



The Deloitte On Cloud Podcast

David Linthicum, Managing Director, Chief Cloud Strategy Officer, Deloitte Consulting LLP

Title: To cut cloud costs, don't repatriate; optimize in place instead

Description: Rising cloud costs are driving many companies to consider repatriating cloud apps back to on-premise environments. In this episode, David Linthicum and The CTO Advisor founder Keith Townsend discuss why repatriation might not always be the best option. Keith believes that the better choice is to optimize in cloud. To start the process, define what optimization means to you and pick strategies that fit your situation and budget, then get the right team and take things cloud native when you can.

Duration: 00:25:05

David Linthicum:

Welcome back to the On Cloud podcast. Today on the show I'm joined by Keith Townsend, Principal and Founder of The CTO Advisor. Keith, thanks for joining us today, and welcome to the show.

Keith Townsend:

Thanks for having me, Dave.

David Linthicum:

Yeah, it's great to have you on. It's funny, I've been following you on your social media for a long time. You are prolific. I don't know if you're sleeping at all, but you get stuff out there all the time. So, kind of give us the Keith Townsend story. How'd you get to doing what you're doing now? What's your background? What have you been focusing on?

Keith Townsend:

So, it's pretty typical, the first half of my career is spent in the trenches, hands on keyboard, fixing broken servers and designing security schemes at Fortune 100 banks, kind of going through patch Tuesdays and firewall change Wednesdays to the point where I got enough of the political capital behind me that I set out as an architect for a defense contractor. Spent a couple of years kind of doing the whole enterprise architecture thing. Got a little bit burnt out from keyboarding, and I think this might be the first time you heard someone who gets burnt out from keyboarding and decides to become a management consultant.

Worked for one of the Big 4 management consultancies out there. Decided to go back to what us Big 4 consultants like to call industry to take it easy. Discovered that my keyboarding skills weren't quite what they used to be. I had the hard realization that I could describe how to integrate Kubernetes—well at the time containers—and open stack, but I could not tell you the beginning of how to do it technically, like how to actually do it. I spent more time in boardrooms than I realized. So, we rebranded ourselves, The CTO Advisor, because I've gotten to the point where I was no longer necessarily keyboarding but I had this deep technical knowledge that I could then combine with business processes needed to transform IT organizations and spent the past probably seven or eight years having the conversations around how to strategically make these changes and what this means to not just the executive level but to the lay frontline engineer.

David Linthicum:

Yeah, you're having an impact on the industry, that's for sure, saying a lot of very profound things and a lot of people are listening to, including myself, and that's how we got together and became social media friends, and now friends in real life. But moving forward, we have a similar background. I came from a coding background. I was a programmer for a long period of time, wrote books on C++ and object-oriented design, and then kind of moved into the architecture trade and now looking at really advising around technical direction where things are going. And I think that's kind of a progression where a lot of people go with their careers. In other words, they focus on a lot of the details and getting things to work from a very tedious point of view and then getting things to work from a macro point of view. And I think the reality is that there's a lot of good coders around for sure, and, of course, there's always demand for good coders, but there's more of a demand for people who understand the strategy behind the technology and how all this stuff should work and play well together and how you get to an optimized endpoint, whatever technology stack that you're building. Have you found that as well?

Keith Townsend:

Yes, it's really difficult to decipher the forest from the trees, to borrow that analogy. But it's true. If you have deeply technical people and then you have I think a lack of talent to understand, "Should we even deploy Kubernetes? Is this something for us? Does this solve the problem? What is the problem?" Just going back and asking the problem. There has to be a layer in-between to help mitigate between the two needs.

David Linthicum:

So, we're going to talk about something that's near and dear to my heart. We've been talking about repatriation and the reason to do it and when not to do it, and you suggested a topic that I thought was brilliant, the ability to look at optimization in place of repatriation and looking at the tradeoffs in doing that. This is something I don't think people are talking about. In other words, people are kind of making a binary decision as to whether or not they should repatriate some data, looking for lower-cost storage systems, things like that, basically moving them back on-premise, which is a tough decision to make, certainly after you've spent the money in moving them in the cloud. And you brought up something really relevant. In other words, why don't we just optimize the stuff and fix the stuff and refactor the stuff and tune the stuff and make it work better in place in the cloud so it burns less money? It's a more optimized architecture versus just making the decision to drag it back on-premise, and that's certainly a viable option. So, what's your concept of this thing? What are people looking at? How would you define the topic?

Keith Townsend:

Yeah, so it's a carry-on from a conversation that me and you had around repatriation, this idea that, man, cloud is so much more expensive than what we thought it was. And the cloud providers will tell you that cloud on the whole is cheaper than on-prem, and that may or may not be true. The TCO may prove out that cloud is cheaper from an organization perspective, but budgets don't work that way. We've both been at enterprise long enough to know that if IT infrastructure pays for the cloud bill, it's coming out of their budget, and they don't magically get the savings from security or facilities or any of these other—or even hard savings that the cloud providers advertise. If my department's budget is bigger than what it was, my department's budget is bigger than what it was. So, the idea is to repatriate the workloads, but we find that we no longer have that skill to run the workloads off—or on-prem or off cloud. So, we have to ask ourselves: How do we answer the bell? How do we actually save money in the public cloud if repatriation isn't an option or something that we don't desire to do? And the best thing to do is to do what my friends in the cost optimization space talk about, which is optimize in place.

David Linthicum:

Yeah, and I think that's a good option. In fact, if you look at the positives and negatives around doing this, there's some negatives I don't think people really kind of understand. Number one, you have to remember you move these application workloads and these data sets into the cloud, you may have just lifted and shifted them, however, there's an amount of expense and risk in moving that, and now you're going back hat in hand saying, well, we need to move back on-premise because the cost has been way more than we thought in running them in the cloud, and therefore we may have to make a binary decision in changing the platform.

So, you're losing credibility if you're changing directions in the system. And the other point is, if we're moving them back on-premise, we have some risks to deal with there that weren't there when we moved them in the cloud, mainly the skill sets may not be there to maintain them on-premise, and the cost of power and the cost of data center space and the cost of hardware may be much more than we think. So, we could end up kind of jumping out of the frying pan into the fire in the fact that we're running into much more cost issues in repatriating than not. And, so, therefore, it's more of a case to optimize in place.

Keith Townsend:

Yeah, exactly. And in a lot of cases, we're talking to companies that were born in the cloud. I was at a big cloud conference last year. I ran into a backup company that told me that they were able to cut their storage costs by two-thirds by going to a lower tier storage option, basically tape in the cloud, because they were seeing that all the petabytes of data that they were restoring during—or that they were backing up during the day, they were only restoring a few gig per day. So, the tier of object storage that they were backing up to made no sense for their data patterns. So, it's this optimization that doesn't just happen for those of us that's lifted and shifted to the cloud. It's optimization for those of us who might even have been born in the cloud.

David Linthicum:

So, ultimately let's define what optimization is, and I assume that just thinking about it's the ability to kind of look at the core patterns within a particular application and data set and optimize it so it's able to burn less money on the cloud providers and the ability to leverage management systems and performance management systems and best of FinOps to normalize these systems to a much more optimized state, but what's your definition of optimization and what should people consider when they're looking to optimize systems in place?

Keith Townsend:

So, you know, we're always driving for optimization in enterprise IT in general. We're adopting d-dup (data de-duplication) for storage arrays or virtualization to leverage—get the most out of our hardware. So, it's that level of optimization we're looking at, looking at the new services that cloud providers are constantly coming out of, and looking to see if it makes sense to migrate the existing services that we have today to some type of cloud native or native cloud operating model or functionality. For example, cloud-based load balancers were a thing that we deployed ten years ago. Now load balancing is kind of built into the DNS that cloud providers offer, so we still may be looking at the architecture from ten years ago where we had a bunch of cloud instance load balancers, and now it's a given part of the service value add of the DNS server, and that may cut costs quite significantly for a simple web application, or simply for those of us that lifted and shifted. Should we be looking at using centralized databases, cloud databases versus a database in each different cloud instance OS—VMOS? So, optimization is, again, refactoring our applications for native cloud services.

David Linthicum:

Now, who would be on an optimization team? In other words, we're going to leave the code there, the workload there, and we're looking to optimize the code and optimize the data sets. To your point, lots of things we can do, d-dup the database, reduce the amount of storage, physically design the database differently so it reduces the amount of storage and how much cost it burns—expense it burns. And then also rebuilding the application so it's much more leveraging cloud native features, therefore it's going to run leaner and burn less cash. So, who would be part of the team that's going to make the decision and make the changes?

Keith Townsend:

Well, you know, this is the hard part because when we look at it on paper, yes, obviously moving from load balancers to a native service will make sense, or looking at a gaming service if you're a game organization and you have a bunch of different terraforms and cloud formations or whatever automation that you're using to automate the scripting of creation and tearing down environments. If you just use the game service provided by the cloud provider, on paper, yes, why not. But this is where we run into the real problem. Those resources are typically developers, and I mean—and we need to define developers. These are the folks that the business analysts go to with requirements to build a net new application or net new features for business value. There's not enough of them. So, when we're thinking about re-platforming our applications, these are the traditional resources that we go to. I'll put the question to you. How often does IT—traditional enterprise operations—get those folks time to go back and re-platform an application that's no longer adding net new business value?

David Linthicum:

That's going to be tough because it's normally something that most people don't work on. They're either developing net new applications on the latest development platforms, cloud native for example, and we'll talk about that next as an option, or they're basically redoing something completely from scratch. In other words, when you say refactoring sometimes and containerization, that means writing the thing from the ground up. And getting in there and doing optimization I think is a specialized skill. In other words, I don't think when people go through developer school—if there's such a thing—learn how to build systems and deploy systems and build databases, things like that. They may go through some optimization training, but they really don't do—bring that into the DNA of what they do as a job, whether it's on-premise or in the cloud. So, you're looking for a different breed of cat, so to speak, in how we're going to bring skills and someone who can make something cheaper, more efficient, which is going to be a skill set that's part of a skill set that's part of a skill set, and so scarce as hen's teeth, so to speak. We're not going to find a lot of people out there capable of doing that. What are your thoughts?

Keith Townsend:

Yeah, so hence we find the talent typically in enterprise architect or SRE system cyber reliability engineer, these folks that have cross-discipline skills, that understand storage compute networks, security, compliance, some application development skills, some automation skill. So, these folks that can look at the system holistically with a DevOps type of lens and be able to make just enough changes to the application to maybe not completely refactor it but definitely see the low-hanging fruit, i.e., changing the ODBC drivers, getting technical, to point to a cloud database versus a legacy database sitting on a virtual machine. I call that the platform group, like a group who is specifically designed at building platforms that scale across multiple applications. We've formed them in the past in enterprise IT, so think of the groups that handle the centralized databases and message queue servers, et cetera. That group or concept morphs into a group that is armed with understanding both the application development process and kind of the deep knowledge of infrastructure and infrastructure costing. It is an expensive group to have, but if you can think about some of these businesses that have \$4 million a month cloud spends, the math kind of makes sense.

David Linthicum:

It does make sense. That's a lot of money, and I'm finding that's the tip of the iceberg. People are spending way more than that on cloud services these days as they're migrating into the cloud and finding the costs are a little higher than they thought. So, if you're getting into this particular area, I think it's going to be the "it depends" answer, it depends on what platform you're on, the application, how it was built, and the technologies you labor. But I like the fact—it was very profound when you said you need an architecture person who's overlooking everything in the system and looking for obvious areas of optimization.

And I find that it's the 80-20 rule, that 20 percent of the system is burning 80 percent of the resources because it wasn't architected correctly and they're doing something like you mentioned. It could be an ODBC driver where we could go to a native API system which makes us 50 percent more efficient in terms of cost spending or moving a middleware layer that really isn't needed because it was there in the legacy system, people migrated to the cloud just doing a lift and shift and now it's just overhead. It's really kind of making simple—I wouldn't say common sense because I don't think everybody understands middleware layers and efficiencies and tuning around those things, but looking for the obvious things to change to optimize them in place. Am I missing something?

Keith Townsend:

No, you know, simple things like egress charges. I mean, how much of the surprise is egress and knowing the basics of networking to know that I can get a private connection from one cloud provider to another and not have to change anything at the application level but get significant cost savings from having this dedicated circuit?

David Linthicum:

Absolutely. So, we're talking about optimization. We can't leave cloud native off the menu because it's something I hear all the time. The ability to containerize, leverage containerization, container orchestration, moving into a fact we're able to provide these—or change these applications to make them more scalable, work like a cloud. Cloud native can be confusing sometimes. And, also, the ability to become operationalized in such a way that it's going to reduce the cost of maintaining the system, it's going to reduce the cost of securing the system, it's going to reduce the cost of governing the system, and it's going to take fewer cloud resources. However, a great deal of work needs to occur typically to make someone from a more traditional state to a cloud native state. So, what skills do we need to do that?

Keith Townsend:

So, a lot of the same skills, surprisingly, and a little bit more. If you look at the CNC landscape, the folks who define cloud native from an open-source perspective, it is an eye chart. Back in my consulting days, if I would have presented this to—basically to you and said we're going to present this to the customer, you'd send me back and have me working all night to make sure that I distilled it down to something that's simpler. And it takes that level of skill to understand what you're doing. I don't—there's very few, if any, turnkey cloud native platforms that you can just go to a vendor and say, hey, give me a cloud native platform, give me a reference design, and I can deploy that for all of my applications. It's not that simple.

This is not as mature as our VM environments in which we can take a generic design and apply it across all the VM hypervisor providers and reasonably know that that can be deployed. The nuance between each provider, creating a load balancer in one provider is very different from creating a load balancer in another environment versus an on-premise environment. And while that Kubernetes or cloud native part might be abstracted and makes the service easier to consume, from an operating or platform team perspective, someone has to do that low-level transformation, and it's difficult.

David Linthicum:

Yeah, it gets complex quick, and I think even coming up to a common definition of what it means to go to an optimized cloud native architecture is one discussion, and then the tactics you use to get there from whatever Point A is, and of course, we're talking about any applications. It could be applications that are written in C++ twenty years ago to applications that are written in Python ten years ago or some of the newer applications today that maybe AI enabled where you're trying to find analogs in a particular cloud provider and analogs that are in a cloud native capability that work with containers. And that gets into some pretty geeky, nerdy stuff at the end of the day, and there's not a lot of people running around who know how to do that.

Keith Townsend:

Nor a lot of people that want to do it. How many people want to take your ERP and stick it in a container?

David Linthicum:

Few but it's not a lot, to your point. So, finally, what other alternatives do we have? Other public cloud providers, you mentioned that someone found a cheaper storage solution I guess using mag tape, which is interesting to me. And what are some of the alternatives to kind of fall back from instead of moving it back to a traditional system on-premise in a data center, an enterprise data center, but move it to another second tier or third tier cloud provider that may have a cheaper solution that's more optimized for the applications and data that you're running?

Keith Townsend:

So, again this is one of the funny things where, as we talk about that second tier of cloud provider that are specializing in industry type workloads where they're going to provide your VM environment until you're ready to change from that VM environment, and their costs are optimized around that. They're deferring some of that egress charge to get the data in and out. And then even in place, without reoptimizing your applications, we all have applications that have super-low SLA. Why am I paying full price for a VM instance and that SLA for that VM instance where I can use these spot type instances that come with basically no SLA but will serve the SLA of my business applications? This can save you up to 90 percent on VM cost alone, so there are plenty of in-place optimizations of just relooking at the services in the different forms that they're offered in without needing to either move to another cloud provider or change the application architectures, just right-sizing the environment.

One of the other things is we're kind of lazy in the data center. When a developer asks for a 32-gig RAM and 8 VCPUs, we give them 32 gig of RAM and 8 VCPUs, regardless of if that's what the application required. In the cloud, we pay for that lost optimization or that loss of efficiency. We pay for it, where in the data center we didn't. So, these are just simple things, low-hanging fruit as I would call, to reduce the cloud spend in place alternatives to just re-platforming.

David Linthicum:

Yeah, I think that's spot on thinking, and you think about it, too, the number of purpose-built clouds are starting to emerge. We just don't see much of them because they haven't made kind of a market-moving impact, but the rise of industry clouds that are independent from the larger cloud providers that just focus on doing something around retail and healthcare and manufacturing and providing core capabilities there can become a better place to optimize these various systems because we're able to leverage these native capabilities around these verticalized services that are specialized for these industries, and I think that's one way to go. And, also, I mentioned the second tier, third tier cloud providers that may not be the brand name cloud providers out there but may just focus on doing storage as efficiently as they can, and therefore they provide mass storage for information retrieval, AI training databases that may be loosely coupled to these systems.

All these options are out there that people should consider, and I understand there's risk in moving things and putting your trust in another company. Managed service provider's another one, but ultimately these are things we have to consider if we're considering optimizing the system on the particular cloud providers, we may start shifting cloud providers to get to the best optimization. I think that's going to happen in the next ten years, and we're just seeing the beginning of it now with the addition of the smaller players out there that are purpose-built to provide a service around a specific niche, and that's where things are going to go. So, where can we find out more about your company and yourself on the web?

Keith Townsend:

Well, you want to find out more about the company, the website is thectoadvisor.com. We put most of the prolific content there. If you're missing some content, it's definitely on LinkedIn. You can just look me up, Keith Townsend. And on Twitter I'm @CTOAdvisor. My DMs are open, so if you want to talk about how we used internet to help cut egress charges for a publicly traded company that's not an academic organization, that's an example of just thinking out of the box, and I'm more than happy to share the example via DM.

David Linthicum:

Smart people save you a lot of money and follow Keith on LinkedIn and Twitter because he's posting, like, every ten minutes, so he's got some—has some great deep content out there in the great idea, and he's killing it out there in the world of cloud computing thought leadership.

So, if you enjoyed this podcast, make sure to like us, rate us, and subscribe. You can also check out our past episodes, including those hosted by my good friend, Mike Kavis. Find out more at deloittecloudpodcast.com. If you'd like to contact me directly, you can e-mail me at dlinthicum@deloitte.com. So, until next time, best of luck on your cloud journey. You guys stay safe. Cheers.

Operator:

This podcast is produced by Deloitte. The views and opinions expressed by podcast speakers and guests are solely their own and do not reflect the opinions of Deloitte. This podcast provides general information only and is not intended to constitute advice or services of any kind. For additional information about Deloitte, go to Deloitte.com/about.

This publication contains general information only and Deloitte is not, by means of this publication, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor.

Deloitte shall not be responsible for any loss sustained by any person who relies on this publication.

Visit the On Cloud library
www.deloitte.com/us/cloud-podcast

About Deloitte

As used in this podcast, "Deloitte" means Deloitte Consulting LLP, a subsidiary of Deloitte LLP. Please see www.deloitte.com/us/about for a detailed description of our legal structure. Certain services may not be available to attest clients under the rules and regulations of public accounting.

Please see www.deloitte.com/about to learn more about our global network of member firms. Copyright © 2023 Deloitte Development LLC. All rights reserved.