



## The Deloitte On Cloud Podcast

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

**Title:** Solving the cloud-cost conundrum to drive higher cloud value

**Description:** Despite cloud's many benefits, organizations are often shocked at its operational costs. In this episode, David Linthicum talks with Erik Peterson, co-founder and CTO of CloudZero, about how companies can leverage software engineering and focus on costs to optimize cloud spend and drive value. According to Erik, success lies in cooperation between finance and engineering to better understand the financial impact of cloud and close the feedback loop in terms of mapping costs to features.

**Duration:** 00:22:45

**David Linthicum:**

Welcome back to the On Cloud Podcast. Today on the show I am joined by Erik Peterson, co-founder and CTO of CloudZero. How are you doing Erik?

**Erik Peterson:**

I'm doing excellent, David. How are you?

**David Linthicum:**

Great, and I was just reading your bio here. So, kind of catch us up with what you've been doing lately, what CloudZero is, and kind of how you evolved to get into this space. Love to hear your story.

**Erik Peterson:**

Oh, there's always a story of evolution, right? Like any good origin story, I think everything starts with a little pain and suffering. I think probably yourself and many of our listeners have kind of lived through that transition from the datacenter to the cloud. And I was on the edge of that in my previous company. I was given a really ridiculous budget of about \$3,000, and I became a cost-obsessed engineer. And as I started to build out some stuff, I realized, man, we are missing a key metric when we think about what we're building in the cloud, and there's nobody really talking about how every engineering decision is really a buying decision, and there's no solution that's really solving that problem. So, I became super obsessed with that and I ended up founding CloudZero in 2016.

**David Linthicum:**

Yeah, what's interesting about this is it says, "cloud cost operations solutions for DevOps." And you normally—even though there are cost checks that you do within a DevOps toolchain, you typically don't hear the two concepts leveraged the same way. And, so, what do you—do you test for cost efficiency or are you looking for better ways to utilize resources, things like that?

**Erik Peterson:**

I mean, there's a lot of different ways to slice it, but the kind of fundamental truth to everything that we're doing today is that your engineers, the folks who are making the architecture decisions, who are writing the code—every line of code essentially is a buying decision when you have the ability to spin-up resources with an API call or a CloudFormation template or a Terraform template, that whoever wrote that made a buying decision. And the kind of key observation to me was that nobody was managing the feedback loop for those buying decisions. There was no software solution or really anything in place to guide people to make better buying decisions.

And when you look at cloud costs today and just all the buzz about, oh, my gosh, I can't believe they're spending this much money, or why does it cost so much, I think it's obvious, why it costs so much. Imagine you're writing code and you didn't get any feedback that you had bugs until the end of the month, right? You'd have some pretty buggy code. And that's exactly why we have some pretty expensive software running in the cloud, because we haven't figured out that feedback loop.

**David Linthicum:**

Yeah, and I think we're going to focus more on this. I just posted a blog today on InfoWorld that talked about autoscaling and when you should use it and when you shouldn't. And the reality is everybody seems to turn that on, or leverage serverless computing where the cloud, in essence, makes the decision for you in terms of how many resources you need to attach to a particular application based on heuristics and different behavioral points and things like that. And the reality is we may need to have control over that.

And, ultimately, if we're turning on autoscaling and we're getting a bill which is \$20,000 per application, and we could probably get away with \$1,000 per application—that's actually a real scenario that I saw—it may make sense to apply some more disciplined software engineering to us configuring the resources specifically rather than letting the cloud autoscaling system and serverless features do it for you. Is there a movement afoot to kind of be a little bit more disciplined, a little more cheap, I guess, in how we're using these resources efficiently?

**Erik Peterson:**

Yeah. There are a couple things out there that are pushing on this. There's some pretty interesting stuff going on over at the FinOps Foundation and what they're trying to push in terms of getting financial concepts to be a part of the software development process. But there's a couple of things that are actually going on in this world, right? The first is that as people start to build, they think about—the typical software engineering processes is, "let's use all the resources available in order to solve the problem," right? And every software engineer I've spoken to, they love solving problems, they want to solve these problems, and they want to use the resources available to them. But we've dropped them into this environment which has pretty much infinite capacity, and the thing that we obviously don't have is infinite wallet. I don't know if you've figured that one out, but if you do let me know, because –

**David Linthicum:**

What? You mean there's not an unlimited amount of money?

**Erik Peterson:**

*[Laughter]* Yeah.

**David Linthicum:**

That's not a real thing? What's going on?

**Erik Peterson:**

I mean, who knew, right? And I talked about this at a QCon presentation I gave, gosh, maybe two, three years ago where I put forth this concept called denial of wallet, right? From a security perspective people needed to start thinking about this, because you sit down and start building and your goal is to build this super-elastic, super-scaling system, but you kind of forget that limits and constraints are actually really a powerful thing, and sometimes they're necessary to create stable systems. But nobody was putting any kind of constraint on the costs, and so now you've kind of replaced the denial-of-service problem with the denial of wallet problem. So, that's one problem.

The other one, though, is just this feedback loop I was talking about, right? Software engineers, they make these decisions, they don't have a feedback loop, and if you think about what it means to build efficient software or well-architected software in kind of this cloud era, if you don't have some sort of metric for efficiency in terms of what you're doing—and you can think of it in terms of impact to the world from a climate change perspective. You can think of it as impact to your wallet. You can think of it as impact on the resources available for other projects in your team. If you don't have some sort of feedback on efficiency, then you're not necessarily building good software for the cloud.

And a lot of people get tripped up by this cost number. I encourage engineers a lot of times to just ignore the dollar sign and treat that number as an indication of that efficiency in terms of what you're building. Make that part of your process, and you'll find that your software's more efficient and actually runs faster typically. You have better outcomes. And, ultimately, you've got a more profitable company coming out of that process as well.

**David Linthicum:**

So, I come from a world where we had to write everything efficiently. In other words, when I first started in computing—I'm old, so it was you had a limited number of resources you could leverage. Limited number of storage, limited number of RAM, and the stack was a very tough thing to manage at the time. And, so, efficiency was kind of built into that, in other words, when you leveraged the systems. The engineering cultures that I see, a lot of that's kind of gone out the window. Even when we've owned the hardware and software, we kind of consider that as commodity, scalability, that kind of thing. Now we've moved to the cloud, we view services as fairly chief, and my time I'm going to waste, and—they view it as wasting. Some—most—some do, most don't—in making this application more cost-efficient is going to not necessarily be a blip. It's going to be around in a year; therefore, I can go ahead and do it. So, what do you say to them?

**Erik Peterson:**

Yeah. I mean, I think what you're talking about, David—and, I mean, we came from that same world where I prided myself on building efficient software. And the core of it was—what you're talking about there is the power of constraints, right? We had constraints. We had only so much RAM. We had only so much disk. We didn't have a million servers that we could just call into being with an API call. We had to kind of make do with what we had. In fact, you look at some of the most elegant systems in history and they had amazing constraints. Like take the Apollo 11 guidance computer. It's unbelievable what it was able to achieve with, like, ridiculously small amounts of resources, so it's probably one of the more efficient systems ever built.

But now we're living in this world where the constraints have gone out the window, right? The only constraint for a lot of people is time, and so they are literally trying to buy time with money. And I kind of object to this notion, though, that it's cheaper, right? I think a lot of people started their cloud journey thinking that it was going to be cheaper, or they saw the prices and they go, "Oh, I'm only going to pay ten cents an hour for this," and they didn't realize that it all ultimately adds up. It's almost like we're all collectively bad at math. And—because it ultimately isn't cheaper. You're still paying for it, right?

And whether you bring that constraint into the environment in the form of a budget, which is kind of an anti-pattern sometimes for engineering—like, folks don't want to have a budget. I had a budget when I first started building on the cloud, and I found it actually really empowering for this kind of focus on developing with constraints. But I think the big trend is people are starting to think about unit costs, right? What does it cost to deliver value? What does it cost from even an economic perspective? Like, if I'm delivering rides, my software engineer should be really crisp on what it costs to deliver a ride. Or what does it cost for a flight, to book a flight? Or I'm a bank; what does it cost for a transaction? Understanding the cost to deliver value is this missing metric that I think the cloud, and cloud engineering in general, has been lacking all this time.

**David Linthicum:**

Yeah, and I think that it is a bit of a change in thinking. By the way, just to go on the record, I never said cloud was going to be cheaper, so don't put those words in my mouth. Not that you did, but some other people have over the years.

**Erik Peterson:**

Yeah, they have, for sure.

**David Linthicum:**

I love the fact that we're thinking in terms of value units and we're thinking in terms of value, because some of the things that—if we look at some of the costs, the cloud may be cheaper, may be more expensive. You're typically going to get more for it. But if you're able to deliver the value back to the business with the utilization of those resources and spending the cost, then you can justify the cost. And there doesn't seem to be visibility into that. You get into—and the reality is that the value of cloud is soft values, meaning it's the value of innovation, value of agility, value of speed to market value of leveraging more innovative technologies because everybody's spending their R&D budgets on building cloud-based products now. How do we do that? How do we gather the metrics and how do we have an ongoing understanding of how much value say a particular application is delivering that's running in the cloud or maybe a suite of applications or a system? What do you think?

**Erik Peterson:**

Yeah. There were two kind of key challenges that CloudZero set out to solve when we were first getting off the ground. The first was close this decision loop. I'm a big fan of OODA loops: observe, orient, decide, act. What's the loop between making an engineering decision and getting the feedback that it was expensive and being able to make a better decision? So, that was the first problem.

And then the second one, though, was how do we better understand the cost to deliver value? Because that is ultimately going to be the metric that determines if I'm building a sustainable piece of software or not. And, so, a lot of people are starting to realize that that's a missing piece and they're asking the same question, right? How on earth are we going to achieve this because it's kind of complicated? Or worse, I talk to some people and they think that they're already achieving it and when you get into the details, you realize they're actually fooling themselves, right? Everybody has that spreadsheet that's laying around where they've done some funny math, and they've basically glued it together and said, "Oh, okay, I've solved my problem," but they really haven't.

And the answer to solving this problem is, one, start with the data that you actually have. It exists. How much stuff is costing you. You have your cloud bill. You have that cloud data. If you're a SaaS company or delivering any sort of value on the internet, you have also information that tells you how many daily active users you have. If you're a message-processing company, how many messages are you processing, right? That data I guarantee you is available to you.

Now the hard part in all this is how you glue that information together, and this is where people who think they're doing this get it wrong, right? They take that, they drop it into a spreadsheet, and they just do a straight divide, right? Total cost divided by number of active users, or total cost divided by number of components. And this misses the big, complicated, unsolved problem, which is that nobody is mapping their cloud costs back to the business units, the features, the teams, the components within their environment.

So, you've got to spend some time actually mapping those costs, because honestly I don't think anybody cares about how much EC2, you're spending or you're spending on EC2. You should care how much you're spending on feature one or feature two. And if you can do that, then you're in a position where you now can take that data that you have number of transactions per second or number of users per second, and use that against that data source to get a more accurate picture of who are your most expensive customers, what's your most expensive transaction, what's the most expensive query. Put it all in dollar figures. Take those performance metrics. Like, get that dollar number associated with it so you can get at the unit economics of the systems that you're building.

**David Linthicum:**

So, how are we getting to value with this? I mean, that seems to be the harder gap to look at. So, in other words—and to be honest with you, I've been wrestling with this as a problem since I got into cloud 20 years ago. And in other words, we're looking to get to something that's going to provide additional value to the business, and I always say we have hard costs and soft costs. Hard costs are easy to understand and you just mentioned some of them. Usage cost on compute, usage cost on storage, whatever public cloud provider, things like that, so we know those and it's pretty easy to find those out, and how many resources we use, things like that. As far as how those resources track back to some sort of soft cost or some sort of soft value that we have within the enterprise seems to be the harder problem to solve. And, certainly, if you look at cost governance and some of the FinOps processes, it gets into cost control. It gets into understanding cost monitoring and the ability to look at what's going on, have observability into cost, and all that cool stuff.

**Erik Peterson:**

Yeah.

**David Linthicum:**

But I rarely see people mapping this back to really kind of soft values that are a little difficult to understand.

**Erik Peterson:**

Yeah. The first—I would say the first kind of generation of trying to understand your cloud costs was really focused on helping people try to buy things better: reserved instances, savings plans, maybe an upfront commit an enterprise discount or credits, or any of these things that can try to lower the cost. And those processes happen outside of engineering, right? And it's no kind of wonder that people started here first, because when finance first started trying to understand cloud costs, they looked at it and said, "I don't understand what EC2 or compute or storage—I don't understand any of that stuff. What can I do to just buy better?" It was a procurement problem, right? So, all of that kind of, like, advancement or kind of savings has been kind of squeezed out of the sponge already, and now here we are going, "Hey, I still don't have understanding of the value that I'm getting out of this. How can I make sure that I'm not wasting money? How can I make sure that I'm not making bad decisions?"

And, so, the finance team has now gotten themselves into kind of the second generation where they're trying to understand what's going on, and they're never going to be successful, right? Finance is an enormously complex discipline; software engineering is an enormously complex discipline. Trying to glue those two worlds together is tough. But you can create a common language between these two groups that I believe is really tied into the value or the unit economics that you work in.

But it's important you don't just stop with the cost value. Like, all right, it costs 20 cents per transaction. Well, that doesn't give me enough information to actually know if that's a good or a bad thing. I need to bring one more data stream into that, which is what's the revenue or the subscription revenue or some—the money that I'm making from that? And make that part of the understanding so I can get at what most people want to get at, which is what are my gross margins? What's my profit margin in all of this? Because if I understand that it costs 20 cents per message to send as an engineer, and then finance provides their part and says, "Well, we only charge five cents per message," then I don't have to think too hard to realize I've got a problem, that if I don't solve this, the company is going to be out of business soon.

**David Linthicum:**

So, moving forward, where is all this going? In other words, we're paying more attention to cloud costs. We think cloud's too expensive, which is probably true. We're finding we're not monitoring the costs, or we don't have a lot of discipline around cloud cost consumption and usage and usage tracking. That's where CloudZero approaches the problem. I kind of like the fact that it approaches from proactively checking things before they're deployed. And, so, is it going to be focused on monitoring and management, cost of certain capabilities? Is it going to be focused on efficiencies in terms of building software? Are we going to get into auditing software, so looking at stuff that was migrated probably with a lift and shift approach which may not be efficient? In fact, I can probably guarantee if you did it that way, it's not going to be efficient.

**Erik Peterson:**

That's the most expensive way to get into cloud for sure.

**David Linthicum:**

Yeah, it is. It is. It's the cheapest way to get into the cloud from moving the application, but they don't really factor in the operational inefficiencies and getting to the cost inefficiencies moving forward. So, all these things we can focus on, certainly the rise of FinOps as a concept and the rise of cost governance and all different technologies we're using to control cloud costs and monitor cloud costs and have observability around cloud costs. Over the next few years, what are we going to be focused on?

**Erik Peterson:**

Yeah. I mean, we're going to be focused on building better with cost as an efficiency metric and broken down by the value to the business, right? What am I delivering? We're going to be thinking about how every engineering decision is a buying decision, but it's not going to be just looking at the big topline number and kind of wondering what's going on there. It's going to be more focused on how can I use this to build better? When you think about it, if I understand that my code runs faster than your code, we can all kind of collectively agree, oh, okay, it runs faster. That sounds better. Right? If it runs cheaper, that also is a really effective way in describing the quality of any kind of cloud software-developed product.

And, so, I expect that teams are going to become much more obsessed with this as an operational metric, as a development metric, that FinOps is going to become part of the DevOps process. It's all going to be subsumed. It's actually kind of amusing to see how FinOps as a practice—I mean, there's over—I think in just the last year, that organization has gone from maybe a few—just shy of a thousand to—I think it's approaching now six thousand people, and it's growing every day. It's kind of growing like wildfire, and it's DevOps people and it's finance people, and it really is this combination of these two worlds coming together trying to figure out how they're going to operate in this new model.

Because at the end of the day, I actually really—as much as I love to talk about cost here, I think we need to have a much more strategic conversation, which is my cloud bill is actually not a sunk cost, or it should never be a sunk cost. It's an investment. And if I'm making an investment, I better make sure that I'm managing that investment. I'd better make sure that I'm getting a return on that investment. And, so, I see a lot of the platforms out there—CloudZero's platform is I think leading the charge here in getting people to think about this as not cost but how can we engineer profit out of this and manage the return on that investment?

**David Linthicum:**

Yeah, an engineer ultimately finding value and the ability to—also an understanding of how the way in which we engineer software has consequences. The way in which we leverage resources from these systems has consequences. I mean, not only cost, but sustainability, because we're leveraging more hardware assets and need more things that generate power and things like that, and in other areas—the cost moving forward. So, where can we find out more about CloudZero on the web?

**Erik Peterson:**

So, obviously you can find us at CloudZero.com. We've got a ton of great information there, lots of things to link to. Probably the most interesting is to go and take a look at our customer success stories. We've got a ton of them out there, great people who've figured out how to get a good return on their cloud investment using our solution. And I'm super active on Twitter. You can find me at @Silvexis, Twitter or @CloudZeroInc for CloudZero's Twitter.

**David Linthicum:**

You'd better spell that.

**Erik Peterson:**

Oh, yeah. So, @Silvexis has been with me a long time. I'll give you a little bit of a story on that real quick because it's—it's kind of a hidden detail. That was my Dungeons & Dragons name from almost, like, 25 years ago, and somehow it's stuck with me this whole time. And nobody can spell it, so it's Silvexis, S-I-L-V-E-X-I-S, and I probably just completely revealed myself as an old-school geek. *[Laughter]*

**David Linthicum:**

Well, that's who listens to the podcast. You'll be fine. Anyway, this is a great conversation. I can't emphasize enough the fact that we need to be more efficient and effective in how we leverage software in the cloud. You've got to remember it's a utility service, very much like electricity and water and things like that, and we're going to pay per use. And our ability to bring efficiency to this equation is going to provide us with an unlimited benefit, and we certainly set up these sorts of automated techniques to monitor costs and to make sure we're putting efficiency into the system. They're going to pay back very quickly.

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](https://DeloitteCloudPodcast.com), all one word. And if you'd like to contact me directly, e-mail me at [DLinthicum@Deloitte.com](mailto:DLinthicum@Deloitte.com), L-I-N-T-H-I-C-U-M. So, until next time, best of luck with your cloud journey and everybody stay real 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](https://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](https://www.deloitte.com/us/cloud-podcast)

As used in this podcast, "Deloitte" means Deloitte Consulting LLP, a subsidiary of Deloitte LLP. Please see [www.deloitte.com/us/about](http://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](http://www.deloitte.com/about) to learn more about our global network of member firms. Copyright © 2022 Deloitte Development LLC. All rights reserved.