

Database on *AWS*

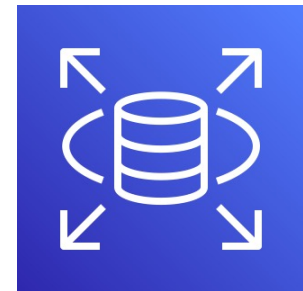
Nội dung

- RDS Introduction
- DB Multi-AZ and Replicas
- Aurora
- DynamoDB
- ElasticCache
- Caching Strategy on AWS
- Redshift
- Right DB solution
- DMS

RDS Introduction

RDS Introduction

- RDS viết tắt của Relational Database Service.
- RDS là dịch vụ DB service do AWS quản lý (AWS Managed Service)
- Hỗ trợ các DB Engine
 - MySQL
 - MariaDB
 - PostgreSQL
 - Oracle
 - Microsoft SQL Server
 - Aurora



Amazon Relational Database
Service (Amazon RDS)

RDS và Self-Managed DB

- Tự động cập nhật các bản vá OS (hệ điều hành)
- Thiết lập khung thời gian cho các hoạt động bảo trì (Window Maintenance)
- Tự động backup và restore dữ liệu tới một thời điểm cụ thể (Point in Time Restore)
- Tính có sẵn cao (HA) và hỗ trợ Multi-AZ deployment
- Thêm Read Replica cho phép tăng hiệu năng DB (Read Performance)
- Cho phép theo dõi chi tiết hiệu năng của DB (Insight Monitoring)

Backups và Snapshot

- **Automated Backups**

- Thực hiện tạo Snapshot của DB Instances trong khoảng thời gian được chỉ định (Window Time)
- Lưu lại Transaction Logs and Backup 5 phút một lần
- Khôi phục lại DB tại bất cứ thời điểm nào dựa vào Daily Snapshot + Transaction Logs
- Dữ liệu Backup lưu mặc định được trong vòng 7 ngày (Retention Period)

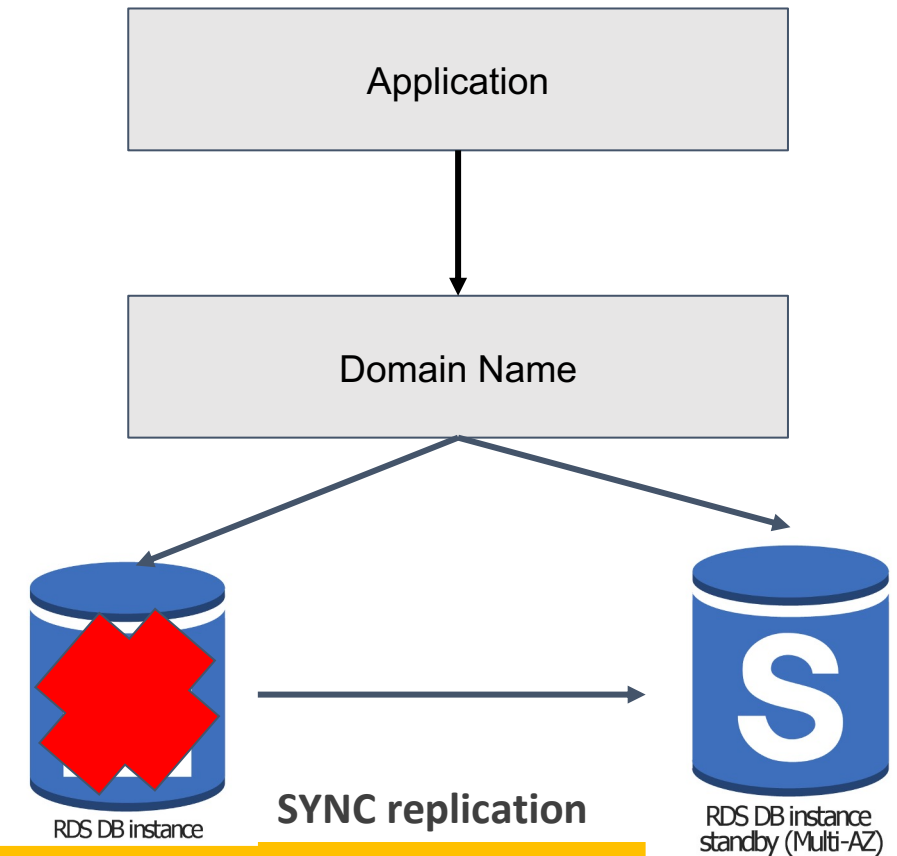
- **DB snapshot**

- Được tạo ra khi người dùng chủ động tạo Snapshot (Manual)
- Retention Period có thể tùy chọn bởi người dùng

Multi-AZ and Read Replicas

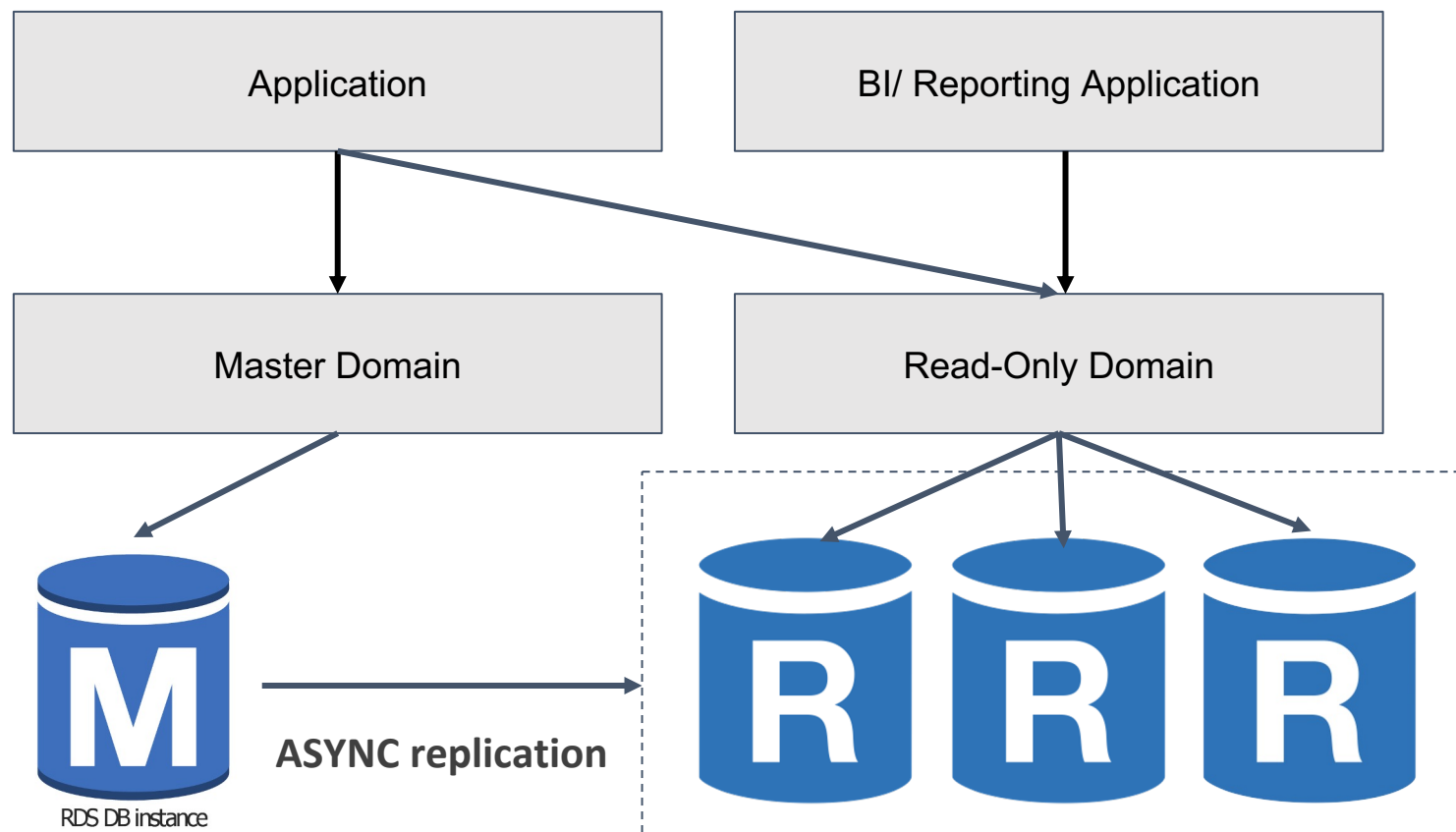
Multi-AZ

- Multi-AZ làm tăng tính có sẵn cho RDS Instances (High Availability)
- Sử dụng một DNS name duy nhất cho Active và Standby RDS
- Dữ liệu đồng bộ sử dụng SYNC replication



Read Replicas

- Read Replica giúp tăng hiệu năng và mở rộng (Scalability) của DB Instances
- Sử dụng **ASYNC** để đồng bộ dữ liệu
- Mỗi Read Replica sẽ có một DNS Endpoint riêng biệt



Read Replicas (cont.)

- Dữ liệu đọc từ Read Replica có thể bị cũ (Read Eventually Consistent)
- Có thể tạo Replica của RDS Instance chạy chế độ Multi-AZ Deployment
- Replicas có thể được tạo từ một Replica Instance khác. Tuy nhiên cân nhắc độ trễ đồng bộ dữ liệu (Replica Lags)

Aurora

Aurora

- Aurora là giải pháp Relational DB Engine được phát triển bởi AWS
- Aurora tương thích với MySQL và PostgreSQL Engine
- Hiệu năng x5 lần so với MySQL và 3x lần so với PostgreSQL
- Dữ liệu được clone thành nhiều bản, phân tán trên 3 AZs

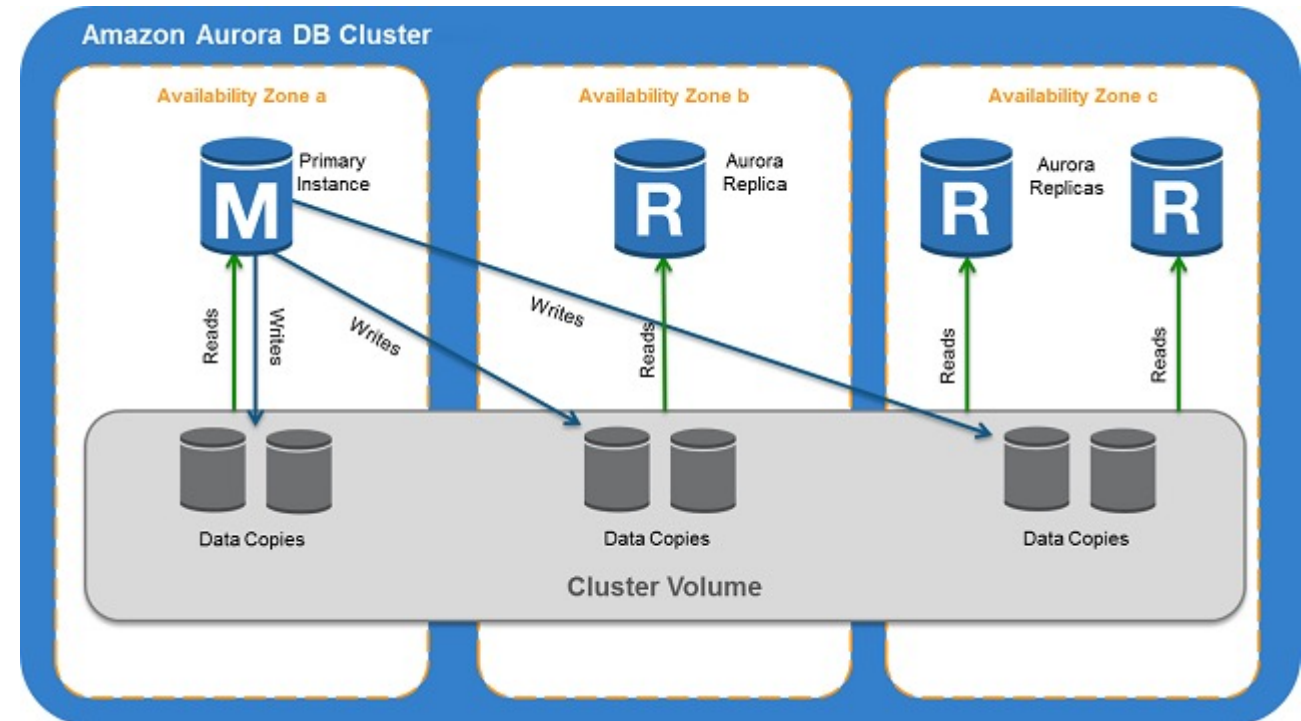


Aurora (cont.)

- Multiz AZ, Auto Scaling Read Replicas, Auto Scaling Storage (up to 128 TiB)
- Tự động xử lý sự cố (Auto failover).
- Các đặc điểm khác tương tự với RDS (Security, Monitoring, Maintenance)
- **Use case:**
 - Yêu cầu cho dịch vụ cần hiệu năng cao hơn so với MySQL, Postgres (Chi phí sẽ đắt hơn)

Aurora Cluster

- Aurora cluster bao gồm Primary và Replica Instances
- Cluster tự động khôi phục (Auto-Failover) trong trường hợp Primary Instance bị Downtime



Exam Tips

- Dữ liệu DB được clone thành nhiều bản, mỗi AZ sẽ có 2 bản copy và phân tán trên 3 AZs
- Quá trình tạo Snapshot không ảnh hưởng tới Performance của DB Primary
- Tự động khôi phục (Auto-Failover), tự động scale kích thước Storage và số lượng Read Replicas Instances
- 3 types of Aurora replica (currently)
 - Aurora replica (up to 15 replicas), Aurora MySQL replica (up to 5 replicas), Aurora PostgreSQL replica (up to 1)
- Dữ liệu DB mặc định được tự động sao lưu (Automated Backup)
- Sử dụng Aurora Serverless cho các Workload có tính chất không đoán định trước được (unpredictable)

DynamoDB

DynamoDB?

- Hệ cơ sở dữ liệu NoSQL do AWS quản lý (AWS managed NoSQL database)
- Độ trễ thấp (1-digit millisecond latency)
- Hỗ trợ kiểu dữ liệu Key-Value và Document
- Tính có sẵn cao (HA). Dữ liệu trong DynamoDB được copy sang 3 AZs
- Eventually Consistency Read (Default)
- Có thể thiết lập Strong Consistency (Cân nhắc về độ trễ latency)



Amazon DynamoDB

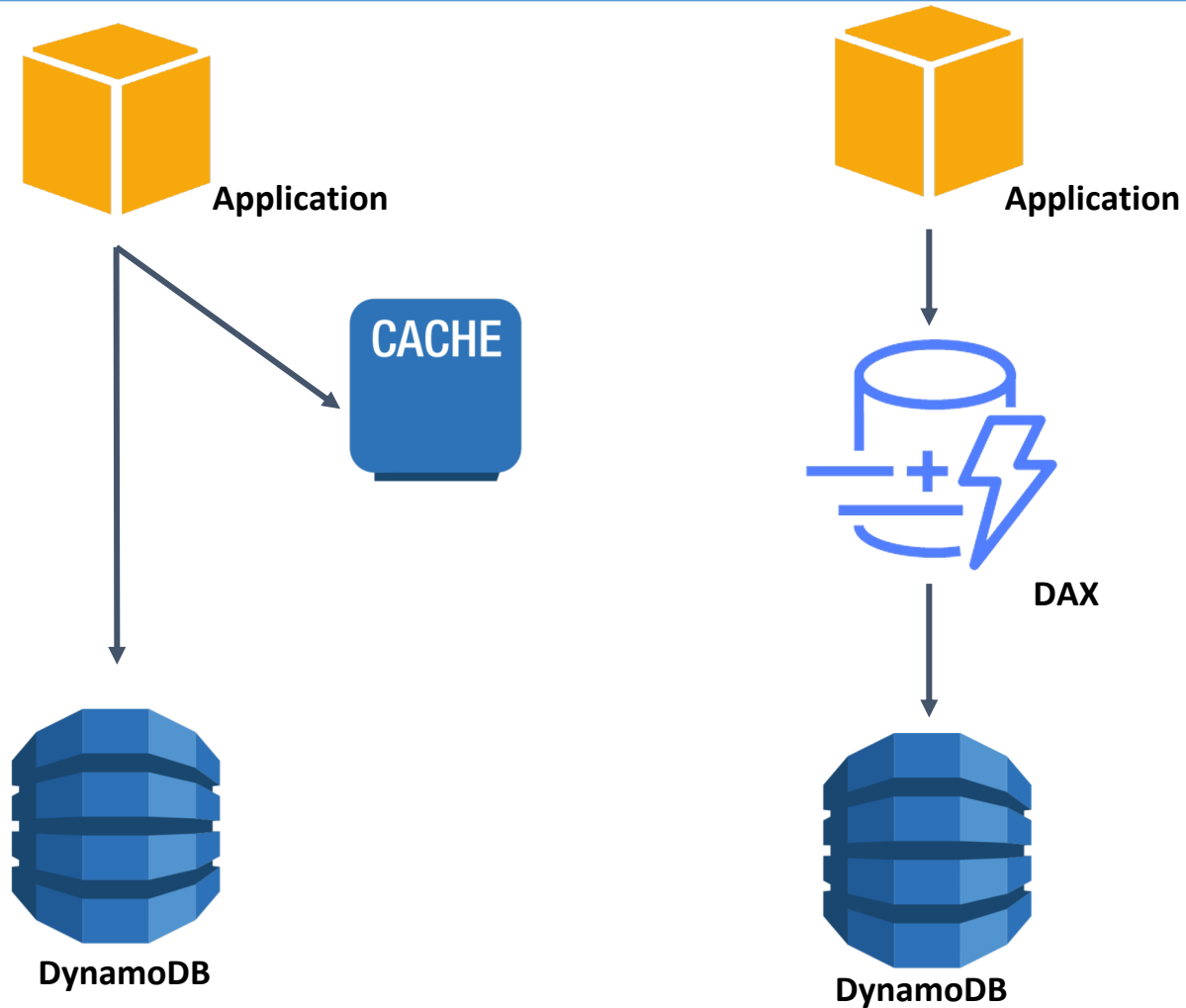
DynamoDB Accelerator (DAX)?

- Sử dụng làm Cache cho DynamoDB (In-Memory Cache)
- High Availability, do AWS quản lý (AWS fully managed)
- Hiệu năng tăng gấp x10 lần (milliseconds to microsecond)
- Developer không cần xử lý Caching Logic
- Tương thích với tất cả các lời gọi API của DynamoDB

DynamoDB Accelerator use cases?

- Sử dụng khi có yêu cầu phản hồi nhanh (fast response time) cho các yêu cầu đọc dữ liệu (Read)
- Lặp đi, lặp lại quá trình đọc cho một tập dữ liệu lớn (large set of data)

DynamoDB Accelerator use cases?

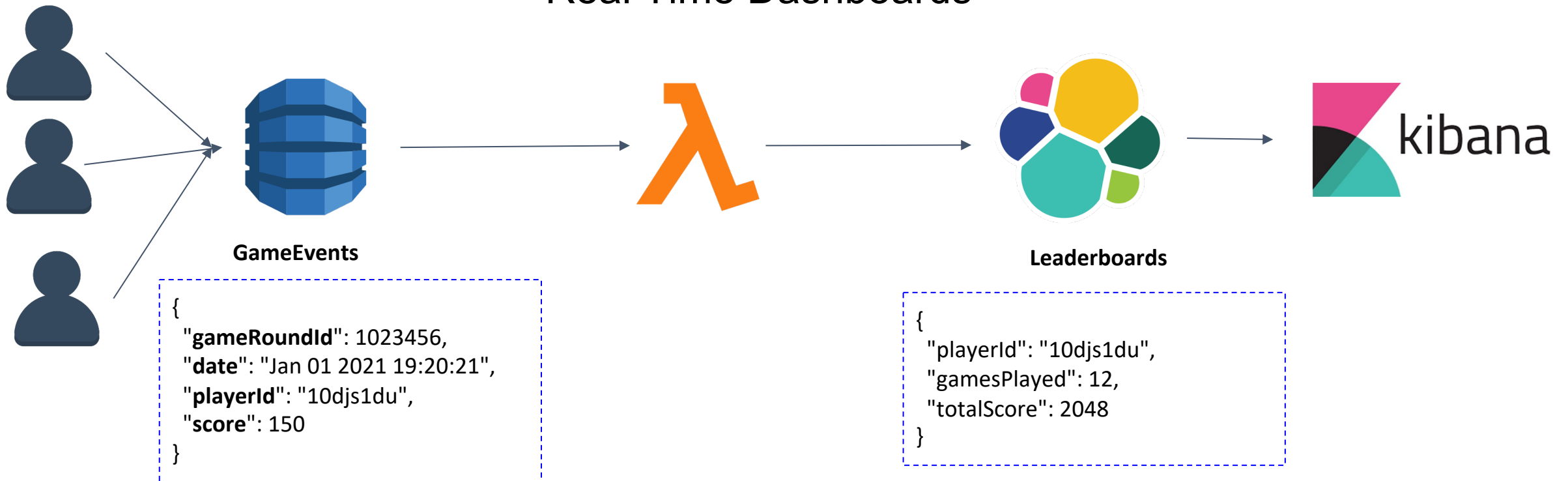


DynamoDB Stream?

- Cho phép tạo ra một event khi có một row bị thay đổi trong DynamoDB table
- Events có thể là hành động INSERT, UPDATE hoặc REMOVE
- Events có thể cho phép thu thập nội dung của row bị thay đổi
- Events được đảm bảo theo đúng trình tự thời gian xảy ra của việc thay đổi
- Xử lý theo Batch (Batch processing)
- Không ảnh hưởng tới hiệu năng của DB

DynamoDB Stream Use cases?

Real Time Dashboards



Elasticache

Elasticache

- AWS managed Key/Value Cache
- Giúp tăng hiệu năng của ứng dụng
- Hiện tại Elasticache hỗ trợ 2 loại Opensource in-memory engine



Memcached và Redis?

Requirement	Memcached	Redis
Sub-millisecond	Yes	Yes
Data partitioning	Yes	Yes
Scale horizontally	Yes	Yes
Multithreaded architecture	Yes	-
Advanced data structures	-	Yes
Persistence	-	Yes
Multi-AZ - Yes	-	Yes
Replication	-	Yes
Transactions	-	Yes
Pub/Sub	-	Yes
Backup and restore capability	-	Yes

Elasticache Security

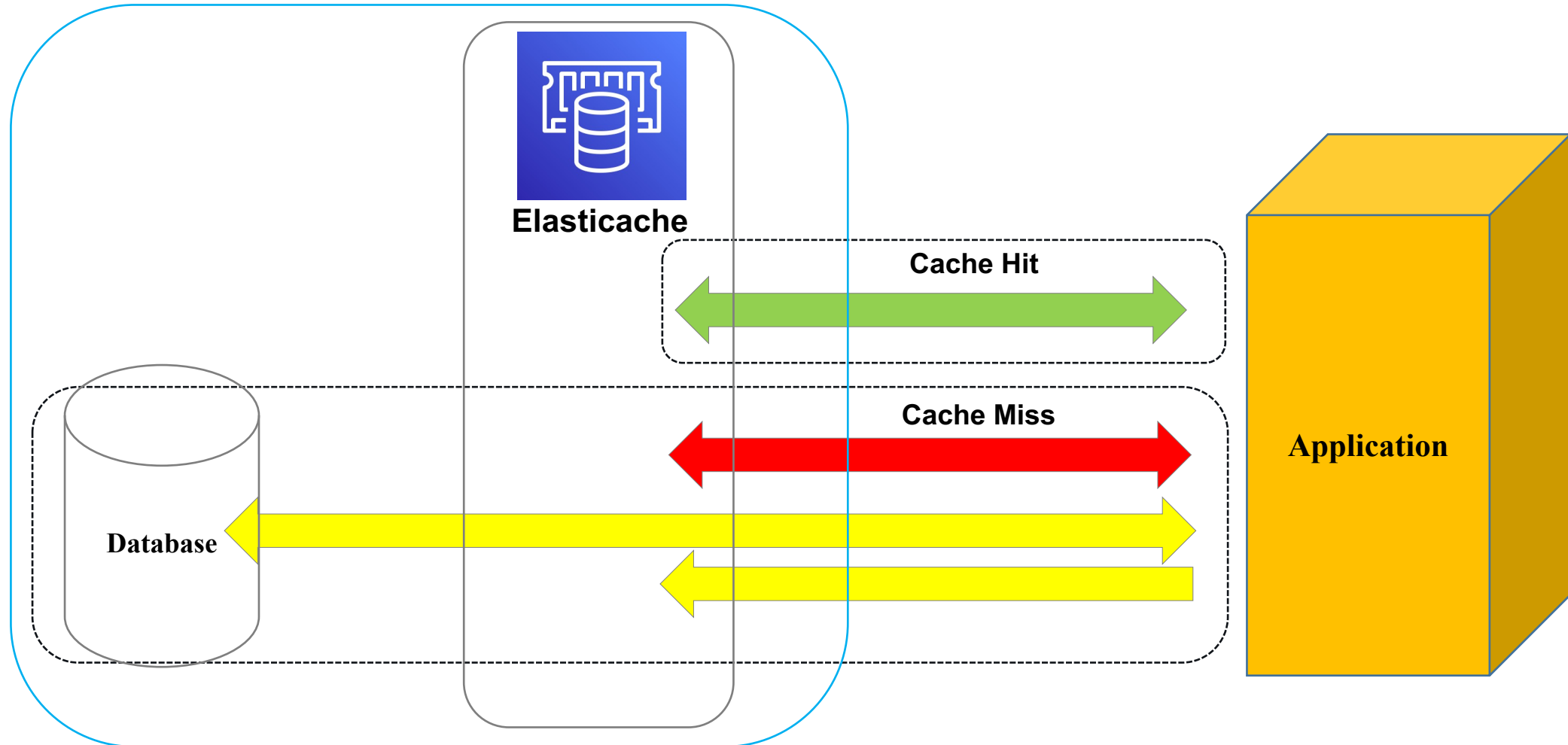
- Hỗ trợ mã hoá đường truyền (In-transit Encryption)
- Không hỗ trợ IAM cho Authentication (Chỉ cho phép trong quá trình tạo Cluster)
- REDIS AUTH có thể tạo **password/token** trong quá trình tạo Cluster
- REDIS AUTH là cho phép tăng thêm tính bảo mật bên cạnh Security Groups
- Memcached hỗ trợ Authentication dựa vào SAML (SAML-based authentication)

Caching strategy

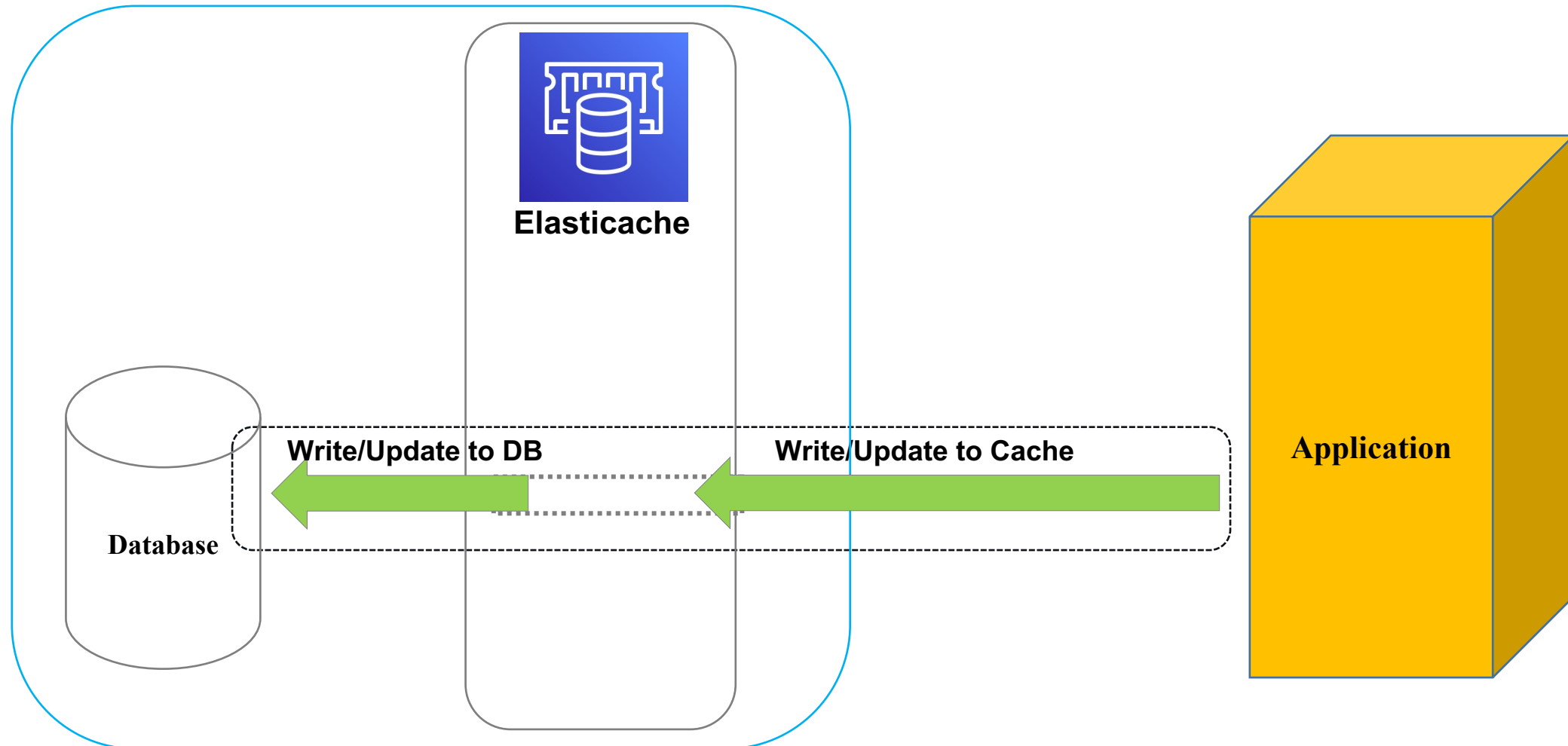
Caching Strategy

- Lady Loading
- Write-Through
- TTL

Lady Loading



Write-Through



TTL (Time To Live)

- Dữ liệu trong Cache được thiết một khoảng thời gian trước khi hết hạn
(Expiration) gọi là Time To Live (TTL)
- Sau khi dữ liệu bị hết hạn (Expired), ứng dụng sẽ không tìm thấy dữ liệu đó nữa
(Data not found)
- Có thể kết hợp Lazy-Loading và Write-Through để phù hợp với mục đích

Caching strategy comparison

	Lazy-Loading	Write-Through
Advantaged	<ul style="list-style-type: none">• Chỉ Cache các dữ liệu được truy vấn• Node failure không ảnh hưởng tới ứng dụng	<ul style="list-style-type: none">• Dữ liệu không bao giờ bị cũ (Stale)
Dissavantaged	<ul style="list-style-type: none">• Cache miss => delay time• Dữ liệu bị cũ	<ul style="list-style-type: none">• Dữ liệu bị lỗi khi node bị lỗi• Lãng phí tài nguyên

Exam tips

- Elasticache được sử dụng để tăng hiệu năng của ứng dụng
- Memcached hỗ trợ multi-thread trong khi Redis chỉ hỗ trợ Single-Thread
- Redis cluster có thể trải ra nhiều AZ (Multi-AZ), tự động failover
- Sub-milliseconds response cho các yêu cầu đọc
- Chi phí (Cost) được tính gồm phí **EC2 Instances + Storage**

Redshift

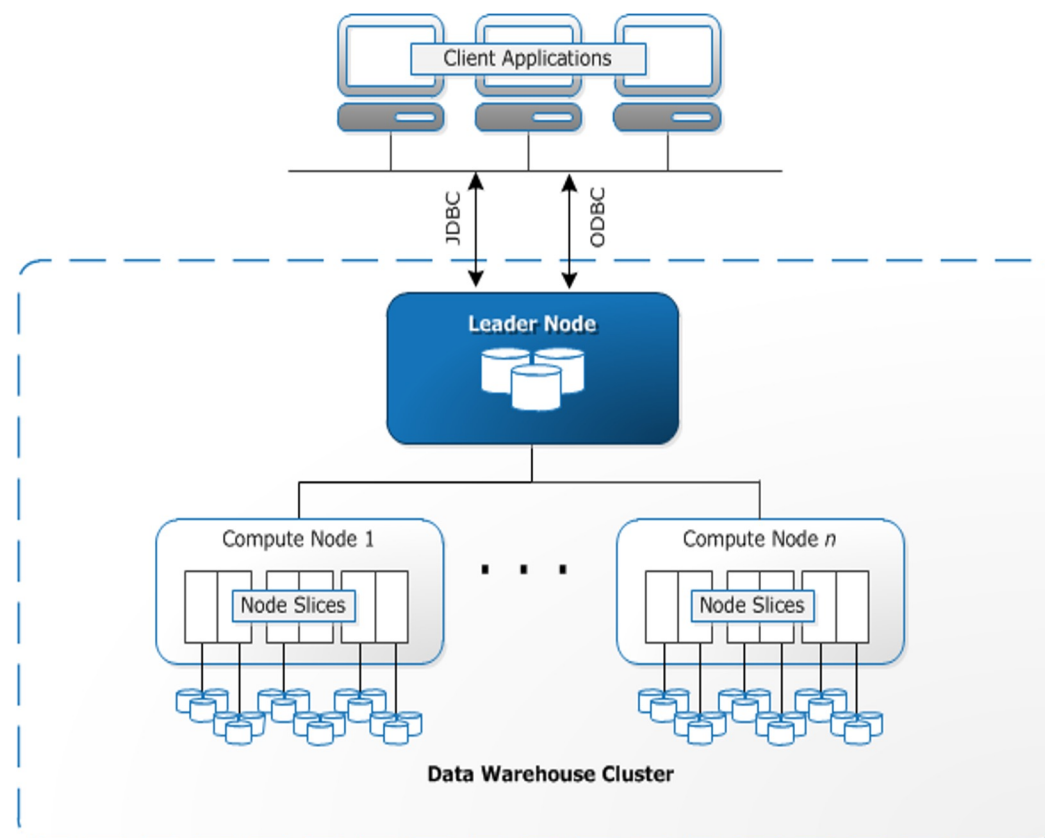
Redshift

- Redshift là OLAP (Online Analytical Processing), RDS là OLTP (Online Transaction Processing)
- Redshift được sử dụng cho BI và Data Warehousing
- Lưu trữ dạng Columnar Data
- Massively Parallel Processing (MPP)



Redshift Configuration

- Single Node (master + compute trong cùng 1 Instance)
- Multi-Node
 - **Leader Node:** Quản lý kết nối từ Client, nhận các Queries và tổng hợp kết quả (aggregation)
 - **Compute Node:** Lưu trữ dữ liệu và thực hiện các Queries và tính toán (computation)



Redshift Availability and Backup



- **Availability**

- Hiện tại chỉ cho phép Setup Cluster trong cùng một AZ
- Có thể khôi phục Cluster sang một AZ khác sử dụng Snapshot

- **Backup**

- Mặc định được enable với thời gian lưu (Retention) là 1 ngày
- Lưu tối đa 35 ngày (up to 35 days retention)
- Snapshot được lưu trên S3 và có thể đồng bộ sang Region khác

Security for Redshift

- Mã hoá đường truyền (In-transit Encryption)
- Mã hoá lưu trữ (Encrypt at Rest)
 - Key qua HSM
 - KMS (Key Management Service)



Exam Tips

- Redshift được sử dụng cho BI và Data Warehousing
- Redshift là OLAP (Online Analytical Processing)
- Chỉ setup Cluster được trong 1 AZ (Không span được giữa các AZ)
- Mặc định Backup được enable và lưu trong vòng 1 ngày, tối đa 35 ngày
- Redshift cho phép tạo bản backup snapshot sang S3 ở region khác phục vụ Disaster Recovery

Database Migration Service

Database Migration Service (DMS)

- Dịch vụ cho phép migrate Databases, NoSQL, Data Warehouse in/out AWS Cloud
- Hỗ trợ 2 kiểu migration

- Cùng DB engine (homogenous)

Ex: Oracle => Oracle

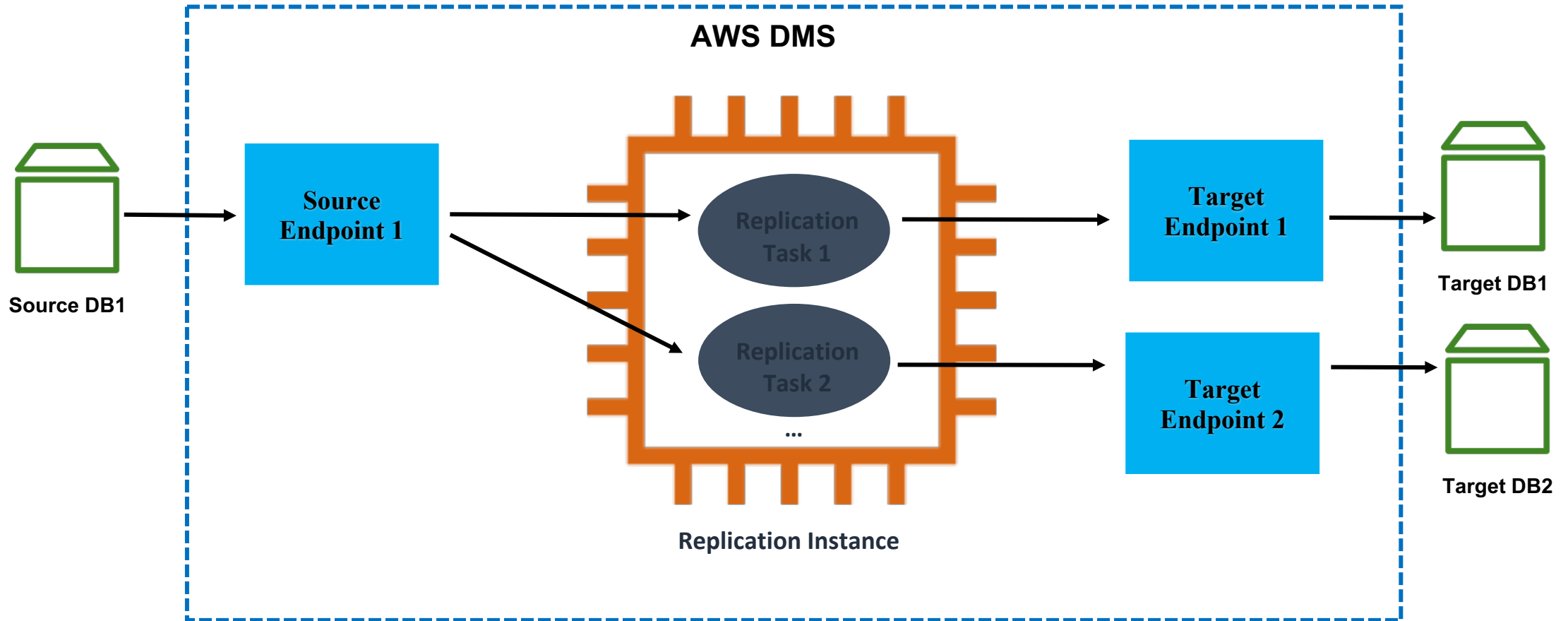
- Khác DB engine (**heterogeneous**)

Ex: SQL Server => Amazon Aurora



DMS

Components of DMS










Exam Tips

- DMS cho phép migrate Relational DB, Data Warehouse, NoSQL in/out AWS
- Source DB có thể đặt ở On-Premise, AWS cloud, Azure...
- Có thể migrate với kiểu cùng DB engine (homogenous) hoặc khác DB engine (Heterogeneous)
- Cần sử dụng **AWS SCT** (Schemal Conversion Tool) để chuyển đổi Schema trong trường hợp migrate kiểu Heterogeneous (Khác DB engine)

Choosing Right DB solution

Do **NOT** fit all things in one place

Common data categories and use cases

						
Relational	Key-value	Document	In-memory	Graph	Time-series	Ledger
Referential integrity, ACID transactions, schema-on-write	High throughput, low-latency reads and writes, endless scale	Store documents and quickly access querying on any attribute	Query by key with microsecond latency	Quickly and easily create and navigate relationships between data	Collect, store, and process data sequenced by time	Complete, immutable, and verifiable history of all changes to application data
Lift and shift, ERP, CRM, finance	Real-time bidding, shopping cart, social, product catalog, customer preferences	Content management, personalization, mobile	Leaderboards, real-time analytics, caching	Fraud detection, social networking, recommendation engine	IoT applications, event tracking	Systems of record, supply chain, health care, registrations, financial