Cost Optimization Pillar

AWS Well-Architected Framework

November 2016

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

The focus of this paper is the Cost Optimization pillar of the Amazon Web Services (AWS) Well-Architected Framework. It provides guidance to help customers apply best practices to optimize costs in AWS environments in all phases of development: design, delivery, and maintenance.

Introduction

At Amazon Web Services, we understand the value of educating our customers about architectural best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud. As part of this effort, we developed the AWS Well-Architected Framework, which helps you understand the pros and cons of decisions you make while building systems on AWS. We believe well-architected systems greatly increase the likelihood of business success.

The framework is based on five pillars:

- Security
- Reliability
- Performance Efficiency
- Cost Optimization
- Operational Excellence

This paper focuses on the Cost Optimization pillar and how to architect systems with the most effective use of services and resources, and at a minimal cost.

You'll learn how to apply the best practices of the Cost Optimization pillar to your solutions. Cost optimization can be challenging in traditional on-premises solutions because you have to guess capacity and complexity in attributing costs and procurement models. By adopting the practices in this paper you will be able to build architectures in which you can do the following:

- You can ensure that your costs move in line with demand
- You can use appropriate resource types to minimize costs
- You can analyze and attribute costs
- You can reduce costs over time

This paper is intended for those in technology roles, such as chief technology officers (CTOs), architects, developers, and operations team members. This paper does not provide implementation details or architectural patterns; however, it does include references to appropriate resources for this information.

Cost Optimization

Cost optimization is a continual process of refinement and improvement of a system over its entire lifecycle. From the initial design of your very first proof of concept to the ongoing operation of production workloads, you can adopt the practices in this paper to build and operate cost-aware systems that achieve business outcomes while minimizing costs, allowing your business to maximize its return on investment.

A cost-optimized system will fully utilize all resources, achieve an outcome at the lowest possible price point, and meet your functional requirements. This paper will provide in-depth guidance for designing your workload, selecting your services, configuring and operating the services, and applying cost optimization levers.

Design Principles

Keep these design principles in mind as we discuss best practices for cost optimization:

- **Adopt a consumption model:** Pay only for the computing resources you consume, and increase or decrease usage depending on business requirements, not with elaborate forecasting. For example, development and test environments are typically only used for eight hours a day during the work week. You can stop these resources when they are not in use for a potential cost savings of 75 percent (40 hours versus 168 hours).
- **Benefit from economies of scale:** By using cloud computing, you might achieve a lower variable cost than you could on your own because AWS can achieve higher economies of scale. Hundreds of thousands of customers are aggregated in the AWS Cloud, which translates into lower pay-as-you-go prices.
- **Stop spending money on data center operations:** AWS does the heavy lifting of racking, stacking, and powering servers, so you can focus on your customers and business projects rather than on IT infrastructure.
- **Analyze and attribute expenditure:** The cloud makes it easier to accurately identify the usage and cost of systems, which then allows transparent attribution of IT costs to individual business owners. This

helps measure return on investment (ROI) and gives system owners an opportunity to optimize their resources and reduce costs.

- **Use managed services to reduce cost of ownership:** In the cloud, managed services remove the operational burden of maintaining servers for tasks like sending email or managing databases. And because managed services operate at cloud scale, they can offer a lower cost per transaction or service.

Definition

Cost Optimization in the cloud is composed of four areas:

1. Cost-effective resources
2. Matching supply with demand
3. Expenditure awareness
4. Optimizing over time

As with the other pillars, there are trade-offs to consider, for example, whether to optimize for speed to market or for cost. In some cases, it is best to optimize for speed—going to market quickly, shipping new features, or simply meeting a deadline—rather than investing in upfront cost optimization. Design decisions are sometimes directed by haste rather than data, and the temptation always exists to overcompensate "just in case" rather than spend time benchmarking for the most cost-optimal deployment. This might lead to over-provisioned and under-optimized deployments. However, this is a reasonable choice when you need to "lift and shift" resources from your on-premises environment to the cloud and then optimize afterwards. Generally, the best practice is to avoid making hasty decisions without an explicit cost optimization strategy. The following sections provide techniques and best practices for both the initial and ongoing cost optimization of your deployment and environment.

Cost-Effective Resources

Using the appropriate services, resources, and configurations for your workloads is key to cost savings. In AWS there are a number of different approaches:

- Appropriately provisioned
- Right sizing
- Purchasing options: On Demand Instances, Spot Instances, and Reservations
- Geographic selection
- Managed services

You can use AWS Solutions Architects, AWS Reference Architectures, and AWS Partners to help you choose an architecture based on what you have learned, but data obtained through benchmarking or load testing will be required to optimize your architecture. After you have identified your architectural approach, you should use a data-driven process to refine your selection of resource types and configuration options. Use benchmarking and load testing to obtain this data. For best practices on how to do this, see the Review section of *The Performance Efficiency Pillar of the AWS Well-Architected Framework* whitepaper.

Appropriately Provisioned

In general, when you use managed services the vendor provisions and manages the underlying resources. There may be attributes of the service that can be set that allow you to ensure you have sufficient capacity to meet your need. You will want to ensure that you have appropriately set these attributes, to ensure excess capacity is kept to a minimum and performance is maintained for end users.

AWS provides access to modify the attributes of its managed services using the AWS Management Console or using the AWS APIs and SDKs so you can ensure that alignment with changing demand is met. For example, the number of nodes on an Amazon EMR or an Amazon RedShift cluster can be increased or decreased to scale out or in.

You can also pack multiple instances on an AWS resource to enable higher density usage. For example, you can provision multiple small databases on a single Amazon Relational Database Service (RDS) DB instance. Then, as usage grows you can migrate one of the databases to a dedicated RDS DB instance using a snapshot and restore process.

When provisioning systems on managed services you need to understand the requirements of adjusting the service capacity. These requirements are typically time, effort, and any impact to normal system operation. If the time to adjust is longer than you want, consider over-provisioning just a bit to allow for growth. The ongoing effort required to modify services can be reduced to virtually zero by using APIs and SDKs that are integrated with system and monitoring tools such as Amazon CloudWatch. AWS Trusted Advisor also monitors services such as Amazon Redshift and Amazon RDS for resource utilization and active connections. You may need to provision additional capacity during system maintenance windows.

Key AWS Services

The key AWS service that supports an appropriately provisioned approach is Amazon CloudWatch, which enables you to collect and track metrics.

Right Sizing

Right sizing is using the lowest cost resource that still meets the technical specifications of a specific workload. You can right size iteratively by adjusting the size of resources to optimize for costs. Right-sizing activities take into account all of the resources of a system and all of the attributes of each individual resource. Right sizing can be an iterative process, triggered by changes in usage patterns and external factors such as AWS price drops or new AWS resource types.

AWS provides APIs, SDKs, and features that allow resources to be modified as demands change. For example on Amazon Elastic Compute Cloud (EC2) a stop-and-start can be performed to allow a change of instance size or instance type. Or on Amazon Elastic Block Store (EBS) a volume can restored from a snapshot to a different volume type offering greater performance through increased IOPS and/or throughput.

Monitor resources and alarms to provide the data for right sizing. This monitoring can also provide triggers for the next execution cycle. Analysis of resources can be performed using Amazon CloudWatch and CloudWatch custom logs, where customized metrics from any component of your system can be captured and utilized to perform right-sizing analysis. For example, memory utilization or system connections can be monitored and analyzed.

Keep in mind three key considerations when you perform right-sizing exercises.

- The **monitoring must accurately reflect the end-user experience**. Select the correct granularity for the time period and thoughtfully choose the maximum or 99th percentile instead of the average.
- Select the correct granularity for the **time period** of analysis that is required to cover any system cycles. For example, if a two-week analysis is performed, you might be overlooking a monthly cycle of high utilization, which could lead to under-provisioning.
- Assess the **cost of modification** against the benefit of right sizing. For example, take into account business costs associated with making the change, such as manual system testing and verification.

Key AWS Services

The key AWS service that supports a right-sizing approach is Amazon CloudWatch, which allows you to set up monitoring so you can understand the utilization of your resources. The following service is also important:

- **AWS Trusted Advisor** performs analysis of resources and reports on any under- or over-utilized resources. For example, on Amazon EC2 Trusted Advisor performs analysis on CPU utilization and network traffic over a period of time. On Amazon EBS it analyses the input/output operations per second (IOPS) activity on a volume and provides a list of resources that could benefit from a right-sizing activity.

Purchasing Options

In AWS, there are a number of different purchasing models that allow you to acquire Amazon EC2 instances in a cost-effective way that suits your business needs. The following section describes each purchasing model:

- On-Demand Instances
- Spot Instances
- Reservations

On-Demand Instances

When you purchase EC2 instances on demand you pay a flat hourly rate, and you have no long-term commitments. You can increase or decrease the compute capacity of your instances based on the demands of your application. You only pay the specified hourly rate for the instances you use. On-Demand Instances are recommended for applications with short-term (such as a four-month project) workloads that spike periodically, or unpredictable workloads that can't be interrupted. On-Demand Instances are also suitable for workloads such as development and test environments, which run longer than a block of Spot Instances and are often shorter than the time required for application of a Reserved Instance.

Spot Instances

Spot Instances are useful for short-duration workloads (up to six hours) or for applications that can be run on a flexible schedule. Spot Instances allow you to bid on unused EC2 instances, and they can significantly lower your Amazon EC2 costs. You could also launch Spot Instances with a required duration (Spot blocks), which are not interrupted due to changes in the Spot price. Spot blocks could provide savings of up to 45 percent, and traditional Spot Instances could potentially save you up to 90 percent, compared to On-Demand rates. Spot Instances are ideal for use cases such as batch processing, scientific research, image or video processing, financial analysis, and testing. In addition to reducing the cost of running your applications, you can use Spot Instances to increase your computing scale and throughput for the same budget.

Spot Instances use a market-based approach where you set the maximum price you are willing to pay for an instance. This is your bid. When your bid meets or exceeds the current Spot price, your request is fulfilled and your instances will run until you choose to terminate them or until the Spot price increases above your bid price (whichever occurs first). Each instance type, size, and Availability Zone is a different market and prices and availability will vary.

Consider these three key things when you use Spot Instances:

- The instance type: If you can be flexible in both the family and size of the instance type this will provide the lowest possible cost and the lowest chance of interruption of the workload.

- Where the resource is located: Each Availability Zone is a different market. If you are flexible on where the workload runs you can seek the lowest cost and lowest chance of interruption.
- Design for continuity: Design the workload to accommodate a potential interruption by frequently saving any state and gracefully handling termination notices.

Reservations

AWS provides a number of ways for you to reduce your costs by reserving or committing to use a certain amount of resources. For example, you can use Reserved Capacity Pricing with Amazon Cloud Front or Reserved Instances with Amazon EC2. With Reserved Instances, you commit to a period of usage (one or three years) and you can save up to 75 percent over equivalent On-Demand hourly rates. When your Reserved Instances are assigned to a specified Availability Zone, they also provide you with capacity reservation (a standard Reserved Instance) so that you can launch the number of instances you need when you need them. For applications that have steady state or predictable usage, Reserved Instances can provide significant savings compared to using On-Demand Instances.

There are three types of Amazon EC2 Reserved Instances:

- Standard
- Regional Benefit
- Convertible

Standard Reserved Instances provide both a billing benefit and capacity reservation when the instance family, size, and Availability Zone are specified. Regional Benefit provides billing benefits for your Reserved Instances across any Availability Zone when you specify the instance family and size. However there is no capacity reservation. Convertible Reserved Instances are provided over a three-year commitment and allow conversion to different families, new pricing, different instance sizes, different platforms, or tenancy during the period. They also provide the billing benefit of a regional benefit Reserved Instance.

There are three payment options for Reserved Instances:

- No up-front payment– There is no up-front payment, and a moderate discount is provided on future hourly usage.
- Partial up-front payment– Part of the usage is paid up front, and a significant discount on future hourly usage is provided over the remaining period.
- Full up-front payment – Usage for the entire period is paid up front, and no other costs are incurred for the remainder of the term regardless of the number of hours used. You can apply any combination of these three purchasing options to the different workloads in your organization.

Key AWS Services

The key AWS service for purchasing payment options is Amazon EC2 Reserved Instances. Using Reserved Instances can save up to 75 percent compared to On-Demand Instance hourly rates. The following service is also important:

- **Spot Instances** can save you up to 90 percent compared to On-Demand Instances.

Resources

Refer to the following resources to learn more about AWS best practices for purchasing Reserved Instances and Spot Instances.

Documentation

- Spot Instances Best Practices
- Amazon EC2 Reserved Instances
- How Reserved Instances Work

Tools

- Spot Bid Advisor
- Spot Instance Pricing History
- How to Purchase Reserved Instances

Geographic Selection

Reduced latency can be a key factor in improving usage of your e-commerce or other websites. When architecting solutions a best practice is to seek to place computing resources closer to users to provide lower latency and strong data sovereignty. For global audiences you might need to use multiple locations to meet these needs. In addition, you want to select the geographic location that minimizes your costs.

AWS operates within different regions across the globe. Each AWS Region operates within local market conditions, and resource pricing can be different in each region. Choose a specific region to operate a component, or your entire solution, so you can run at the lowest possible price globally.

Using the AWS simple monthly calculator you can model a solution in different regions and compare costs. You can also use services such as Amazon CloudFront or AWS CodeDeploy to provision a proof of concept environment in different regions, run workloads through the environments, and analyze the exact and complete system costs in each region.

Key AWS Services

The AWS Region is a foundational concept for AWS. The AWS infrastructure is built arounds Regions and Availability Zones. When you choose a specific region to operate a component, or your entire solution, you have the opportunity to locate your solution with the same resources at the lowest possible price globally.

The following services and features are also important when you consider a geographic location:

- **Amazon Route 53**: When you use multiple AWS Regions, you can reduce latency for your users by serving their requests from the AWS Region for which network latency is lowest. Amazon Route 53 latency-based routing lets you use Domain Name System (DNS) to route user requests to the AWS Region that will give your users the fastest response.
- **Amazon CloudFront**: When you select your AWS Regions, you can use Amazon CloudFront to further reduce latency. Amazon CloudFront can be used to deliver your entire website, including dynamic, static,

streaming, and interactive content using a global network of edge locations. Requests for your content are automatically routed to the nearest edge location, so content is delivered with the best possible performance.

Resources

Refer to the following resources to learn more about AWS best practices for geographic selection.

Documentation

- AWS Global Infrastructure
- Regions and Availability Zones

Tools

- Amazon Web Services Simple Monthly Calculator

Managed Services

By using the managed services offered by AWS, you remove the operational burden of managing and maintaining the resources required to provide that service. Additionally, because managed services operate at cloud scale, they can offer a lower cost per transaction or service.

AWS provides managed services for databases, such as Amazon RDS and Amazon DynamoDB. By using these managed services you can reduce or remove much of your administrative and operational overhead, freeing you to work on value adding activities.

You can also use serverless or application-level services such as AWS Lambda, Amazon Simple Queue Service (SQS), Amazon Simple Notification Service (SNS), and Amazon Simple Email Service (SES). These various services remove the need for you to manage virtual servers for code execution, queuing services, and message delivery.

Use managed services to reduce the cost of managing infrastructure, but you should also consider the time savings that will allow your team to focus on

value-adding features. You should also consider the use of managed services if you are constrained by time pressures. For example, you might need to "lift and shift" your on-premises environment to the cloud now, and plan to optimize later. Finally it's worth exploring the savings you can realize by using managed services that remove or reduce license costs.

Key AWS Services

The key AWS services that support a managed approach are AWS database services, such as Amazon RDS and Amazon DynamoDB. These services reduce the cost of database capabilities, and also free up time for your developers and database administrators. The following managed services and features are also important:

- **AWS Lambda**: This serverless compute service executes your code without the base-level infrastructure costs. Charges are based on the compute time that you consume.
- **Amazon SQS, Amazon SNS, and Amazon SES**: Use these application-level services without paying for base-level infrastructure costs. With AWS pricing, you pay as you go and pay only for what you use.

Matching Supply and Demand

When supply matches demand optimal conditions are created for delivering the lowest costs for a system. However, there must be sufficient extra supply to allow for provisioning time and individual resource failures. Demand can be fixed or variable. Either way, plan for metrics and automation to ensure that management does not become a disproportionately large cost.

In AWS, you can use a number of different approaches to match supply with demand. The following sections describe how to use these approaches:

- Demand-based approach
- Buffer-based approach
- Time-based approach

Demand-Based Approach

Leveraging the elasticity of the cloud to meet demand as it changes, can provide significant cost savings. *Elasticity* refers to the virtually unlimited capacity of the cloud, where the vendor is responsible for capacity management and provisioning of resources. By taking advantage of APIs or service features you can programmatically vary the amount of cloud resources in your architecture dynamically. This allows you to scale components in your architecture, and automatically increase the number of resources during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.

Within AWS this is normally accomplished using Auto Scaling, which helps you to scale your Amazon EC2 instance capacity up or down automatically according to conditions you define. Auto Scaling is generally used with Elastic Load Balancing (ELB) to distribute incoming application traffic across multiple Amazon EC2 instances in an Auto Scaling group. Auto Scaling is triggered using scaling plans that include policies that define how to scale (manual, schedule, and demand) and the metrics and alarms to monitor in Amazon CloudWatch. CloudWatch metrics are used to trigger the scaling event. You can use standard Amazon EC2 metrics such as CPU utilization, ELB observed request/response latency, and even custom metrics which might originate from application code on your EC2 instances.

Auto Scaling can use custom code or metrics to trigger auto scaling in response to a business event.

When architecting with a demand-based approach keep in mind two key considerations: First, understand how quickly you need to provision new resources. Second, understand that the size of margin between supply and demand will shift. You need to be ready to cope with the rate of change in demand and also be ready to resource failures.

You can optimize the speed of provisioning by reducing the startup and configuration tasks your instances run at boot. This is often done using a pre-baked Amazon Machine Image (AMI). Note that this optimization will be at the expense of configurability of these instances. You need to balance speed versus configurability based on your needs. The margin between supply and demand can be very wide when you start development, and can be reduced through

experimentation, load testing, and confidence building as you move into test and production.

Key AWS Services

The key AWS service that supports a demand-based approach is Auto Scaling, which allows you to add or remove resources to match demand without overspending. The following services and features are also important:

- **Amazon CloudWatch**: Use Amazon CloudWatch to collect and track metrics. Your architecture can Use CloudWatch to be aware of demand and utilization, incorporate custom metrics, and trigger and respond to thresholds and events.
- **Elastic Load Balancing**: Use to automatically distribute incoming application traffic across multiple Amazon EC2 instances in the cloud, thus allowing you to scale horizontally.

Buffer-Based Approach

A buffer-based approach to matching supply and demand uses a queue to accept messages (units of work) from producers. For resiliency the queue should use durable storage. A *buffer* is a mechanism to ensure that applications can communicate with each other when they are running at different rates over time. Messages can then be read by consumers, which allows the messages to run at the rate that meets the consumers' business requirements. By using a buffer-based approach, you can decouple the throughput rate of producers from that of consumers. You don't have to worry about producers having to deal with data durability and *backpressure* (where producers slow down because their consumer is running slowly).

When you have a workload that generates significant write load that doesn't need to be processed immediately, you can use a buffer to smooth out demands on consumers.

On AWS, you can choose from multiple services to implement a buffering approach. Amazon Simple Queue Service (SQS) provides a queue that allows a single consumer to read individual messages. Amazon Kinesis provides a stream that allows many consumers to read the same messages.

An Amazon SQS *queue* is a reliable, scalable, and fully managed repository for messages that are awaiting processing. Amazon SQS enables applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component. For this approach, a queue is created that producers post messages to, and which resides at a well-known address.

To reduce cost, producers post messages using Amazon SQS batch API actions. Consumers read messages from the well-known queue using a fixed-sized Auto Scaling group of EC2 instances to cope with instance failures. *Long polling* lets you retrieve messages from your Amazon SQS queue as soon as they become available. Use long polling to reduce the cost of retrieving messages. Amazon SQS uses the *message visibility* feature to hide messages that have been read. Message visibility reduces the risk that you might process the same message twice, though you should be aware that by default Amazon SQS *at-least-once delivery* means that you can receive a message more than once.

You can use Amazon SQS first in/first out (FIFO) queues if you need exactly-once processing. Message visibility allows for messages that have not been processed to reappear (for example, in the case of an instance failure). For long-running tasks you can extend the visibility time-out, and your application will need to delete messages after they have been processed.

An alternative approach is to use Amazon Kinesis to provide buffering. It differs from Amazon SQS in that it allows multiple *consumers* to read the same message at any one time. However, a single message will be read using the Kinesis Client Library, or AWS Lambda, and will be delivered to one-and-only-one consumer for an *Application*, which is a name provided by the consumer when it connects to the Stream. Different *Applications* can all consume the same messages concurrently, providing a “Publish and Subscribe” model.

When architecting with a buffer-based approach keep in mind two key considerations: First, what is the acceptable delay between producing the work and consuming the work? Second, how do you plan to handle duplicate requests for work?

To optimize the speed with which work items are consumed by more consumers, you can use Amazon EC2 Spot Instances. Using Spot Instances, you can bid on spare Amazon EC2 computing capacity. To handle duplicate

messages with Amazon SQS we recommend using Amazon SQS (FIFO) queues, which provide exactly-once processing.

For AWS Kinesis you should consider provisioning a table in Amazon DynamoDB that can track the work items that have already been successfully processed. An alternative to deduplication is to implement idempotent processing, this should be used where you require higher throughput processing. *Idempotence* is an application logic pattern that allows a consumer to process a message multiple times, but when a message is subsequently processed it has no effect on downstream systems or storage.

Key AWS Services

The key AWS services that support a buffer-based approach are Amazon SQS and Amazon Kinesis, which make it simple and cost-effective to decouple the components of a cloud application. The following services and features are also important:

- **Amazon EC2 Spot Instances:** Enable you to bid on spare Amazon EC2 computing capacity that can process work items at a low cost.
- **AWS Lambda**: Provides a serverless approach to processing messages, removing the fixed costs of running EC2 instances.

Time-Based Approach

A time-based approach aligns resource capacity to demand that is predictable or well defined by time. This approach is typically not dependent upon utilization levels of the resources. A time-based approach ensures that resources are available at the specific time they are required, and can be provided without any delays due to start-up procedures and system or consistency checks. Using a time-based approach, you can provide additional resources or increase capacity during busy periods.

Within AWS you can use scheduled scaling in Auto Scaling to put a time-based approach in place. Systems can be scheduled to scale out or in at defined times, such as the start of business hours, thus ensuring that resources are available when users arrive.

You can leverage the AWS APIs and AWS CloudFormation to automatically provision and decommission entire environments as you need them. This usage

is well suited for development or test environments that run only in defined business hours or periods of time.

You can use APIs to scale resources up or down within an environment. For example, you could scale up a production system by changing the instance size or class. This can be achieved by stopping and starting the instance and selecting the different instance size or class. This technique can also be applied to other resources, such as Amazon Elastic Block Store (EBS) volumes. Create a snapshot and restore the volume with increased input/output operations per second (IOPS) or a different volume type.

When architecting with a time-based approach keep in mind two key considerations: first, how consistent is the usage pattern? Second, what is the impact if the pattern changes? You can increase the accuracy of predictions by monitoring your systems, and by using business intelligence. If you see significant changes in the usage pattern, then you can adjust the times to ensure that coverage is provided.

Key AWS Services

The key AWS service that supports a time-based approach is Auto Scaling, which allows you to add or remove resources to match demand without overspending. The following service is also important:

- **AWS CloudFormation**: Provides templates you can use to create AWS resources and provision them in an orderly and predictable fashion. This can be useful for creating short-lived environments, such as test environments.

Expenditure Awareness

An awareness of your business cost drivers is critical to managing your business expenditure effectively and identifying cost-reduction opportunities. Businesses typically operate multiple systems run by multiple teams. These teams can be in different business units, each with its own revenue stream. The capability to attribute resource costs to the individual business or product owners drives efficient usage behavior and helps reduce waste. Accurate cost attribution allows you to understand which business units and products are truly profitable, and allows you to make more informed decisions about where to allocate resources within your business.

Consider taking a multi-faceted approach to becoming aware of your expenditures. Your team needs to gather data, analyze, and then report. These are the key considerations:

- Stakeholders
- Visibility and controls
- Cost attribution
- Tagging
- Entity lifecycle tracking

Stakeholders

Relevant stakeholders within your business need to be involved in expenditure discussions at all stages of your cloud journey. The following are typical stakeholders for engagement and their roles:

- Financial: CFO/Financial controllers must understand the cloud model of consumption, purchasing options, and the monthly billing process and accompanying data (detailed billing and/or usage files). Because there are fundamental differences between the cloud (such as pay as you go pricing, and detailed billing and usage information) and an on-premises operation it is essential that the financial team can continue its functions with this new mode of operation and information.
- Management: The management team must understand the cloud business model so it can provide direction to the business units and the whole company. This cloud knowledge is critical when management needs to forecast for growth and system usage, and also when management needs to assess different purchasing options, such as Reservations.
- Tech Leads: Tech leads must translate external business, usage, or cost factors from the financial and management stakeholders into system attributes or adjustments. This allows the system to be implemented to achieve the desired goals of the business.
- Third parties: If your business uses third parties, ensure that they are aligned to your financial goals and can demonstrate the alignment

through their engagement models. Typically third parties will contribute to reporting and analysis of any systems that they manage, and they will provide cost analysis of any systems that they design.

Visibility and Controls

By being able to understand the breakdown of your costs, predict future costs, and planning to react before you spend more than you expected, you can maintain control of your cost base.

AWS provides a suite of reports and tools for you to estimate, monitor, plan, report on, and analyze your AWS spend. The AWS Billing and Cost Management service enables you to do the following:

- Estimate and forecast your AWS costs
- Receive alerts if your costs exceed a threshold that you set
- Assess your biggest investments in AWS resources
- Analyze your spend and usage data

Consider using the free Cost Explorer tool to visualize and analyze your costs. This tool allows you to graphically view your costs over a year, and forecast how much you are likely to spend for the coming months with a more than eighty percent confidence interval. You can use Cost Explorer to see patterns in how much you spend on AWS resources over time, identify areas that need further inquiry, and see trends that you can use to understand your costs.

The AWS Billing and Cost Management service provides Budgets you can use to define a monthly budget for your AWS costs, at an aggregate cost level (that is, all costs) or at a more granular level where you can include only specific dimensions such as linked accounts, services, tags, or Availability Zones. You can also attach email notifications to your budgets, which will trigger when your current or forecasted costs exceed a percentage threshold that you define.

You can use Amazon CloudWatch alerts and notifications to create billing alerts that notify you when usage of your services exceeds financial thresholds that you define. You can define thresholds at an overall account level, or you can define discrete dimensions such as services, linked accounts, or both linked account and service.

The Detailed Billing Report with resources and tags (DBR-rt) and the Cost and Usage Reports provide the most granular level of reporting. They provide hourly usages and charges for all chargeable AWS services. These reports offer insights into both the cost and usage of your systems, on an hour-by-hour basis. To analyze and report on this data you can have Amazon Redshift ingest your billing data, use a third-party application, build an application that uses your billing data, and import your billing data into your existing financial systems.

The Billing and Cost Management service is integrated with the AWS Identity and Access Management (IAM) service. You use the IAM service in conjunction with Billing and Cost Management to control access to your financial data and to the AWS tools in the billing console.

Cost Attribution

You can use cost attribution to drive cost management by assigning costs to the parts of your organization, such as departments, business units, products, or internal teams.

To organize and manage your AWS costs consider using account structuring and tagging. Either of these or a combination of the two can provide the cost attribution required for your business

Account structuring. AWS has a one-parent-to-many-children account structure that is commonly known as a payer (parent) account-linked (child) account. You can implement either a single AWS account or a multiple AWS account structure. There is no one-size-fits-all answer for how many AWS accounts you should have. You should first assess your current and future operational and cost models to ensure that the structure of your AWS accounts reflects that of your organization. Some companies will want to create more than one AWS account for business reasons, for example, in the following situations:

- Administrative and/or fiscal and billing isolation is required between business units or cost centers or specific workloads.
- AWS service limits are set to be specific to particular workloads.
- Specific instance reservations are required for certain workloads (such as high availability and disaster recovery).

Consolidated billing is used to create the construct between one or more linked accounts and the payer account. Linked accounts allow you to isolate and distinguish your usage and billing by groups defined by you. A common practice is to have separate linked accounts for each business unit (such as finance, marketing, and sales), or for each environment-lifecycle (such as development, test, and production), or for each project (project a, b, and c), and then aggregate these linked accounts using consolidated billing. Consolidated billing allows you to consolidate payment for multiple AWS accounts under a single payer account, while still providing visibility for each linked account's activity. Because costs and usage are aggregated at the payer account, this allows you to maximize your service volume discounts, Reserved Instances volume discounts, and Reserved Instances benefits.

Tagging

Tags allow you to overlay business and organizational information onto your billing and usage data. This is useful in helping you categorize and track your costs by meaningful, relevant business information. You can apply tags that represent business categories (such as cost centers, application names, projects, or owners) to organize your costs across multiple services and teams.

When you apply tags to your AWS resources (such as Amazon EC2 instances or Amazon Simple Storage Service (S3) buckets) and activate the tags, AWS adds this information to the detailed billing report and cost allocation reports. There is also a cost allocation report which contains a billing summary with this tagging information.

Assigning tags to resources allows higher levels of automation and ease of management. You can execute management tasks at scale by listing resources with specific tags, then executing the appropriate actions. For example you can list all resources with the tag and value of `environment:test`, then for each of the resources either delete or terminate them. This is useful to automate shutdown or removal of a test environment at the end of the working day. Running reports on tagged and more importantly untagged resources allows greater compliance with internal cost management policies.

Creating and implementing an AWS tagging standard across your organization's accounts will enable you to manage and govern your AWS environments in a consistent and uniform manner.

Entity Lifecycle Tracking

When you manage a list of projects, employees, and technology resources over time you will be able to identify resources that are no longer being used or orphaned projects that no longer have an owner.

AWS provides a number of services you can use for entity lifecycle tracking, such as AWS Config, which provides a detailed inventory of your AWS resources and configuration, and continuously records configuration changes. AWS CloudTrail and Amazon CloudWatch allow you to establish a record of resource lifecycle events. IAM allows you to manage groups, users, and roles as well as establish trust with existing identity providers using federation already tied to your organization's workforce source of record. Use federation to reduce the need to create users in IAM, and still use the existing identities, credentials, and role-based access you have already established in your organization.

You can integrate with your existing project or asset management systems to keep track of active projects and products with your organization. Combining your current system with the rich set of events and metrics provided by AWS will allow you to build a view of significant lifecycle events and proactively manage resources to reduce unnecessary costs.

Key AWS Services

The key AWS service for expenditure awareness is AWS Billing and Cost Management. The following services and features are also important:

- **Cost Explorer**: Use this tool to visualize and analyze your costs.
- **Tagging**: Use tags to overlay business and organizational information onto your billing and usage data.
- **Amazon CloudWatch alerts**: Create billing alerts that notify you when usage of your services exceeds financial thresholds you define.

Resources

Refer to the following resources to learn more about AWS best practices for expenditure awareness.

Documentation

- AWS Tagging Strategies
- AWS Billing and Cost Management
- Amazon CloudWatch Alarms

Tools

- AWS Cost Explorer

Optimizing Over Time

In AWS, you can use a number of different approaches to optimize over time. The following sections describe how to use these approaches:

- Measure, monitor, and improve
- Staying ever green (move to the newest services, features, and instance types)

Measure, Monitor, and Improve

When you measure and monitor your users and applications, and combine the data you collect with data from AWS platform monitoring, you can perform a gap analysis that tells you how closely aligned your system utilization is to your requirements. By working continually to minimize this utilization gap you can ensure that your systems are cost effective. There are four key ways to set this up:

- Establish a cost optimization function
- Establish goals and metrics
- Gather insight and perform analysis
- Report and validate

For best practices for metric collection and setting up monitoring, see the Monitoring section of *The Reliability Pillar of the AWS Well-Architected Framework* whitepaper.

Establish a Cost Optimization Function

Consider establishing a function within your organization with the responsibility and accountability of cost optimization for the entire

organization. This function can be performed by an existing team, such as a Cloud Center of Excellence, or you can create a new team of key stakeholders from business units across the organization. This function will coordinate and manage all aspects of cost optimization, from your technical systems to your people and processes. Investing in the correct level of AWS Support can provide assistance with execution of this function. AWS Enterprise Support will provide a Technical Account Manager (TAM) who specializes in optimization. The TAM has the technical depth and tools to provide reporting and analysis for your organization.

Establish Goals and Metrics

Establish goals and metrics that your organization can use to measure its progress. These will be at both the macro and micro level and must be aligned to the end users of the platforms and systems. At a macro level the goals will be across accounts and organizational units, at the micro level they will be more aligned to individual systems.

The following are some examples of goals for organizations:

- On-Demand EC2 Elasticity: The percentage of your On-Demand EC2 instances that are turned on and off every day. This should be as close to 100 percent as is feasible, and be in the range of 80-100 percent. To measure, take the number of instances started during the day and divide it by the maximum number of instances running at any time.
- Overall Elasticity: The percentage of On-Demand and Reserved Instances that are turned on and off every day. This should be as close to 100 percent as is feasible and be in the range of 80-100 percent. However, if you operate systems that are in use 24 hours 7 days a week the range will be broader. If your systems peak from 20 percent to 100 percent of usage during the day, then elasticity should be driving towards 80 percent.
- Baseload Reserved Instance coverage: The number of “always on” instances that are running as reserved capacity. Organizations should aim for 80-90 percent coverage.

This list can be used as a guide to develop goals for other resources, such as Amazon EBS volume utilization, Amazon RDS storage utilization, or Amazon ElastiCache memory utilization.

Gather Insight and Perform Analysis

Tooling allows you to measure and monitor your platform usage, and provides insight and analysis into your business. The most suitable tools will be those that tell you the most about what your customers are doing and how your system components respond to that usage.

AWS provides the following tools to measure and monitor usage: The Billing and Cost Management Dashboard (including Cost Explorer and Budgets), Amazon CloudWatch, and AWS Trusted Advisor.

Report and Validate

Reports using the tooling and analysis described above should be reviewed regularly to ensure alignment with and progress toward your defined goals. It is also essential to validate any implemented cost measures, for example savings due to reserved capacity, and adjust them if required.

Staying Ever Green

As AWS releases new services and features, it is a best practice to review your existing architectural decisions to ensure that they remain cost effective and stay ever green. As your requirements change, be aggressive in decommissioning resources and entire services, or systems that you no longer require.

Managed services from AWS can often significantly optimize a solution, so it is good to be aware of new managed services as they become available. For example, running an Amazon RDS database can be cheaper than running your own database on Amazon EC2.

Key AWS Services

The key AWS features that support optimizing over time are the AWS Blog and the *What's New* section on the AWS website. These are the places you can visit regularly to learn about newly launched features and services.

The following service is also important for optimizing over time:

- **AWS Trusted Advisor**: Inspects your AWS environment and finds opportunities to save money by eliminating unused or idle resources or committing to Reserved Instance capacity.

Resources

Refer to the following resources to learn more about AWS best practices for optimizing over time.

Documentation

- AWS Blog
- What's New at AWS
- AWS Trusted Advisor

Tools

- Billing & Cost Management Dashboard

Conclusion

Cost optimization is an ongoing effort. From the first review of a pilot or test workload to deploying a mature production infrastructure on AWS, you should regularly review your architectural approach and component selection. This effort is easier thanks to the programmatic functions and AWS features and services discussed in this paper.

AWS strives to help you minimize cost while you build highly resilient, responsive, and adaptive deployments. To make your deployment truly cost-optimized, take advantage of the tools, techniques, and best practices discussed in this paper.

Contributors

The following individuals and organizations contributed to this document:

- Philip Fitzsimons, Sr Manager Well-Architected, Amazon Web Services
- Nathan Besh, Enterprise Support Manager, Amazon Web Services
- PT Ng, Commercial Architect, Amazon Web Services

- Arthur Basbaum, Business Developer Manager, Amazon Web Services
- Jarman Hauser, Commercial Architect, Amazon Web Services

Further Reading

For additional help, please consult the following sources:

- AWS Well-Architected Framework