BEST PRACTICES FOR CLOUD OPS SUCCESS

I've been spending a fair amount of time recently speaking with enterprises about their cloud strategy. As I sit in airport terminals, mindlessly people-watching and purposefully avoiding my inbox (what is this "inbox zero" sorcery of which you speak?), my thoughts gravitate to common themes from those meetings. While practically every enterprise already uses the public cloud, they recognize they can be doing better; I mean, it's why they took the meeting in the first place. Struggles with technical and cultural transformations are even more prevalent than I expected going in. Cost is a challenge, security is a raging concern, and automation is needed to remove manpower bottlenecks. Let's throw some thoughts on digital paper on what a successful cloud strategy looks like. I will make every effort to avoid the obvious or cliché, so if I do so here, I beg your indulgence, that the perspective is not only mine but also comes directly from customer interviews. Just looking to share common themes and practical guidance on moving your cloud journey forward. From Gate A24, here goes…

Enable Agility, Don't Hinder It

"Enable agility, don't hinder it." I've been hammering this mantra for quite some time now, and for the first time, in one of my meetings, a security leader led the conversation to this mantra before I could even show a slide with those very words on it. It was like some sort of peculiar magic trick, and we all had a good laugh. (Aside: most of our meetings were conversational, with very few PowerPoint slides, and in most cases, none.) If the governance team – or security, or security and compliance, or a rose by any other name – makes cloud usage difficult, then the scrum teams will simply go around them.

And you already know you do not want that. The answer is not more, but better, restrictions.

Pro tip: do not over-rotate on governance. Cloud usage outside of centralized governance is bad. Very bad. Like an existential threat to your business competitive advantage bad (e.g., exposed IP, fines, brand damage, loss of talent/turnover). If you over-rotate, the scrum teams will just go around you and then you won't know what is out in the cloud. And by the time customer data or IP is inadvertently exposed in an unsecured Elasticsearch service, it really will not matter whose "fault" it is. It was reassuring to hear how receptive everyone was to this theme.

If "enable agility, do not hinder it" is the guiding principle, then the Cloud Center of Excellence (CCOE) will go a long way in effectively supporting the business strategy.

Enable the Scrum Teams to Self-Serve

A few months back, a developer bristled a bit when I wrote that they were responsible for the configuration of their cloud services. He pointed out that the tools forced upon him by the CCOE were numerous and not easy to use. I'm quick to admit when I am wrong. In this case, we were both RIGHT. And I was wrong to not give the weight that ease of use deserves in this cultural transformation. To lead a successful cultural and technical transformation which the public cloud requires, the CCOE needs to deliver fewer complicated, point-specific tools requiring a Ph.D. to use; and the customers – the dev teams – need to be accountable for their own cloud spend and cloud security posture management.

The practical advice I can offer here is to have executive sponsorship from the application services leadership, the CCOE, and the CIO. Executive leadership is helpful in leading cultural change. Plus, if you've chosen easy-to-use solutions that help devs do their job, then agreeing to use those solutions should not be a burden.

Cloud services must be appropriately configured if they are to be secure. The speed and scale of change in the enterprise's cloud footprint is not anything that a centralized security and compliance could, or even should, be responsible for managing. That responsibility rests upon the user themselves. Cloud Operations should not be a traffic cop. Instead, autonomous and self-organizing scrum teams should be able to spin up cloud resources on-demand, provided they have the tools to manage their cloud resources.

Automate Public Cloud Security

And while the developers are spinning up those cloud resources, the Cloud Governance team can, and should, have deployed policies which automatically test that the cloud resources are securely configured. If they are not so configured, then the organization can decide the next step. Should they block deployment to PROD? Probably, unless an exception is agreed upon. Who fixes the misconfiguration? Well, that would be the developers. How do they do it? Doing so should be as simple as clicking a button; that is, remediation should also be automated.

DO use best practice recommendations from the Center for Internet Security as the standard for configuring your cloud resources. DO automate security checks and remediation – doing so will not only ensure consistent cloud resource configurations across your large, growing, and rapidly changing cloud footprint, but also radically diminish risk. DO let the scrum teams own their security posture. DO provide security and compliance policies for the appropriate configuration of cloud resources.

Misconfiguration of cloud resources remains the #1 cause of cloud security failures.

Orchestrate Change Management

While the developers are busy deploying new resources, and/or changing existing ones, it is important to manage this process so that change management does not itself become a manual bottleneck. Automate change request tickets. Have an agile change management workflow which keeps the devs innovating and accelerating along sensible guardrails. DO keep that CMDB updated.

If you can get cloud security and change management working together synchronously and automatically, you will be doing this better than 99% of your competition. Agility and security are not mutually exclusive. It's just hard to imagine this future state, when cloud security failures from open S3 buckets still routinely make the news. Automate cloud security. Orchestrate change management.

Asset Discovery and Tagging Taxonomy

I've been trying to weave this important topic in, but it didn't seem a good time until now. You cannot manage what you cannot see. And from what customers and prospects are saying, visibility remains a big challenge. Of course, a lot of this, but not all, is the unintended consequence (shadow cloud usage) of draconian governance as previously discussed. A successful cloud strategy begins with knowing what resources you have and how they interconnect. Application mapping is crucial, as is enforcing an asset tagging taxonomy. Taxonomy enforcement – gosh, that sounds draconian itself, sorry – is easily codified with compliance policies. For example, IF a resource is deployed without a tag to an approved department, THEN take down the resource AND notify the user. In this case, the developer will quickly add the department tag and redeploy: no harm, no foul. I'd also have a tag for the application, and the reader can easily decide which tags make sense for their organization.

Cloud Expense Management

You are overprovisioned on-premises, so why would you expect to be any different in the cloud? My apologies if you changed your process, recognizing that cloud OPEX charged back to your department for resources not being used is particularly painful. Setting money on fire is generally frowned upon by financial controllers.

A successful cloud strategy will empower application owners themselves to proactively manage their own cloud budget. Machine learning will reveal usage patterns, improve forecasting, and predict when budget excursions could materialize, so that the scrum team can intervene before the cost overage happens. Were I in the CIO's shoes, I would also want machine learning to proactively identify cost savings opportunities – show me WHERE resources are overprovisioned, and how much money we could save. Show me how right-sizing these resources would affect application performance, so my team can factor that into its decision-making. Also, release cloud resources that are not being used so you are not burning OPEX. In short, show cloud cost by application, proactively manage budget, improve forecasting, and proactively recommend cost saving opportunities.

Gamification

Friendly competition among teammates raises everyone's performance level. This is the theory behind gamification of cloud operations management. Since we've established above that cloud ops

should enable agility, and that scrum teams should actively manage their own cloud budget and security posture, let's challenge everyone to raise their game. Why not have a dashboard (or, "information radiator" in the parlance of scrum) which shows how the different app teams are performing relative to each other? You'll identify and share best practices from the highest performing teams, as well as get help to where it will have the most impact. Over time, the leaderboard will read like a bunch of Olympic swimmers: even the one at the bottom is still a top-notch world athlete.

Summary

In short, we need AI to reveal insights buried under a mountain of data and automation to put those insights into action, removing bottlenecks and accelerating agility. Enabling scrum teams to self-serve within sensible policy guardrails and to proactively manage their cloud cost and security will drive the organization to new heights. Friendly competition and sharing best practices will raise everyone's game. Hopefully these themes from recent meetings are helpful to you in managing your own cloud journey. I'd like to hear your thoughts. What's missing? What cloud challenges are your organization facing? What's working well, and what could be better? Tweet me @rickbosworth8 (#cloudops), or even share this post with your thoughts. When we expand the conversation, everyone benefits.