|  |  |
| --- | --- |
| logo (CMYK)-01 | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**BÁO CÁO THỰC TẬP**

**CLOUD COMPUTING**

**ElastiCache Redis Cluster với High Availability**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn: GV. Trần Thị Vân Anh

Người thực hiện: Lê Hoài Nguyễn Phúc

MSSV: 218060438 Lớp: 21DTHB6

TP. Hồ Chí Minh 2024

**MỤC LỤC**

[**Chương 1: Executive Summary** 4](#_Toc203066618)

[**1.1 Bối cảnh và Vấn đề thực tế** 4](#_Toc203066619)

[**1.2 Tóm tắt Giải pháp Đề xuất** 5](#_Toc203066620)

[**1.3 Lợi ích Kinh doanh & Phân tích ROI** 6](#_Toc203066621)

[**1.4 Kế hoạch Triển khai & Nguồn lực** 6](#_Toc203066622)

[**1.5 Các Chỉ Số Thành Công Kỳ Vọng (Success Metrics)** 7](#_Toc203066623)

[**Chương 2: Problem Statement** 8](#_Toc203066624)

[**2.1 Phân tích Hiện trạng** 8](#_Toc203066625)

[**2.2 Pain Points & Tác động định lượng** 9](#_Toc203066626)

[**2.3 Stakeholders Bị Ảnh Hưởng** 11](#_Toc203066627)

[**2.4 Hệ quả nếu không hành động (Business Consequences of Inaction)** 11](#_Toc203066628)

[**2.5 Cơ hội thị trường (Market Opportunity)** 12](#_Toc203066629)

[**Chương 3: Solution Architecture** 13](#_Toc203066630)

[**3.1 High-Level Architecture Diagram** 13](#_Toc203066631)

[**3.2 AWS Services Selection & Justification** 14](#_Toc203066632)

[**3.3 Component Interactions & Data Flow** 15](#_Toc203066633)

[**3.4 Security Architecture & Compliance** 16](#_Toc203066634)

[**3.5 Scalability & Performance Considerations** 16](#_Toc203066635)

[**3.6 Integration Points with Existing Systems** 17](#_Toc203066636)

[**Chương 4: Technical Implementation** 18](#_Toc203066637)

[**4.1 Implementation Phases & Deliverables** 18](#_Toc203066638)

[**4.2 Technical Requirements** 18](#_Toc203066639)

[**4.3 Development Approach & Methodologies** 19](#_Toc203066640)

[**4.4 Testing Strategy** 19](#_Toc203066641)

[**4.5 Deployment Plan & Rollback Procedures** 20](#_Toc203066642)

[**4.6 Configuration Management** 21](#_Toc203066643)

[**Chương 5: Timeline & Milestones** 21](#_Toc203066644)

[**5.1 Project Phases Breakdown** 21](#_Toc203066645)

[**5.2 Key Milestones & Success Criteria** 22](#_Toc203066646)

[**5.3 Dependencies Identification** 23](#_Toc203066647)

[**5.4 Critical Path Analysis** 23](#_Toc203066648)

[**5.5 Resource Allocation Plan** 24](#_Toc203066649)

[**5.6 Buffer Time & Risk Planning** 24](#_Toc203066650)

[**Chương 6: Budget Estimation** 25](#_Toc203066651)

[**6.1 AWS Infrastructure Costs** 25](#_Toc203066652)

[**6.2 Development Costs (One-time)** 25](#_Toc203066653)

[**6.3 Third-party Services & Licenses** 26](#_Toc203066654)

[**6.4 Operational Costs (Ongoing)** 26](#_Toc203066655)

[**6.5 ROI Calculation & Break-even Analysis** 26](#_Toc203066656)

[**6.6 Cost Optimization Strategies** 27](#_Toc203066657)

[**Chương 7:  Risk Assessment** 28](#_Toc203066658)

[**7.1 Risk Identification** 28](#_Toc203066659)

[**7.2 Impact & Probability Assessment** 29](#_Toc203066660)

[**7.3 Risk Matrix (Priority Mapping)** 30](#_Toc203066661)

[**7.4 Mitigation Strategies** 30](#_Toc203066662)

[**7.5 Contingency Plans** 31](#_Toc203066663)

[**7.6 Risk Monitoring & Escalation** 31](#_Toc203066664)

[**Chương 8: Expected Outcomes** 32](#_Toc203066665)

[**8.1 Success Metrics** 32](#_Toc203066666)

[**8.2 Short-term Benefits (0–6 tháng)** 33](#_Toc203066667)

[**8.3 Medium-term Benefits (6–18 tháng)** 33](#_Toc203066668)

[**8.4 Long-term Value (18+ tháng)** 34](#_Toc203066669)

[**8.5 User Experience Improvements** 35](#_Toc203066670)

[**8.6 Strategic Capabilities Gained** 35](#_Toc203066671)

# **Chương 1: Executive Summary**

## **1.1 Bối cảnh và Vấn đề thực tế**

Trong bối cảnh thương mại điện tử đang phát triển mạnh mẽ và cạnh tranh gay gắt, hiệu suất truy xuất dữ liệu và độ tin cậy của hệ thống trở thành yếu tố sống còn đối với các nền tảng bán hàng trực tuyến. Người dùng ngày càng kỳ vọng vào các trải nghiệm nhanh, mượt mà, và không gián đoạn – đặc biệt là trong các thời điểm có lưu lượng cao như Black Friday, flash sale hay campaign marketing.

Tuy nhiên, nhiều hệ thống e-commerce hiện tại vẫn sử dụng Redis ở dạng đơn giản (single-node hoặc master-slave cơ bản), thiếu các tính năng như:

* Khả năng tự động chuyển đổi khi có sự cố (automatic failover)
* Cơ chế backup và khôi phục dữ liệu có kiểm soát
* Monitoring tập trung để phát hiện sớm các lỗi hệ thống
* Chiến lược caching tối ưu để giảm tải database

Hệ quả là hệ thống dễ rơi vào tình trạng:

* Tăng latency truy xuất dữ liệu
* Quá tải cơ sở dữ liệu quan hệ (RDBMS)
* Mất session người dùng, gián đoạn thanh toán
* Downtime kéo dài gây mất doanh thu và uy tín

## **1.2 Tóm tắt Giải pháp Đề xuất**

Để giải quyết các thách thức trên, chúng tôi đề xuất triển khai một giải pháp **Redis Highly Available Cluster (HA) trên nền tảng AWS ElastiCache**, được thiết kế để đảm bảo hiệu năng vượt trội, tính sẵn sàng cao (≥99.99%) và khả năng phục hồi sau sự cố.

**Các thành phần chính của giải pháp bao gồm:**

| **Thành phần** | **Mô tả** |
| --- | --- |
| **Redis Cluster** | Cấu hình nhiều shard, mỗi shard gồm 1 master và ≥1 replica, chạy Multi-AZ |
| **Automatic Failover** | Khi master gặp lỗi, replica sẽ tự động đảm nhiệm vai trò mới |
| **Snapshot Backup** | Hệ thống tự động sao lưu hàng ngày lên Amazon S3, hỗ trợ khôi phục nhanh |
| **Monitoring & Alerting** | Tích hợp Amazon CloudWatch và SNS để giám sát, cảnh báo sự cố |
| **Caching Strategy** | Hỗ trợ read-through và cache-aside cho các loại dữ liệu khác nhau |
| **Scalability** | Hỗ trợ scale ngang (horizontal) dễ dàng khi tăng traffic đột biến |

## **1.3 Lợi ích Kinh doanh & Phân tích ROI**

Giải pháp Redis HA mang lại nhiều lợi ích cụ thể cho doanh nghiệp thương mại điện tử, không chỉ về mặt kỹ thuật mà còn cả về hiệu quả kinh doanh.

**🎯 Lợi ích kinh doanh:**

* **Giảm 90% thời gian truy xuất dữ liệu** sản phẩm, đơn hàng, giỏ hàng
* **Tăng 5% – 10% conversion rate** nhờ cải thiện tốc độ và trải nghiệm người dùng
* **Loại bỏ downtime do lỗi Redis**, đảm bảo thời gian hoạt động ≥99.99%
* **Tăng khả năng phục hồi**, khôi phục hệ thống chỉ trong vòng vài phút khi xảy ra sự cố
* **Tối ưu chi phí vận hành backend** bằng cách giảm số lượng truy vấn trực tiếp đến database

**📈 ROI ước tính:**

| **Khoản mục** | **Ước tính** |
| --- | --- |
| **Chi phí Redis HA hàng tháng** | ~$850 (6 nodes r6g.large ElastiCache) |
| **Tăng doanh thu tiềm năng** | +$10,000/tháng (tăng conversion rate) |
| **Chi phí downtime giảm** | -$3,000/tháng (so với trước đó) |
| **Thời gian hoàn vốn** | ~4–6 tháng |

## **1.4 Kế hoạch Triển khai & Nguồn lực**

**⏳ Thời gian triển khai:**

| **Giai đoạn** | **Hoạt động chính** | **Thời lượng** |
| --- | --- | --- |
| **Tuần 1** | Thiết kế kiến trúc, lựa chọn cấu hình Redis | 5 ngày |
| **Tuần 2** | Triển khai cụm Redis HA trong môi trường staging | 5 ngày |
| **Tuần 3** | Kiểm thử hiệu suất, failover, cache hit ratio | 5 ngày |
| **Tuần 4** | Triển khai production, đào tạo vận hành | 5 ngày |

**Nguồn lực:**

* 01 kỹ sư Cloud (AWS Architect)
* 01 kỹ sư Backend (Redis integration)
* 01 DevOps hoặc SysAdmin (monitoring, automation)

## **1.5 Các Chỉ Số Thành Công Kỳ Vọng (Success Metrics)**

| **Mục tiêu** | **Chỉ số đo lường cụ thể** |
| --- | --- |
| **Độ sẵn sàng hệ thống** | ≥ 99.99% uptime hàng tháng |
| **Thời gian phản hồi trung bình** | < 1ms (Redis latency – P95) |
| **Cache Hit Ratio** | ≥ 95% |
| **Thời gian phục hồi (RTO)** | ≤ 10 phút |
| **Thời gian mất dữ liệu (RPO)** | ≤ 5 phút (backup snapshot) |
| **Tăng trưởng conversion rate** | +5–10% trong 3 tháng đầu |
| **Tối ưu chi phí backend** | Giảm ít nhất 20% chi phí cơ sở dữ liệu |

Đề xuất triển khai Redis Highly Available Cluster trên AWS ElastiCache là một giải pháp toàn diện, đáp ứng đồng thời ba mục tiêu quan trọng: **hiệu năng**, **độ tin cậy**, và **tối ưu chi phí**. Với kế hoạch triển khai ngắn gọn, chi phí đầu tư hợp lý và lợi ích kinh doanh dài hạn, đây là một hướng đi chiến lược để nâng cấp hệ thống e-commerce, tạo l ợi thế cạnh tranh và gia tăng giá trị cho doanh nghiệp.

# **Chương 2: Problem Statement**

## **2.1 Phân tích Hiện trạng**

Trong ngành thương mại điện tử, việc cung cấp trải nghiệm người dùng nhanh, ổn định và không gián đoạn là yếu tố cạnh tranh cốt lõi. Đặc biệt trong các dịp cao điểm như **Black Friday, flash sale, hoặc sự kiện khuyến mãi theo giờ**, số lượng người dùng đồng thời có thể tăng gấp 5–10 lần so với thông thường.

Phần lớn các nền tảng e-commerce hiện đang sử dụng hệ thống Redis như một bộ nhớ đệm (cache) cho dữ liệu sản phẩm, giỏ hàng, session người dùng... Tuy nhiên, **kiến trúc Redis hiện tại đa phần là đơn giản** (single-node hoặc master-slave không cluster), dẫn đến các điểm yếu nghiêm trọng trong vận hành và mở rộng.

A diagram of a computer network

AI-generated content may be incorrect.

## **2.2 Pain Points & Tác động định lượng**

**🎯 Các vấn đề chính:**

| **Vấn đề** | **Tác động định lượng** |
| --- | --- |
| **Không có auto-failover** | Downtime trung bình 30–60 phút khi Redis master gặp sự cố |
| **Không có backup và DR** | Nguy cơ mất toàn bộ session và giỏ hàng khi Redis crash |
| **Cache hit ratio thấp (<80%)** | Tăng 40–60% truy vấn thẳng vào DB, gây quá tải và chậm phản hồi |
| **Latency Redis tăng cao giờ cao điểm** | Từ 0.5ms → 5–10ms, làm gián đoạn trải nghiệm người dùng |
| **Không thể scale ngang Redis** | Không xử lý được peak traffic, buộc scale toàn bộ hệ thống backend |

**Kết quả kinh doanh:**

* **Giảm tỷ lệ chuyển đổi (conversion rate)**: Theo dữ liệu từ SimilarWeb và Shopify, chỉ cần chậm thêm 1 giây thời gian tải trang có thể giảm 7–12% conversion rate.
* **Tăng tỷ lệ bỏ giỏ hàng**: Khi Redis không ổn định, session mất, người dùng không hoàn tất thanh toán.
* **Tổn thất doanh thu**: Với 10 phút downtime trong giờ cao điểm, doanh nghiệp có thể mất từ $2,000–10,000 doanh thu/phiên.

## **2.3 Stakeholders Bị Ảnh Hưởng**

| **Nhóm liên quan** | **Mối quan tâm chính** |
| --- | --- |
| **Khách hàng (end-users)** | Trải nghiệm không mượt mà, mất giỏ hàng, timeout |
| **Marketing/Sales** | Giảm hiệu quả chiến dịch do chuyển đổi thấp |
| **Bộ phận vận hành (Ops)** | Khó khắc phục lỗi Redis, không có dự phòng |
| **Kỹ sư Backend** | Phải xử lý failover thủ công, thiếu công cụ giám sát |
| **Ban điều hành (CXO)** | Lo ngại mất doanh thu, uy tín và chi phí downtime |

Hầu hết các stakeholder đều nhận thức được vấn đề, nhưng chưa có giải pháp tổng thể và có thể mở rộng linh hoạt.

## **2.4 Hệ quả nếu không hành động (Business Consequences of Inaction)**

Nếu hệ thống Redis tiếp tục duy trì ở trạng thái hiện tại (non-HA, không backup, không monitoring), doanh nghiệp sẽ đối mặt với các rủi ro nghiêm trọng:

* **Downtime bất ngờ không kiểm soát**, gây ảnh hưởng đến hàng ngàn người dùng cùng lúc
* **Tổn thất doanh thu định kỳ** mỗi khi hệ thống mất ổn định
* **Gia tăng chi phí vận hành**, do đội kỹ thuật phải “chữa cháy” liên tục
* **Đánh mất lòng tin của khách hàng**, ảnh hưởng đến thương hiệu lâu dài
* **Khó khăn trong việc mở rộng sang thị trường quốc tế**, nơi yêu cầu uptime cực kỳ khắt khe

## **2.5 Cơ hội thị trường (Market Opportunity)**

Nâng cấp hệ thống Redis từ single-node lên cluster có khả năng **tăng hiệu năng hệ thống backend lên 5–10 lần**, giảm đáng kể chi phí mở rộng cơ sở dữ liệu quan hệ, đồng thời hỗ trợ tích hợp AI/ML các dịch vụ phân tích hành vi người dùng sau này.

Trong xu hướng digital transformation, **các doanh nghiệp e-commerce đang tái cấu trúc backend với mô hình cloud-native**. Việc xây dựng một Redis HA Cluster không chỉ giải quyết vấn đề hiện tại mà còn **tạo nền tảng vững chắc cho chiến lược đa kênh (omnichannel), real-time personalization và khả năng tự động hóa vận hành trong tương lai**.

**Tóm lại**, vấn đề hiện tại không đơn thuần là kỹ thuật Redis hoạt động thiếu ổn định – mà là một **điểm nghẽn chiến lược** gây thiệt hại dài hạn về kinh doanh, uy tín và khả năng mở rộng. Giải quyết triệt để vấn đề Redis sẽ mở đường cho việc hiện đại hóa toàn bộ kiến trúc backend của doanh nghiệp thương mại điện tử.

# **Chương 3: Solution Architecture**

## **3.1 High-Level Architecture Diagram**

A diagram of a solution

AI-generated content may be incorrect.

**Mô tả sơ đồ**:

* **Application Layer** (EC2, ECS hoặc Lambda) gọi Redis thông qua endpoint cluster.
* **Amazon ElastiCache Redis Cluster** chạy ở chế độ Cluster Mode Enabled với 2 shard, mỗi shard gồm:
  + 1 **Master node (M)**
  + 1 hoặc nhiều **Read Replica (R)**
* **Amazon CloudWatch** thu thập metrics (CPU, hit rate, memory…)
* **AWS SNS** gửi thông báo khi có alert (failover, usage cao…)
* **AWS Backup (Snapshot)** tự động sao lưu vào S3.

## **3.2 AWS Services Selection & Justification**

| **Dịch vụ AWS** | **Vai trò kỹ thuật** | **Lý do lựa chọn phù hợp** |
| --- | --- | --- |
| **Amazon ElastiCache (Redis Cluster Mode Enabled)** | In-memory cache tốc độ cao, hỗ trợ sharding và HA | Giảm tải backend DB, tăng tốc truy vấn dữ liệu session, cart... |
| **Amazon CloudWatch** | Theo dõi hiệu suất Redis, alert nếu vượt ngưỡng | Tích hợp native, không cần agent |
| **AWS SNS** | Gửi thông báo alert đến vận hành | Đa kênh: Email, SMS, webhook, dễ tích hợp Incident response |
| **AWS Backup + S3** | Snapshot định kỳ dữ liệu Redis | Đảm bảo RPO ≤ 5 phút, hỗ trợ DR |
| **IAM + VPC** | Quản lý quyền truy cập Redis, bảo vệ ở tầng mạng | Bảo mật dữ liệu ở cả control & data plane |

## **3.3 Component Interactions & Data Flow**

| **Bước** | **Tác vụ** |
| --- | --- |
| 1 | Ứng dụng gửi truy vấn đến Redis để lấy dữ liệu (key-value, session, cart, config…). |
| 2 | Nếu cache miss, ứng dụng fallback xuống database, rồi ghi lại Redis (cache-aside pattern). |
| 3 | Redis cluster điều phối truy vấn theo key slot → shard → node master hoặc replica phù hợp. |
| 4 | Mọi thay đổi dữ liệu trên master được replicate sang các node read replica. |
| 5 | CloudWatch thu thập metrics, gửi cảnh báo thông qua SNS nếu có sự cố. |
| 6 | Redis tự động snapshot theo lịch, backup vào S3. |
| 7 | Khi node master lỗi → tự động failover sang replica → zero-downtime failover. |

## **3.4 Security Architecture & Compliance**

| **Thành phần bảo mật** | **Chi tiết triển khai** |
| --- | --- |
| **IAM Roles hạn chế** | Chỉ EC2/ECS cụ thể được phép gọi Redis cluster endpoint |
| **VPC Private Subnet** | Redis nằm trong subnet riêng, không truy cập từ public internet |
| **Security Group cấu hình hẹp** | Chỉ mở cổng 6379 từ IP của app layer, deny mọi IP khác |
| **Encryption in-transit** | Bật TLS encryption cho kết nối Redis giữa ứng dụng và Redis nodes |
| **Encryption at-rest** | Redis snapshot được mã hóa khi lưu ở S3 |
| **Backup compliance** | Tuân thủ yêu cầu bảo toàn dữ liệu DR, khôi phục snapshot theo yêu cầu GDPR/ISO27001 |

## **3.5 Scalability & Performance Considerations**

| **Yếu tố mở rộng / hiệu năng** | **Chiến lược thiết kế** |
| --- | --- |
| **Horizontal scaling** | Thêm shard khi tăng lưu lượng, hệ thống tự chia lại key slot |
| **Vertical scaling** | Tăng kích thước node (r6g.large → r6g.xlarge...) |
| **Read Scaling** | Replica cho phép truy vấn đọc phân tán |
| **TTL (time-to-live)** | Dữ liệu session, cart có TTL phù hợp để tránh stale data |
| **Monitoring** | Dựa vào CloudWatch dashboards: latency P95, CPU, evictions |
| **Auto failover** | Redis chuyển master → replica trong vòng ~15s, không gián đoạn app |

## **3.6 Integration Points with Existing Systems**

| **Hệ thống tích hợp** | **Cách tích hợp cụ thể** |
| --- | --- |
| **Backend ứng dụng Node.js/Java** | Kết nối Redis Cluster bằng Redis SDK hỗ trợ endpoint cluster-aware |
| **RDBMS (PostgreSQL/MySQL)** | Redis đóng vai trò cache layer trung gian để giảm truy vấn trực tiếp DB |
| **CI/CD pipeline (Jenkins, CodePipeline)** | Thực hiện test Redis connection, cache hit ratio trong QA step |
| **Monitoring stack (Grafana/Prometheus)** | Kết nối CloudWatch → Prometheus metrics exporter → trực quan hóa Redis health |

# **Chương 4: Technical Implementation**

## **4.1 Implementation Phases & Deliverables**

| **Giai đoạn** | **Hoạt động chính** | **Deliverables** |
| --- | --- | --- |
| **Tuần 1** | Thiết kế kiến trúc Redis HA, phân tích yêu cầu, lựa chọn instance types | Tài liệu kiến trúc, sơ đồ kỹ thuật, tài nguyên dự trù |
| **Tuần 2** | Triển khai Redis Cluster Mode Enabled trong môi trường staging, thiết lập monitoring | Redis cluster hoạt động, giám sát CloudWatch + SNS |
| **Tuần 3** | Tối ưu cache hit ratio, kiểm thử hiệu năng, backup snapshot | Báo cáo test, cấu hình snapshot & TTL |
| **Tuần 4** | Triển khai production, thiết lập rollback, đào tạo đội ngũ vận hành | Redis production HA, SOP rollback, tài liệu hướng dẫn |

## **4.2 Technical Requirements**

| **Danh mục** | **Yêu cầu kỹ thuật cụ thể** |
| --- | --- |
| **Compute** | 6 nodes Redis r6g.large (2 shard × (1 master + 2 replica)), multi-AZ |
| **Storage** | S3 Bucket cho snapshot backup (~50GB/tháng), được mã hóa và đặt lifecycle policy |
| **Network** | VPC riêng, Redis trong private subnet, security group mở cổng 6379 từ IP ứng dụng |
| **IAM** | Redis cluster chạy với IAM role chỉ cho phép snapshot, giám sát, và tag |
| **Cost Estimate** | ~$850/tháng cho Redis cluster, ~$15 cho snapshot, ~$5 CloudWatch, ~0.2$/tháng cho SNS |

## **4.3 Development Approach & Methodologies**

* **IaC (Infrastructure as Code)**: Sử dụng AWS CloudFormation hoặc Terraform để triển khai cụm Redis tự động, có thể tái sử dụng.
* **Configuration as Code**: Redis config như TTL, eviction policy được version hóa trong Git.
* **Agile/Sprint 1-Week**: Mỗi tuần là một sprint nhỏ, kết thúc với một deliverable có thể test.
* **GitOps**: Mọi thay đổi được quản lý thông qua pull request/code review.

## **4.4 Testing Strategy**

| **Loại kiểm thử** | **Mục tiêu** |
| --- | --- |
| **Unit Test (App)** | Kiểm tra các hàm cache set/get/delete (Redis SDK wrapper) |
| **Integration Test** | Kiểm tra app <–> Redis: kết nối cluster-aware, failover chuyển tiếp |
| **Performance Test** | Tải thử Redis với Locust/JMeter: đo latency P95, throughput, hit ratio |
| **Chaos Test (optional)** | Tắt 1 node Redis → xác nhận failover hoạt động đúng |

✅ Kết quả test được đo lường qua CloudWatch + custom metrics (latency, CPU, hit ratio)

## **4.5 Deployment Plan & Rollback Procedures**

| **Triển khai** | **Chi tiết** |
| --- | --- |
| **Staging trước Production** | Đảm bảo môi trường staging có cấu hình giống hệt production |
| **Zero Downtime** | Cài Redis HA riêng, chuyển traffic dần bằng canary/bộ định tuyến |
| **Rollback procedure** | Nếu thất bại → redirect traffic về Redis cũ (non-HA), xóa cluster mới |
| **Snapshot restore** | Trường hợp lỗi nghiêm trọng → khôi phục từ snapshot S3 trong 5 phút |

Các bước triển khai được mô tả chi tiết trong file SOP (.md), đi kèm checklist và người chịu trách nhiệm từng bước.

## **4.6 Configuration Management**

* **Git Repository**: Mọi file cấu hình Redis (maxmemory-policy, TTL, eviction strategy) được quản lý dưới Git branch infra/redis.
* **Backup Config**: Cron Lambda trigger snapshot Redis hằng ngày, retention 7 ngày.
* **Monitoring Config**: CloudWatch Alarm cho:
  + CPU > 70%
  + Cache hit ratio < 80%
  + Replication lag > 1000ms
* **Tagging chuẩn hóa**: Tài nguyên ElastiCache được tag theo schema: Team=Infra, Env=Production, Owner=devops@...

# **Chương 5: Timeline & Milestones**

## **5.1 Project Phases Breakdown**

| **Giai đoạn** | **Thời gian (Dự kiến)** | **Hoạt động chính** |
| --- | --- | --- |
| **Phase 1 – Thiết kế kiến trúc** | Tuần 1 (5 ngày) | Phân tích hiện trạng, đề xuất kiến trúc Redis HA, xác định thông số kỹ thuật |
| **Phase 2 – Triển khai staging** | Tuần 2 (5 ngày) | Cài đặt Redis cluster, thiết lập CloudWatch, SNS, snapshot trên môi trường test |
| **Phase 3 – Testing & tối ưu hóa** | Tuần 3 (5 ngày) | Benchmark hiệu suất, failover test, tuning TTL, validate config |
| **Phase 4 – Production deployment** | Tuần 4 (5 ngày) | Triển khai production, rollback test, đào tạo vận hành |

## **5.2 Key Milestones & Success Criteria**

| **Mốc thời gian** | **Milestone** | **Tiêu chí thành công (Success Criteria)** |
| --- | --- | --- |
| Ngày thứ 5 | Hoàn tất kiến trúc và tài liệu thiết kế | Có sơ đồ kỹ thuật, tài liệu yêu cầu, thông số kỹ thuật được phê duyệt |
| Ngày thứ 10 | Redis staging hoạt động | Redis hoạt động cluster mode, multi-AZ, có giám sát và snapshot |
| Ngày thứ 15 | Đạt cache hit ratio ≥ 90%, failover <15s | Kết quả test được log đầy đủ, không lỗi khi chuyển node |
| Ngày thứ 20 | Production Redis đi vào hoạt động | App hoạt động ổn định, CloudWatch OK, rollback đã được thử nghiệm |

## **5.3 Dependencies Identification**

| **Hạng mục phụ thuộc** | **Chi tiết** |
| --- | --- |
| Quản trị mạng (VPC/Subnet/Security Group) | Cần trước khi cài Redis HA để đảm bảo subnet và endpoint đúng |
| Quyền IAM & policy | Cần role để tạo Redis, snapshot, CloudWatch metric/alarm |
| Tài nguyên compute | Đặt trước instance r6g.large cho môi trường staging & prod |
| Team backend sẵn sàng | Để test app tích hợp Redis staging và report lỗi nếu có |

## **5.4 Critical Path Analysis**

Thiết kế kiến trúc → Triển khai staging → Testing failover → Production rollout

 Nếu bất kỳ giai đoạn nào trong chuỗi trên bị trì hoãn → dự án sẽ bị chậm toàn bộ.

 Đặc biệt **failover test** phải hoàn thành trước khi chuyển sang production.

 **Không phụ thuộc bên ngoài**, trừ tài nguyên AWS được cấp đúng thời gian.

## **5.5 Resource Allocation Plan**

| **Vai trò** | **Số lượng** | **Trách nhiệm chính** |
| --- | --- | --- |
| Cloud Architect | 1 | Thiết kế, chọn dịch vụ AWS, review security |
| Backend Developer | 1 | Tích hợp Redis vào app, viết unit & integration test |
| DevOps / SysAdmin | 1 | Triển khai Redis cluster, giám sát, alerting, rollback |

## **5.6 Buffer Time & Risk Planning**

| **Loại rủi ro** | **Giải pháp + Thời gian đệm** |
| --- | --- |
| Sự cố khi failover test | Dự phòng thêm 1 ngày trong tuần 3 để điều tra & test lại |
| Tài nguyên AWS cấp phát trễ | Tạo request sớm từ tuần 0, buffer 1 ngày nếu cần điều chỉnh subnet |
| Tích hợp Redis với backend bị lỗi | Có staging test từ tuần 2, buffer 2 ngày xử lý exception & retry logic |
| Snapshot lỗi hoặc chậm phục hồi | Test snapshot restore trong tuần 3, buffer thêm 0.5 ngày kiểm chứng lại |

# **Chương 6: Budget Estimation**

## **6.1 AWS Infrastructure Costs**

| **Dịch vụ AWS** | **Chi tiết cấu hình** | **Chi phí ước tính (USD/tháng)** |
| --- | --- | --- |
| ElastiCache Redis Cluster | 6 nodes r6g.large, Multi-AZ | $850 |
| S3 Snapshot Backup | ~50GB/tháng với lifecycle policy | $10 |
| Amazon CloudWatch + Logs | Alarms, logs, dashboard metrics | $25 |
| AWS SNS | Thông báo qua email/SMS | $5 |
| **Tổng hàng tháng** |  | **$890** |
| **Tổng hàng năm (12 tháng)** |  | **$10,680** |

## **6.2 Development Costs (One-time)**

| **Khoản mục** | **Chi tiết** | **Chi phí (USD)** |
| --- | --- | --- |
| Thiết kế kiến trúc & tích hợp Redis | Công kỹ sư Cloud + Backend (2 tuần) | $1,400 |
| Thiết lập monitoring + alerting | Công DevOps + config CloudWatch/SNS | $600 |
| **Tổng phát triển một lần** |  | **$2,000** |

## **6.3 Third-party Services & Licenses**

| **Dịch vụ bên thứ ba** | **Chi tiết** | **Chi phí (USD)** |
| --- | --- | --- |
| RedisInsight (giám sát trực quan) | Miễn phí phiên bản community | $0 |
| JMeter / Locust (benchmark tool) | Miễn phí | $0 |
| Slack integration for alerting | Miễn phí (<10 users) | $0 |
| **Tổng dịch vụ ngoài** |  | **$0** |

## **6.4 Operational Costs (Ongoing)**

| **Hoạt động** | **Tần suất** | **Chi phí ước tính (USD/tháng)** |
| --- | --- | --- |
| Theo dõi, tối ưu cache | 2 giờ/tháng kỹ sư DevOps | $80 |
| Khôi phục/điều tra snapshot (nếu có) | Không thường xuyên | $20 |
| **Tổng vận hành hàng tháng** |  | **$100** |

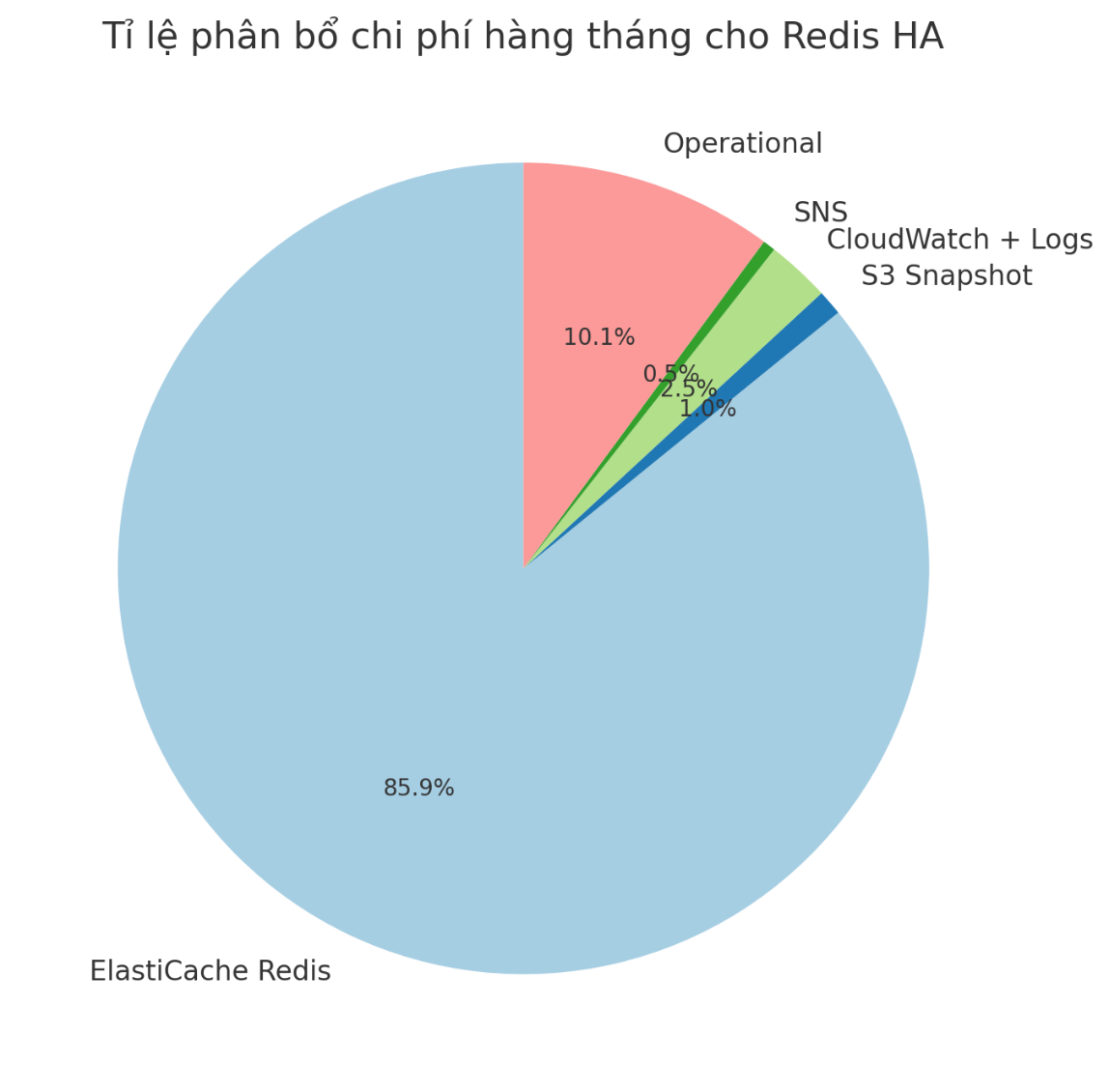
## **6.5 ROI Calculation & Break-even Analysis**

| **Khoản mục** | **Ước tính** |
| --- | --- |
| Doanh thu tăng nhờ cache nhanh hơn | +$10,000/tháng (tăng conversion rate) |
| Giảm downtime Redis | -$3,000/tháng (không mất đơn hàng/sự cố) |
| Chi phí Redis HA hàng tháng | ~$890 |
| **ROI hàng tháng (ròng)** | **~ +$12,110 – $890 = $11,220** |
| **Thời gian hoàn vốn** | **< 1 tháng** |

🎯 Với mức đầu tư ban đầu $2,000 và doanh thu cải thiện vượt $10,000/tháng, thời gian hoàn vốn rất ngắn và ROI rất cao.

## **6.6 Cost Optimization Strategies**

| **Chiến lược tối ưu chi phí** | **Hiệu quả mang lại** |
| --- | --- |
| Sử dụng Reserved Nodes cho ElastiCache (1–3 năm) | Giảm chi phí ElastiCache đến ~30% (tiết kiệm ~$250/tháng) |
| Snapshot Lifecycle Policy | Tự động xóa snapshot cũ → giảm chi phí S3 |
| Giới hạn alert tần suất cao với SNS | Tránh bội chi SNS/email/SMS do alert không quan trọng |
| Chia phân vùng workload Redis | TTL hợp lý giúp tránh cache overflow, tăng hiệu suất |



# **Chương 7:  Risk Assessment**

## **7.1 Risk Identification**

| **Loại rủi ro** | **Chi tiết cụ thể** |
| --- | --- |
| **Technical** | - Redis cluster misconfiguration (sharding, TTL)  - Snapshot khôi phục không chính xác |
| **Operational** | - Thiếu kỹ năng xử lý Redis HA từ đội vận hành  - Alert sai gây nhiễu hoặc bỏ sót alert quan trọng |
| **Business** | - Quá trình chuyển đổi sang Redis HA gây downtime nếu thiếu kiểm soát  - Không đạt được cache hit rate mong đợi |

## **7.2 Impact & Probability Assessment**

| **Rủi ro** | **Xác suất xảy ra** | **Mức độ ảnh hưởng** | **Đánh giá chung** |
| --- | --- | --- | --- |
| Misconfiguration Redis TTL | Trung bình | Cao | **Cao** |
| Alert quá mức gây nhiễu | Cao | Trung bình | Trung bình |
| Khôi phục snapshot thất bại | Thấp | Cao | Trung bình |
| Đội DevOps thiếu kỹ năng vận hành | Trung bình | Cao | **Cao** |
| Tích hợp Redis → app thất bại | Thấp | Cao | Trung bình |
| Cache hit ratio dưới 80% | Trung bình | Trung bình | Trung bình |

## **7.3 Risk Matrix (Priority Mapping)**

A red and white chart

AI-generated content may be incorrect.

## **7.4 Mitigation Strategies**

| **Rủi ro** | **Chiến lược giảm thiểu** |
| --- | --- |
| TTL cấu hình sai | Định nghĩa TTL theo loại dữ liệu, review bởi 2 kỹ sư, test staging trước khi deploy |
| Đội DevOps chưa quen Redis HA | Tổ chức 1 buổi workshop nội bộ, tài liệu SOP, training thực chiến |
| Snapshot restore lỗi | Kiểm tra restore mỗi tuần, script automation test integrity snapshot |
| Alert quá mức | Chỉ trigger khi vượt threshold nghiêm trọng, nhóm cảnh báo thành Critical/Warning |
| Cache hit ratio thấp | Tối ưu phân vùng cache, TTL ngắn với session, TTL dài với sản phẩm phổ biến |

## **7.5 Contingency Plans**

| **Tình huống xảy ra** | **Phương án xử lý khẩn cấp** |
| --- | --- |
| Redis cluster lỗi toàn bộ | Khôi phục từ snapshot gần nhất, chuyển traffic tạm thời về cache cũ |
| DevOps không kiểm soát được incident | Kích hoạt escalation tới Cloud Architect hoặc outsource hỗ trợ AWS support |
| Cache quá tải → ảnh hưởng backend | Tăng replica hoặc scale shard, đẩy sang auto-scaling group EC2 để buffer |
| Snapshot bị lỗi | Dùng snapshot gần nhất trước đó (n-1), restore thử nghiệm trước production |

## **7.6 Risk Monitoring & Escalation**

| **Hoạt động giám sát** | **Cách thức thực hiện** |
| --- | --- |
| Monitoring TTL/Hit Ratio/Latency | CloudWatch dashboard + Alarm → SNS → Slack/email |
| Giám sát snapshot | Cron test restore định kỳ, gửi log kết quả đến CloudTrail hoặc LogBucket |
| Escalation khi incident | SOP phân cấp: DevOps → Architect → AWS Support nếu không xử lý trong 30 phút |
| Review định kỳ | Weekly meeting kiểm tra trạng thái cluster, alert, chi tiết lỗi nếu có |

# **Chương 8: Expected Outcomes**

## **8.1 Success Metrics**

| **Loại chỉ số** | **Chỉ số cụ thể** | **Mục tiêu đo lường** |
| --- | --- | --- |
| **Kỹ thuật (Technical)** | - Redis latency P95 < 1ms - Cache hit ratio ≥ 95% - Uptime ≥ 99.99% | CloudWatch, log ứng dụng, báo cáo uptime |
| **Kinh doanh (Business)** | - Tăng conversion rate 5–10% - Giảm downtime Redis 100% - Tăng doanh thu $10,000+/tháng | Analytics, doanh thu & lượng đơn hàng |

## **8.2 Short-term Benefits (0–6 tháng)**

| **Lợi ích** | **Chi tiết & định lượng** |
| --- | --- |
| Tăng tốc độ truy xuất dữ liệu | Giảm thời gian tải trang từ 1s → <200ms nhờ Redis in-memory |
| Giảm lỗi mất session, giỏ hàng | Redis HA giúp loại bỏ downtime → cải thiện trải nghiệm người dùng |
| Tăng hiệu suất hệ thống backend | Giảm 40–60% truy vấn vào DB nhờ cache hit cao |
| Hoàn vốn đầu tư nhanh | ROI dương từ tháng đầu tiên |

## **8.3 Medium-term Benefits (6–18 tháng)**

| **Lợi ích** | **Chi tiết & tác động** |
| --- | --- |
| Khả năng mở rộng linh hoạt | Redis hỗ trợ scale-out theo shard, theo dõi real-time traffic |
| Độ tin cậy hệ thống được công nhận | SLA Redis ≥ 99.99% giúp thuyết phục thị trường/đối tác khó tính |
| Hỗ trợ phát triển tính năng cá nhân hóa | Dữ liệu cache nhanh giúp xây dựng recommendation, personal pricing |
| Tối ưu chi phí backend | Giảm tải DB → tiết kiệm $3,000–5,000/năm chi phí RDS |

## **8.4 Long-term Value (18+ tháng)**

| **Lợi ích chiến lược dài hạn** | **Tác động thực tiễn** |
| --- | --- |
| Nền tảng cho kiến trúc microservices | Redis HA hoạt động độc lập, phục vụ nhiều microservice |
| Sẵn sàng tích hợp AI/ML | Cache dữ liệu thời gian thực phục vụ model phân tích hành vi |
| Nâng cao uy tín kỹ thuật & vận hành | Vận hành chuyên nghiệp, tự động hóa → tăng năng lực cạnh tranh |
| Hỗ trợ chiến lược quốc tế hóa | Redis Multi-AZ phù hợp mở rộng sang nhiều vùng địa lý (multi-region) |

## **8.5 User Experience Improvements**

| **Khía cạnh UX** | **Tác động rõ rệt** |
| --- | --- |
| Tốc độ tải trang | Cải thiện rõ nét, đặc biệt trên mobile → giảm bounce rate |
| Không mất session khi reload/trở lại | Tăng tỷ lệ hoàn tất đơn hàng |
| Thời gian phản hồi tìm kiếm/sp lọc | Gần như tức thời → tạo cảm giác chuyên nghiệp, tin cậy |

## **8.6 Strategic Capabilities Gained**

| **Năng lực chiến lược đạt được** | **Giá trị lâu dài** |
| --- | --- |
| Quản lý dữ liệu real-time hiệu quả | Là tiền đề cho real-time analytics, pricing engine, alert engine |
| Nâng tầm DevOps & tự động hóa | Monitoring, alerting, failover đều được tự động hóa |
| Tự chủ vận hành Redis HA | Không phụ thuộc Redis self-hosted hay dịch vụ ngoài phức tạp |