

Solution Brief: Cost Management

Introduction

Many companies turn to the cloud with organizational productivity in mind and soon find that tendencies of over-provisioning, a lack of financial visibility and abandoned infrastructure result in inflated costs. Cost management soon becomes a crucial factor in a successful cloud strategy.

When enterprises attempt to define the scope of the “Cloud Cost Management” challenge they often arrive at three fundamental components:

Budgeting - The ability to create budgets and enforce them on projects

Visibility - The ability to track spend at a granular level

Accountability - The ability to perform chargeback/showback in an efficient manner

However, there are more issues to keep in mind.

First, the information must be actionable. The difference between a useless report and actionable data is context, if a tool reports on an overspending machine, it's vital to know which application the machine belongs to and what will happen if we shrink or terminate it.

Second, if you hold enterprise groups accountable for operating within budget, it's useful to create an environment that makes it easy to follow financial guidelines and hard to overspend.

Preventative and reactive controls are required to build such environments.

Common Causes For Overspend

Overprovisioning

When given a choice, end-users often prefer to err on the side of over-provisioning to ensure uptime and avoid wasting time scaling up infrastructure. Provisioned resources are also frequently left running for longer than required, for example, a test stack that's left running over the weekend.

Solution:

An intelligent provisioning workflow that guides end-users to the right resources for the job. A lifecycle policy that scales-in non-production workloads after-hours.

Abandoned Infrastructure

Application stacks are regularly left running after a developer is let go or moves to another project, or a training session concludes. Abandonment quickly drives up spend but can be easily avoided.

Solution:

Reclamation Policies that assign each provisioned resource with a lifetime. Once the lifetime is exceeded, the application is terminated, but it's desired state is preserved. If users wish to keep the application running, they must submit business justification for it.

Lack of Visibility

The lack of visibility into cloud spend compounds the challenges described above. Optimization becomes impossible without clear, low-level financial visibility into the company's multi-cloud and datacenter deployments.

Solution:

Hierarchical visibility into the organization's environments:
Business Unit/Department -> Project -> Application -> Tier -> Instance/Service.

Controlling Costs in the Public Cloud

The public cloud presents some key differences in cost management, namely the lack of fixed capacity. Each public cloud offering has different infrastructure options with various costs associated with them.

The challenge becomes managing associated costs rather than capacity, so the motivation shifts to preventing “runaway” cost rather than traditional capacity planning.

Controlling Cost in a Hybrid Cloud

The organization’s approach to cost management should also account for the hybrid cloud reality that most every enterprise operates in. For example, some applications might run in a private cloud or datacenter environment for security, performance or licensing reasons.

Perhaps there’s a strategy in place to eventually migrate, re-architect and optimize these applications to the public cloud, but maybe not. Different enterprise business units might leverage various cloud platforms according to their needs, leaving IT to serve a unified front-end to a hybrid back-end.

In any event, most enterprises will have a mix of infrastructure, with applications running in the public cloud, private cloud and datacenter, across multiple providers. Costs might be managed differently for each platform.

Cost Management “Shopping List”

What are today’s enterprises looking for when facing the challenges of managing and controlling cloud costs?

Optimized Provisioning Process

Cost management is more than budgeting and reporting. Businesses are integrating financial policies into the provisioning process through budget enforcement, automated tagging, and reclamation policies. Both preventative and reactive controls are put in place to create an effective guardrail around resource consumption. An optimized provisioning workflow keeps usage within budget while making users more productive.

Application Level Visibility Across Platforms

With traditional datacenters, Finance departments lacked low-level visibility and often worked with a lump sum representing infrastructure spend. Since anything that’s hard to measure is hard to optimize, not much could be done in the way of reducing and controlling costs. A multi-provider cloud environment presents a similar issue, so enterprises seek centralized, application-level visibility and tracking capabilities across all cloud platforms.

Contextual Financial Reporting

When identifying overspending applications, administrators require context to make informed decisions. A report indicating an over-budget machine will not be useful unless the administrator can quickly determine which application the machine belongs to. Enterprises require in-line tools that provide contextual, actionable data, as opposed to scanning the cloud environment from the outside.

Federated Accountability

From business units to teams and users, large organizations hold internal customers accountable to a financial plan. Enterprises seek efficient chargeback/showback models to ensure accountability and budget compliance at multiple organizational levels. Therefore solutions that map to the organizational structure are required.

Scalr's Approach to Cloud Cost Management

Actionable Data Lives in The Workflow

Scalr acts as a bridge between the end user and the Enterprise Cloud. Developers, DevOps engineers or any other type of user go through Scalr for self-service cloud resources, and the Policy Engine ensures all usage is within company policy.

Scalr's central role in the provisioning workflow lends context to financial reporting and tracking. Instead of getting a notification about an overspending machine, you learn that an application is overspending, and will then be able to determine which tier or particular machine is the root of the problem.

With this top-down approach, you're able to predict how terminating an instance or making capacity changes will impact the application.

Cost Control through Intelligent Workflows

Cost-efficient systems are often a result of workflows that guide end-users to the right decisions. Scalr's Policy Engine empowers you to bake best practices into intelligent policies to make your users and their workflows more productive.

For example:

Allow DevOps engineers more operational freedom and the ability to choose their own instance types and services within an assigned budget and in compliance with the reserved instance profile.

For QA engineers who need to spin up the same test stack multiple times, and other users with repetitive requests - create a simplified storefront experience that encourages them to choose inexpensive options whenever possible.

Visibility, Accountability, and Responsibility

Most cost-control issues in a decentralized cloud environment stem from a lack of application-level visibility. With Scalr, you're able to hold business units, departments and teams accountable, while providing them with the tools to responsibly manage their budgets and deliver services to end-users.

Cost Control As Policy

The Scalr Enterprise-Grade CMP works by creating a guardrail around the consumption of cloud resources. Scalr provides end-users with a modern, streamlined self-service experience. Scalr policies fall into five general categories:

Access Policies

Example: Use RBAC to ensure that new developers can only edit applications owned by their team, preventing new hires from making costly mistakes on production workloads.

Workload Placement Policies

Example: Create a placement policy that only allows users to provision instances that match the Reserved Instance profile.

Integration Policies

Example: Track license usage across your organization and increase cost transparency by reflecting associated costs in spend reports.

Financial Policies

Example: Track spend across business units and cloud platforms, communicate financial costs to users before provisioning, and set notifications around budget consumption.

Lifecycle Policies

Example: Create an automation policy that scales down relevant applications after work-hours. Create a reclamation policy that assigns a 5 day lifetime to all QA workloads, ensuring they don't persist when they're no longer used.