**Cloud Cost Optimization Plan**

**Loic Ndeh**

**1. Introduction**

Cloud adoption has given businesses unprecedented scalability, agility, and access to global markets. However, rapid growth often results in inefficient spending patterns. For a mid-sized e-commerce company, cloud costs have increased by 150% in the past year, while revenue has grown only 75%. This misalignment presents a strategic challenge: cloud resources must be optimized to reduce unnecessary costs while maintaining the performance levels necessary for customer satisfaction and peak shopping demand.

The purpose of this plan is to analyze current cloud resource utilization, recommend cost-saving strategies, develop a phased implementation roadmap, and design a monitoring framework that ensures ongoing efficiency. The recommendations draw upon cost optimization principles outlined in the AWS Well-Architected Framework, the Google Cloud Architecture Framework, and Azure’s Cloud Adoption Framework (CAF)

A screenshot of a computer

AI-generated content may be incorrect.

**2. Analysis of Current Resource Utilization**

| **Resource** | **Current State** | **Observed Inefficiencies** |
| --- | --- | --- |
| Web Servers | 50 servers running continuously | No auto-scaling; over-provisioned during off-peak hours |
| Database Servers | 4 instances @ 40% average utilization | Under-utilization; potential right-sizing or serverless migration |
| Storage | 500 TB with mixed access patterns | Flat storage model; no tiering for cold data |
| CDN | Global delivery network | Possibly overextended with unused regions |
| Backup/DR | Standard backups only | No lifecycle management; archival opportunities missed |

**Key Observations**

* Compute resources are over-provisioned, running 24/7 without scaling, causing unnecessary costs.
* Databases run at less than half capacity, leading to wasted resources.
* Storage is treated as hot data across all 500 TB, inflating costs by not separating infrequent or archival data.
* CDN footprint includes unused regions, increasing egress and request costs.
* Backups are stored in the same expensive storage class as production data instead of Glacier or other archival solutions.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**3. Recommended Cost Optimization Strategies**

**3.1 Compute Optimization**

* **Right-Sizing:** Reduced baseline to 12 EC2 instances under Savings Plans.
* **Auto-Scaling:** Added 38 burst EC2 instances that run only during peak demand.
* **Reserved Commitments:** Implemented Savings Plans for 1-year term on baseline servers, cutting compute costs by ~60%.

**EC2 Optimized Configuration]**

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

* Before optimization, it showed 50 always-on servers.
* After optimization showing split between 12 reserved and 38 burst instances.

**3.2 Database Optimization**

* **Right-Sizing:** Scaled down database instances from db.r5.xlarge to db.r5.large.
* **Reserved Pricing:** Converted on-demand databases to 1-year reserved instances.
* Resulting cost reduction: **62%**, dropping from $9,515 → $3,632 per month.

**RDS Configuration**  
 *the final configuration screen showing RDS reserved instances with reduced instance type and pricing.*

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**3.3 Storage Optimization**

* **Tiered Storage Model Implemented:**
  + **S3 Standard (Hot Data):** 200 TB
  + **S3 Standard-IA (Warm Data):** 200 TB
  + **S3 Glacier Deep Archive (Cold Data):** 100 TB
* Lifecycle policies automatically transition aging data to lower-cost tiers.
* Savings achieved:
  + Glacier Deep Archive now costs **$123/month** vs. **$2,300/month** previously.
  + Warm tier costs **$5,302/month**, appropriately priced for occasional access.

**S3 Storage Tiering**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. S3 Standard (Hot 200 TB)
2. S3 Standard-IA (Warm 200 TB)
3. S3 Glacier Deep Archive (Cold 100 TB)

**3.4 Network & CDN Optimization**

* Reviewed CloudFront data transfer and request patterns:
  + **North America:** 79 TB/month
  + **Europe:** 20 TB/month
  + Asia disabled for cost savings.
* Cleaned up duplicate request entries and applied a **90% cache hit ratio**.
* Monthly CDN costs reduced by ~11% from $8,634 → $7,708.

**CloudFront Final Configuration**

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**3.5 Backup & DR**

* **Hot Backups:** 10 TB kept in S3 Standard for immediate restores.
* **Archived Backups:** 90 TB migrated to Glacier Deep Archive, cutting costs by 85%:
  + $2,304 → $112/month.

**Backup Configuration**

A screenshot of a computer

AI-generated content may be incorrect.

**4. Implementation Plan & Risk Assessment**

| **Phase** | **Activity** | **Timeline** | **Risks** | **Mitigation** |
| --- | --- | --- | --- | --- |
| 1 | Assess workloads with Cost Explorer & CloudWatch | 2 weeks | Incomplete utilization data | Use 30-day monitoring baseline |
| 2 | Implement right-sizing & auto-scaling for EC2 | 1 month | Performance dips during rollout | Simulate traffic in staging before production |
| 3 | Purchase Savings Plans for baseline workloads | 1 month | Overcommitment risk | Start with 50% of baseline before expanding |
| 4 | Apply S3 storage tiering and lifecycle rules | 1 month | Retrieval delays for cold data | Define retrieval SLAs with stakeholders |
| 5 | Optimize CDN endpoints & DR archival processes | 2 weeks | Downtime or latency issues | Make changes during low-traffic periods |

**5. Performance Monitoring Framework**

* **Compute:**
  + AWS CloudWatch metrics for CPU/memory utilization and scaling events.
  + Alarms for under/over-utilization.
* **Databases:**
  + RDS Performance Insights for query optimization.
* **Storage:**
  + Monitor transitions between S3 tiers.
  + Watch retrieval costs for Glacier data.
* **CDN:**
  + Monitor cache hit ratios, latency, and egress volumes.
* **Cost Monitoring:**
  + AWS Cost Explorer for monthly cost trends.
  + Tag resources for cost allocation.
* **Governance:**
  + Adopt FinOps practices for cloud financial management.

**6. ROI Analysis**

| **Strategy** | **Current Monthly Cost** | **Optimized Monthly Cost** | **Monthly Savings** | **% Savings** |
| --- | --- | --- | --- | --- |
| Compute Right-Sizing + Auto-Scaling | $3,504 | $1,424 | $2,080 | 60% |
| Database Optimization (RDS) | $9,515 | $3,632 | $5,883 | 62% |
| Storage Tiering (500 TB) | $19,592 | $18,259 | $1,333 | 7% |
| CDN Optimization | $8,634 | $7,708 | $926 | 11% |
| Backup Optimization | $2,304 | $348 | $1,956 | 85% |
| **TOTAL** | **$43,550** | **$31,373** | **$12,177** | **27.9%** |

**Final Optimized Summary - Figure 10**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**7. Conclusion**

The optimized cloud architecture reduces the company’s monthly spending from **$43,550 to $31,373**, achieving nearly **28% cost savings** without impacting performance.

Key drivers of savings include:

* EC2 right-sizing and auto-scaling with Savings Plans.
* Database consolidation and reserved pricing.
* Tiered S3 storage and lifecycle management.
* Efficient CDN configuration and regional optimization.
* Glacier archival for long-term backup storage.

These improvements ensure that as traffic grows, costs remain predictable and aligned with business growth. With ongoing monitoring and FinOps practices, the company can continue to refine cloud costs while maintaining excellent customer experiences.

**8. References**

* Amazon Web Services. (2023). [AWS Well-Architected Framework](https://aws.amazon.com/architecture/well-architected/?utm_source=chatgpt.com)
* Amazon Web Services. (2024). [AWS TCO Calculator](https://aws.amazon.com/tco-calculator/?utm_source=chatgpt.com)
* Google Cloud. (2024). [Cloud Architecture Framework](https://cloud.google.com/architecture/framework?utm_source=chatgpt.com)
* Microsoft Azure. (2024). [Azure Cost Management + Billing](https://azure.microsoft.com/en-us/pricing/calculator/?utm_source=chatgpt.com)
* FinOps Foundation. (2024). [Cloud Financial Management Best Practices](https://www.finops.org?utm_source=chatgpt.com)