AUREA
CLOUDFIX

The #1 challenge to AWS cost control:

How to adopt an ongoing approach to cost optimization

WHITEPAPER

Executive summary

As cloud-native organizations approach optimizing costs on AWS, they can encounter a very challenging roadblock: They've done it already.

At least, that's the conception at the leadership level. But just as AWS has evolved over time, so has the complexity of the services AWS offers and the associated costs. Just because a fix was made in the past to optimize AWS costs doesn't mean it's kept pace with the changes.

Given the evolving nature of AWS services, it's simply not possible to address costs with a one-and-done solution, such as scripting a program to track usage. If you're going to successfully optimize costs on AWS, it has to be part of an ongoing practice. To do that, leadership has to go beyond the one-and-done thinking that assumes turning on a solution for cost optimization once will save money perpetually. Not only are there too many changes being made to their environment day after day and week after week for that to work, AWS is also constantly updating its services.

In this white paper, we will show why organizations have to move away from one-and-done thinking about cost optimization and embrace a multidimensional approach that takes into account the ever-evolving nature of AWS.

Table of contents

The evolving nature of AWS

There's a clear reason why AWS continues to dominate the cloud market: AWS didn't just make the cloud popular, AWS invented it.

That legacy remains a fundamental reason why organizations continue to make AWS their sole cloud service provider, driving the company's market dominance. **By the end of Q4 2021**, AWS had a 33% share of all cloud spending, dwarfing Azure's 22% share and Google Cloud's 9%.

But what many users may not understand is that AWS hasn't always been the all-in-one service provider that it is today. In many ways, the missteps that organizations make when first using AWS can be traced back to misunderstanding the difference between AWS as it once was and the AWS of today.

AWS Then

When AWS officially launched in 2006, its premier service was Amazon Simple Storage Service (Amazon S3), offering scalable object storage that was durable and inexpensive. This largely generated the conception of the cloud as merely a data storehouse.

For organizations with existing infrastructure, the cloud offered a virtualized version of their infrastructure to house and run their applications and data, with a pay-as-you-go model replacing big capital expenditures for appliances. Many companies with legacy operations still see the cloud through that basic lens, even though there are now more advanced services available on AWS. For these organizations, just being in the cloud represents cost savings—which is something of a misconception in itself.

For cloud-native organizations, their relationship with AWS has always been fundamentally different, as there was never a need to replace on-premises infrastructure. The cloud was a cradle, and its services provided ways to operate that would otherwise have been out of reach or cost-prohibitive. Rather than just seeing the cloud as a cheaper alternative to on-premises hardware, what the cloud offered was viability.

In both cases—cloud-native and cloud-adopter—there was a lot to be desired in the early days of AWS. Not everything was available at the click of a button the way it is today. Consider the fact that the AWS Management Console wasn't introduced until 2009; until then, there wasn't a graphical interface for users to control their resources on the service. Since the drive to optimize costs at legacy organizations was achieved simply by using the cloud, there wasn't as much motivation behind it. At cloud-native operations, however, there was a higher chance that users would try more innovative methods to optimize costs, such as developing in-house scripts to monitor usage.

That's what the company that would become CloudFix did. We wrote a program that could act as a graphical interface for deploying AWS instances. This console allowed users to log into AWS, provision EC2 instances, and define working hours. When working hours ended, the instances would automatically spin down, saving on costs. However, our team first had to build multitenancy into the program to be able to safely shut down those instances and make sure data was not lost in the process. It was complex, but it was possible.

This project was essentially inventing our own wheel before the wheel existed. Eventually, AWS would provide its own console. They have since added many more services, which brings us to the AWS that we have today: a company and suite of services that continue to grow in size and complexity. Inventing that wheel would be much harder today.

AWS Now

Today, AWS offers a wide range of services that go far beyond the initial IaaS workloads it presented at its launch. Everything-as-a-service is the new mode of operation at AWS, which now has pay-as-you-go and managed service options for machine learning (Amazon SageMaker), service meshing (AWS App Mesh), analytics (Amazon EMR), Kubernetes (Amazon EKS), virtual desktops (Amazon WorkSpaces), internet of things (AWS IoT), and much more.

While this is by no means an exhaustive list, the message is clear: As AWS has grown, so has the number, complexity, and interconnectedness of its services. And as such, the rules for operating in AWS that may have worked in the past will most likely not work today, and definitely won't work in the future—including one-and-done cost optimization.

Trying to control costs with that simple Python script that was written in-house may still work if a company only has a small AWS footprint and limited resources to invest in better solutions. But for operations that are fully vested in the AWS platform and make use of multiple services, running a script that can track costs simply will not suffice. Given the pace of AWS development, as soon as such a script begins to work, it would likely need to be rewritten and run again as existing services are updated and new ones are adopted.

Even if the organization did decide to optimize costs by creating a new in-house script, the growing complexity of AWS services means that any actions the script could inform users to take may have far-reaching effects and unforeseen impacts on other services. Developer teams would have to thoroughly vet any changes before they're put into effect, meaning they would have even more to worry about on top of their full work schedules. The one-and-done solution won't work here because it's too goal-oriented.

In the cloud today, the goalposts are always moving.

The result is that many companies adopting AWS today never attempt to solve their problems when it comes to cost optimization. They seem simply too big to solve. Organizations that have been using AWS for a while find themselves overwhelmed with changes. Services therefore go over-provisioned, underutilized, and overcharged.

AWS in the Past	*AWS Today*
— Traditional on-premises deployments	— Cloud-native
— Reasonable fixes implemented by users	— Service complexity and scale limits user fixes
— First-gen services: storage, compute, etc.	— Everything as a service

AWS Timeline

Early 2000s

Amazon comes up with the idea to help port its leading e-commerce system for other retail users; this led to the first effort to design web services.

2004

Amazon launches its first web service, Simple Queue Service (SQS); it has limited regional use and is very much a test case.

2006

The official launch of Amazon Web Services takes place with the release of Simple Storage Service (Amazon S3); many other first-gen services follow.

2009

This marks the debut of the AWS Management Console; before this point, users had to work with AWS resources programmatically.

2010

The entirety of Amazon.com's retail business operates on AWS; the tail has begun to wag the dog.

2012

First re:Invent takes place.

2014

AWS Partner Network launches to help companies using AWS better master the technology and onboarding.

2018

AWS begins to offer machine learning and artificial intelligence services, representing a huge shift in services.

2022

Despite more competition, AWS continues to dominate cloud market share, accounting for 33% of all cloud spending.

The three biggest obstacles

to AWS cost optimization

Given this shifting landscape, modern businesses must address the cost dilemma. Changing things at AWS isn’t going to be possible, but you can identify elements within your company that are hindering cost-savings opportunities and do something about them.

There are three main organizational issues that impact how costs are addressed on AWS: **lack of experience**, **one-and-done thinking**, and the **growing complexity** of AWS itself.

Obstacle #1:

Surface knowledge of AWS

> ***There's no single AWS user, and every deployment is different. While some companies are using AWS to port their existing on-premises technology, like NetApp or VMware, others are fully vested in AWS and leveraging hundreds of native services. Even the use of a single service such as EC2 is actually broader than it seems: There are so many different instance types and SLAs involved; there's a much wider range of services than what appears on the surface.***

No matter the AWS footprint, if a user wants to optimize costs, they typically turn to a visualization tool first. It's a good move; visualization is one of the basic steps that need to be taken to realize savings. However, interpreting the results and understanding how they can be translated into savings is a different matter altogether. This is where a surface-level understanding of AWS services will impact your ability to optimize costs.

Consider what happens when a cost-minded C-level director takes a look at an AWS visualization tool and sees that EC2 instances are only being utilized at 10%. The obvious answer this director would reach is to adjust the sizing. This will address costs and optimize for usage. Problem solved, right? What that user doesn't realize is that the change will set off a chain reaction that will affect disk performance, IOPS, memory, the data in use, and the network.

Users who aren't seeing the full picture of the affected AWS services can't get that knowledge from a simple data visualization tool. Knowing the right first step doesn't necessarily mean you'll reach the right result.

With a platform as popular as AWS, this lack of experience is surprisingly widespread, and the wrong move can have lasting business repercussions. Making any decision about a deployment without a real understanding of how AWS works and a decision framework in place is simply not safe.

Visualization tools can simplify how users see their AWS operations, and the console makes it easy to use. However, just as you wouldn't try to fix your microwave without a background as an electrician or in industrial design, it's a dangerous situation to change anything about your AWS usage without a complete understanding of how it will affect your operations. Data visualization tools simply can't give you that level of insight.

This lack of experience leads directly to the next obstacle to cloud cost optimization: one-and-done thinking.

Obstacle #2:

Goal-focused thinking

Goal-focused thinking is goal-oriented: It wants to spend less now. If this person could hit a button to cut costs, they would do it. While that goal is valid, the days of single-use solutions on AWS are a thing of the past.

In the end-results worldview of the one-and-done thinker, it would make sense to create that Python script in-house to control AWS usage. But AWS isn't standing still for anyone. The nature of AWS usage is too multidimensional for the one-and-done thinker to solve with a single solution. Real cost savings are only possible when taking a multidimensional view of the AWS surface area and constructing a decision-making framework composed of multiple parts to properly understand it—and building that kind of experience with AWS is only possible if you've already tried and failed, tried again, and built on that experience.

Obstacle #3:

Growing complexity

In the early days of AWS, building custom solutions was a much more serviceable way to achieve goals that the platform didn't currently specialize in.

For example, at smaller organizations, it was perfectly sensible to write a version of the AWS console before it existed, like we did at CloudFix.

Such solutions could work, but not for very long. As companies grow, so does AWS. And as the nature of the services offered by the cloud has grown more complex, so too is the initial deployment that companies adopt in the cloud-native approach. A simple Python script won't be enough to solve all of your problems anymore.

Consider what happened at the company that would eventually grow into CloudFix. While the solution worked, it took a considerable amount of effort to build the multitenancy around it to make sure that when it was shut down, all the data wouldn't be lost.

That was a lot of work, and it was done when AWS was still in its infancy. Today, writing that script would be a far more intensive undertaking, one that would detract from the main goals of any business and quickly become out of date given the rapid pace of innovation at AWS. For a small operation that doesn't utilize a large AWS surface area, that option might still be feasible, but for growing companies that are reaching a greater level of service spend and interconnectivity, another type of solution is in order.

Finding the Right solutions

Between the growing complexity of services and the inexperience of users, cost optimization can seem impossible.

Some of these problems are evident even at AWS, which has rolled out solutions to try to help users address the way their resources are being consumed:

— **AWS Athena** gives users a way to dive deeper into the results that are produced by various AWS services, which can lead to finding places to adjust for costs.

— **Amazon CloudWatch Metrics** can provide insight into the metrics of different AWS services in use. CloudWatch data can directly show which resources are being underutilized, which is a prime area for optimization. This data also lets users see how any potential changes will affect overall system health.

— **AWS Compute Optimizer** reports automatically on how compute via EC2 instances is being utilized, giving users tips on where to optimize.

— **AWS Cost Explorer** is a free-to-use feature provided as part of the AWS console. It provides visualizations of data usage, costs, APIs in use, and more. This, however, may not be the most useful tool since there isn't visibility into important services such as your compute and storage resources.

— **AWS Cost and Usage Reports** are a bit more effective than AWS Cost Explorer at pinpointing your actual cost factors. Cost and usage reports are produced as often as three times a day, with details on usage across all of your AWS services. But here is where complexity becomes an issue: Once your AWS usage reaches a certain scale, these reports will become too detailed to manually gain any insights from them.

— **Amazon S3 Storage Lens** gives S3 users a free way to analyze their bucket usage from a single pane of glass. With the insights users gain from these dashboards, S3 usage can be optimized. Note that this isn't a part of the AWS console, but a completely independent dashboard, which in itself adds to overhead, as there is now an additional window to juggle.

— **AWS Trusted Advisor** scans your AWS deployment and finds recommended places to make adjustments. For investigating costs, the service has a specific Cost Optimizations tab, which pinpoints underused resources. This isn't a free service since you will need to have an AWS Support account set up first and not every option for that service is free.

This panoply of services makes it clear that addressing the issue of costs for a highly complex AWS infrastructure is itself overly complicated.

In addition, the visibility that these solutions provide doesn't necessarily translate into effective actions. Juggling between these different tools, coordinating the different information they produce, and acting upon it isn't a scalable solution. As part of a manual process, you can keep your eye on one metric while losing sight of all the rest. But once you try to make changes, there isn't a clear way of knowing that you aren't going to affect operations throughout your deployment, which will only compound your problems.

Find and fix

AWS-recommended changes with CloudFix

The only way to track changes and usage and make actionable adjustments is with a solution automated to find and implement changes. Enter CloudFix.

With CloudFix, your AWS resources are easily monitored and your fixes clearly reported, taking a full view of every level of your deployment. One of the reasons that CloudFix works is that its information is derived directly from the AWS services with which it integrates.

WITH CLOUDFIX, YOU CAN:

— **Automatically fix cost issues in your account.**
Whereas data visualization tools simply find the problems, CloudFix actively helps you solve them.

— **Run optimizations with zero downtime.**
All of the changes that CloudFix finds to optimize costs are tested against your entire deployment, making sure nothing breaks as problems are resolved.

— **Constantly monitor and respond.**
CloudFix gives you a way to constantly monitor changes to your environment and implement fixes to adjust.

— **Stay in control.**
With CloudFix, users are in complete control, getting a full say in how fixes are put in place and integrated with AWS Change Manager.

Say goodbye to **one-and-done** *cost control solutions*

A one-and-done approach to AWS cost optimization is no longer viable. Developers' time is far better spent developing new products and services than inventing ways to make AWS more efficient.

CloudFix gives organizations a way to think about AWS from a multidimensional perspective, gaining automatic insights into fixes that take the entirety of your AWS deployment into account. With CloudFix, you don't just get visualizations, but actual fixes that take place seamlessly without any disruption to your operations.

Start optimizing your AWS costs quickly, easily, and risk-free

SCHEDULE A DEMO TO SEE HOW CLOUDFIX CAN WORK FOR YOU TODAY.

AUREA
CLOUDFIX