Beyond FinOps Dashboards

Shift left to prevent cloud budget overruns

Shift FinOps Left
Since 2021

According to the FinOps Foundation's 2025 State of FinOps report, workload optimization and waste reduction remains the number one priority for FinOps teams for the second consecutive year. The urgency is justified. 69% of organizations experienced budget overruns in their cloud spending. Often the delta is significant, with 33% of businesses exceeding their budget by 40% or more.

There are hundreds of available FinOps tools designed to manage cloud budgets. The visibility they provide is essential, but they struggle to keep up with modern deployment workflows. Infrastructure as Code (IaC) tools like Terraform and CloudFormation democratize deployment, making it difficult to wrangle in costs.

> Infrastructure automation through IaC requires teams to take a more proactive rather than reactive approach to cloud waste and budget overruns.

Platform and FinOps teams need tools to prevent cloud waste, not just provide visibility after the fact. This whitepaper outlines:

- Why using traditional FinOps tooling alone fails to address budget overruns
- The advantages deploying infrastructure as code (IaC) provides for governance and optimization, including for FinOps teams
- How shifting FinOps left in the CI/CD workflow enables better governance, and what this looks like in practice.

You will learn how current IaC practices often lead to overspending and how to address it with practical solutions. It also outlines how to shift FinOps left to prevent overspending without slowing your teams down.

Visibility alone doesn't solve budget overruns

Traditional FinOps tools provide essential visibility across all major cloud providers or business units. This visibility is key for budget forecasting year over year. However, visibility alone cannot prevent the budget overruns that affect the majority of organizations. The gap between seeing the problem and actually remediating the problem represents the fundamental challenge facing FinOps and Engineering teams today.

By the time these FinOps teams can communicate findings to engineers, infrastructure is already provisioned and the money is spent. Visibility without prevention creates a reactive cycle of identifying waste, coordinating remediation, and waiting for changes—only to repeat the process next month.

Visibility alone doesn't solve budget overruns

Traditional FinOps tools provide essential visibility across all major cloud providers or business units. This visibility is key for budget forecasting year over year. However, visibility alone cannot prevent the budget overruns that affect the majority of organizations. The gap between seeing the problem and actually remediating the problem represents the fundamental challenge facing FinOps and Engineering teams today.

The Velocity-Cost Paradox

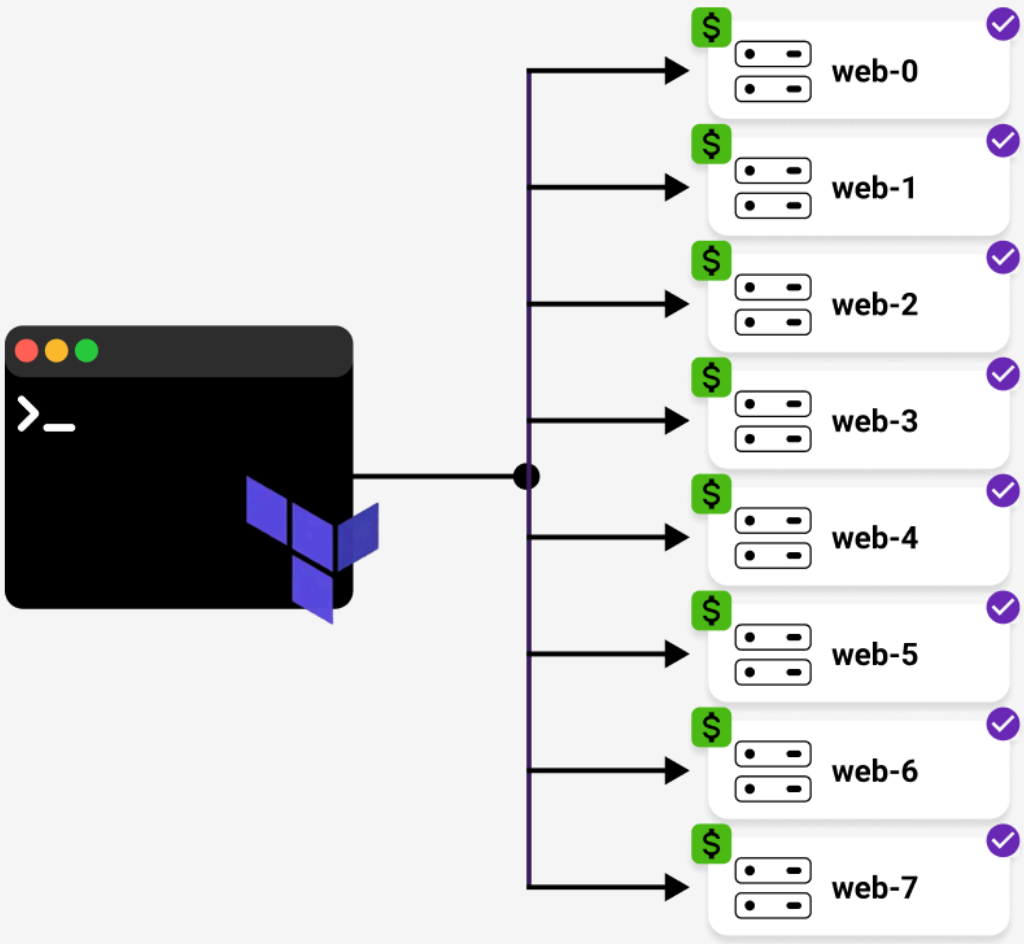
Why current FinOps processes can't keep pace with infrastructure automation

The deployment methods have changed significantly since FinOps started as a practice. Infrastructure as Code (IaC) tools like Terraform and CloudFormation enable unprecedented provisioning speed and automation across all major cloud providers. They've become a standard in less than ten years. According to recent market research, the IaC market is projected to grow from $1.74 billion in 2024 to $12.86 billion by 2032.

> Over 80% of enterprises now integrate Infrastructure as Code into their CI/CD pipelines, enabling self-service infrastructure across distributed teams.

This is precisely why FinOps teams struggle to keep pace. Infrastructure as Code enables ondemand provisioning. Gone are the days of submitting tickets and waiting six months for a server. Engineers can use pre-written, pre-approved IaC modules or templates to provision their infrastructure in minutes. So how does this impact cloud budgets?

> The engineers who are now empowered to provision their infrastructure often have little to no visibility into the costs of the infrastructure they provision.



IaC enables provisioning velocity, but also speeds up the rate of cloud waste.

Consider the paradox: organizations invest in comprehensive FinOps tools for budgeting while simultaneously empowering any developer to provision infrastructure in minutes with almost no insight into how much they can spend or what everything costs. How can FinOps teams build processes to reduce spend when cost prevention isn't part of the provisioning workflow? By the time the waste is identified, remediation requires coordination, meetings, and rework – all while the infrastructure continues to accrue costs.

The automation introduced by infrastructure as code requires an automated approach to governance and compliance. The industry established shift-left security, testing, and compliance. Now FinOps must follow the same path through shift-left cost management tools like Infracost.

What is Shift-Left FinOps?

The solution lies in bringing FinOps governance to where infrastructure decisions are actually made: the code itself. Shift-left is the practice of enforcing compliance and governance as part of the CI/CD workflow. Security and compliance teams have successfully applied this principle for years. Tools like Snyk and Trivy scan code for vulnerabilities, check containers for security issues, and validate infrastructure configurations—all before deployment to production.

Infrastructure as Code provides a key advantage for a shift-left strategy:

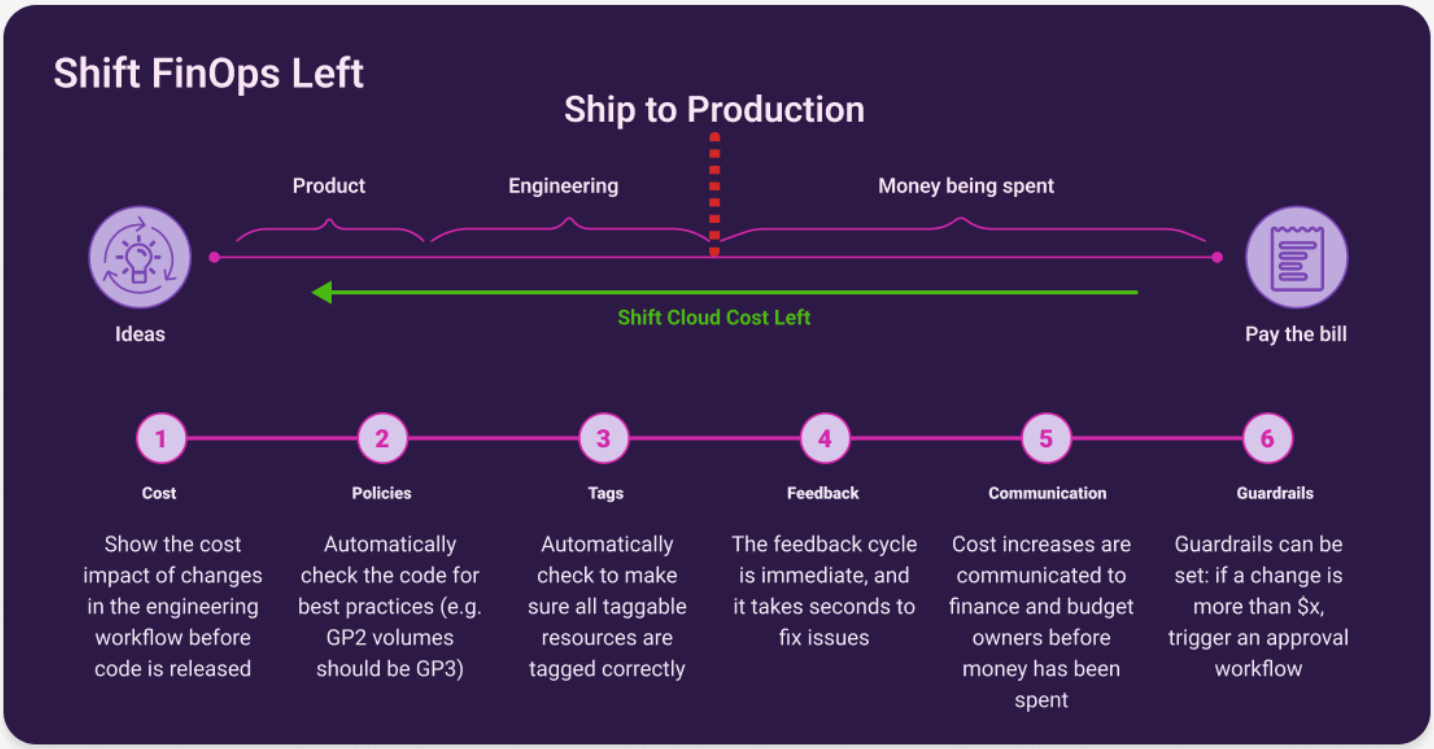
> Codifying infrastructure creates a single source of truth that makes governance scalable and repeatable. Instead of chasing down misconfigured resources after deployment, teams can enforce tagging standards, cost policies, and security controls directly in the code review process.

FinOps teams should take advantage of this approach to reduce the volatility of cloud spending. Instead of scheduling meetings and sending tickets to reclaim resources after deployment, FinOps teams can establish standardized or custom policies that enforce automatically in every code review—ensuring consistent cost controls across hundreds of teams and repositories. This prevents overspending before it occurs while eliminating unnecessary meetings and tickets that slow down provisioning velocity.

Shift-Left vs. Traditional FinOps Workflows

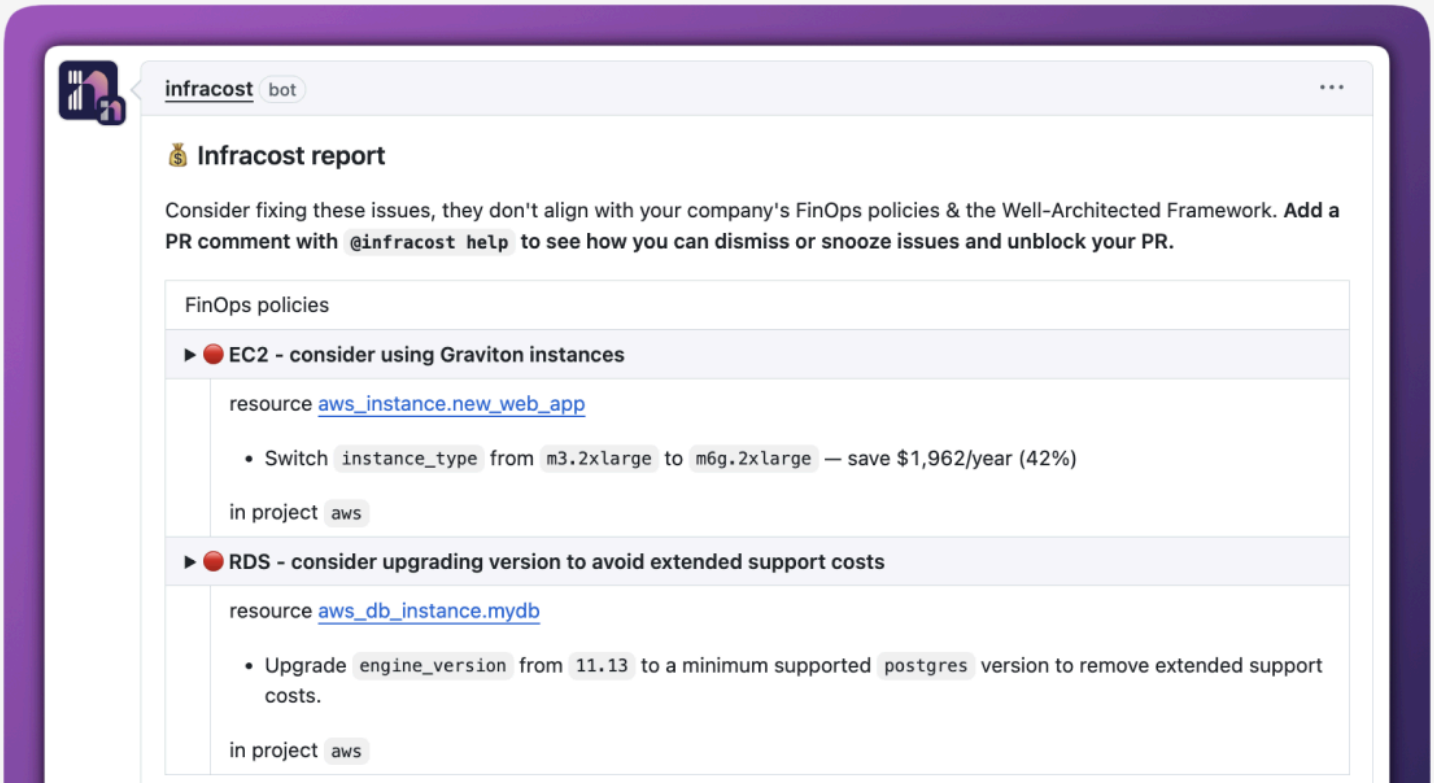
Like Infrastructure as Code itself, using FinOps governance tools requires a cultural shift within an organization. It starts with providing visibility into the costs of each deployment, both within the IDE and as part of the CI/CD workflow. Once this workflow is established, it's easier to integrate policies and cost guardrails into the CI/CD pipeline.

Requirement	Traditional FinOps	Shift-Left FinOps
Timing	FinOps addresses spend after infrastructure is provisioned and costs are incurred by other teams	FinOps creates policies and guardrails which are visible during code review, before resources are created
Cost Visibility	Monthly bills and dashboards are only seen by FinOps teams	Pull requests show engineers the cost impact before deployment
Waste Response	Identify waste, create tickets, coordinate remediation with teams	Prevent waste by catching issues before deployment
Policy Enforcement	Manual audits and remediation campaigns	Automated checks in CI/CD prevent policy violations
Developer Experience	Interrupted by tickets and meetings about deployed infrastructure	Cost feedback integrated into existing workflow
Budget Control	React to budget overruns after they occur	Prevent budget overruns through pre-deployment validation
FinOps Team Focus	Reactive firefighting and coordination	Strategic optimization and policy refinement



What does this look like in practice?

With new cloud services and initiatives introduced monthly, a FinOps team's job is never done. By implementing a Shift-Left strategy with tools like Infracost Cloud, they can highlight the cost visibility, policies, and guardrails directly into the CI/CD pipeline. This means IaC practitioners can help stay under budget without slowing down provisioning velocity.



infracost bot

💰 Infracost report

Consider fixing these issues, they don't align with your company's FinOps policies & the Well-Architected Framework. **Add a PR comment with `@infracost help` to see how you can dismiss or snooze issues and unblock your PR.**

FinOps policies

▶ 🔴 **EC2 - consider using Graviton instances**

resource aws_instance.new_web_app

- Switch `instance_type` from `m3.2xlarge` to `m6g.2xlarge` — save $1,962/year (42%)

in project `aws`

▶ 🔴 **RDS - consider upgrading version to avoid extended support costs**

resource aws_db_instance.mydb

- Upgrade `engine_version` from `11.13` to a minimum supported `postgres` version to remove extended support costs.

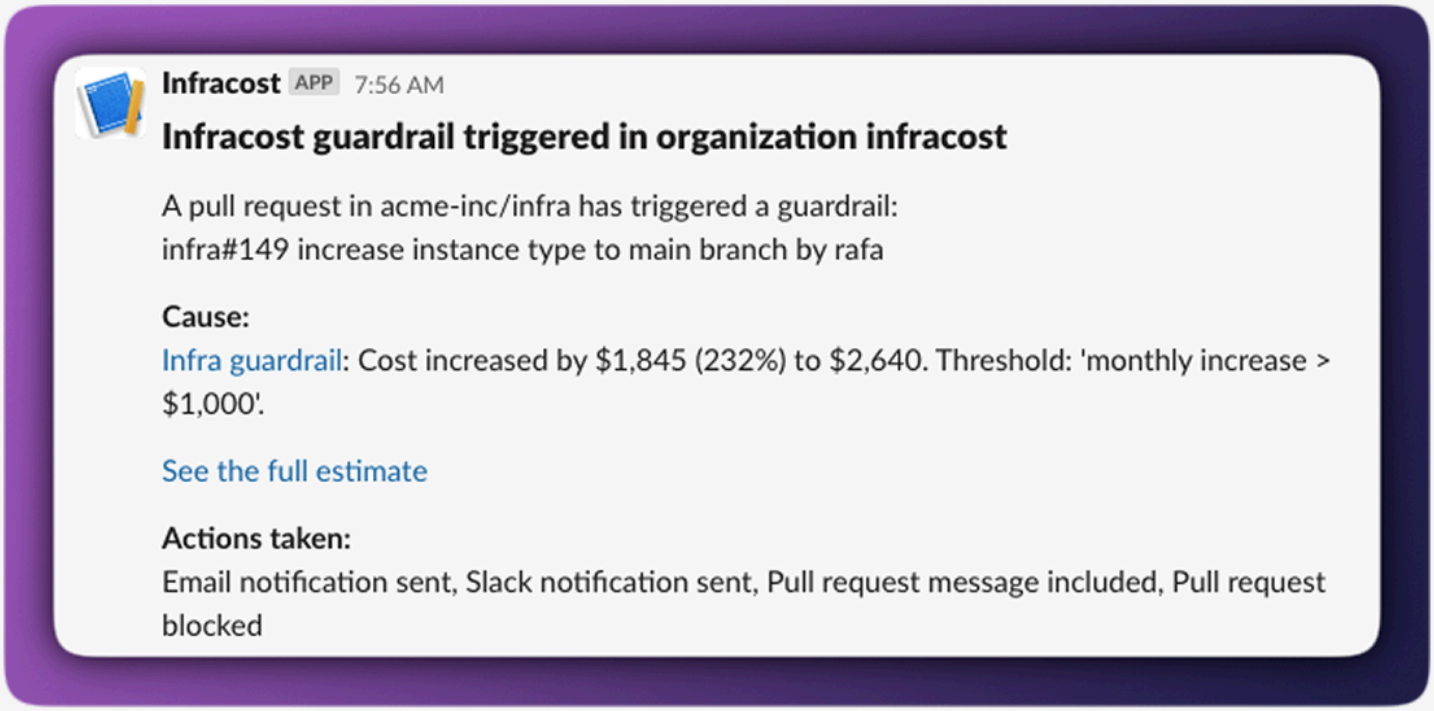
in project `aws`

FinOps governance delivered directly in pull requests

FinOps for IaC workflows is no small feat. It requires code parsing of the IaC file itself and extracting any cost-related parameters. There are hundreds of possible cloud services defined in IaC files and literally millions of price points across the three major cloud providers. With this complexity, platform teams must embrace battle-tested tooling and automation that can keep pace with updates across cloud providers. Writing custom rules built with OPA or other policy engines would require platform teams to constantly monitor for pricing updates and new services across cloud providers.

> By putting these costs as well as best practices directly into the workflow engineers use every day, they build a “cost muscle" within their IaC practice. Engineers can prevent overprovisioning and cloud waste before it is deployed.

This workflow doesn’t just provide basic cost visibility and recommendations. FinOps teams can use it to provide budget guardrails and policies for engineers as well. When provided with a centralized console to manage governance, FinOps teams can create hard and soft requirements to reduce cost across deployments, as well as approval processes that work asynchronously. This enables FinOps teams to meet their goals without slowing engineers down.



With a shift-left strategy, FinOps teams are armed with the tools and asynchronous processes to reduce not only open issues, but prevent new issues later.

Tagging Policies	FinOps teams can validate that resources have required tags with correct keys and values before deployment to ensure cost allocation. They can create rules that block deployment if cost center, owner, or environment tags are missing or misspelled.
Budget Guardrails	FinOps teams can set budget thresholds that trigger notifications or block pull requests when cost increases exceed defined limits. For example, FinOps teams can block any pull request for a certain VCS repo if monthly costs exceed $10,000.
Optimization policies	FinOps teams can leverage hundreds of out-of-the-box policies that help optimize instance types, data lifecycle policies, log retention policies, as well as avoiding deprecated software versions with additional extended support fees.

Additional benefits: Optimizing existing workloads for potential cost savings

Infrastructure as code enables proactive cost control for new deployments. It also provides FinOps teams with a powerful mechanism to evaluate and optimize existing workloads. When infrastructure is codified, the entire cloud footprint becomes queryable and analyzable at scale. FinOps teams can scan thousands of Terraform files across hundreds of repositories to identify cost optimization opportunities like outdated instance types, inefficient storage configurations, or missing lifecycle policies. This code-level visibility eliminates the gap between cloud billing tools that show what is expensive and the engineering changes needed to fix it.

Instead of manually tracking down resource owners and coordinating remediation through tickets and meetings, FinOps teams can pinpoint exactly which line of code needs to change, prioritize fixes based on potential savings, and even automate pull requests that implement the improvements—turning what used to be a months-long audit process into a continuous optimization workflow.

Moving From Visibility to Prevention

The automation and velocity enabled by Infrastructure as Code has fundamentally changed how infrastructure gets provisioned. This requires a corresponding change in how costs are managed. Shift-left FinOps brings cost visibility and governance directly into the engineering workflow—where infrastructure decisions are made. FinOps and DevOps teams are empowered to take action and reduce cloud waste before it's provisioned.

The question is no longer whether shift-left FinOps is necessary, but rather how quickly organizations can implement these practices to control cloud costs before the next budget cycle

How to get started with a shift-left FinOps strategy

Infracost provides FinOps for infrastructure automation. It integrates directly into your CI/CD workflow, so engineers using Infrastructure as Code catch expensive mistakes before they hit production. Beyond cost visibility, Infracost helps FinOps teams enforce tagging policies, evaluate infrastructure against best practices, and generate reports for ongoing optimization.