FinOps Case Study: AWS Static Website Cost Modeling and Optimization

Overview

This case study examines the cloud economics of a static website hosted on AWS. It explores cost modeling, usage assumptions, service-level breakdowns, and the impact of basic cost optimizations.

Architecture Breakdown

| Layer | AWS Service(s) | |---|---| | **Frontend** | Amazon S3 (static HTML/CSS/JS) | | **CI/CD** | GitHub Actions (dry-run + fallback) | | **Storage** | S3 with bucket policies (no ACLs) | | **Networking** | CloudFront CDN + Route 53 DNS | | **Logic** | Lambda Function URL (visitor tracking) | | **Database** | DynamoDB (on-demand + TTL) | | **Monitoring** | S3 Access Logs + CloudWatch (Lambda) | | **Security** | IAM roles, least privilege, CORS |

Usage Assumptions

- **Traffic Pattern**: 2,500 visitors/month steady-state
- **Asset Load**: ~1 MB per visitor
- **Lambda**: x86, 120ms average duration
- **DynamoDB**: On-demand reads/writes with TTL enabled
- **Route 53**: One hosted zone, standard query volume
- **Monitoring**: CloudWatch logs for Lambda + S3 access logs

---

Monthly Cost @ 2,500 Visitors

2500 Visitor Optimization Impact

Yearly Cost by AWS Service

12-Month Cumulative Cost

Summary Analysis

Cost Drivers

- **CloudFront** and **Lambda** were the top cost contributors under initial modeling.
- **Route 53** remained flat across usage levels.

Optimization Techniques Applied

- Enabled **aggressive CloudFront caching** to reduce origin fetches.
- Reduced **Lambda memory** and execution time.
- Converted S3 to **intelligent tiering** to reduce storage cost.
- Added TTL and partitioning to **DynamoDB** for storage efficiency.

Results

- Achieved a **~25% reduction** in overall monthly spend at low traffic levels.
- Demonstrated significant **annual savings** in a stable usage scenario.

FinOps Takeaways

| Principle | Action Taken |
|---|---| | **Right-sizing** | Tuned Lambda memory and timeout | | **Elasticity** | Used on-demand DynamoDB,

Lambda scaling | | **Cost Visibility** | AWS Budgets, dashboards, usage alerts enabled | | **Optimization** | CDN cache tuning, asset compression | | **Cost Allocation** | Layer-to-service Sankey flow chart (see appendix) |

Appendix

Cost Allocation Flow

Sankey Diagram

Optimization Impact at 50K Visitors

Optimization Impact Labeled