

# Optimizing Cloud Efficiency with AI Powered Prompt Engineering: Cost-Effective Strategies for Cloud Native Platforms



Hari Yerramsetty  
Software Engineer, Infrastructure  
Flexport

# Table of Contents

- Introduction to *Cloud-Native Cost Optimization*
- Key cost drivers in *Cloud-Native Platforms*
- Resource Management Techniques
- Container and Serverless Optimization
- Storage and Data Management
- Monitoring and Cost Visibility
- FinOps - A Cultural Approach to Cost Optimization
- Leverage Prompt Engineering & Finops to drive Cloud Efficiency
- Conclusion

# Introduction To Cloud-Native Cost Optimization

- Cloud-native platforms provide unmatched flexibility and scalability, revolutionizing how applications are built and deployed.
- However, the dynamic and complex nature of these environments introduces challenges in cost management.
- **Statistics:** Studies show up to 30% of cloud spending is wasted due to resource mismanagement.
- **Objective:** This presentation covers strategies for optimizing costs in cloud-native setups, enabling financial responsibility without compromising on agility or scalability

# Key cost drivers in Cloud-Native Platforms

- **Compute:** Costs from virtual machines, containers, and serverless executions vary with instance types, runtime durations, and usage patterns
- **Storage:** Expenses from various storage types (object, block, and managed databases), impacted by data volume and access frequency
- **Data Transfer:** Costs from network egress charges, which increase with data-intensive applications or multi-region deployments
- **Third-Party Services:** Integration with managed services or third-party APIs, each with unique pricing models

## Common Challenges

- **Over-Provisioning:** Excess resources can lead to unnecessary expenses
- **Lack of Visibility:** Resource usage is often not clear
- **Multi-Cloud Complexity:** Managing costs across different cloud providers adds to the challenge

# Resource Management Techniques

**Right-Sizing:** Ensures resource allocation matches workload needs. Tools like AWS Trusted Advisor and Azure Advisor provide insights on ideal instance sizes

**Auto-Scaling:** Dynamically adjusts resources based on real-time demand, saving up to 30% compared to static provisioning.

# Resource Management Techniques

## Instance Type Variety

**Spot Instances:** Cost-effective for fault tolerant tasks like batch processing, CI/CD

**Reserved Instances:** Suitable for predictable workloads with long-term requirements, offering substantial discounts

**On-Demand Instances:** Used for short-term, variable needs.

# Container and Serverless Optimization

- **Resource Limits & Requests:** Define minimum and maximum usage to prevent over-provisioning.
- **Density Optimization:** Increase container density per node while balancing cost and performance
- **Node Autoscaling:** Automatically adjusts node count based on demand, improving resource utilization



# Container and Serverless Optimization

- **Monitoring Execution Times:** Regularly analyze execution times to identify and resolve performance bottlenecks
- **Reducing Idle Times:** Implement efficient cold start strategies to minimize idle costs
- **Code Optimization:** Minimize dependencies and optimize code for better memory and CPU usage.

# Storage and Data Management

- **Data Lifecycle Management:** Implement policies to move data across storage tiers based on access patterns. For example, transition infrequently accessed data to low-cost storage options
- **Infrequent Access Storage Tiers:** Use storage classes optimized for infrequent access, providing significant cost savings without sacrificing availability
- **Compression & Deduplication:** Techniques like data compression and deduplication reduce storage needs, especially effective for repetitive or compressible datasets
- **Regular Audits:** Routine audits help identify and remove unnecessary data, maintaining storage efficiency and compliance

# Monitoring And Cost Visibility

- **Continuous Monitoring:** Essential for identifying unexpected usage spikes and tracking resource utilization trends.
- **Cost Visibility Tools:**
  - *AWS Cost Explorer:* Provides detailed insights into cost patterns, drivers, and forecast expenses.
  - *Google Cloud Cost Management:* Offers tools for cost forecasting, budget alerts, and detailed expense breakdowns.
  - *Azure Cost Management + Billing:* Delivers cost analysis, budgeting, and insights across multi cloud and hybrid environments
- **Budgets and Alerts:** Set spending thresholds and receive alerts when nearing limits to proactively prevent overspending.
- **Cost Allocation Tags:** Attribute expenses to specific projects or departments, supporting more granular tracking and optimization.

# FinOps - A Cultural Approach To Cost Optimization

- **FinOps (Financial Operations):** A framework for cost management that integrates financial accountability into cloud spending
- **Key Principles:**
  - **Cross-functional collaboration:** Involves finance, engineering, and operations teams, fostering shared responsibility
  - **Continuous Monitoring:** Enables real-time tracking of usage and costs for informed optimization
  - **Developer Cost Awareness:** Educate developers on cost implications, encouraging cost-conscious design choices
- **Benefits:**
  - Aligns financial goals with technical decisions, supporting sustainable growth.
  - Enhances transparency and enables data-driven decisionmaking

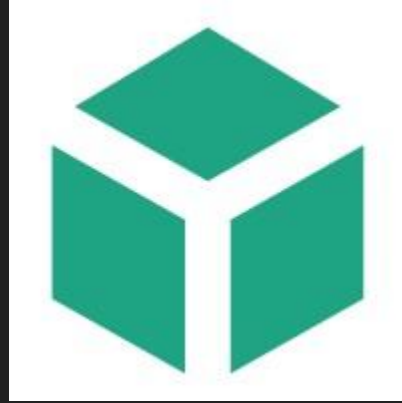
# Leverage AI & Finops to drive Cloud Efficiency

- **Real-time Cost Monitoring and Analytics:** AI-powered tools can continuously monitor cloud usage and expenses in real time, providing visibility into spending patterns and identifying anomalies. According to Gartner, AI-driven cloud cost optimization will drive significant advancements in predictive analytics
- **Anomaly Detection:** AI algorithms can analyze historical spending data to detect cost anomalies and irregularities, allowing businesses to address these issues promptly
- **Eliminating Inefficient Resource Use:** AI can help eliminate idle cloud resources and unused resources by automatically shutting down or resizing them, reducing wasteful spending
- **Demand Forecasting and Autoscaling:** AI can predict future demand for cloud resources, enabling better planning and deployment, and dynamically adjusting resources to avoid over-provisioning

# Using Prompt Engineering for achieving Cloud Efficiency

- **Role Prompting:** Role prompting is a technique in prompt engineering to control the output generated by the model by assigning a specific role to the model.

We can make use of roles like FinOps Expert and craft prompts by providing more context to the prompt



# Some prompts using FinOps expert role

ChatGPT 4o ▾

Share

As a finops expert, Can you please help deep dive into our AWS cloud bill. Please explain what is the different unblended costs and amortized costs



Certainly! Here's a breakdown of the two cost types in AWS billing:

## 1. Unblended Costs:

- Unblended costs refer to the raw costs for AWS resources without any spreading out or redistribution. This means the costs shown are exactly what is incurred for each specific service or resource for a given period.
- Unblended costs do not account for Reserved Instance (RI) or Savings Plan discounts that are spread across multiple accounts or services.
- For example, if you purchase an RI in a single month, its full cost will show up in that month unblended costs, even though the benefit might spread over a year.

# Analyze AWS invoices with AI models

ChatGPT 4o ▾

↑ Share



invoice495046353.pdf  
PDF

As a finops expert, Can you please analyze our AWS invoice and let us why was our bill higher than usual. Can you please provide details report on how could we reduce our AWS spend?



The AWS invoice you provided has the following breakdown:

- **Total Amount Due:** \$0.53
- **Billing Period:** June 1 - June 30, 2020
- **Service Charges:**
  - **Amazon Route 53:** \$0.53 (includes \$0.50 in charges and \$0.03 estimated sales tax)



Message ChatGPT





# AI Agentic Approach

- We can build AI agents like Simple Reflex Agents using open source frameworks like LangChain (<https://python.langchain.com/docs/introduction/>) and CrewAI (<https://www.crewai.com/open-source>) that can act as part of the automation
- These agents can automatically resize the workloads, shutdown unused resources, and take actions on behalf of the infrastructure team
- These agents can be built using any of the AI models like ChatGpt, Claude, Gemini and LLama



 **LangChain**

**crewai**

# Conclusion

- Cost optimization in cloud-native platforms requires a comprehensive approach that balances technical strategies with organizational shifts
- By understanding the primary cost drivers—such as compute resources, storage, data transfer, and third-party services—organizations can implement tailored techniques to reduce expenses without compromising performance
- Resource management practices, like right-sizing, auto-scaling, and diversified instance types, form the backbone of effective cost optimization. Additionally, container and serverless optimization methods, coupled with efficient storage and data lifecycle management, are crucial to sustaining cost-efficient operations as demand fluctuates
- A strong monitoring and cost visibility framework is equally essential, with tools that enable real-time tracking of expenses and resource utilization
- FinOps principles bring a cultural dimension to this strategy, encouraging cross-functional collaboration and fostering cost awareness within development teams.
- By embedding financial accountability into every stage of cloud management, organizations can align spending with business goals, creating a culture of continuous improvement.