

# AWS Cloud for beginner

Instructor: Linh Nguyen

(Engineering Consultant, AWS Cloud Solution Architect)

Level: Beginner

*“Không có việc gì khó, chỉ sợ không biết làm”*

# Relational Database Service (RDS)

Copyright@Linh Nguyen on Udemy

# Target

- Hiểu được RDS là gì, đặc trưng, usecase của RDS
- Biết cách tạo một RDS Cluster có chức năng HA, Read-replica
- Thực hành kết nối với RDS và thực hiện các thao tác cơ bản.
- Biết cách sử dụng parameter group để custom setting ở cấp độ database
- Tìm hiểu về Aurora – công nghệ Database độc quyền của AWS

Copyright@Linh Nguyen on Udemy

# What is RDS

Viết tắt của **Relational Database Service**

Là một service giúp người dùng tạo và quản lý các Relational Database.



**Amazon Relational  
Database Service (RDS)**

Copyright@Linh Nguyen on Udemy

# Đặc trưng cơ bản của RDS

- Là một Database as a Service. User không cần quan tâm tới hạ tầng ở bên dưới.
- Cho phép người dùng tạo ra các database instance độc lập hoặc cụm instance hoạt động dưới mode Cluster.
- Không thể login vào instance level (khác với việc tự cài DB lên 1 EC2 instance).
- Có thể scale theo 2 hình thức
  - Scale vertical: tăng hoặc giảm cấu hình instance.
  - Scale horizontal: thêm hoặc bớt node tuy nhiên node này chỉ có thể read (read-replica).
- Có giới hạn về dung lượng ổ cứng tối đa (64TB đối với MySQL, Maria,... 16TB đối với Microsoft SQL).

# Features of RDS





## **RDS cung cấp các tính năng cơ bản sau:**

- Cho phép tạo các DB Instance hoặc cụm cluster một cách nhanh chóng.
- Tự động fail-over giữa master-slave instance khi có sự cố.
- High-Availability: tự động cấu hình instance stand by, người dùng chỉ cần chọn.
- Tự động scale dung lượng lưu trữ (optional)
- Liên kết với CloudWatch để giám sát dễ dàng.
- Automate backup & manage retention.
- Dễ dàng chỉnh sửa setting ở cấp độ Database sử dụng parameter group.

# RDS supported engines

RDS Hỗ trợ các engines sau:

**Engine options**

<input type="radio"/> Amazon Aurora <b>Amazon Aurora</b>	<input type="radio"/> MySQL 	<input type="radio"/> MariaDB 
<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle <b>ORACLE</b>	<input checked="" type="radio"/> Microsoft SQL Server  <b>Microsoft SQL Server</b>

\*Aurora hiện tại chỉ hỗ trợ 2 engine là Aurora MySQL và Aurora PostgreSQL

# RDS Usecase

RDS được sử dụng trong hầu hết các trường hợp cần database dạng quan hệ. Vd: lưu trữ thông tin của user, website e-commerce, education,...

RDS thích hợp cho các bài toán OLAP (Online Analytical Processing) nhờ khả năng truy vấn mạnh mẽ, cấu hình có thể scale theo yêu cầu.

Copyright@Linh Nguyen on Udemy



# RDS Pricing

Về cơ bản RDS tính tiền dựa trên các thông số

- Instance size. Instance size càng lớn cost càng cao. Có hỗ trợ reserve instance tương tự EC2.
- Lượng data lưu trữ (GB/month)
- Dung lượng các bản snapshot được tạo ra.
- Các tính năng khác vd Backtracking đối với Aurora.

Copyright@Linh Nguyen on Udemy

# Mô hình triển khai RDS

**RDS Có thể được triển khai theo một số mô hình sau**

- Single Instance
- Single Instance with Multi-AZ option = yes
- Master – Read Only cluster
- Master – Read Only cluster with Multi-AZ option = yes
- Master – Multi Read cluster

Copyright@Linh Nguyen on Udemy

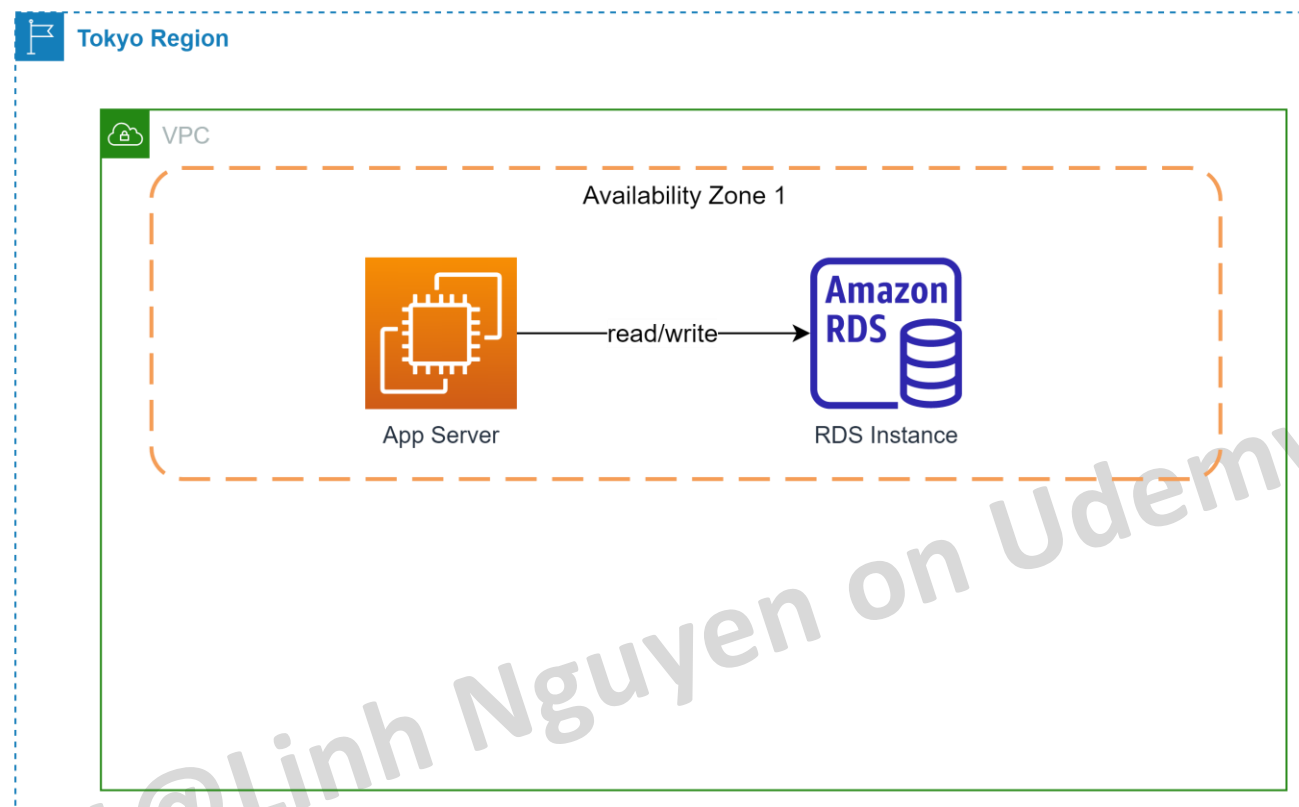
# Mô hình triển khai RDS

## Single Instance

Chỉ có 1 database instance duy nhất được tạo ra trên 1 Availability Zone (AZ)

Nếu sự cố xảy ra ở cấp độ AZ, database không thể truy cập.

Phù hợp cho môi trường Dev-Test để tiết kiệm chi phí

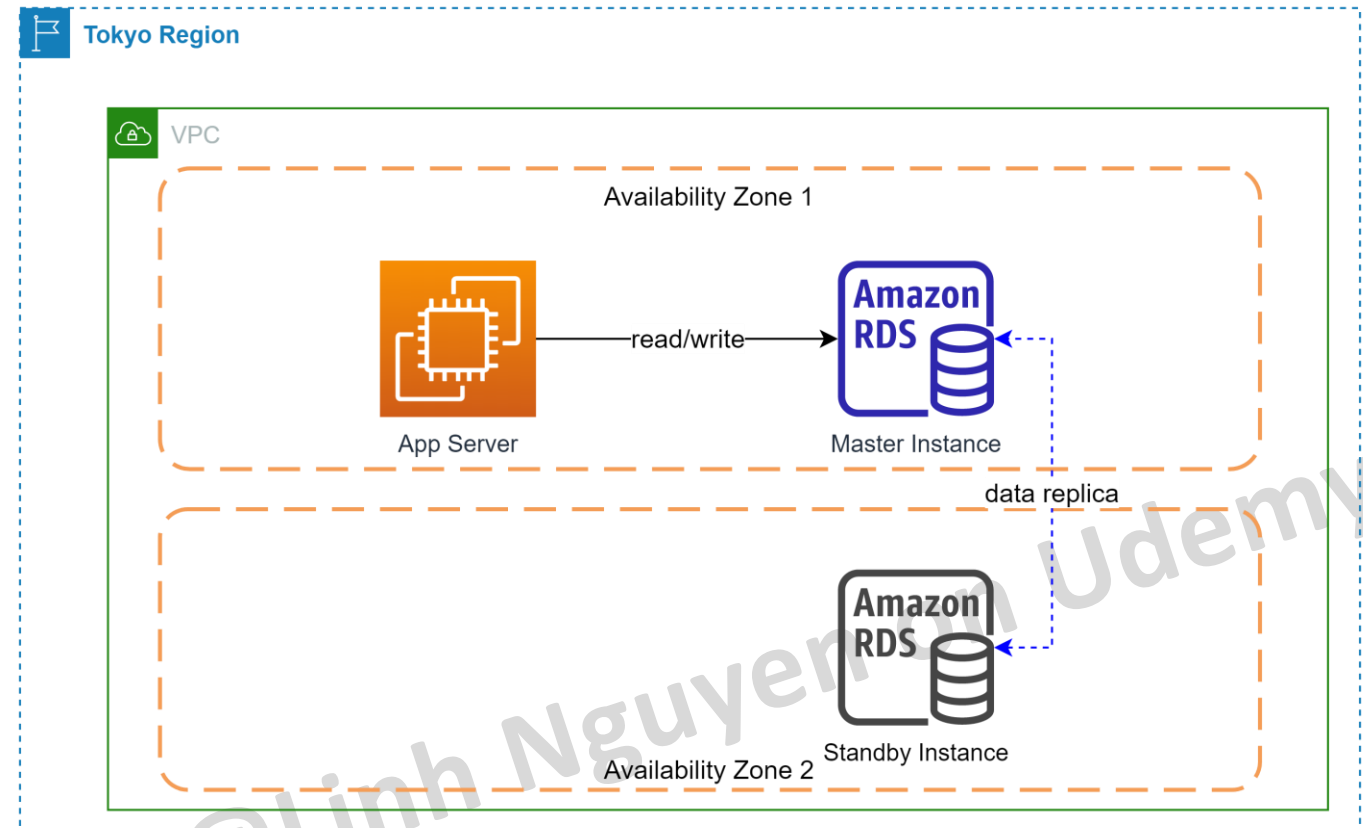


# Mô hình triển khai RDS

## Single Instance with Multi-AZ option enabled

Một bản sao của instance sẽ được tạo ra ở AZ khác và hoạt động dưới mode standby.

Nhiệm vụ của instance standby này là sync data từ master, không thể truy cập instance này.



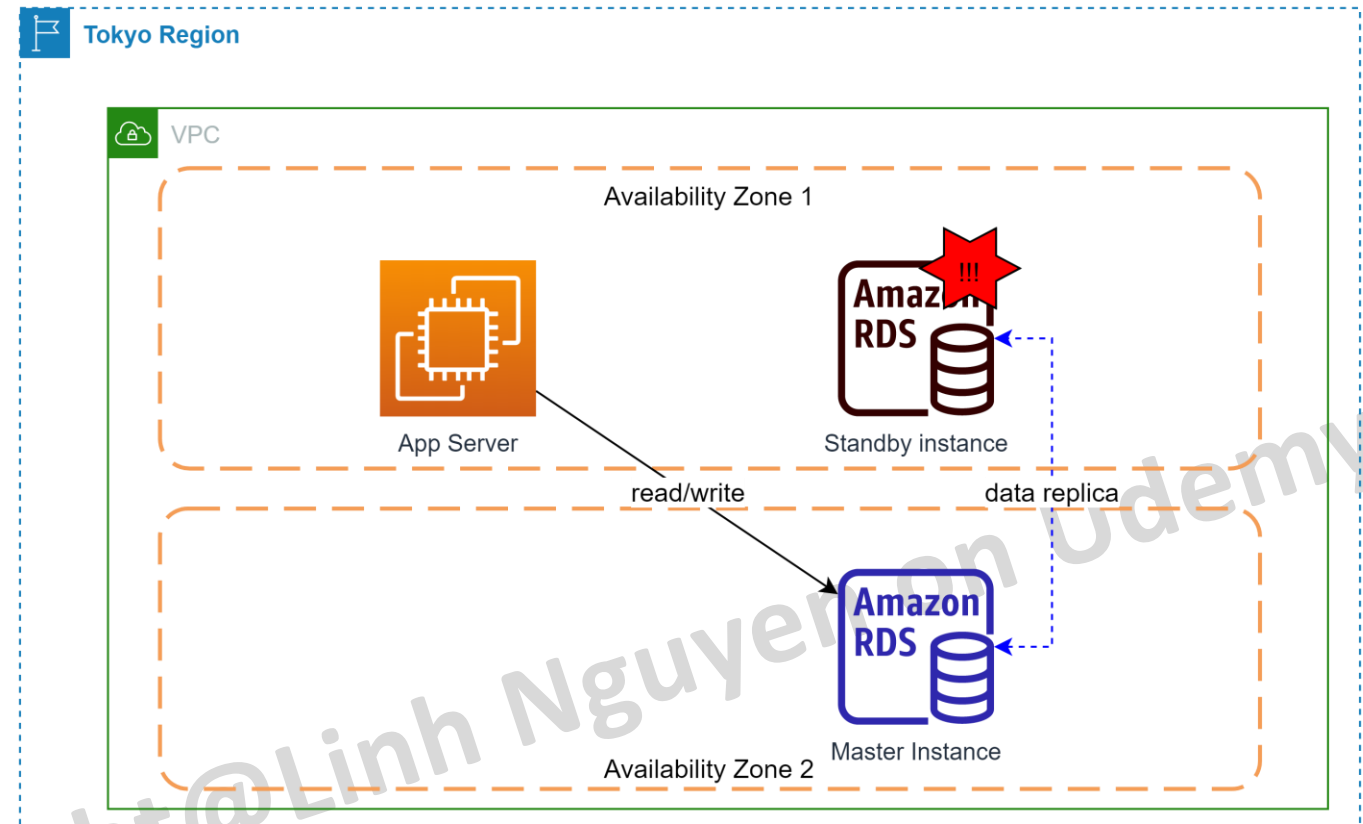
# Mô hình triển khai RDS

## Single Instance with Multi-AZ option enabled

Khi có sự cố, instance standby sẽ được chuyển lên thành master (việc này được AWS thực hiện tự động, endpoint url được giữ nguyên).

Nếu enable multi AZ, số tiền bỏ ra sẽ x2.

Phù hợp cho Database Production.



\*Instance standby được AWS tạo ra tuy nhiên user không thể kết nối tới instance này.

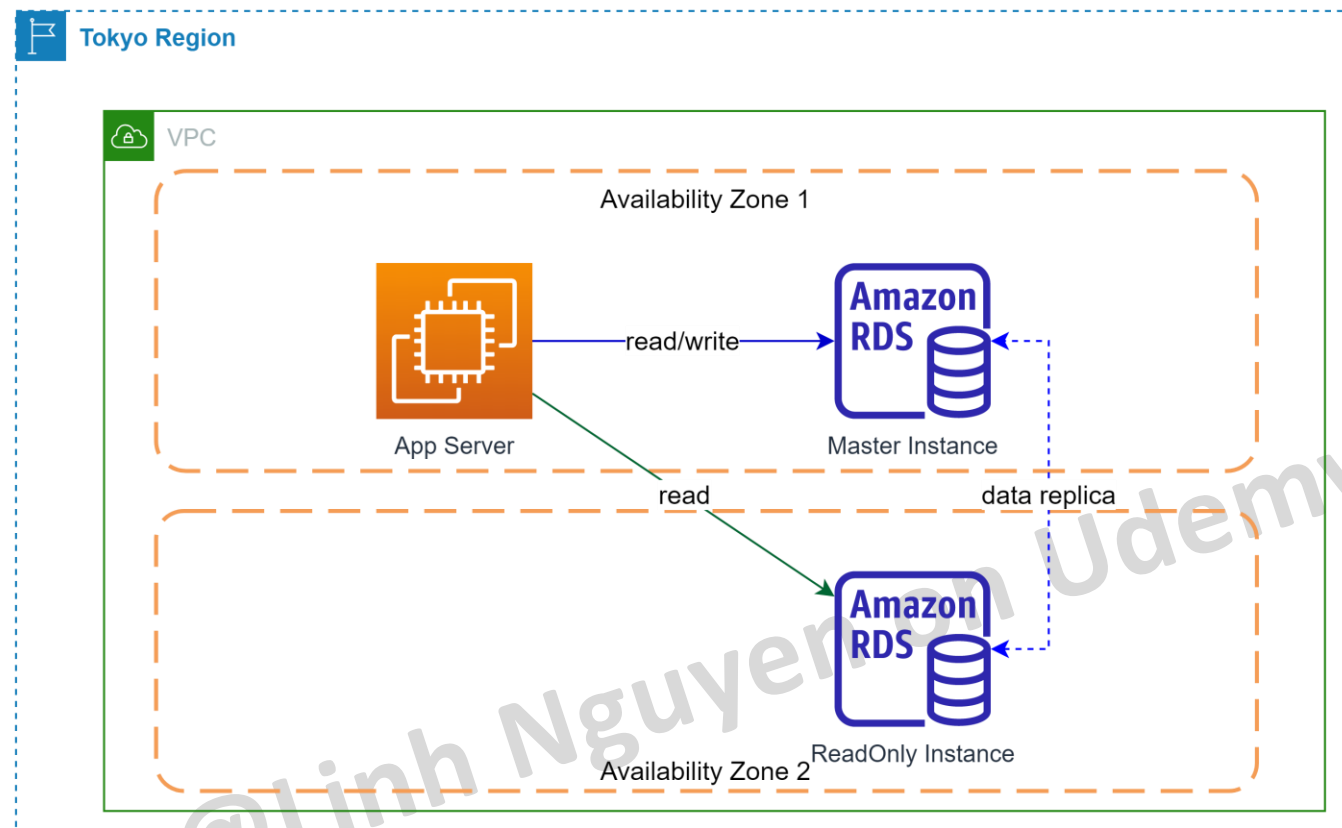
# Mô hình triển khai RDS

## Master – Read Only cluster

Một instance với mode ReadOnly sẽ được tạo ra và liên tục replica data từ master instance.

Instance này chỉ có thể đọc data. Phù hợp cho hệ thống có workload read>write, muốn tối ưu performance của Database.

Sau khi thiết lập quan hệ, instance được tạo ra sẽ kết hợp thành 1 cluster.



\*Instance read-only không nhất thiết phải khác AZ với Master.

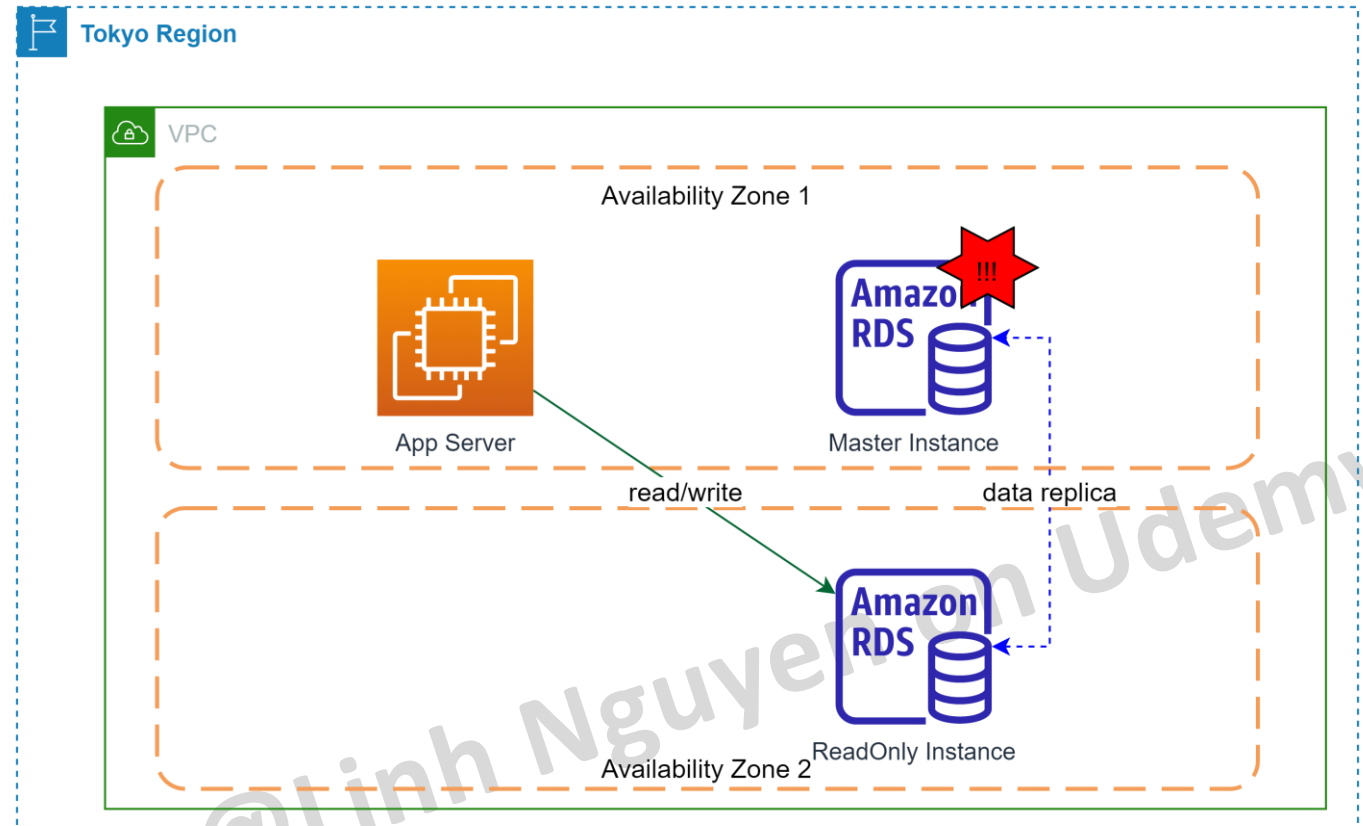
# Mô hình triển khai RDS

## Master – Read Only cluster

Trong trạng thái 2 instance đã hình thành cluster, nếu Master instance gặp sự cố, failover sẽ được tự động thực hiện, ReadOnly instance được promote lên làm Master.

\*Lưu ý: Nếu 2 instance được tạo ra riêng biệt sau đó mới thiết lập quan hệ read-replica, endpoint của 2 instance sẽ riêng biệt nên sau khi failover, cần chỉnh lại connection từ App.

\*Nên tạo cluster sau đó mới add node read vào để quản lý connection ở cluster level (số lượng node read có thể tùy chọn)

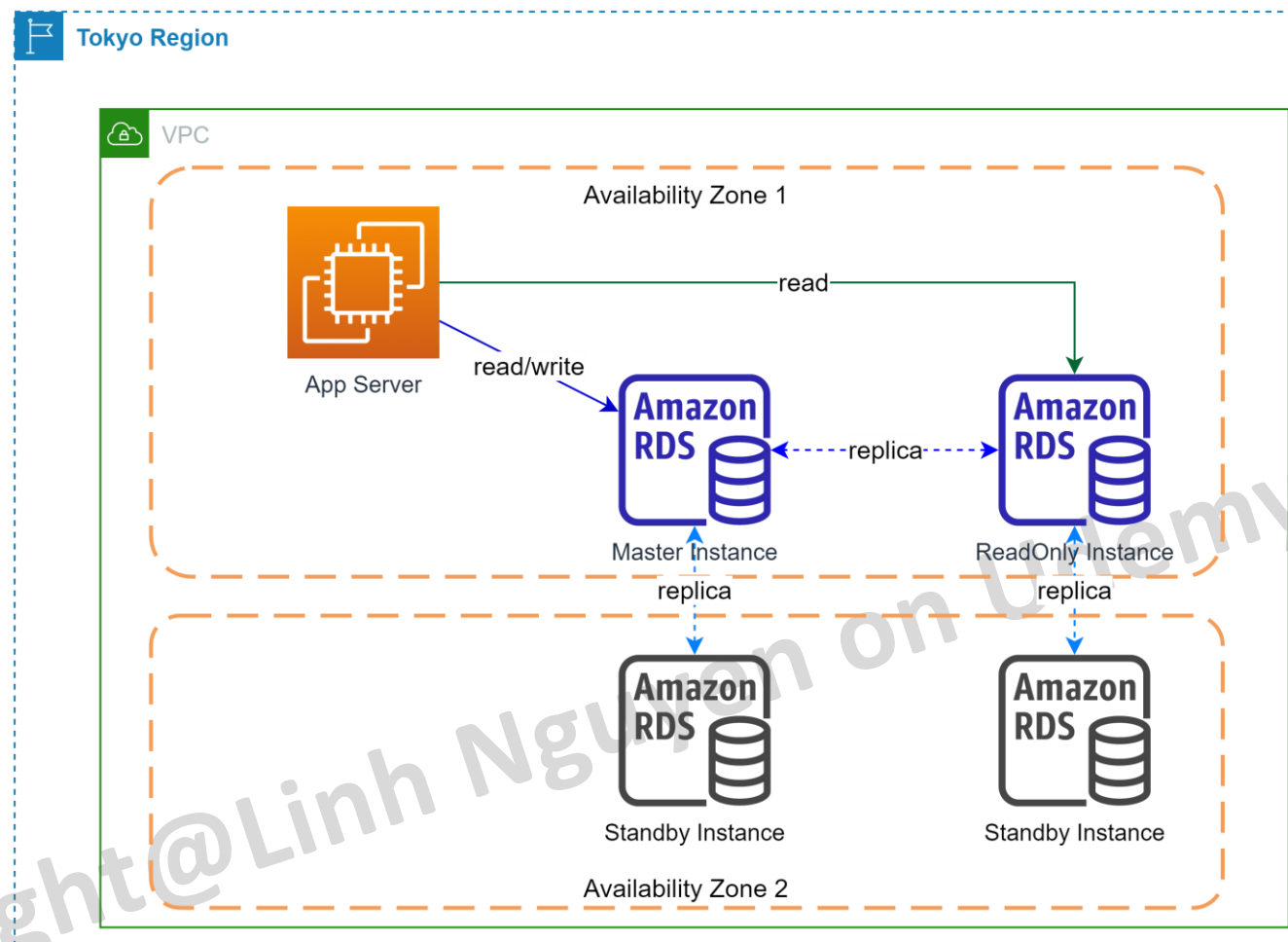


# Mô hình triển khai RDS

## Master – Read Only cluster with Multi-AZ option = yes

Tương tự mô hình Master – Read Only tuy nhiên các node đều được bật multi-AZ enabled.

Chi phí sẽ gấp 4 lần mô hình Single Instance.

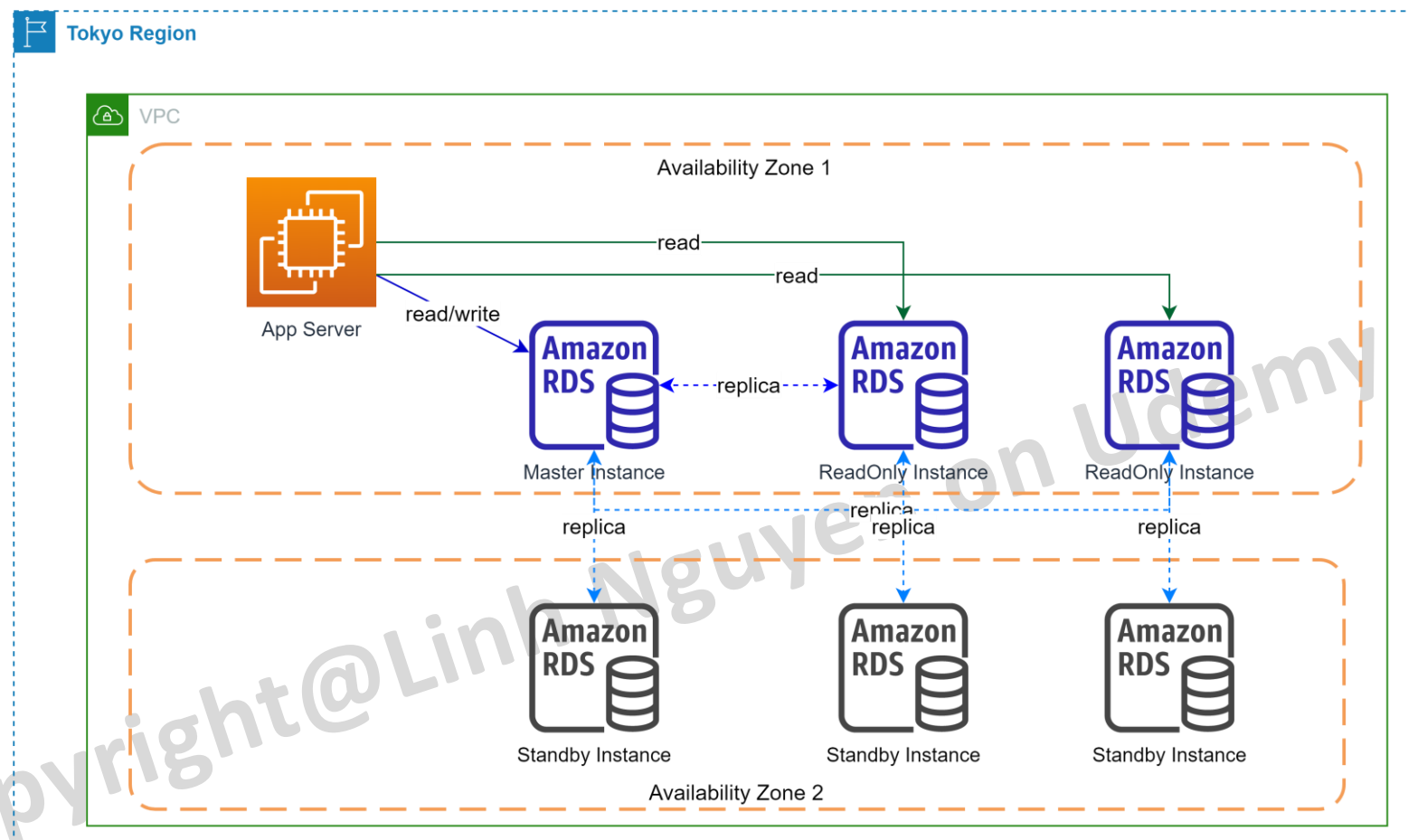




# Mô hình triển khai RDS

## Master – Multi Read cluster

Với mô hình này, nhiều hơn 1 reader instance sẽ được tạo ra.



# Mô hình triển khai RDS

## Nên tạo RDS Cluster hay RDS Instance?

AWS cung cấp cơ chế cho phép tạo ra 1 cluster RDS giúp quản lý node và failover dễ dàng hơn.

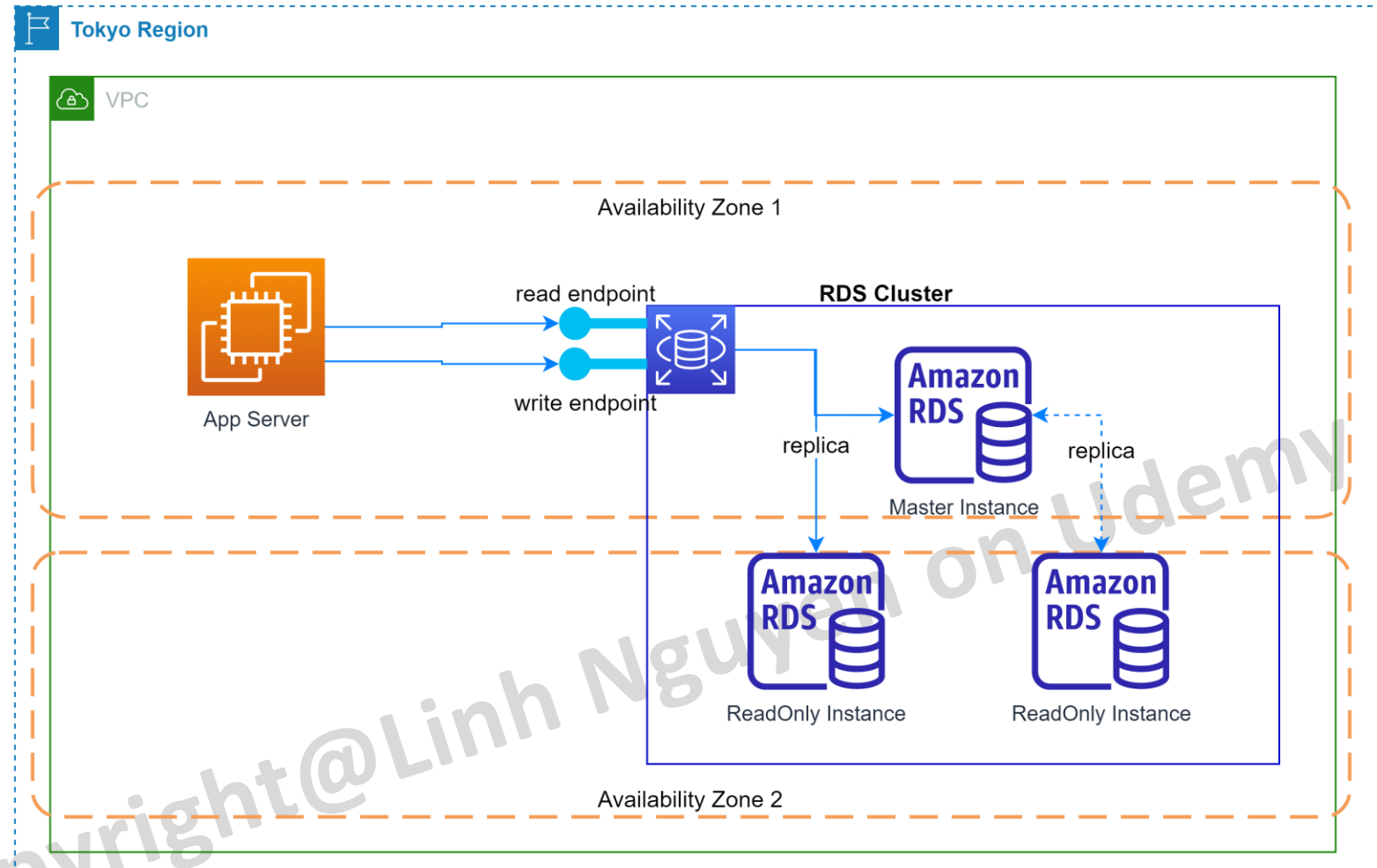
Ưu điểm so với việc tạo RDS instance thông thường:

- Quản lý endpoint ở cấp độ cluster, không bị thay đổi khi instance trong cluster gặp sự cố.
- Failover tự động.
- Scale read instance dễ dàng.

Copyright@Linh Nguyen on Udemy

# Mô hình triển khai RDS

Nên tạo RDS Cluster hay  
RDS Instance?



# Lab 1 – RDS Basic Operation

## \*Yêu cầu cài sẵn MySQL Workbench

1. Login to AWS console, navigate to RDS
2. Tạo 1 RDS instance với cấu hình cơ bản db.t3.medium, engine MySQL
3. Kết nối tới instance sử dụng MySQL Workbench
4. Thực hiện 1 số câu lệnh CRUD cơ bản.
5. Tạo thêm instance với role Read Replica.
6. Kết nối tới Read instance sử dụng MySQL Workbench
7. Thực hiện 1 số câu lệnh CRUD cơ bản => Read only.
8. Test delete Master Instance.

Copyright@Linh Nguyen on Udemy

# Lab 1 – RDS Basic Operation

NOTE: Clear resource sau bài lab!!!

Copyright@Linh Nguyen on Udemy

## Lab 2 – RDS Cluster

1. Login to AWS console, navigate to RDS
2. Tạo 1 RDS Cluster.
3. Kết nối với Cluster thông qua Cluster Write Endpoint.
4. Thực hiện 1 số câu lệnh CRUD cơ bản.
5. Kết nối với Cluster thông qua Cluster Read Endpoint.
6. Thực hiện 1 số câu lệnh CRUD cơ bản.

Copyright@Linh Nguyen on Udemy

# Lab 2 – RDS Cluster

NOTE: Clear resource sau bài lab!!!

Copyright@Linh Nguyen on Udemy

# Aurora

- Aurora là công nghệ database độc quyền của AWS, hỗ trợ 2 loại engine là MySQL và PostgreSQL.
- Aurora có 2 hình thức triển khai:
  - Cluster (Master + Multi Read Replica)
  - Serverless
- Những tính năng vượt trội của Aurora so với RDS thông thường
  - **Hiệu năng tốt hơn** (so với RDS instance cùng cấu hình. \*Cái này là AWS quảng cáo 😊)
  - **Hỗ trợ backtracking** (1 tính năng cho phép revert database về trạng thái trong quá khứ tối đa 72h). Khác với restore từ snapshot đòi hỏi tạo instance mới, backtracking restore ngay trên chính cluster đang chạy.
  - Tự động quản lý Write endpoint , Read endpoint ở cấp độ cluster.



# Mô hình cluster của Aurora

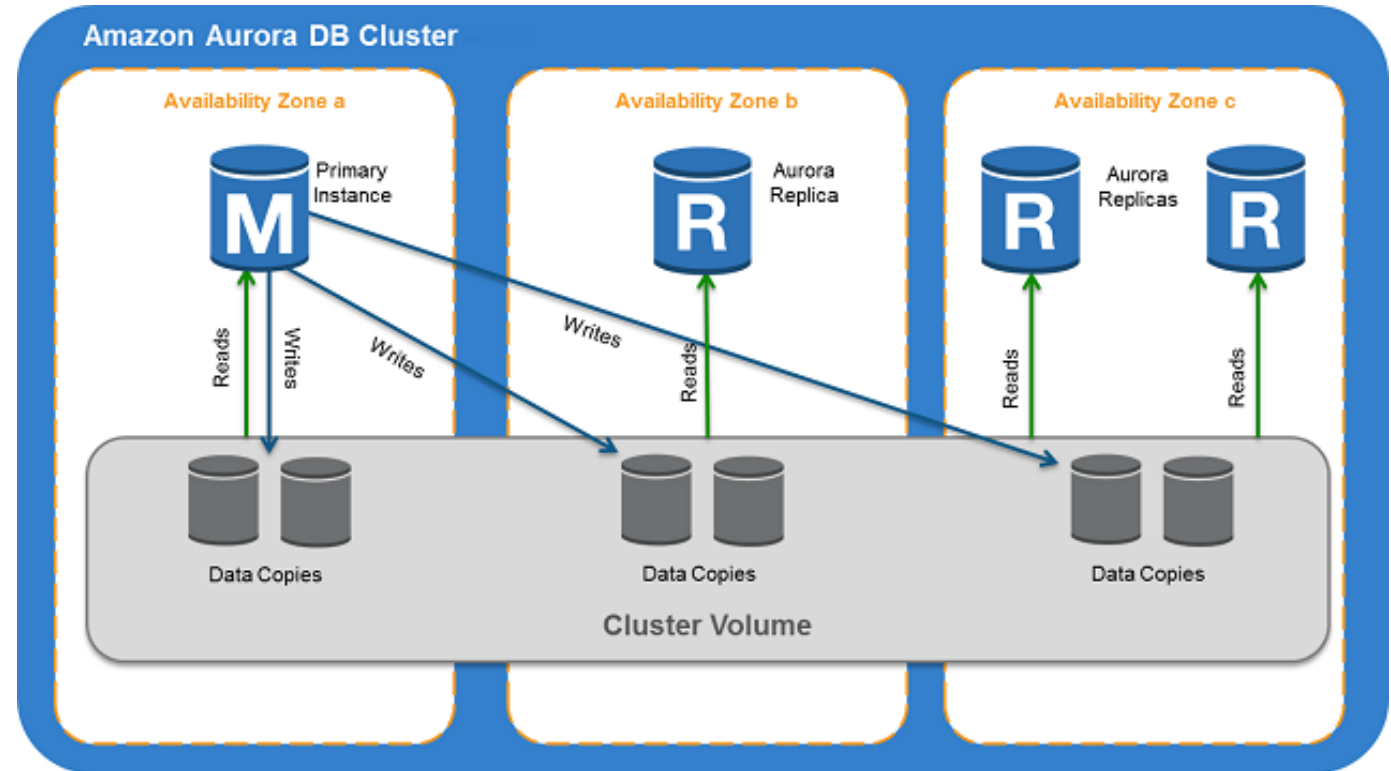
Cluster Aurora bao gồm:

- **1 Master node** (Primary instance)
- 1 hoặc nhiều **Replica node** (tối đa 15)

Data của Aurora cluster được lưu trên một storage layer gọi là Cluster Volume span trên nhiều AZ để tăng HA.

Data được Cluster Volume tự động replicate trên nhiều AZ. \*Số lượng copy không phụ thuộc vào số lượng instance của cluster.

Cluster Volume tự động tăng size dựa vào nhu cầu của người dùng (không thể fix cứng size).



Nguồn: doc.aws.amazon.com

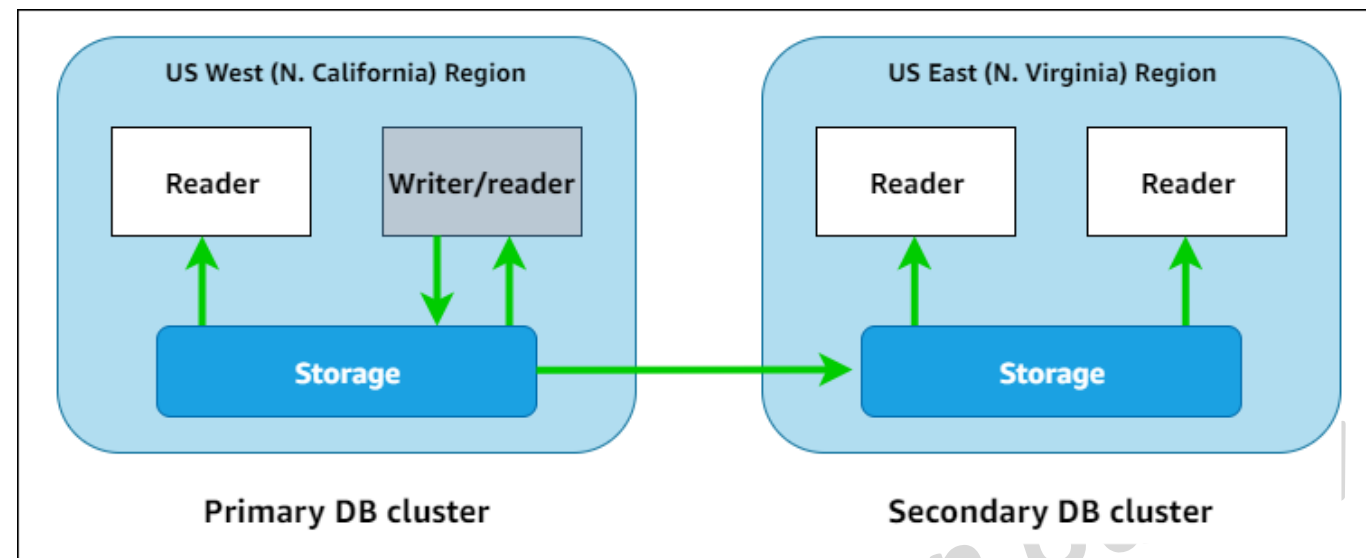
# Mô hình cluster của Aurora

## Aurora Global Cluster

Là một cơ chế cho phép tạo ra cụm cluster cross trên nhiều regions.

- Tăng tốc độ read tại mỗi region tương đương với local read.
- Mở rộng khả năng scale số lượng node read (limit 15 read nodes cho 1 cluster).
- Failover, Disaster recovery: rút ngắn RTO và giảm thiểu RPO khi xảy ra sự cố ở cấp độ region.

Mặc định cluster ở region thứ 2 trở đi chỉ có thể read, tuy nhiên có thể enable **write forwarding** để điều hướng request tới primary cluster.



Nguồn: doc.aws.amazon.com

\*Khác với RDS thông thường, Aurora sử dụng tầng Storage để replicate data giữa master-read nodes.

# Mô hình Serverless của Aurora

- Aurora Serverless là 1 công nghệ cho phép tạo Database dưới dạng serverless.
- Thay vì điều chỉnh cấu hình của DB instance, người dùng sẽ điều chỉnh ACU (Aurora Capacity Unit), ACU càng cao hiệu suất DB càng mạnh.
- Phù hợp cho các hệ thống chưa biết rõ workload, hoặc workload có đặc trưng thay đổi lên xuống thường xuyên.

Copyright@Linh Nguyen on Udemy

## Lab 3 – Aurora

1. Tạo Aurora Cluster với 2 instance (một Write, một Read-replica).
2. Kết nối đến Cluster sử dụng Write endpoint.
3. Thực hiện một số thao tác đơn giản (CRUD).
4. Enable tính năng back tracking
5. Đợi khoảng 5 phút.
6. Xoá toàn bộ data.
7. Thực hiện back tracking về 5 phút trước đó.
8. Kiểm tra data đã được phục hồi lại trạng thái trước khi bị xoá.

Copyright@Linh Nguyen on Udemy

# Parameter Groups

- RDS là một managed service do đó không thể login vào instance. Nếu muốn can thiệp vào setting ở cấp độ DB (không phải setting OS) ta cần thông qua 1 cơ chế gián tiếp là Parameter Groups.
- Khi tạo RDS nếu không chỉ định gì AWS sẽ sử dụng Parameter Group default của hệ database đang chọn. **Default Parameter Group** không thể chỉnh sửa.
- Custom Parameter Group được tạo ra bằng cách copy default Parameter Group sau đó chỉnh sửa những tham số phù hợp với nhu cầu.
- Parameter Groups có 2 loại là **Cluster Parameter Groups** và **Instance Parameter Group**. Hai loại này khác nhau về scope có thể apply.

# Parameter Groups

Một số Parameter thường được custom riêng theo nhu cầu hệ thống

Tên Parameter	Ý Nghĩa	Default value	Allowed value
max_connections	Số lượng connection tối đa mà Database có thể tiếp nhận đồng thời.	Tuỳ theo size của instance	50-12000
long_query_time	Query chạy vượt quá số này sẽ được tính là slow query.	0	0-31536000
slow_query_log	Có log ra những câu query bị coi là slow query hay không?	0	0, 1
max_allowed_packet	Package tối đa có thể gửi nhận tới database	1073741824	1024-1073741824
general_log	Có log ra general query hay không?	0	0, 1
time_zone	Setting time zone cho database.	Same as region	Ex: Asia/Tokyo

# Option Groups

- Tương tự Parameter Group tuy nhiên Option Group định nghĩa những setting liên quan tới Security là chủ yếu
- Một số ví dụ về option group:
  - SERVER\_AUDIT\_EVENTS: Loại action sẽ được log ra, vd: CONNECT, QUERY, QUERY\_DDL, QUERY\_DML, QUER
  - SERVER\_AUDIT\_INCL\_USERS: Include users vào audit log.
  - SERVER\_AUDIT\_EXCL\_USERS: Loại trừ users khỏi audit log (vd system user).
  - SERVER\_AUDIT\_LOGGING: Bật tắt audit logging.

Copyright@Linh Nguyen on Udemy

# Lab 4 – RDS Parameter Groups

1. Tạo 1 Parameter Group từ default MySQL 5.x
2. Setting các thông số như bảng sau
3. Apply Parameter Group đó cho 1 cluster.

Parameter name	Value
max_connections	100
max_allowed_packet	1000000000 (~100MB)
long_query_time	0.5
slow_query_log	1

Copyright@Linh

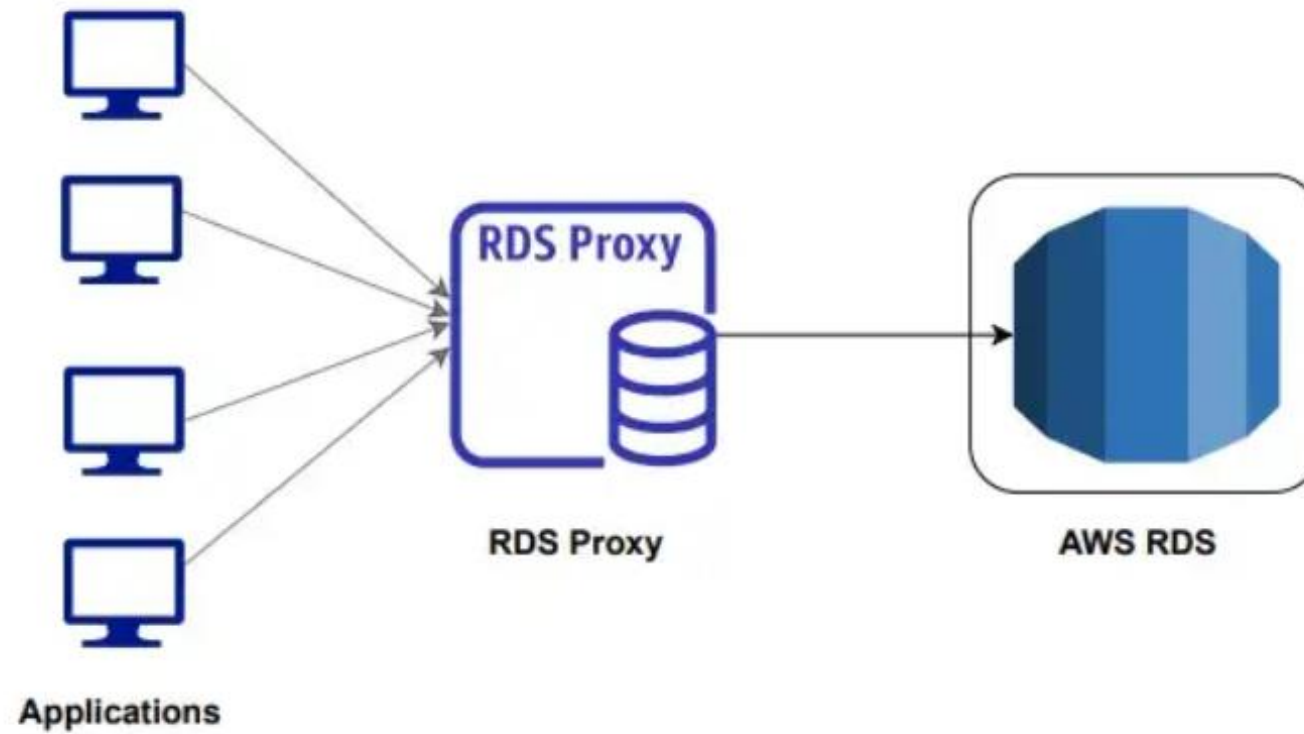


# RDS Proxy

- RDS cung cấp cơ chế proxy giúp quản lý connection tới các instance một cách hiệu quả, hạn chế bottle neck (VD: do application quản lý connection không tốt gây ra).
- Khi sử dụng proxy, application sẽ không kết nối trực tiếp tới RDS mà thông qua proxy endpoint.
- Chi phí sẽ phát sinh thêm cho proxy.
- Hiện tại hỗ trợ 3 engine: MySQL, PostgreSQL, SQL Server

Copyright@Linh Nguyen on Udemy

# RDS Proxy



Copyright@Linh Nguyen

# Clear resources

Login vào AWS Console và thực hiện những nội dung sau

1. Xoá RDS Instance
2. Xoá RDS Cluster
3. Xoá Aurora Cluster
4. Xoá các bản snapshot còn sót lại
5. Terminate EC2 instance nếu có

Copyright@Linh Nguyen on Udemy