Deloitte.

For Cloud Professionals
May 2021

For Cloud Professionals, part of the On Cloud Podcast

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

Title: **Mitigating the multicloud complexity risk**

Description: Multicloud has become a reality for most organizations, but there's a tremendous amount of complexity and risk inherent in a multicloud environment. How companies deal with the complexity and risk can largely determine the success of their cloud journey. In this episode, David Linthicum and guest, Prudential Financial's Global Head of Cloud Services, Drew Tatarian, discuss keys to reduce cloud complexity and mitigate risk. Drew's advice: pick a core vendor and augment with complementary vendors, govern and nurture your cloud ecosystem effectively, automate what you can, and have strong guiding principles that help you make decisions that maximize business value.

Duration: **00:22:34**

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. Welcome to On Cloud, the podcast for cloud professionals, where we break down the state of cloud computing today and how you can unleash the power of cloud for your enterprise. Now here is your host David Linthicum.

David Linthicum:

Welcome back to the On Cloud Podcast, your one place how to find out how to make cloud computing work for your enterprise. This is an objective discussion with industry thought leaders who provide their own unique perspective around the pragmatic use of cloud-based technology. Today on the show we are joined by Drew Tatarian, Global Head of Cloud Services at Prudential Financial. Drew, welcome to the show.

Drew Tatarian:
Thanks, David, great to be here. So, tell me what do you do at Prudential? What's the say in the life of Drew like?

Drew Tatarian:
Sure. I have responsibility for the cloud, as you mentioned, and what I really enjoy is it's everything from strategy about what we should be doing in the cloud through engineering, deployments, and operations, so the full life cycle. And we've moved to an Agile way of working where I'm basically the product owner; I'm responsible end-to-end for that delivery and management of the team and the insurance that we create business outcomes for our customers.

David Linthicum:
So, what would be your priority these days? What do you typically—like, considering your schedule, what are you scheduling first as number one priority, second priority, third priority, those sorts of things?

Drew Tatarian:
Sure. So, we're almost in year three, coming up on year three of our journey, so we're beyond the basic building blocks. I will say that still number one in the list is always security, so it's never something you walk away from in the cloud. The constant innovation of what's happening and what the cloud vendors are delivering requires you to be on top of changes and how that changes the view you need to have in security. But I would say more near term, we're really working on self-service enablement, which is trying to get our developers and users that are trained and capable to enjoy the privileges of using the cloud firsthand, to provide them more services and really build those kind of guardrails and controls in the background to enable them to be productive, but still stay compliant with where we want to be from an enterprise perspective.

David Linthicum:
So, you've had to make some pretty challenging technology decisions moving forward, including when to leverage agnostic versus native services during the cloud deployments, things like that. So, what have you found to be the tradeoffs between those two different kinds of approaches?

Drew Tatarian:
If you read all the literature, everybody's got a different view here, and some say you should be agnostic because it allows you to move from cloud to cloud, and others say you need to stay with one provider, or whatever providers you're using, and really go deep and leverage the benefits that they bring and the integration of those services. And I kind of fall into that latter camp, and we've been trying to promote that philosophy as one that we think is the right way to head, which is really go deep, use whatever cloud provider you're using, use all the services in their native fashion, because the pace of innovation, we're not even at the knee of the curve with respect to cloud providers and what they could do. The changes that they make and not being able to leverage them, or the dependency some services may have on each other, requires you to be in that native landscape.

And on the flipside, that agnostic view, in some ways, I think is an illusion of portability, right? You believe you can move, but it's almost impossible to engineer an application that will never be able to use something native and will easily be portable, right? So, it's tough. Even with things like Docker or other insulation layers, you're bound to leveraging some underlying service that makes portability challenging.

David Linthicum:
Yeah, I think that's great advice, I mean, the least common denominator approach to building applications in the cloud. And, basically, you're not going to perform well, or be 100 percent, or optimized anywhere. And, so, you're making the tradeoff of becoming mediocre just to provide portability that you typically may never even use moving forward. So, my advice to people who are moving to multicloud is basically the same as yours. Look at the native services. Take advantage of what the cloud features have to offer. Get deep into the particular technology or platform that you're leveraging. So, what advice would you give to those moving in this direction?

Drew Tatarian:
Yeah, I mean, I think like you said, embrace the native services. Understand that comes with a steeper learning curve and more of a burden on application refactoring, because you are going to try and take applications that might be running in your datacenter, or that might be more traditional in how they were architected, and moving them to a way of working that people are not very comfortable with. But I like to look at it as accept that lock-in is really the cost of exit, right? There's no technology decision you could make that is free from some long-term implications, right? So, the thing that I think that has worked from an advice perspective is we look at these decisions, and if there is something that makes sense to be shared by multiple clouds or by on-prem in the cloud, you might give it some thought. But you have to really weigh that risk of using that versus maybe where you'll be long term and what that might do to your strategy.

But I will say that I think in the cloud in general, the applications that are designed, if they are native, are already decomposed in some way. They're no longer monolithic, which means moving them to a different vendor or to a different place can be done in piece-parts. And since all vendors, remarkably, have very similar services—you want to know how that happens—and in many cases those services are compatible, you actually do have some degree of portability even across clouds, if you do want to move. I mean, sure, there's energy and cost to do it, but you'll find similar services in different places. And I think people should just realize they're picking a partner, or multiple partners, and they should embrace that native service and go along for the ride, because it's not slowing down anytime soon.

David Linthicum:
Yeah, I think one of the considerations out there is just a good design. I mean, there's ways in which you can build applications and data stores where you put the native volatility into a single domain. And, in essence, if you have to move from platform to platform, it's a much easier change to make, and you'll still get the access to the core native services, which is what you're looking for. So, design and architecture is very important as well.

So, five years ago, not a lot of multicloud going on. People were typically deploying on monolithic cloud deployments: AWS, Microsoft, or Google, one or the other. And now people are using two or all three, 95 percent according to a recent survey that I just saw. So, what are some of the issues related to deploying on multicloud versus single cloud that you've found?

Drew Tatarian:
I'm actually starting to become skilled in more than one, and what I learned is, even just from the basics, they all have different nomenclatures, right? So, what one cloud provider calls roles in security another one calls policies. So, there's an increased burden there on your training and skillsets, and that translates into additional cost to try and be good enough at all of these to be enterprise grade. I think the decision on why you'd use a particular cloud vendor has become something that's in the forefront of my mind. I'm not trying to provide the same services with multiple cloud vendors. To me that's chasing an outcome that I don't think delivers business value.

What I think delivers business value is picking a core provider that delivers the majority of the services that you think you need, and then supplementing those services with other providers that may have a distinct view on a particular domain of technology, or play very nicely to augment one of your existing cloud vendor's decisions. But trying to create multiple clouds to deliver the same services to me is probably not the best use of resources in the cloud world.

David Linthicum:
Yeah, ultimately, I think you have to pick and choose different services, and you have to look at the way in which they're coupled, and I think if you're going to go to heterogenous distribution of services and building cross-cloud applications, it has to has an abundance of caution that has to occur.

So, moving forward, as people are moving into multicloud, they're trying to make these core decisions around common services, common security systems, common governance systems, common management and monitoring systems. And they have, in essence, a richness of choices right now, which is how one CIO put it I talked to a few weeks ago. So, are there too many choices in dealing with multicloud, and is this leading us to very complex systems because people are picking different services, or are we getting away from common services? And how are we mitigating this risk right now?

Drew Tatarian:
Yeah. I mean, I'm not really sure we're seeing that in my day to day, but I do see that there's lots of choices. And with every technologist out there, there's lots of information for people to consume, and people then gravitate towards technologies that they think are going to be the silver bullet technology to help them with their solution. I do think we're seeing a normalization around things like Docker and Kubernetes and people believing that there's some, like I said before, illusion of portability with that. Although there is some developer normalization that happens if you pick some underlying common capabilities that make your CI/CD pipelines a little bit more straightforward and maybe some of the governance.

I think things like centralized logging solutions and be ing able to ensure that you get all of your relevant security information in a place where you can govern it and look at it with tools, and trying to maintain compliance with industry security policies and guidelines that would be generic and cross-cloud, are ways that we've looked at it. Not so much focusing on a single tool to do it, but trying to ensure that each of the cloud providers are held to the same standard with respect to what we're trying to deliver from an enterprise service.

David Linthicum:
You said something interesting in terms of dealing with containers, and really there is a—I wrote an article a couple of years ago talking about a container tax. There's more effort in designing, deploying, and building these various systems if you're going to get to a level of portability, which is what the promise is. And ultimately there may be a performance hit. There may be different security software standards. You're going to end up paying more and encountering a bit more risk in leveraging containers, versus just kind of building in a native fashion. So, do people understand that out there? Is it still kind of very pro-containers and it's very difficult to get people to kind of understand the tradeoffs of leveraging that technology?

Drew Tatarian:
Yeah, I think that there's maybe a view that containers is the end state, and I try and tell people containers are just the next thing on the menu, on the road of many other developments that will happen in the future. But to your point more specifically, things like how do you manage secrets in an application, and how do you manage parameters, and how do you configure queueing systems that need to—the container needs to talk to. All of these surrounding things that are the ecosystem of what a container uses are not part of the container, right?

So, by nature of—unless you build every one of those in a container and then put that also into your environment, now you've basically gone to the extreme opposite of native and agnostic and you're basically managing a datacenter in the cloud, which doesn't really give you the advantages. The container only gives you some amount of portability, and then the rest of the ecosystem is really challenging to move. But it does deliver the value of the developers knowing how to build and create a single unit of work. But that single unit of work doesn't live in isolation and that's where the portability challenge lies.

David Linthicum:
So, a few years ago when we were migrating things to the cloud, it was really kind of migration at scale, and I think we even called it that. And people were opting for more lift-and-shift capabilities, where they were, in essence, moving it from one platform to a platform analog that exists on a public cloud and doing a minimal amount of code changes if any code changes, coupling it to a database, typically the same database—it could be a cloud version of MySQL, something like that—and calling it good.

They found out those applications didn't run well. They weren't performing well. And the reality is they had to go back and do a full refactoring, which means rewriting significant portions of the applications and changing the database, or a partial refactoring where we're just kind of a tuning the thing up to leverage some cloud-native features, but not all cloud-native features. So, how do we understand the tradeoffs of this? How does lift-and-shift versus partial or fully refactoring of applications come in and how do we make these decisions now in 2021?

Drew Tatarian:
Yeah, I know. It's always been a challenge, right? You're trying to balance, in some cases, speed versus efficiency, right? So, I think there's a time and a place for lift-and-shift. I really do. If you need to leave a datacenter, if you have a need to transform to a single operating model, as opposed to keeping an on-prem operating model and a cloud operating model, because the skillsets you have and the journey you want from a human resources transformation

would be faster. I think that there's some value in lift-and-shift, and if done properly, you can try and minimize some of the downsides to make it maybe cost neutral, or maybe a slightly higher cost than the on-prem depending on your size, scale, and complexity.

But I think the important thing to realize is it's not an end state, right? It's just a step along the way. So, I think when you—whatever strategy you have with respect to migration to the cloud, lift-and-shift could be a good first step. It could be a good first step for applications that are destined to be retired or rebuilt or replaced at some point in the future, but you don't want to modify them for the cloud if they're going to otherwise dispositioned in another manner. But the real value comes, I think as you said, from the elasticity and the real benefits of using the cloud-native capabilities that allow you to flex the cost and compute to meet the demands of the business.

David Linthicum:
So, in other words, what we're trying to avoid is migrating twice, migrating once via lift-and-shift and find that was not the right approach for this particular application, and having to go back and make changes to the application, in essence making the migration again.

Drew Tatarian:
Yeah, I mean, I think that's fair. But I think if you had a datacenter that was reaching end of life, and you had to make investments there versus lifting and shifting the apps and then doing something again, I think that's where it becomes more of a business decision at that point than a technology decision. So, how do you achieve the business objectives, which I think I've kind of threaded some comments throughout this conversation, is to me I'm very focused on business outcomes. That's all I focus on. And whatever the technology solution that drives the business outcome, I'm not a purist when it comes to having to have a particular solution as much as making sure that whatever we build delivers the value. So, to me, if we needed to close a datacenter and do it quickly and I had to employ a lift-and-shift strategy, then that would be sensible if the economics were positive.

David Linthicum:
So, moving forward we're looking at people who are migrating to the cloud, and certainly the pandemic has provided some challenges, because, actually, I think people were doing so in a very orderly, probably slower fashion, and then suddenly the pandemic hit. People were rushing to public-cloud-based resources because they viewed them as lower risk and more reliable than some of the stuff they had on-premise. There were people who couldn't get to their on-premise systems because they were quarantined and stuff like that. But the result was suddenly everybody started to accelerate the migration to cloud and migrate in a rapid pace, and not necessarily planning, or thinking in terms of how these things were going to be done. And I suspect a lot of the existing migrations are going to have to loop back and fix some of the mistakes we made either moving too fast, picked the wrong technology, did lift-and-shift versus cloud-native when we really should've done cloud-native, and those sorts of things. So, what advice would you give to others that are attempting the journey into the cloud?

Drew Tatarian:
Yeah, sure. I mean, look, in three years there's obviously things I think we've done really well and things that I think everybody would like to look back and say, "Man, I could've done that one or two things differently." I would say the thing that's kept us on our course was having really strong guiding principles for what we wanted our cloud solution set to look like, so things like be cloud-native and don't deliver overlay solutions in the cloud, and design your applications with operations in mind—establish projects using an MVP outcome approach. These were some of the principles that we employed at the beginning that kept us from the Amazon term, “undifferentiated heavy lifting,” that AWS likes to use, right? We didn't want to be in that space, right? And I think whenever we were at an inflection point, or we were unsure what to do, the guiding principles really helped us refocus on the journey that we were on and helped us make some of those decisions.

Another thing that I think we learned along the way was ruthless automation, right, everything as code. Trying to make a change in the console should be the thing you never want to do. You want to, if you see something wrong, find a way to write infrastructure as code to fix that problem so that it's persistently fixed, not fixed on time, and trying to find those high-frequency, low-complexity items that you can keep automating away so your team can focus more on the high-value outcomes.

And then one thing that I'm calling the N-minus-two problem, which is if you walk through a datacenter, anybody who's done that anytime recently, you see the history of every technology that's ever been bought, and applications that were built for that technology, and they can't get rid of it because it's locked into that particular footprint.

And when you look at the cloud, you have that same risk and, in some cases, pitfalls that you fall into where you start out on a journey, and you make a design for your network, or your identity and access management, or your virtual private cloud environment, and then you mature and/or the cloud provider matures in some way, and now you say, "Hey, let's move to this new version of how we do it.” And that becomes the landing area for all the new applications. And then six months later maybe you do it yet again, and now you have this problem where applications that were built at some point in the future are very, very similar to that datacenter that I mentioned at the beginning there, where they're just sitting there on an older way of doing things. And how do we really promote this constant updating of applications to bring them no older than two iterations back from how we're doing it today, and really trying to use that infrastructure as code methodology to make that happen. It's a real tough problem.

David Linthicum:
In other words, make the applications autonomous unto themselves so they can define the platform they're going to run on, define the security that they need, define the governance that they need? Am I getting close?

Drew Tatarian:
Yeah. I mean, I think that would be the dream world. But if I had them when they came to me at some point in the past, I would've said, "Here's how I'd like you to build containers," and then they went and they built their containers that way. And then I'm like, "Well, look at this. Amazon or AWS released this next thing. Let's do containers this way," or, "Azure has this feature that we want to leverage, and now all I want all new applications to start building that way." How do I get those first set of applications to update their deployment to the new model, so I'm not supporting multiple different ways of doing anything and introducing an operational complexity that we really try and avoid in the cloud.

David Linthicum:

Yeah, and complexity is getting to be a problem right now. And I agree. I mean, people over the years, we have a methodology of managing by magazine. Of course, we don't have magazines anymore, so we manage by blogposts, where we're, in essence, moving toward where the crowds are moving, or the tech press is moving in promoting a particular technology. And, so, we'll build it in containers a couple of years ago, and then containers that are in Kubernetes clusters now, and the ability to leverage federated Kubernetes in a year or two down the line, the ability to operate using traditional procedural operations systems; now it's AIOps systems moving forward, and we're leaving these legacies behind and we still have to maintain and operate these things. They typically never go away, to the point you make. So, in essence, we're kind of painting ourselves into a corner. Is that a good way to put it, Drew?

Drew Tatarian:
Yeah, and I guess my point there is the cloud is not solving that problem. We have to actively solve it, right? So, it just provides us another playground for people to make those same choices, and we really need to get ruthless about that, in addition to the automation around bringing forward some of these prior implementations. And given they're written as infrastructure as code and pipelines that do the deployments, the challenges shouldn't be as high as migrating from one legacy database to another in the historical infrastructure world, but it needs to be cared for, right? If you don't care for it, we're going to end up, even within a single cloud provider, with generations of different ways of doing things which are going to add to complexity and cost.

David Linthicum:
Absolutely. That's why planning and architecture is so important, and also people who do things using common sense approaches, which it appears you do. So, if you enjoyed this podcast make sure to like and subscribe on iTunes or wherever you get your podcasts. Also don't forget to rate us. Also check out our past episodes including the On Cloud Podcast hosted by Mike Kavis and his show Architecting the Cloud. If you'd like to learn more about Deloitte's cloud capabilities, check out DeloitteCloudPodcast.com, all one word. And if you'd like to contact me directly you can reach me at DLinthicum@Deloitte.com, L-I-N-T-H-I-C-U-M. So, until next time, best of luck with your cloud projects. We'll talk again real soon. You guys take good care.

Operator:
Thank you for listening to On Cloud for Cloud Professionals with David Linthicum. Connect with David on Twitter and LinkedIn and visit the Deloitte On Cloud blog at www.deloitte.com/us/deloitte-on-cloud-blog. Be sure to rate and review the show on your favorite podcast app.

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