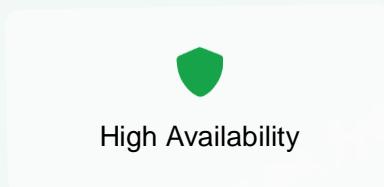
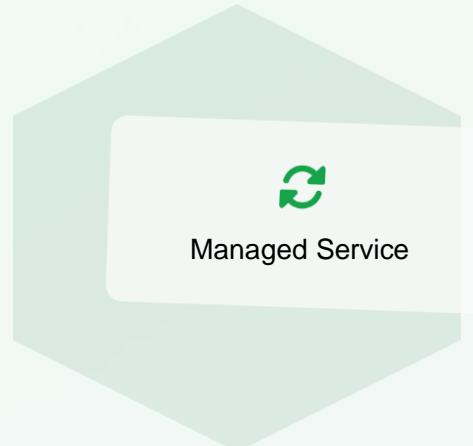




Amazon RDS & Các Dịch vụ Cơ sở Dữ liệu Liên Quan

Dịch vụ cơ sở dữ liệu quan hệ được AWS quản lý hoàn toàn



Ưu điểm khi dùng RDS so với tự triển khai DB trên EC2



RDS là dịch vụ được quản lý

AWS quản lý toàn bộ infrastructure dưới tầng RDS, bao gồm cả việc cấu hình, triển khai và bảo trì.



Tự triển khai trên EC2

Yêu cầu quản trị thủ công, cấu hình phức tạp và bảo trì thường xuyên.



Provisioning tự động & OS patching

AWS tự động quản lý infrastructure, bao gồm provisionning server và áp dụng patch bảo mật.



Backup liên tục & Point in Time Restore

Tự động backup và khôi phục tại bất kỳ thời điểm nào trong khoảng thời gian lưu trữ.



Monitoring dashboards tích hợp sẵn

Theo dõi hiệu năng và trạng thái DB thông qua Amazon CloudWatch.



Read Replicas

Cần cấu hình manual, replication sync/async cần thiết lập.



Multi-AZ & HA

Cần cấu hình manual, failover thủ công khi có sự cố.



Scaling

Cần manual scaling, downtime cho việc thay đổi instance type.

RDS là managed service → giảm thiểu quản trị, tăng hiệu năng

RDS – Storage Auto Scaling



Tự động mở rộng lưu trữ

Amazon RDS tự động điều chỉnh dung lượng lưu trữ dựa trên nhu cầu thực tế, tránh việc phải quản trị thủ công.

Điều kiện kích hoạt



- Dung lượng trống < 10% tổng dung lượng
- Trạng thái thiếu dung lượng kéo dài ≥ 5 phút
- 6 giờ kể từ lần thay đổi gần nhất



Ngưỡng tối đa

Người dùng cần đặt Maximum Storage Threshold để giới hạn tăng trưởng tối đa của lưu trữ.

Quy trình tự động mở rộng



Kiểm tra dung
lượng



Phát hiện thiếu
dung lượng



Tự động mở
rộng

- ✓ Hỗ trợ ứng dụng có nhu cầu lưu trữ biến động
- ✓ Hạn chế over-provisioning

- ✓ Không cần can thiệp thủ công
- ✓ Hỗ trợ tất cả các database engines của RDS

Storage Auto Scaling giúp tối ưu hóa chi phí và hiệu năng

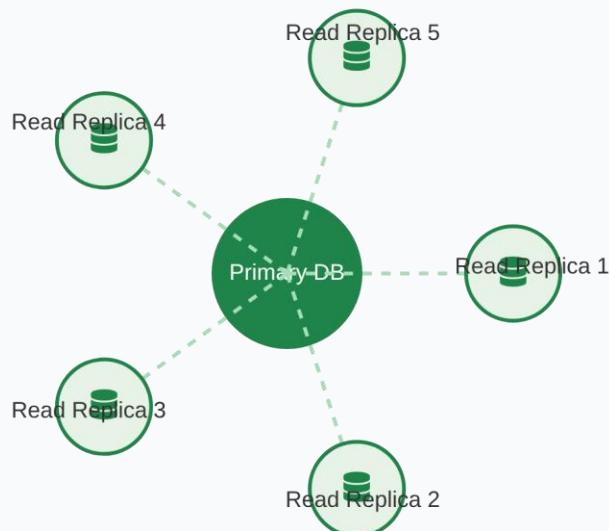
RDS Read Replicas – Read Scalability



Read Replicas

Các Read Replica cho phép bạn hướng dẫn lưu lượng đọc đến các bản sao riêng biệt của instance cơ sở dữ liệu chính.

Primary DB → Read Replicas: ASYNC replication



Max 15 Read Replicas per source DB



Số lượng Read Replicas

Có thể tạo tối đa 15 Read Replicas từ một instance cơ sở dữ liệu nguồn.



Triển khai

Có thể triển khai Within AZ, Cross AZ hoặc Cross Region.



Replication bất đồng bộ

Replication được thực hiện bất đồng bộ → dữ liệu là eventually consistent.



Promotion

Read Replica có thể được promote thành DB độc lập nếu cần thiết.

RDS Read Replicas – Use Cases

Read Replicas giúp cải thiện hiệu năng đọc bằng cách phân tán tải cho các truy vấn đọc đến các bản sao riêng biệt của instance chính.



Báo cáo & Analytics

Read Replicas xử lý các workload báo cáo và phân tích nặng nề, giúp giảm tải cho DB chính mà không ảnh hưởng đến hiệu năng.

- ✓ Tốt cho các ứng dụng reporting



Tìm kiếm & Filter

Các truy vấn tìm kiếm và filter phức tạp có thể được hướng đến Read Replicas, cải thiện hiệu năng của các thao tác đọc.

- ✓ Thích hợp cho search functionality



Ứng dụng đa vùng

Read Replicas có thể được triển khai ở nhiều vùng AWS khác nhau, giúp giảm latency cho người dùng phân tán địa lý.

- ✓ Tăng tốc độ truy vấn toàn cầu



Development & Testing

Môi trường phát triển và testing có thể sử dụng Read Replicas để truy vấn dữ liệu mà không ảnh hưởng đến DB chính.

- ✓ An toàn cho môi trường dev

! Lưu ý quan trọng

Read Replicas chỉ hỗ trợ các thao tác đọc (SELECT). Các thao tác ghi (INSERT, UPDATE, DELETE) phải được thực hiện trên DB chính.

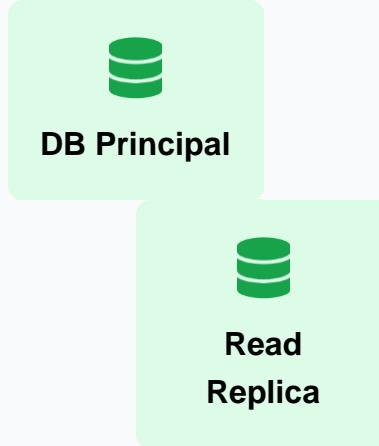
RDS Read Replicas – Chi Phí Mạng



Lưu ý quan trọng: Trong AWS, có chi phí mạng khi dữ liệu truyền giữa các AZ khác nhau, nhưng **Read Replicas** trong cùng Region sẽ không tính phí.



Replica trong cùng Region



\$ Không tính phí mạng

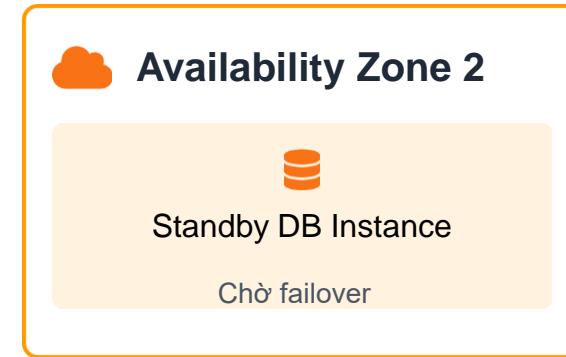
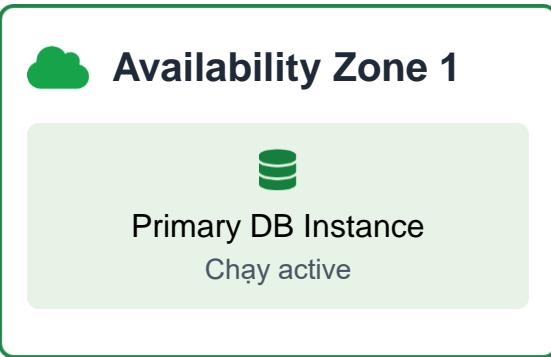


Replica giữa các Region khác nhau



\$ Tính phí mạng

RDS Multi-AZ (Disaster Recovery)



Replication đồng bộ

Instance chính được replication đồng bộ sang instance dự phòng trong AZ khác.

Mỗi transaction phải được xác nhận trên cả hai instance trước khi được chấp nhận.



Handle tự động

Khi instance chính gặp sự cố (panne AZ, mất mạng, hỏng instance hoặc storage), RDS tự động bascilenie sang instance dự phòng. Quá trình bascilenie thường mất từ 60-120 giây.



Lưu ý: Multi-AZ không được sử dụng để scale đọc, mà chỉ phục vụ Disaster Recovery. Read Replicas có thể được cấu hình Multi-AZ để tăng tính sẵn sàng.

RDS – Từ Single-AZ sang Multi-AZ



Quá trình chuyển đổi không downtime

Chuyển đổi từ Single-AZ sang Multi-AZ là quá trình được thiết kế để thực hiện mà không gây gián đoạn dịch vụ.



Bước 1: Tạo Snapshot

Tạo snapshot của instance DB hiện tại



Bước 2: Restore sang AZ mới

Khôi phục snapshot vào AZ mới để tạo instance secondary



Bước 3: Thiết lập Đồng bộ

Đồng bộ hóa giữa instance chính và instance secondary



Bước 4: Hoàn tất

Multi-AZ đã sẵn sàng, chờ failover nếu cần

Quá trình nội bộ

- ✓ Modification initiated via AWS console with a simple click
- ✓ Snapshot creation, restoration in new AZ, and synchronization

Lưu ý quan trọng

- ⚠ Process is designed for zero downtime
- ⚠ Improves database resilience without interrupting operations

Amazon Aurora Overview



Aurora là gì?

Công nghệ cơ sở dữ liệu relationnelle độc quyền của AWS, không open-source, được tối ưu specifically cho môi trường AWS Cloud.



Hỗ trợ MySQL & PostgreSQL

Tương thích với các database engines phổ biến, giúp dễ dàng di chuyển từ RDS sang Aurora.



Storage tự động tăng

Tự động tăng theo block 10GB, tối đa 128TB, tránh phải quản lý storage thủ công.



High Availability Native

Basculement quasi instantané, không cần can thiệp thủ công khi có sự cố.



So sánh hiệu năng

MySQL trên RDS

1x

1x

Aurora MySQL

5x

5x

PostgreSQL trên RDS

1x

1x

Aurora PostgreSQL

3x

3x



Replicas

Hỗ trợ đến 15 replicas với độ trễ dưới 10ms.



Chi phí

Đắt hơn RDS khoảng 20%, nhưng hiệu năng và tính sẵn sàng tốt hơn.

Aurora High Availability & Read Scaling



High Availability Architecture

- ✓ Lưu 6 bản sao dữ liệu trên 3 AZ khác nhau
- ✓ 4/6 bản sao cần cho write, 3/6 cho read
- ✓ Self-healing qua replication ngang hàng
- ✓ Storage striping trên hàng trăm volumes

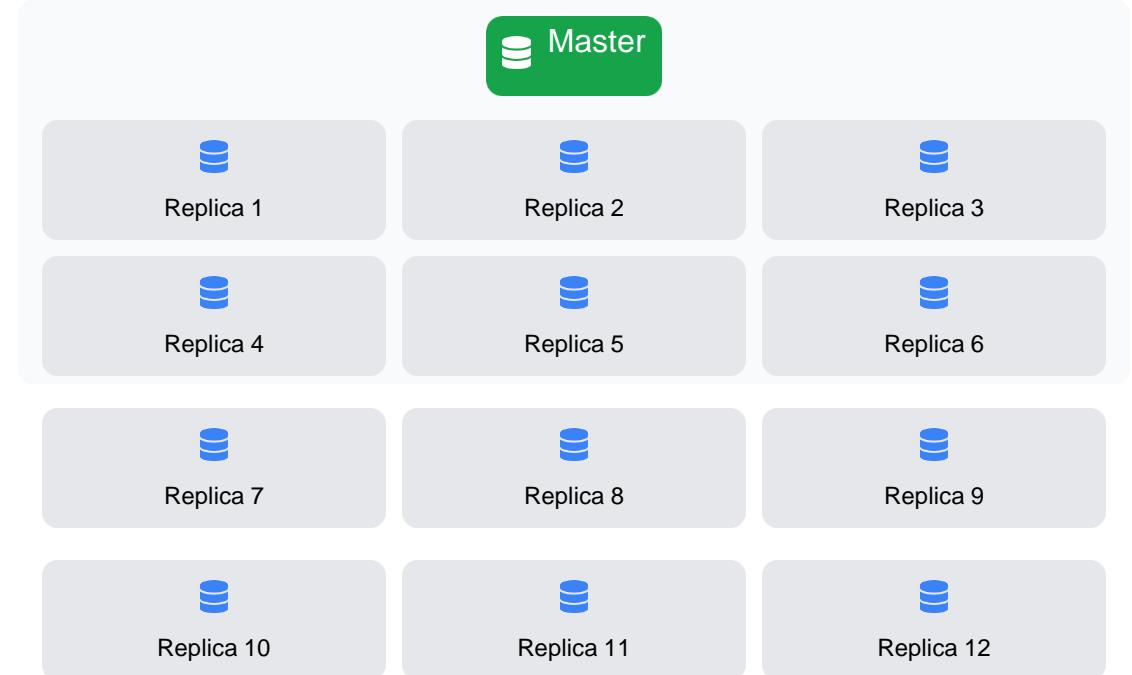


Read Scaling Architecture

- ✓ Tối đa 15 Read Replicas với độ trễ <10ms
- ✓ 1 Aurora Instance đảm nhiệm write (master)
- ✓ Failover tự động <30 giây khi master lỗi
- ✓ Hỗ trợ Cross-Region Replication



Aurora: High Availability & Read Scaling in the cloud



Aurora Features

Các tính năng nổi bật của Amazon Aurora



Automatic Failover

Tự động chuyển đổi trong vòng chưa đến 30 giây khi instance chính gặp sự cố.



Backup & Recovery

Sao lưu tự động và khôi phục theo thời điểm với khả năng lưu giữ dữ liệu lên đến 35 ngày.



Isolation & Security

Kiểm soát truy cập chi tiết, mã hóa dữ liệu và nhóm bảo mật để bảo vệ dữ liệu của bạn.



Industry Compliance

Tuân thủ các quy định của ngành nhằm đáp ứng các yêu cầu về bảo mật và quyền riêng tư.



Push-button Scaling

Mở rộng tài nguyên dễ dàng và nhanh chóng mà không có thời gian ngừng hoạt động hay gián đoạn dịch vụ.



Zero-downtime Patching

Triển khai các bản cập nhật và bản vá mà không làm gián đoạn dịch vụ, ngay cả đối với các bản nâng cấp lớn.



Backtrack

Cho phép khôi phục dữ liệu tại bất kỳ thời điểm nào mà không cần thực hiện khôi phục từ bản sao lưu truyền thống.

RDS & Aurora Security



Mã Hóa Dữ Liệu Khi Lưu Trữ

- ✓ Sử dụng AWS KMS để mã hóa dữ liệu.
- ✓ Phải được kích hoạt khi tạo instance.
- ✓ Các bản sao (replica) không thể được mã hóa nếu instance chính chưa được mã hóa
- ✓ Amazon RDS sử dụng thuật toán AES-256.



Mã hóa trong quá trình truyền

- ✓ Kết nối được mã hóa bằng SSL/TLS
- ✓ Dữ liệu được mã hóa tự động khi truyền giữa các vùng AWS.



Xác Thực Bằng IAM

- ✓ Sử dụng Role IAM để kết nối
- ✓ Kiểm soát truy cập chi tiết
- ✓ Quản lý danh tính tập trung



Security Groups & Audit Logs

- ✓ Security Groups kiểm soát lưu lượng vào/ra (inbound/outbound)
- ✓ **Audit Logs:** là nhật ký kiểm tra, ghi lại các hoạt động truy cập và thay đổi trong cơ sở dữ liệu để phục vụ giám sát, bảo mật và tuân thủ quy định.

La sécurité des bases de données RDS et Aurora est assurée par plusieurs couches de protection

Amazon RDS Proxy

Dịch vụ proxy database được AWS quản lý hoàn toàn, giúp tối ưu kết nối và tăng hiệu suất cho RDS.



Connection Pooling

Pool & share kết nối đến DB → giảm tải tài nguyên (CPU, RAM)

Tăng hiệu năng

Tối ưu hiệu suất DB, giảm timeout và connection overhead

Giảm failover time

Giảm thời gian failover đến 66% cho Aurora & RDS

Serverless

Auto-scaling, không cần provision infrastructure

Security

Tích hợp IAM Authentication & AWS Secrets Manager

Hỗ trợ nhiều DB

MySQL, PostgreSQL, MariaDB, SQL Server

Amazon ElastiCache Overview



ElastiCache là gì?

Dịch vụ cache in-memory được AWS quản lý hoàn toàn, designed để cấu hình, quản lý và mở rộng các môi trường cache phân tán trong cloud.



Ứng dụng

Truy vấn ElastiCache trước, sau đó RDS nếu cache miss



ElastiCache

Cache in-memory với độ trễ thấp



RDS

Cơ sở dữ liệu quan hệ gốc



Redis

Multi-AZ, Read Replicas, Data durability



Memcached

Multi-node partitioning, multi-threaded

Hiệu năng cao

Accès dữ liệu ở memory → độ trễ thấp ms

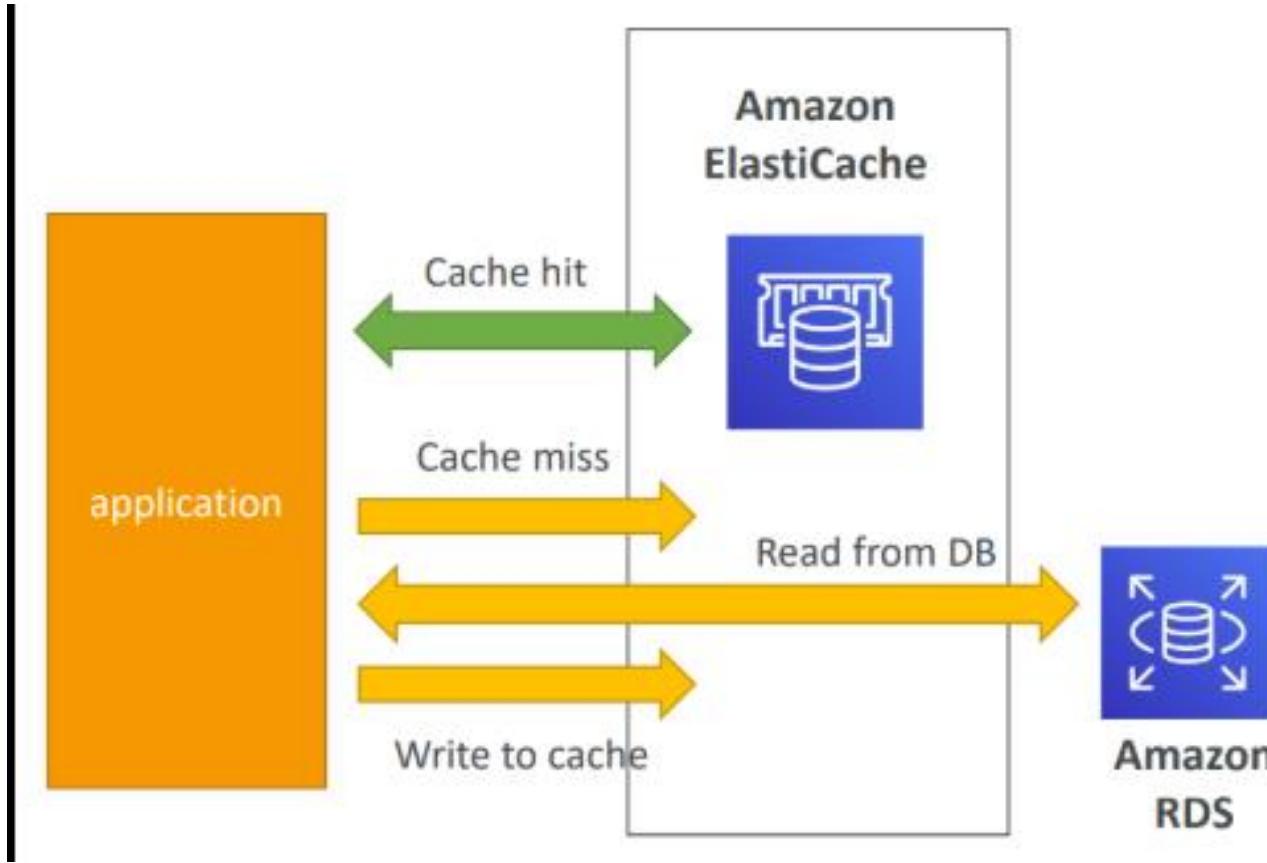
Scalability

Tự động scaling theo nhu cầu

AWS Management

AWS quản lý OS, maintenance, patching

ElastiCache Solution Architecture – DB Cache



Cách hoạt động

- ✓ Ứng dụng truy vấn ElastiCache trước
- ✓ Nếu có dữ liệu (cache hit) → trả về luôn
- ✓ Nếu không có (cache miss) → truy vấn RDS rồi lưu vào cache

Lợi ích

- ✓ Giảm tải cho RDS đáng kể
- ✓ Tăng tốc độ truy vấn dữ liệu
- ✓ Cần chiến lược invalidation cache để dữ liệu luôn mới nhất

ElastiCache Solution Architecture – User Session Store

Cách sử dụng ElastiCache để lưu trữ session người dùng, giúp ứng dụng trở nên stateless và dễ mở rộng.



✓ Quy trình lưu trữ session

- Người dùng đăng nhập vào ứng dụng
- Ứng dụng tạo session data
- Lưu session vào ElastiCache
- Ứng dụng trả về response cho người dùng

✓ Lợi ích của việc sử dụng ElastiCache cho session

- Giúp ứng dụng trở nên stateless
- Dễ dàng mở rộng ứng dụng
- Giảm tải cho cơ sở dữ liệu
- Độ trễ thấp cho việc truy cập session

ElastiCache cho session store là giải pháp hiệu quả cho ứng dụng web mac

ElastiCache – Redis vs Memcached



Redis



Multi-AZ với Auto-Failover

Tự động failover giữa các Availability Zones



Read Replicas

Hỗ trợ Read Replicas để scale đọc và tăng HA



Data Durability

AOF persistence để lưu trữ dữ liệu bền vững



Backup & Restore

Hỗ trợ backup và restore dữ liệu



Data Structures

Hỗ trợ Sets và Sorted Sets



Memcached



Multi-node Partitioning

Sharding tự nhiên giữa nhiều node



No High Availability

Không có mechanism replication hoặc failover



No Persistence

Dữ liệu chỉ lưu trong memory, không persistent



No Backup/Restore

Không có cơ chế backup hoặc restore



Multi-threaded

Architecture multi-threaded cho hiệu năng cao

Lựa chọn Redis khi cần data durability và advanced features; Memcached khi cần hiệu năng đơn giản và sharding

Cache Strategies & TTL

Chiến lược caching



Lazy Loading / Cache-Aside

- Lazy Loading: tải dữ liệu vào cache khi cần
- Cache-Aside: kiểm tra cache trước, nếu không có thì load từ DB
- Phù hợp cho hầu hết các use case đọc dữ liệu



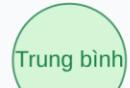
Write-through

- Viết dữ liệu vào cache và DB cùng lúc
- Đảm bảo consistency giữa cache và DB
- Thường dùng kết hợp với Lazy Loading cho workload đặc thù



Time-to-Live (TTL)

Ví dụ về TTL khác nhau



TTL Best Practices

- Đặt hợp lý, tránh cache lỗi thời
- Cache dữ liệu phù hợp: user profiles, blog content, metadata...
- Chỉ cache dữ liệu cần thiết, tránh lãng phí bộ nhớ

Cache Invalidation

- Cần cache invalidation strategy để đảm bảo dữ liệu mới nhất
- Invalidating khi dữ liệu được cập nhật để tránh inconsistency

Chọn chiến lược caching phù hợp với workload để tối ưu hiệu năng

Final Words of Wisdom



Lazy Loading

- ✓ Foundation tốt cho các tình huống đọc dữ liệu
- ✓ Easy implementation, phù hợp hầu hết use case



Write-through & TTL

- ✓ Write-through kết hợp với TTL cho workload tối ưu
- ✓ Balance giữa consistency và performance



Cache Selection

- ✓ Chỉ cache dữ liệu cần thiết, tránh lãng phí bộ nhớ
- ✓ user profiles, blog content, metadata là lựa chọn tốt



Key Takeaway

Cache là công cụ mạnh mẽ nhưng cần được sử dụng một cách thông minh. Chọn đúng chiến lược cache và dữ liệu phù hợp sẽ giúp ứng dụng của bạn trở nên nhanh chóng và hiệu năng cao hơn.

Cảm ơn bạn đã xem!