

[Home](#) > [Architecting](#)

FinOps explained: a mindset, not a cost-cutting exercise

3 December 2025 - 6 min. read

FinOps

Introduction

What does FinOps mean? Or rather, what is it? In recent years, we have heard more and more about cost optimization and bill analysis. However, how many of you have taken it a step further, considering not just the action itself but also its meaning? And how to apply these practices effectively?

If you are one of those who has always put this topic aside, well, this article is specifically for you; if, instead, you already know the topic, keep reading anyway: there are some gems from our experience you won't want to miss!

The FinOps Framework

Like all self-respecting standards, the FinOps Framework has also been well-documented and defined; it is, in fact, an iterative approach, divided into three different phases:



Inform

When making decisions for the future, we must first understand the present and the past, as well. Indeed, in this phase, we will focus on collecting, analyzing, and standardizing the information.

As a wise man said, "Data tells a story, but you at least need to know how to read it".

A crucial goal that we must set for ourselves, beyond data collection and its standardization, is how to make it comparable across different working groups or projects. KPIs, therefore, turn out to be as important as knowing how to interpret the data. Let's imagine that data collection is the takeoff of a plane, and the analysis is the flight itself; we must still manage to land safely and soundly, otherwise the value of what we have done would be lost!

Optimize

At this point, we finally have clear ideas on what is superfluous or on which components we need to cut costs; we just have to make it concrete.

The Cloud makes this process simpler for us: some options, like 'Reserved Instances', 'Savings Plans', or 'Committed Use Discounts', contribute to cutting bill costs, but they concern the remediation of a situation already in progress; the true FinOps process must start right away, from the birth of a project. One of the major difficulties of this phase is interfacing with different People or Teams within the same or different

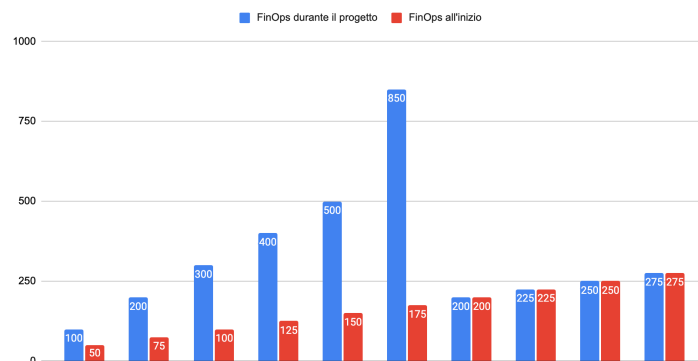
companies. This can lead to moments of tension and competition, but let's always remember that, in the end, we are all rowing in the same direction.

Operate

At this point of the journey, we have made informed decisions based on data and reduced the bill, so we have solved the problem, right?

Well, not exactly.

Let's compare these two charts:



In the first example (blue series), we didn't adopt a FinOps approach from the beginning, incurring avoidable costs, only to reach an unsustainable peak for the project and then refactor everything. This drastically lowered costs, but the initial cost bubble still increased spending significantly, bringing the total to \$ 3,300.

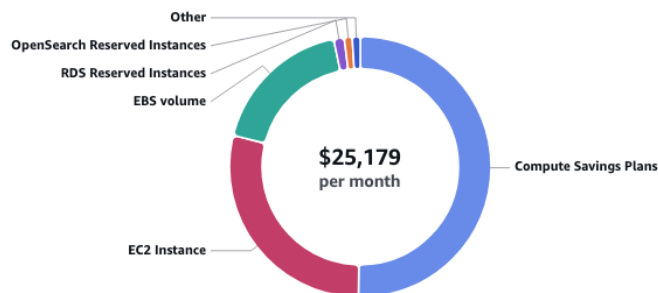
Let's consider the second case (red series), where from the start, we had a mindset geared towards FinOps 'Best Practices', maintaining an optimized cost from the outset. Since no major changes were required during the project lifecycle, the total project cost is \$ 1,625.

In these cases, it proves very useful to start with a TCO (Total Cost of Ownership), which enables us to estimate the cost of our implementation right away, ultimately streamlining areas that require it, and reducing development and project maintenance costs. It therefore turns out to be fundamental to have and cultivate a culture around FinOps practices, so as to consistently guarantee sustainable and efficient projects from an economic perspective.

How can we apply it on AWS?

We've discussed the theory; now let's try to understand how to put it into practice.

Let's start with the services that can help us understand the situation of our accounts; we can always start from a high-level but complete view with Cost Optimization Hub:



Here, we can observe the options available to start performing some optimizations. The Cost Optimization Hub does not perform complex operations, but rather checks which resources are not in use, such as EBS not in use. It also determines if we can leverage Savings Plans or Reserved Instances, a simple yet effective approach.

The service leverages a set of other data, for example, Compute Optimizer. However, we are only scratching the surface. For example, we can subscribe to Savings Plans, thus optimizing the cost, but we must not stop once the first cleanup is done.

Instead of passively using AWS services, we should try to ask ourselves targeted and reasoned questions, for example: "Can we modernize this application? "Are there managed solutions that allow us to lower costs? Can we decouple the monolith and dynamically manage its peaks with scaling policies?"

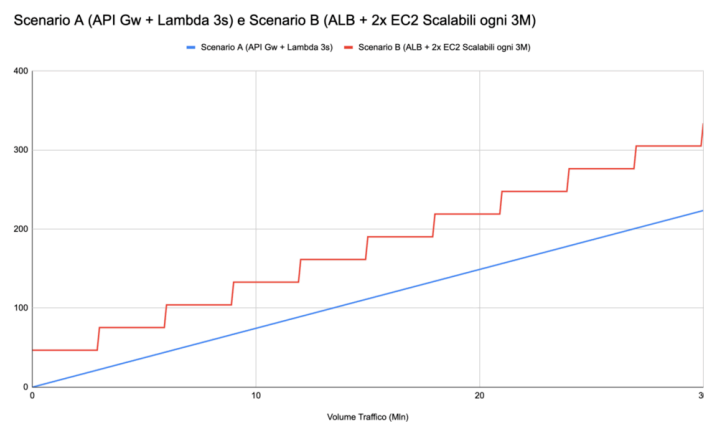
By doing so, we will ensure that we do not accept the current infrastructure without awareness, but rather understand and define the rationale behind these decisions and actions. Let's make a concrete example: if I had a penny for every time I migrated a monolithic web server used to host a static site from on-premise to AWS, I'd have two. Which isn't a lot, but it's weird that it happened twice.

If, instead of directly migrating the static site, we had addressed the context and modernized the solution, leveraging CloudFront and S3, we could have provided a better service at a fraction of the cost of a dedicated EC2. Obviously, it is not always possible to perform these operations due to technical limitations or unforeseen situations; in this case, however, one can weigh the cost of maintenance against the cost of modernization.

This strategy not only can, but must be applied continuously, precisely because of how the cloud provider AWS has conceived the Pricing model. Let's make a concrete example, without going too much into the details of the use case, but focusing on the cost structure:

In the graph below, we have two use cases:

- An API Gateway that exposes APIs and forwards requests to Lambdas, whose average response time is 3 seconds for every call.
- An autoscaling group composed of two EC2 instances that expose APIs through an ALB. We hypothesized the creation of two instances for every 5 Million requests.



In this case, we can see how some managed services can offer the same, or in some cases, more functionalities at a lower cost, being optimized exactly for that use case. In this specific case, however, we should also count a cost delta for the refactor of the application to make it usable by API GW and Lambda; it will therefore be necessary to evaluate, on a case-by-case basis, what is the best and most efficient solution, both economically and technologically!

In some cases, even if it may seem counterintuitive, serverless infrastructures can present themselves as more costly than some serverfull ones. That's not the case, but consider what would happen if the Lambda, instead of being executed in 3 seconds, took 10. The cost would be more than tripled, moving the 'Blue Line' of the graph much higher.

This article is an introduction to the FinOps standard, but also to our approach on it. It was also an overview of what we consider most valuable to the client and of the key questions that can push us towards a better approach to the cloud.

But that's not the whole story. Part II will be out soon to see how these questions and approaches allowed us to optimize, and in some cases even modernize some customers' Workloads.

See you in a few weeks!

About Proud2beCloud

Proud2beCloud is a blog by **beSharp**, an Italian APN Premier Consulting Partner expert in designing, implementing, and managing complex Cloud infrastructures and advanced services on AWS. Before being writers, we are Cloud Experts working daily with AWS services since 2007. We are hungry readers, innovative builders, and gem-seekers. On Proud2beCloud, we regularly share our best AWS pro tips, configuration insights, in-depth news, tips&tricks, how-tos, and many other resources. Take part in the discussion!



Stefano Salvaneschi

DevOps Engineer @ beSharp. I'm the kind of guy that can't let a question go unanswered and I can't resist making them too! Passionate about IT since i was a child, now landed on the Cloud side. In my free time i enjoy playing videogames, dungeons & dragons and everything geeky in between, all accompanied by a nice beer!
