



Architecting the Cloud, part of the On Cloud Podcast

Mike Kavis, Managing Director, Deloitte Consulting LLP

What is your approach to integrating the latest technologies available in the ever-evolving cloud toolkit?

Duration: 0:18:35

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:

Welcome to Deloitte's Architecting the Cloud podcast, I'm your host Mike Kavis and I am here with Red Hat's Director of Strategy Brian

Gracely. Brian, welcome to the show, tell us a little bit about your background and your role at Red Hat and tell us a little bit about your Podcast.

Brian Gracely:

Hey, Mike good morning, great to be on with you and with all your listeners. My name is Brian Gracely I'm the Director of Product strategy at Red Hat, my focus is primarily around our Cloud platform so OpenShift and OpenStack as well as virtualization, a lot of our focus is around these new types of Cloud Native applications, Kubernetes and so forth. My background has been around technology for a long time I've been lucky to work around the Cloud for the last six, seven, eight years and in terms of podcast, I do a weekly one called the "CloudCast", and I also do a Kubernetes specific one now over the last year called PodCTL, P-O-D-C-T-L. So look forward to talking to you today.

Mike Kavis:

Yes, I will have to start following that Kubernetes one, I didn't realize you were doing that one, – I was familiar with the other one. So, I was at GoogleNext couple weeks ago the big news out there, lot of new buzz words, Hybrid Cloud, my favorite one Serverless containers like Serverless and containers aren't confusing enough now we have Serverless containers. But at the end of the day what the buzz was really about was Kubernetes, Kubernetes as the platform or platforms.

Brian Gracely:

Yes.

Mike Kavis:

And the announcement of a few corresponding open source projects Knative and Istio and without saying anymore, I'm going to let you explain what each one of those does, what their function is and then how is that kind of changing the game as far as – PaaS and portability and on prem versus Cloud and all that good stuff so let's start with, describe what Istio is then we'll get to Knative after that.

Brian Gracely:

The Google folks definitely did an excellent job of hitting on all the really hot buzz words these days, so obviously Google has been heavily involved with Kubernetes, a lot of us have been involved with Kubernetes now for four years or so now, that project has become very stable, very well adopted and I think the Kubernetes community ultimately went through a question a couple of years ago and said "Do we keep adding to the Kubernetes projects, keep making it bigger and broader, or do we kind of narrow the scope, focus on stability and performance and then build some capabilities that allow it to be extended for new types of applications or new ideas that come along and ultimately they chose the latter road, so as part of that they created some extensions that allowed for projects like Istio and so Istio is a project that was announced as being kind of graduated to 1.0 kind of GA maturity wise. But it's a project, it's been around for more than a year, I think closer to 18 months, Google and IBM got started with it, a lot of other vendors have been contributing to it quite a bit. Ultimately in the simplest terms Istio is a what they call a service mesh, which means it's really kind of sophisticated routing that can be used if you are building new Cloud Native distributed applications. So if you want to have sort of sophisticated level seven routing, proxying capabilities, the ability to inject like what they call side cars or agents and stuff into your application. But also the ability to have some logic that says, "How do I handle retries or slow response times or failures or service discovery". So it's technically people are calling it a service mesh, Istio is a service mesh, and it's there to be kind of sophisticated routing for applications and again it's probably one of the first things that we say Kubernetes didn't put this into the main project, but it's an extension that can be added on to the Kubernetes projects.

Mike Kavis:

A lot of people I have talked to kind of liking it to, a resource to manage micro services, is that called...

Brian Gracely:

Yes.

Mike Kavis:

.. a good definition of that?

Brian Gracely:

Yes, I think so, I think it's there to sort of help the big thing that keeps coming out of these projects is they are saying, instead of having to keep adding more and more code to my applications making them more complicated, making them slower to deploy, what if we could just make these kind of native services that run on top of the Kubernetes platform that every application could take advantage of. So, yes it's very much targeted at micro services types of applications, distributed applications and it's really there to say instead of every application building all this sophisticated logic in, make it a native service of the platform, simplify things for developers, make your application code smaller, tighter, faster and let the platform just deliver a bunch of services for you.

Mike Kavis:

And let the plumbers do the plumbing.

Brian Gracely:

You, bet. You bet.

Mike Kavis:

And then keep it simple, right that's a lot of what – people are preaching now, because everything is so distributed, it gets so complex, the simpler we can keep things the better the resilience is.

Brian Gracely:

Right, absolutely.

Mike Kavis:

So what about Knative, what does that bring to the table?

Brian Gracely:

Yes, so Knative is another example of people saying well, you know, there is more that we would like to enable for our developers so, maybe we can add this functionality, Knative is unlike Istio which is like I said kind of mature to 1.0 - Knative is a brand new project so think of it as 0.1 in terms of maturity and newness, it's a project that sort of has three parts to it. I think the headline around it was, it's essentially kind of Cloud independent Serverless, if we dig into that just a hair more it's really, it's a framework to let you plugin Serverless projects, so Knative could be its own Serverless project but you could plugin other Serverless projects that run on Kubernetes so, Fission, Kubeless, Openwhisk, Riff there is a bunch of those. The second thing that it does is it creates this concept that they call "Build", and Build is the idea that if my developers are going to ultimately put their applications on Kubernetes, but we don't want to make them learn about containers, can we just have it push code to this build service and the build service will build the application and then put it into containers, and that's a concept that's existed for a little while concepts like Buildpacks from Heroku and Cloud Foundry, S2I from OpenShift there has been some other tools that have done this, but again this is trying to be an independent way of doing build, so developers don't have to know about containers. And then the third piece of it is, it introduces a sort of an events framework to allow to people to say, "If I'm building applications that need to plug into events so Serverless being a great example, here is a kind of framework for doing that allows you to have different source events output events and again all of this is designed to be independent of the Cloud where you run it, so you could run it on prem or you could run it in the future". So new project people are kind of excited about it and it does potentially a lot of different functionality that to a certain extent kind of sounds like we're bringing back, PaaS or PaaS like opinions on top of Kubernetes.

Mike Kavis:

The value statement I got out of it was Serverless is taken off, but people are little reluctant because its propriety to the different Cloud providers where this sounds like it's open and I can run Knative on top of Kubernetes anywhere, even on prem, that's seems where the value was. But the scary part to me is all the people getting excited about it, almost feels like this private Cloud 2.0, all right? "Oh! I can build everything myself now", so I just wanted your opinion on that.

Brian Gracely:

Well, I think we're definitely, we've been going through this thing for the last couple of years where every time there is a new project that comes out there is this fascination of "Oh Google is releasing another part of what they've done internally and we're going to be able to just immediately be like Google" I think the reality is number one; most of these new tools are explicitly designed for new applications, Cloud Native distributed applications, and so that's sort of the first check box you have to ask yourself is what we are doing tied to new applications because if it's not, if it's lift and shift, if it's modernize, some of this won't be relevant to you. Yes, and the other part this stuff is so new, trying to learn how to deal with this may or may not be in your best interest, it might be best to either acquire it and make it part of some technology that you acquire, you don't have to build it yourself or you just consume it as a managed service. So I think people get excited when there is free access to new toys and new software, but then I think the reality has to kick in the pragmatism that says, "Are we sophisticated enough to deal with this, does this make any sense for applications" and hopefully that will bring people back to reality a little bit.

Mike Kavis:

And also I've been thinking a lot about containers like – if you go back you know to dotCloud who is now Docker, who was using containers under the covers to do live migrations just like all the other PaaS solutions we're doing, right? And they open sourced it and it made it easy for developers to stand-up containers, standardize things, but then we wanted to go production and do tougher scale then this whole

ecosystem built around it, and then Serverless came up and this containers under the covers, and my question to you is, are containers going back underneath PaaS where they started and where they belong or we going to be still mucking around the containers in Kubernetes five years from now?

Brian Gracely:

I don't think there is one answer to that, I think at the end of day the goal of all of these technologies whichever one you pick is hopefully to say software is becoming a bigger part of how our business interacts with the market just like everything else in the market, the faster we can get things into our customers hands, the faster we can make adjustments the better off we'll be, right. So all these technologies whether it's containers or PaaS or Serverless or whatever, are kind of all about making it hopefully easier for developers to be productive and successful and then what you get into is kind of all the different gradients, all the different shades along a spectrum that says "Yes that's the goal but what else do developers need?" Do they need to pick and choose what languages are supported, do they get to pick and choose what other things they need – access to file systems, access to storage, does the operation's team have to be able to do certain things that they see under the covers. I think what we're going to see is there will be customers and companies who build a bunch of applications that are just Serverless functions we've already seen that happen but we're also going to see people that five years from now, are using containers, they're using Kubernetes, the things that those things allow people to have access to will make sense to them and they'll use. And quite honestly you and I will talk in five years and there will be a – there will be a segment of the population who goes, I have no idea what Kubernetes is, we don't, we still run in a mainframe.

Mike Kavis:

It will be like, remember we used to talk about Kubernetes.

Brian Gracely:

Yes, exactly.

Mike Kavis:

Like yourself, I've been in this Cloud space for about a decade now and when I started it was IaS and there was the big PaaS player was Heroku and I think IaS won most of those battles and then PaaS became prevalent and then Docker came with containers and then it was all containers as a service was a rage and then Serverless and now like Kubernetes that's a platform – this stuff changing so fast, what advice do you give to a client who is trying to make a bet on a platform or can they even make a bet on a platform anymore?

Brian Gracely:

You know, whenever these sort of new platforms come along because again like you said most people today if you said where are you, probably virtualized a lot of their environments to save money and their developers are somewhere between, we do some waterfall stuff, we'd like to go faster and we have package applications. Yes there are the bleeding edge people – that's kind of the majority, if you're on the bleeding edge you're always taking a chance of saying "Does this platform with whatever limited capabilities it has, does that make sense right now?". And then for folks that aren't on the bleeding edge, they tend to be looking at and they're going to let things mature a little bit, and I think the reality is if we look at the platforms, and those platforms could be anything from Amazon and the broad set of services that AWS has or the broad set of services Google has or I'll use OpenShift as an example the broad set of things you can do with that. What it turns out is you have platforms, people make a decision that they like capabilities of the platform and the platforms are expanding and they're going if you want to run containers great, I'll make it easier for you to run containers and some Kubernetes will be a part of your life if you want a PaaS experience great, push some code and we'll hide all the rest for you. And then people are saying "Well what if I want to give you functions and do Serverless stuff you go "that's great that's just a feature, that's just another service or a feature that we deliver." And so I think the reality is the evolving platforms, the maturing platforms in the market today people don't really have to make a choice I think what they're finding is "Oh okay, these emerging platforms the ones that are leading tend to have all the stuff for me and I'm not forced to only make one decision".

Mike Kavis:

Sounds a lot easier than it is, especially when you get into large organizations where there's a lot of disparate business units and they are all making their own technology choices you wind up with the kind of alphabet soup of technology.

Brian Gracely:

Yes we do, and that's not but I don't know that's ever going to change in IT, you have groups that want to control their own destiny, you have times when – we see in every technology and in every organization things ping pong between distribute all the learning and the capabilities out to the edge and then at some point that becomes too tangled and costly and things come back and they are centralized we see that both in organizations, but we also see it in and how technology is deployed. So that's not going to change I think, we do the best we can to educate people about, what are good choices we try and educate them on how well prepared they are to take on new stuff, but I think look

there's never been a better time if you are a team that develops applications to go "boy, I don't lack for any tools I – maybe I lack for how to sort through the choices, but man there are a bunch of really good options out there for people today."

Mike Kavis:

So last question and then you kind of alluded to this a lot of these technologies we're talking about are geared towards Cloud Native Solutions but the bulk of the portfolio is that people are trying to get to the Cloud are not Cloud Native, they are legacy and traditionally PaaS hasn't really been a good place to migrate old stuff to. I was at a Red Hat conference it must be three months ago, two, three months ago and it looked like there was a lot of new features that made it possible for some types of legacy workloads to actually port there. So what are some of the answers for the legacy, and we'll take mainframe out and risk boxes, but...

Brian Gracely:

Sure.

Mike Kavis:

...it's got more legacy apps is PaaS now a home for that, and if so what it what would it take to get it there?

Brian Gracely:

Yes, I think I'll speak for us but I think there's some prevailing kind of consistency in the industry I think what we found was people initially said, if legacy applications today were running in VM's do I think about moving them into containers the kind of clouderati part of the market the everything old is terrible crowd says, "No containers don't make any sense, they're only good for stateless applications". We've been much more pragmatic and we said, "Look containers really can be looked at as a place to lift and shift applications think about them as lightweight virtualization there are more and more tools that run sort of container Native if you will, that are helping people modernize those applications, make them work with some of the newer technologies." So for us we look at it as, containers are a good place if you're using them as containers as a service, Kubernetes helps that because it's sort of built-in automation built-in, scaling if you need it, but just automation of deployment which is going to save you some cost, it's going to make it easier for you. We're making it – we're seeing it's easier to attach storage, it's easier to attach networks than we had with virtualization. So containers is a service maybe more so than platform as a service, because platform as a service tends to be very opinionated doesn't necessarily work with older applications as well, is becoming the place that more and more companies are lifting and shifting to modernize, lifting and shifting to help them get into a public cloud environment and it's not the most whiz bang of technology some purist will tell you "Oh that's not purely cloud", but it is saving people a lot of money, it's building an automation for people and it is doing a lot of good for a lot of companies kind of across the industry. So I think it really boils down to, are you interested in just doing new stuff and that's fine there's great tools for that, but if you're trying to take some cost out of the old stuff you're trying to, bring some consistency to your operations, containers are proven out to be really good technology for doing that and then Kubernetes helps make that operationalized.

Mike Kavis:

Well thanks for that, so Brian, thanks for your time today. Where can we find you on Twitter and where can we go to find your podcast and blogs?

Brian Gracely:

Yes you bet, join the conversation, my Twitter is @bgracely so like my first initial last name, I love talk to anybody, DM's are open, the podcast the, it's called The Cloudcast or at @thecloudcastnet on Twitter and the Kubernetes podcast is called PodCTL, so @PodCtl on Twitter as well.

Mike Kavis:

Well thanks again I'm sure we'll get you back on next year, you are "P" defender with my podcast.

Brian Gracely:

Thank you. No it's great to talk to you and appreciate the time today.

Operator:

Thank you for listening to Architecting the Cloud, part of the On Cloud Podcast with Mike Kavis. Connect with Mike on Twitter, 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

About Deloitte

As used in this podcast, "Deloitte" means Deloitte Consulting LLP, a subsidiary of Deloitte LLP. Please see 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 to learn more about our global network of member firms.

Copyright © 2018 Deloitte Development LLC. All rights reserved.