2 MODELS FOR OPTIMIZING CLOUD COST

Introduction

At some point in their cloud journey, most organizations hit a major obstacle. After experimenting with a workload or two, organizations begin to scale their cloud investment and estate. It is only then that executives realize they have a problem: their teams lack the skills, tools, and processes they need to track, manage, and optimize expenditures. Consequently, costs rise rapidly and often dramatically exceed budgets and expectations.

In this post, we look at some key strategies and tactics that teams can employ to start optimizing their cloud investments and resources.

Cloud Optimization Models

In pursuing cloud cost and resource optimization, IT and cloud teams can choose from two models:

- **Daily operational optimization.** Employing this model, team members will look at tools on a daily basis, and see a prioritized list of tasks for saving costs. Every day, they'll execute those tasks they can, and then start the process again the next day.
- **Cycle-based optimization.** With this model, teams will aggregate all recommendations and establish optimization cycles. These cycles can be based on a range of intervals, typically monthly or bi-monthly. At a high level, the process entails establishing baselines, executing recommended optimization tasks, evaluating and auditing progress, and, at the end of the cycle, establishing new baselines to track progress.

For smaller cloud estates, the daily operational approach is great. On the other hand, for most large enterprises, cycle-based optimization will be the best approach. The following sections offer more details on this cycle-based approach.

Cycle-based Optimization: Four Key Steps

There are 4 steps to cycle-based optimization, which is a continuous process.

Step 1: Recognize. The first step is for teams to establish an awareness of what resources are in place. This is vital to establish a complete, current picture of all the resources in place—both on premises in and public clouds. Teams should look to establish visibility and insights by business service and application in order to most intelligently determine what resources exist and how they are used. This will serve as an initial baseline for measurement and changes.

Step 2: Analyze. In this phase, teams assess prescriptive recommendations and prioritize efforts. Tasks should be broken out by service tiers, for example, distinguishing between production, non-production, shared services, and so on. Often, efforts will be prioritized by cost savings. Teams can come up with standard guidelines to help ensure the payoff is aligned with the economic models. For example, guidelines could indicate that those efforts that net savings of $1000 or more are given the highest priority, those less than $1000 will be lower priority, and those less than $100 will not be pursued.

Step 3: Optimize. The goal of this phase is to ensure the most efficient use of resources and budget. Teams will typically apply a predetermined method for identifying and then implementing optimization recommendations. Optimization recommendations can include options for right-sizing cloud resources, placing intermittently used resources on schedules, and using on-demand, discounted, or spot pricing for purchasing cloud resources.

Step 4: Evaluate. The last phase is where teams compare metrics against the previously established baseline, calculating actual cost savings realized based on recommendations implemented. In this effort, it is important to establish apples-to-apples comparisons between baselines and current state. This means factoring in changes that may have occurred, for example, accounting for changes in pricing or new instances that may have been provisioned since the baseline was established. Finally, the new metrics gathered then become the new baseline for the next cycle, and the process begins again.

Cycle-based Optimization: How it Works in Practice

Cycle-based optimization needs to follow a well-formulated series of tasks. To provide an example, we'll look at how the process would work when doing cycle-based optimization for computing resources. Also, it is important to note that, given computing and storage typically represent about 70% of what organizations consume from an infrastructure perspective, this is a good place to begin for any team getting started with cycle-based optimization.

Stage 1. Workload examination

Teams start by assessing the following characteristics:

- **Aging.** Teams assess how long a given instance has been running. Why look at that? Instances that have been running for a long time may be good candidates for upgrading or tuning.

- **Idle.** This involves looking at utilization rates at different times. Often, there are time-series patterns that emerge. For example, a server may be running at 2% utilization most of the time, except for Wednesdays at 11:00 p.m., when utilization hits 100% for an hour. In this type of case, rather than keeping the instance up at all times, an organization may want to put it on a schedule so it's automatically powered on for that one hour on Wednesdays, and then powered down the rest of the time.
- **Inactive.** Inactive resources are important to flag. Often, if an inactive resource is detected, it may make sense to ascertain if the associated service still makes sense to support. Often, teams inherit processes and don't take time to validate whether they are all still valid. Resource inactivity will frequently be a sign that there's an inherited process that's not needed any more, and cloud migrations can be a good time to uncover these cases.

Stage 2. Generation migration

Teams next need to look at the current generation instance, and determine whether it makes sense to migrate to another generation that's cheaper or faster. For example, through this analysis, a team may determine that an instance running AWS C4 should be migrated to AWS C5.

Stage 3. Resizing

Once teams have determined whether to migrate or not, they'll next want to ascertain whether the instance needs to be resized. After an evaluation, the determination may be to upsize or downsize. Here again, sequence is important. If a new generation is faster, a team may not have to upsize or may even be able to downsize after migrating to that newer variant.

Stage 4. Purchasing

At this point, the team can move forward with the purchase. It is important to note that many teams early in their cloud journeys will skip the prior steps and go straight to purchasing. For example, a team will sign up for an RI for one year. However, once you've made the purchase, you can't go back and perform steps 1-3. As a result, teams that skip those prior steps can lose a lot of money.

Conclusion

As IT and cloud operations teams start to work on optimizing their cloud implementations, many find that the efforts required are more complicated than they anticipated. By applying the principles above, you can start to control spending, establish realistic cloud operations budgets, and ultimately reduce costs and waste.