Deloitte.

For Cloud Professionals
August 2019

Architecting the Cloud, part of the On Cloud Podcast

Mike Kavis, Managing Director, Deloitte Consulting LLP

Title:	**Build the future with scalable, resilient cloud architectures**
Description:	In its early days, the cloud functioned as a virtual datacenter. However, as the cloud matures, the imperative is to make using the cloud easier, more reliable, and more accessible. Consequently, cloud strategies are highly-focused on architecture. Infrastructure as Code, with its ability to help companies automate the software development process and enforce security and policies, is quickly becoming the backbone of highly scalable and resilient cloud architectures. Other technologies, such as Kubernetes, are also being widely adopted to take advantage of the cloud's promise, but companies have to take care to manage complexity to achieve value. **Disclaimer:** As referenced in this podcast, "Amazon" refers to AWS (Amazon Web Services) and "Google" refers to GCP (Google Cloud Platform).
Duration:	**00:23:29**

Operator:
The views, thoughts and opinions expressed by speakers or guests on this podcast belong solely to them and do not necessarily reflect those of the hosts, the moderators, or Deloitte. Welcome to Architecting the Cloud, part of the On Cloud Podcast, where we get real about Cloud Technology what works, what doesn't and why. Now here is your host Mike Kavis.

Mike Kavis:

Hey, everyone. Welcome back to Architecting the Cloud Podcast where we get real about cloud technology. We discuss what's new in the cloud, how to use it, why to use it, but we do it with people in the field who are doing the work. I'm Mike Kavis, your host and chief cloud architect over at Deloitte, and today I'm joined with Joe Duffy of Pulumi. Joe's the founder and CEO of Pulumi. Joe, tell us a little bit about your background, what drove you to found Pulumi, and what's going on in your world today.

Joe Duffy:
Yeah, thanks for having me, Mike. It's great to be here. My background – you know, I started – you know, mostly on developer tools. You know, I was at Microsoft for 13 years before starting Pulumi, early engineer on the .NET framework, worked on operating systems and Visual Studio, and ended up managing all the languages groups and helping with the .NET open source effort. And, so, my background is really about making developers as productive as possible, and I've seen sort of the evolution of developer tools over the years. And I was really excited about cloud because, you know, cloud is really changing everything about how we build software. And, so, when I was looking at doing a thing of my own, starting a company, I really – you know, a lot of the innovation in Docker, container, serverless really caught my attention and made me excited.

The challenge that I found, though, is when I came to the space and actually started talking with practitioners and getting my hands dirty building software, it was clear that the tools and techniques we use to this day really treat the cloud as an afterthought. It's not woven into the fabric of how we build software. And it's clear, especially when we move to more serverless or container-based workloads, really it needs to be deeply engrained. And, so, with Pulumi we made a few fundamental bets. You know, we wanted to use a lot of the same developer tools and techniques around programming languages, building abstractions and being able to reuse them, how we architect things, how we test things. Basically, the whole spectrum of software engineering, we wanted to bring that to build in cloud applications. And that sort of led us to where we are today.

Mike Kavis:
Yeah, really cool stuff, and the first topic – we're going to kind of talk about the cloud journey. We've both been in it for a while, and I think when cloud first started to rise, people treated it kind of like a virtual datacenter, right? So, they kind of – their development practices or operational practices were the same as they've always been; just now they could fire up a server instead of rack and stack, it and really didn't get most of the benefits of the cloud. And then as groups mature, now infrastructure code has kind of become a competitive advantage. So, what I wanted to ask you is, what has your experience been when you've seen customers and friends and peers working with the cloud at the beginning to where they are now, and what are some of the gaps in making that transition easy?

Joe Duffy:
Yeah, I look at sort of the transition of how we get to where we are, exactly what you're saying. Where originally it was, you know, we had Java applications or three-tier applications or what have you, and then we had an IT organization who would then go put them in VMs for us, right? And in that world, this separation between operations and development kind of made sense, right? It was – you know, you'd have three VMs for an application. I mean, so it made sense to really think about each one as its own independent thing that was configured and managed by a different team. And when you go to the cloud that same model can work, right? So, that's what a lot of people will start with. They'll basically just take their VMs from on premises and use an EC2 instance instead of a vSphere on prem instance.

But now you get into things like, hey, the security perimeter is different. I need to think about networking and security, and that gets you into some of the more fine-grained aspects of managing cloud resources. But you know, that starts to even break down. The VM model starts to break down when you want to modernize. You want to have more elasticity. You want to have more scale. That fundamentally changes the way you architect things, and now you actually need operations teams and developers working together to figure out how to build scalable, efficient cloud software. And that demands a lot more engineering rigor. You can't go into the console and point and click. You know, and it's also the notion of repeatability. It's not just scalability within one region. We work with a lot of customers that have to be geo-distributed across the globe, multiple production environments to manage, and at that level of scale, the sort of old school manual techniques really don't work anymore. We need automated techniques that have – not only more efficient for engineers to work with, but also have things like policy in place so you don't accidentally make a mistake, particularly when it comes to security, like accidentally opening a database to the internet. You know, these are things that are easy mistakes to make that automation can help prevent.

Mike Kavis:
Yeah, and I've been exposed a few places where because of that risk, the platform teams lock things down so much that a lot of the agility of doing development gets kind of stolen from you, because you have to go through these processes. And I think – and I just had that conversation today with someone on the platform team, and it really comes down to, "Yeah, Mike, you can do this, but there's a lot of people that can't." And, so, until we get that trust where we can consistently not release stuff with gaping security holes like that, it's a challenge. So, how – I think you mentioned earlier about abstractions. I think that's the answer, you know, taking some of the plumbing – you know, building stuff at scale has kind of been commoditized over the years, but you have to layer it with all your company's policies, and those are unique, right? So, I think that's part of what you guys do, but talk about the importance of abstractions in cloud platforms.

Joe Duffy:
Yeah. Honestly this is something a lot of organizations we work with want to accomplish. One of the things – I like to say we – Pulumi actually empowers operations teams to empower their developers, or infrastructure or SRE teams, you know, whichever sort of discipline your organization uses. The notion that, like, you want to empower your developers but with guard rails, so they don't actually do the wrong thing. So, we actually launched earlier this week something we call Pulumi Crosswalk for AWS, which is effectively a collection of AWS best practices that out of the box are just easy to use, but they do the right thing. So, a good example – and it really comes down to what you're saying around abstractions, right? A good example I like to cite is, you know, setting up a virtual private cloud, a VPC in Amazon, and probably configuring public/private subnets, making sure you don't accidentally expose something to the internet or different IP addresses than you intended, and locking it down.

Today most teams, they'll go read a 20-page document, a PDF on Amazon's website, and then try to manually accomplish what that whitepaper tells them to do, and so it's kind of like a blueprint. But I can't tell you – we work with lots and lots of organizations and everyone has to go recreate the wheel there, stand up a VPC, configure it, and then once you've got one and you want to spin up new environments or new applications, now you have to go replicate that. And it's a pretty rudimentary process to just manually copy and paste cloud formation files or what have you, and that's just really – it's not – you know, cloud formation is a great piece of tech. It's really just about, hey, the tools we're using are too primitive. YAML has no abstraction.

And, so, the idea with Pulumi is now you have these abstractions, and so you can have a – what is a well-configured VPC can now be encoded in code rather than text, and now you can share it as a package. And we actually have organizations that create their own to enforce their own best practices. And at that point now you can trust it's not even, just empowering developers, but it's also ensuring consistency across an organization. And, so, actually having a true component model with real abstractions is really sort of secret sauce in accomplishing that.

Mike Kavis:
Yeah, and all the work it takes to do that isn't really business-value related, right? It's something you have to do, but it takes away cycles from delivering what the users are asking for, so yeah, anything we can do to abstract that stuff. And the other thing I always bring up is, like, when we were in the datacenter, regardless if the datacenter processes were good or not, they were known, right? So, we'd been doing it for 30 years. Even with all the warts, we knew how to fix the warts. Then you go to the cloud and it's a blank sheet of paper. You have no network. S o this is the once-in-a-lifetime opportunity to get it right and automate it. So, how do you see your customers looking at that?

Joe Duffy:
Yeah, so I do see – you know, there's almost like a maturity cycle for most organizations that are going into the cloud. I think a lot of them start very simply, and then, for better or for worse, an organization has to make a few mistakes before they learn from them, and then decide that, hey, we need to get serious about security or infrastructure as code. You know, the first time your organization has a severe outage for multiple days because you lost an environment, but you didn't have infrastructure as code so that you could quickly spin up a new one, you learn from that mistake, right? And, so, I'd say most people that have been on the cloud journey for, you know, a couple years, which most people we talk to these days are, they understand the importance of automation and infrastructure as code.

And, so, what we do is we usually start with the very low-level infrastructure as code building blocks, right, because most people are coming from not having any abstraction. And we usually don't come in and start just layering on abstractions. We usually start by saying, "Hey, describe your current infrastructure in infrastructure as code, and then we can sort of map things." And then from there the nice thing about using real programming languages to encapsulate some of this is you've got refactoring tools. So, you can now start refactoring. Where are the places that we're repeating ourselves? Okay, take that, swap it out, use a reusable component. Oh, here's a place where we got it wrong. Okay, we can swap out – you know, use the VPC abstraction over there so that it's correct. So, it's more of an incremental approach. I think that's another key lesson, is many projects that really try to boil the ocean end up failing, and so it's really important to find those ways to incrementally add value and improve.

Mike Kavis:
Yeah, I like what you said there – learn from doing. (Inaudible) are nice, but nothing teaches you more than paint, right?

Joe Duffy:
Right, absolutely. [Laughter]

Mike Kavis:
So, the next question I've seen some of your talks in the past and I picked up on a theme you kept mentioning. It was how can we move from working at the assembly level to the architecture level?" So, explain what that means and explain how you help with that.

Joe Duffy:
Yeah, I think of, you know, YAML, which today if you look at a Kubernetes application, or you look at a cloud formation file, or even a lot of existing infrastructure as code tools, you're sort of describing at the very lowest level the bits and bytes of, like, every excruciating detail, every property, how these resources are figured, whether those are IM roles, or S3 buckets, or VMs, or containers, or clusters, or Kubernetes objects themselves. And a lot of that detail, you're just repeating over and over again. And it reminds me of, you know, back when we were programming in assembly language, and now all of a sudden you have C, right? And now you just stop thinking about certain things. Like you don't have to think about how to manage registers suddenly, right, when you have C. And then eventually you've got like Visual Basic, right, so you have an even higher level of abstraction.

And that to me is the journey that I think we're destined to go down with cloud programming. You know, the idea of abstraction is you can encapsulate and hide details that you don't want to think about time and time again. And this gets to the best practices conversation we were just having. If you have to repeat yourself over and over again, you're having to rediscover and restate the same thing. And if you make a mistake, now you have to go track down the thousands of places you made the same mistake, versus if you have an abstraction you can actually go to one place, one definition, and fix it. And, so, I think there are a lot of analogies between how programming languages grew up over the last 30 years that I think foreshadow a lot of what we're going to see in the infrastructure-as-code space.

Mike Kavis:
I remember working in the mainframe days, and the amount of work I had to do to get a screen page up and down was phenomenal, right? And then, all of a sudden, I drag this thing and go click, and I'm like, "Holy cow." But one of the challenges of abstraction – and I think abstractions is definitely the way we need to go – is that if you've only worked on the abstraction and you don't know what the plumbing is, you don't really know architecture that well. So, I'll give you an example. Again, I was managing a couple teams and we used to write drivers that worked down at the point-of-sale level. Now these guys are writing GUIs and they're asking their customers to buy faster laptops. And it's like, "Well, wait." And a lot of it's because they don't know how the computer works. They only know it at the abstraction levels. So, that's one of the things – I think abstractions is the only way we're going to get the agility. The danger is if all you ever know is abstraction, what is the result of some of the architectural decisions you make? Thoughts on that?

Joe Duffy:
I agree. One of the key elements when we designed Pulumi that was kind of unique – you know, I loved platforms like Heroku and these PaaSs and, like, you know, Google App Engine and Elastic Beanstalk in Amazon – you know, a lot of these higher-level PaaSs really are great for productivity. But the problem is, once you hit the wall, once the abstraction isn't working for you anymore for some particular reason, you kind of have to abandon the platform and go pick a different solution. So, what we wanted to do is, like, mix abstraction where you want it, but then where you don't want it, give you full access to the

underlying platform. And, so, that's why I was saying most people will start at the low level without the abstraction. They need to learn where are the patterns, and then they appreciate the abstraction a bit more.

I probably take these analogies too far, just because of my background, but you know, it's funny. When we were starting Azure at Microsoft, it was called Red Dog at the time. Dave Cutler, who's actually the architect of Windows NT, he was one of the primary architects of Red Dog back in the day, and they would describe it as an operating system, a cloud operating system. And what eventually became Azure was a cloud operating system. And when you start thinking of it that way, you really think differently, because I think of Windows as a good analogy, right? Like you program Win 32, there's all these APIs to do certain things, and then eventually, you know, the .Net framework came along and it helped make that platform more and more accessible. And I look at Amazon where you've got all of these APIs. They're highly programmable. And the real trick there is you just need to package it up and give it to developers in a form – and infrastructure engineers in a form that they can consume and build bigger things out of smaller things.

Mike Kavis:
That is a good analogy. And I think the other thing – a lot of times people's thinking is a little too binary, right? You don't always have to use Amazon's stuff. Use the right tool for the toolbox. So, sometimes you use the PaaS. Sometimes you're at IS. Sometimes maybe you are writing a little assembly module. I hope not, but use the best tool for the job, right?

Joe Duffy:
Absolutely, yep.

Mike Kavis:
All right, the next topic – Kubernetes, like the hottest thing on the planet here. And I remember I didn't attend but I was following a live Twitter stream at – I think it was at KubeCon or everything, and there were all kinds of presentations on all these heroic efforts people were doing and, you know, all this stuff. There's so much involved, making that work. And my Twitter response is, "That's great. Give me an API." Right? So, all this stuff these people are doing to get their cluster to work, and I'm like, "But, why are we doing that?" So, a lot of solutions are out there trying to kind of abstract that abstraction. That's an abstraction, but we're trying to abstract that because it's complex. So, what are you guys doing in that space? And what's your recommendation? Someone a little late to the game, wants to get into container and Kubernetes, what would you recommend? Now that we've spent two, three years learning it, what would be your recommendation to them?

Joe Duffy:
Yeah, it's a great question. We work with a lot of customers. We're in a unique position because Pulumi is multi-cloud, right? So, we support AWS, Azure, Google Cloud, you know, on-prem technologies as well, and so we actually work with a lot of customers to help them deploy container workloads, including Kubernetes workloads. And the thing I would say is, you know, with Kubernetes, really, it's a commitment to becoming an expert in managing Kubernetes clusters. And you need to understand that's the game you're getting into. There's no easy, "Hey, I'm just going to do Kubernetes, and I'm just going to throw a container into Kubernetes, and I'm never going to have to worry about Kubernetes stuff." Like, you really do need to have a team who's going to understand how to operate it. So, a lot of customers actually we work with, if they're going to Amazon will choose ECS Fargate, which is the native container technology in Amazon, just because it's simple. But a lot of people do elect to go with Kubernetes, whether it's they want to – they're betting on Kubernetes to help with their hybrid cloud transition. Maybe they want to remain agile between different cloud providers. There's a lot of reasons why you might go with Kubernetes.

I think the key thing is, it's one thing to just get a cluster up and running and start playing. It's a very different thing to go to production with Kubernetes. You know, a lot of people don't realize that until they get pretty far into whatever proof of concept they're doing, and it's important to realize that upfront. You really do need to think about networking, security, how it relates to however you're managing security in the cloud already, like, you probably don't want to have a mirror RBAC system. You probably want to integrate with your cloud's IM out of the box. And, so, one thing we've done, you know, this Crosswalk that we launched basically supports hosted Kubernetes clusters that are configured securely out of the box, so it's a lot easier to do because today it's still a lot of point and clicking, a lot of documents you have to go read. It's similar to the VPC example I talked about earlier. If you want to learn how to secure your Kubernetes cluster, you have to basically go read the internet and, like on Twitter, on Hacker News, on whatever cloud provider's documentation, on the Kubernetes website itself. We're still very early days, and so it's still difficult to find out what are the best practices, how do I adopt those best practices, and so that's one thing that we help our customers understand when they're going to production.

Mike Kavis:
It is very early days I remember when - I mean, there's no ifs, ands, or buts about it. This stuff's hard, and how can we make this as seamless as possible, so we can work on business requirements and spend less time managing infrastructure? Because what I see too often at clients – you know, they're just trying so hard to be agnostic, that they're actually spending as much time with clusters as they were standing up physical infrastructure in their datacenters.

Joe Duffy:
Yeah, I agree 100 percent. It's important to realize that there's new technology, and that's pushing the boundaries of kind of what you can do. That doesn't always mean that you have to, right?

Mike Kavis:
Yeah.

Joe Duffy:
And I think there's a difference between some of the innovation in things like serverless technologies, and what people really should be doing today. And I think they're – you know, Kubernetes is a good example where – Kubernetes I think in three years we're going to look back and say, "Okay, we've got a stable platform. We can depend on it. It's great for hybrid workloads" – but that doesn't mean you have to go adopt Kubernetes today, right? There are simpler ways of doing things. I think also, you know, I love the Docker story, because that was one of the things that hooked me, honestly. You know, Docker on your desktop was amazing, and then – but the problem was you wanted to go to production. Now you're managing a swarm cluster or a Kubernetes cluster. You're thinking about (Inaudible), like consensus protocols, and you know, as a developer, you know, I was coming – remember, I was coming from a developer standpoint. I was kind of like, "This is not what I signed up for, right?"

Mike Kavis:
Right. [Laughter]

Joe Duffy:
I thought I was going to be able to deploy a container to production, and now I'm knee-deep in networking stuff. And, so, one of the things that we've done actually is package up some very basic patterns of how to go and just take a container-based app and get it up and running in the Amazon or Azure – wherever you're going to. And this gets back to, you know, my discussion earlier about some of the AWS. If you're going to AWS, you know, ECS is going to be way easier. If you all you want to do is take a container application and get it up and running, it's a lot easier. You don't have to become an expert in managing clusters. And, so, I think, you know, people should question complexity a lot more than they do. I think we've all become too tolerant of complex technologies.

Mike Kavis:
Yeah, my colleague Dave Linthicum who also blogs on this channel, right now his big paper that he's working on and the thing he's focused on is complexity killing the cloud, and that's where we're at. There are so many choices, and a lot of these things are new, and you could quickly architect yourself into some real spaghetti real fast.

Joe Duffy:
Yeah, yeah. The good news is I think, you know, we're sort of in the – you know, think of the big bang, right? We're still in the expansion phase. At some point we will start contracting and get back to something that's simple and that's stable and that we can depend on. I mean, frankly we've already made progress, right? If you look back two years ago, three years ago the question was, like, hey, should I bet on mesas? Should I bet on swarms? Should I bet on Kubernetes?"

Mike Kavis:
Right, right.

Joe Duffy:
At least now we've all rallied around one platform, and I think that's moving in the right direction. It takes time, though.

Mike Kavis:
Yeah, hopefully I didn't make this all sound like doom and gloom. It's just – you know, we've just got new problems, right? There's always challenges. We're getting stuff out the door faster. We just – we're creating new challenge as we're learning. It's a pretty exciting time to be alive in this space. So, Joe, appreciate your time today. Where can we find you on Twitter and where can we read some of your stuff and see some of your videos?

Joe Duffy:
Yeah, so definitely follow the Pulumi Corp Twitter account, and my personal handle is @FunkOfJoe, which is a little programming language pun there. And I've got a personal blog, JoeDuffyBlog.com. I've been remiss in keeping it up to date but I'm going to plan to do some fun stuff there pretty way. And then the official Pulumi blog is always fun to follow, too, Blog.Pulumi.com. I should mention Pulumi is open source, so it's free to download and use for infrastructure as code, so definitely give it a try. We've got a community Slack. We've got thousands of people there, practitioners doing cool stuff with Kubernetes and Amazon and on prem stuff, and the community is really growing and – to the point where people in the community where people are helping out each other, and so that's really awesome to see. So, definitely give that a try and let us know what you think.

Mike Kavis:
Cool, I'll do that. So, that's it – today's episode, Architecting the Cloud with Joe Duffy. You can find podcasts like this by me and my colleague Dave Linthicum just by searching for Deloitte On Cloud Podcast on iTunes or wherever you get your podcasts. I'm your host Mike Kavis. You can find me on Twitter, @MadGreek65. And if you need to get ahold of me directly it's MKavis@Deloitte.com. Thanks for listening and we'll catch you next time on Architecting the Cloud.

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