APPTIO
an IBM Company

Shift Left Guide

How to Build Cost Awareness into DevOps Workflows

4 stages to reduce wasted cloud spend—and engineer hours

Closing the Gap Between DevOps and FinOps

Every engineer knows the value of making informed decisions in the earlier stages of application development instead of having to backtrack and redo what's been done. The "shift left" philosophy has already pushed security and testing considerations earlier in the development cycle, driven by the DevSecOps community. Now, it's time for FinOps to shift left as well.

In this guide, you'll learn how DevOps and FinOps teams can work together to make cloud cost management and optimization an integral part of development. By building proactive financial guidance and guardrails into the workflows engineers already use, you can save both cost and time—while building a culture of efficiency for your organization.

This guide covers:

What Is Shift Left for FinOps?

FinOps has traditionally operated as a reactive function. FinOps teams review trends in cloud spend and usage, identify waste and opportunities for optimization, and meet with engineering teams to discuss these findings and implement recommendations.

While this approach can help organizations understand how they're wasting money, it does far too little to prevent waste from accumulating in the first place.

- By the time FinOps gets involved, money has already been spent, and misconfigurations have been deployed throughout environments.
- Organizations must expend considerable effort uncovering occurrences of waste and assigning responsibility to specific teams.
- Even after waste has been identified and connected to teams, resolving this technical debt remains challenging; engineers may have lost the context of their original decisions and need to prioritize the remediation against other work.

Now, FinOps practitioners are evolving to a more proactive approach: helping engineers make the best decisions about cloud resources from the very beginning, before deployment—instead of giving them a list of issues to fix post-launch.

By shifting FinOps left, organizations can save time and money while letting engineers focus on building new applications and features, not fixing existing ones.

The diagram in Figure 1 is familiar to many engineers and testers. As software progresses through the development lifecycle, the work required to fix a defect increases dramatically. Fixing late-stage defects often forces teams to rework previously completed design, code, tests, and documentation while also dealing with new issues that these changes can introduce.

Typical Cost of Resolving Defects in the SDLC

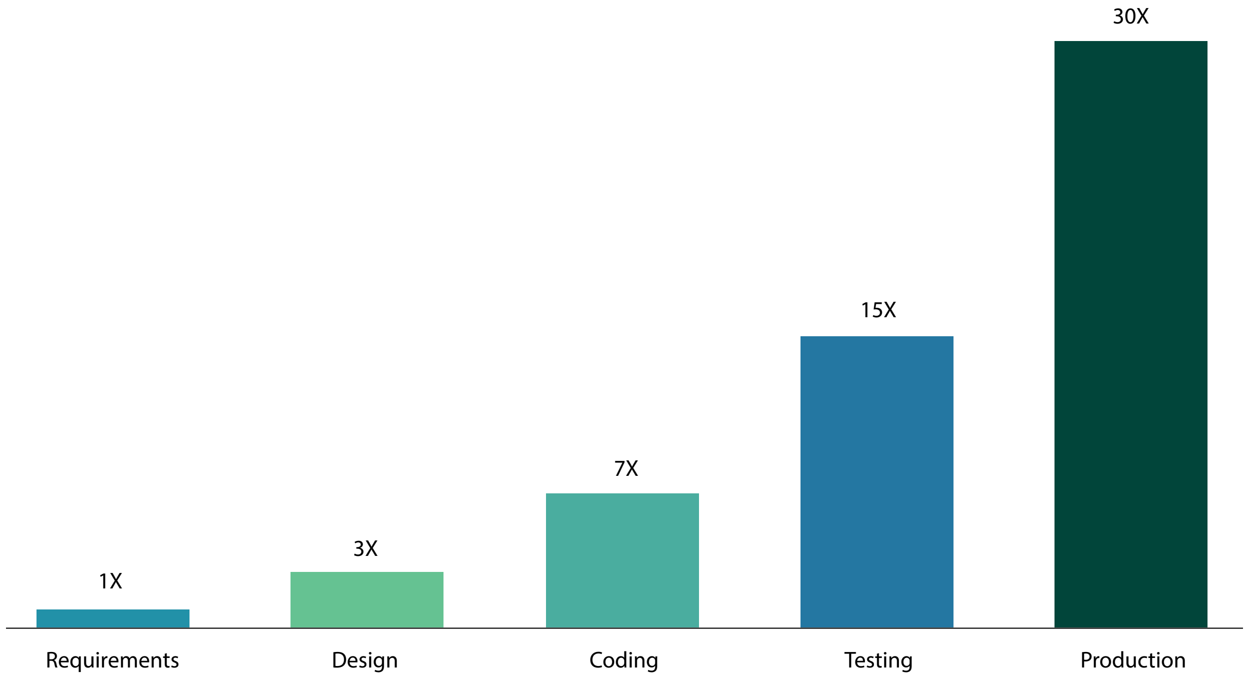


Figure 1: Typical cost of resolving defects in the software development lifecycle (SDLC)

Why Shift FinOps Left?

Cloud waste and inefficiency are nothing new—and neither is FinOps. For nearly a decade, finance and engineering teams have collaborated to manage cloud costs more effectively. Still, in the race to market, cost optimization often remains an afterthought. And once an application has entered production, the best opportunities for maximizing savings have already passed.

The inability to root out wasted spend has real consequences for an organization's bottom line. Facing tight budgets, fierce competition, and the looming cost of AI transformation, organizations need their cloud resources to go toward real business value—not rework, unnecessary toil, and operational overhead.

As every doctor knows, prevention is better than a cure. That means helping engineering teams prevent unnecessary spend and budget overruns by providing guardrails and cost visibility before deployment, when issues are simpler and cheaper to fix.

Workload optimization/waste reduction is by far the top priority of FinOps practitioners.
FinOps Foundation

25% of organizations will experience significant dissatisfaction with their cloud adoption by 2028, due to unrealistic expectations, suboptimal implementation, and/or uncontrolled costs.
Gartner

Most organizations have untapped cloud cost savings of 10–20%.
McKinsey

The 4 Stages of Shift Left for FinOps

Shift left for FinOps means exactly what it sounds like: moving cost analysis and optimization upstream to earlier stages of development. Compounded efficiency across the cycle yields the greatest possible savings, in terms of both cost and engineering hours.

Progressively shifting left for FinOps typically proceeds in four stages:

1. **Identifying waste and mapping it to teams:** ensuring all existing cloud resources are assigned to teams and insights into cost saving are available
2. **Institutionalizing insights to drive action:** embedding optimization recommendations into developer workflows
3. **Establishing proactive insights and guardrails:** integrating cost insights and guardrails into build pipelines
4. **Maximizing business value with cost-aware product decisions:** ensuring cost and business value are key considerations in what to build

What an optimized state can look like:

- Appropriately sized virtual machines (VMs), databases, disks, and containers
- No idle resources
- Well-placed containers
- Cloud-native development
- Dynamically scaled IaaS, PaaS, and Kubernetes resources
- Development and test environments turned off when not in use

Stage 1

Identifying waste and mapping it to teams

No matter how vigilant your organization is about efficiency, some amount of waste almost inevitably exists in your cloud environments. To begin shifting left, FinOps teams must address two key challenges:

1. The specific resources driving waste can be difficult to identify, and the appropriate resolving actions aren't always straightforward.
2. Even when wasteful resources are identified, establishing their ownership can be an onerous task due to the combination of distributed cloud usage and consolidated billing. Monthly billing exports can encompass billions of rows spanning many teams and applications.

In general, it's best to address the ownership problem first. To root out existing waste and poor resource utilization, FinOps teams first need to understand which engineering teams are responsible for the workloads driving this excessive cost.

A strategy built around tags and the vendor hosting construct (e.g., accounts, subscriptions, or projects) can build accountability into your cloud environment by connecting your organizational structure to each cloud resource being deployed:

- **Tagging:** requiring all resources to be tagged with top-level ownership, whether that be the team, cost center, or business unit deploying it
- **Accounts:** creating dedicated member accounts for each team or business area deploying cloud resources
- **Business rules:** applying business rules across tags, accounts, and other billing attributes to capture organizational complexities and provide fallback mechanisms when assigning costs

By keeping track of who's spending what, for what, you can create accountability and provide cost insights and optimization recommendations to the right teams. These recommendations typically fall into two broad areas:

- **Idle resource termination:** Idle resources can be identified using status information (e.g., unattached volumes) or categorical utilization information (e.g., zero database connections). Heuristics are necessary for some resource types, such as flagging VMs with less than 2% sustained CPU utilization.

- **Rightsizing overprovisioned resources:** Resources should be closely aligned to the requirements of their underlying workloads. Common mismatches include being oversized—such as deploying an m8g.16xlarge EC2 instance when an m8g.xlarge would suffice—or being misshaped, such as using a memory-optimized r8g.xlarge instance when a cheaper compute-optimized c8g.xlarge would be more appropriate.

IBM Cloudability helps create accountability with:

- Tag mapping and account groups for accurate cost allocation
  Learn more

- Business mapping to connect IT, finance, and line of business teams
  Learn more

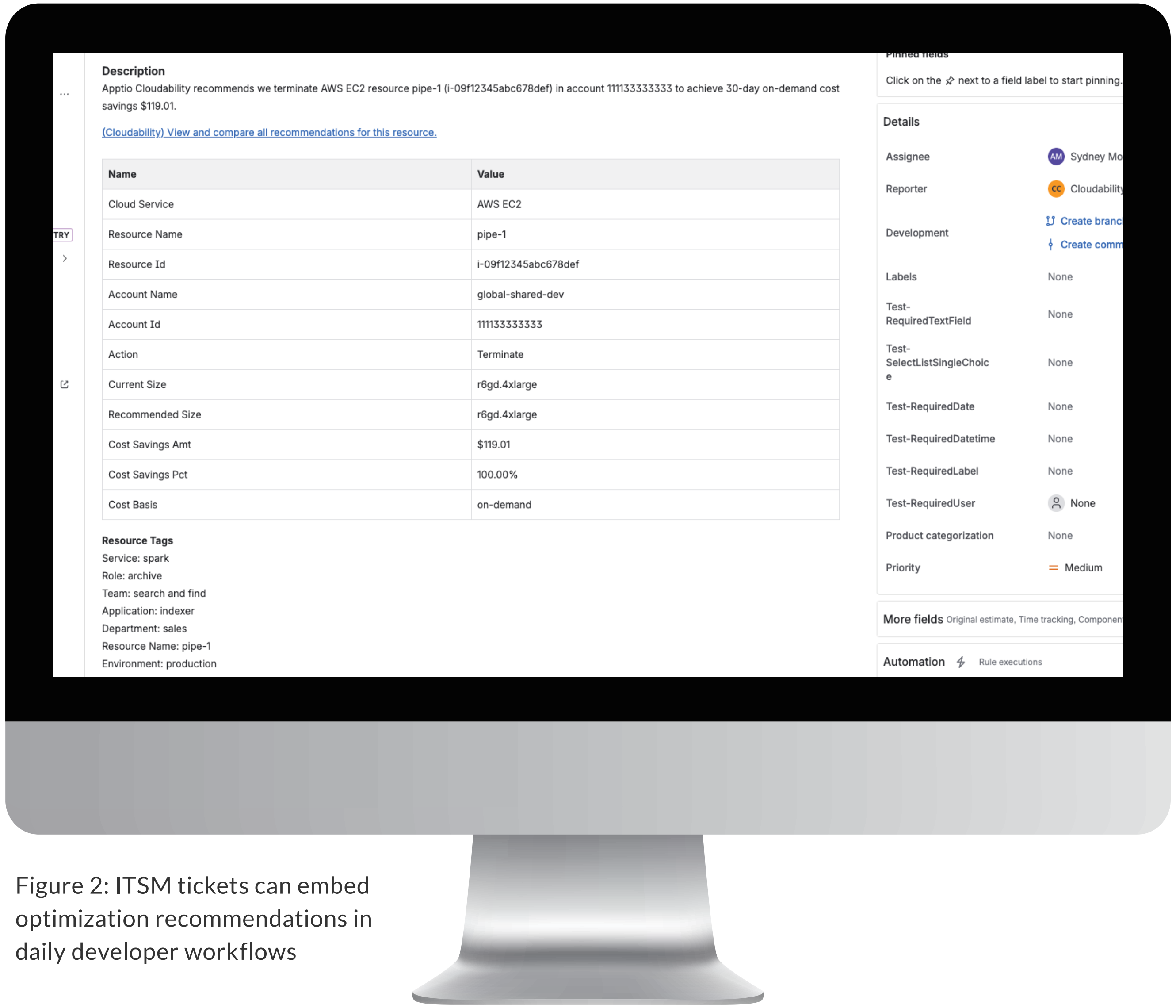
- Rightsizing recommendation across a broad set of service types
  Learn more

Stage 2

Institutionalizing insights to drive action

Uncovering existing waste and making recommendations for optimization is an important first step—but if these insights remain siloed, they won't influence engineering decisions or get actioned. To drive dependable cost savings, they must be embedded directly into workflows where engineers spend their time.

One approach is to push optimization actions and insights into an ITSM tool (such as Jira or ServiceNow), creating tickets for each sizeable rightsizing and idle resource savings opportunity. This lets engineers address waste as part of their daily routine, or work through the recommendations for a given application during their next DevOps sprint. Assigning and tracking these actions can provide an incentive and enable gamification strategies.



Description

Apptio Cloudability recommends we terminate AWS EC2 resource pipe-1 (i-09f12345abc678def) in account 111133333333 to achieve 30-day on-demand cost savings $119.01.

(Cloudability) View and compare all recommendations for this resource.

Name	Value
Cloud Service	AWS EC2
Resource Name	pipe-1
Resource Id	i-09f12345abc678def
Account Name	global-shared-dev
Account Id	111133333333
Action	Terminate
Current Size	r6gd.4xlarge
Recommended Size	r6gd.4xlarge
Cost Savings Amt	$119.01
Cost Savings Pct	100.00%
Cost Basis	on-demand

Resource Tags
Service: spark
Role: archive
Team: search and find
Application: indexer
Department: sales
Resource Name: pipe-1
Environment: production

Pinned fields

Click on the next to a field label to start pinning.

Details

Assignee	AM Sydney Mo
Reporter	CC Cloudability
Development	Create branc Create comm
Labels	None
Test-RequiredTextField	None
Test-SelectListSingleChoice	None
Test-RequiredDate	None
Test-RequiredDatetime	None
Test-RequiredLabel	None
Test-RequiredUser	None
Product categorization	None
Priority	Medium

More fields Original estimate, Time tracking, Componen

Automation Rule executions

Figure 2: ITSM tickets can embed optimization recommendations in daily developer workflows

While making optimization part of developers' business as usual reinforces accountability, it can also receive pushback if they perceive it as adding work to engineering queues. To ease acceptance, emphasize that the practice will benefit DevOps as much as it does FinOps. By eliminating waste more effectively, engineers can free up resources to use for more interesting and meaningful projects.

Make optimization fun with FinHack sessions and gamification strategies like leaderboards and badges.

Keep in mind that not every optimization task is worth pursuing compared to other priorities. The most successful FinOps practices create an automated policy that only creates tickets in cases where the cost savings will be large enough to warrant the required effort.

At this stage, the function of FinOps has remained reactive: identifying and addressing existing waste. In the next stages, we'll move earlier in the development lifecycle to prevent waste from accumulating in the first place.

IBM Cloudability helps engineers prioritize optimization with policy-driven rightsizing insights and ROI tracking. Learn more

Stage 3

Establishing proactive insights and guardrails

In the first two stages, we discussed strategies for addressing existing waste more effectively. While these are crucial stages to work through, it's worth remembering that most cost inefficiencies are introduced during development. If we only catch these issues after deployment, it's already too late to address them at optimal cost and effort.

The core problem is well known: while modern CI/CD processes and Infrastructure as Code (IaC) empower engineers to deploy rapidly through automated build pipelines, these processes typically lack integrated checks for cost efficiency or infrastructure compliance. As a result, any modifications engineers make to IaC files in source control are typically applied to cloud environments without safeguards to prevent overspending or non-compliant configurations.

To address this, we'll shift further left, making FinOps proactive for the first time. Within the build pipeline, whenever an engineer makes an IaC change, we'll help with advisories or guardrails across concerns.

Concern	Advisory or Guardrail	Benefit
Uncertain cost impact	Show expected cost delta	Avoid major cost surprises
Missing required tag	Highlight missing tag key	Improve chargeback accuracy
Unapproved tag value	List approved tag values	Improve chargeback accuracy
Unapproved instance type	List approved instance types	Stay compliant with policies
Unapproved region	List approved regions	Stay compliant with policies
Budget violation	Show budget and forecasted violation	Avoid major cost surprises
Optimal resource selection	Highlight cheaper resource options	Improve cost efficiency

There are several options for where these advisories and guardrails could be implemented, to be triggered before the configuration change goes live:

- The most obvious is to embed them into the pull request process, having them surfaced alongside the proposed commits in source code management platforms like GitHub or GitLab.

- Depending on your situation, another option may be to surface them to engineers in an IaC platform like HCP Terraform.

Building insights and policies into the CI/CD pipeline can help foster a culture of cost-effective development.

A key decision for the FinOps team is whether the checks—for example, a check on a mandatory Cost Center tag—would be purely informational or actually prevent the pull requests from proceeding.

IBM Cloudability provides cost insights whenever engineers make GitHub pull requests to create or change infrastructure. Learn more

Stage 4

Maximizing business value with cost-aware product decisions

Proactive advisories and guardrails help engineers make the right decisions for the applications they're building. But a more fundamental question remains: How can we ensure that the business value of each cloud application will justify its costs?

In the final stage of shifting FinOps left, we reach the very beginning of the product design and development process. Here, product, engineering, and FinOps teams work together to make strategic decisions about what to build and how to build it. Using the granular data provided by a FinOps platform, these teams can:

- Identify the most cost-effective way to build an application, including the optimal cloud or on-premises architecture and infrastructure
- Evaluate the resulting expectations for performance, scalability, cost of goods sold (COGS), and ROI
- Decide whether the project will meet the strategic goals and financial targets set by C-level leadership

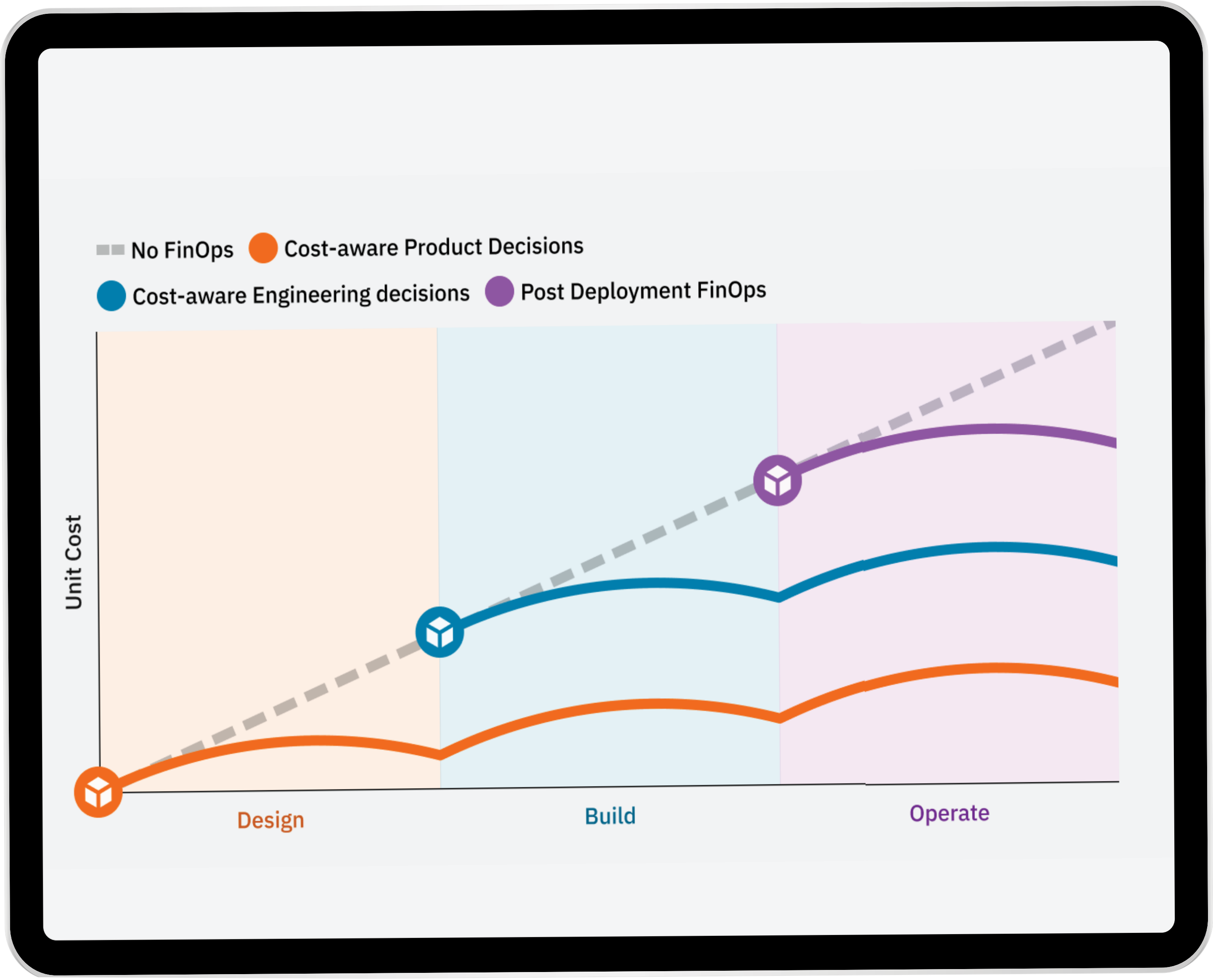


Figure 3: Shifting cost focus along the product development lifecycle by the FinOps Foundation

If the project's economics prove worthwhile, the team can proceed with the initiative and engineering should build in guardrails to ensure it stays on target.

Sometimes it's just not possible to optimize an application to the point of profitability. Pulling the plug once in production can limit losses, but making a no-go decision at the outset prevents time and resources from being squandered in the first place.

Questions to ask:

- How can we build this as efficiently as possible?
- What's the resulting unit cost?
- Are we likely to achieve worthwhile ROI?
- Do these economics make sense for our business?

Getting Started

By implementing a shift left approach to FinOps, organizations can reduce waste, realize greater value from their cloud spend, and build a culture of financial accountability.

As you shift left, collaboration is key every step of the way. Engineers understand technical requirements. FinOps practitioners understand vendor contracts and budget constraints. Product teams understand business goals and financial expectations. Working together to develop processes helps build trust and confidence on all sides. And having the right FinOps tooling partner can ensure the best outcomes for all involved.

As a leader in the FinOps space, IBM Cloudability provides tools and services to embed cost insights and policies throughout the development lifecycle. Learn more about shifting FinOps left with IBM Cloudability.

APPTIO®
an IBM Company

Apptio, an IBM company, is the leading technology spend and value management software provider. Its AI-powered data insights empower leaders to make smarter financial and operational decisions. A pioneer and category leader in TBM, FinOps, and Agile Portfolio Management, Apptio works with thousands of customers, partners, and community members worldwide.

Start Your Free Trial