



CODESTAR

Database

CodeStar Academy

Nội dung chính

- RDS
- Các thành phần phụ trợ cho DB
- SQL - NoSQL



RDS

RDS là dịch vụ giúp chúng ta có thể dễ dàng khởi tạo và quản lý DB instance.



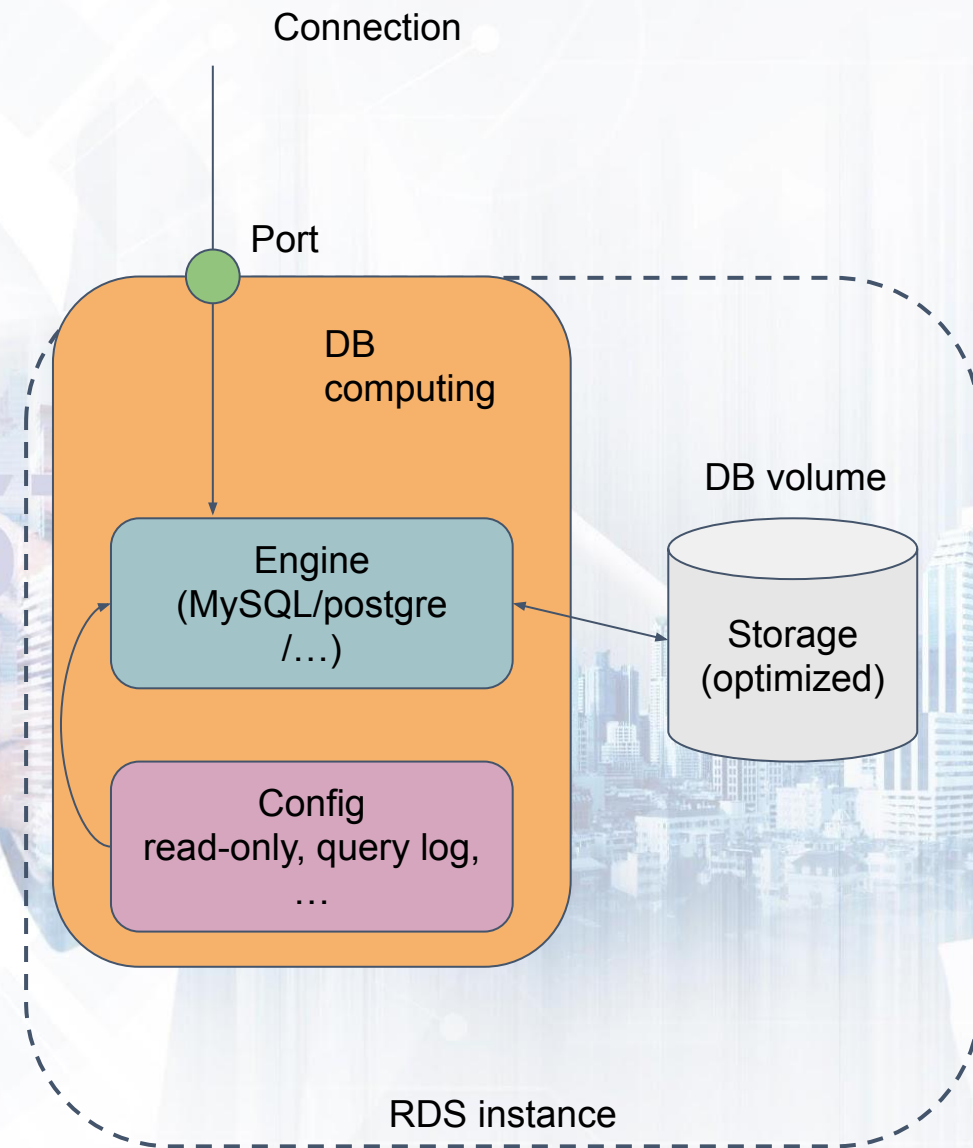
RDS vs EC2 Instance cài đặt DB

RDS	EC2 Instance with DB
Khởi tạo tự động theo lựa chọn đã chọn sẵn	Lựa chọn AMI, sau đó cài đặt phiên bản DB mong muốn.
Có sẵn các phương án backup	Phải tự thiết lập các phương án backup thủ công
Mở rộng dung lượng tự động	Mở rộng ổ đĩa thủ công
Có thể tạo các Read Replica	Thiết lập Replica thủ công
Bị giới hạn các Engine có thể khởi tạo	Không giới hạn các Engine có thể khởi tạo
Các thành phần đã được tối ưu cho việc truy xuất DB	Tùy ý lựa chọn

RDS

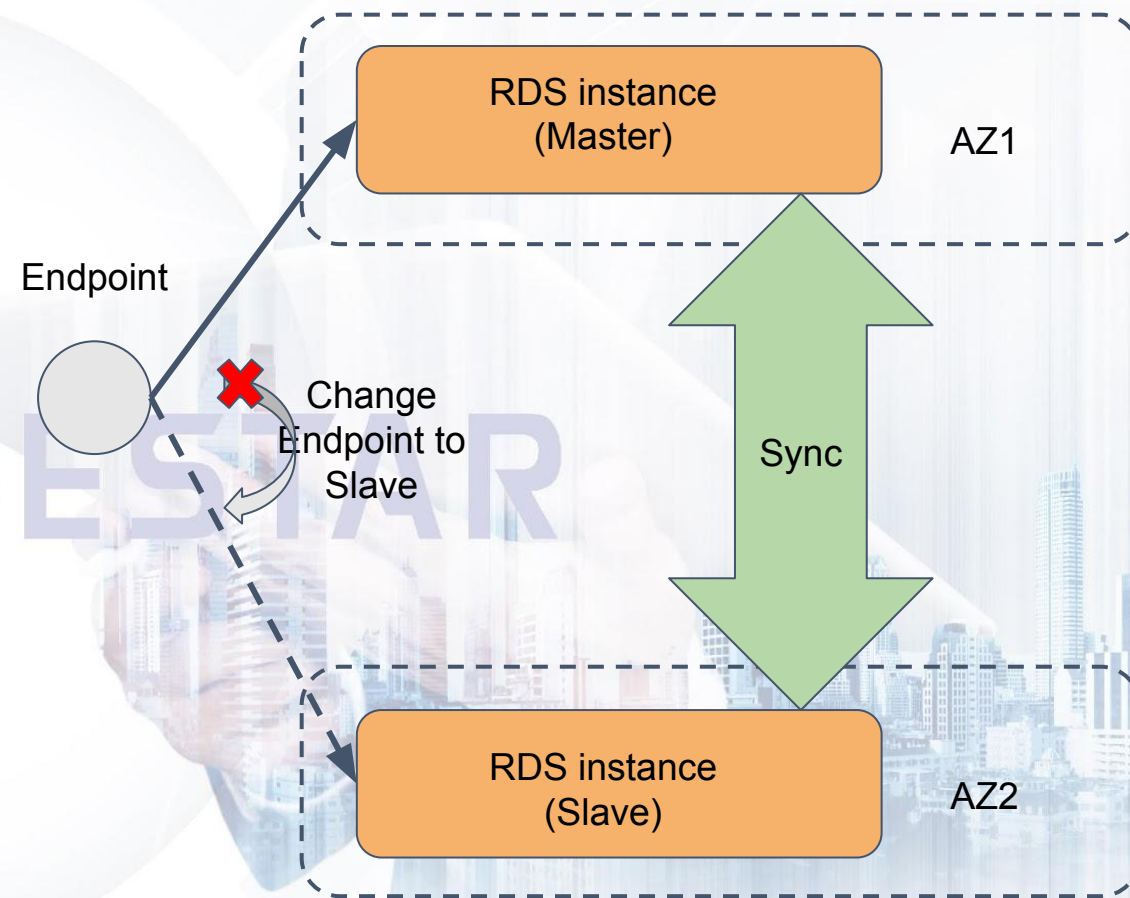
DB Instance

- Là một instance (tương tự EC2), đã thiết lập sẵn các thành phần chuyên biệt cho hoạt động của DB
- Cung cấp kết nối đến DB thông qua các port thông thường
- Không thể SSH trực tiếp tới DB host (như EC2) trong trường hợp bình thường.



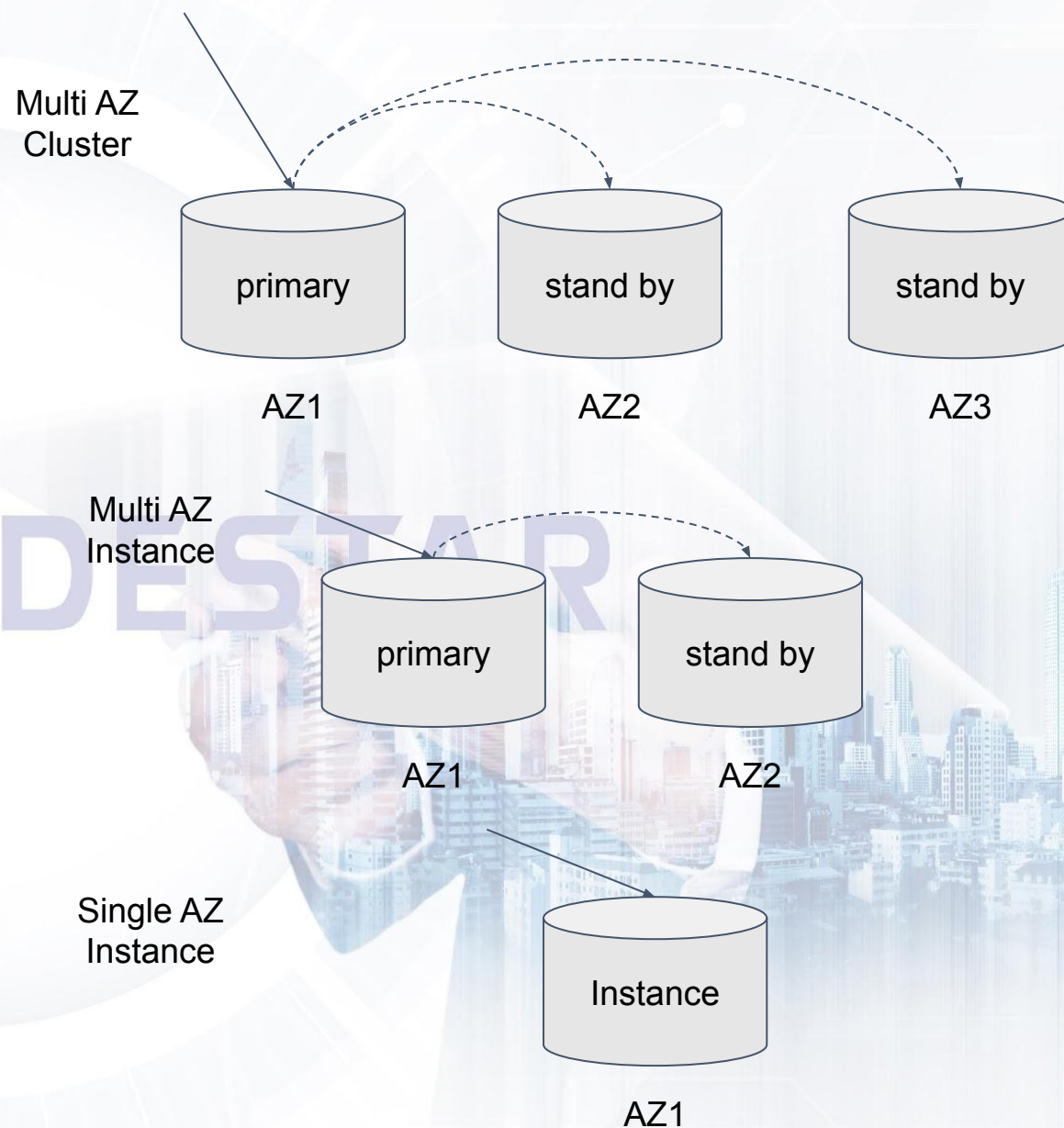
RDS: Multi AZ

- **Multi AZ** là tính năng cho phép RDS instance HA.
- Khi hệ thống khởi tạo RDS instance, sẽ tạo ra thêm một instance nữa tạo AZ khác.
- Chi phí tính như sử dụng 2 instance (gấp đôi)
- Đồng bộ giữa 2 RDS instance là synchronous replication, trong đó lúc nào 2 instance cũng có trạng thái giống nhau.



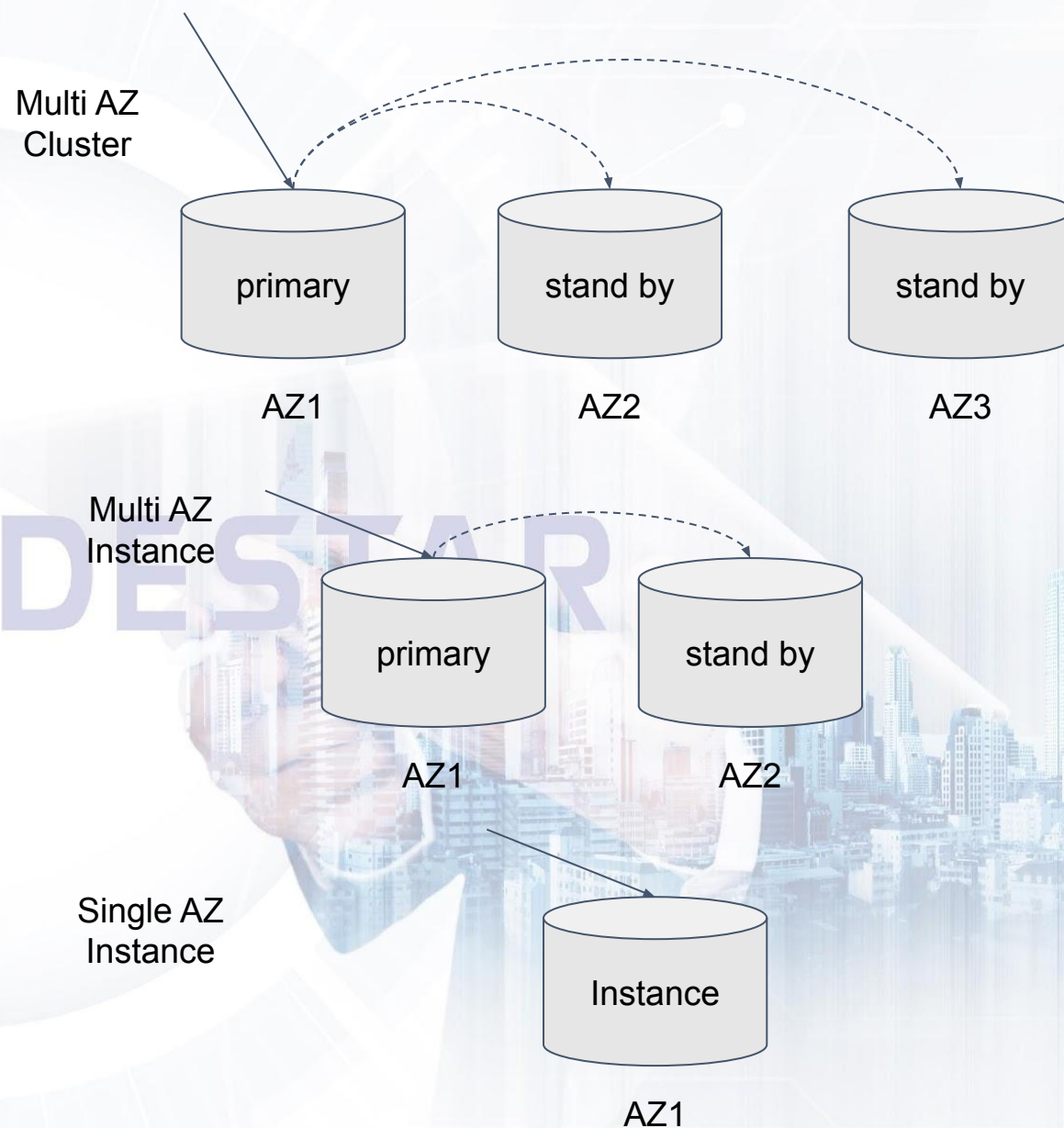
RDS: HA

- Để tăng tính HA cho các loại Database Engine, RDS sử dụng các công nghệ khác nhau.
- **Multi AZ Cluster:** Tạo 1 primary instance, và các stand by instance khác ở các AZ khác.
- **Multi AZ Instance:** 1 primary instance và 1 stand by instance
- **Single AZ Instance:** chỉ có 1 instance duy nhất.



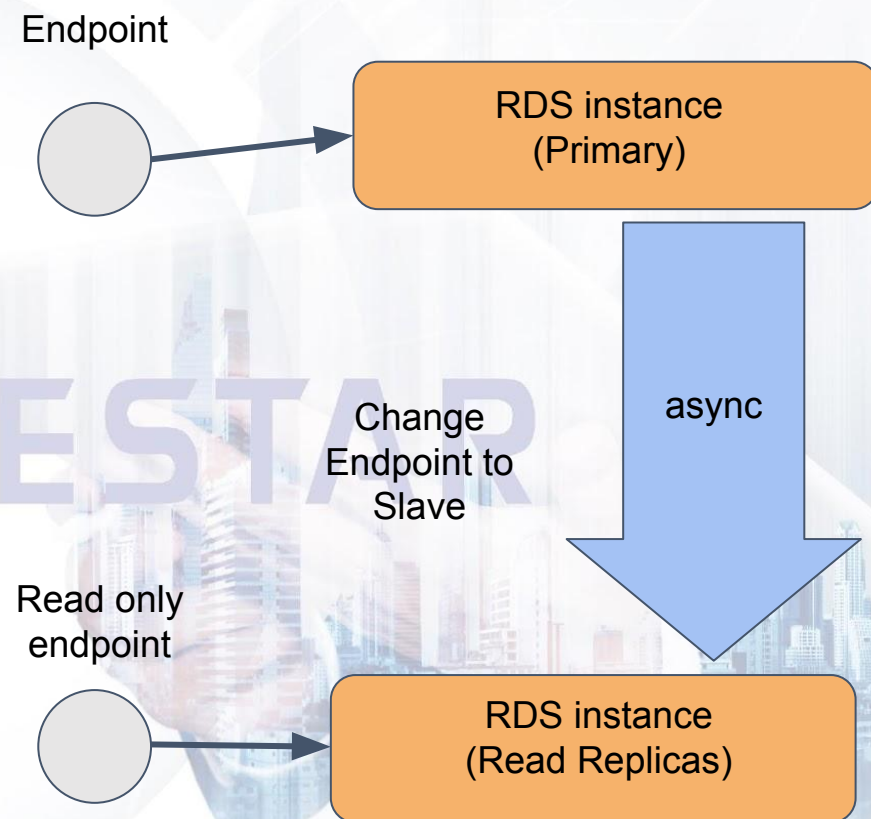
RDS: HA

- Để tăng tính HA cho các loại Database Engine, RDS sử dụng các công nghệ khác nhau.
- **RDS SQL Server:** Sử dụng chức năng Mirroring (nhưng chỉ có phiên bản Standard và Enterprise mới có)
- **Aurora:** Sử dụng Aurora Replica hoặc Reader node ở AZ khác.
- **MariaDB/Postgres/IBM:** Tạo Standby Instance
-



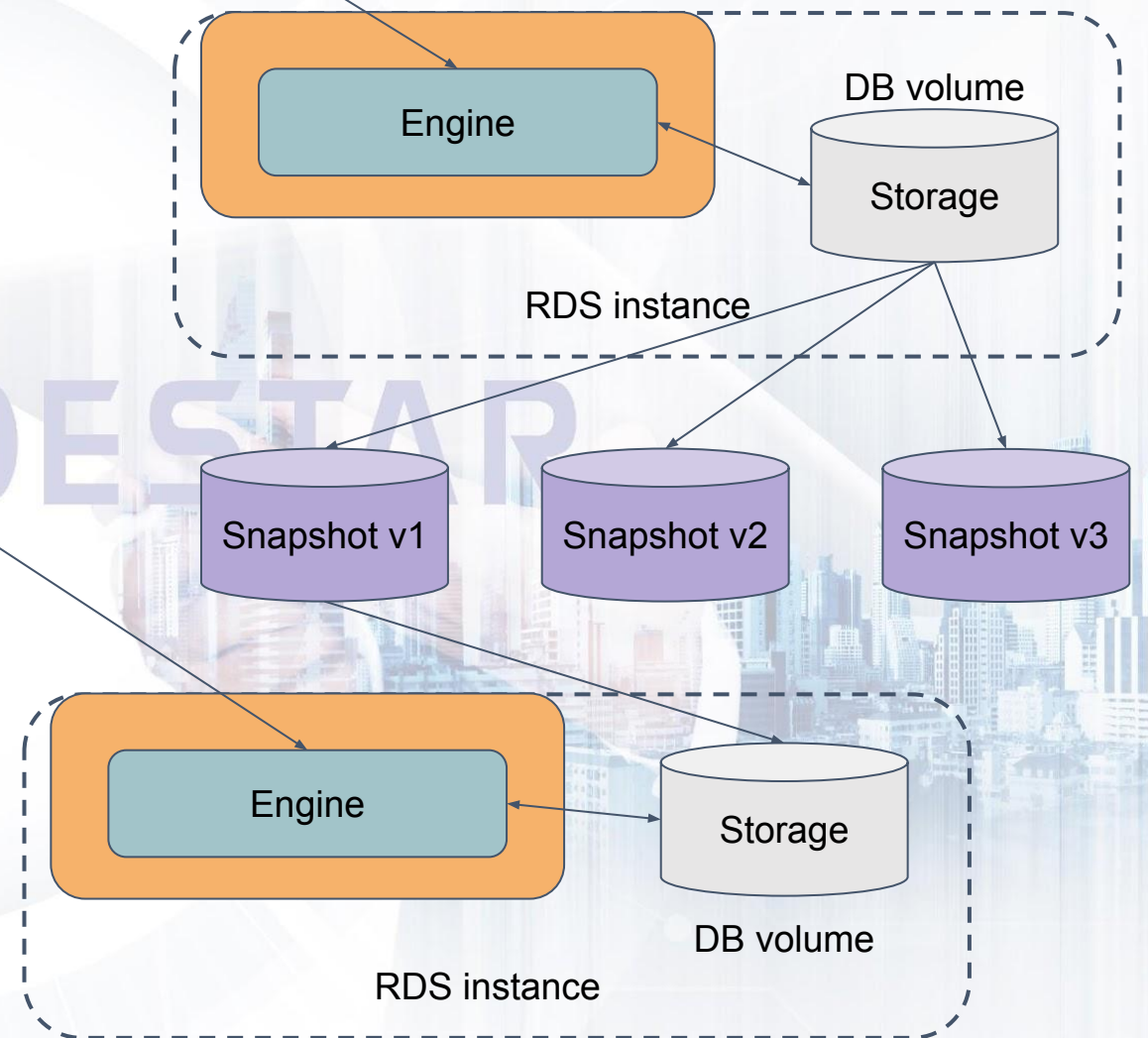
RDS: Read Replica

- **Read Replica:** tính năng cho phép tạo ra thêm một instance nữa, để tránh đưa toàn bộ request tới Primary instance.
- Read Replica sử dụng asynchronous replication, do đó có độ trễ nhất định so với Primary instance.
- Có một endpoint riêng phục vụ cho các tác vụ Read-only.



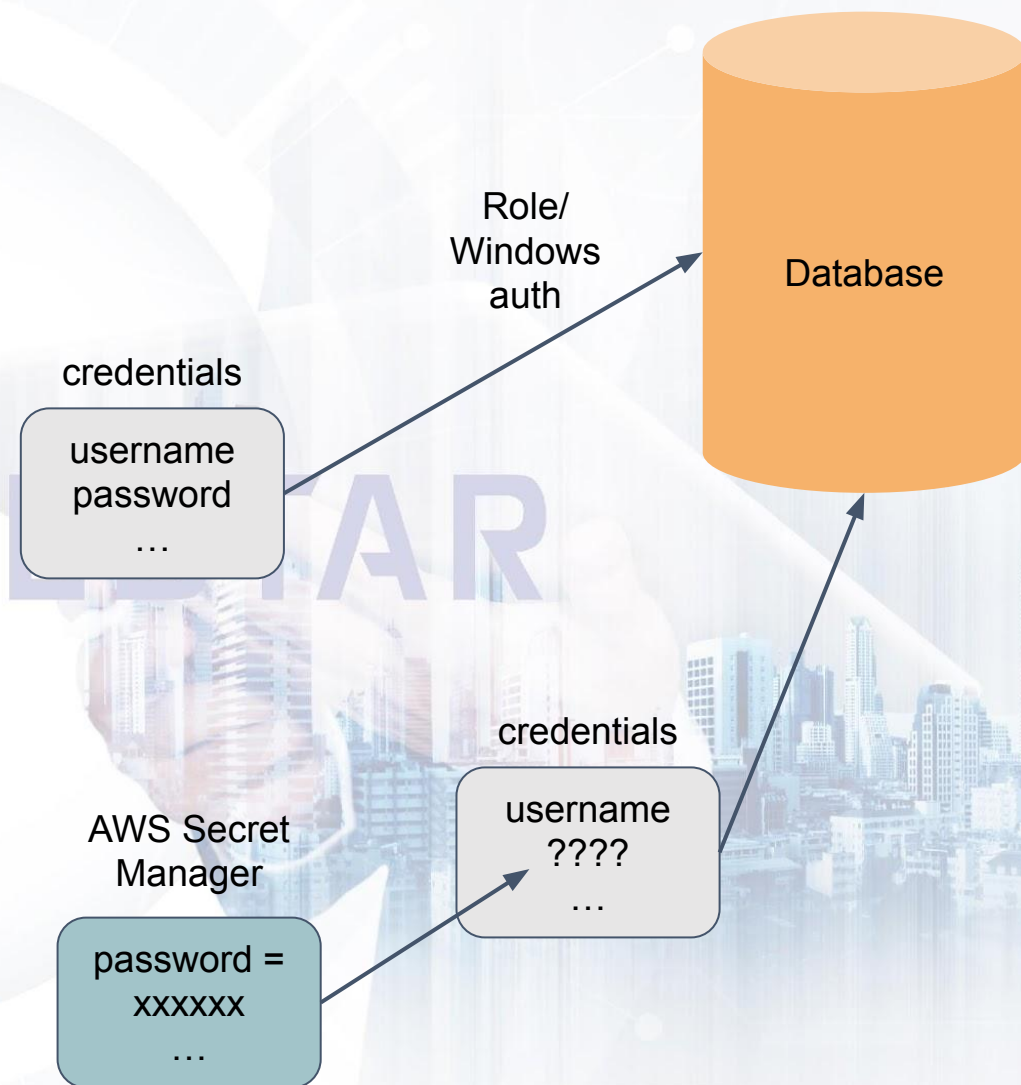
RDS: Backup

- RDS backup sử dụng Snapshot (tương tự EC2 Instances)
- Snapshot có thể được sử dụng để tạo lại một RDS instance khác tương đương.
- Snapshot sẽ được tạo ra định kỳ -> Automation Snapshot. Snapshot tạo ra tại thời điểm bất kỳ -> Manual Snapshot.



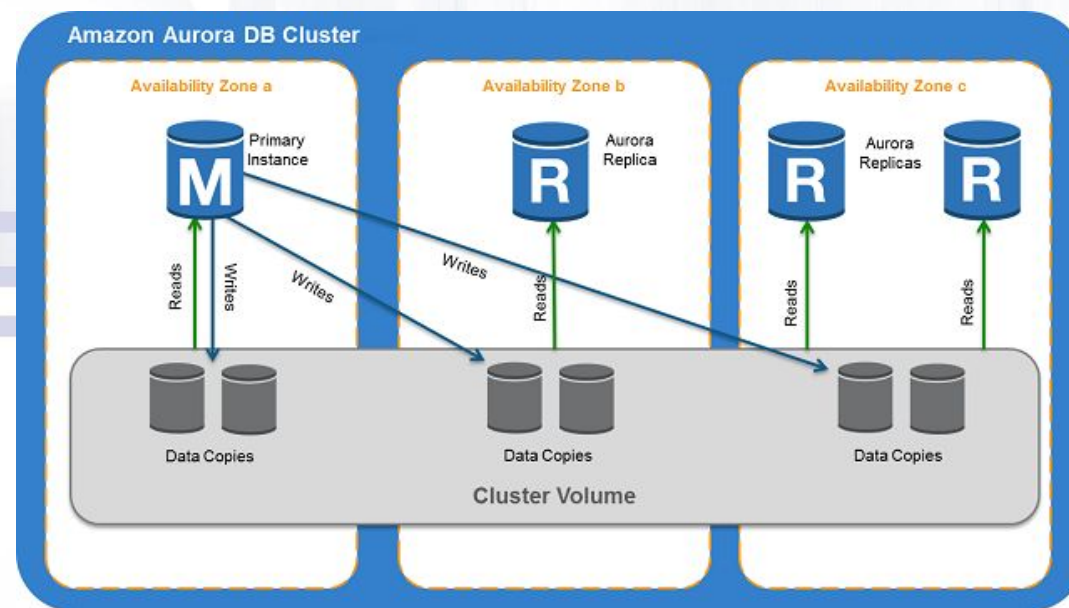
RDS: DB credentials

- RDS sử dụng credentials chủ yếu là **username** và **password**.
- Có thể tăng thêm bảo mật bằng IAM Role, Windows auth/ ... tùy vào loại Database.
- RDS credential có thể sử dụng **AWS Secret Manager**, một dịch vụ giúp quản lý các thông tin bí mật như password, key một cách hiệu quả. Password có thể đổi định kỳ sau 1 khoảng thời gian tự động.



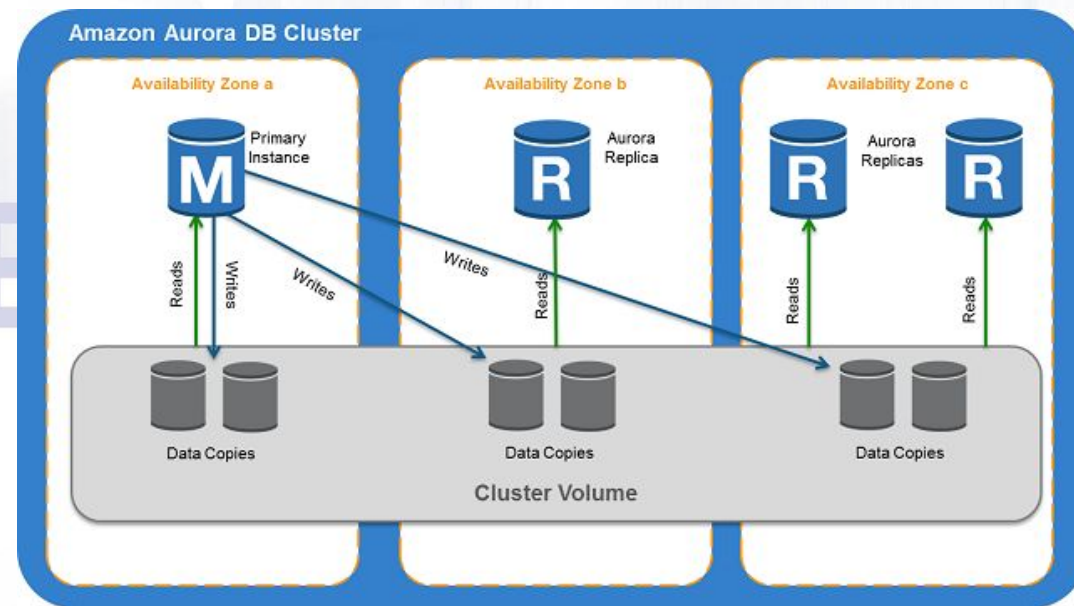
Aurora

- Aurora là loại RDS Engine đặc biệt do AWS phát triển
- Tính năng nổi trội so với RDS:
 - + Cho phép nhiều Replica hơn (15)
 - + Scale nhanh hơn, phù hợp cho trường hợp đọc nhiều.
 - + Giá cao hơn khoảng 20%
 - + Aurora sẽ phụ thuộc vào phía AWS đối với 1 số lỗi sẽ cần Support từ bên AWS.
 - + Aurora cho MySQL có 1 số giới hạn về engine, plugin.



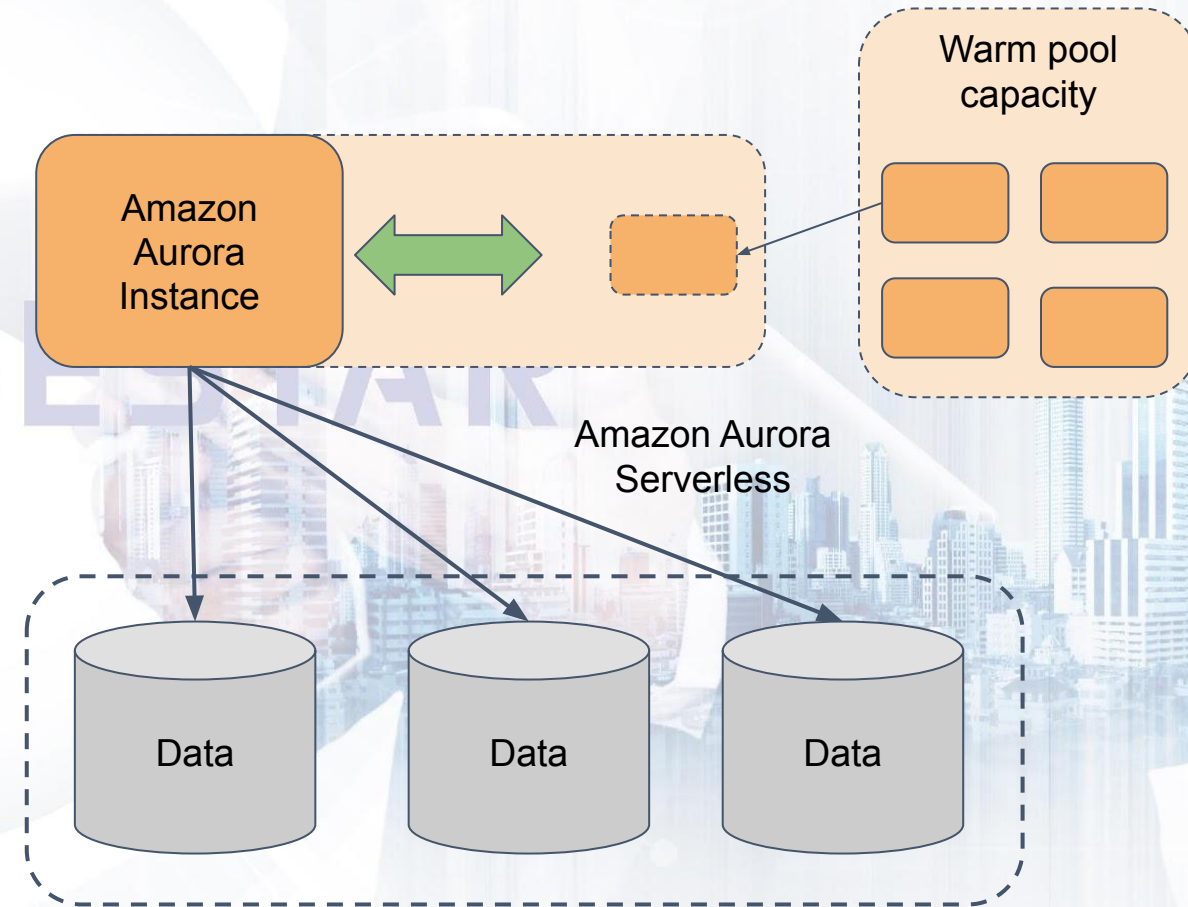
Aurora: Architecture

- Aurora mặc định sử dụng nhiều Cluster Volume nên có tính HA ngay trong thiết kế.
- Data được lưu trữ nhiều AZ, bất kể chúng ta sử dụng bao nhiêu Instance.
- Chỉ phải chi trả cho lượng data lưu trữ trong Aurora.
- Aurora cung cấp các Endpoint riêng biệt cho các instance Primary và Instance Replica.



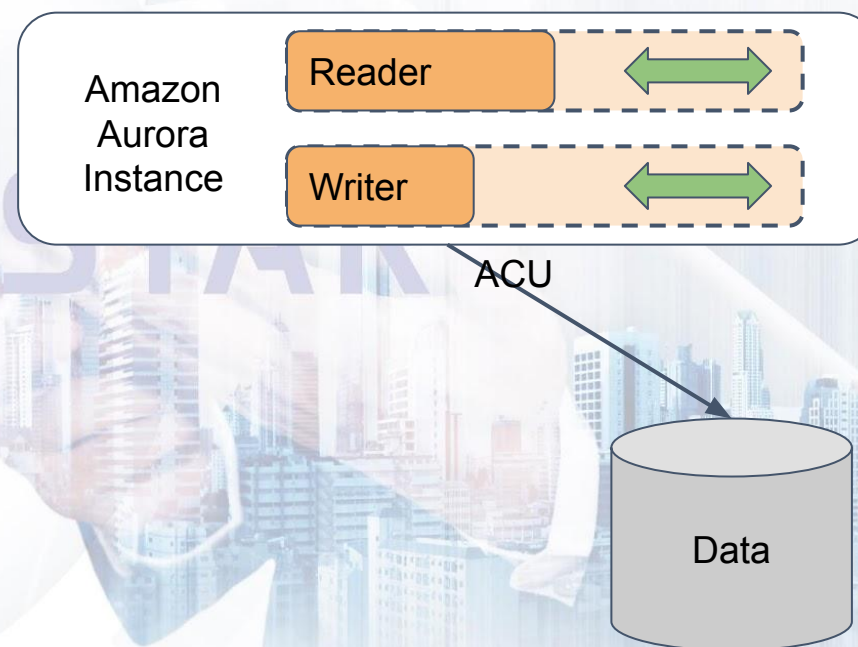
Aurora: Serverless v2

- Amazon Aurora Serverless có thể tự điều chỉnh lượng compute resource cần thiết.
- Khi cần tăng resource, Aurora sử dụng capacity từ Warm pool để scale gần như ngay lập tức.
- Khi không cần sử dụng computing resource, lượng capacity sẽ được release cho warm pool.
- Chúng ta sẽ chỉ phải chi trả cho lượng resource tối thiểu để duy trì instance.



Aurora: ACU (Aurora capacity unit)

- Aurora sử dụng một đơn vị đặc biệt để tính toán lượng resource sử dụng, đó là ACU.
- Mỗi ACU là tổng hợp của việc sử dụng 2GB RAM, kết hợp với lượng CPU, Network và 1 số thành phần khác tương ứng.
- Bộ phận ghi và bộ phận đọc sẽ có con số ACU khác nhau.
- ACU min là 0.5, max là 128, được đo trên giây.



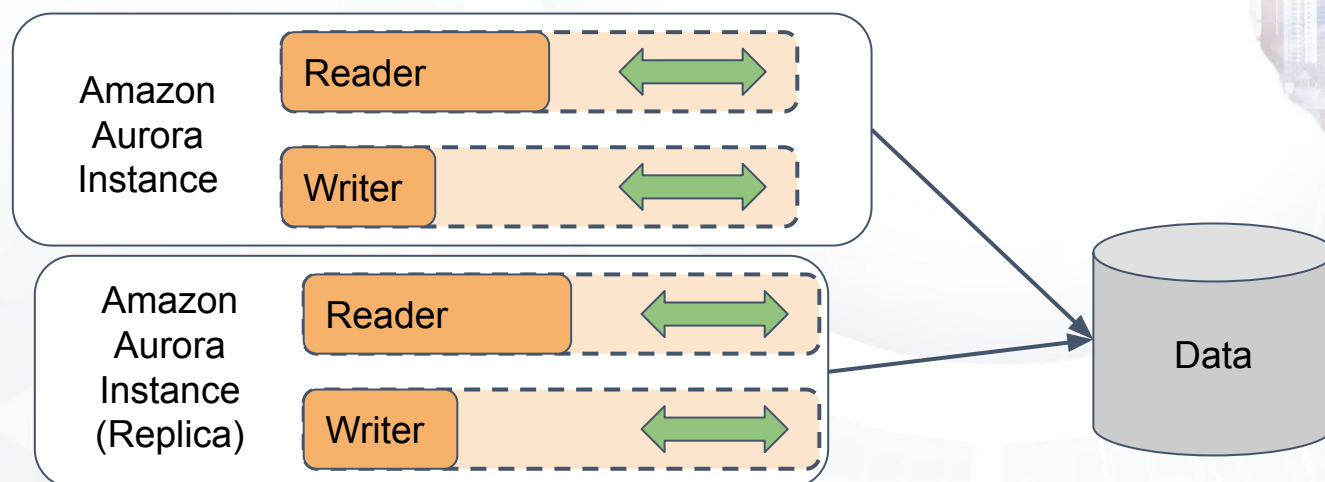
Note: Aurora Serverless v2 sử dụng Reader và Writer capacity riêng biệt

Aurora: ACU (Aurora capacity unit)

Khi sử dụng Multi AZ, Lượng Reader sử dụng sẽ là lượng ACU minimum đã thiết lập cho toàn bộ Replica (ở trạng thái nghỉ)

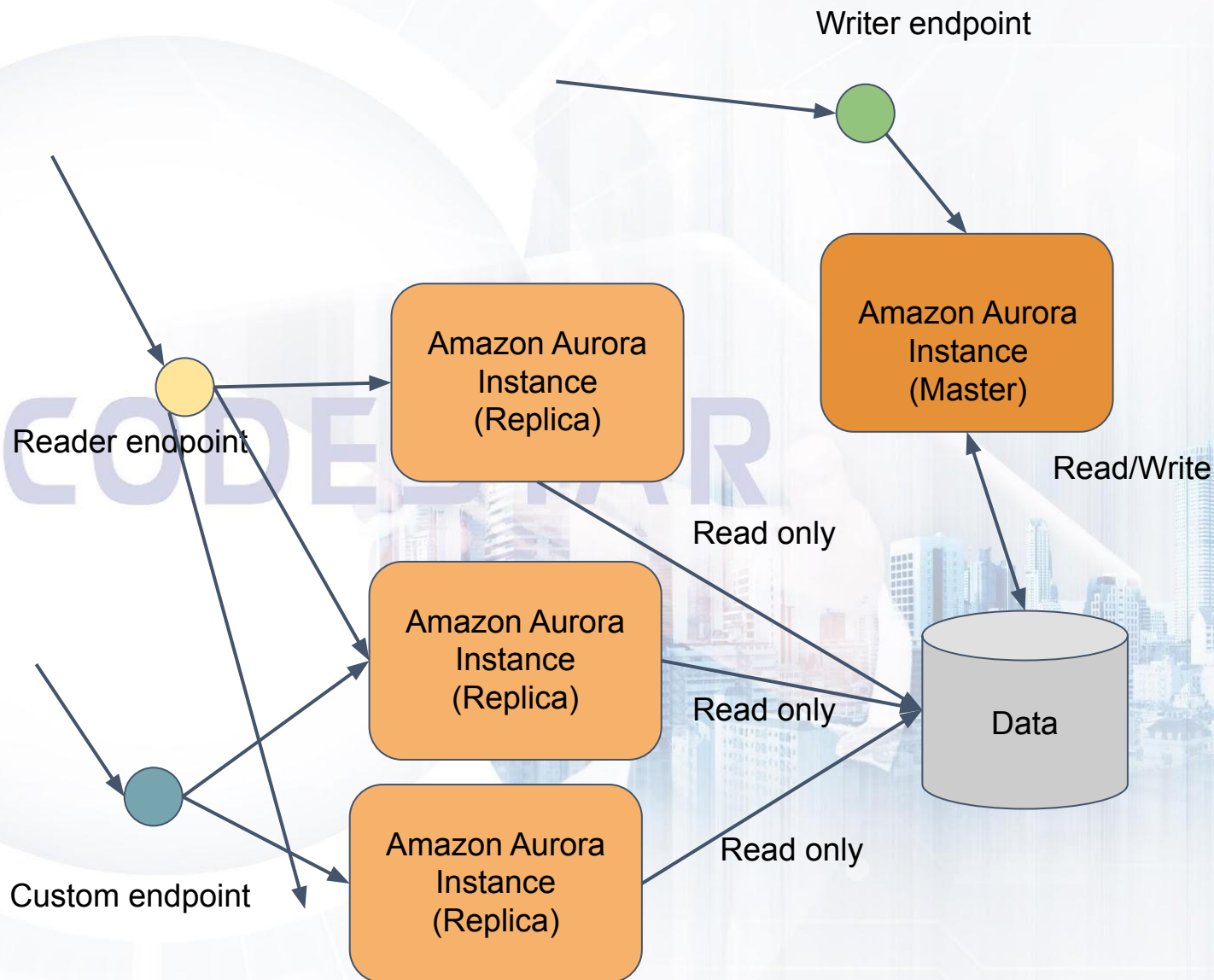
Số lượng Connection được phép kết nối tới Aurora Instance cũng phụ thuộc vào số lượng ACU maximum.

Max ACUs	Default max connections on Aurora MySQL	Default maximum connections on Aurora PostgreSQL
1	90	189
4	135	823
8	1,000	1,669
16	2,000	3,360
32	3,000	5,000
64	4,000	5,000
128	5,000	5,000



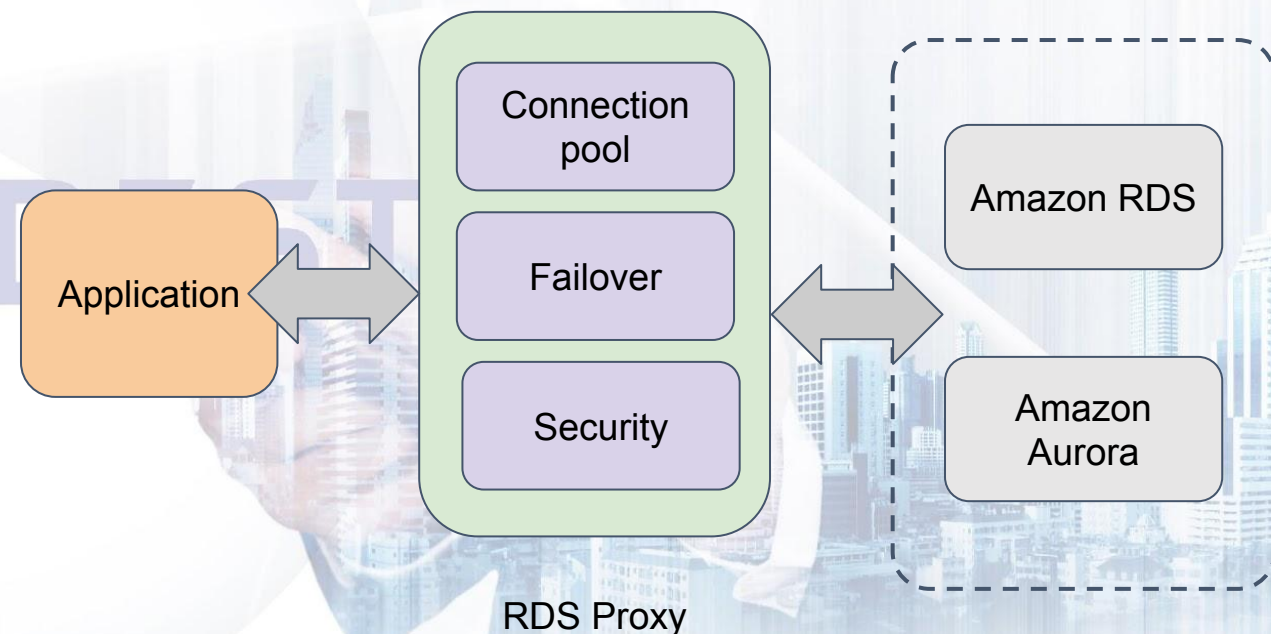
Aurora Endpoint

- Aurora sử dụng Endpoint để trở tới các DB Instance.
- Có 3 loại Endpoint:
 - + Writer endpoint: trở tới master DB Instance
 - + Reader Endpoint: trở tới tất cả Replica DB Instance
 - + Custom Endpoint trở tới một số Replica Instance.



RDS Proxy

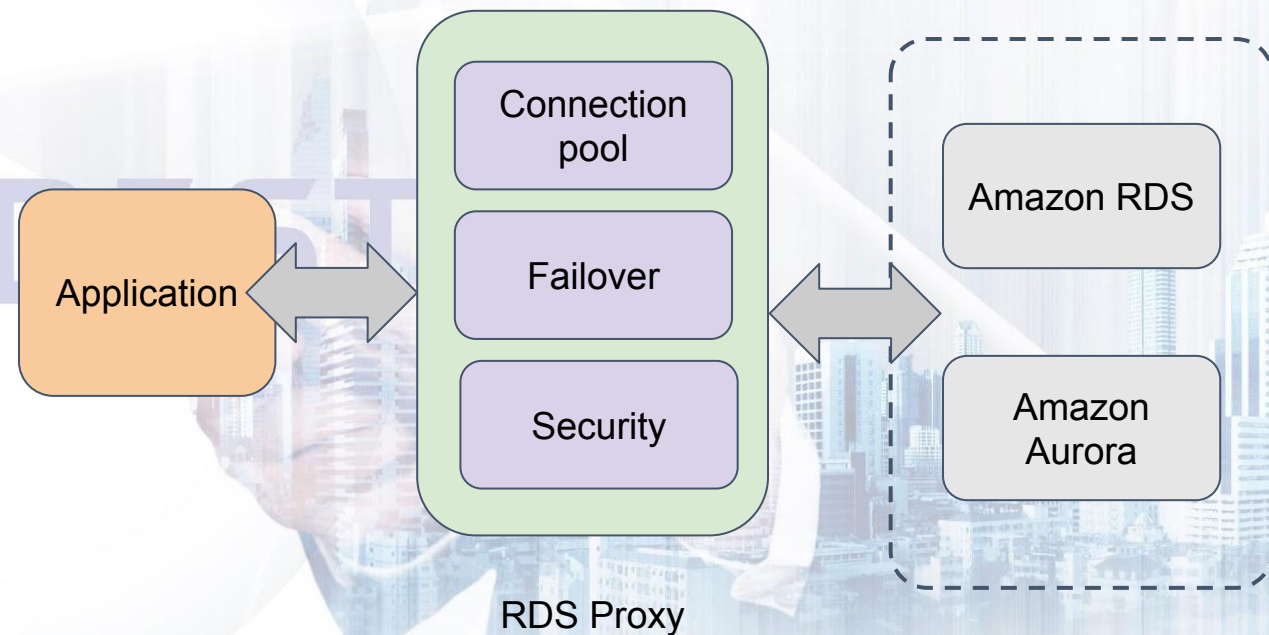
- RDS Proxy là một thành phần nằm giữa Application và Database
- RDS Proxy hỗ trợ kết nối tới DB một cách hiệu quả, tăng khả năng chịu lỗi.
- RDS Proxy tái sử dụng Connection tới DB hạn chế resource sử dụng cho các connection này khi dùng Lambda.
- Tạo lớp Security bảo vệ phía ngoài DB.



RDS Proxy

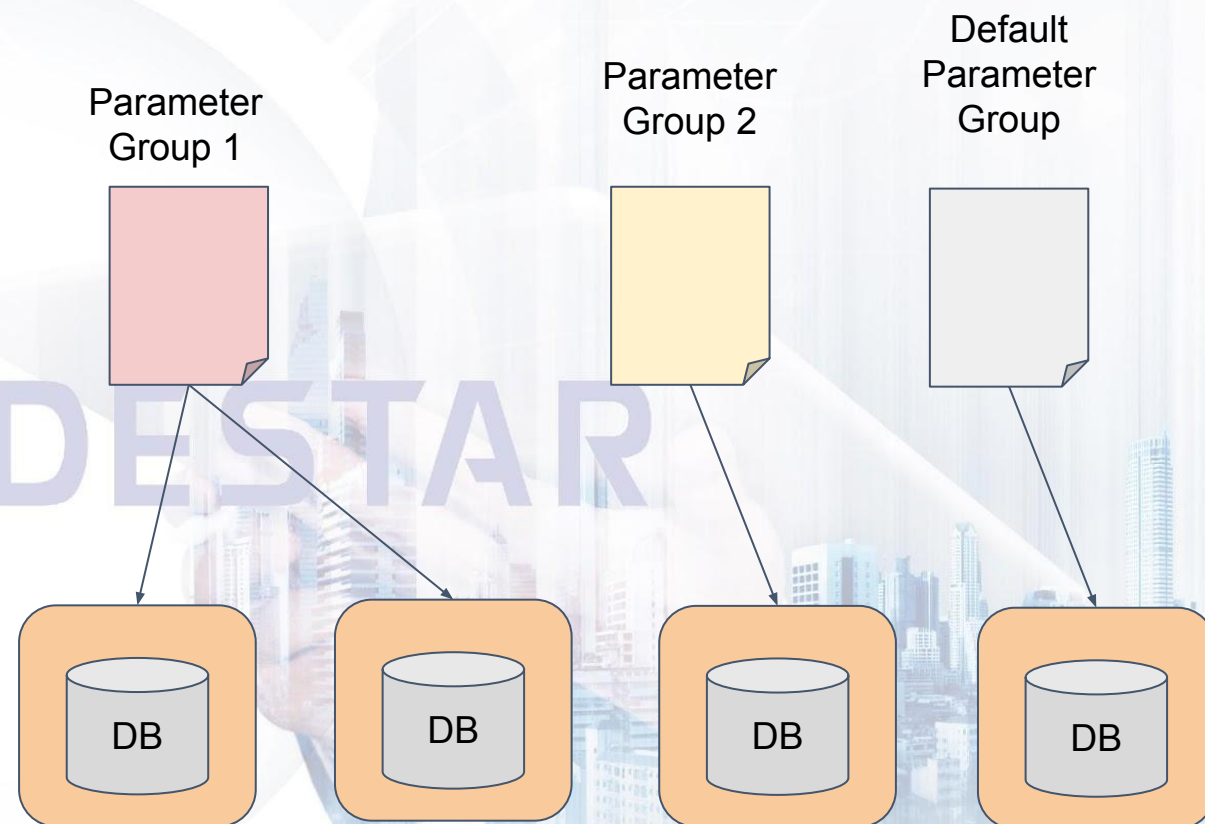
Use-case:

- Sử dụng Lambda kết nối tới DB ở tần suất cao, có thể có quá nhiều Connection kết nối tới DB -> Bị lỗi Too many connections.
- Có nhiều DB đồng bộ mà Application có thể kết nối tới, nếu 1 kết nối tới DB bị fail, có thể kết nối tới DB khác thay thế ngay.
- Tránh bị lộ DNS kết nối trực tiếp tới Database.



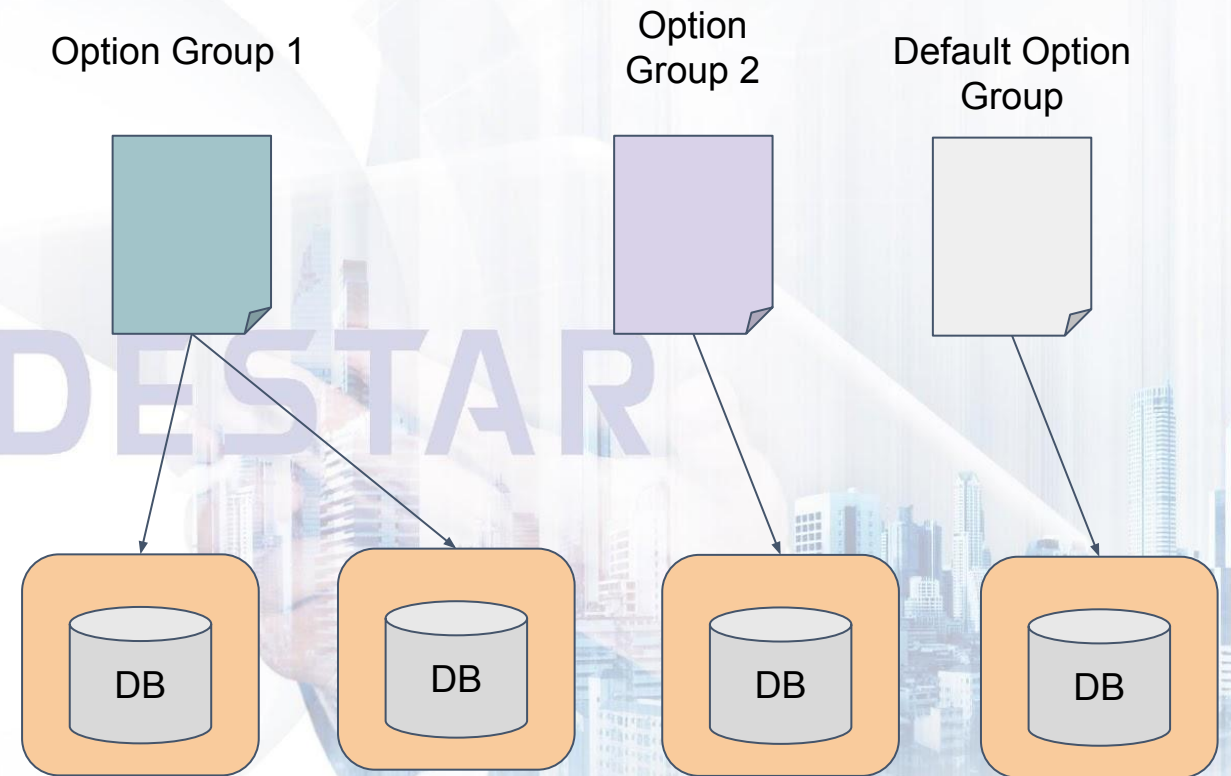
RDS: Parameter Groups - Option Groups

- Parameter Groups là tập hợp các tham số điều chỉnh cho DB engine.
- Chúng ta có thể gắn Parameter Groups vào các DB tương ứng.
- Khi chỉnh sửa các thông số trên một Parameter Group, các DB liên kết với Parameter Group đó đều sẽ thay đổi theo.



RDS: Parameter Groups - Option Groups

- Option Groups tương tự như Parameter Groups, nhưng đặc trưng cho việc bật/tắt các tính năng của DB.

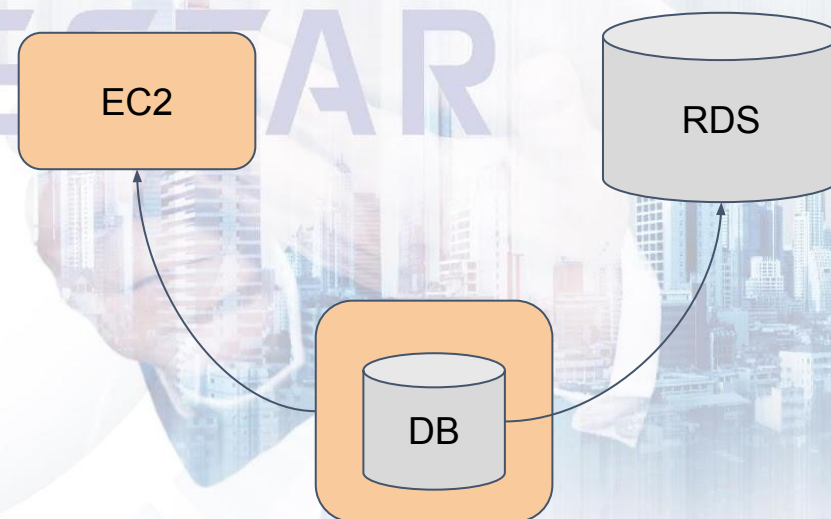


RDS: RDS Custom

- RDS Custom là tính năng hỗ trợ cho phép tùy chỉnh instance đang sử dụng để làm DB tại tầng OS.
- Hiện thị 1 EC2 riêng biệt cho phép SSH vào môi trường của instance.

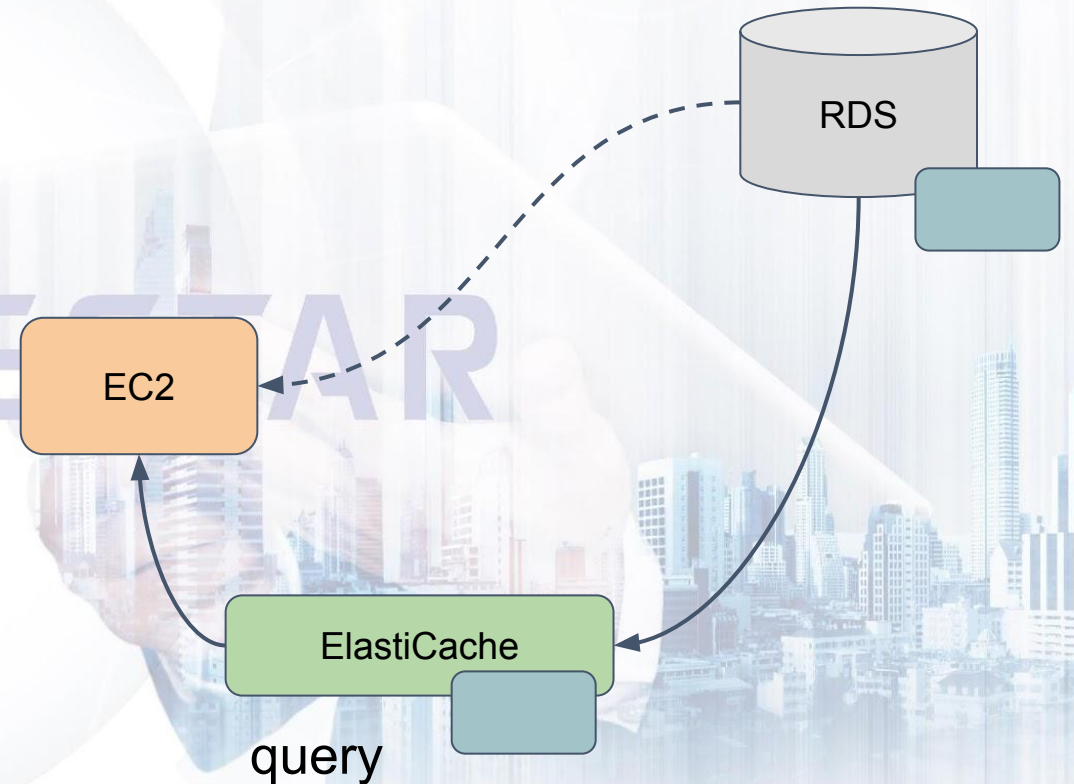
Use case:

- Cần truy cập, cài đặt các thông số tương tự như hệ thống cũ
- Cần customize các thành phần bên trong instance ở tầng OS.



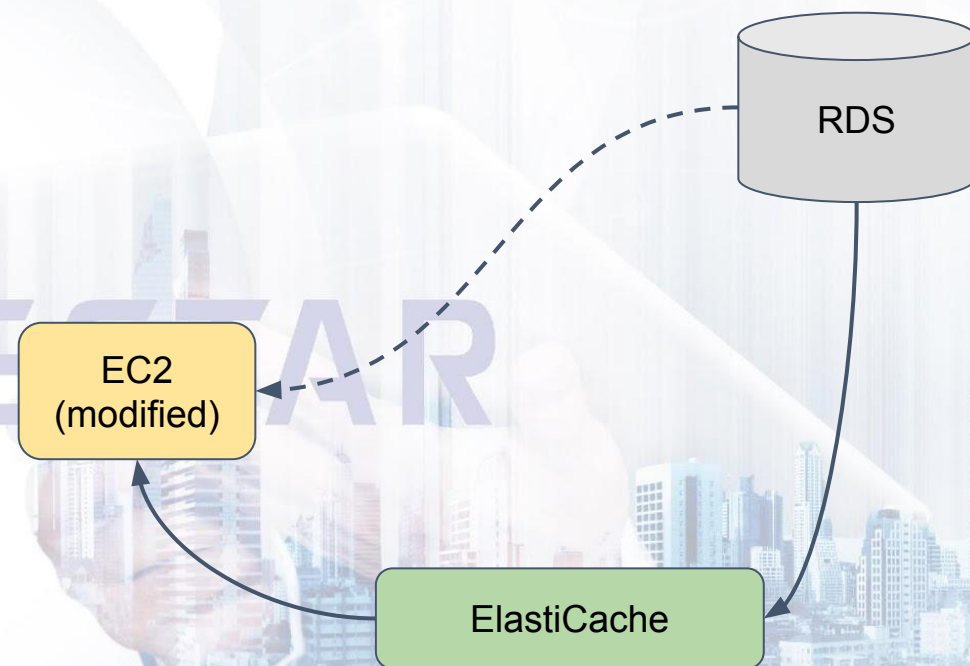
RDS: ElastiCache

- ElastiCache là dịch vụ hỗ trợ tăng hiệu năng khi sử dụng DB RDS.
- ElastiCache thay vì gọi thông tin trực tiếp từ RDS thì gọi thông qua ElastiCache.
- ElastiCache sẽ lưu lại những câu query thường xuyên gọi.
- Nếu ElastiCache trả về luôn data đã lưu trữ nếu có câu query cần gọi giống hệt.



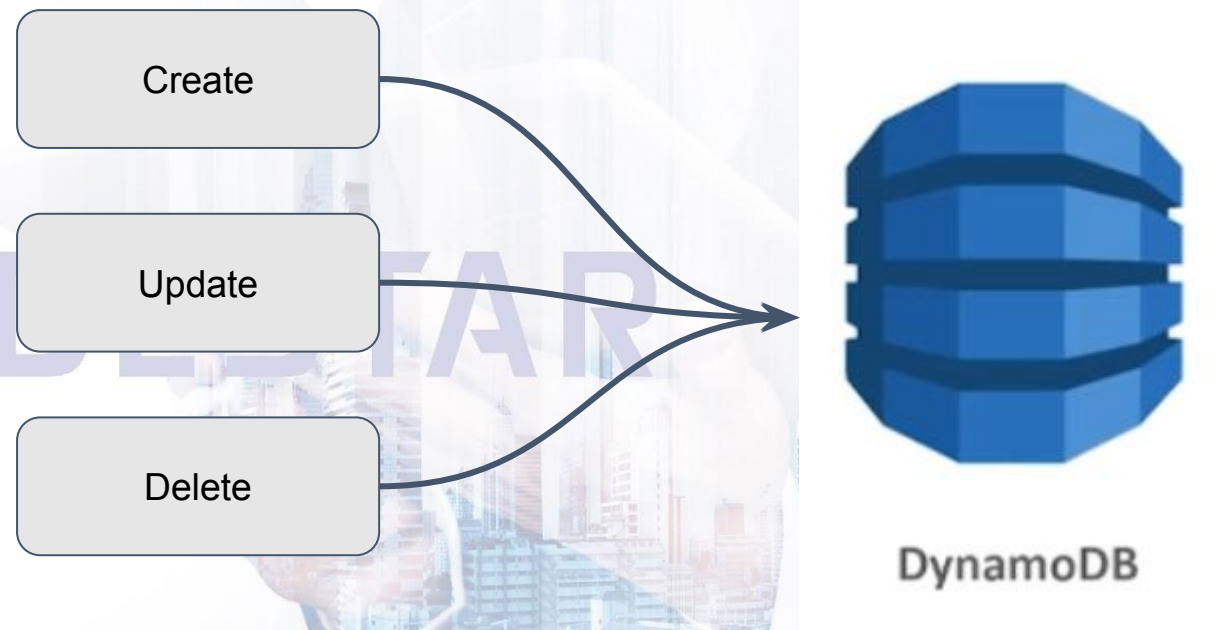
RDS: ElastiCache

- Đây cũng là phương án thường sử dụng với song song kèm với Read Replica, tuy nhiên ElastiCache cần update Code (thay đổi cách viết Code trên EC2/invoker)
- Không thể tùy ý thay đổi dễ dàng các instance DB.



DynamoDB

- Dịch vụ NoSQL
- Thuộc nhóm Serverless.
- Đã đảm bảo HA.
- Các hoạt động như create, insert, update, query, scan và delete được thực hiện trong bảng qua các API



DynamoDB

Use-case:

- Sử dụng cho những hệ thống đòi hỏi độ trễ cỡ mili giây
- Dữ liệu lưu trữ dạng phi cấu trúc hoặc bán cấu trúc, thường là dữ liệu nhỏ.
- Lưu trữ session data, hoặc những hệ thống nhỏ.

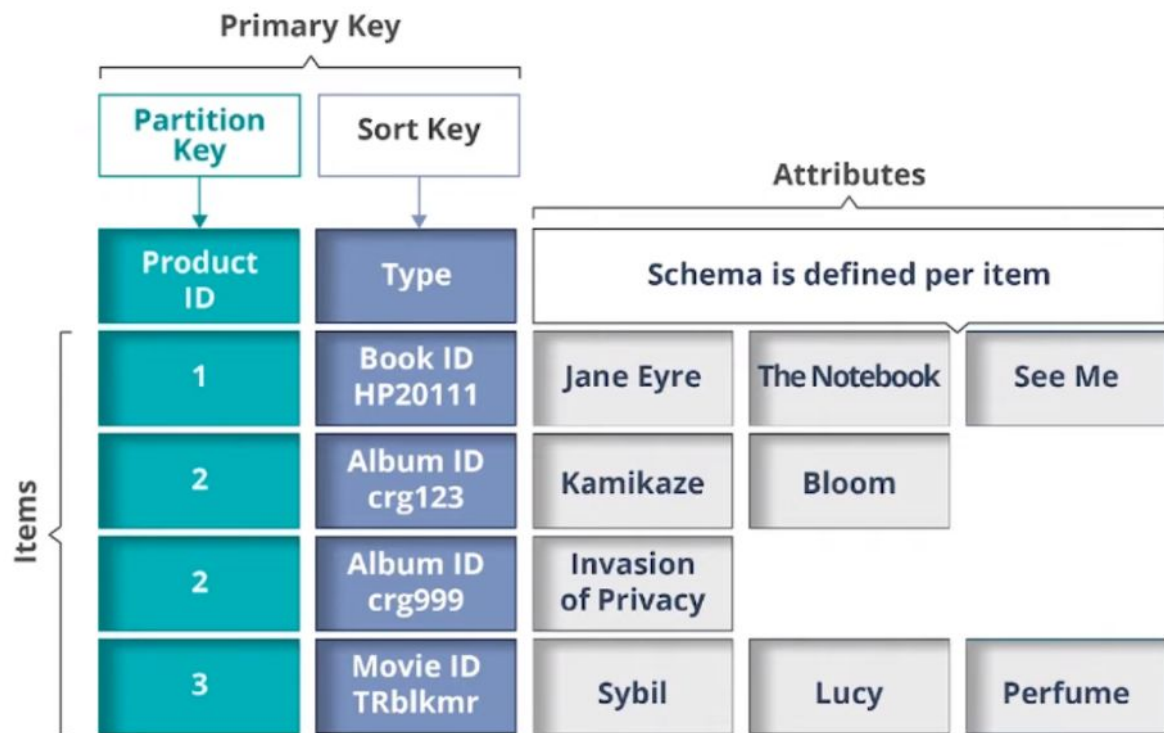


DynamoDB

DynamoDB

Đặc trưng của dữ liệu:

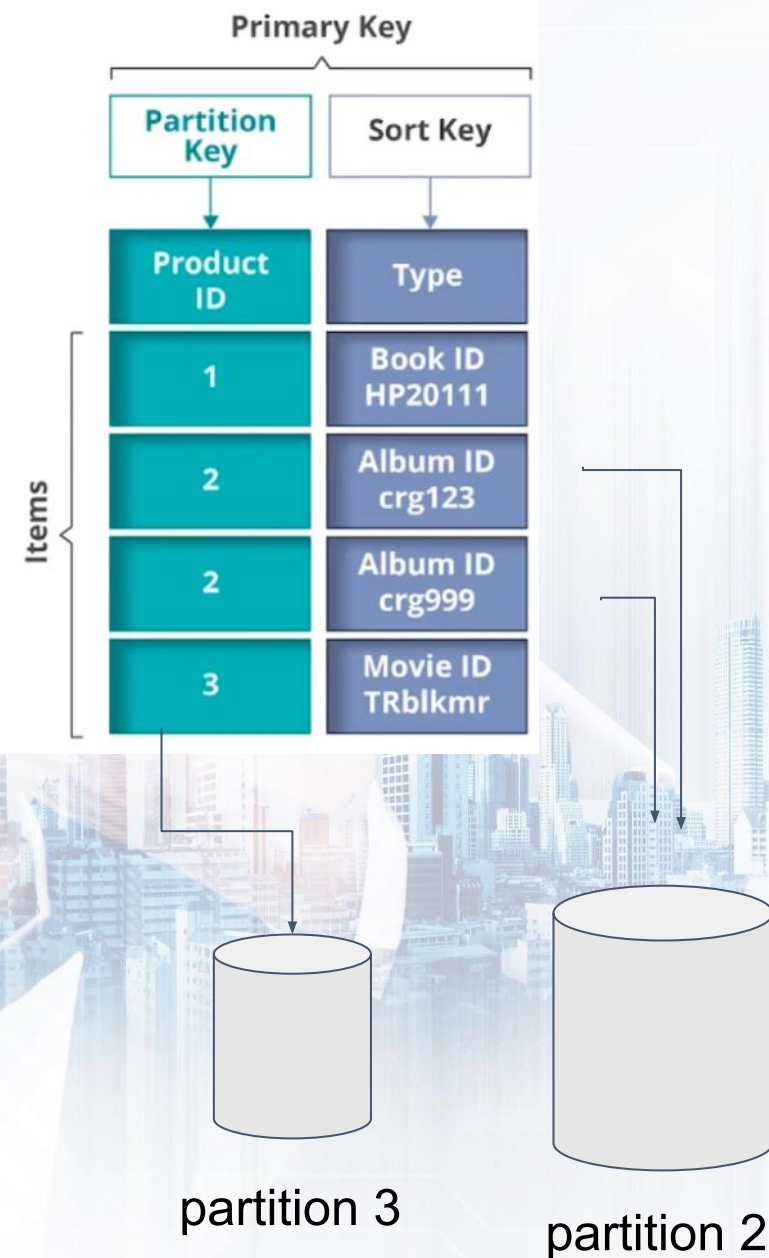
- DynamoDB có 2 thành phần là **Primary Key** và **Attributes**
- **Primary Key** được sử dụng làm khóa chính của bảng. Khi thao tác, chúng ta bắt buộc phải cung cấp thông tin này.
- **Attributes**: Là các thuộc tính của dữ liệu trong bảng. Có thể sử dụng Attributes để tìm kiếm, nhưng sẽ lâu hơn.



DynamoDB

Primary Key

- **Primary Key** có thể là **Partition Key** hoặc tổ hợp **Partition Key + Sort Key**.
- **Partition Key**: Là thông tin của một hash function nội bộ, giúp DynamoDB xác định dữ liệu đang nằm trong Partition nào (trong kho lưu trữ nội bộ DynamoDB) để truy xuất.
- **Sort Key**: Trong trường hợp Primary Key là tổ hợp, các Partition Key có thể giống nhau, nhưng Sort Key sẽ khác nhau, nhằm xác định vị trí lưu trữ trên Partition



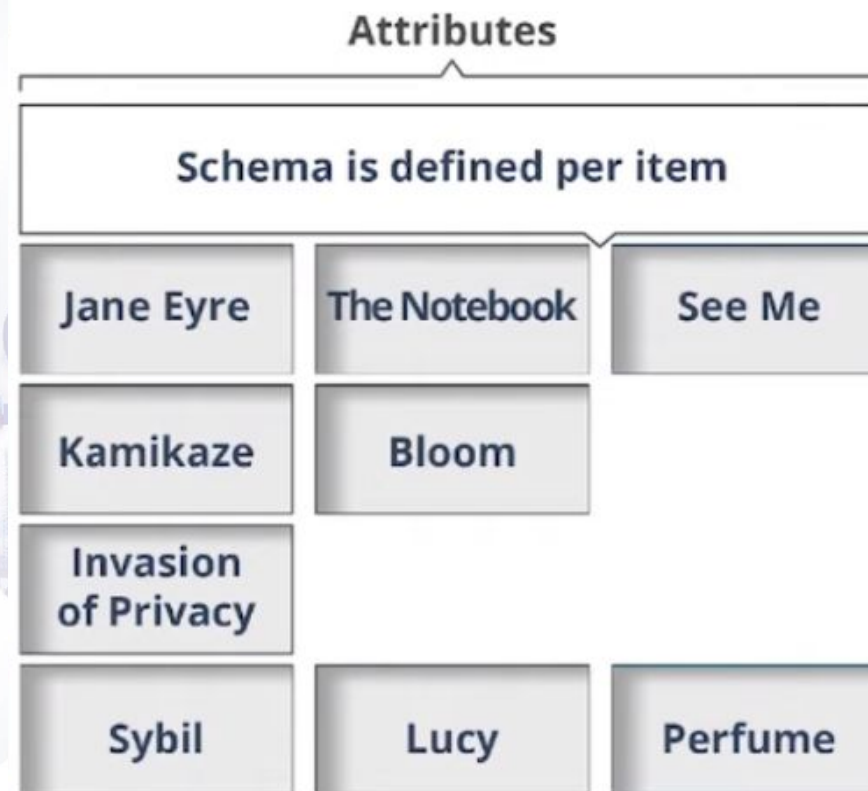
DynamoDB

Attributes

Attributes là các thông tin của một Items trong DynamoDB tables.

Các thông tin này có thể khác nhau, không cần theo một Schema nào.

Các attribute trong data thuộc loại Scalar, do vậy chỉ có một số kiểu như Boolean, Number, Strings, không các kiểu data gộp như Datetime, ...



DynamoDB

RCU/WCU

Capacity Unit là đơn vị đo mức độ sử dụng tài nguyên (RCU là đọc, WCU là ghi) của hệ thống đối với các tương tác tới DynamoDB table.

Capacity
đặt trước

Capacity
sử dụng

Mức vượt



DynamoDB

Pricing

On Demand: Dùng cho trường hợp lượng sử dụng ít, tuy nhiên thì thoảng có thể tăng cao, nhưng tốc độ vượt không quá nhanh.

Provision: Dùng cho trường hợp khi có nhu cầu tăng cao đột ngột, vẫn có thể đáp ứng.

Capacity
đặt trước

Capacity
sử dụng

Mức vượt





THANK YOU