**Cấu hình Amazon ElastiCache for Redis để đạt độ sẵn sàng cao**

**Amazon ElastiCache** đã trở thành công nghệ đi đôi với các ứng dụng thời gian thực. Redis nổi bật với hiệu năng cao, tính đơn giản, và khả năng hỗ trợ nhiều cấu trúc dữ liệu khác nhau — tất cả đã khiến nó trở thành một trong những hệ thống **key-value store phi quan hệ** phổ biến nhất hiện nay.

Với sự phát triển của các ứng dụng thời gian thực mang tính chất **then chốt đối với doanh nghiệp**, việc **đảm bảo tính sẵn sàng (availability)** trở thành yếu tố cực kỳ quan trọng.

Để đáp ứng nhu cầu này, **Amazon ElastiCache for Redis hỗ trợ cấu hình Redis Cluster**, mang lại khả năng **mở rộng vượt trội và tính sẵn sàng cao**.  
Bên cạnh đó, **ElastiCache cung cấp hỗ trợ Multi-AZ (nhiều vùng khả dụng) kết hợp với cơ chế tự động chuyển đổi dự phòng (auto failover)** — cho phép bạn thiết lập một cụm Redis với **một hoặc nhiều bản sao (replica)** được phân phối trên các vùng khác nhau.  
Khi xảy ra lỗi ở **nút chính (primary)**, ElastiCache sẽ tự động **chuyển đổi (failover)** sang một **nút replica** để đảm bảo hệ thống tiếp tục hoạt động **không gián đoạn**.

Amazon ElastiCache for Redis cũng đã công bố nhiều tính năng mới, góp phần nâng cao **tính sẵn sàng tổng thể** cho các ứng dụng Redis của bạn.

**Tăng cường tính sẵn sàng của cụm trong quá trình bảo trì định kỳ**

Khả năng sẵn sàng của cụm Redis được cải thiện khi bật **tự động chuyển đổi dự phòng (auto-failover)**, đặc biệt trong các hoạt động như vá lỗi, cập nhật phần mềm và các tác vụ bảo trì khác liên quan đến việc thay thế node.

Đối với các cấu hình Redis Cluster sử dụng **Redis Cluster client**, việc bảo trì định kỳ và thay thế node **giờ đây có thể hoàn tất mà không gây gián đoạn việc ghi dữ liệu**.  
Còn với các cấu hình **không dùng Redis Cluster (non-sharded)**, bạn có thể thấy **một chút gián đoạn khi ghi dữ liệu**, thường chỉ kéo dài vài giây — nguyên nhân chủ yếu là do cập nhật DNS.

**Tự chủ trong cập nhật (Self-service updates)**

Tính năng **tự chủ trong cập nhật** cho phép bạn **kiểm soát thời điểm bắt đầu các bản cập nhật bảo trì**, từ đó **giảm thiểu tối đa ảnh hưởng đến hệ thống**.  
Bạn có thể tìm thêm thông tin và các câu hỏi thường gặp về lịch bảo trì tại **trang FAQ chuyên biệt** của dịch vụ.

**Endpoint đọc chung cho cấu hình không dùng Redis Cluster**

Với các cụm Redis **không sử dụng Cluster**, ElastiCache cung cấp một **endpoint đọc duy nhất**. Nhờ đó, bạn có thể **điều hướng lưu lượng đọc mà không cần theo dõi thủ công endpoint của từng replica**.

Điều này giúp **nâng cao tính sẵn sàng**, vì ứng dụng của bạn không cần xử lý các thay đổi endpoint khi replica thay đổi.  
Còn với Redis Cluster, tính năng này thường **đã được xử lý tự động bởi Redis Cluster client thông minh**.  
Ngoài ra, nhiều client Redis còn hỗ trợ tùy chọn READONLY — rất hữu ích nếu bạn muốn tận dụng **các replica trong cụm Redis để mở rộng khả năng xử lý truy vấn đọc (read scalability)** cho ứng dụng.

**Đổi tên lệnh Redis một cách linh hoạt**

Tính năng **đổi tên lệnh Redis (Dynamic rename)** cho phép bạn **đổi tên các lệnh Redis ngay khi hệ thống đang chạy**, **không cần khởi động lại hay gây ảnh hưởng đến tính sẵn sàng**.

Với Redis 6, Amazon ElastiCache for Redis còn hỗ trợ **kiểm soát truy cập theo vai trò (Role-Based Access Control)**, cho phép bạn:

* Giới hạn quyền truy cập vào các lệnh cụ thể,
* Và giới hạn quyền truy cập vào từng khóa dữ liệu (key) thông qua **access string** dựa trên tài khoản người dùng.

Để tận dụng tối đa những cải tiến này cũng như đảm bảo **tính sẵn sàng tổng thể**, bạn nên **xem xét lại cấu hình hiện tại** và đảm bảo rằng hệ thống đã được thiết lập đúng cách để mang lại **hiệu quả cao nhất về độ sẵn sàng**.

Các phần tiếp theo sẽ hướng dẫn bạn **những phương pháp cấu hình tốt nhất (best practices)** dành cho:

* **Cụm ElastiCache for Redis**,
* **Redis client**,
* Cũng như các **mẹo chung cho ứng dụng** nhằm tăng tính ổn định và độ sẵn sàng của hệ thống.

**Cấu hình Amazon ElastiCache for Redis**

Bạn có thể thiết lập Amazon ElastiCache for Redis bằng cách chọn đúng loại node, cấu hình Redis phù hợp (Redis Cluster hoặc không), số lượng bản sao (replica), cùng các tính năng bổ sung tùy chọn.

**Bước đầu tiên** là **xem xét lại cấu hình cụm ElastiCache for Redis** của bạn:

🔹 **Bật Multi-AZ với tính năng tự động chuyển đổi dự phòng (auto failover)**  
Việc bật Multi-AZ giúp **giảm thiểu thời gian gián đoạn** bằng cách **tự động chuyển vai trò từ node chính sang các replica**, nếu xảy ra sự cố — dù là có kế hoạch hay đột xuất.

🔗 Xem thêm: [Tự động chuyển đổi Multi-AZ](https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/AutoFailover.html)

🔹 **Tối thiểu 3 shard cho Redis Cluster**  
Thiết lập **tối thiểu 3 shard** giúp cải thiện đáng kể **tính sẵn sàng**, đặc biệt trong các tình huống failover, nhờ khả năng phục hồi nhanh hơn.  
Amazon ElastiCache for Redis hỗ trợ lên đến **500 node trong một cụm** (bao gồm cả shard và replica).

🔹 **Thiết lập 2 hoặc nhiều replica phân tán trên các Availability Zone**  
Có từ **hai replica trở lên** giúp:

* **Tăng khả năng mở rộng xử lý truy vấn đọc (read scalability)**,
* **Duy trì tính sẵn sàng cho truy vấn đọc**, ngay cả khi một replica đang bảo trì.

Điều này đặc biệt quan trọng nếu bạn **không sử dụng endpoint đọc chung** và chỉ định client truy cập trực tiếp từng replica.

🔹 **Sử dụng phiên bản Redis và loại node mới nhất**  
Các instance thế hệ mới như **R6g và M6g** sử dụng kiến trúc **Graviton2**, tận dụng nền tảng **Nitro** để mang lại hiệu suất gần như không khác biệt so với máy vật lý (bare metal), và khả năng xử lý mạng vượt trội.

Bên cạnh việc tối ưu hiệu suất và chi phí, **Redis phiên bản 6** còn mang đến nhiều cải tiến về:

* replication (sao chép),
* snapshot (sao lưu nhanh),
* eviction (thu hồi dữ liệu),
* và độ trễ (latency).

🔹 **Theo dõi và điều chỉnh tài nguyên theo lưu lượng truy cập dự kiến**  
Khi tải quá cao, Redis có thể mất phản hồi, gây ảnh hưởng đến tính sẵn sàng. Hãy theo dõi các chỉ số:

* **DatabaseMemoryUsagePercentage**: thể hiện dung lượng bộ nhớ đang sử dụng trên mỗi node,
* **ReplicationLag**: cho biết tình trạng sao chép dữ liệu, đặc biệt khi tốc độ ghi tăng cao.

Bạn có thể **thiết lập cảnh báo hoặc auto scaling cụm** dựa trên các chỉ số này.

🔗 Tham khảo thêm: [Blog về giám sát Redis workloads](https://aws.amazon.com/blogs/database/monitoring-your-amazon-elasticache-for-redis-workloads/)

🔹 **Tránh bảo trì và nâng cấp vào giờ cao điểm**  
Thực hiện bảo trì khi hệ thống ít ghi dữ liệu giúp quá trình **failover nhanh hơn và ít ảnh hưởng đến ứng dụng** hơn.

🔹 **Sử dụng Amazon ElastiCache for Redis Global Datastore**  
Tính năng **Global Datastore** cho phép bạn:

* **Sao chép dữ liệu cụm Redis từ vùng chính (Primary Region) sang vùng phụ (Secondary Region)**,
* Hỗ trợ khôi phục khi thảm họa (disaster recovery),
* Làm điểm chuyển đổi dự phòng nếu xảy ra sự cố trên toàn vùng.

**Cấu hình Redis Client**

Redis cung cấp một **hệ sinh thái client phong phú**, giúp bạn linh hoạt lựa chọn công cụ phù hợp với nhu cầu và ngôn ngữ lập trình của mình. Dưới đây là một số hướng dẫn chung, có thể áp dụng cho hầu hết các Redis client:

🔹 **Redis Cluster Mode:**  
Nếu bạn sử dụng Redis ở chế độ Cluster (phân mảnh dữ liệu theo slot), hãy:

* **Sử dụng Redis client có hỗ trợ Redis Cluster** (Cluster-aware client).
* **Kết nối đến cụm Redis thông qua endpoint cấu hình (configuration endpoint)** thay vì endpoint của từng shard.

Điều này cho phép client:

* **Tự động phát hiện các shard và phân bổ slot dữ liệu**,
* Hỗ trợ **resharding online** (tăng/giảm số shard mà không cần dừng hệ thống),
* Và **thực hiện bảo trì hay thay thế node mà không gây gián đoạn khi ghi dữ liệu**.

Ngoài ra, bạn nên:

* **Cập nhật bản đồ cluster (cluster map) cục bộ trong Redis client một cách định kỳ**, đặc biệt khi gặp lỗi **MOVED** – đây là lỗi cho biết key đang được truy cập không nằm ở shard hiện tại.

👉 Tìm hiểu thêm về cách làm việc với Redis Cluster trong Amazon ElastiCache tại blog hướng dẫn chuyên biệt:  
[**Redis Cluster with Amazon ElastiCache**](https://aws.amazon.com/blogs/database/running-redis-cluster-on-amazon-elasticache/)

A computer screen shot of a computer program

AI-generated content may be incorrect.

**🔹Non- Redis Cluster Mode:**

Với các cụm Redis **không sử dụng chế độ Cluster**, bạn nên:

* **Dùng endpoint chính (primary endpoint) cho toàn bộ lưu lượng ghi (write traffic)**  
  Trong trường hợp có thay đổi cấu hình hoặc xảy ra failover, Amazon ElastiCache sẽ **tự động cập nhật DNS của endpoint chính** để luôn trỏ đến node chính hiện tại. Điều này đảm bảo các lệnh ghi luôn được xử lý đúng.
* **Dùng reader endpoint để điều hướng toàn bộ lưu lượng đọc (read traffic)**  
  Amazon ElastiCache sẽ **tự động cập nhật DNS của reader endpoint theo thời gian thực**, đảm bảo nó luôn phản ánh chính xác trạng thái cụm – đặc biệt khi có replica được thêm hoặc xóa.
* **Endpoint của từng node (individual node endpoint)** vẫn có thể được sử dụng, nhưng nếu dùng **reader endpoint**, bạn **không cần theo dõi thủ công từng thay đổi ở từng node**, giúp đơn giản hóa ứng dụng.

Reader endpoint hoạt động bằng cách **cung cấp một bản ghi DNS**, sẽ **trả về IP của một trong các replica theo cơ chế vòng tròn (round robin)**. Vì vậy, cách tốt nhất là:

* **Dùng primary endpoint để ghi**,
* Và **dùng reader endpoint duy nhất để đọc**.

A computer screen shot of a computer

AI-generated content may be incorrect.

🔹 **Thời gian chờ socket (Socket timeout):**  
Hãy đảm bảo **thời gian chờ (timeout) của socket trên client được đặt tối thiểu là 1 giây**, thay vì để mặc định là “không giới hạn” như nhiều client hiện nay.

* Nếu đặt timeout quá thấp → rất dễ xảy ra lỗi timeout khi server đang tải cao.
* Nếu đặt quá cao → ứng dụng sẽ mất nhiều thời gian để nhận biết sự cố kết nối.

🔹 **Tái sử dụng kết nối (Connection pooling):**  
**Bật tính năng connection pooling** để client có thể **tái sử dụng kết nối đã mở**, thay vì tạo mới mỗi lần gửi lệnh.

* Việc này giúp **giảm chi phí thiết lập kết nối**,
* Tránh tình trạng **hết giới hạn kết nối trên mỗi node**,
* Đồng thời **giảm độ trễ và tăng hiệu năng tổng thể** khi giao tiếp với Amazon ElastiCache for Redis.

🔹 **Bộ nhớ đệm DNS (DNS caching):**  
Nếu Redis client của bạn có cơ chế **cache DNS nội bộ**, hãy:

* **Đặt TTL ở mức thấp (5–10 giây)** để đảm bảo client luôn cập nhật địa chỉ node mới nhất khi xảy ra failover hoặc thay đổi topology.
* **Tránh sử dụng tùy chọn “cache mãi mãi (cache forever)”**, vì điều này khiến ứng dụng có thể truy cập sai node hoặc bị lỗi kết nối.

🔹 **Kiểm tra chuyển đổi dự phòng (Test failover):**  
Khi một node chính (primary) trong cụm Redis gặp sự cố, **Amazon ElastiCache sẽ tự động chuyển vai trò chính sang một replica hiện có**.

Bạn có thể **chủ động kiểm tra tình huống này** bằng cách dùng **API TestFailover**. Vì mỗi Redis client có cơ chế xử lý failover khác nhau, bạn cần:

* **Hiểu rõ hành vi của client trong sự kiện failover**,
* **Kiểm tra khả năng tự phục hồi của client và logic quản lý kết nối** trước khi sự cố thực tế xảy ra trong môi trường production.

**Ứng Dụng Thực Tiễn**

Ngoài việc cấu hình **cụm ElastiCache for Redis** và **Redis client**, bạn cũng nên xem lại **logic của ứng dụng** để áp dụng các phương pháp tối ưu và tăng cường độ sẵn sàng như sau:

🔹 **Tránh sử dụng các đoạn mã LUA chạy lâu**  
Các script LUA dài có thể khiến Redis bị **treo hoặc không phản hồi**, làm giảm tính sẵn sàng của hệ thống.  
Nếu bạn buộc phải dùng LUA, hãy **đảm bảo hệ thống đủ tài nguyên (CPU)** để xử lý các đợt tăng tải.

🔹 **Ưu tiên cơ chế hết hạn (expiration) hơn là thu hồi dữ liệu (eviction)**  
Chính sách thu hồi có thể gây tốn tài nguyên hơn so với việc thiết lập thời gian hết hạn cho key.  
→ Để **giảm áp lực bộ nhớ**, hãy áp dụng thời gian **hết hạn tự động (TTL)** cho các key thay vì phụ thuộc vào eviction.

🔹 **Tránh sử dụng các lệnh tốn kém (expensive commands)**  
Một số lệnh như KEYS có thể gây ảnh hưởng lớn đến hiệu năng và **cản trở các hoạt động được quản lý** (như bảo trì, cập nhật).

* Thay vào đó, bạn nên sử dụng SCAN, vì SCAN có độ phức tạp thời gian **ổn định** (constant time), trong khi KEYS là **tuyến tính** (linear).
* Tương tự, nếu bạn sử dụng **Sorted Set hoặc Hash có kích thước lớn**, việc đồng bộ dữ liệu sẽ gặp khó khăn và ảnh hưởng đến các hoạt động như bảo trì và nâng cấp.

📌 Để **tránh việc dùng nhầm các lệnh nặng**, bạn có thể:

* **Bật tính năng kiểm soát truy cập theo vai trò (RBAC)**: ElastiCache sẽ **áp chính sách kiểm soát truy cập** dựa trên người dùng, giới hạn quyền sử dụng các lệnh nhất định.
* Hoặc **đổi tên các lệnh Redis một cách linh hoạt (dynamic rename)** để hạn chế truy cập theo cách của riêng bạn.

**Tóm tắt**

Chúng tôi rất vui được mang đến cho bạn những **cải tiến về tính sẵn sàng và các khuyến nghị thực tiễn** trong bài viết này — và đây mới chỉ là **bước khởi đầu**.

Đội ngũ AWS sẽ tiếp tục **nâng cấp toàn diện khả năng sẵn sàng của hệ thống** trong thời gian tới.

Hãy theo dõi để cập nhật thêm các **best practices mới nhất**.  
Bạn có thể bắt đầu sử dụng **Amazon ElastiCache for Redis** bằng cách truy cập [**Amazon ElastiCache Console**](https://console.aws.amazon.com/elasticache/).

**Thông Tin Về Tác Giả**

A person with sunglasses on her head

AI-generated content may be incorrect.

**Ruchita Arora** là **Quản lý Sản phẩm Cấp cao (Senior Product Manager)** tại **Amazon ElastiCache** và hiện đang phụ trách sát sao tất cả các khía cạnh liên quan đến dịch vụ ElastiCache.

Bên cạnh lĩnh vực cơ sở dữ liệu, cô còn có kinh nghiệm làm việc trong các lĩnh vực khác như: **Lưu trữ dữ liệu (storage)**, **Phát triển ứng dụng doanh nghiệp (enterprise application development)**, Và **viễn thông (telecommunication)**,

Với nhiều vai trò khác nhau trong cả **kỹ thuật (engineering)** và **quản lý sản phẩm (product management)**.