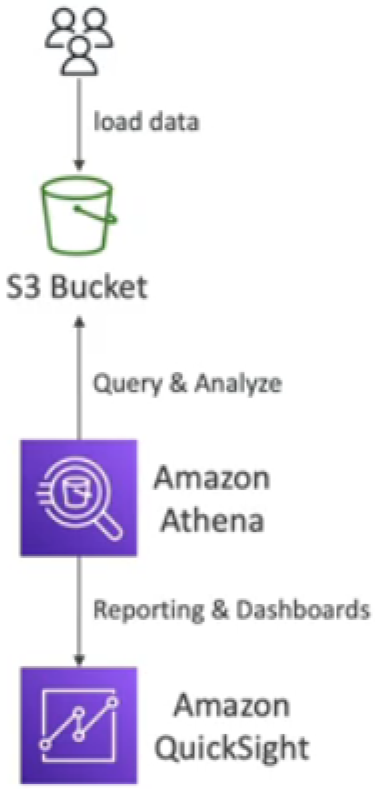
1. **Easy Setup:** Amazon EMR makes it simple to set up and configure a Hadoop cluster. You don't need to worry about managing the underlying infrastructure; AWS takes care of that for you.
2. **Scalability:** You can easily scale your cluster up or down based on your processing needs. If you have a lot of data to analyze, you can add more computing resources to your cluster. If you don't need as much power, you can scale it down to save costs.
3. **Managed Environment:** Amazon EMR provides a managed environment for Hadoop. It takes care of installing, configuring, and optimizing Hadoop, as well as other popular tools like Apache Spark and Apache Hive. This saves you the trouble of dealing with these complex configurations.
4. **Integration with AWS Services:** EMR integrates seamlessly with other AWS services. For example, you can store your input and output data in Amazon S3, and you can use services like AWS Glue for ETL (Extract, Transform, Load) tasks.
5. **Cost Efficiency:** With Amazon EMR, you only pay for the resources you use. Since it's a cloud service, you don't need to invest in and maintain your own physical hardware. This can lead to cost savings, especially if your data processing needs vary over time.

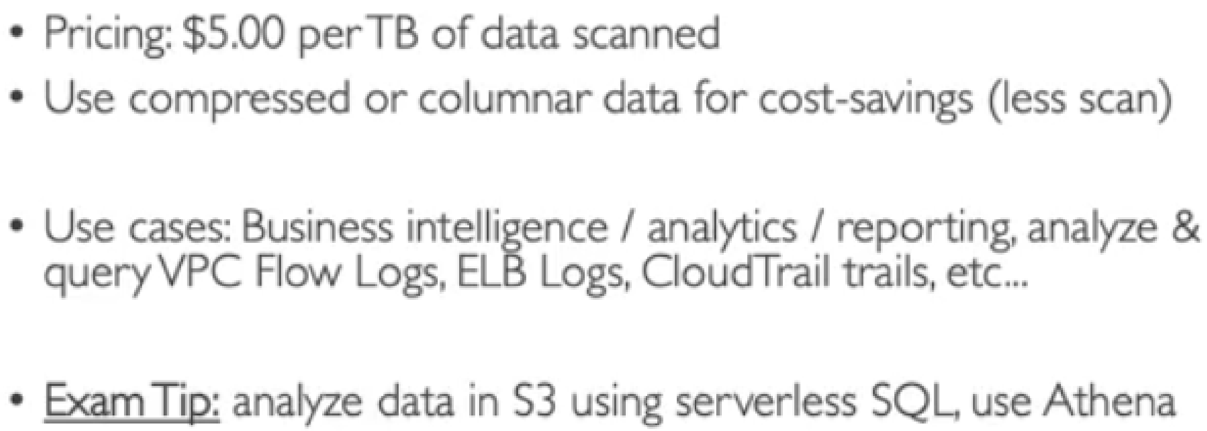
In summary, Amazon EMR simplifies the process of setting up and managing Hadoop clusters, making it easier for businesses to process and analyze large volumes of data in a scalable and cost-effective manner.

- Using EMR, you can create clusters of hundreds of EC2 instances that work together to analyse big data.
- It also supports Apache spark, HBase, etc.
- It allows autoscaling and can be integrated with spot instances.
- exam keyword: hadoop cluster = amazon emr

Amazon Athena

- Amazon athena is a serverless query engine that works to perform analytics for objects stored in S3.
- It is built on presto.
- It uses SQL to query the files or objects and supports CSV, Json, Avro, etc.
- If you want, you can also integrate the whole process with amazon QuickSight service for reporting the analysed data through a dashboard.





- Pricing: $5.00 per TB of data scanned
- Use compressed or columnar data for cost-savings (less scan)
- Use cases: Business intelligence / analytics / reporting, analyze & query VPC Flow Logs, ELB Logs, CloudTrail trails, etc...
- Exam Tip: analyze data in S3 using serverless SQL, use Athena

- Keywords for exam - serverless data analytics using SQL, s3 objects data analytics

Amazon QuickSight

- In simple words, this service lets you create dashboards on your databases which makes reading or drawing conclusions from the analysed data easier.
- Helps you visualize data with graphs, charts, etc.
- Pricing is per-session, you don’t have to provision any servers.
- use cases: BI, BA, Data Visualization
- QuickSight can run on top of a lot of databases and storage like S3.

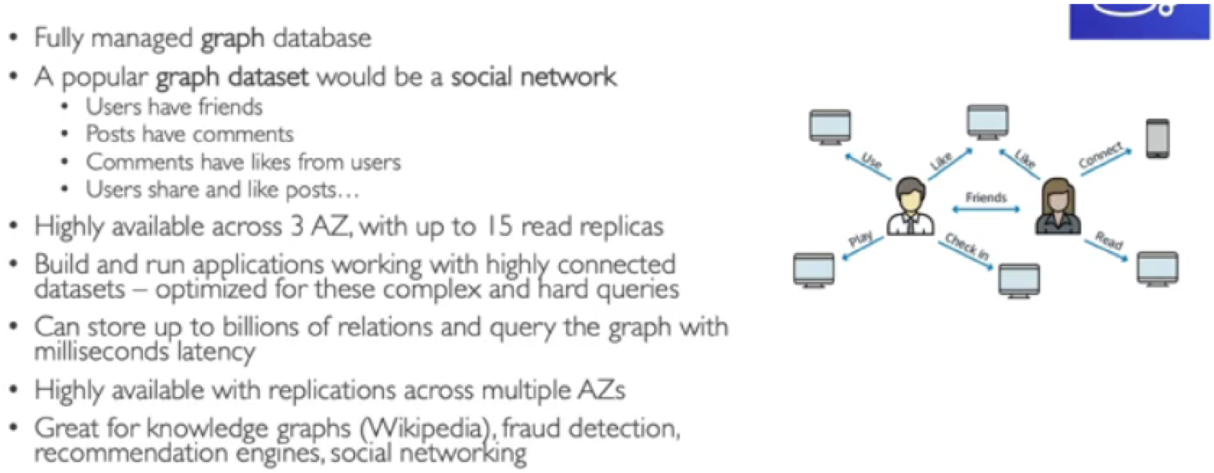
DocumentDB

- What Aurora is for PostgreSQL and MySQL, DocumentDB is for MongoDB.
- It is a cloud native, NoSQL database that enhances the efficiency of MongoDB. MongoDB is used to store, query and index JSON documents.
- Managed by AWS, highly available (replicas across 3 AZ)

- Storage scales automatically with increments of 10GB (up to 64TB)

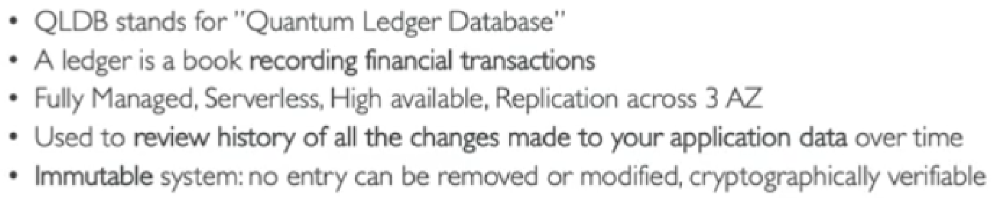
Amazon Neptune

- Graph databases = neptune
- Can handle interconnected, complex, graph-like structured queries.



- Fully managed graph database
- A popular graph dataset would be a social network
  - Users have friends
  - Posts have comments
  - Comments have likes from users
  - Users share and like posts…
- Highly available across 3 AZ, with up to 15 read replicas
- Build and run applications working with highly connected datasets – optimized for these complex and hard queries
- Can store up to billions of relations and query the graph with milliseconds latency
- Highly available with replications across multiple AZs
- Great for knowledge graphs (Wikipedia), fraud detection, recommendation engines, social networking

QLDB (Quantum Ledger Database)

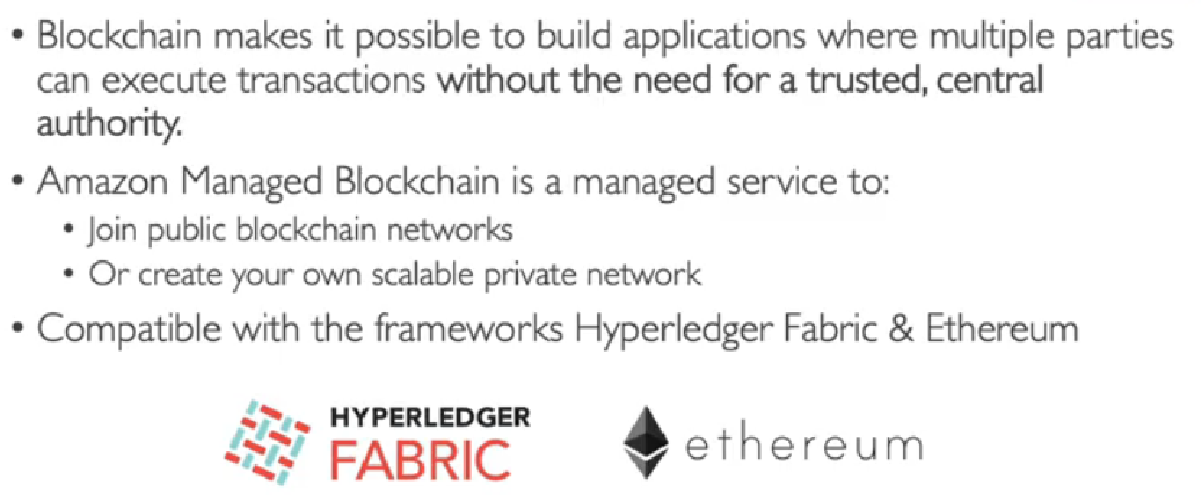


- QLDB stands for "Quantum Ledger Database"
- A ledger is a book recording financial transactions
- Fully Managed, Serverless, High available, Replication across 3 AZ
- Used to review history of all the changes made to your application data over time
- Immutable system: no entry can be removed or modified,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

- It works by maintaining a QLDB journal at the backend, which guarantees nothing has been modified or deleted. (similar to how blockchains work, with hash codes and stuff except QLDB is not decentralized)

- 2-3x better performance than common ledger blockchain frameworks
- You can query data using SQL.
- keywords: financial transactions record, ledger, centralized by aws = QLDB.

Managed Blockchain Overview



- Blockchain makes it possible to build applications where multiple parties can execute transactions **without the need for a trusted, central authority.**
- Amazon Managed Blockchain is a managed service to:
  - Join public blockchain networks
  - Or create your own scalable private network
- Compatible with the frameworks Hyperledger Fabric & Ethereum

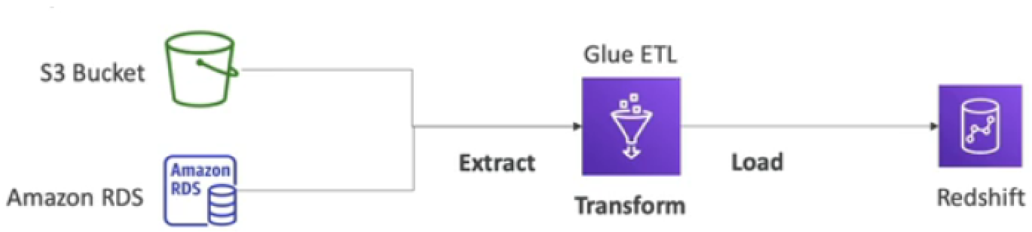
HYPERLEDGER
FABRIC
ethereum

- The major difference between QLDB and Managed blockchain is that QLDB is centralized while this is decentralized.

AWS Glue

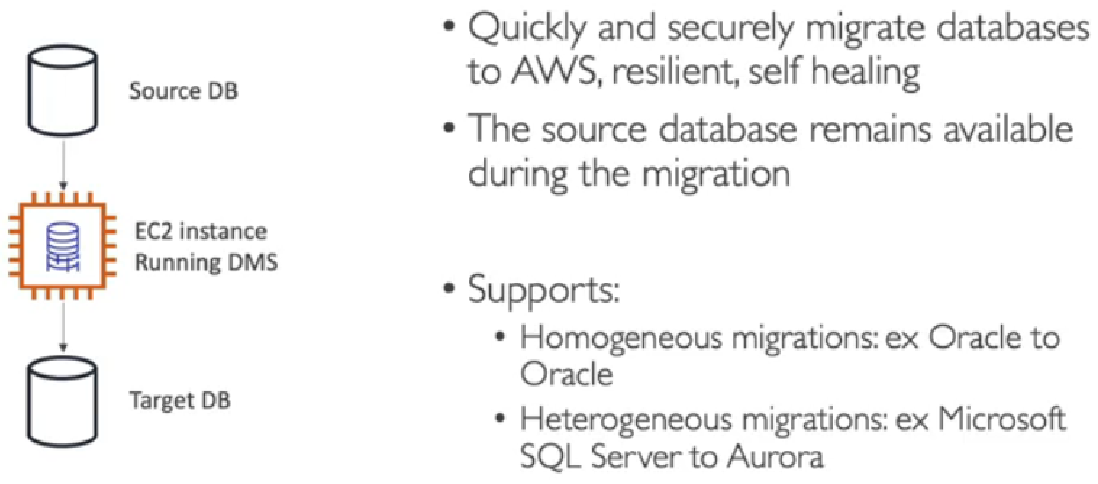
- A managed ETL Service (extract, transform, load)
- ETL is helpful when you have data but it isn’t in the right format or it isn’t prepared enough to have analytics performed on it yet.
- Traditionally you would use servers to do this but with AWS Glue, you can go serverless for the same

- AWS Glue is an ETL service that makes it easy to prepare and load data for analysis.
- For example, suppose you need to **extract** data both from a S3 bucket and RDS, glue would extract those for you then you can write a **transform** script using which Glue will transform the data and then you can **load** it onto an analysis service like redshift which would analyse the data.



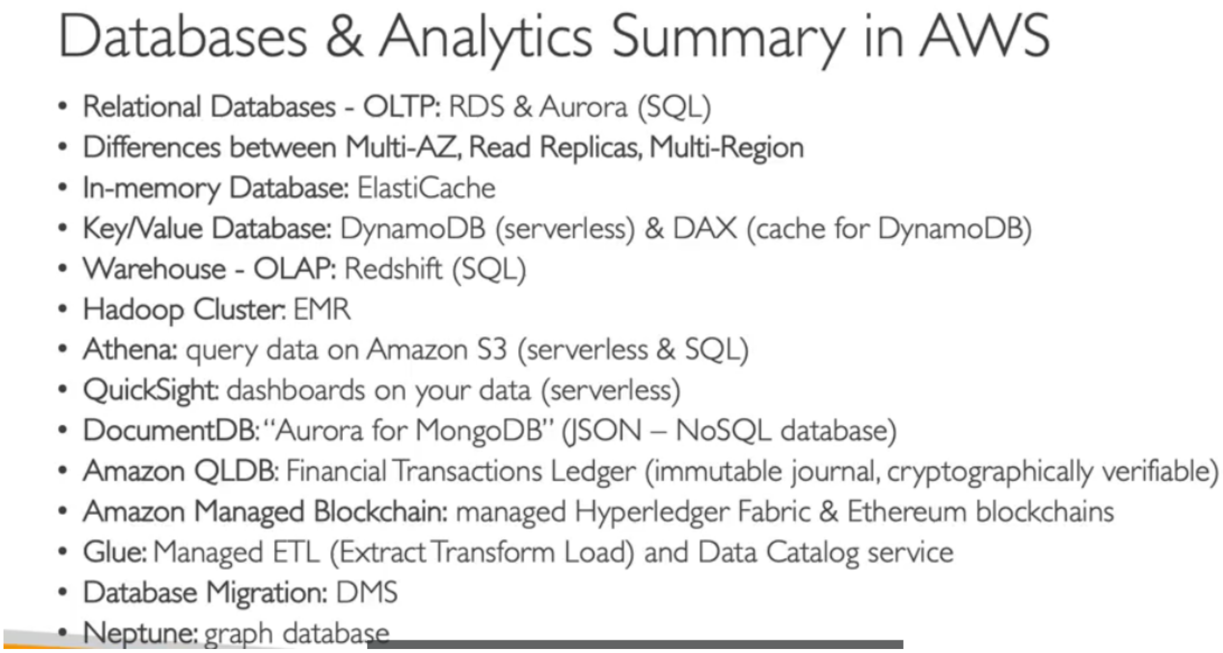
DMS (Database Migration Service)

- To migrate data from one database to another.



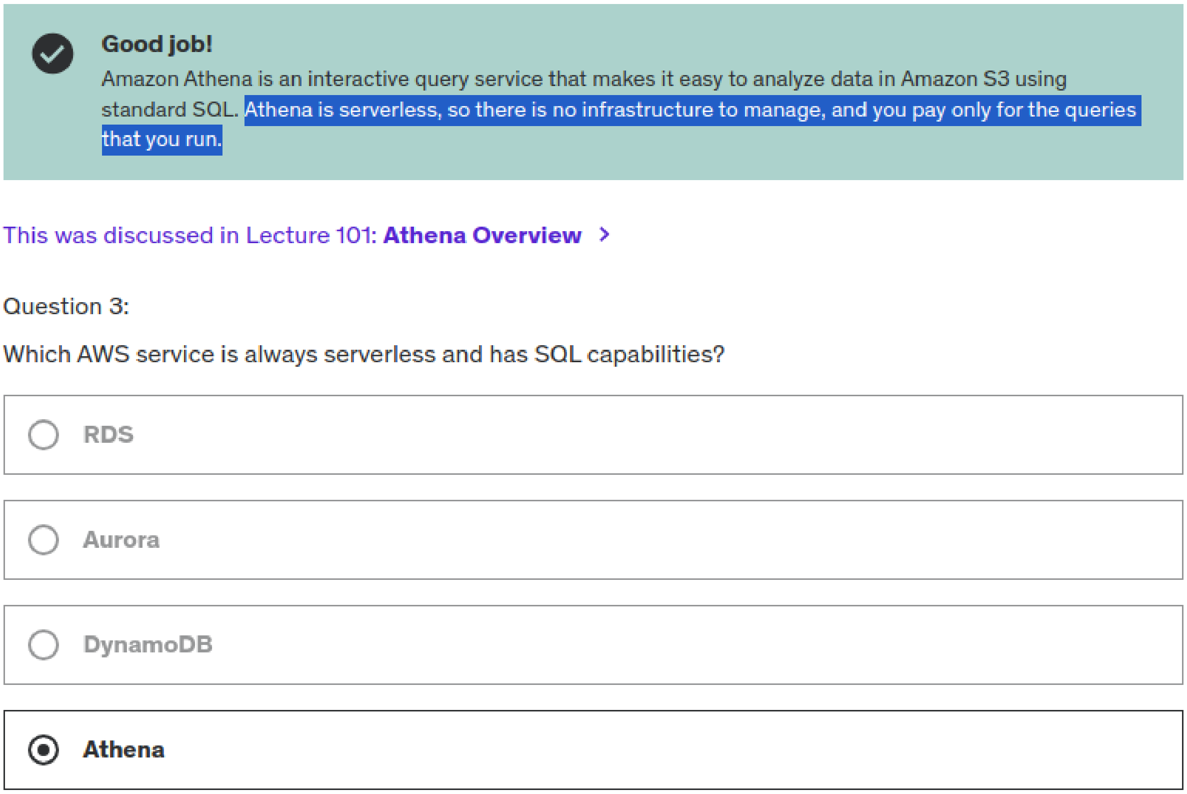
- Quickly and securely migrate databases to AWS, resilient, self healing
- The source database remains available during the migration
- Supports:
  - Homogeneous migrations: ex Oracle to Oracle
  - Heterogeneous migrations: ex Microsoft SQL Server to Aurora

Concluding



Databases & Analytics Summary in AWS

- **Relational Databases - OLTP:** RDS & Aurora (SQL)
- **Differences between Multi-AZ, Read Replicas, Multi-Region**
- **In-memory Database:** ElastiCache
- **Key/Value Database:** DynamoDB (serverless) & DAX (cache for DynamoDB)
- **Warehouse - OLAP:** Redshift (SQL)
- **Hadoop Cluster:** EMR
- **Athena:** query data on Amazon S3 (serverless & SQL)
- **QuickSight:** dashboards on your data (serverless)
- **DocumentDB:** "Aurora for MongoDB" (JSON – NoSQL database)
- **Amazon QLDB:** Financial Transactions Ledger (immutable journal, cryptographically verifiable)
- **Amazon Managed Blockchain:** managed Hyperledger Fabric & Ethereum blockchains
- **Glue:** Managed ETL (Extract Transform Load) and Data Catalog service
- **Database Migration:** DMS
- **Neptune:** graph database



✓ **Good job!**

Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run.

This was discussed in Lecture 101: **Athena Overview** >

Question 3:

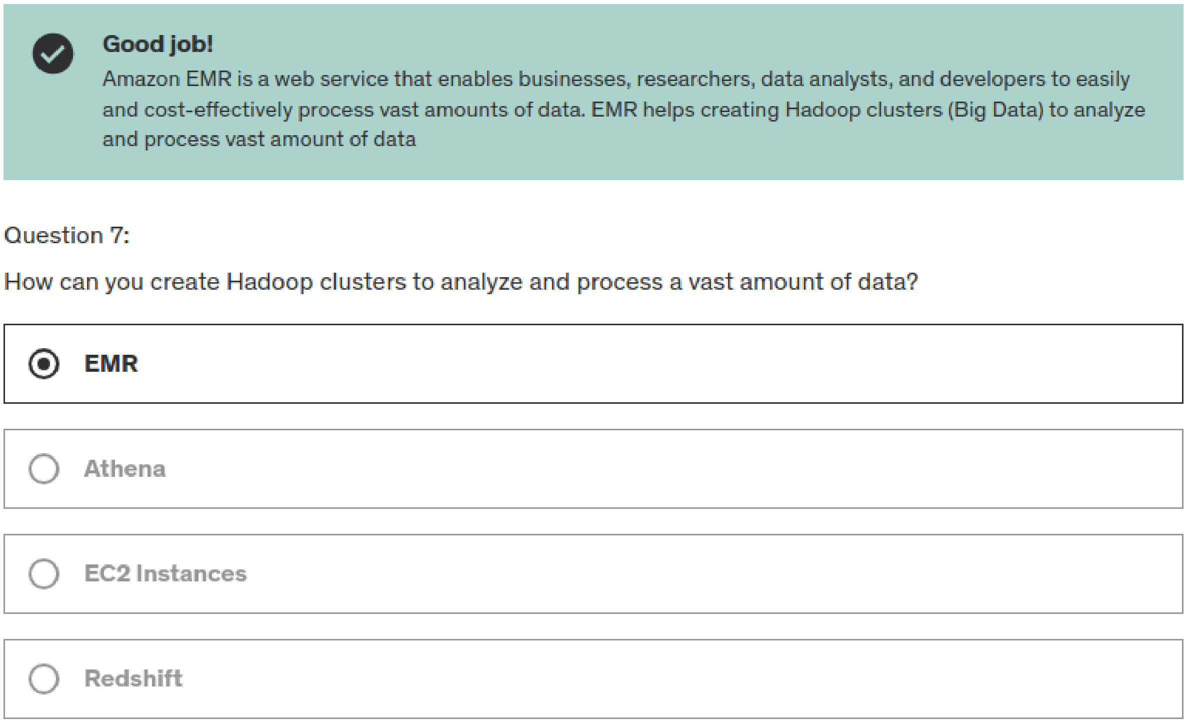
Which AWS service is always serverless and has SQL capabilities?

○ RDS

○ Aurora

○ DynamoDB

● **Athena**

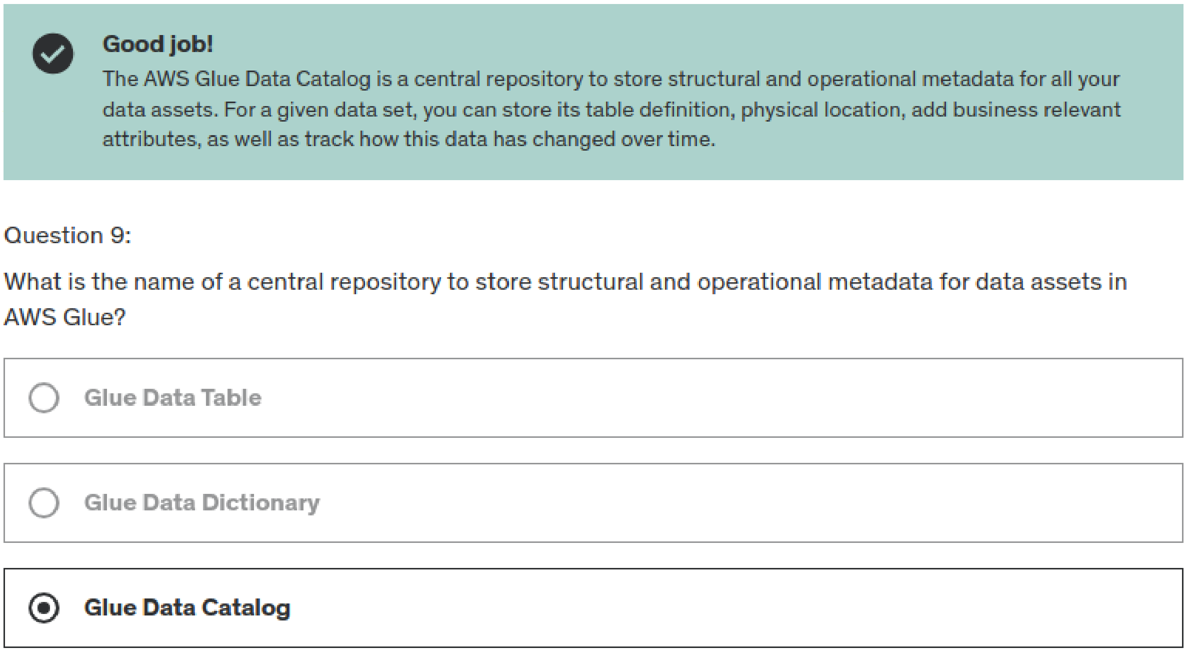


Good job!
Amazon EMR is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data. EMR helps creating Hadoop clusters (Big Data) to analyze and process vast amount of data

Question 7:

How can you create Hadoop clusters to analyze and process a vast amount of data?

- ● **EMR**
- ○ **Athena**
- ○ **EC2 Instances**
- ○ **Redshift**



✓ **Good job!**

The AWS Glue Data Catalog is a central repository to store structural and operational metadata for all your data assets. For a given data set, you can store its table definition, physical location, add business relevant attributes, as well as track how this data has changed over time.

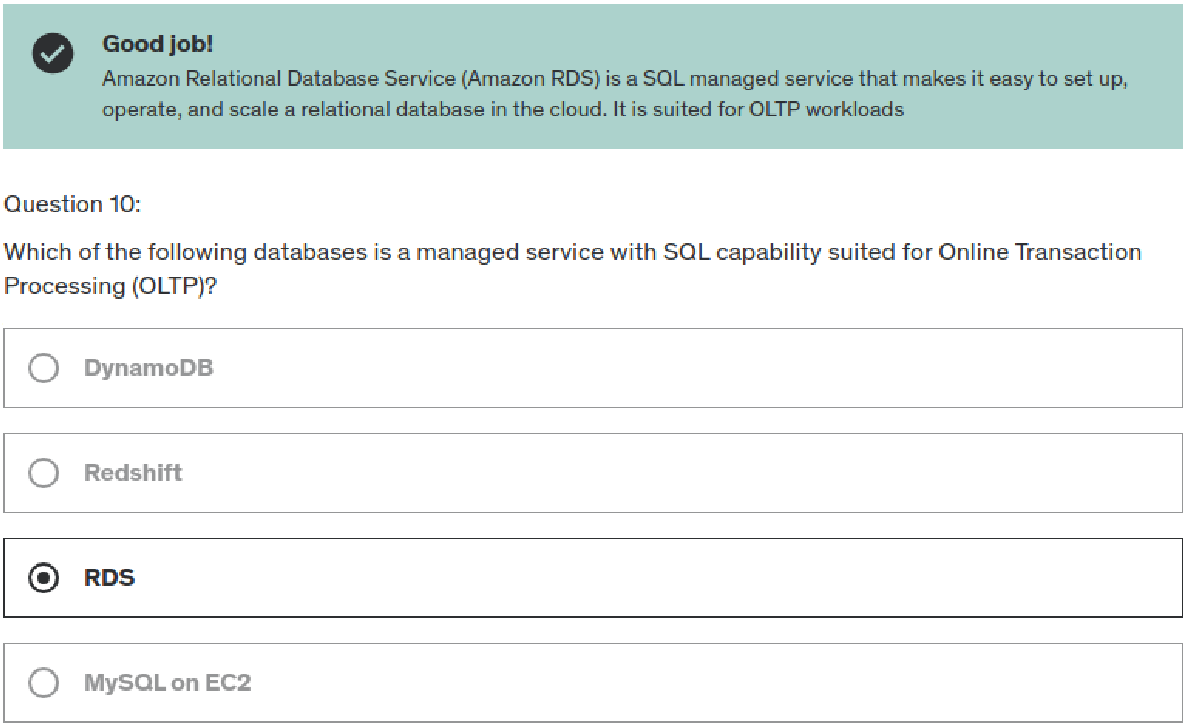
Question 9:

What is the name of a central repository to store structural and operational metadata for data assets in AWS Glue?

○ Glue Data Table

○ Glue Data Dictionary

● **Glue Data Catalog**



✓ **Good job!**

Amazon Relational Database Service (Amazon RDS) is a SQL managed service that makes it easy to set up, operate, and scale a relational database in the cloud. It is suited for OLTP workloads

Question 10:

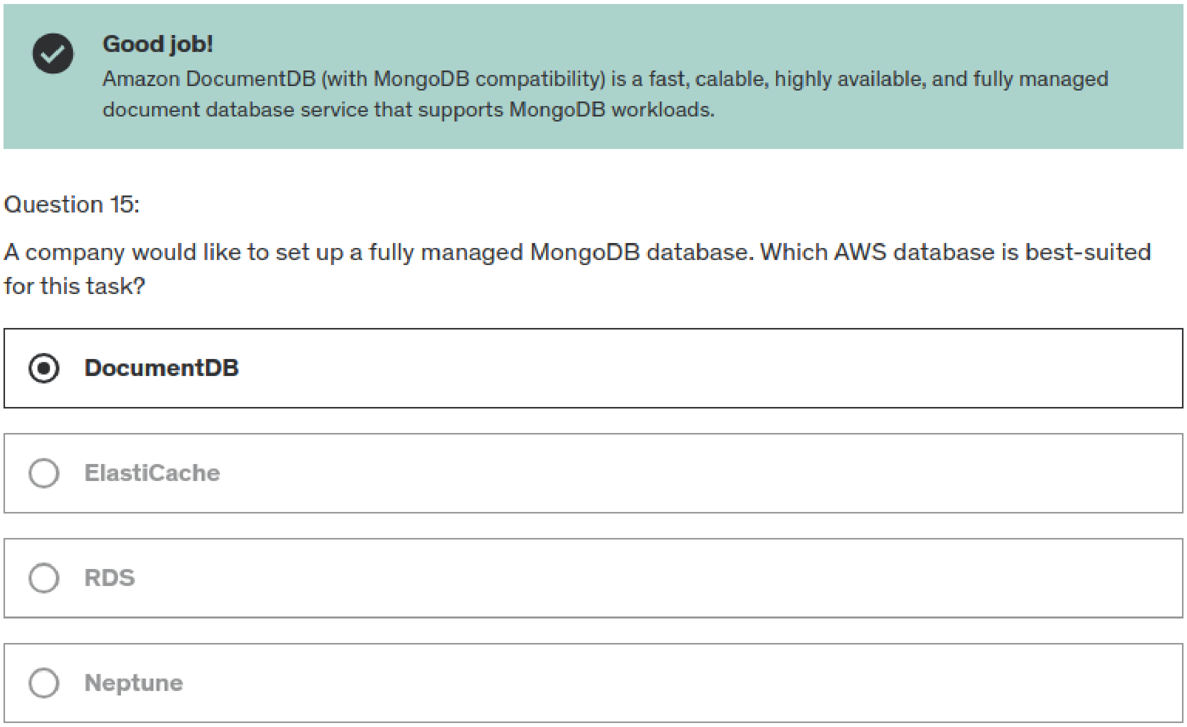
Which of the following databases is a managed service with SQL capability suited for Online Transaction Processing (OLTP)?

○ DynamoDB

○ Redshift

◉ **RDS**

○ MySQL on EC2



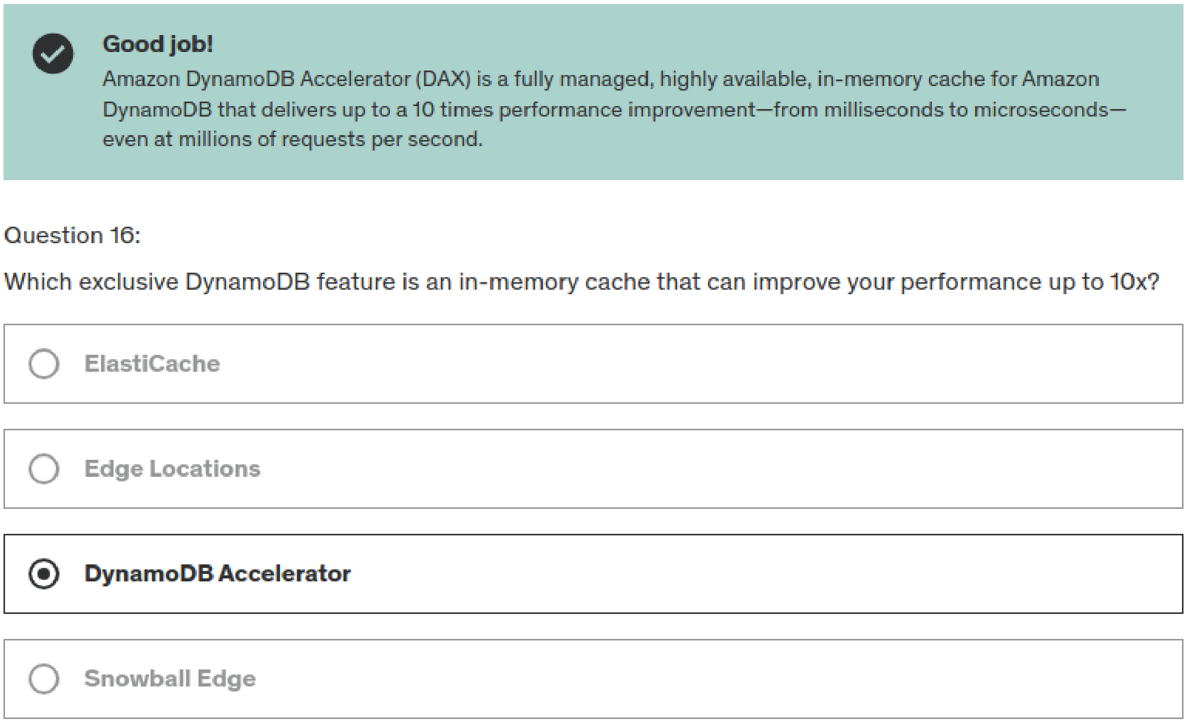
✓ **Good job!**

Amazon DocumentDB (with MongoDB compatibility) is a fast, calable, highly available, and fully managed document database service that supports MongoDB workloads.

Question 15:

A company would like to set up a fully managed MongoDB database. Which AWS database is best-suited for this task?

- ● **DocumentDB**
- ○ ElastiCache
- ○ RDS
- ○ Neptune



✓ **Good job!**
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.

Question 16:

Which exclusive DynamoDB feature is an in-memory cache that can improve your performance up to 10x?

○ ElastiCache

○ Edge Locations

● **DynamoDB Accelerator**

○ Snowball Edge

✓ **Good job!**

RDS Multi-AZ deployments’ main purpose is high availability, and RDS Read replicas’ main purpose is scalability. Moreover, Multi-Region deployments’ main purpose is disaster recovery and local performance.

Question 17:

RDS Multi-AZ deployments’ main purpose is high availability, while RDS Read replicas’ main purpose is scalability.

● **True**

○ **False**