Open Source Voices | Interview Series
Podcast, Episode #02: Going Native: Diving into Containerized Apps & Orchestration
English Transcript

[Black Text: Nicole Huesman]

Hey, welcome to Open Source Voices. My name is Nicole Huesman.

I don't think there's any secret that there's been a lot of buzz around containers and container orchestration solutions.

I'm really excited to welcome Jose Palafox, Cloud Solutions Architect at Intel, and Kelsey Hightower, Developer Advocate for the Google Cloud Platform, to give us their insights. Jose, Kelsey, thanks for joining us today!

Before we dive in, Jose, can you introduce yourself and tell us a little bit about what you do at Intel.
Jose: Sure. I manage a team of engineers who contribute to the CNCF projects. We're mostly contributing to Kubernetes but we also contribute to a number of other projects under the CNCF umbrella.

And Kelsey, tell us a little bit about yourself and what your current focuses are.
Kelsey: My current focus is making Google Cloud easier to use for our customers, so that's exposing open source technologies like MySQL, Kubernetes, and also some of the best of Google, things that we use traditionally internally but have made available for use for our customers on Google Cloud.

Q: So, let's start at a high level. Kelsey, why containers? What do they bring to customers?
Kelsey: There's a lot of talk about containers, I think, mainly because most people don't understand it. So, it's easy to see containers as a placeholder for an easier way to package and distribute my applications. So, whatever you're building, traditionally, if you're building in a specific programming language—let's take Python or the Go programming language—you'd have a very specific workflow for getting the application packaged, getting the application deployed, and that would dictate a lot of the ways you go about your day-to-day development activities. And then, containers show up, and there's new tooling that says, 'Hey, let's abstract that away and give people a different way of thinking about packaging and deploying their applications in a standard way,' which has kind of unlocked this new ecosystem given this area of standardization. You can now start to target better tools for doing the things you were doing before.

Q: Are there any drawbacks around containers that you've observed along the way?
Kelsey: Like any technology that's new, you usually see mis-use. If people don't understand what it is, then it's really hard for them to use it correctly, right? Once you package your application in a certain way, there's responsibilities for any software, right? If you are building an application, you're responsible for ensuring that things inside the container are up to date, meaning updated for security vulnerabilities. You can't just build a container six months ago and ignore it, right? There's things that need to be patched and updated, just like you would see in traditional deployment methodologies. So, you kind of see some mistakes there where people believe that the container can be treated like a black box. I think I used to quote, 'If you use a container like a black box, eventually you'll be left in the dark.' And I think that people need to understand that there's still responsibility there even if it makes day-to-day easier.

Q: So, Jose, let's bring you into the conversation. What have been your observations?
Jose: Well, I think I agree with what Kelsey is saying—it's a great packaging methodology and where the conversation has shifted for me in my role has been off into the logic to manage all of those new units. Kelsey and I both worked at a company called Puppet Labs a while ago where we were thinking about helping systems administrators manage the amount of virtual machines that they were creating because the number of discrete things they had to manage exploded. And I think we're seeing a similar trend in containers in that there's a whole bunch of extra stuff now to manage. The bits that people are managing are more discrete and smaller, and there's more of them, and so a lot of my interest is the move towards these management tools, and containers are the way of delivering some of those smaller features but they're not the whole problem, or even the solution.

Q: Can you tell us a little bit about how containers and container orchestration, like things with Kubernetes, have really changed or upended traditional config management?
Kelsey: You know, we launched this new paradigm or compute unit and closely followed with tools to deal with it, and some of the best patterns we've seen over the decades at a few companies that think about things like service discovery for everything that's built around the application. So, the more applications you deploy, the system or the orchestration platforms like Kubernetes have all these built-in practices about how to deal with those things, which is drastically different than how people were treating virtual machines, right? People would take physical machines, made them virtual, and for the most part, most people kept the same automation tools, the same practices, for dealing with virtual machines that they did with physical. But in the container world, these platforms also brought in the best-of-breed approaches to dealing with applications, not necessarily servers. And I think the difference here is, configuration management is borne in the server world, right? Everything is server-centric, you put applications on servers, and then servers become the anchor of all that technology, all that automation, whereas in the Kubernetes world, Kubernetes tries to abstract away the server as much as possible, and then what you're left with is tools that deal with the application, which is probably the biggest difference between what we were doing before with configuration management and what we're doing now with containers and orchestration platforms.

Q: So, there's really then this shift to abstraction and orchestration with Kubernetes. Jose, can you talk a little bit about how Intel has been involved in that effort?
Jose: What Intel I think is focused on is in this space is making sure that, where there is value added by knowledge of specific underlying hardware, we make that knowledge available to the users. So, there's certain high-value, non-generalized compute platforms that are available in a data center. Whether that's GPUs or other types of accelerators or special-purpose cards for offloading, we want to make sure that those are available to end users to take advantage of in their application without having to be aware of where specifically they all are and which servers they're attached to. So, I think what Intel is really focused on in, at least the short term, is making sure that operators can take advantage of our specific technology differentiation in the context of Kubernetes, in the context of their applications.

Q: And Kelsey, what are your current focuses in relation to Kubernetes, and driving that effort forward?
Kelsey: So, right now Kubernetes definitely does a lot from the operation side. So if you were managing machines before, Kubernetes makes your life a lot better in that regard because it abstracts a lot of that stuff away, it gives you tools to manage a large number of machines. But once you get to the application level, Kubernetes does offer a few things out of the box, like service discovery, managing data volumes,

routing network traffic. But at a higher level, the developer wants a much easier work flow than thinking about all those things. So, given the Kubernetes foundation, which is built on top of the machine foundation, probably more accurately the Linux kernel for example, as we think about the next level of abstraction, what will come and layer on top of Kubernetes to ease the development work flow. And I think that's the job I'm really focused on now. How do we improve where we are now? It's been almost five years since Kubernetes has been out, so we need to start looking at what comes next before we sit around and get complacent.

Q: What do you view as some of the challenges in looking ahead? Jose, I'll give that question to you and then pass it to Kelsey as well.
Jose: I think the biggest challenge we're seeing in the industry still is on the front of the adoption curve. Using Kubernetes and going to a cloud native design format requires a lot of enterprises to redesign how their applications are built and how their development workflows are built. So, retraining people to think in this higher level way where they're dealing with a massive compute rather than dealing with individual servers, and letting them get comfortable with how that changes the way that they design their applications and interact with them. That's still where we're seeing I think the biggest challenge in the enterprise space today.

Q: And Kelsey?
Kelsey: I think the challenges we see ahead are the same challenges we've had forever. I think a lot of people, especially in the software development space, or tech in general, have yet to understand that it's never going to stop changing. You're always going to have to pay attention, you're always going to have to adapt. That is not optional. I think traditionally enterprise IT has bought systems that they believe that, 'If I buy it, it will be good for ten years. We'll use it for ten years, and then we'll look up to see what we can buy next. And the thing that we buy next, we'll figure out how to transition from what we have to the next thing, and all the pain that comes with it.' So, that challenge then starts to burden the future, right? It's like, this thing that was a great idea ten years ago, some of the decisions that we made ten years ago, not with the forward-looking view that, 'Hey, we may need to rewrite this, or we may have to change and modify it to meet the new technologies that come out.' Most people didn't think about those things then. They're still making the same mistakes because they're not thinking about them now. So when new technology shows up, once it gets awareness, a lot of people try to bring the past forward. 'I want the new thing to work like the old thing.' And then we spend a lot of time and energy trying to meet people halfway or all the way to make their previous decisions relevant in the future, and I think that just continues to pose a challenge. So, what we have to do in these new systems, which I think is helping, is they're all mainly declarative API driven, meaning that it's probably going to be easier in the future to translate some of the decisions we're making today into something that's compatible for the future in a more automated way versus a brute force way of hiring legions of people trying to reverse engineer their old stack to run the new stack.

Q: Jose, how do you see this landscape of containers and orchestration evolving such that it supports new usages and necessitates new technologies?
Jose: I guess the big trend we're seeing from my seat inside of Intel is a lot of effort to push some of this new technology to the IoT edge or telco edge. So, that's where a lot of our efforts are focused. It's actually not in larger clusters but in smaller ones. We have a project now to see if we can replace a lot of the microcontrollers in a car cockpit on a single server using Kubernetes and a set of functions to lower

and raise windows and display dashboard information on a driver panel. So, I think there's a lot of work that we can do to take the improvements we've made in the data center and move them out to the edge, and so that's where I think a lot of this headed in the near future. That's the thing—the data center is going to continue to transform and I think probably drive innovation forward for the next couple of years, so we'll see what happens there, and then apply those practices out to the edge where we can.

Kelsey: And I'll jump in there as well. Once you actually have a system like Kubernetes, then the act of using exotic hardware becomes slightly easier. So if you had a data center with 10,000 machines that you bought five years ago, a lot of times you're going to be kind of locked in to those capabilities of five years ago. If you bring in three new servers that have, let's say, FPGAs, some machine learning, some specialized chips, you can expose those things, but the challenge then becomes, how do you roll those three new machines into your existing fleet and then make it easy for the various teams to consume or share—especially when we only have a limited set of those—and Kubernetes, one of its core capabilities is this idea of a scheduler. So, as you register those three new machines into your fleet, then they can expose new capabilities. So, this machine has ML capabilities or it has the new Intel larger memory that's persistent between reboots. So, once you label those set of machines with these new capabilities, it's easier for the developer now just to describe very discretely, to say 'Hey, I would like to use some of these things' and let Kubernetes decide how to get that application on to that new hardware without necessarily retraining the entire organization or changing the tools you use to do your deployment. So, I think that's where the ability for people to bring in new hardware pretty quickly without thinking about re-provisioning and repurposing everything else. I think that's kind of the thing that Kubernetes enables on the custom exotic hardware front.

So, thank you so much, Jose and Kelsey, it's been fantastic to talk to you today. We've covered a lot of ground and there's still so much left to cover. We'd love to have you back on for a conversation in the future. Appreciate you being here today.