Architecting the Cloud, part of the On Cloud Podcast

Mike Kavis, Managing Director, Deloitte Consulting LLP

Title: **The promise of PaaS**

Description: Platform as a Service (PaaS) offers the benefit of being able to push code to the cloud and get it running quickly. However, its relative inflexibility often causes problems at scale. Enter containerization—especially with Kubernetes. As organizations and applications scale, Kubernetes gives them the flexibility they need to put code into production quickly and scale as needed. In this episode, Mike and guest, Sheng Liang, CEO and co-founder of Rancher Labs, discuss the revolutionary power of containerization (with and without Kubernetes) to deploy applications faster, and to almost any environment. They also discuss common gaps in container orchestration engines and leading practices to close them, as well as management strategies for Kubernetes in the cloud and at the edge. Disclaimer: As referenced in this podcast, "Amazon" refers to AWS (Amazon Web Services) and "Google" refers to GCP (Google Cloud Platform).

Duration: 00:28:39

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 host, 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, and 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 and why, and with people in the field who do the work. I'm Mike Kavis, your host and chief cloud architect over at Deloitte, and today I am joined by longtime friend Sheng Liang. He is a co-founder and CEO of Rancher Labs. Welcome to the show. Tell us a little bit about yourself and about Rancher.

Sheng Liang:

Hey, Mike. It's great to talk to you again. You know, I'm one of the co-founders and CEO of Rancher Labs. We started about four and a half years ago and we're one of the leading providers of enterprise container management platforms. We've been working with technologies like Docker and Kubernetes for a number of years in helping enterprise organizations transition their workload to containers and to the cloud.

Mike Kavis:

Yes, and the first time I met you – it was either the first or second DockerCon.

Sheng Liang:

Right.

Mike Kavis:

I want to say it may have been the first, because there were like 12 vendors at it. They could all fit in the hallway. And I don't even think you guys had a table. I met you guys in a back room and you were discussing stuff. And now you're one of the few that survived the evolution of containers, right? Every DockerCon they would make an announcement and five vendors would disappear. And you're still here and standing and standing on top of Kubernetes at the same time, so congratulations to that.

Sheng Liang:

Yes, it's definitely been an exciting and very dynamic industry. We're not only surviving, but we're thriving. The company's grown quite a bit where we're over 150 people now in more than a dozen countries.

Mike Kavis:

Wow!

Sheng Liang:

Yes, it's – things are really happening.

Mike Kavis:

Yes, that four years ago – it just seemed like it was in the nineties or something. It seems so long ago. That's how fast everything's moving now.

Sheng Liang:

I know.

Mike Kavis:

So, the first question – while we're talking about things changing fast, let's talk about the evolution of PaaS. So, when PaaS first kind of became a thing – you had Salesforce.com. You had Heroku. You had companies like Engine Yard that don't exist anymore. And the original concept of PaaS was more public cloud focused. And, the PaaS was just managed for you. It was just, "Here, build code." And then when we tried to get into the enterprise, the enterprise had this hybrid world so, public cloud's nice, but we still have this private cloud thing. So, things like Cloud Foundry project spun up, and then PaaS kind of became this on-prem thing. But then, "Oh, we want workloads in both," so then those types of applications moved to the cloud as well. So, PaaS has been evolving like crazy, but during that now PaaS became a thing you had to manage, which kind of defeats the original intent of PaaS. Now fast-forward to now, and now containers are starting to – and Kubernetes is starting to feel like it's the – not only have they won the orchestration war but they're starting to become the platform of choice, or at least it's trending that way. But again, managing containers, clusters, is a science. It's hard, right?

Sheng Liang:

Yes, yeah.

Mike Kavis:

So, that's where companies like yourself, they come in and add an extra layer on top of that to make it easier. My question to you is where do you see PaaS going? You know, Google came out with Anthos. You know, there's all kinds of Kubernetes as a service on the public vendors. Where is all this heading? Do you have any glimpse into the future for us?

Sheng Liang:

Yes, that's a great question. I mean, you really can't talk about cloud or containers or Kubernetes without talking about PaaS. As you know, some of the fundamental technologies like Docker came out of – essentially came out of PaaS platforms. And the original idea of PaaS is, you make a platform that's targeting more developers as opposed to operations people, and you basically go straight from source code to a running application really without – without really having to acquire any operational expertise. And then everything would just be managed, then scaled and monitored for you automatically. So, it's really a great vision, and it amounts to…I think that we still think about cloud – a lot of people still think about three layers, IaaS, PaaS and SaaS. I'd say probably PaaS is by far the layer that's the most under-delivered, to put it mildly. But that doesn't mean that the need isn't there, but just for a variety of reasons the platforms and the services that were delivered didn't quite live up to the expectation and didn't quite gain the adoption that they were supposed to gain.

And in some ways, platforms like Docker and Kubernetes really breathe a second life into continuing relevance of PaaS, and in my mind, PaaS remains quite vibrant. It just takes a slightly different form now. So, one form is Kubernetes. So, you could think about Kubernetes as sort of taking the lower layer of PaaS. You know, it doesn't really handle things like building source code or directly managing the applications. It focuses more on the more fundamental building blocks, the containers, the application components, the microservices, and figure out a way to orchestrate them in a portable way in the infrastructure. So, essentially. I'd say the fundamental promise of PaaS is in many ways nowadays delivered by a Kubernetes platform, and that explains why Kubernetes is essentially happening everywhere.

So, almost think about KaaS, Kubernetes as a Service, is a new form of PaaS. And even the private cloud guys, whether it's VMware or the (Inaudible), I mean, they're either supporting already or very quickly going to support basically Kubernetes as a service. You can get Kubernetes clusters very easily out of any type of infrastructure, and that just really has huge implications. Really, for the first time, infrastructure is sort of getting commoditized again, right? Just think about the consequences. So, that's really interesting. So, I would say PaaS hasn't gone away; it just now exists in another form.

And the last thing I want to make a quick comment, is really beyond the Kubernetes layer, some of the other capabilities of PaaS, like the ability to actually manage applications, the ability to build directly from source code into a deployment. What's going on is these days, people just build these very simple add-ons on top of Kubernetes. So, you almost think about it like a micro-PaaS. You know, some people use the term for that. So, the idea is wherever there is Kubernetes, now you can deploy a very simple, lightweight piece of software that essentially brings a very easy-to-use, a very lightweight developer experience, and you can carry it from your laptop to the CI/CD environments, all the way into large-scale production. So, we've actually worked on a project called RIO that integrates software like Istio and KNative, and it's basically a micro- PaaS built for Kubernetes. So in summary I think PaaS continues to exist and that it exists in the form of Kubernetes as a service everywhere and in the form of these micro-PaaS software that can run on any Kubernetes cluster. Mike, does that make sense?

Mike Kavis:

It does, and it brings up a few more questions. I've never heard it phrased that way, the micro-PaaS. So, like, is that a response to the old-school PaaS was always deemed as too prescriptive, and is this now the next evolution where Kubernetes is going to take care of the infrastructure layer, but build the blocks you want on top of that so you're not so prescriptive from the development standpoint. Is that where that's going?

Sheng Liang:

Yes, definitely. I mean, I think you are spot on. And, by the way, you can almost make the analogy between micro-PaaS and microcomputers. So, the traditional PaaS is a cloud, right? It's a – cloud is like the new mainframe and we all know that. So, – and it's great for production but it's not really the greatest experience for a developer. I mean, if you think about it, which developer – I don't mean to undermine great efforts like hosted IDE and stuff. I mean, I think those have a lot of merits. But I would say in general, most developers today really don't like the tethered working experience. And I think I speak for most of the developers. And that's kind of what PaaS, some of the old PaaS, kind of cloud PaaS felt like. Like now you're sort of working with a mainframe in the cloud. And the idea with – that's why it's called micro-PaaS, and it started with – you probably heard of – Mike, a lot of the audience members have heard about it, too, like projects like Flynn and Dais? Do you remember those things?

Mike Kavis:

Yes.

Sheng Liang:

Yes, and then more recently I would put a lot of serverless frameworks in that bucket as well, things like OpenFAST, and like we have a project called RIO. And I think those are all just – and what's interesting is these things are very lightweight and they go wherever Kubernetes goes. And essentially just like there's a computer on every desktop, there's a PaaS, or maybe even multiple PaaS on every Kubernetes cluster.

Mike Kavis:

And essentially on everyone's laptop.

Sheng Liang:

Exactly, exactly. Yes, you run Minikube. You run K3S. Then on top of that you can run OpenFAST. You can run RIO.

Mike Kavis:

Yes, cool stuff. So, the next question we're going to focus a little bit on orchestration engines. So, a few years back there was this big battle, right? Mesos, Kubernetes – and Kubernetes was just released, so I remember taking a class in that and it didn't have 90 percent of the features it has today and I'm like, "Wow, there's a lot to do here." And then Swarm was coming out, and what a lot of people don't realize is that these orchestration engines, they're really good at managing clusters of containers, but they lack a lot of enterprise features, especially around security and compliance and integrating with existing things like AD or what have you. And I think this is an area where you guys are capitalizing on it, so what are some of these gaps that these orchestration engines have and how are you guys addressing that?

Sheng Liang:

Yes, definitely. That's a great question because what we observed was Kubernetes adoptions really going through the work, and Kubernetes adoption happens everywhere: in the cloud, in the datacenter, on the laptop, even on branch offices. So, if you look at it from a perspective of a business, or a large organization, of an enterprise, really managing all these Kubernetes clusters becomes an issue. I mean, these things – Kubernetes clusters are not the easiest thing to operate even though they're really all the same, but it takes a fair amount of expertise to upgrade them, to back them up, to patch for security holes. And it's also not the easiest thing to make sure that it's kept secure because it has a lot of knobs. It has a lot of configuration options. And I think I remember in the early days of Kubernetes, some folks made mistakes like trying to secure the Kubernetes cluster, but somehow left the dashboard, the UI, the admin UI like completely unprotected, so anyone was able to just go in and just be the super admin. So, you really want to make sure things are configured correctly.

You want to make sure that different teams have the right access to the right Kubernetes cluster. And we're not just talking about one cluster; we're talking about multiple clusters. Some clusters are built specifically for dev and test. Some clusters are built for a mission-critical application. And obviously you need to make sure that the right person has the right level of access to these Kubernetes clusters. And Kubernetes provides another level of multi-tenancy inside the same cluster. It's called namespaces. So, you also have to make sure that the right user has the right – you program the right role-based access control rules into every namespace. So, there's this layer of unified control, access control, and unified management that you've really got to take care of.

So, that's basically why we developed the Rancher 2.0 platform. So, we started it like very early on in our company, when the industry was not – has not quite converged towards Kubernetes. We were basically just helping people. We were creating things like Mesos distros and Kubernetes distros, just helping people standing up these different kinds of container clusters. And very quickly we realized – I think it was actually about two years ago we started to realize the future is really Kubernetes. Not only that, the future of Kubernetes – they're not going to be all Rancher Kubernetes, all Red Hat Kubernetes, all Amazon Kubernetes, all Google Kubernetes. In fact, there's going to be Kubernetes everywhere, done by all these different kinds of people, different kinds of vendors. So, the real challenge an enterprise will have is how to manage these Kubernetes clusters that behave the same way, but they are operated and managed and configured very, very differently. So, we started building Rancher 2.0, and it's been on the market for over a year now. It's really done very well.

Mike Kavis:

That’s going to lead me to the next question. We're going to start talking about Kubernetes and the edge, and I recently heard you on a podcast with my good buddy Rob Hirschfeld. I wish I was on it, because I had a bunch of questions, but I'm like, hey, I can ask him now.

Sheng Liang:

Now is the time to talk about it.

Mike Kavis:

Yes, now's the time. So, there's a lot of talk about using Kubernetes on the edge. I'm sure there's reasons or use cases why you would want containers and orchestration on the edge, and then there's sometimes you just want to be right on, the metal or the device. And it's not clear to me, right? I don't spend a lot of time in that space. But when does it make sense to manage clusters out on the edge versus not use them?

Sheng Liang:

Let me explain our involvement in it. We have a great deal of interest in pushing for the growth and supporting the adoption of Kubernetes on the edge, and a lot of it is also for our selfish business reasons. I have to be upfront about that, because our mainline business is to manage Kubernetes everywhere, right? And by everywhere, really, we mean desktop, datacenter, cloud, and branch offices, and the edge. And the reality is in the cloud and in datacenters, even though we still have a business today, like doing Kubernetes distros – it's called Rancher Kubernetes Engine. It's one of the easiest to use and most widely-adopted Kubernetes distros out there. But we really believe the long-term trend is it makes more sense for you to get Kubernetes clusters straight from your infrastructure provider, whether it's VMware or AWS or Azure. So, – and then that's why we naturally have an inclination of looking for maybe there are some other areas that traditional infrastructure vendors or the cloud providers are not going to reach. So, not surprisingly we kind of got there first.

And what was really interesting was about a year ago, I think – at the time we were very surprised. Like, Chick-fil-A, of all the people, published a blog last June describing their experience of building these Kubernetes clusters in every single one of their retail locations. And they were documenting – we didn't even know about it—but they were using Rancher Kubernetes Engine to do it, and they had some questions for us. So, I thought that was quite remarkable, because if you really think about it, what's really going on, was these are the kind of environments typically people run like Windows server – like Windows – actually not even Windows server, like a Windows XP machine, or Windows 7 machine historically, and they run a bunch of retail kind of business apps

And why do they want to use Kubernetes? So, it honestly caught me by surprise. And then it just started happening with – I mean, I'll give you another use case, which actually led to creation of K3S, our edge-optimized Kubernetes distro. So, it was a company called Goldwind and they are the largest – the second-largest wind turbine maker in the world. So, they basically make these very, very tall wind turbines that sit on top of a mountain, or usually in these desolated places like nobody really lives. And then they have dozens of these wind turbines that aggregate into a power generation station. So, it was in that power generation plant that they were – again it's the same story. They used to run a Windows machine. They used to run Windows desktops, really. And then in that Windows machine they were just running some of their apps. And then – and I was looking at what these apps were doing. I mean, these are basically apps that collect data, a lot of sensor data from each wind turbine, and then they do some analytics. They forecast power. They control which direction the wind turbine should turn; how many wind turbines are turned on. And there's just a lot of data input.

So, what happened was I looked at their software stack. It's really amazing. The software stack consisted of Hadoop, consisted of Elasticsearch. It consisted of MongoDB. I think it consisted of Kafka. And there's a bunch of other things I don't even know. I think it's their own software. But just – so it kind of gives you an idea of the level of sophistication that's now getting run in these edge locations. And that's fundamentally why they started looking to Kubernetes, because imagine before. They were like, "How do you really deploy these things? I mean, you deploy them as Windows services?" And it's just a little awkward, right, and let's just say now you deploy Linux, and then you want – you deploy these (Inaudible) units. And by the way, they only have one node, so even like managing the infrastructure isn't even the issue. So, there's something – what they really want is they want, essentially, these services they're deploying, there needs to be a way to manage – to monitor the health, to restart the services if somehow it goes down, and to upgrade these services. So, Kubernetes is a natural fit, and they're basically using Kubernetes as their application management platform. Think about it as the modern version of app server really for the news apps. So, I thought that was really interesting. And that development, by the way – actually we've been working with these guys also for over a year, and it resulted in earlier this year we released K3S, and it's been a huge success. There's just so much commercial and community interest in it, and now people have, every day I hear people talking about running – there was just the other day a surveillance camera company wanting to run these things, not in surveillance cameras but in that box that aggregates, like, 24 surveillance cameras, because again they do a lot of data analysis, right? So, that's not a surprise. So, I would say Kubernetes is probably still a little too big for a true IOT scenarios, like thermostats and cameras. I don't see anyone doing that, but one level above. You know, the gateway that – the surveillance camera controller, like the retail branch office we're talking to, energy platforms, subway stations. You know, these kinds of places, I definitely see a lot of interest running embedded Kubernetes.

Mike Kavis:

Yes, and those are some really great use cases. I'm seeing that a lot. You know, I keep seeing these articles – death of the cloud, the edge is here – and I'm like, okay, you don't know what you're talking about, because you need both, right? You need the edge for some processing and you need cloud for others. So, an example is, say, cruise ships. They're disconnected most of the time, so you need some level of infrastructure and processing out there on the edge, right? But then when they're connected you pull all that stuff back to the cloud, and your mainstream analytics are there, right? So, you used the wind turbine example, and I wrote an article about a similar company years ago, and it was like – I called it "Small Data on the Edge and Big Data Back in the Datacenter." They were, as you said, looking at wind and temperature and vibration, and based on that, they were making real-time decisions, hitting actuators, changing the blade, and basically maximizing output of energy. But then they would bring all that data back to the cloud and they would do deep analytics to figure out why those things happened, and then they would put new instructions out to the edge, to the actuator to make different adjustments. So, there's so many more of these use cases now because you can do this stuff now because storage is cheap, compute's getting cheaper, bandwidth is getting better, and we're seeing use cases – I won't say we've never seen before. We've always had them, but they're easier – I don't know if that's a better word either. They can be solved in different ways now. So, what are your thoughts on that?

Sheng Liang:

Yes, I would say definitely. I just want to agree with you. I mean, just take that wind turbine as an example. You know, these power generation stations do have network connectivity, but it's only one megabit per second. So, it's actually fair reliable but just not a lot of throughput. So, if you think these dozens of hundreds of turbines, they collect so much data. So, that's why it has to be processed locally, right? Like I was listening to a podcast about AI, between you and Simon Crosby, talking about exactly the same kind of scenario. I mean, these days on the edge, you collect so much data that even if it's just traditionally considered small data, it's just the volume is such that you have to process it locally with modern big data AI and analysis software. And that's why they want Kubernetes, and then after that data is processed, then they upstream the information up to the central cloud, where I'm sure they're running additional Kubernetes clusters in the datacenter, or in the cloud, to do additional processing.

Mike Kavis:

Yes, really, really cool stuff. Last question really quick. So, we talked earlier how you guys were filling a gap in the enterprise for what orchestration engines don't have when it comes to security and stuff. I'm sure on the edge there's a whole new bunch of use cases you're trying to fill. Real quick what are some of the issues out on the edge that you guys are trying to plug the holes?

Sheng Liang:

Yes, so right now the things we're going on the edge is – number one is the Kubernetes distro itself, as it turns out that in the cloud or in the datacenter you can solve the operation – Kubernetes cluster operations problem by employing SREs. You know, but on the edge it's a lot tougher. You know, these locations are not widely reachable, and when there's a problem, you're not necessarily talking about fixing the cluster. You're probably talking about rebuilding the cluster or replacing the hardware. You send a technician out there, not an SRE. And on the edge, traditionally you implement HA with say three nodes of XED cluster, but on the edge, it doesn't quite work because a lot of times when you lose power, you actually lose power to all three nodes. So, having three nodes doesn't necessarily even help you. So, it's just a – with K3S we just really did a lot of work to make Kubernetes as easy to operate really in a lights-out scenario as Linux, so you don't really – it's something that will just run without any human intervention. And you can back it up. When it goes bad, you replace it rather than trying to fix it. And all these edge nodes require centralized fleet management and application delivery servers, so that's where we've actually extended. That's built on Rancher. So, Rancher already has capability to manage multiple clusters. But in the enterprise, we were managing hundreds of clusters. On the edge, people wanted to manage thousands, tens of thousands of edge nodes. So, we've got to dramatically improve its scalability and we've got to make the whole thing a lot more lightweight. So, there is just a lot of work we need to do there.

Mike Kavis:

Cool stuff. I'm sure we'll create a nice new buzzword for the technicians. We'll call them device reliability engineers, and we'll be talking about DRE the rest of our lives and have DRE conferences. So anyway, that's our show for today. Thanks, Sheng, for joining us on Architecting the Cloud. Where can we find you on Twitter? You're not ever that much on Twitter, so where can we find your blogs? I'm sure you just go to Rancher.com. Is there any place that we can find some of your content?

Sheng Liang:

Yes, there are a lot of blogs I wrote, and my colleagues wrote about Kubernetes, containers, edge, K3S on our website, Rancher.com. And there's also @Rancher_Labs Twitter account. That's our company Twitter account – a lot of good information there.

Mike Kavis:

Cool. And you can learn more about Deloitte or read today's show notes. Just head over to www.DeloitteCloudPodcast.com, and there you'll find more podcasts by myself and my good friend and 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, @MadGreek65 on Twitter. Thanks for listening and we'll see you next time on Architecting the Cloud.

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

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.

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.