Deloitte.

On Cloud Podcast
September 2022


# The Deloitte On Cloud Podcast

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

**Title:** **Cracking the multi-cloud security puzzle by shifting security left**

**Description**: Multi-cloud security is one of the most difficult pieces of the cloud complexity puzzle to solve for. It's also one of the most critical. In this podcast, David Linthicum talks with Deloitte's Ramesh Menon about how shifting security left—i.e., implementing appropriate security during application development to solve issues before systems go live—can help companies move from a reactive to a more proactive security posture and increase the effectiveness of their security program.

**Duration:** **00:25:40**

**David Linthicum:**
Welcome back to the On Cloud podcast. Today on the show I'm joined by Ramesh Menon. He's a managing director at Deloitte and leader of the Cloud Security Policy Orchestration offering, a solution focused on shifting security left. Ramesh, welcome to the show.

**Ramesh Menon:**
Thank you, David. Good to be here.

**David Linthicum:**
So, let's get into yourself. So, how did you come to Deloitte? What's your background? How did you get into security? What was the siren song to get you into cloud security?

**Ramesh Menon:**
Good question. I started a company along with a couple of my colleagues in 2017, called CloudQuest, focused on taking a cloud native approach to multi-cloud security, and Deloitte acquired the company last year, and, so, that's how I'm at Deloitte. And actually, that's kind of like the core of the Cloud Security Policy Orchestration solution. I got into security about the year 2000. I started my career at Silicon Graphics working on parallel computing. And then after that, into a startup called Narus, where we built some of the first carrier compliance software that helped telcos to comply with some of the requirements of carriers and things of that nature. Several startups, and then along the way in large companies building infrastructure software, moving to the cloud, and helping my teams move from fear of the cloud to cloud first, cloud native software development. That's kind of my background.

**David Linthicum:**
So, what's your current role? What are you doing, what's your day job now?

**Ramesh Menon:**
Yeah, right now I lead the Cloud Security Policy Orchestration solution. One of the things that I think a lot of CISOs, a lot of security folks are really aware of is that tools produce outputs. Solutions provide outcomes. And, so, really trying to focus on how to enable security outcomes for our customers, that is entirely the focus of my effort here. And how do you actually get from moving security from more of a reactive endeavor to a proactive endeavor, and how do we enable our customers to be able to get there?

So, a lot of times there's a lot of material out there on how to do those things, and, so, we try to come at it with an opinionated way of how to enable that, how to make that operational, because security teams are still fairly small compared to application teams, business teams that are actually deploying workloads into the cloud. So, that's kind of the entirety of my focus. I like to be technical and, so, I spend a lot of my time on the product, the engineering efforts, and how to build the product itself.

**David Linthicum:**
So, I just did a press call, and they asked me the question about multi-cloud security. We're talking these days about the super cloud and the metacloud, and the fact of the matter is we're building security now in-between these various cloud providers and not within a particular provider, even if it runs physically within a provider, it may logically span providers. So, multi-cloud security is becoming this very complex thing that's defined differently depending on who you talk to. So, what is multi-cloud security in your definition, and how is, ultimately, the way that we're approaching these technology challenges changing right now?

**Ramesh Menon:**
Yeah, it's a great question, and in fact, you referred to supercloud, and the fact that a lot of these workloads because of things like latency and the amount of data that needs to be backhauled in order to bring it to the cloud, a lot of it's happening on the edge, hybrid, there's edge hybrid, data center, cloud, multi-cloud. And, so, there's definitely a recognition that security is becoming very complex and managing security in a single cloud is complicated enough, and moving to the multi-cloud, it is actually is a big challenge in terms of the way that we've been doing it.

What happens is that a lot of times, the way that we approach it is that we have these workloads deployed into the cloud, and we have had great tools available to find out what are the insecurities that are running in these cloud configurations. And now you multiply that by all the clouds that you're running in, and you extend that to hybrid workloads and to the edge, you're drowning in what we usually call alert fatigue, which is, typically folks think of an insecure configuration as an alert. It actually generates an alert, and that which means that hey, we need to do something about it.

That, typically, at cloud scale, is impossible to deal with in a manner and fashion, and, so, we turn on integrations to ticketing systems, and it actually shows up as a ticket for some person to actually do something about. Then we also have automated remediation for the subset of workloads where it is safe to turn on auto remediation, things of that nature. So, multi-cloud security has become very complex from an operational perspective, where we have a lot of the tools to find out what is happening in the cloud in terms of security, but getting ahead of that has been a big challenge, and there will be a lot of developments now in terms of how do we actually—everyone is asking.

We are drowning in alerts. We don't own auto remediation. We cannot do that for—we don't have the confidence to turn on automated remediation for a large number of these cases. What can we do? How can we do better? That's where the conversation is now leading to, and that is entirely the focus of some of the things that I was talking about. How do we move away from this reactive model where things that are insecure have already ended up in one of our workloads? What do we do? What can we do better? So, that's the whole movement to shift left security, and that's one of the areas that I'm very passionate about, and I think that's one of the solutions to this whole issue of drowning in alerts, alert fatigue, and managing this complexity across the different clouds, there's probably an easier way to address it, there's a better way to address it, and that's actually what I'm focusing on.

**David Linthicum:**
Yeah, it seems like we're dealing with operational aspects of multi-cloud, and security aspects and governance, data integration, things like that, and getting through that complexity. In fact, the industry calls it the complexity wall. In other words—and you just kind of highlighted it. We're in alert fatigue because we're getting all these various alerting systems that are coming from all these various heterogeneous systems that sit within our operational framework and things that we have to secure, and there's really no way that we can have the same humans we have around today that are going to scale up and support that.

And, so, we have a couple of choices. Number one, we can hire more people. Or we can use abstraction, we can use augmentation, we can use automation to start doing these things better. Am I off?

**Ramesh Menon:**

No, you're absolutely right, and I think you hit the nail on its head, right. There are not enough security people to go around, even if we were able to throw people at the problem. Second is that it's not only cloud scale but it's cloud speed. We now have the ability to—a lot of these workloads are not being provisioned or composed in the cloud on a cloud console. They're actually using automation to provision these workloads. So, the fact that we can provision these workloads easily means that we can deprovision them easily as well. And, so, it is practically impossible to do these things manually.

Second, I think of it as a data flow problem. So, by the time the differences are in the different clouds, and when we talk about the super cloud in terms of whether it's going to the edge, or whether it's going to a data center, or it's actually going to one of these on-prem infrastructures such as Kubernetes, things of that nature. But if we think about where is the place where we can intercept it where things are most homogenized, then we start getting the ability to address these things earlier. So, that is where this whole of security effort is going to.

And I can describe what that means because we've done application development for very long. And, so, some of the lessons that we have learned from application development can be very useful in this context because that is the point at which things are much more uniform. Now, of course, everybody's familiar with data source control systems, so the way we develop applications today, a developer checks in code, it triggers, it goes into their version-control system, VCS, a source control system, and then it kicks off a whole bunch of actions, typically, which can trigger continuous integration, continuous deployment, CI/CD pipelines, and we can actually do a lot.

One of those things that we do, we test the code. When we deploy an application, we don't just go and deploy that directly. We have it run through automated tests, we run it through unit tests, functional tests, performance tests, and depending on which part of the pipeline you're on, you may actually run SAST and DAST, static analysis, security tests, dynamic analysis tests, software composition analysis, all those things. And then, once it passes all those things, only then it advances in the software development pipeline.

So, we move it through the process only if it passes those tests. And, so, there is a single point of interception there, and the question that we can ask ourselves is that, to manage all this complexity, can we do these things earlier in the software development life cycle when things are much more homogenized. Everybody's using the same software version-control system, source control system, and are there things that we can do earlier in the life cycle where things are much more manageable where we don't have this explosion in terms of complexity. So, that's where this entire software development life cycle, shifting security left is happening.

And, so, maybe it's obvious when we sketch the software development life cycle, the developer's on the left side of the picture and the deployment happens on the right side of the picture, and as we get more and more to the left, we have better ability to control what's happening there. And that is one of the ways in which we can actually manage security because what happens is that we can actually catch a lot of these things. Just like for applications we catch a lot of these things earlier. The question is can we catch a lot of these security issues earlier in the pipeline. So, that's where this entire effort is going

**David Linthicum:**
So, what does it mean to be reactive in dealing with multi-cloud security, any security approach or model?

**Ramesh Menon:**
Yeah, so reactive is the set of tools that we use today to, once something has been deployed already into the cloud, we have had this cloud security posture management, CSPM tools, cloud workload protection platform, CWPP, and then the cloud infrastructure entitlement management, the CIEM, platform which actually focuses on the identity and access management portions of this.

We've had these tools which actually look at the deployed configuration in the cloud, and when they detect something that is insecure, and we have all had a lot of good standards for looking at deployed configurations. The Center for Internet Security has published, for example, guidelines. We call them guardrails for what constitutes a secure deployment. So, we can actually check whether it meets some of those requirements.

An example would be, I deploy a bucket into one of the public clouds, and let's say I forgot to set encryption on. So, that particular resource would have an attribute called "bucket.encryption equals on." Let's say I forgot to set that, so these tools would actually go look at the deployed configuration and then they raise what we call an alert. And then that alert actually goes typically to either by e-mail or, depending on what integrations are turned on, it would create a ticket, and hopefully the developer would get the ticket, or it actually flows through some automated remediation system to go fix that particular configuration issue.

This is a reactive way of doing it because what happens is that the insecure configuration is already running in the cloud, so the time between when we find that there is an insecure configuration and when we're able to do something about it, which we call the remediation latency, is ripe for exploitation. A bad guy can come in there and say—and there are these scanning tools out there that can look for open buckets, open firewalls, open compute resources, and they do—they can and they do exploit some of those things.

We are actually catching these issues after they have manifested themselves in the cloud, and that's why we call it reactive. So, the idea of this whole shift in security left us asking, "Are there things we can do to prevent these misconfigurations from getting deployed in the first place?" And, so, obviously, this is what we call a proactive approach. That's the shift left security approach.

**David Linthicum:**
So, what can we do better? How do we set this up? If we're kind of in a non- reactive state today, or we're not doing security in an optimal way, what do we need to do better? You have to secure their clouds, what do they need to work on? How do they stop digging themselves into a hole?

**Ramesh Menon:**
Yeah, so it's useful to think about what application development has given us. I talked about the whole model of applications go through the CI/CD pipeline. What is it that allowed us to do that? It is the fact that our application core, the configuration, all of the associated artifacts live in a source-control repository. And when they live in a source-control repository in a version-control system, what does that mean? They exist as code, right.

So, what has happened is that to be able to move security left, we need two things. One is that the infrastructure that is being deployed also need to exist as code, and the security policies that we need to apply to the infrastructure as well as the applications need to exist as code. Sometimes we bandy about this phrase called “everything as code,” and the value proposition of everything as code is that we can use all of the tools that we have developed over decades for software development, application development, and apply that in a similar fashion to how we test applications. We can apply that to security as well, i.e., what we can do is when an application developer checks in a change to the code, it runs through the pipeline, and if some test fails, it doesn't advance in the pipeline and doesn't get deployed.

Similarly, when we create some new infrastructure or when we change some entitlement, we check that in as code, and if we are able to set up our pipeline in such a way that we can actually check that new configuration against all of the security policies, let's say we were able to do that, then that particular infrastructure change or entitlement change would fail in the CI/CD pipeline and doesn't get deployed into the cloud. So, if we were able to do that, then we would accomplish all this idea of proactive security, which is what we refer to as the shift left security model.

So, in that model, the insecure configuration doesn't get deployed to the cloud in the first place. It fails in the CI/CD pipeline, and there's no possibility of exploiting that. So, that's what that enables us. So, what has been missing so far has been we have now had infrastructure as code, IAAC, in the form of Terraform and cloud formation templates and Azure resource manager and things of that nature. That has allowed us to encode infrastructure as code, and that means that we can actually now put infrastructure in our source-control systems, which means that we can actually run all of these things through the CI/CD pipeline.

What is missing so far is security. A lot of times we have a document, we go take that document, and then we go assess it manually, or within the case of the cloud, we use the CSPM tools to assess the security posture of our running cloud configuration. But all of those things, depending on which vendor you're using, have lived within those products. So, we haven't been able to take those security policies and then put them in the source control system. Because if we were able to do that, then we would be able to bring all of the benefits of CI/CD, this whole process of checking if what we are deploying conforms to the security requirements as well.

So, this is where enabling security as policy, as code, has made a big difference in making this whole shifting security left possible. So, if I may just expand a little bit on that, the industry now has really converged on a de facto model called Open Policy Agent, which has a domain-specific language for writing these security policies. So, before, what happened was that one vendor would encode their policies in Yammer, another vendor would encode it in YAML, another vendor would encode it in JSON, yet another vendor, some of the policies would be in some programming language, things of that nature.

And, so, now this whole migration toward this domain-specific language has enabled us to converge on a common language for importing policies, and this really has taken off in the Kubernetes world, where Kubernetes admission controllers use this language to encode these policies. And now that we have this language and we can encode all of these policies in a very standard way, we can plug it into the CI/CD pipeline. When I, for example, develop some new infrastructure template and check it in, it is possible to check it in the CI/CD pipeline against these security policies, and if I'm actually deploying that bucket where I have forgotten to turn on encryption, it checks against the security policies and actually fails. So, that is the whole model that has enabled this whole process.

**David Linthicum:**
So, are we moving away from centralized security, command and control? Security used to be something that I always envisioned it as being like the bridge of the Starship Enterprise where people are looking and monitoring things, things like that. And getting into the as-code scenarios, we're empowering the developers to invoke security and leverage security in different ways that are going to be more productive because they're more specific to the applications and the data sets that we're securing. Is this decentralization or is this recentralization or maybe just a rebalance?

**Ramesh Menon:**
I think of it as almost a reeducation and reorganization, in some sense, because I think the days of the application teams and the security teams being at war doesn't work anymore, and that's a very good realization that that doesn't work anymore. We have to work closely with the development teams. The development teams have to work closely with the security teams. Because at the end of the day, developers are very good at—they are looking at feature velocity, adding new features to their product. What is the value that the customer derives out of that product? That's the topmost of mind.

And security folks are very good at asking the question, “What could go wrong?” They're very good at coming up with those scenarios, and, so, the collaboration between them and now—the difference now is that the medium of collaboration becomes code rather than structures. So, rather than having a document which actually says thou shalt not do this, this is actually now—and this has to be, it often is, and in the enterprises where I've seen this working very well, it's a strong collaboration between security and development teams because security teams can describe what the intent is, and then the developers often can help write, codify, these policies in this domain specific language that I was talking about to implement those security controls.

And it benefits both. And the way it benefits both the developers and the security teams is that in the current scenario where we are not doing that, a lot of times a developer checks in some kind of—let's say I'm actually deploying—picking on that bucket example, I'm actually deploying this new bucket and I did not turn on encryption, I get a ticket on whatever system I'm using saying that there's a ticket for you. And often the developer's “Oh, I didn't know that I'm supposed to do this.” So, with this model of encoding these policies as code, the feedback is much earlier, and it is possible for the developer to understand it. They're not saying, “There is some set of requirements that I'm supposed to subscribe to, but I have no idea what they are.” Now there's no excuse for that.

So, the way that security teams benefit from it is that now they're not drowning in alert fatigue, right. So, you absolutely still need to do monitoring. Monitoring is essential. And as the feedback loop for how well we're doing, but we're able to get in front of the process in this much more collaborative approach between security and development teams. I think you asked a great question. The organizational dynamics obviously need to come along with it. And this is where this whole—ISC has benefitted from this whole moment to DevOps, and now security-as-policy-as-code I think is going to benefit greatly from this moment to DevSecOps.

**David Linthicum:**
So, what resources do you leverage to learn more about cloud security information? How do you keep up to date? Our listeners would love to know.

**Ramesh Menon:**
I read a lot of the cloud security provider blogs. Largely, a lot of the work that I do is in AWS, Azure, and GCP, and all of them have wonderful blogs that describe a lot of the work that they're doing, how they see some of these issues. then on LinkedIn, I follow some of the influential security venture capitalists, as well, who often point to what are the areas that they're looking at because that's kind of a little bit of a future look into where are the areas that they think are areas of disruption because that's where, often, there's a lot of work to be done.

So, above and beyond that, one of the things that I think—we deploy workloads into the cloud— hands-on work is often a great way to understand what are some of the challenges which brings about those things. So, that's how I keep up on the field. And above that, one thing that I didn't mention is that the IBM and Red Hat, by virtue of their participation in the Linux Foundation and CNCF, have very good blogs that describe a lot of the general-purpose security work that is happening within the CNCF cloud native computing forum, and the Linux Foundation also has a lot of topics that discuss a large part of these issues that I'm actually talking about.

**David Linthicum:**
So, where can our listeners go to find out more about your group and yourself on the web?

**Ramesh Menon:**
I am on LinkedIn at: RG Menon is my handle on LinkedIn, and on the web on Deloitte, we have an entry for Cloud Security Management by Deloitte, which is actually one of the overarching security offerings that we have, and Cloud Security Posture Management, Cloud Security Policy Orchestration, these are all component pieces of that cloud security management by Deloitte. So, on our site, if you search for cloud security management, you'll see a lot of the information that we're working on, again working on trying to move away from outputs to moving towards outcomes, solutions that result in outcomes.

**David Linthicum:**
Yeah, this is very important work. We've got to figure this out because as the deployments get more complex and certainly as we go into multi-cloud, we can't keep doing the same things and expect different results. We have to get into reactive shift left kind of things. Infrastructure-as-code, security-as-code, policy-as-code, governance-as-code, so we're a little bit more pragmatic in terms of how we're approaching this stuff.

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

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

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

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