The Deloitte On Cloud Podcast

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

Title: **Intel's Josh Hilliker talks strategies to optimize your cloud architecture**

Description: In this episode, Intel's Senior Director of Cloud Solutions, Josh Hilliker, talks with David Linthicum about strategies companies can use to optimize their cloud architectures to return more value from their investments. According to Josh, optimization is a matter of using optimization-as-code and finely tuning as much of the architecture as possible. To further optimize, it's also important to govern effectively, with policy-as-code helping to automate the process.

Duration: **00:23:54**

David Linthicum:
Hey, guys, welcome back to the On Cloud podcast. Today on the show I am joined by Josh Hilliker, principal engineer, senior director of cloud solutions at Intel. Josh, welcome of the show.

Josh Hilliker:
Right on. Thank you, Dave, for having me.

David Linthicum:
So, what does a senior director of cloud solutions at Intel do? What's the day in the life of Josh like?

Josh Hilliker:
Great question. Boy, so many things. So, understanding what's happening in the cloud native tools, what's happening with our customers in this space from an onboarding and repatriation from the cloud, and then for what partners need and want. So, it's kind of looking at the entire ecosystem, what's available, what's out there. Where can Intel lend a hand to help them on their cloud journey? So, yeah, the day is a busy day.

David Linthicum:
What kinds of tools and solutions, technology, are you normally dealing with on a daily basis?

Josh Hilliker:
As I look at the cloud and see what hardware, software, and availability is there, the toolset around that is: What are my transition points, my activities into the cloud? So, when I look at quality of tools, I'm thinking about, okay, what's the recommendation, meaning whether it's cloud-to-cloud, cloud-to-multi-cloud, cloud vector on-prem with repatriation? It's like what are the recommendation tools? The next step I'm looking at is: What are the provisioning tools, so the creation? Back in the on-prem days, I just absolutely loved provisioning, creating it up from base nothing to full OS, BIOS, everything.

That creation stage still exists in the cloud, is still super-important on how do you do it. What are the BKMs. Then the next area is that whole modernization, changing, updating, refreshing, being mindful of different architectures, mindful of different software packaging when things get updated, when new roadmap items happen. Then the last, and we'll talk a little more about this last one, but it is governance. It's compliance. It's what's really happening. How's the drift? How much drift? What has been changed? And having that compliant. So, for the tools, I'm looking at those four different transition actions to see what's in the market, who's out there. How do we work together? And how do we make it an easier journey for both customers and partners?

David Linthicum:

This concept of optimization, certainly as an architect I use it a ton, talk about getting to the most optimized architecture, which means that we're trying to build as much efficiency as we can in it. It means different things to different people. So, when you deal with optimization, leveraging technology to do optimization and optimizing systems, how do you define it? What does it mean to you specifically in terms of how it's going to be implemented at Intel?

Josh Hilliker:
Dave, you're so right on with respect to—I ask half a dozen customers what optimization means, and I get six different answers on what does it mean, so great question. When I look at the word optimization, I'm thinking about workload optimization, and specifically we've been looking at workload optimization-as-code. So, what is the workload? What is the stack? What is the hardware layer underneath? And then whether that translates from an activity. What we're doing is what are the different tunables? What knobs are available to tune to make it more optimized? So, it's how am I handling buffer? How am I handling cache? How am I handling Q-links?

Basically, all those different tunable knobs and saying, "How do I tune it in based on the workload and based on the use case?" How is it being leveraged for that specific customer or that specific industry? So, when I'm looking at optimization, I'm looking at the workload. What is the workload? Is it a MySQL database? Is it Oracle? Is it MSSQL? Is it in that data family or is it in AI workload? What are all the tunables and how do I dial that in?

David Linthicum:
So, this is kind of a dynamic concept at this point. So, in other words, there's optimization we think about, almost like tuning an engine. So, we're going to put different points and plugs, and make the engine run as best as it possibly can, and therefore it's fully optimized. But workloads are a different beast because they're going to change behaviors and change usage patterns, and change different ways in which we're leveraging data and different ways in which we're leveraging the processor and the network and those sorts of things. Many times while they're running, and therefore the optimization are the tunables, are going to change over time. So, how do we deal with that?

Josh Hilliker:
Great question. It brings up another part of my history as well, which is the robustness of telemetry matters, and knowing your performance matters, and being able to keep track of it with the right level of granularity based on what's happening. So, when we start looking at workload optimization-as-code, it's like, "Okay, but what's the starting point?" What should it be? What are we seeing as the BKC, the best-known configuration?

David Linthicum:
Before we move on, tell us the details behind optimization-as-code. How are you defining that now?

Josh Hilliker:
Optimization-as-code is what are the different configurable parameters. Okay, let me back up. There's service optimization. There is cost optimization and performance optimization. When we say optimization-as-code, we're looking at the cost and the performance element of it. Cost is like: Are you right-sized? Do you have the right configuration and the right hardware platform for the workload? Then on the performance side it's: Have you truly dialed that workload in to operate the most efficiently as possible? Then those all lead to of course a better service to the end customer, whether that's patient care, whether it's booking interface, what have you. It's being able to tun those correctly. So, when I look at optimization, I'm looking at the whole stack, from hardware, software, middleware, runtime, app experience.

David Linthicum:
Listeners are probably asking the question, "I've heard of infrastructure-as-code. So, infrastructure-as-code to optimization-as-code. And what are the differences and what things do I need to consider as I use both concepts?"

Josh Hilliker:
All right. Let's take a stroll through IaC and where we're at today. So, infrastructure-as-code, great movement. We're absolutely seeing more and more companies go after that, where they're codifying the infrastructure so that it's got better predictability, better reliability, less human error, and teams can all work together to revision their infrastructure in a way that I'd say is more—it's more scalable. It's more aware of what they're doing. So, infrastructure-as-code is kind of that first big move. The next big move is really policy-as-code. That's about a year and a half to two years old. So, how do I govern in a whole different way? So, initially, policy-as-code, we had thought a lot about ports and protocols. Am I shutting down SSH? Am I making sure that I'm restricting everything from 0.0.0? Am I restricting all that correctly?

But policy-as-code is really more to do with a different level of governance. Am I applying the right workload to the right infrastructure? Am I applying the right tunables? Am I thinking about this the right way? That's that PaC journey that is, again, about a year and a half or two years old. This is the next progression of writing optimizations into code, and making it so that it's the same kind of repeatable with IaC, executed through PaC, so there's this formal relationship with the two. Now in contrast, when we started the journey on optimizations, we would provide everything in a PDF. It would be anywhere from a 40 to 80 to a couple hundred-page PDF of, "Here are the optimizations. Here's what they all mean," point-by-point, "Here's what that parameter does when you tune it. Here's how you would tune it," and it goes through that. That is not optimization-as-code.

Now taking that PDF, transitioning it to a code set where I can do an automated recipe, a deployment module, and putting those pieces together is what optimization-as-code really is all about. It's like how do I get it in a way that the easy button is there, the ability just to say, "I want this optimized as a starting point," and be able to pull from that right at the gate, not, "Let's go figure out how we tune it later," or, "Hey, I'm having a performance issue. How do I configure it?" No. You're doing it right from the start. That's the migration or this journey we've been on around the "aCs," IaC to PaC to OaC.

David Linthicum:
I've got to ask the question. Is AI applicable here? Is AI helping us optimize systems? If so, how does it fit in?

Josh Hilliker:

Another great question. So, AI. There is absolutely a role with AI in this space. The first thing is the tunable parameters. How we've done it in the past is we have workload owners that really know their stuff. They know what's going on with the workload. They know all the tunables, what I can do inside the software, what I can do for the database engine, what I can do for hardware. We have brought AI in on top of that to really push the system and say, "Okay. What else? What other permutations, combinations, and what is the BKC, the best-known configuration?" So, we've used AI to really tune these parameters in. So, when you look at an optimization-as-code module and a recipe, it's actually the output of using both the human startup and AI to make that possible.

So, as we're looking at new workloads, we're using AI to help us with what are the tunable parameters. How much do I tune it? What are the ratios? What should that configuration be for the different level of services, the landing zones? Am I doing bare-metal-as-a-service? Am I using IaaS? Am I using PaaS? Et cetera. It's like where is the landing zone? What are the tunables? And AI plays a huge role in making that possible.

David Linthicum:
Right. It's able just to look at a lot of information and tell us the gist of it. Therefore, some people who are doing the optimization of code, it just makes that job easier?

Josh Hilliker:
Yeah, it does. For those that are tuning, it helps. Let's say we start with a tuning place, a workload expert. It comes in and fine-tunes at a whole, I'd say, deeper breadth of tunable parameters. The amount of combinations it can run is humanly not possible. It would take us months. Instead, it's minutes and hours, and say, "Okay. Here's your best-known configuration," and we apply that into the optimization-as-code.

David Linthicum:
What are business benefits? How do we explain this to a CEO or a CFO or people that are concerned about, "You're going to spend so much money on doing this. What kind of business benefits can we realize from this activity or from this concept, from moving forward with these concepts?"

Josh Hilliker:
I believe that in the cloud space we are also moving through the big shock and awe of how much does it cost. How much does this service really cost? Oh my gosh, it auto-scaled and that now 2x'd, 3x'd my bill. So, how should I be running the cloud? So, from a business standpoint, it's like: How can I move my development team faster, keep them going on the product front, but optimizing my infrastructure so I can lower cost and increase performance for them? That's where, to me, optimization-as-code fits right in. It's like we can help with the development time, help with the automated recipe, help with deploying the model, be able to give you things where you're not reading a bunch of documentation, and you're not seeing all those development costs. What I like to say is that if the level of effort is greater than the ROI, why do it?

And that is where optimization-as-code fits right in. We're going to help decrease the level of effort, so you can focus on your time to market of your product. You can focus on other use cases of your product because we're giving you the ability to get the easy path or easier path to it. There's that benefit right there from a C-suite standpoint of saying, "Hey, it works this way." So, to me, that is number one. The other thing is that for CFOs that are saying, "I've got a mature business and it's going well. I've already got all this infrastructure-as-code. I've already got my DevOps, and I've already set this function. I don't see how you guys can help." Well, guess what. In optimization-as-code we have got the right brown-filled hooks, where you can take current infrastructure and not redo it, but add in those optimizations and be able to get to that next state. So, again, back to that shock and awe, the squeeze, the bill. It's like, "No, you can see this will improve performance."

This will lower your cost even in brown-filled. As I was out talking to customers this last week, some of the things I heard was optimization automation is invaluable for the industry, from a C-suite player. They said, "This is where we're going. This is where the industry is going. This is where we are headed, and it's great to see it going down this space." Now another part, back to the CEO and the CFO, the world of provisioning and creation in the cloud is starting to migrate even more to machine-to-machine communication, where machines are talking about placement. Machines are doing it. They're setting up. They're repositioning. They're reprovisioning. They're making sure that auto-scaling is correct. And preparing ourselves for that is super-important.

So, as Intel we're providing the building blocks, like, "Here are the modules that will help you when that machine-to-machine communication gets to that 100 percent marker. This will help you from a scaling correctly, scaling back correctly, being able to understand what the next architecture is and the immediate benefit of the next architecture." Because in a lot of cases it's going to cost you less to move to the next architecture. That should be something that you don't have to think about. It just works. It just does the job for you.

David Linthicum:
Yeah. I think there's a pent-up demand right now. If you look at 2022, it was one survey after the other where people were just very shocked at what they were paying for cloud-based systems and even some infrastructure—I mean some on-premise-based stuff as well. So, in other words, we moved massive amounts of things during the pandemic and prior to that, and say 30 to 40 percent of the applications and the workloads existing on the cloud, but the bills that came back from these systems was way out of whack from what people expected. If you look at it, it was more self-inflicted, and a lot of this stuff was grossly under-optimized. Do you think that's a use case for this, the ability to get those things under control?

Josh Hilliker:
Absolutely. And grossly under-optimized is absolutely right on par. It is vanilla configuration as of yesterday. You had to dig in a little bit more. It's not just recreational digging in, but digging in because of that cost benefit, because of the performance benefit. We live in a two-pronged world. It's cost/performance. I want great performance, but I've got to look at cost. So, yeah, for how services were run, how they were configured. One of the most kind of frustrating points that customers say to me is, "Hey, I auto-scaled and I didn't need to." It's like, "Yeah. And how much did that cost you?" That shock. Now as we went through the pandemic, that push to the cloud and there was this—I don't want to say Wild West, but it was like, "Hey, everybody go and go fast," because we've got employees at home.

We've got new customers. We've got new demand. How do I now rein that in? And that's where governance, to me, plays a big role, the policy-as-code, of having that relationship of policy-as-code to feed into policies, and policies to be able to go look at infrastructure. Because as an executive, the question is: How bad is it? Yeah, as to the money part, but how bad? Are we talking six months to go reconfigure and reoptimize? Are we talking two years? Where are we at? That's where that policy-as-code really helps out with, "Hey, give me a scan. Tell me. Okay, cool, got it. A one-month project and we can save 50 percent. Let's execute it." That's governance, policy-as-code coming back in.

David Linthicum:
That's what I love about this. It allows these workloads to be self-optimized. In other words, to be autonomously optimized. They're able to take care of themselves. They're able to improve their efficiency internal to the application, so it's not another process that's acting upon it. It's the ability for it to make a decision based on what it's doing, and the applications are going to know what they're doing better than any human being. There are systems that are surrounding that to make the decision in terms of how we need to optimize these systems, and how we're leveraging policies to kick these things off, and AI capabilities to have some ability to learn as we go, and make decisions based on better data and understanding the different patterns of the data. So, what are some of the use cases that you see out there, where this is kind of killer for optimization-as-code? How are people leveraging this? What kinds of applications are leveraging this?

Josh Hilliker:
Top workloads, I call them the high flyers. Those top workloads are things that are in the data sector. So, we talk about databases. We talk about open-source on the MySQL, Postgres. We look at things like MSSQL and Oracle from commercial. These are the ones that are heavily being used in the cloud, and we've got optimizations on the shelf and, for most cases, not being leveraged completely. So, again, the perfect opportunity is like that database, how you use the database. Are you doing key-value pairs? Are you doing booking reservation? Are you doing user management for logins for websites? A lot of those use cases still resonate and hit, but it's that database where we can come in and pick those optimizations up and leverage them quickly.

In some cases, it's just something that we've up-streamed. We have put it in the code. It is available, but it's a question of: Are you selecting the right version? Are you selecting the right version on top of the software and on top of the hardware stack? So, a lot of it is that: What is the recipe? What is that configuration? What is that gold configuration of how to run? So, database is the top of the list. Next down is this little workload called AI, just a little one. That was a joke, sorry, but the little workload around AI. We look at things like: How do I do ChatGPT? How do I do it quickly? How do I do tomorrow, if not today? Also, stable diffusion for imaging, ChatGPT/FastChat has been top of the list of, okay, I want to do this and I want to dabble. How quickly can I do it, and how quickly can I show it? What do I need to go fine-tune the model, so that I can start using it inside of my company?

That's when all those, from an optimization standpoint, now you're picking up part of the hardware, part of the accelerators that are available in the market, part of the libraries that have been optimized, part of the code language. Runtime has been optimized, and the software stack, even additionally above that, which is the UI. So, those two are kind of the top flyers. So In optimization, there's got to be automation, and there's got to be that closed loop. That's one of the underpinnings of optimization-as-code. It's not just good enough to have tunable parameters. I've got to be able to set up that closed loop, where my infrastructure is telling me stuff and I'm able to adapt to that.

A new phrase, we've been kicking this conversation around automated instance selection for: How does your infrastructure pick up that real-time telemetry? How does it use it to make decisions about where your infrastructure should be, and what type of infrastructure it should be on? Then it automatically takes care of that and readjusts itself on an ongoing basis. That is truly the win at the end of the day for optimization-as-code. It's just I'm constantly learning, listening, changing, and reflecting that back without human intervention.

David Linthicum:
Where can our listeners learn more about you and also this topic in general? Where can we find more information about optimization?

Josh Hilliker:
A couple things I would go out to Intel.com. We have our /cloud, so Intel.com/cloud. That will take you to our cloud tools and what we have available, from our developer tools, our business tools, our different transition tools. So, you can go out there. As I said, all the tools, technologies are listed there. Also, if you're interested to hear more as I dig into automated instance selection or optimization-as-code, go to Cloud TV. It's on the Intel Partner Alliance. It's Cloud TV Program. I'll be deep diving, along with one of my peers, Sarah Musick. She's a cloud solutions architect as well. We talk through very specific things, what the value, the benefit is. Why would you want to do it? And then a little bit about the code and the technology underneath it. So, join me there or jump on LinkedIn and add me, and let's talk more about optimization-as-code, automated instance selection, and where this is headed.

David Linthicum:
Josh, this is a great concept, because if we need anything right now it's the ability to look at this in a scientific way, in a repeatable process way. Everybody wants optimization, and the ability to do it in the proper way, and the ability to build it into your systems is going to return a huge amount of value, just kind of based on the fact that we're dealing with very under-optimized systems right now. They're eating too much money. They're not using resources. You recommended generative AI and some other capabilities. Your ability to optimize these things in a dynamic and autonomous way is really going to be what's critical to the success in getting this stuff up and running, and having it operating so it brings back the eye to the business. 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, all one word. If you'd like to contact me directly, e-mail me at dlinthicum@deloitte.com. So, until next time, best of luck on your cloud journey. You guys stay safe. Cheers.

Operator:

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