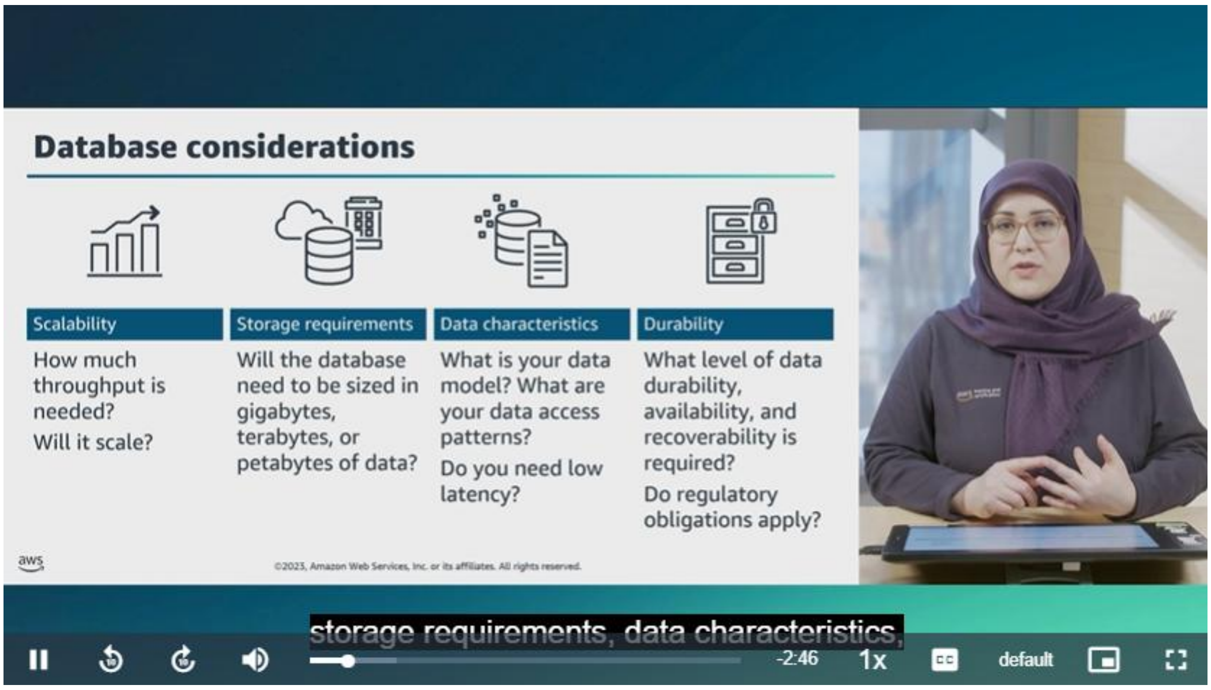
Module 6 – Adding a Database Layer

Evaluasi opsi database yang tersedia sebelum memilih data management sehingga bisa optimisasi performa.

Harus amankan infrastruktur secara efektif agar data tetap terjaga durability nya dan aman dari ancaman.

Database layer considerations



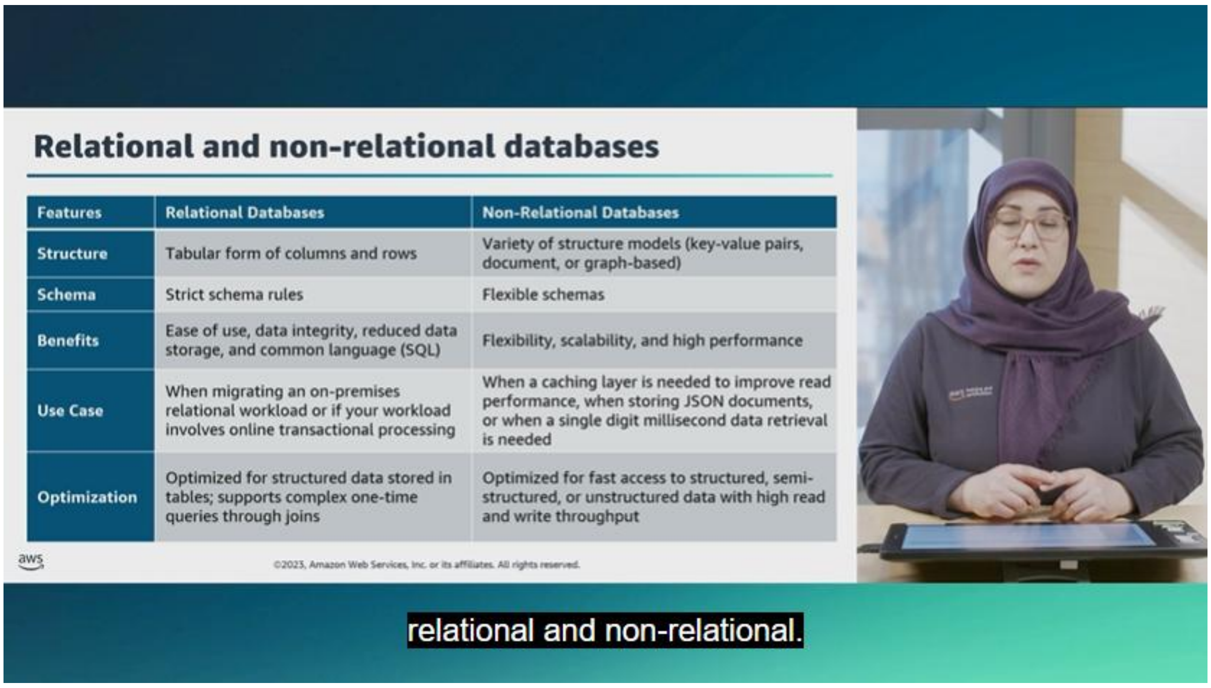
Database considerations

Scalability	Storage requirements	Data characteristics	Durability
How much throughput is needed? Will it scale?	Will the database need to be sized in gigabytes, terabytes, or petabytes of data?	What is your data model? What are your data access patterns? Do you need low latency?	What level of data durability, availability, and recoverability is required? Do regulatory obligations apply?

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

storage requirements, data characteristics,

- Scalability
- Persyaratan data
- Karakteristik data
- Durability

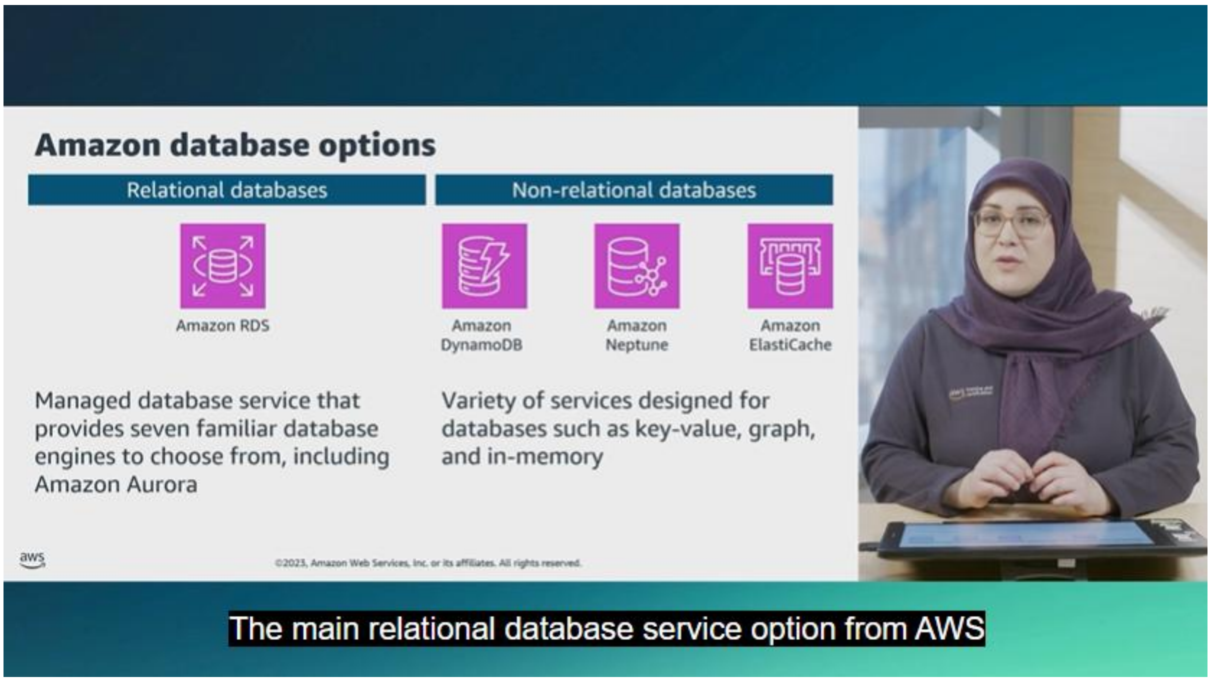


Relational and non-relational databases

Features	Relational Databases	Non-Relational Databases
Structure	Tabular form of columns and rows	Variety of structure models (key-value pairs, document, or graph-based)
Schema	Strict schema rules	Flexible schemas
Benefits	Ease of use, data integrity, reduced data storage, and common language (SQL)	Flexibility, scalability, and high performance
Use Case	When migrating an on-premises relational workload or if your workload involves online transactional processing	When a caching layer is needed to improve read performance, when storing JSON documents, or when a single digit millisecond data retrieval is needed
Optimization	Optimized for structured data stored in tables; supports complex one-time queries through joins	Optimized for fast access to structured, semi-structured, or unstructured data with high read and write throughput

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

relational and non-relational.



Amazon database options

Relational databases

Amazon RDS

Managed database service that provides seven familiar database engines to choose from, including Amazon Aurora

Non-relational databases

Amazon DynamoDB

Amazon Neptune

Amazon ElastiCache

Variety of services designed for databases such as key-value, graph, and in-memory

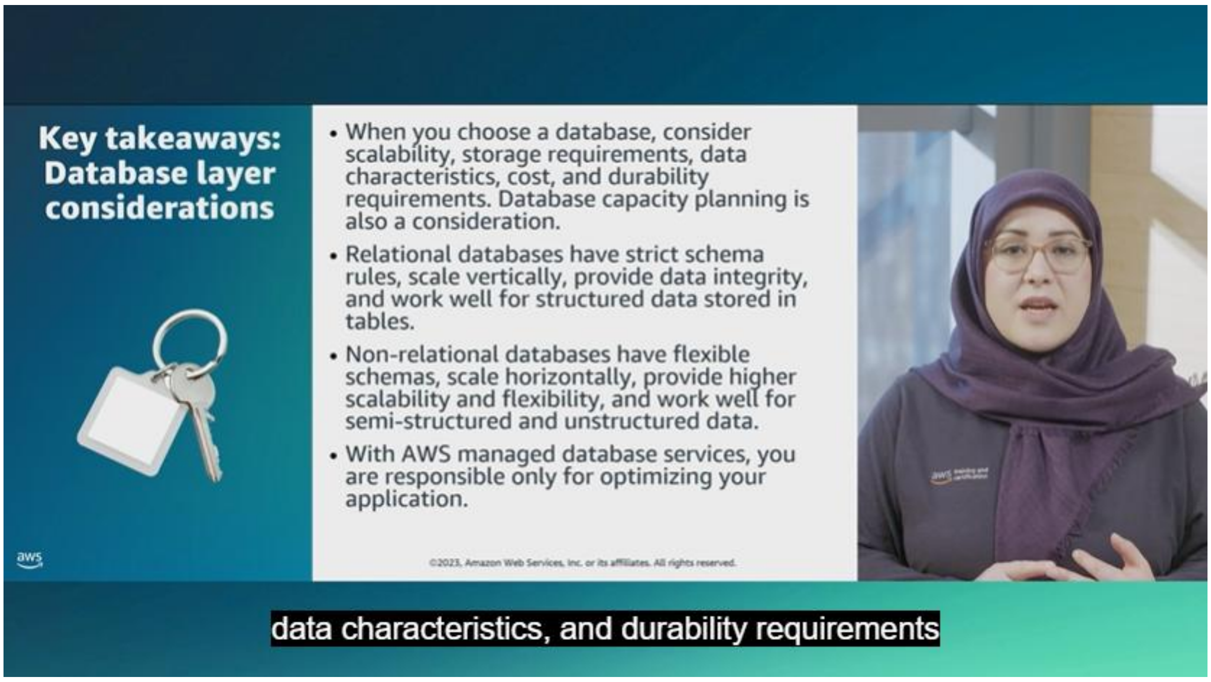
©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The main relational database service option from AWS

Amazon Relational Database Service = Amazon RDS

Berguna untuk optimisasi aplikasi

Amazon Non-relational Database = Amazon DynamoDB, Amazon Neptune, Amazon ElastiCache.

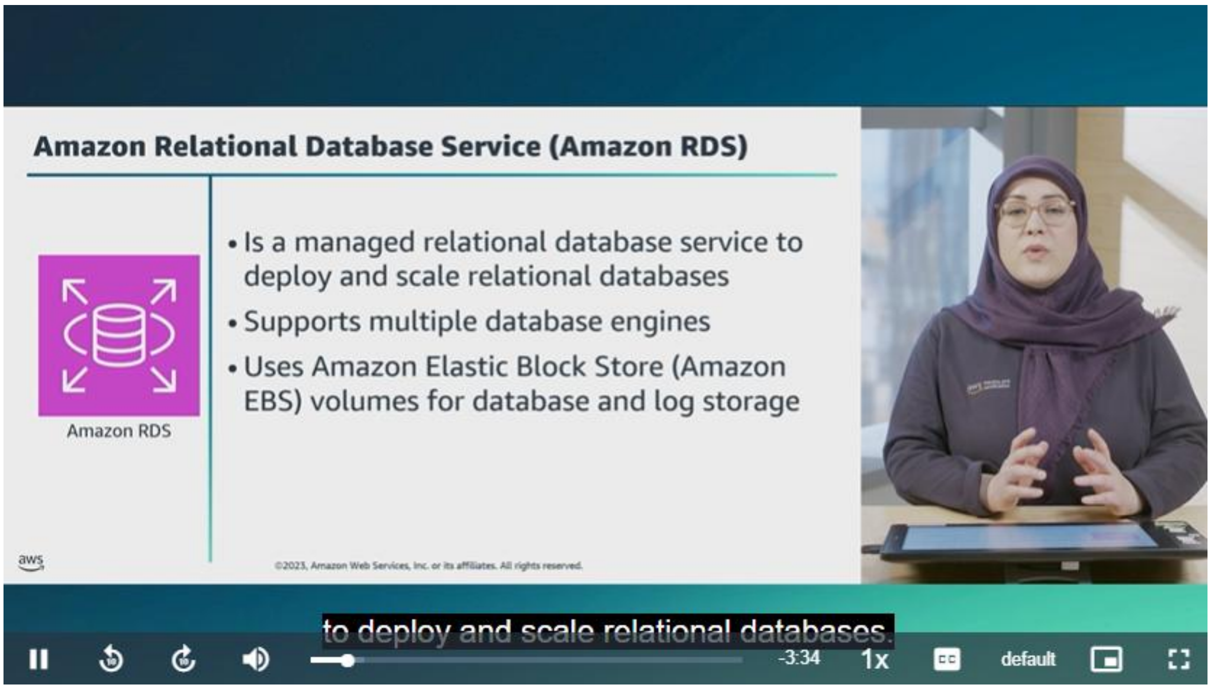

Key takeaways: Database layer considerations

- When you choose a database, consider scalability, storage requirements, data characteristics, cost, and durability requirements. Database capacity planning is also a consideration.
- Relational databases have strict schema rules, scale vertically, provide data integrity, and work well for structured data stored in tables.
- Non-relational databases have flexible schemas, scale horizontally, provide higher scalability and flexibility, and work well for semi-structured and unstructured data.
- With AWS managed database services, you are responsible only for optimizing your application.

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

data characteristics, and durability requirements

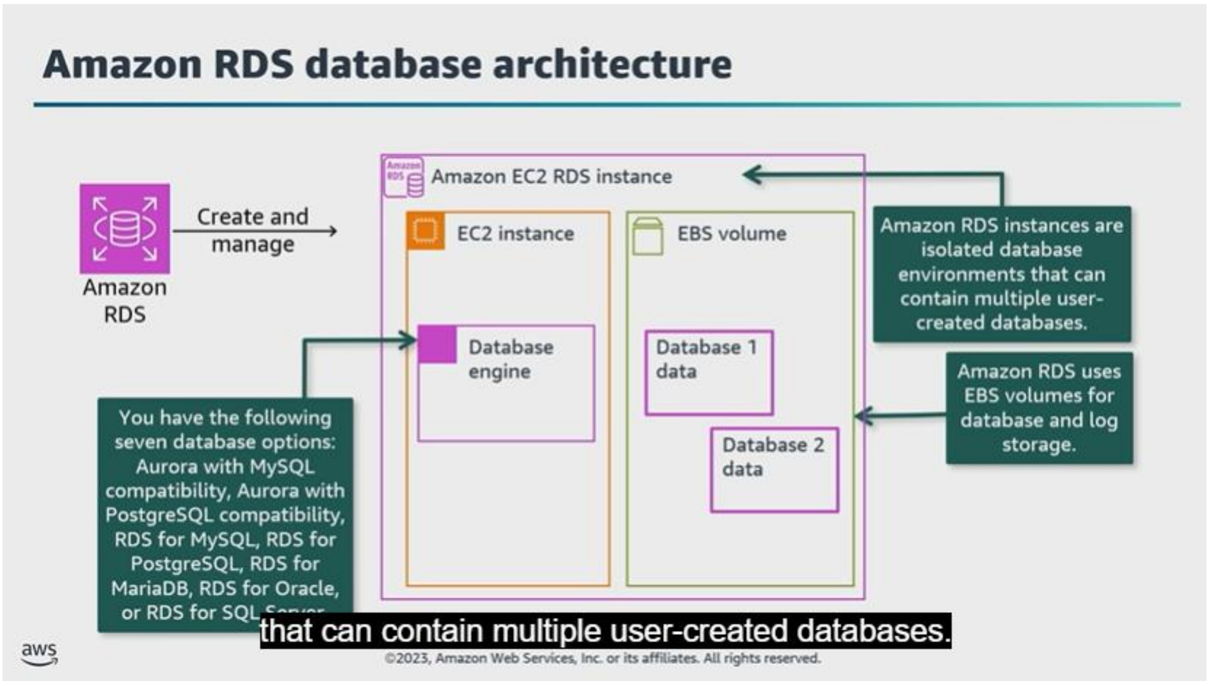
Amazon RDS = relational database untuk menyebarkan dan mengatur skala dari relational databases.


Amazon Relational Database Service (Amazon RDS)

- Is a managed relational database service to deploy and scale relational databases
- Supports multiple database engines
- Uses Amazon Elastic Block Store (Amazon EBS) volumes for database and log storage

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

to deploy and scale relational databases.


Amazon RDS database architecture

Amazon RDS instances are isolated database environments that can contain multiple user-created databases.

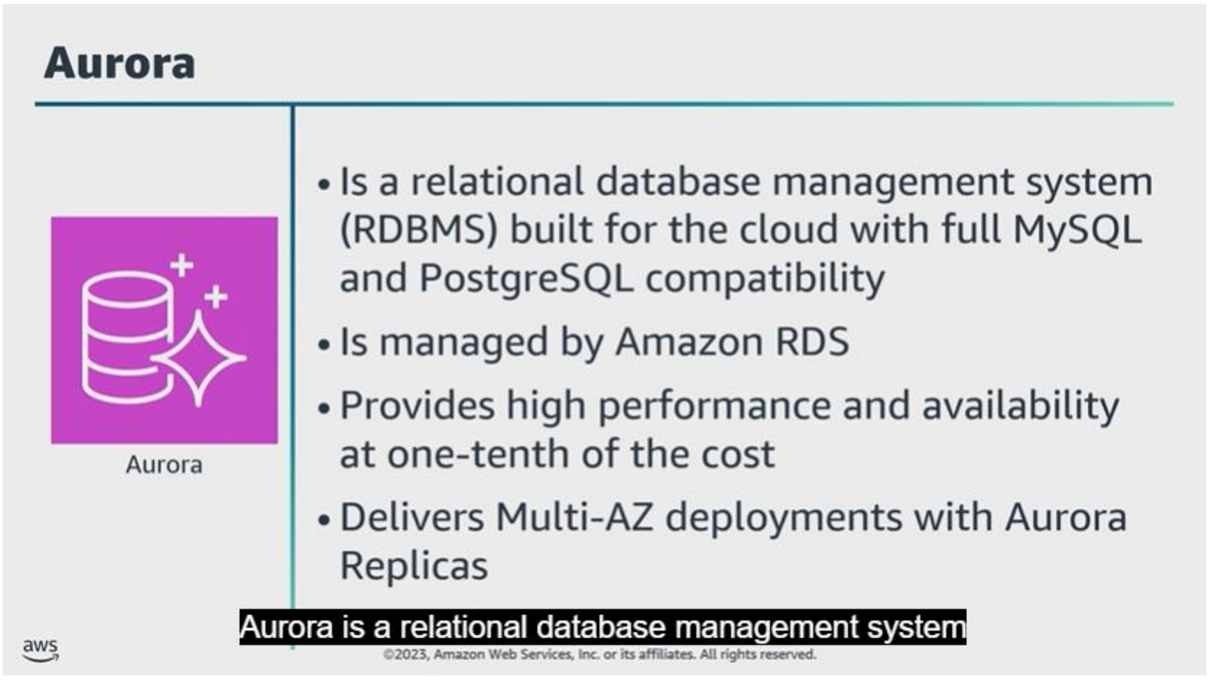
Amazon RDS uses EBS volumes for database and log storage.

You have the following seven database options: Aurora with MySQL compatibility, Aurora with PostgreSQL compatibility, RDS for MySQL, RDS for PostgreSQL, RDS for MariaDB, RDS for Oracle, or RDS for SQL Server

that can contain multiple user-created databases.

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EC2 RDS instance > EC2 instance [Database engine] & EBS Volume [database 1 data & database 2 data].

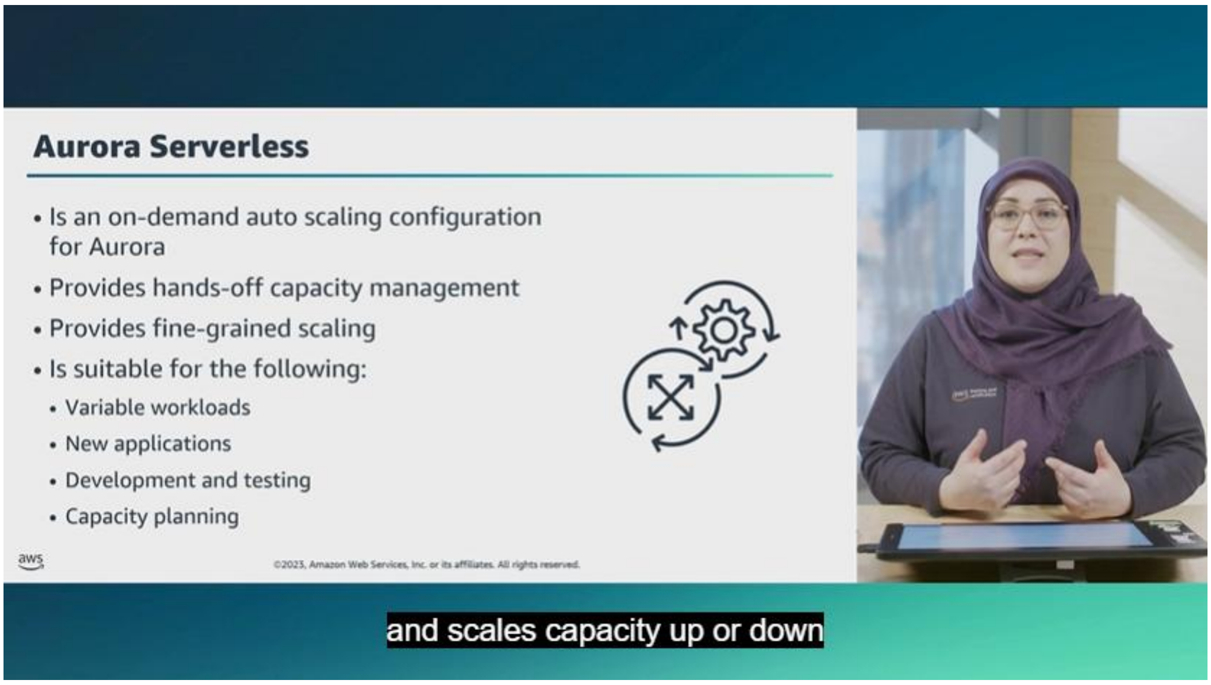

Aurora

- Is a relational database management system (RDBMS) built for the cloud with full MySQL and PostgreSQL compatibility
- Is managed by Amazon RDS
- Provides high performance and availability at one-tenth of the cost
- Delivers Multi-AZ deployments with Aurora Replicas

Aurora is a relational database management system

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

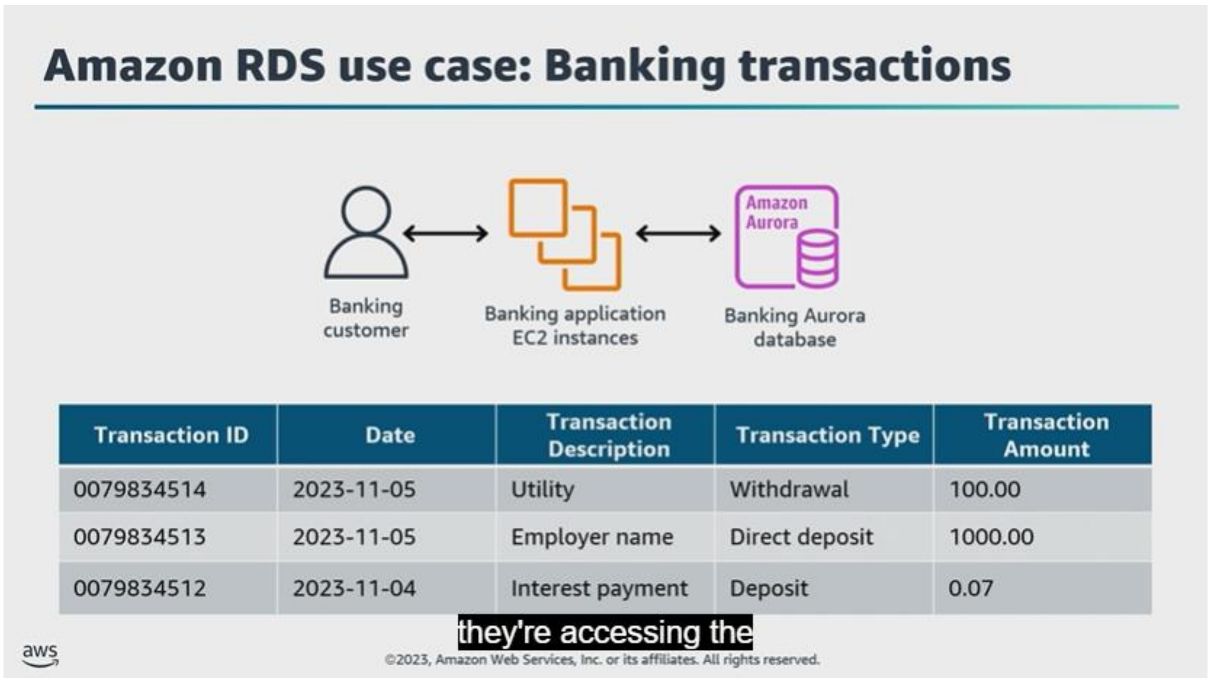
Aurora = relasional database management system yang dibuat untuk cloud dan disesuaikan dengan MySQL dan PostgreSQL.



Aurora Serverless

- Is an on-demand auto scaling configuration for Aurora
- Provides hands-off capacity management
- Provides fine-grained scaling
- Is suitable for the following:
  - Variable workloads
  - New applications
  - Development and testing
  - Capacity planning

and scales capacity up or down



Amazon RDS use case: Banking transactions

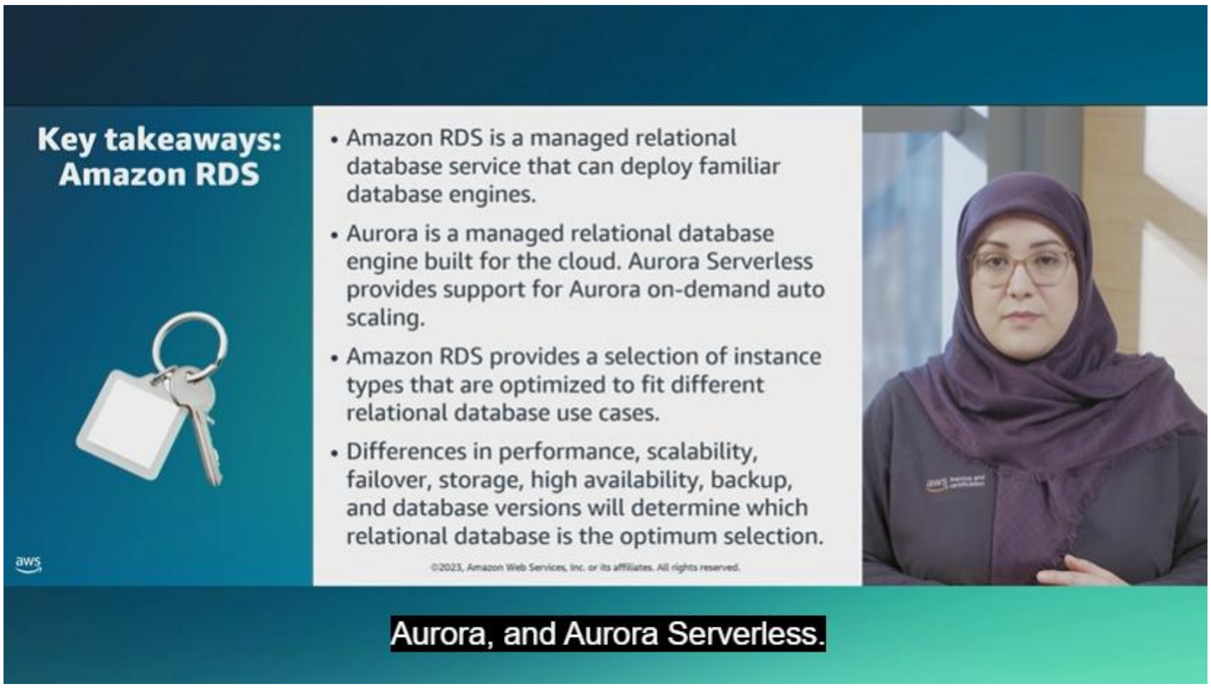
Transaction ID	Date	Transaction Description	Transaction Type	Transaction Amount
0079834514	2023-11-05	Utility	Withdrawal	100.00
0079834513	2023-11-05	Employer name	Direct deposit	1000.00
0079834512	2023-11-04	Interest payment	Deposit	0.07

they're accessing the

Contohnya = Hosted in banking application EC2 instances

Mendukung Scalability, bisa diintegrasikan dengan service yang lain

Adanya memory intensive atau compute intensive



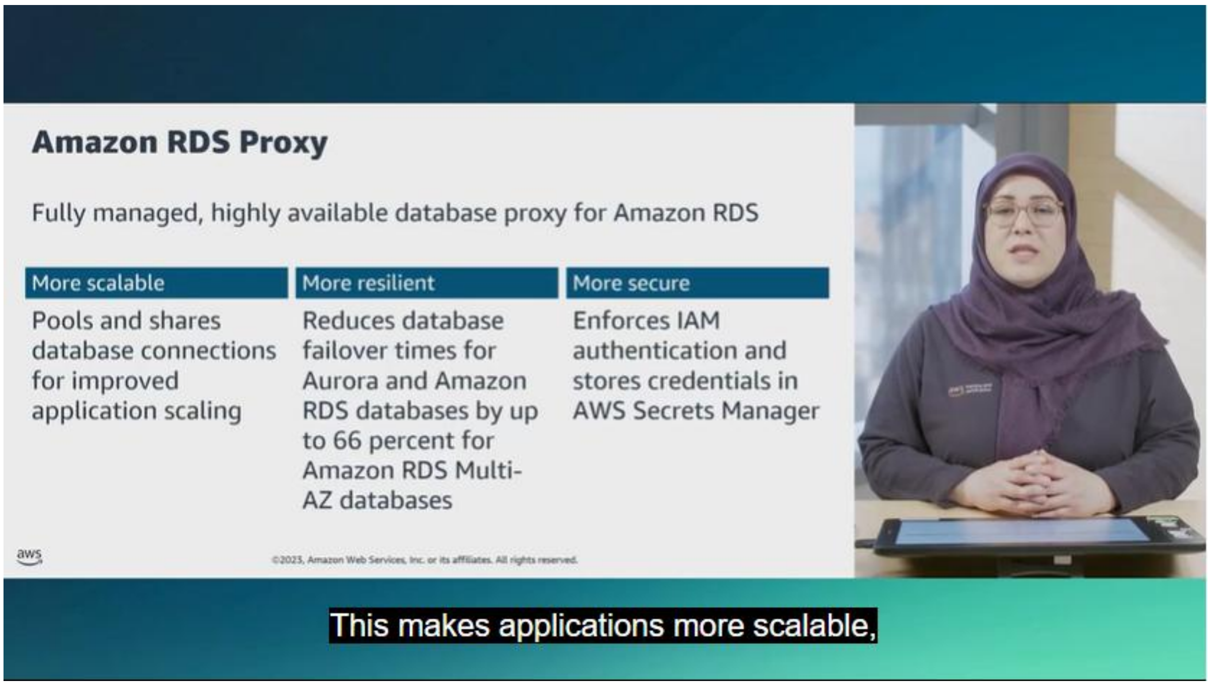
Key takeaways: Amazon RDS

- Amazon RDS is a managed relational database service that can deploy familiar database engines.
- Aurora is a managed relational database engine built for the cloud. Aurora Serverless provides support for Aurora on-demand auto scaling.
- Amazon RDS provides a selection of instance types that are optimized to fit different relational database use cases.
- Differences in performance, scalability, failover, storage, high availability, backup, and database versions will determine which relational database is the optimum selection.

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Aurora, and Aurora Serverless.

Amazon RDS proxy connection management



Amazon RDS Proxy

Fully managed, highly available database proxy for Amazon RDS

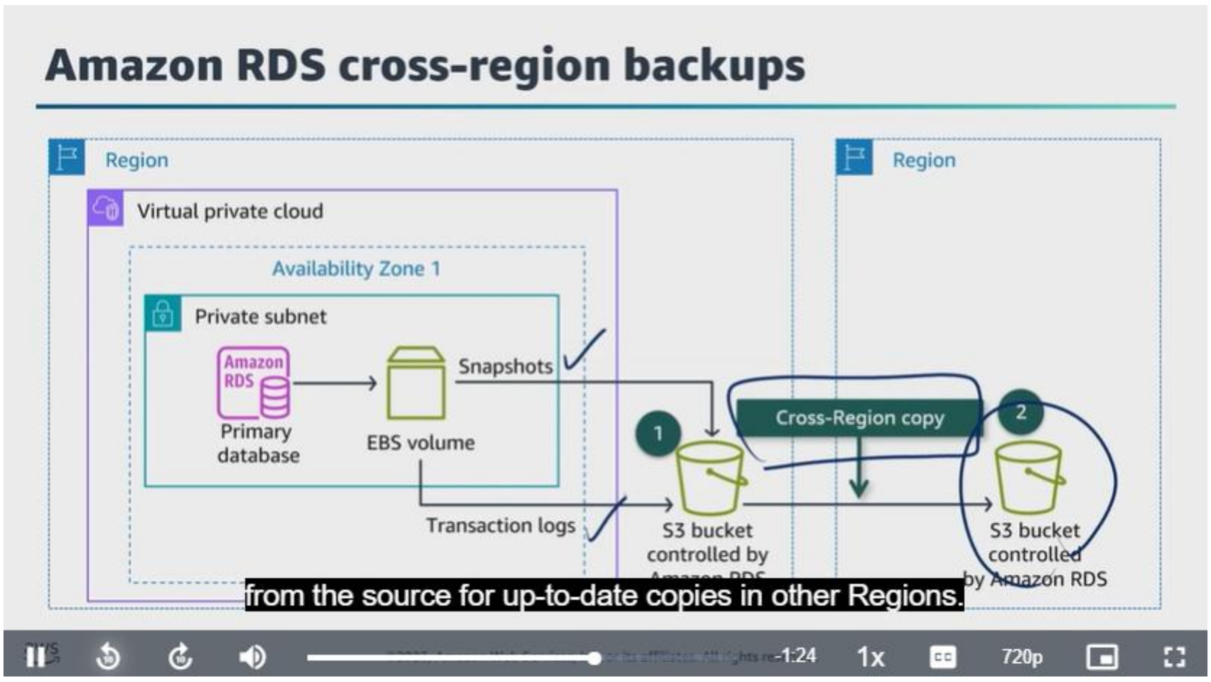
More scalable	More resilient	More secure
Pools and shares database connections for improved application scaling	Reduces database failover times for Aurora and Amazon RDS databases by up to 66 percent for Amazon RDS Multi-AZ databases	Enforces IAM authentication and stores credentials in AWS Secrets Manager

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

This makes applications more scalable,

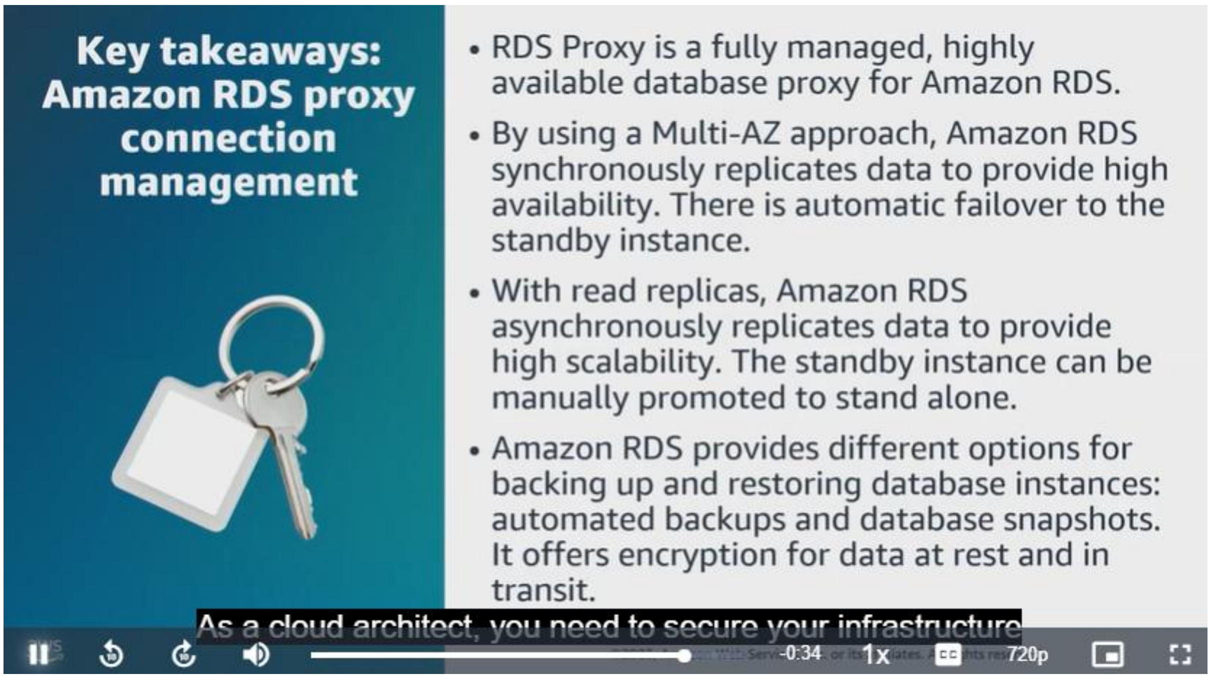
Improve scalability, availability dan bisa back up serta menyimpan database instance di known state lalu memulihkannya (restore) ke state yang spesifik.

[Can back up dan restored a database instance in a known state and then restore it to specific state]



Data yang dikirim antara aplikasi akan dienkripsi selama ditransfer

Untuk back up database yang tidak dienkripsi, harus buat snapshot, copy lalu enkrip copyannya lalu buat database yang telah dienkripsi dari snapshot tersebut.



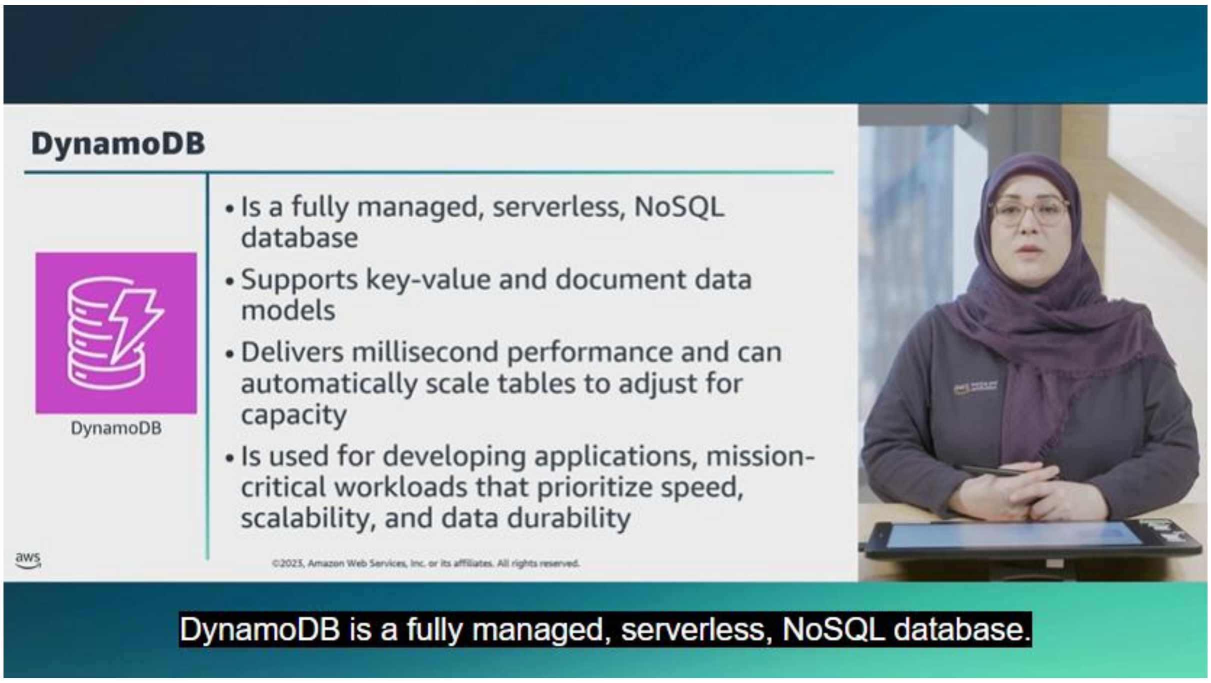
Key takeaways: Amazon RDS proxy connection management

- RDS Proxy is a fully managed, highly available database proxy for Amazon RDS.
- By using a Multi-AZ approach, Amazon RDS synchronously replicates data to provide high availability. There is automatic failover to the standby instance.
- With read replicas, Amazon RDS asynchronously replicates data to provide high scalability. The standby instance can be manually promoted to stand alone.
- Amazon RDS provides different options for backing up and restoring database instances: automated backups and database snapshots. It offers encryption for data at rest and in transit.

As a cloud architect, you need to secure your infrastructure

Demo: Amazon RDS Automated Backup and Read Replicas

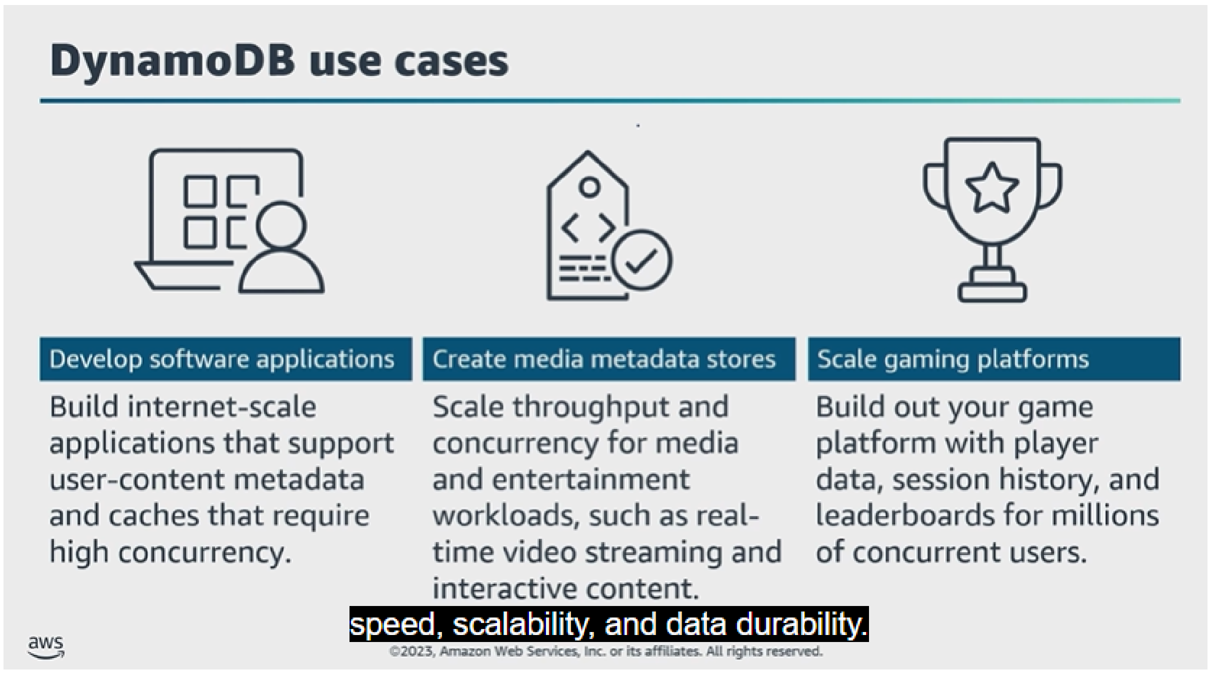
Amazon DynamoDB



DynamoDB

- Is a fully managed, serverless, NoSQL database
- Supports key-value and document data models
- Delivers millisecond performance and can automatically scale tables to adjust for capacity
- Is used for developing applications, mission-critical workloads that prioritize speed, scalability, and data durability

DynamoDB is a fully managed, serverless, NoSQL database.



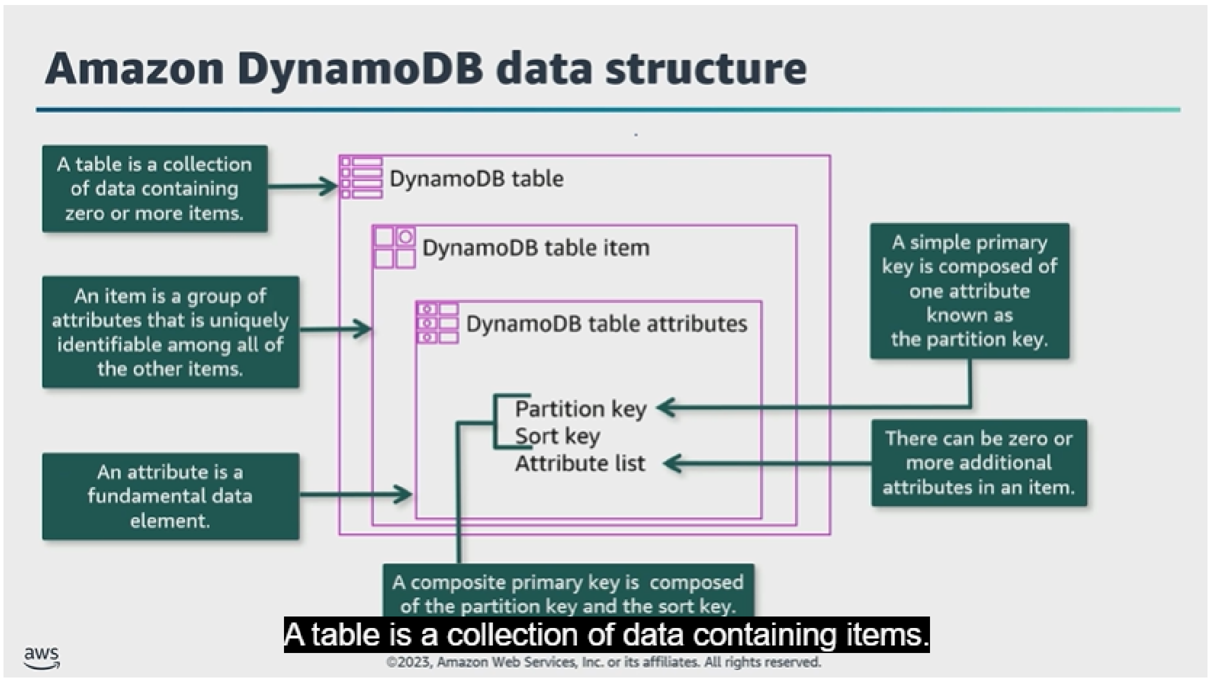
DynamoDB use cases

Develop software applications
Build internet-scale applications that support user-content metadata and caches that require high concurrency.

Create media metadata stores
Scale throughput and concurrency for media and entertainment workloads, such as real-time video streaming and interactive content.

Scale gaming platforms
Build out your game platform with player data, session history, and leaderboards for millions of concurrent users.

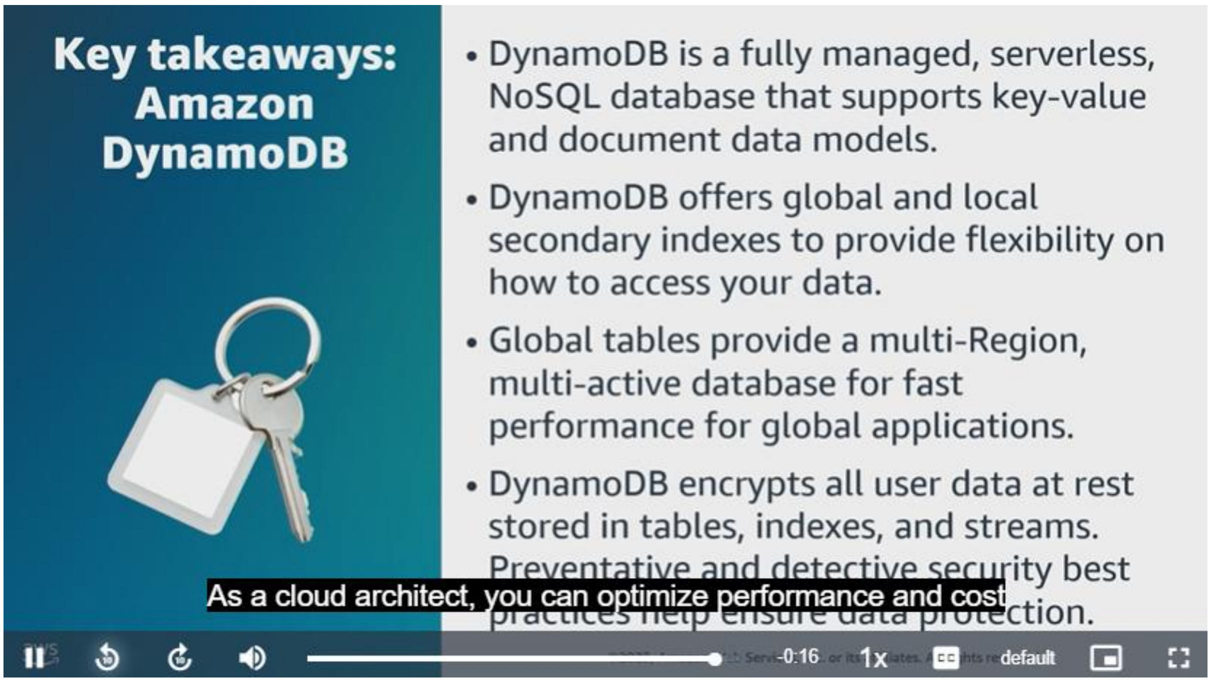
speed, scalability, and data durability.



Amazon DynamoDB data structure

A table is a collection of data containing zero or more items.
DynamoDB table
DynamoDB table item
An item is a group of attributes that is uniquely identifiable among all of the other items.
DynamoDB table attributes
A simple primary key is composed of one attribute known as the partition key.
Partition key
Sort key
Attribute list
There can be zero or more additional attributes in an item.
An attribute is a fundamental data element.
A composite primary key is composed of the partition key and the sort key.
A table is a collection of data containing items.
aws

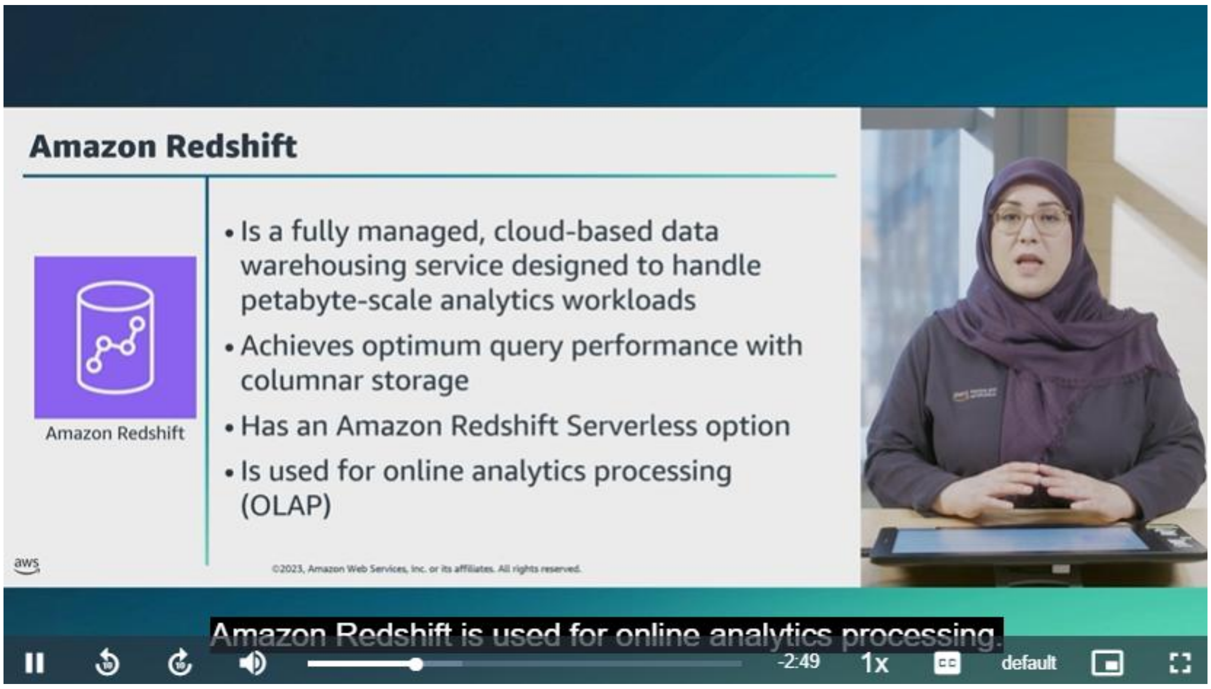
Table > item > attributes (partition key, sort key, attribute key)



Key takeaways: Amazon DynamoDB

- DynamoDB is a fully managed, serverless, NoSQL database that supports key-value and document data models.
- DynamoDB offers global and local secondary indexes to provide flexibility on how to access your data.
- Global tables provide a multi-Region, multi-active database for fast performance for global applications.
- DynamoDB encrypts all user data at rest stored in tables, indexes, and streams. Preventative and detective security best practices help ensure data protection.

Purpose built-database

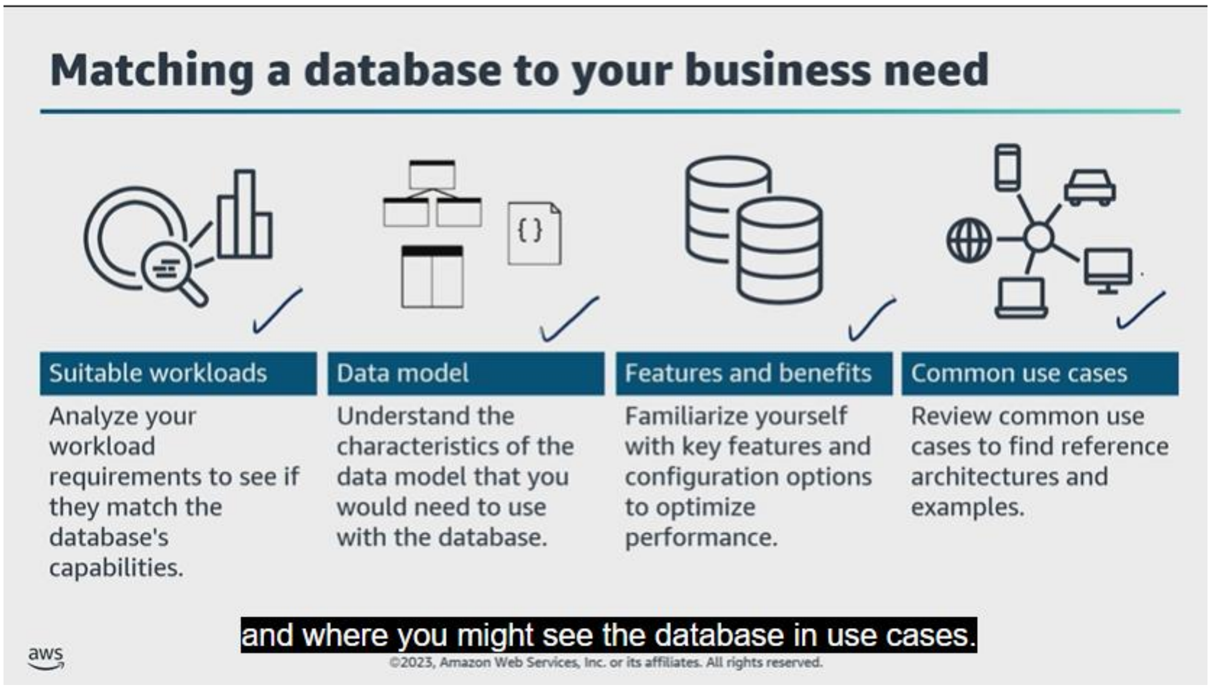


Amazon Redshift

Amazon Redshift

- Is a fully managed, cloud-based data warehousing service designed to handle petabyte-scale analytics workloads
- Achieves optimum query performance with columnar storage
- Has an Amazon Redshift Serverless option
- Is used for online analytics processing (OLAP)

Amazon Redshift is used for online analytics processing.



Matching a database to your business need

Suitable workloads	Data model	Features and benefits	Common use cases
Analyze your workload requirements to see if they match the database's capabilities.	Understand the characteristics of the data model that you would need to use with the database.	Familiarize yourself with key features and configuration options to optimize performance.	Review common use cases to find reference architectures and examples.

and where you might see the database in use cases.

Amazon DB = document database, didesain untuk menyimpan dan query data dalam format JSON documents, contohnya menggunakan MongoDB

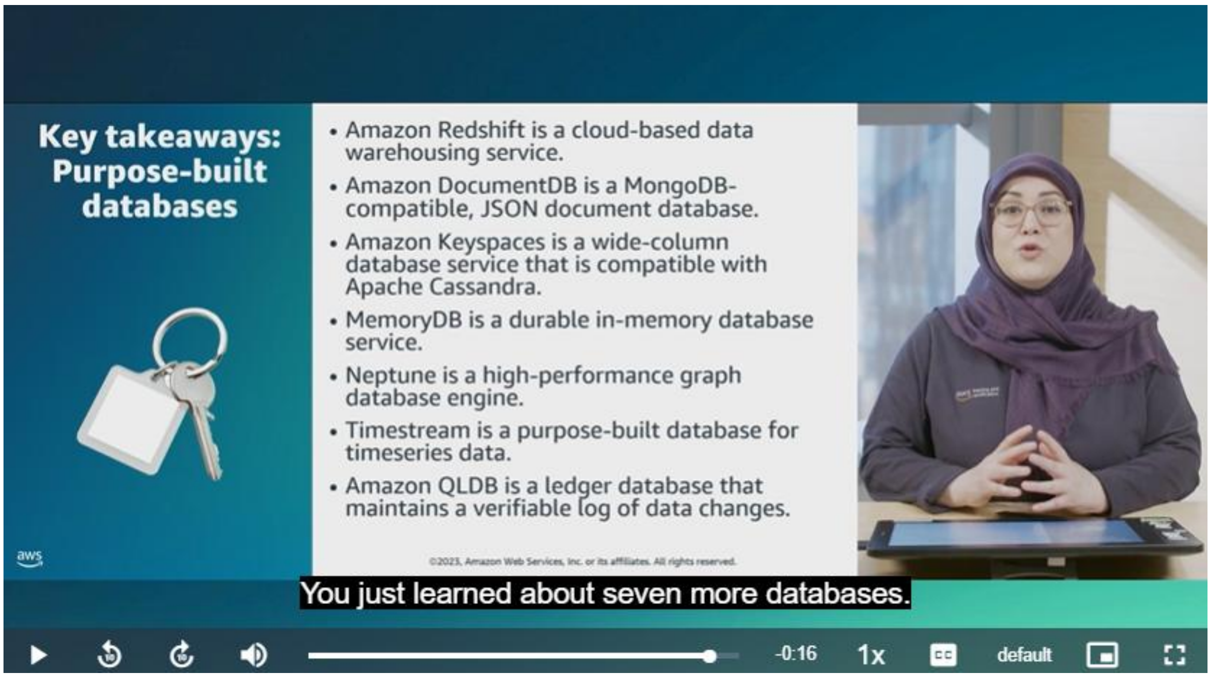
MemoryDB = In memory database service, meminimalisasi response time dengan mengeliminasi keperluan untuk mengakses disks. Menyimpan keseluruhan dataset di dalam memori. Hal ini bisa digunakan untuk caching dan game leaderboards.

Amazon Keyspaces = managed Apache Cassandra-compatible database service yang bisa memproses data dengan kecepatan tinggi untuk aplikasi yang membutuhkan delay/latency yang minim, seperti trade monitoring.

Neptune = graph database, menyimpan data yang memiliki relasi dengan data yang lain. Dengan cepat membuat dan menavigasi relasi-relasi data. Digunakan dalam recommendation engines, fraud detection, drug discovery dan social networking.

Timestream = timeseries database, didesain berdasarkan data yang dibatasi waktu. Punya built-in functions untuk analisis yang cepat. Penggunaannya dalam analyzing timeseries data generated by IoT applications.

Quantum Ledger Database (QLDB) = ledger database, provides transparent, immutable, cryptographically verifiable transaction log. Digunakan dalam maintain claim history dan menyimpan transaksi finansial.



Key takeaways: Purpose-built databases

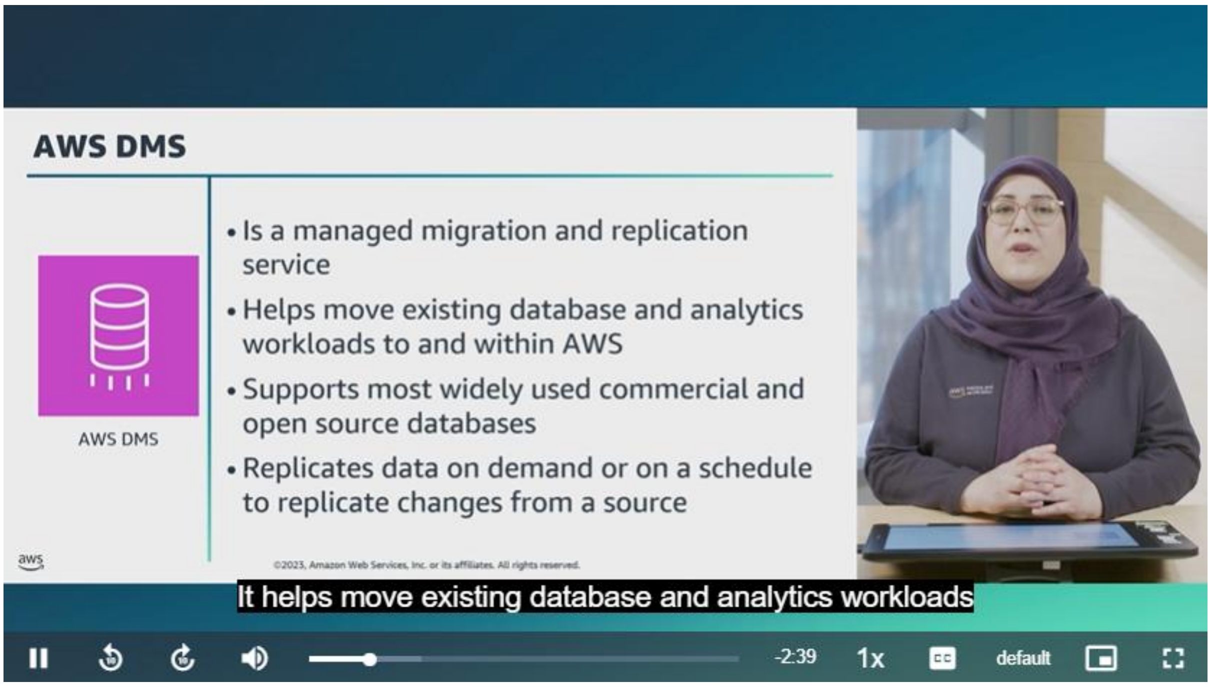
- Amazon Redshift is a cloud-based data warehousing service.
- Amazon DocumentDB is a MongoDB-compatible, JSON document database.
- Amazon Keyspaces is a wide-column database service that is compatible with Apache Cassandra.
- MemoryDB is a durable in-memory database service.
- Neptune is a high-performance graph database engine.
- Timestream is a purpose-built database for timeseries data.
- Amazon QLDB is a ledger database that maintains a verifiable log of data changes.

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

You just learned about seven more databases.

Migrating data into AWS Database

AWS Database berguna untuk migration and replication data.

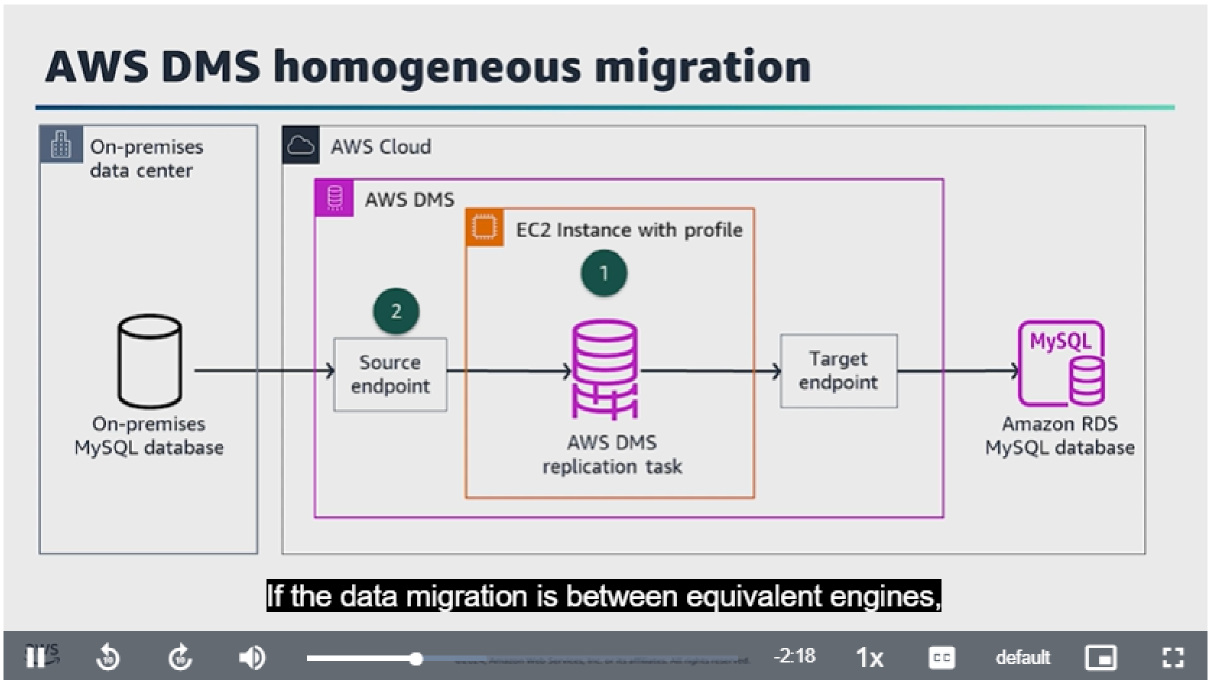


AWS DMS

- Is a managed migration and replication service
- Helps move existing database and analytics workloads to and within AWS
- Supports most widely used commercial and open source databases
- Replicates data on demand or on a schedule to replicate changes from a source

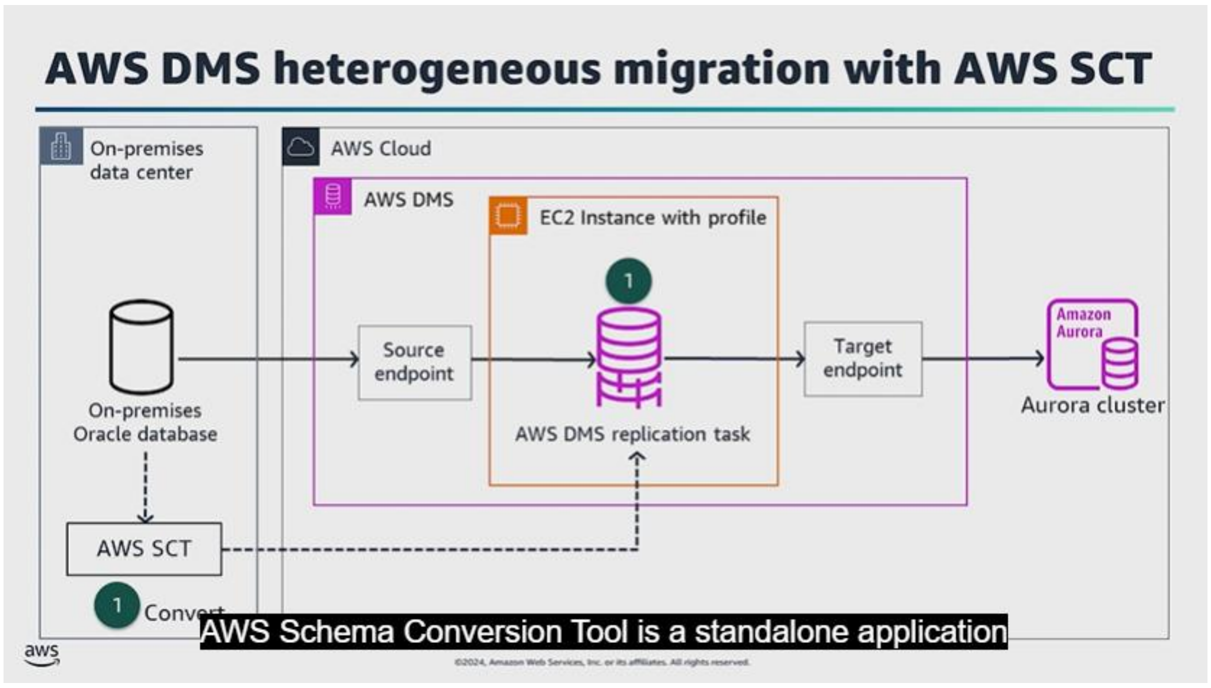
It helps move existing database and analytics workloads

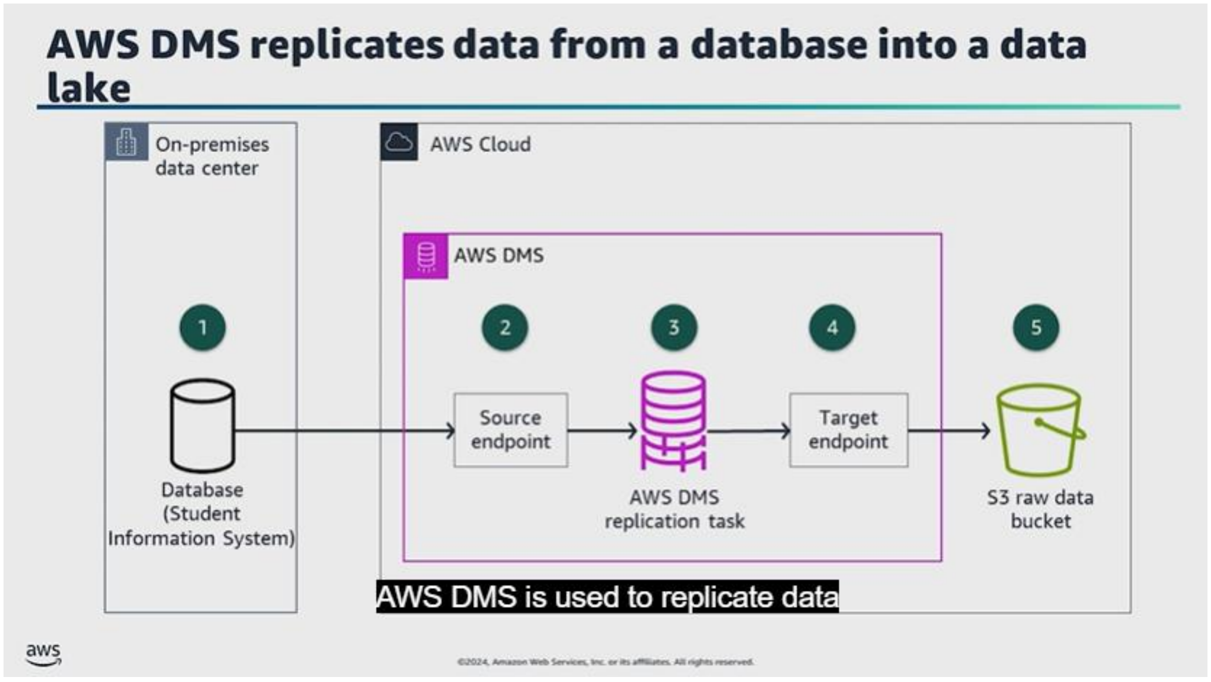
AWS DMS = Web service yang bisa digunakan untuk migrasi data dari source data store ke a target data store.



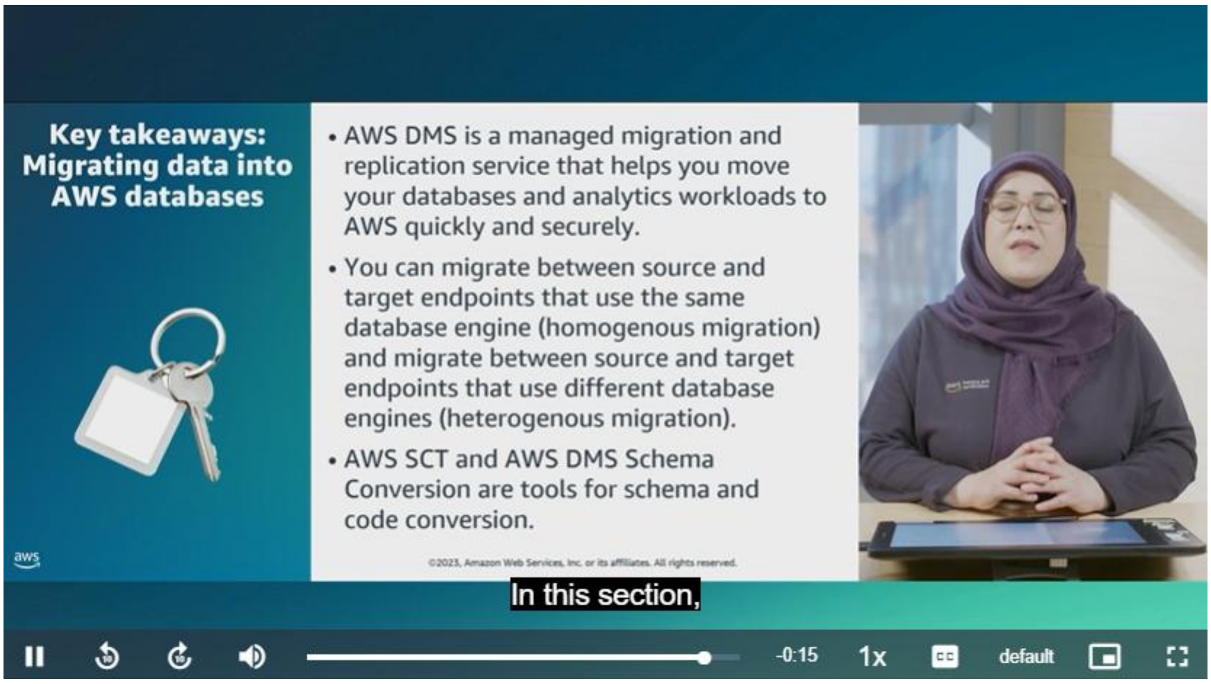
AWS DMS homogeneous migration

If the data migration is between equivalent engines,


AWS DMS heterogeneous migration with AWS SCT
On-premises data center
AWS Cloud
AWS DMS
EC2 Instance with profile
1
Source endpoint
AWS DMS replication task
Target endpoint
Amazon Aurora
Aurora cluster
On-premises Oracle database
AWS SCT
1 Convert
AWS Schema Conversion Tool is a standalone application
©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.


AWS DMS replicates data from a database into a data lake
On-premises data center
AWS Cloud
AWS DMS
1
2
3
4
5
Database (Student Information System)
Source endpoint
AWS DMS replication task
Target endpoint
S3 raw data bucket
AWS DMS is used to replicate data
©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Dalam tingkat higher education = AWS DMS digunakan untuk replikasi data dari database ke data link



Key takeaways: Migrating data into AWS databases

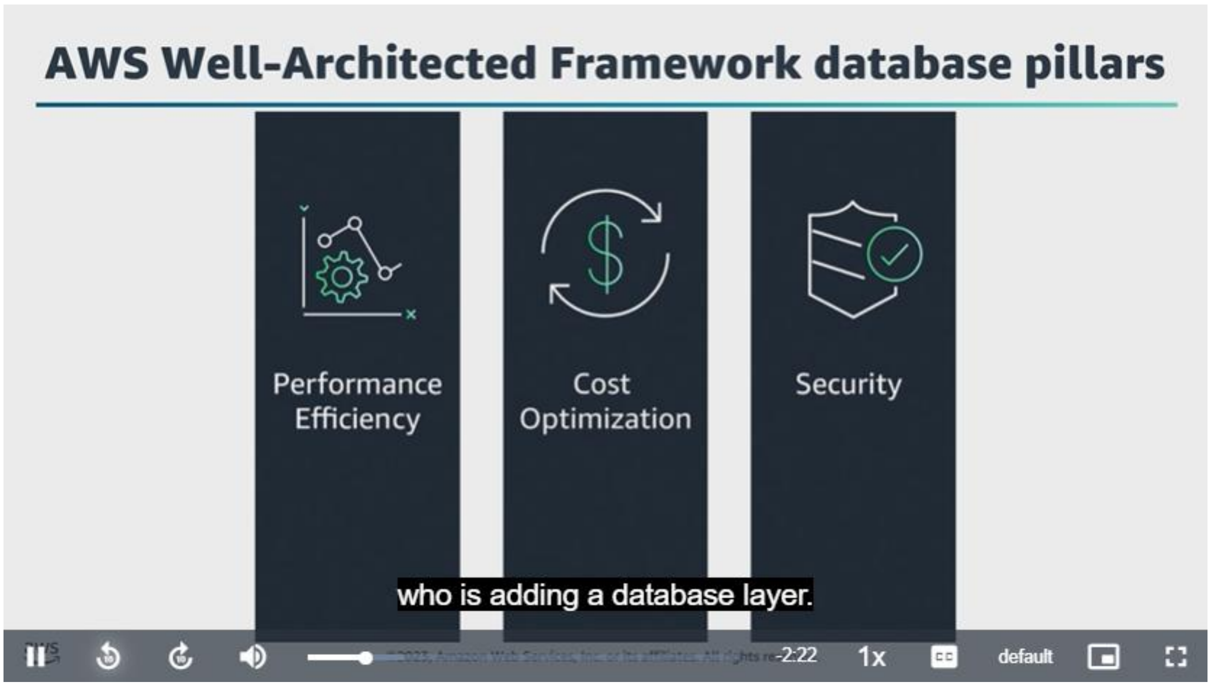
- AWS DMS is a managed migration and replication service that helps you move your databases and analytics workloads to AWS quickly and securely.
- You can migrate between source and target endpoints that use the same database engine (homogenous migration) and migrate between source and target endpoints that use different database engines (heterogenous migration).
- AWS SCT and AWS DMS Schema Conversion are tools for schema and code conversion.

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In this section,

-0:15 1x default

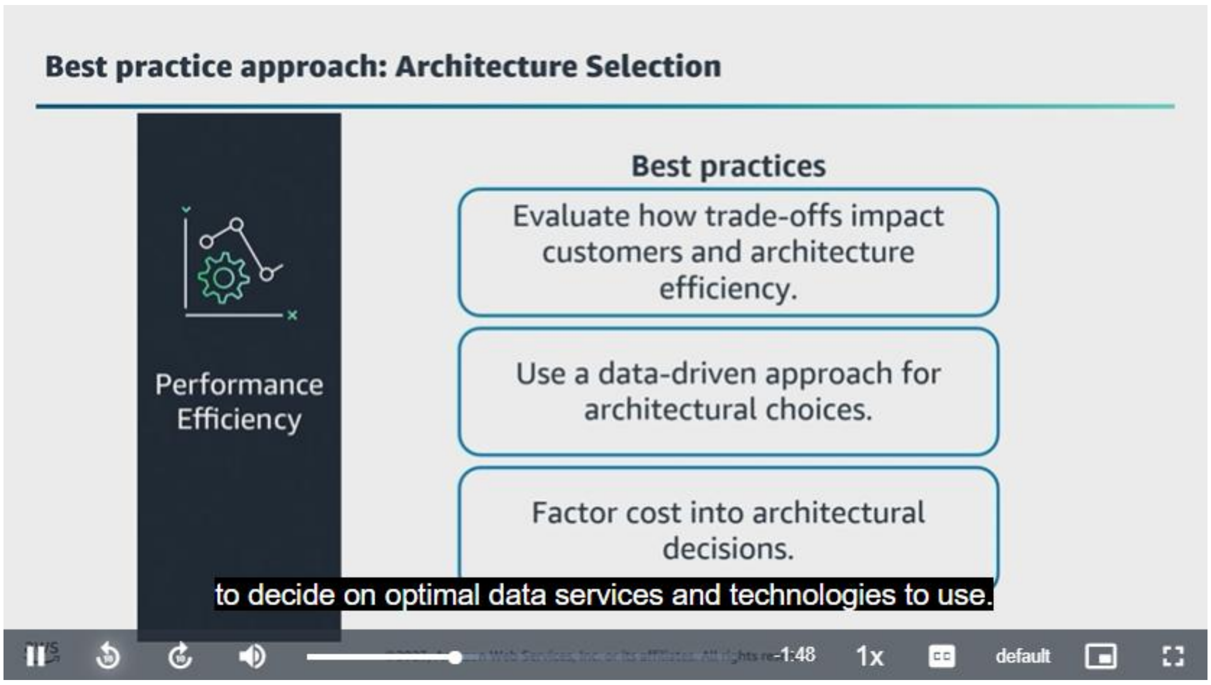
Applying Well-Architected Framework Principles to the Database Layer



AWS Well-Architected Framework database pillars

who is adding a database layer.

-2:22 1x default



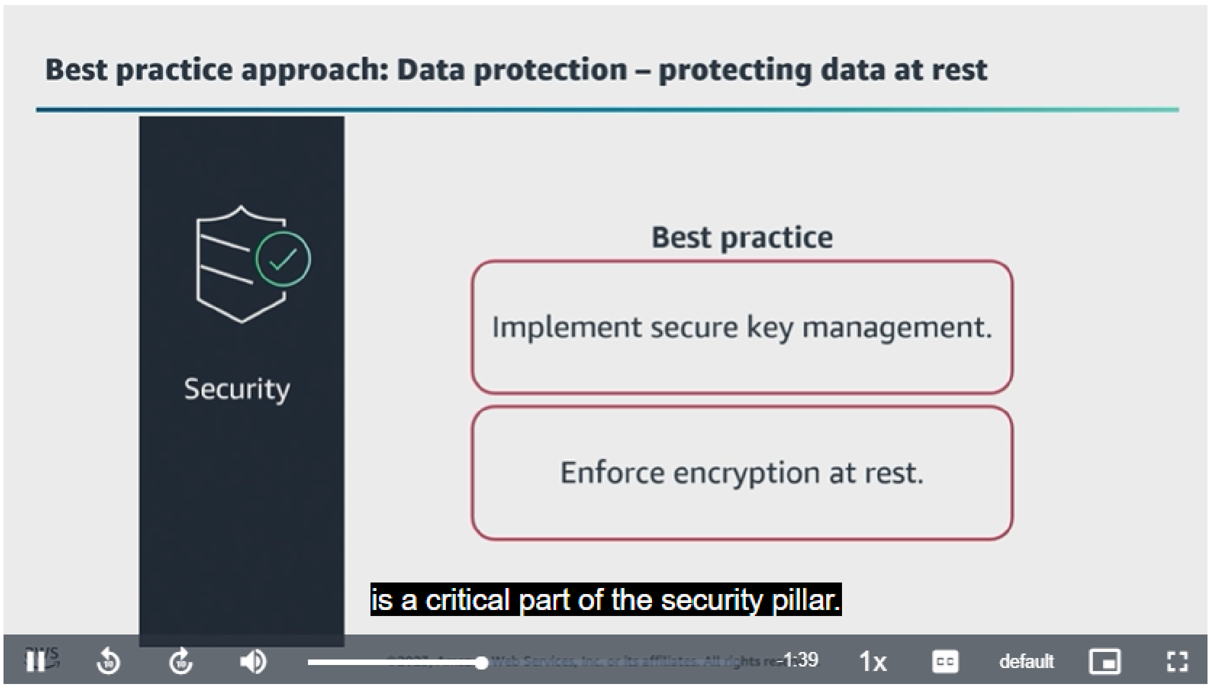
Best practice approach: Architecture Selection

Performance Efficiency

Best practices

- Evaluate how trade-offs impact customers and architecture efficiency.
- Use a data-driven approach for architectural choices.
- Factor cost into architectural decisions.

to decide on optimal data services and technologies to use.



Best practice approach: Data protection – protecting data at rest

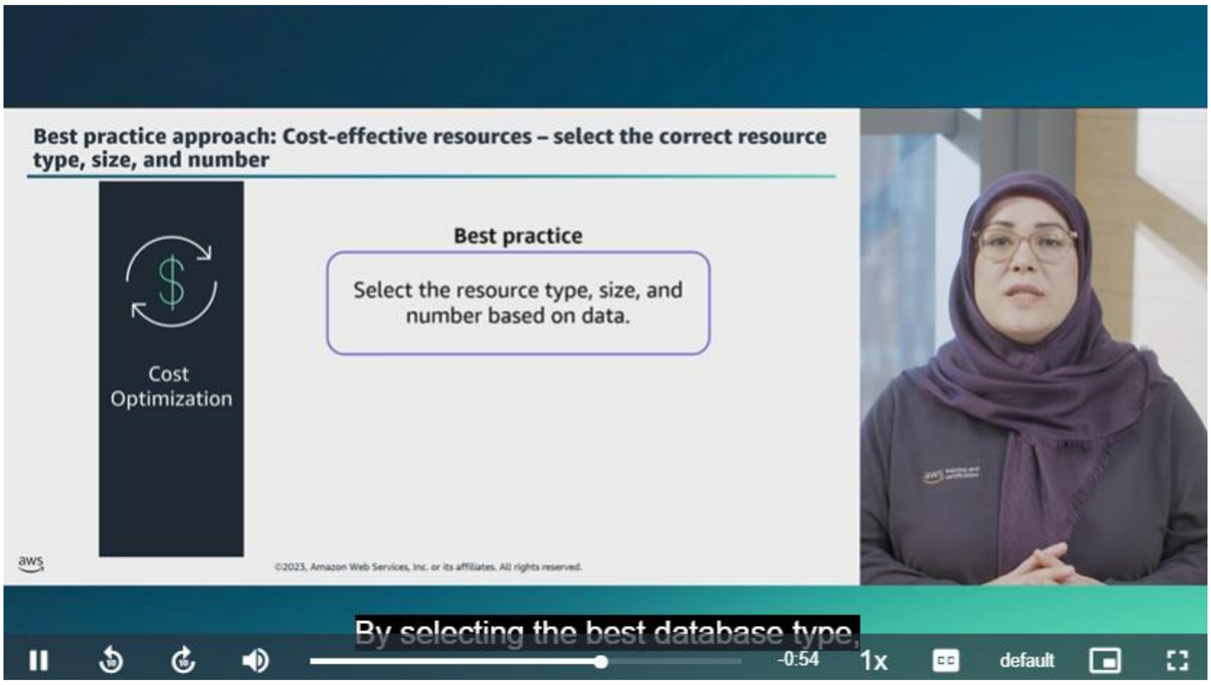
Security

Best practice

- Implement secure key management.
- Enforce encryption at rest.

is a critical part of the security pillar.

Amazon RDBS – RDS, DynamoDB menggunakan AWS key management service (KMS). Dynamo DB encrypts semua user data at rest stored in tables, indexes, streams, backups dengan menggunakan ecryption keys yang disimpan di AWS KMS.



Best practice approach: Cost-effective resources – select the correct resource type, size, and number

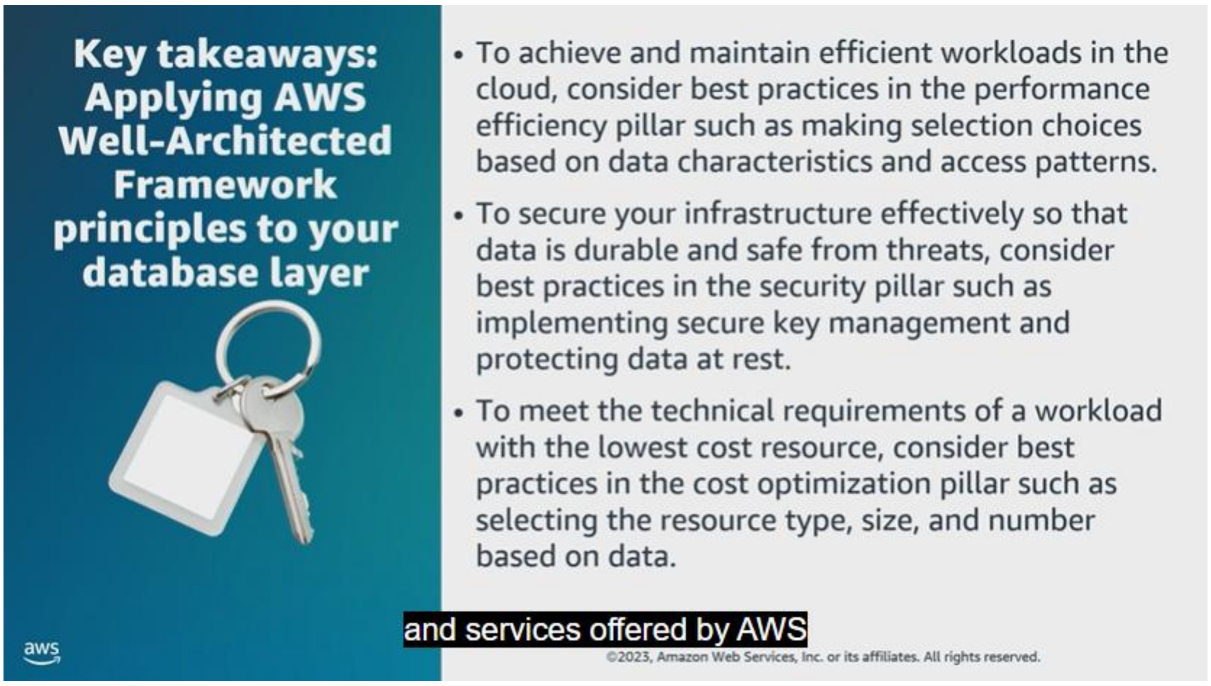
Cost Optimization

Best practice

Select the resource type, size, and number based on data.

By selecting the best database type,

how different database resources can scale



Key takeaways: Applying AWS Well-Architected Framework principles to your database layer

- To achieve and maintain efficient workloads in the cloud, consider best practices in the performance efficiency pillar such as making selection choices based on data characteristics and access patterns.
- To secure your infrastructure effectively so that data is durable and safe from threats, consider best practices in the security pillar such as implementing secure key management and protecting data at rest.
- To meet the technical requirements of a workload with the lowest cost resource, consider best practices in the cost optimization pillar such as selecting the resource type, size, and number based on data.

and services offered by AWS