Cisco Cloud Unfiltered Podcast Series, Episode 11: Ken Owens

This week Cisco's CTO of Cloud Native Platforms, Ken Owens, weighs in on the benefits of containerization, security in the cloud native space, the divide between infrastructure and apps teams, and more.

Niki Acosta: Good morning, good afternoon, wherever you might be. This is Cloud Unfiltered, I'm Niki Acosta, you're hostess with the most-est, and I've got a really great guest with us today, our very own Ken Owens!

Ken Owens: Hey, Niki, thanks for having me. I'm looking forward to being unfiltered today.

Niki Acosta: Yes, and a much thinner Ken Owens. I couldn't believe it, I haven't seen you in a while, but I was like, "Man, Ken looks fit, good for him."

Ken Owens: Yeah. It's amazing what not traveling as often can help you with.

Niki Acosta: That's where I usually get tripped up. I find that being on the road it's really hard to eat healthy, especially when there's just so many good things you want to eat. And regional things, and people you want to go eat with.

But let's get to it. You know, we typically start out by asking, you know, "How'd you get to be where you are today?" You've obviously been in technology for a long time, but take us back to when you were younger, what were you like?

Ken Owens: Yeah, well, as a kid I always liked to break things and then try to fix them, and so, you know, my parents always had me out of the house trying to fix other things besides things that I'd break in their house. So engineering became very natural to me, I loved doing engineering, which I do electrical engineering. So I started off my career doing sort of telecommunications and some of the early days of ATM and frame relay and the convergence of the two, and then moved into MPLS. And so ... Moved from a telco to a tel-labs where I was building routers and switches, to a start-up company building network processor ASICs and switch fabric ASICs to sort of when the economy crashed moved into IT.

So I kind of got a background with an enterprise company working at two different banks, sort of leading up architecture and engineering within the bank environment. And then when an opportunity in St. Louis opened up with a hosting provider called Savvis Communications, I jumped over to Savvis and spent a lot of my early cloud days and,

you know, kind of cloud computing and platform as a service type of efforts at Savvis and then moved over to Cisco to do Intercloud.

Everywhere I go I sort of keep the same theme of, you know, focus on customers, focus on what their needs are, what they're looking for, and try to stay as connected to them as possible, to their needs and to what they're looking to accomplish. And then organize your teams around speed and focusing on execution and delivery, because, my motto is, working in code will out beat a PowerPoint anytime. And so I try to make sure I live by that mantra and have running code for demos and not just a bunch of PowerPoints.

Niki Acosta: You know, Ken, one thing that I think is really sort of fascinating about you is you've managed to kind of work on, I'll say, kind of a traditional infrastructure and now you're kind of this guy who's at the bleeding edge of technology. I know that you work with our DevNet team, but you're also a Technical Oversight Committee member, correct?

Ken Owens: Right. Yep, that's correct on the Cloud Native Computing Foundation.

Niki Acosta: And what is that foundation? What y'all work on, what do you do?

Ken Owens: So in the Cloud Native Computing Foundation, about a year ago, Google was looking at taking Kubernetes and Open Sourcing Net, and they weren't sure exactly the right way to do that. So, they worked with the Linux foundation to put together a Cloud Native Foundation, where Kubernetes is one of the many projects in the foundation today. The whole concept around Cloud Native is the industry is confused. There's so many terms, and so many ways you can say something is Cloud Native, and over the years, the confusion just seems to keep growing in the way terminology gets thrown around and open source projects take off and mature.

So, the goal of the Cloud Native Foundation is to help find what Cloud Native means. There's an executive steering committee that Cisco is one of the founding members for, and then there's a group of technologists that were voted in by the community at large, as to individuals that should represent what the technical aspects of Cloud Natives means, and I'm on that technical oversight committee.

Niki Acosta: So what kind of things do you get to make decisions about?

Ken Owens: In the first year of the foundation, it was really about the defining what we want the foundation to do, and so the first decision we made was we're not going to allow to happen to Cloud Native, what happened to OpenStack, and with like other projects and other foundations in the Linux foundation. Meaning, we don't want to create a big tent, or create a bunch of projects that people go off and develop their own technology.

What we wanted to do was sort of, say, there are multiple ways to accomplish Cloud Native solutions, and there are multiple projects that are having success and the enterprise space doing that. So, let's go talk with these companies and these projects and say, "Here are best practices and things that have worked for these enterprises. Here's how you should try to start about this journey towards Cloud Native." And we'll get more on that journey later.

But, the whole aspect of, let's start with what's popular, what's getting adoption, and what's being successful, and let's kind of build a foundation around that ecosystem, and try to help enterprises and users how to go down this path, and give them some support as they're going down this path. That's kind of the first decision we made.

The second decision, and what I've been involved in, is bringing projects into the CNCF, and so I work closely with the container networking interface work group. They've defined, sort of, some very basic primitives around what you should expect from your

underlying networking interface for Cloud Native. So, we brought CNI into the CNCF as a project that's part of our foundation, and I'm working with other projects across the industry right now, to look at how do we support them, how do we bring them into Cloud Native Foundation? Do they make sense in the foundation? Do they meet our criteria for what we want as a project in the foundation?

So, I spent a lot of time working with individual projects and talking with users of Cloud Native technology to understand and help determine what direction we want to take the foundation.

Niki Acosta: So what is CNI in a nutshell?

Ken Owens: So, CNI in a nutshell is just an interface specification for, how do you request a network interface, how do you assign an IP address. If you're familiar with Linux commands, it's very similar to if interface up and do net stat dash and you get the routes for the container network, right? So, it's basically taking some of the basic primitives of a networking interface, like defining your gateway, defining your DNS, defining your net mask. So, the basic IP configuration list, and it starts with that, just making it so networking individuals, or developers, can understand how to request a specific IP address, or request a specific DNS server to be their DNS server.

Niki Acosta: So, from the Cisco viewpoint, obviously Cisco is a company that has it's roots in networking, and in a lot of ways, defines the space around networking ... But, what do you see between Cisco's traditional product lines and, sort of, moving to this container networking model, like what advantages does it have over some of the more traditional networking models?

Ken Owens: I think there's two ways I like to look at that question. The first way is, depending on the use case, traditional networking is what you have today. So, in most cases, I don't really see it as an advantage or disadvantage. I see it as an integration point; how do you take on a containerized environment into your existing network without rewriting how networking should work, right? The example I like to give is, when everything is working well and you have no problems, you don't care where your container lives, or where your services live, right? But in the real world, things break, and things don't function the way you expect them to. And you need to know where that service is that just went down, because your other services, while they're loosely coupled and don't require that service to be available, you can't create new services without that service, right?

So, you've got to ... To some extent, you kind of have to understand what is happening at that underlying network layer, whether it's an overlay network or a physical network, you want to have visibility into that underlying infrastructure layer. So, you want to kind of extend your existing networking capabilities into this container space, if for no other reason than to just have a better view of what's happening.

The second aspect is really important to your question, right, why containerize? Why look at Cloud Native, right? I think there's two ways that I like to answer that question, too. The first one is, as you look at structuring your application and the way application availability is so critical today, and the way your reputation sits, in a lot of times, on the way that application behaves and performs. So, Cloud Native -and just containerizing and breaking up your monolithic application into multiple independent services that can all be run and maintained, and distributed independently of each other and scaled independently of each other- allows you to be much more reactive to your customers, allows you to be much more available, and get your uptime a lot higher, and allows you to, in a lot of ways, extract yourself from that underlying hardware or services that you're buying from a provider. Because, you're now ... You're writing your code at a function level or a service level that doesn't require any underlying definition of hardware or services from a provider.

The other aspect of that, that I think is important, especially with the networking question you have, when you're connecting your services together, you don't want to have to train people how to do some kind of new network model, some new network way of doing things. You don't want to make it so someone has to stop and think, "Okay, I need an IP address. How would I configure an IP address for a container?" And then make them go off and type commands that have no reality in what the rest of the world does today for networking.

So, I think it's really important that as you look at developing containers and microservices, you want that networking to be consistent with what you would typically do on any sort of Linux device or Windows device today, so that it's familiar to you when you go to type in those commands. That's why I think, if you keep those two views in mind, you're extending your existing network in a way that makes sense for your developers, and is similar to what you would do for any type of a networking configuration today. That's why CNI made so much sense to bring in to the CNCF.

Niki Acosta: Got it. So, how does security fit into all this? You know, often times when you're hearing about in networking, there's always this idea that our containers at the network layer are secure, are they secure enough? How does this impact other security measures you might take? It seems like that would be a high priority concern.

Ken Owens: It definitely is. There is some, I guess, agreement across the industry that, in general, software development is more secure as you limit that attack surface, or building things as a container limits that attack surface area. Now, that doesn't mean that just having a container makes you more secure, you still have to write your software in a way that that surface area is not open to attack, right? So, there's still development needed, and secure processes to keep in place when you're developing in Cloud Native, but decreasing that attack surface is important.

The area that is really important, now, in Cloud Native that is just starting to be discussed is, in the same way that you looked at cloud computing many years ago, networking and security were some of the last components to move towards that cloud model, right? Security is just now becoming more virtualized and more cloud-enabled. And we're automatically starting to change this now to Cloud Native and containers, and most security solutions don't understand containers today. They kind of operate at a layer below the container, or way above the container, and don't really understand that they have a set of services underneath them.

So, one of my concerns, and it's a little bit off topic, an example of where I can see a concern is where in IOT, as you have these services being registered automatically through Kubernetes, how do you know that that service is really the service that it says it is? There's no way to validate that. There's no secrets, there's no security model parameters in place to say, "Yes, this is a DNS service, and it is the right one I'm connecting to." So, I think over time, you're going to see security bobbing quite a bit in the Cloud Native space to address some of these, I guess, security by design, is what I like to call it, aspects of when I connect into an interface, I'm going to validate that's the right interface and that I'm not in the middle of an attack by some service that is ad hoc connecting into me.

Niki Acosta: Which is crazy, I think there's a lot of parallels between, sort of, that and what's happening, just in general, at the consumer level, ya know? There's a lot of crazy things happening with security, it seems like there's some people that have a, sort of, nefarious intent, that seems to just stay a step ahead. I read articles on security, even when it comes to clicking on links in Twitter. Things that would seem relatively innocuous. I've certainly got some suspect stuff form family members that I'm like, "Is your computer working? Did you really send me this?"

So, I think that it's always good to think up that stuff at the beginning. It's good that the foundation is taking measures so that it's not an afterthought, I guess.

Ken Owens: And we're taking it from the standpoint of open source projects today. So, we're trying to work with the communities of interest that are doing open source projects, but we're not completely closed off to companies that have products in this space, either. So, we have a landscape document on ... If you go to GitHub CNCF, you'll see that there's a landscape that we keep updating on GitHub, that has a list of security projects and products that we are interested in working with and trying to figure out how to make Cloud Native more secure.

So, again, kind of different from other foundations in the industry, we're not trying to go off and create a whole new security model. We're trying to take what the industry accepts as good security and working security today, and help amplify that message and help integrate it with the other projects in interest, and that customers are having success with.

Niki Acosta: On the topic of, sort of ... This is kind of related to what you're talking about ... I gotta give OpenStack a lot of credit, obviously, I might be a little biased because I spent a lot of time in OpenStack and in a lot of ways, I think OpenStack had to start from scratch, when it comes to creating a massively scalable platform and collection of projects and a community. And they did a really good job, and they still continue to do a good job on the community.

But, it sounds like there's almost a case study at this point. How do you think that ... Where do you think OpenStack has fallen short, in terms of, you mentioned big tent, but in a lot of ways, do you think OpenStack is still on the right path?

Ken Owens: I think the nice thing about OpenStack is that it keeps correcting itself, and working on making it better. So, I think Mike and the OpenStack board does a great job of listening to feedback, trying to adjust the way they're going forward. To me, where OpenStack has struggled, recently, is the industry is moving towards smaller containerized set of services, and OpenStack, as you know, is a bunch of big, large monolithic services. The projects you have to bundle together, and the complexity is difficult and a lot of people ask me why would I deploy OpenStack? Why would I not just jump straight into Kubernetes, and just kind of build my own set of services? The conversation is usually around ... There's a big difference between OpenStack projects and Cloud Native projects in that *Cloud Native projects sit within an OpenStack framework. They're not separate, they're not outside of it.*

They're not a whole new way of doing infrastructure, they're just a new way of developing applications and they need to leverage the existing infrastructure, whether it's OpenStack, or Microsoft or Amazon or any other cloud provider, or any infrastructure that you have internally. You want it, kind of like I mentioned with CNI, you want to compliment and integrate your existing deployments with Cloud native models and methodologies and practices. You don't want to try to recreate the wheel, and go off and build your own new set of capabilities. And so, OpenStack, it's still very relevant and there are a lot of large enterprises that want to run and operate their own private cloud, or their own hybrid destiny.

They're not going to feel comfortable putting everything out in the public cloud, and so ... I mean, OpenStack has a great home, and what OpenStack has done recently is listen to the community and trying to fit Kubernetes and Cloud Native patterns into OpenStack. I think that's what is really important here, we figure out a way to integrate Cloud Native and Kubernetes with OpenStack, versus trying to make them separate discussions.

Niki Acosta: So, are we fighting for a, I'll say, control point mind share? Where should the control point be? Depending on who you talk to, it seems like ... If you talk to networking folks, they seem the control point is the network. If you talk to application folks, they want to be able to define their control point through their app, or through the deployment tools they're using for their app. Where does that control point, where does it live?

Ken Owens: So, I mean, there's simple ways that you can look at that question, Niki. You know, in my mind, the control point almost always lives with the business, and with who owns the P and L for the business. So, yes the infrastructure team is important, yes the networking team is important ... And take that to the next level, companies like Cisco, and Arista and EMC are all very important in the future of where this industry is heading. There's no taking away from that, but I think we got to this point of a big divide between infrastructure and applications because the application team has a job to do. They're making money; they're trying to grow the business.

The infrastructure team is a cost to them. They're basically trying to put processors and secure the infrastructure, and help make sure the company isn't going to put themselves at risk. Those two things collide with each other. When the infrastructure team is delaying a deployment of a very important application or update to an application, for very good reason, I'm not saying they had bad reasons for doing that, but when they hold it up and say, "I need a change window on the second Thursday of the month to make this networking change, so you can then deploy your application," the application team is like, "That's bullshit. I can just go to Amazon and deploy it right now, I don't need you. You're slowing me down because you have this process that I don't care about, and it doesn't help my business grow."

So, we got to this over years, as you know, with infrastructure and admin application teams fighting about this, and it's, to some extent, good we got to this point because I think, what I see with Cloud Native now, and what I want to get into this conversation with you is exactly that point of, there's a transition that's happening in the industry. I don't think the infrastructure team is bad, and I don't think the application team is bad, and there used to be a point in time, because I remember back in my IT days, you would sit down in a room together, with the architects and with the lead developers, and figure out how to make this work the best. And we miss that now, and there are a lot of things you can do with infrastructure that the application team has no idea about. And there's a lot of things the application team can do that the infrastructure team has no idea about.

So, if you're both kind of like ships in the night, and you both doing best practices and you never share those with each other, you end up with an inferior product in the market place, and that's where, to me, helping as a business, to come together with your infrastructure team and your application architects and say, "How do we work together in a way that drives the business at the most optimal speed with the least amount of risk that we can tolerate?" And at the same time how do you involve your networking team and your storage team and make them more Cloud Native architects and Cloud Native solutions engineers, that are solving problems, and applying their vast knowledge of improvements and performance and security at that infrastructure layer to software development practices with software defined networking, software defined storage, software defined whatever you want to call it, right? And how the application teams would know what interface ... How do you know what defines this API between the infrastructure team and the app team, so that the app team doesn't have to worry about what storage vendor or networking vendor you've deployed, and what its specific interfaces are, it just knows interface up, and I get a network.

I need this DNS, put it into the config, and you get that DNS, right? The more automatic you can make that infrastructure responsive to that application requirements, and the more that infrastructure team understands the value they bring to the application, I think those two worlds coming together is what this transformation is exciting for, because

we're seeing that now. It's not just Ken being crazy, saying, "Why can't we all just get along?" It's really these two teams are coming together again and figuring out what's the best way to be successful as an organization versus silos that existed for many years now in IT and in the business that either don't operate or don't share those practices, and don't work well together.

They're now breaking down those silos together, and that's what's really great about this whole transformation that we're seeing happen.

Niki Acosta: As far as this transformation goes, it definitely seems like this isn't just ... The things that are impeding, I'll say that the transition, don't seem to be technology so much as it is culture, and processing.

Ken Owens: [crosstalk 00:24:07] We always talk about technology, process and the time. Those three things, right, technology, process and people, those three things I mean. The people ... My view is, you have to start with the people, and you have to lay out this vision of what you want to get to. And some people may not get to come along that journey with you. That's not what you're hoping for, you're hoping everyone gets on board and says, "Let's go." But, you have to be honest with your teams and your individuals and make sure that they're ready for this transformation.

Then the process, like I've mentioned multiple times in presentations, we have processes in place for a good reason. We didn't ... I think application teams are younger now, so they don't realize that there's been lots of big problems that have happened in the internet because process didn't exist. So, the problem is that process was defined ten, fifteen years ago, and nobody has ever gone back to look at that process and see, is it still valid today, or probably better question is, "How do I modify this process to be more supportive of what we're trying to do today?" I hate to say this, but I have told enterprises to kind of go back and question every process and every procedure and try to understand, not why they have it, but how does it apply to me today? Is this outdated and useless procedure ... What was the intent behind this procedure? Is that intent still valid going forward, okay? How do we make that intent relevant for today?

And then put that into some form of CICD pipeline so that you're continuously evolving, and continuously looking at those processes and every iteration you do, and if that process needs to be update or changed, it becomes part of your development cycle. Not part of your afterthought, but not a, "Oh, we forgot about this process and now it's ten years again, and we have the same problem we have today."

You laughed, but we've seen this happen so many times. People forget why they do something, and it becomes an inhibitance to moving forward, and no one can tell you why, because the person who wrote it has been gone for ten years, and no one knows why he wrote it in the first place, but we have to follow it.

Niki Acosta: I think that's applicable, not only just to developmentive apps and infrastructure, but I think it affects most organizations in other areas. I've definitely seen there's always departments, and it seems the older the company, the harder it is to shift. But when it comes to sort of control and policies and what, really, they're trying to do is avert risk, and make sure there's not a bunch of rogue stuff going on. One example, we were talking about it before was you guys are using the DevNet. The DevNet Create site has a site on Medium, right? And you've made the decision to kind of pull that off of the Cisco.com domain, but use Medium instead. What was the driver behind that decision, and how do you make folks who may not agree that's the right thing to do, how do you get them comfortable with that type of decision?

Ken Owens: Definitely. Well, the decision to move to Medium, in the case with DevNet, was, I think started with we want to have a focus on a developer audience, and so, for those of you who don't know Cisco DevNet, we are a group within Cisco, engineering team, that is

focused on developers, and what developers are trying to do, and the multiple areas that Cisco functions in. So, Cisco is one of these large, as Niki mentioned, very large companies and we have IOT, we have collaboration, we have networking switches and routers and access points, and we could go on for 20 minutes on all the networking stuff we have.

We have security solutions, we have different business and different verticals, like in the government and in healthcare, and in finance. So, we are a very large company, but across all of that is a focus on, how do we enable developers to leverage Cisco solutions through our APIs and through our networking interfaces. To do what I kind of mentioned earlier, how do we make it easier to do unique performance to kind of differentiate yourself from your competitor? How do we make it easier for you to take advantage of advances that Cisco has made in networking and security to improve your product availability and your product's performance, so you can outshine against your competition.

So, DevNet started out just three years ago, if you can believe that, it's a very young group within the environment of Cisco. We work with all businesses across Cisco to just, sort of, enable easy integration. We have sandboxes and learning labs. You can learn about technology, and get hands-on in user technology, and understand how to use the APIs, and so ... DevNet is kind of focused on this specific audience that doesn't go to Cisco.com. So, we could've kept using Cisco.com, but we never would reach the audience we're trying to reach, so the business decision was made to move to where our audience is, which is on Medium.

So, we moved blogging and content to Medium, and started publishing and promoting what we're doing on Medium. And we've had tremendous response, much more followers on Medium than we ever had on Cisco.com and it's not because Cisco.com is bad or ... Don't take that I'm saying Cisco.com couldn't reach that audience, it's just that that audience doesn't go to Cisco.com. But they do go to Medium. Now, we're able to cross-link things to Cisco.com and Cisco.com can post that we're doing things on Medium, and we're kind of bringing those groups together now, and the philosophy is that if we bring the two groups together, you're gonna get growth in both Cisco.com visits and DevNet to community because you're driving more and more users to look at what Cisco's doing in the space.

Niki Acosta: So I have Tim Crawford on the podcast, he's a well-known cloud guy, and he was making a point just in general about the fact hat companies are always eager to go and "market" to CIOs. I think in a lot of ways, there's a lot of companies out there that wanna go "market" to developers. But that's really ... I don't see the words "marketing" and "developers" ...

Ken Owens: They don't go together.

Niki Acosta: First of all, why is that?

Ken Owens: I think the main reason is developers ... I think of developers in two ways, there's the ones that you and I are thinking about the most, which are the guys that are head down, or gals, heads down writing code. They don't look up, they don't communicate with other people around them, other than their team, and they're just focused on driving revenue in the business. Then you have the other group of developer guys and gals, I look at are more of the architects. They look across the industry, they try to understand trends, they look at open source projects, what are other people doing? And they try to bring that in to the heads-down developers that are just writing code all day long, and just love to write code.

So, as you look at these guys and gals, they're not just interested in what the market is trying to tell them, right? They know what the market needs, they know what they're

working on. They're looking at what's happening around them and they're trying to figure out how to take advantage of that. They don't have time, or want someone to coming in and tell them what they should be thinking.

So, to answer your first question, developers don't have the time of day for marketing, because it just wastes their cycles, and they don't have time to waste cycles.

Niki Acosta: But, if you could Google something, and end up on a DevNet site and it has a snippet of code, or a pro-tip of some kind, then that becomes valuable, right?

Ken Owens: Right, and I think that's a difference in what I was mentioning before. I didn't say we were marketing to the developers, we're trying to reach the developers by going to where they are. So, to your point, if we can provide relevance and content, and they pick up that content in their day to day, how do I configure X or Y, or what's the right way to ... I have this Cisco product, how would I write code to make that Cisco product do something? They're gonna find us on Medium, because that's where they're searching. And they're going to be happy that they found us, because we're gonna have sample code, we're gonna have examples of how to do it, and what else you can do with that device, that they can then take and use right away, without having to reach out to Cisco or ask Cisco for help. They can just take what we provide, and just go with it.

Niki Acosta: I see it very much, in exercise and efficiency. It seems like developers are going to go wherever they can get the fastest, most complete, most reliable answer for the question that they have that doesn't involve reaching out to a bunch of people and waiting.

Ken Owens: Yep. I was joking with a couple of my friends a couple weeks ago at one of the conferences I was at, one of the DevNet, not DevNet sorry, at Dockercon, I was talking with a couple of my friends, and we were commenting on how we haven't written any fresh code in years. I mean, we write code all day long, but we pretty much almost always Google something, find something that's close to what we're trying to do, and we start with something that's already been done. We don't try and reinvent it ourselves.

I don't think that's the common for ... I'm not trying to put down application developers, cause I know they're inventing new code all the time, and they do a great job at inventing code, but to your point, if they're trying to figure out how to take their new invention and work with something like a Cisco product, they don't want to invent that. They want to go borrow some code that's been written, have ideas of snippets of code that have been successful in doing what they wanna do. And you're absolutely right, it's all about efficiency and the least amount of time it takes to get product out the door.

Niki Acosta: Which, arguably, is happening on the consumer. I think about all my favorite apps that I use, and all the apps that I download are ones that are going to save me time or make my life easier, or help me accomplish whatever I need to accomplish the fastest. Like, I used to drive to the store to get stuff scanned and faxed, and then I found the Scanner Pro app, and I'm like, "Oh my gosh, what a life changer. I can do all this from my phone. I don't need to go through this process of hooking up a fax machine or wondering if ... "

Ken Owens: This might be for another discussion time, cause I think you're onto something very important here, and I watch my kids and I think your kids are probably too young for this, but my kids are all about the games and how they can waste more time, by downloading stupid things to their phone.

I guess the point I wanna give, is that the difference between us and the people who don't have jobs, like kids, is there's a different view of what they can do with applications, and how applications can make their lives better versus ... Like, you and I want to be more efficient, they wanna just waste time and have fun. And I think that culture, going back to your security question, I bring it up because I think that culture is why we're in some of the security concerns that we have today, is that that generation is

growing up not worried about privacy, not worried about security. Just worry about, what can I do to have fun? How can I do to make this more fun for me?

And that group is coming into the work force now, and I think that's ... It's going to be very interesting to see how that mindset that has shifted from, more about efficiency to more about how do I get pleasure in what I'm doing ... It's going to be a major shift over the next decade that I think we should be watching closely.

Niki Acosta: Well, I'll tell you what, if everything is as good as avocado toast, then millennials need to bring it, because I've been on an avocado toast kick. I've been posting pictures of all my avocado toast inventions, and I'm like, "Man! These millennials had it right, like, where has this been my whole life?"

Ken Owens: They have a lot of great ideas, a lot of great things for sure.

Niki Acosta: It's a different perspective, most definitely. I mean, growing up, I don't think I was exposed to technology until I was in my teens, for sure. We didn't have all these cool iPads and games to play with.

Ken Owens: I always joke with my wife, when she was in medical school, I was in engineering school and we would use IOC to chat with each other. And I've explained to my kids what IOC is, it's original instant messaging, except you type something, and you had no idea if it got there. You just waited, and waited, and waited, and about ten minutes later, you'd get something back. And then you would type another message, and you'd wait and wait and wait. You know, five or six text messages would take you an hour because you had no idea if it got there, and you had to wait there forever for a response.

Niki Acosta: You know, one thing that I'm surprised hasn't come back in fashion is walkie-talkies. In a previous life, I worked in construction and we had Nextel phones, and you'd call your builder and say, "Hey, you've got a broken ceiling fan ... " and you'd get, "10-4." That was pretty handy; I guess we have text messaging now, which is a lot less disruptive to the people around you ...

Ken Owens: My kids like using Snapchat. They take a stupid picture and send it to someone. That's the way they communicate now.

Niki Acosta: A [inaudible 00:37:27] communication, which is wild. It's all so wild. I just devised this great plan, speaking of technology. With my kid, I'm trying to get him to learn how to spell stuff, and get better at writing and spelling. So, I was like, "Man, how can I do this over the summer?" So, I had this great plan to set him up with a Minecraft account, and then I'll set up a Minecraft account and join his game and send him messages. He thought it was the coolest thing ever. He was like, "Hey, you wrote me, I'm going to write you back!" But I learned how to type from TelNet chat, so I could probably use this technology to my favor, and to help my kid out a little bit. Oh, Minecraft, how times have changed.

So, switching gears, you obviously talked to a lot of companies, something that you absolutely love doing and I'm sure you talk to people who are ... One of my customers in a topic game called "Old School and New School Folks" and you kind of see where these worlds are colliding, but especially with regards to, I'll say Cisco's traditional customer base, how are people and companies evolving to this Cloud Native model? I mean, they're not just going to get rid of their old stuff overnight, but the evolution is inevitable at this point, I think, and it seems like a lot of companies are struggling with this.

Ken Owens: Yeah, there's a lot of confusion, and a lot of frustration in the companies I talk to because ... I've heard some CIOs and VPs of infrastructure say, "We just have figured

out cloud, and now this how Cloud Native thing is coming." So, it's interesting from that regard.

But, to answer your question more directly, the ... What I'm seeing being successful is starting with the small projects. It's no different than with any time in the past, and how you'd kind of take on new technology. But the difference I'm pushing, and when I'm talking to customers, is when you do this new project, think about what are the people impacts, and what are the process impact, and what are the technology impacts? Because, when you think of new technology, you think of containers, and Kubernetes, and they are new technologies, but you also have an existing new technology that you probably just implemented in your enterprise.

Maybe something like [inaudible 00:39:57] isn't new to everyone, but it might be new to your enterprise. So, how do you deliver a message bus that is taking care of a lot of your enterprise back office stuff, with Kubernetes. How do you integrate those to message and orchestration and solutions simultaneously? Do they replace each other, do they overlap, do you integrate them? So, I try to help enterprise customers understand that it's a journey, and you have to start with helping your people see how they fit into the direction you're going. Helping to figure out which process is going to be impacted by this new model of development that is gonna take over, as you and I both know is taking over already, so it's going to continue to grow.

Probably most importantly, how do you look at the impact it has at your existing technologies, and how do you integrate, or how do you ... Maybe you have a project that is a six month or eight month, I'm going to move off of this SAP Oracle 11 database. How do you take a new project that you know is going to take multiple years, and how do you table that project, and rescope it to be more of a Cloud Native model. What would be the impacts of doing that? What would be the benefits, what would be the drawbacks? What processes would be completely useless when you did that, and what processes would you need to create when you do that?

Obviously, a database that has personal information in it, isn't something that you just haphazardly throw away. There's a lot, when you start thinking about real projects and real aspects of what you're trying to do, it becomes a lot more real to you that just saying, "I'm gonna build a pet store app on my laptop, and see how this technology can help me." No one builds a pet store app on their laptop, so, take something that really is gonna make a difference, and help you understand the impact that it can or cannot have on what you're trying to do.

What I like to tell my kids is, take a tough problem and try to solve it. And when you do that, you may not solve the problem the way you thought were gonna solve it, but you're gonna learn so much about what you're trying to do by taking on a tough problem.

Niki Acosta: It seems like that's a trait of people that I've seen, sort of make these successful career transitions, is that they're the type of people that love to learn and take it upon themselves to learn and to explore, and to try things, and people aren't afraid to break things. If there was one thing that I would want my kid to learn, for sure ... It's not always what you know, it's your passion for knowing. Probably more than anything else.

So, we just blew through forty-five minutes, which is crazy. That was fast. Serverless is coming [crosstalk 00:42:49]

Ken Owens: I can always come back, Niki, I always love talking to you.

Niki Acosta: Yes, I would love you to come back, and we can talk about the serverless thing, that people are just now wrapping their heads around cloud, and now that they've gone from traditional infrastructure to [inaudible 00:43:02] and now they're doing cloud, which still

has servers, but now you're talking serverless, and I think people's heads are exploding. I'd love to read more about that, and have you back on to talk about that.

Ken Owens: Definitely.

Niki Acosta: Where are you gonna be in the next few months? Where can we find you?

Ken Owens: So, I'm going to be at the OpenStack summit, in LA, coming up in a couple months, and then I'll be at ...

Niki Acosta: Wait, OpenStack Summit?

Ken Owens: Open Source summit, sorry.

Niki Acosta: Open Source summit, yes.

Ken Owens: Open Source summit, yeah, in LA.

Niki Acosta: The next OpenStack summit is in Australia, in Sydney.

Ken Owens: Yep! I've been looking at that. But I'll definitely be at the Open Source summit, and I'll be at the Cloud Native CubeCon in Austin later this year.

Niki Acosta: Yay, you'll have to call me! We'll have to hang out!

Ken Owens: Definitely.

Niki Acosta: You can check out DevNet Create. The blog is at Medium.com/atDevNetCreate. You can find Ken on Twitter at KenOwens12. With one 'n' correct?

Ken Owens: Correct.

Niki Acosta: Where else can we find you? I mean, the DevNet site just has a ton of stuff on it, super awesome, and ...

Ken Owens: I'm on LinkedIn at KenOwens12 as well.

Niki Acosta: LinkedIn, who uses that? I'm just kidding, I use it too. It seems like the DevNet folks in general, the one thing I found consistent amongst the DevNet crew is that they're so helpful, like, they're always willing to help. And they usually respond to stuff super fast, which is super handy. Look for Ken online, check out DevNet Creates, subscribe, if you haven't subscribed, we'd love to hear your comments, feedback. If you have an idea for guests, let me know. Hopefully I'll be back with Vallard next week. So, Ken, thank you so much for your time. I hope ...

Ken Owens: Thanks for having me!

Niki Acosta: We'll tune in later, and we'll definitely get you back on the show, Ken.

Ken Owens: Cool, take care.

Niki Acosta: Bye!

Ken Owens: Bye bye.

For More Information

Find more Cisco Cloud Unfiltered podcasts.

Learn more about Cisco Cloud solutions.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Printed in USA 06/17