QBurst | A High AI-Q™ Company

AWS Cost Optimization via Smart Autoscaling

Implementing advanced DevOps strategies, including scheduled shutdowns and Karpenter/KEDA autoscaling, across AWS EKS infrastructure to significantly reduce cloud spend for a B2B hotel booking platform.

Overview

The client needed to solve the problem of rising cloud costs tied directly to increased business connectivity on their B2B hotel booking platform. The QBurst DevOps team implemented a comprehensive, multi-faceted optimization strategy across their AWS infrastructure.

- Achieved nearly 60% cost savings on development environments through strategic scheduled EC2 shutdowns managed by EventBridge and Lambda.
- Enhanced scalability and cost-efficiency by adopting Karpenter for intelligent EKS node autoscaling and KEDA for event-driven scaling of intermittent workloads (RabbitMQ/Redis).
- Reduced overall compute costs by 37% using a No Upfront EC2 Savings Plan and achieved an additional 20% cost reduction via strategic Instance Type Optimization (T4g over T3).

Client Profile

The client is a B2B platform operating in the hospitality industry managing the connectivity, contracts, and booking flow between hotels, third-party suppliers, and travel agencies.

Challenges: Uncontrolled Scaling and Idle Costs

The client's primary challenge was linking business growth (new connections) directly to unsustainable infrastructure costs:

- **Linear Cost Growth:** Each new supplier or customer connection led to higher computing power demands, resulting in proportional cloud cost increases.

- **Idle Resource Waste:** Development/UAT environments were running 24/7 despite only being needed during business hours, leading to significant wasted spend.
- **Inefficient Scaling:** Production EKS nodes were not optimally scaled for real-time usage, resulting in underutilized EC2 instances.
- **Storage Bloat:** Hidden costs were accumulating from obsolete snapshots, oversized EBS volumes, and outdated container images.

Our Solution: Strategic Cost Optimization and Autoscaling Mechanisms

QBurst executed a multi-year, strategic program focused on integrating intelligent autoscaling mechanisms and enforcing strict cost governance across the client's AWS landscape.

Cost Optimization Initiatives

- **Scheduled Shutdowns:** Implemented EventBridge schedules and Lambda functions to power down UAT EC2 instances during idle times (full weekends, overnight on weekdays).
- **Instance Type Optimization:** Migrated compute instances to the T4g instance family for a better price-performance ratio.
- **Savings Plans:** Adopted No Upfront EC2 Savings Plans for the T3 instance family based on AWS recommendations.
- **Cleanup Policies:** Configured ECR Lifecycle Policies to automatically remove unused images and executed manual cleanup of obsolete snapshots and oversized EBS volumes.

Scalability Enhancements

- **Karpenter Autoscaler:** Deployed Karpenter to replace legacy autoscaling, ensuring optimal EKS node autoscaling based on pod resource requests, preventing node underutilization.

- **Event-Driven Scaling (KEDA):** Used KEDA (Kubernetes-based Event Driven Autoscaling) to activate workloads only in response to specific events, optimizing resource usage for intermittent workloads.
    - **RabbitMQ Queue Scaler:** Dynamically scaled worker pods based on message count thresholds.
    - **Redis List Scaler:** Enabled scaling only when Redis lists contained messages (on-demand processing).
- **Fargate for Workers:** Integrated Fargate for low-usage or intermittent worker pods, ensuring the client only pays for actual CPU usage and reducing overall node resource consumption.

Impact

- **Sustainable Scaling:** The platform can now add more capacity (suppliers/customers) with a comparably small incremental cloud cost, decoupling business growth from linear cost increase.
- **Immediate Cost Savings:** Achieved nearly 60% savings on development and UAT environments due to scheduled shutdowns.
- **Compute Cost Efficiency:** Secured a total compute cost reduction of over 57% (37% from Savings Plan + 20% from Instance Optimization) compared to initial on-demand pricing.
- **Optimized Resource Utilization:** Karpenter and KEDA ensure that resources (EC2 instances and pods) are not provisioned or charged for when they are idle, maximizing efficiency.
- **Enhanced Agility:** The foundation is now set for sustainable, high-performance scaling in a cost-effective manner.