

---

# Laying the Foundation: Setting Up Your Environment for Cost Optimization

**AWS Whitepaper**



# **Laying the Foundation: Setting Up Your Environment for Cost Optimization: AWS Whitepaper**

Copyright © 2018 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

## Table of Contents

Laying the Foundation: Setting Up Your Environment for Cost Optimization .....	1
Abstract .....	1
Introduction .....	1
Cost Optimization Pillars .....	2
Cost Allocation Supports Ownership and Accountability .....	3
Use Tagging to Organize Your Environment and Drive Accountability .....	4
Define Mandatory Cost Tagging .....	4
Enforce Quality of Tagging .....	4
Tagging Tools .....	4
AWS Account Structure Considerations .....	6
Measurement and Management .....	7
Enable Teams to Architect for Cost .....	8
Create a Cloud Center of Excellence .....	10
AWS Tools for Reporting and Cost Optimization .....	11
Conclusion .....	12
Resources .....	13
Document Details .....	14
Contributors .....	14
.....	14
AWS Glossary .....	15

# Laying the Foundation: Setting Up Your Environment for Cost Optimization

Publication date: **March 2018** ([Document Details \(p. 14\)](#))

## Abstract

This is the first in a series of whitepapers designed to support your cloud journey. This paper seeks to empower you to maximize value from your investments, improve forecasting accuracy and cost predictability, create a culture of ownership and cost transparency, and continuously measure your optimization status.

This paper discusses the pillars of cost optimization and the key mechanisms for achieving them.

## Introduction

The cloud offers tremendous opportunity for increased agility, faster innovation, and lower total cost of ownership. The organizations that are the most successful in moving from on-premises environments to the cloud are those that establish a well-defined strategy for approaching this new IT operating model early in their journey. Moving from a model of large upfront investment in data centers to the consumption-based model of AWS requires changes to tools, processes, and mindsets to ensure that costs are effectively managed.

The most important practice for effectively optimizing costs with AWS is to start early. Although many cost optimization practices are relatively easy to implement in small environments, you need new operational best practices, automation, and organizational incentives to be successful at scale across large environments and enterprises. Establishing these best practices early in your journey can help you establish the right processes and behaviors to ensure success when you hit scale.

# Cost Optimization Pillars

Regardless of your workload or architecture, there are five cost optimization pillars that apply across nearly all environments. The pillars of cost optimization are:

- **Right size** – Ensure that what you provision matches what you need. For example, for compute, you provision for CPU, memory, storage, and network throughput.
- **Increase elasticity** – Traditional IT costs and hardware requirements are tailored for peak usage and are rarely turned off. In the cloud, you can optimize cost to meet dynamic needs and turn resources off when they are not needed. For example, you can usually turn off non-production instances for 70% or more of any given week.
- **Leverage the right pricing model** – AWS provides a range of pricing models (On-Demand and Spot Instances for variable workloads and Reserved Instances for predictable workloads). Choose the right pricing model to optimize costs based on the nature of your workload.
- **Optimize storage** – AWS provides multiple storage tiers at prices designed to meet performance. By identifying the most appropriate destination for specific types of data, you can reduce Amazon Elastic Block Store (Amazon EBS) and Amazon Simple Storage Service (Amazon S3) while maintaining the required performance and availability. For example, where performance requirements are lower, using Amazon EBS Throughput Optimized HDD (st1) storage typically costs half as much as the default General Purpose SSD (gp2) storage option.
- **Measure, monitor, and improve** – To ensure that you extract the full economic potential of the AWS Cloud at any scale, you want to:
  - Define and enforce cost allocation tagging.
  - Define metrics, set targets, and review at a reasonable cadence.
  - Enable teams to architect for cost via training, visualization of progress goals, and a balance of incentives.
  - Assign optimization responsibility to an individual or to a team.

# Cost Allocation Supports Ownership and Accountability

In a traditional data center, teams have to either architect within the limitations of existing investments or request the deployment of additional, capital-intensive resources that are often spread across many projects and organizations.

The cloud is fundamentally different. AWS provides much greater capacity, flexibility, and ease of deployment. However, with these benefits comes the requirement to manage costs. With AWS resource tagging, it is possible for you to have direct accountability and visibility into IT costs by team and application. The cloud also has the potential to decentralize IT procurement, which can lead to a lack of accountability if you do not set up the right visibility and control mechanisms within the cloud environment.

# Use Tagging to Organize Your Environment and Drive Accountability

Tagging your AWS resources lets you assign custom metadata to instances, images, and other resources. For example, you can categorize resources by owner, purpose, or environment, which helps you organize them and assign cost accountability.

## Define Mandatory Cost Tagging

An effective tagging strategy will give you improved visibility and monitoring, help you create accurate chargeback/showback models, and get more granular and precise insights into usage and spend by applications and teams. The following tag categories can help you achieve these goals:

- **Environment** – Distinguishes between development, test, and production infrastructure. Specifying an environment tag reduces analysis time, post-processing, and the need to maintain a separate mapping file of production versus non-production accounts.
- **Application ID** – Identifies resources that are related to a specific application for easy tracking of spend change and turn-off at the end of projects.
- **Automation Opt-In/Opt-Out** – Indicates whether a resource should be included in an automated activity such as starting, stopping, or resizing instances.
- **Cost Center/Business Unit** – Identifies the cost center or business unit associated with a resource, typically for cost allocation and tracking.
- **Owner** – Used to identify who is responsible for the resource. This is typically the technical owner. If needed, you can add a separate business owner tag. You can specify the owner as an email address. Using an email address supports automated notifications to both the technical and business owners as required (e.g., if the resource is a candidate for elasticity or right sizing).

## Enforce Quality of Tagging

Without enforcement, tagging quality will be low, and reporting will be manual, time-consuming, and subject to debate. There are two general types of tagging enforcement: soft and hard. Soft enforcement notifies users when they have not followed policies. Hard enforcement terminates resources that are not tagged to the company standard (usually within hours after they're launched). Mature organizations find hard enforcement of tagging to be the best way to ensure that quality tagging is maintained.

## Tagging Tools

The following tools can help you manage your tags:

- [AWS Tag Editor](#) – Finds resources with search criteria (including missing and misspelled tags) and allows you to edit tags via the AWS Management Console

- [AWS Config](#) – Identifies resources that do not comply to tagging policies
- [Capital One's Cloud Custodian](#) (open source) – Ensures tagging compliance and remediation



# AWS Account Structure Considerations

A well-defined AWS account structure that your teams agree on will help you understand and optimize costs. As with tagging, it is important that you implement a deliberate account strategy early on and allow it to evolve in response to changing needs. With multiple accounts, an organization can manage costs by:

- Grouping resources that require different payment instruments
- Providing groups with different levels of administrative control over AWS resources
- Better controlling Reserved Instances for specific workloads
- Identifying untaggable costs such as data transfer
- Using accounts associated with different business units or functional teams

For many organizations, a consolidated billing strategy where all AWS accounts are paid through one master account facilitates simplified payments, maximizes volume discounts, and enables the sharing of Reserved Instance benefits across linked accounts. [AWS Organizations](#) lets you create groups of AWS accounts with central policy management and consolidated billing. For more information on how to effectively structure accounts, see the [AWS Multiple Account Billing Strategy](#) article.

# Measurement and Management

Investment in advanced, granular metrics delivers value in larger organizations operating at scale. Advanced metrics can include:

- Measures that are normalized by spend, or that show how spend changes in response to changes in other metrics
- Measures and targets specific to business units
- Developer-focused or engineer-focused metrics that provide feedback on progress toward a goal (such as the percentage of workloads migrated to a target operating system)
- Metrics normalized by usage and business outcome, such as dollars per developer (This helps reduce data noise when dealing with a combination of static and dynamic costs.)
- Tagging based on architecture patterns to ensure that services conform to expected outcomes, such as expected peak CPU or memory utilization

# Enable Teams to Architect for Cost

Cost optimization is a pillar of the [Well-Architected Framework](#). It prevents developers and engineering teams from having to optimize workloads after the fact and when it is often too late and not economical to address issues built into the environment at early decision points.

Teams that are empowered to architect for cost can iterate quickly and learn over time so that best practices become embedded in day-to-day operations. The following practices can help teams architect for cost:

- Drive and foster transparency by creating visibility and by using tools to promote consistent reporting, measurement, and accountability.
- Drive the right type of behavior by creating positive incentives when the right actions are taken (e.g., email from management highlighting optimization wins).
- Establish control policies while maintaining agility (e.g., have a process to identify and address oversized resources, have an opt-out policy for non-production resources to switch off outside of work hours).

The following are some ideas that can help you drive cost-optimization behaviors:

- **Incentives** – These include visualization and gamification of metrics, as well as positive communication from leadership based on results. They encourage teams to understand that efficiency and frugality are valued and help developers and engineers consider the cost implications of their decisions. They also provide a way to discourage inefficiency.
- **Chargeback of costs to users** – Chargeback creates incentive for business users to care about IT efficiency. This results in treating IT as a resource that's used by and paid for by the business instead of as a cost center.
- **Removal of process barriers** – Occasionally, there are barriers that limit developers and engineers from undertaking optimization. For example, policies may be in place that require that any change made to the environment must go through a change review process. This will hamper initiatives to promote right sizing and elasticity. An amendment of such policies can streamline the optimization effort.
- **Agile working methods** – If design iteration cycles include cost as a metric, then your organization's ability to deliver the same or better outcomes at a lower cost will improve over time.
- **Training and onboarding** – Individuals typically solve problems using the tools and techniques they know. This can be addressed through training and onboarding that incorporate the latest practices to maximize efficiency (e.g., using serverless architectures, using Amazon CloudFront to reduce compute demand).

The following approaches can also be effective, but present risks to agility if not implemented with care:

- **Executive support/pressure** – Support for best practices is preferred over cost pressure due to their positive impact on staff satisfaction. Cost pressure can create an incentive to hide inefficiency and can lead to budget lockdown, resulting in loss of agility and ability to innovate.
- **Architectural review** – There is typically a reasonable balance between no architecture review (or optional review) and mandatory review. Excessive mandatory reviews can create bottlenecks. High-consequence and high-cost projects may require review with boundaries that are defined by each organization.
- **Orchestration control** – Approval workflows for projects and resources place agility and innovation at risk to protect finances and budget. One way to balance control and agility is to place fewer (or no)

cost controls on revenue-generating services. You can counterbalance this by having advanced metrics in place for these services.

# Create a Cloud Center of Excellence

A Cloud Center of Excellence (CCoE) team can take responsibility for cost optimization. Effective CCoE teams start small, develop an approach for implementing cloud technology at scale for your organization, and can become the fulcrum by which your organization transforms the way technology serves the business.

The following tenets are key guiding principles for creating a CCoE:

- The CCoE structure will evolve and change as the organization changes.
- Treat the cloud as your product and application team leaders as the customers you are enabling.
- Build company culture into everything you do.
- Organizational change management is central to business transformation. Use intentional and targeted organizational change management to change company culture and norms.
- Embrace a change-as-normal mindset. Changes in applications, IT systems, and business direction is expected.
- Operating model decisions will determine how people fill roles that achieve business outcomes.

# AWS Tools for Reporting and Cost Optimization

To help you track, report, and analyze costs over time, AWS provides several reporting and cost-optimization tools:

- [Cost Explorer](#) – See patterns in AWS spend over time, project future costs, identify areas that need further inquiry, observe Reserved Instance utilization, observe Reserved Instance coverage, and receive Reserved Instance recommendations.
- [AWS Trusted Advisor](#) – Get real-time identification of potential areas for optimization.
- [AWS Budgets](#) – Set custom budgets that trigger alerts when cost or usage exceed (or are forecasted to exceed) a budgeted amount. Budgets can be set based on tags and accounts as well as resource types.
- [Amazon CloudWatch](#) – Collect and track metrics, monitor log files, set alarms, and automatically react to changes in AWS resources.
- [AWS CloudTrail](#) – Log, continuously monitor, and retain account activity related to actions across AWS infrastructure at low cost.
- [Amazon S3 Analytics](#) – Automated analysis and visualization of Amazon S3 storage patterns to help you decide when to shift data to a different storage class.
- [Cost Optimization Monitor](#) – Automatically process detailed billing reports to get granular metrics that can be searched, analyzed, and visualized in a customizable dashboard.
- [EC2 Right Sizing](#) – Analyze EC2 instance utilization data and receive reporting recommendations for right sizing EC2 instances. The tool recommends instances that better match your usage.
- [AWS Cost and Usage Report](#) – Granular raw data files detailing your hourly AWS usage across accounts used for Do-It-Yourself (DIY) analysis (e.g., determining which S3 bucket is driving data transfer spend). The AWS Cost and Usage Report has dynamic columns that populate depending on the services you use.
- [Detailed Billing Report](#) – This report is similar to the AWS Cost and Usage Report, but it has static columns. This report is eventually deprecated. We recommend that you use the AWS Cost and Usage Report instead.

# Conclusion

This whitepaper has provided an overview of key pillars and tools for controlling costs. You can unlock the potential of cost optimization in the cloud through your initiatives. You will benefit significantly through interactions between cross-functional teams within your organization (for example, engineering and finance teams). AWS provides powerful tools to simplify these efforts, but organizational recognition and commitment to the process are critical to success.

# Resources

- [AWS Architecture Center](#)
- [AWS Whitepapers](#)
- [AWS Architecture Monthly](#)
- [AWS Architecture Blog](#)
- [This Is My Architecture videos](#)
- [AWS Answers](#)
- [AWS Documentation](#)



# Document Details

## Contributors

The following individuals and organizations contributed to this document:

- Amilcar Alfaro, Sr. Product Marketing Manager, AWS
- Erin Carlson, Marketing Manager, AWS
- Keith Jarrett, WW BD Lead - Cost Optimization, AWS Business Development

Date	First publication
March 2018	First publication

# AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the *AWS General Reference*.