

EPISODE 764

[INTRODUCTION]

[00:00:00] JM: Virtualization software allows companies to get better utilization from their physical servers. A single physical host can manage multiple virtual machines using a hypervisor. VMware brought virtualization software to market creating popular tools for allowing enterprises to deploy virtual machines throughout the organization.

Containers provide another improvement to server utilization. A virtual machine can be broken up into containers allowing multiple services to run within a single VM. Containers proliferated after the popularization of Docker and the Kubernetes open source container orchestration system grew to be the most common way for managing the large numbers of containers that were running throughout an organization.

As Kubernetes has risen to prominence, software infrastructure companies have developed Kubernetes services to allow enterprises to use Kubernetes more easily. PKS is one example of a managed Kubernetes service. PKS comes out of a joint project between VMware and Pivotal Software. Brad

Meiseles is a senior director of engineering at VMware with more than nine years of experience with the company. He joins the show to discuss virtualization, Kubernetes, containers and the strategy of a large infrastructure provider like VMware.

[SPONSOR MESSAGE]

[00:01:30] JM: DigitalOcean is a reliable, easy to use cloud provider. I've used DigitalOcean for years whenever I want to get an application off the ground quickly, and I've always loved the focus on user experience, the great documentation and the simple user interface. More and more people are finding out about DigitalOcean and realizing that DigitalOcean is perfect for their application workloads.

This year, DigitalOcean is making that even easier with new node types. A \$15 flexible droplet that can mix and match different configurations of CPU and RAM to get the perfect amount of resources for your application. There are also CPU optimized droplets, perfect for highly active frontend servers or CI/CD workloads, and running on the cloud can get expensive, which is why DigitalOcean makes it easy to choose the right size instance. The prices on standard instances have gone down too. You can check out all their new deals by going to do.co/sedaily, and as a bonus to our listeners, you will get \$100 in credit to use over 60 days. That's a lot of money to experiment with. You can make a hundred dollars go pretty far on DigitalOcean. You can use the credit for hosting, or infrastructure, and that includes load balancers, object storage. DigitalOcean Spaces is a great new product that provides object storage, of course, computation.

Get your free \$100 credit at do.co/sedaily, and thanks to DigitalOcean for being a sponsor. The cofounder of DigitalOcean, Moisey Uretsky, was one of the first people I interviewed, and his interview was really inspirational for me. So I've always thought of DigitalOcean as a pretty inspirational company. So thank you, DigitalOcean.

[INTERVIEW]

[00:03:37] JM: Brad Meiseles, you are a vice president at VMware. Welcome to Software Engineering Daily .

[00:03:42] BM: Hello, Jeff. Good to be here.

[00:03:44] JM: You were director of R&D at VMware from 2008 to 2013 and you are currently working on VMware, the container platform, as well as some other products that we'll get into. I want to go back to 2008 to 2013, because I think there are infrastructure lessons we could discuss from that time that apply to today. Back in 2008, 2013 timeframe, how was infrastructure software changing?

[00:04:15] BM: Yeah. Well, that was not quite early days for VMware, but early enough. We'd already been discovered. I joined in 2008 when we had already had our IPO and we're in growth mode, and a lot of what started the company on its success path was essentially the

magic associated with the idea of essentially virtualizing computers and having them run as virtual machines within other computers and in doing like magic things like VMotion. So that's I think what got everybody's attention and imagination going.

But the thing that we discovered relatively early on in the company's history was that magic gets attention, but it's solving real business problems that gets people to use and love your products, and we have been on a journey to do as much as we can to make it easy to operate data centers, and that's where a lot of our energy has gone in terms of monitoring, troubleshooting, solving compliance problems and automation.

[00:05:18] JM: Describe the competitive advantage that VMware developed in the early 2000s.

[00:05:25] BM: Well, I think it was largely a realization that where the very first value proposition for virtualization was this notion of server consolidation and where companies could save money by having fewer physical servers and thus have the decreased cost associated with the hardware itself, associated with electricity and cooling and those sorts of operational expenses.

But the real realization is that the total cost of ownership has a very, very large, often overshadowing by a wide margin those other costs is the human cost. So what you can do to automate processes, reduce the number of touch points needed to satisfy the business needs that those computers are there for is the thing that really gave us the advantage.

[00:06:22] JM: Virtualization software allowed companies to get better economies of scale out of their infrastructure, and that is what you're referring to with what VMware allowed for. More recently, companies have gotten even more economies of scale from containerization. Is the use case for enterprise customers who want a container today, is that the same as an enterprise they wanted a VM back in early 2000s?

[00:06:52] BM: There are certainly a lot of similarities. So with any platform that's going to run applications, whether they're containerized running on Kubernetes, or another container runtime, or you're running virtual machine images on our core virtualization platform, you're going to have a lot of the same benefits around automation that it was talking about earlier.

What I'd say is different as companies move into the containerization world, is that they're extending the need for automation and the benefits of automation from operating their data centers and their applications in production to earlier and earlier in the software development lifecycle. What I mean by that is developer write some code, tests it locally and the process from going from a tested working piece of software on a developers development environment to that software running in production in traditional virtual machine worlds often involve a lot of steps and a lot of people. I think what we're seeing is that automation of those software development lifecycle processes is the next frontier of productivity and costs savings.

[00:08:07] JM: Are there any other lessons from the enterprise adoption of virtualization that can be applied to the adoption of containers and Kubernetes?

[00:08:17] BM: So there's a few things that I think have been really instrumental in the success that VMware has realized in the virtualization of a computer's world, and some of them relate to ecosystem development. So we haven't tried to solve every problem ourselves. In fact, we have been very, very committed to involving all sorts of partners from the hardware level, to system integrators, to other software platforms. I think that commitment to building a community of partners as well as the neutrality that put us in were really successful in our early efforts in the virtualization world.

I think a lot of those same things will translate into the container world with an added twist that in the container world in particular, that ecosystem is not just with partners that we nurture and select, but with the open source community as a whole and so much of the software that is really making a difference in the container world is open.

[00:09:30] JM: Ecosystem can refer to open source projects, can refer to platforms like OpenShift or Mesosphere. It can refer to cloud providers, like AWS, or Google. It can refer to standalone vendors, like Datadog, or Dynatrace. There're so many different individual companies and projects that can be referred to as part of the ecosystem. Maintaining or fostering something like neutrality, that can be hard to do, because you have these different types of players in the ecosystem that have different amounts of resources and represent different competitive threats, maybe not competitive threats, but competitive players. So what is your modern approach to ecosystem management?

[00:10:25] BM: Yeah. I mean, you make some really good points there, and if you think about some of the major partnerships that we've been in the news for recently, you can see that we don't – Just because a company may have some other product or offering in a space that overlaps with ours, that doesn't make us shy away from seeking them as a partner and trying to look for ways to collaborate, and just probably no better examples than Amazon and IBM, and both of those partners have been absolutely instrumental in helping our growth and continue to be, even though we certainly do compete in some areas. I think that's been our strategy from the start and I don't see any change in that.

[00:11:09] JM: Yeah, and I think it makes a lot of sense, because the market is so gigantic. The number of enterprises who are opening up their wallets to buy new container orchestration software, or monitoring tools, or whatever, they're just a lot of dollars and they are going to be turned off by an overall atmosphere of companies not cooperating with each other. So this pretty positive-sum development. I don't know if you would agree with that or maybe you can draw some contrast between less collaborative environments of the past.

[00:11:44] BM: I actually can't think of a scenario where that might be the case.

[00:11:48] JM: So like in the past though, do you feel like things have been like just a more competitive software environment? Like today, like I did a show earlier this week with somebody from Microsoft who does open source in Microsoft. Previously he worked at Google, and the way he framed it is the fact that you have these open source projects that are kind of like a neutral ground for companies to all collaborate, I think that combined with lots of enterprise dollars that want to move into a cloud native environment has led to kind of a more positive-sum environment. But maybe I'm wrong. Maybe it's always been kind of collaborative and positive-sum.

[00:12:27] BM: You're probably right that the open source movement and the adoption of open source by companies that have traditionally been close source, and Microsoft is a great example, and I think the VMware is another great example, where our early success all came from proprietary software and now we collaborate with all those companies including Microsoft on Kubernetes itself and other projects. I think that that does open up more willingness

throughout their entire organizations to collaborate in general and find opportunities to work together. In fact, we had a release of PKS, VMware's Kubernetes offering just earlier this week, and in that one we announced support for Microsoft's Azure platform and we're hoping that that translates into a lot of shared customers between us.

[00:13:18] JM: There have been infrastructure management tools prior to Kubernetes, like OpenStack is an example to people talk about a lot. How did the early days of OpenStack compare to those of Kubernetes?

[00:13:32] BM: Yeah, I think there are a lot of similarities in the early days. They're both platforms that are trying to solve a very broad and wide group of different business problems. There's a lot of different vendors involved in both of those who were looking not only to contribute to the open source project, but also with commercial offerings to try to monetize that open source.

So I think there were significant similarities in the early days of both projects, and comparatively it still is early days for Kubernetes. I guess one big question is will they follow similar trajectories, right?

[00:14:11] JM: Yeah. Do you think that risk of kind of following some trajectory that leads Kubernetes towards a less productive path, do you think that risk has been removed yet or is Kubernetes still at risk of having some kind of issue in losing adoption or fracturing?

[00:14:31] BM: I was not directly involved in a project associated with OpenStack. So I don't have sort of first-hand knowledge to point to what were the inflection points along the trajectory of OpenStack that led it to sort of not realize this potential, because there was a point in time there where you could've have believed that that was going to be the de facto way to manage data centers, and it hasn't happened. So what went wrong?

Rather than comment on that, maybe I can tell you what I think is going well in the Kubernetes open source community.

[00:15:10] JM: Please. Yeah.

[00:15:12] BM: And hope that that continues. Our participation and observation in the Kubernetes community with major players, and you know who all the major players are. Some of whom are very friendly towards each other and some not necessarily. But when you go to the meetings of the leaders of the Kubernetes community, when you go to smaller group meetings in sigs and conferences like KubeCon, what you see is a genuine commitment to try to build the best platform, and I don't see any sort of private agendas being pushed in ways that are contrary to the way a health community should be run. That's certainly our approach towards our participation in the community, and I think there're a lot of signs that that will continue just based on the leadership.

[00:16:06] JM: When did VMware start to evaluate Kubernetes as something to build a product around and how did that evaluation process precede?

[00:16:15] BM: Well, we've had an eye on the development of containers as a way of deploying applications when, I'd say, Docker first re-popularized the concept. Somewhere maybe in 2013, 2014, we started investigating how we should be participating in that ecosystem, and back in that timeframe and then maybe leading all the way up to 2015, it was very sort of neck and neck between whether Docker with Swarm, Mesos or Kubernetes might emerge as the leader. So that was about when we started having Kubernetes on our radar and recognizing its potential.

[SPONSOR MESSAGE]

[00:17:08] JM: Triplebyte fast-tracks your path to a great new career. Take the Triplebyte quiz and interview and then skip straight to final interview opportunities with over 450 top tech companies, such as Dropbox, Asana and Reddit. After you're in the Triplebyte system, you stay there, saving you tons of time and energy.

We ran an experiment earlier this year and Software Engineering Daily listeners who have taken the test are three times more likely to be in their top bracket of quiz scores. So take the quiz yourself anytime even just for fun at triplebyte.com/sedaily. It's free for engineers, and as you make it through the process, Triplebyte will even cover the cost of your flights and hotels for final interviews at the hiring companies. That's pretty sweet.

Triplebyte helps engineers identify high-growth opportunities, get a foot in the door and negotiate multiple offers. I recommend checking out triplebyte.com/sedaily, because going through the hiring process is really painful and really time-consuming. So Triplebyte saves you a lot of time. I'm a big fan of what they're doing over there and they're also doing a lot of research. You can check out the Triplebyte blog. You can check out some of the episodes we've done with Triplebyte founders. It's just a fascinating company and I think they're doing something that's really useful to engineers. So check out Triplebyte. That's T-R-I-P-L-E-B-Y-T-E.com/sedaily. Triplebyte. Byte as in 8 bits.

Thanks to Triplebyte, and check it out.

[INTERVIEW CONTINUED]

[00:18:57] JM: Different companies that have built a Kubernetes offering have built the product in different ways, and like the Kubernetes offering for one cloud provider is different than another. They are very similar and you have this Kubernetes conformant for having a similar Kubernetes type of set of APIs or whatever so you can port your Kubernetes stuff from one cloud to another, for example. But in any case, for VMware, you have a different customer base than the major cloud providers. How did the average customer base of VMware, how did that affect your approach to architecting the PKS, your Kubernetes service?

[00:19:39] BM: So first of all, the conformist test that you're referring to is the [inaudible 00:19:42] test that all of us who have conformant Kubernetes offerings, we run those tests. In our case, we run them daily to make sure that you don't ever come out of compliance. But with each new release, we do that. You're right, there are lots and lots of vendors out there.

What those tests do is within a given cluster, they verify that the major operations that are expected actually run. So that does create this level of portability, which is quite reliable. If your application runs on one instance of Kubernetes, assuming that you're sticking with relatively similar version and underlying capabilities in the hardware, if you're relying on something particular, I guess in that case, then you should be able to run from one to the other.

So when it comes to making an offering like PKS, we look to what are the operational needs of our customers. So if you think back on some of the earlier comments that I made about enterprise customers trying to drive value from a platform, it's important for us to apply some of those learning. So how can we make this easy to operate? How can we make it easy to automate? How can we make it easy to monitor? So those are the areas that we've been investing our engineering effort into.

Another thing that has become really apparent is that many enterprises running Kubernetes view their clusters as ephemeral. When we talk about cattle, the whole pets versus cattle analogy when it comes to your applications themselves and your containers, but a lot of customers are viewing that same paradigm in their clusters themselves where the clusters can come and go. So one of the things that was really important for us to do in the product is to make it very easy to lifecycle manage a large number of clusters and a lot of energy has gone into making that super simple.

[00:21:39] JM: Cattle not pets development, or the way that people are using Kubernetes, that was something that I didn't actually understand until more recently, where just like people treat Docker containers as these dispensable entities that should be stateless and you can have them rotate out or go down or A-B test them anytime, people are looking at their clusters the same way as these dispensable things that can be torn down. Do that surprise you that people started using clusters as cattle not pets?

[00:22:14] BM: Well, it's something that we learned very early on in talking to the early adopters of our product, and it was actually part of our design from the beginning. So we were happy that we had recognized that need early. By the way, it's not like every cluster is cattle. If it's a production cluster that has stateful sets, you might not think of it that way.

But where analogy applies is that if your Kubernetes – Whatever your Kubernetes platform makes it easy, why not spin up clusters for each developer to work in a sandbox. Why not spin up clusters for each time that you're running your test suite? Why not spin up clusters every time there's a new version of Kubernetes and make sure there're no incompatibilities with your apps? So for all those reasons, we see it as being a widely used pattern and with a lot of

benefits as long as it doesn't require a whole lot of operational effort to get them to come and go.

[00:23:13] JM: For customers who are already on VMware virtualization, did you want to provide some kind of on-ramp for them to easily go from virtualization to containerization?

[00:23:29] BM: There is a term that's been thrown around quite a bit around the container space, which is this notion of lift and shift or some people say shift and lift, but that's the notion of saying, "I've got an application already running in VM's, and I want to start running it in containers and attempt to realize some of the benefits of containers without having to re-architect an app," and I think that's what you're referring to.

[00:23:55] JM: Also, actually, I should have been more clear, but most are referring to the fact that you can take your VM's and break them up into containers and get more economies of scale that way, but I guess I'm just trying to understand the customer a little bit better in terms of what they want out of a Kubernetes solution.

[00:24:12] BM: Yeah. So I'd say there is a little bit of this lift and shift activity where there is an existing app with or without re-factoring, where the idea of portability and maybe the integration with CI/CD pipelines will actually provide a benefit without a major architecture. There's definitely a use case for that. In fact, some commercial software offerings are now being made available not just as a virtual machine images, but also as container images. So there's that way in which Kubernetes can be leveraged.

But I think the vast majority of at least some of the early adopters of Kubernetes are actually building new applications and trying to architect them following the cloud native principles that they've been reading about and trying to get the full benefit.

[00:25:05] JM: Right. This is one of these trends that's kind of interesting that it's not just the adoption of public cloud or hybrid cloud that is increasing. There's also just the expansion of on-prem deployments, the expansion of deployments everywhere, because people aren't shutting down their old applications. They're deploying more applications alongside their existing applications.

[00:25:33] BM: Yeah, I think that's right, and with any application – And I'm an engineering manager, so I have to think about this a lot in my own job and not just what our customers are doing, re-architecture is expensive and you better be sure that you're going to get the benefit from it if you're going to embark on that kind of a project versus just adding new features and value that is in your backlog.

So I think our customers are going through that same equation, and in some systems really should be re-architected because the velocity that their engineering teams are able to deliver is unacceptably slow. That's when it's justified. So I think that that's the same decision process that we all go through, and if you're going to re-architect, you may as well leapfrog into the cloud native world and try to reap the benefits of scalability and more automation.

[00:26:26] JM: I will say I would much rather be on a project that is kind of a greenfield just deploying a new application alongside an existing application. But that said, I completely agree. There are times where you just have to refactor. My sense of talking to people is that there is I guess a growing tolerance of, "Okay, we've got this piece of software, and maybe it's not built bill perfectly, but we need to move on to build us new features, and so we're just going to leave this old piece of software as it is and just kind of let it continue to run and build some new stuff."

But is that leading to an IT sprawl where you just have these old applications and then just continued expansion and sprawl of new applications to the point where nobody really knows like where everything is running and this becomes a sprawling problem?

[00:27:24] BM: I think you've hit on a need in the market for management, right? So they say that every organization, at least any reasonably sized organization, would already view their application portfolio as having sprawl today. So it's a problem that's already endemic. So I don't think that that necessarily factors into the decision of what to do with any particular app.

[00:27:52] JM: How are the buying patterns of enterprises changing in this cloud native world?

[00:28:00] BM: So when it comes to making a platform decision, that's something that's going to be impactful, because you're not just choosing something that's going to impact usually a

small number of people. It could impact the entire enterprise. All those being equal, you'd rather have a single platform. Platforms, they often come with a learning curve. They are often difficult to operationalize. So it's a big decision.

The way companies are evaluating their choices in container platforms, I think isn't that much different from the way any other platform decision has been done historically. Find a few choices that you think satisfy your core needs from a functional standpoint and make sure you're working with a vendor that you trust and that has a track record of being reliable and has a track record of innovating so that you don't get stuck behind it.

[00:29:02] JM: If I'm a large enterprise like a bank or an insurance company, am I looking for just one platform provider to help me deploy Kubernetes or have a modern platform, or am I saying, "Let's just have carte blanche and one part of the bank is going to have one platform as a service, another part of the bank has a different platform as a service." Are most of the organizations you're talking to, are they only looking for one of these platforms?

[00:29:32] BM: Let me just go back to something that we've touched on earlier, which is one of the benefits of Kubernetes as a platform is that workloads are portable from one Kubernetes cluster to another and almost always from one version to Kubernetes to another unless something significant has changed, there's a huge gap.

So whether you got your cluster through just installing the open source project yourself or through a vendor or through a cloud provider, ideally the Kubernetes control plane itself will run your app. So I think that has led to there being less of an imperative that we must have only one, and large organizations such as the ones you mentioned, banks, and insurance companies, etc., they may prefer to have one, but it's often impossible. Different groups have a lot of autonomy and they have some business imperative for which they must act fast and coordinating a purchase or a decision across an entire enterprise can be very difficult.

So I think that the reality is that many companies find themselves with more than one even though they didn't specifically seek more than one, because they do view the ability to run their apps as relatively uniform. That said, I do think that over time we'll see that the operational cost

of having more than one will eventually become a large enough problem that companies do want to solve it in and maybe consolidate on the one that's giving them the best results.

[00:31:05] JM: What about the cloud provider conversation? You see lots of enterprises that are starting to play with Cloud providers. They're increasingly opening up their wallets to public cloud providers, like AWS, and they want to be able to work with some of the cloud provider services, but they also want privately hosted on-premise Kubernetes. They want to be able to take advantage of the data centers that they've hardly invested in. How are you seeing companies that want some exposure to the cloud managing their Kubernetes platform selection?

[00:31:50] BM: Yeah. So what you're touching on has already happened to virtualization in the VM world, which is that hybrid is basically the way that companies are going to run for the foreseeable future. There was a point in time where cloud was just emerging and then there was this period of time where the narrative was the data center is dead and cloud will rule. What VMware has predicted and what our strategy has moved towards is that hybrid is actually the right choice, because there are economies of scale that can be achieved on-prem and there are data protection benefits from having an on-prem data center. But for many reasons, it's good to have cloud presence as well.

So this whole hybrid approach I think is here to stay, and that's true for virtual machines, and I think it's going to be true for container platforms as well. Even in public cloud, there's quite a lot of choice. Major cloud providers all have their own Kubernetes services. VMware has cloud PKS. So we have PKS for on-prem and we also have cloud PKS, and I think there are vendors that can deliver Kubernetes clusters on public clouds as well. So I do think that we're going to see a hybrid footprint for Kubernetes for a lot of the same reasons that we see it for virtual machines.

[00:33:23] JM: The VMware cloud PKF, is that of version of your Kubernetes service that gets deployed to cloud, like clouds like Amazon or Azure?

[00:33:35] BM: Yeah, that's exactly right. So we offer a SaaS service, where through a user interface and an API you can spin up clusters in public clouds.

[00:33:48] JM: Okay. If I'm an enterprise, like a bank, maybe I want to spin up a cluster on AWS so that my cluster is closer to some AWS managed service? If I'm a developer at one enterprise like a bank, why am I choosing to – If I've got PKS deployed on the cloud and also on-prem, how am I choosing where to spin up a cluster? Why does that matter to me?

[00:34:21] BM: Well, we talked earlier about all the different reasons why companies will have multiple clusters. So there are many different lines in which that decision can be made. One very simple one is, well, our production applications run on-prem and that's where we have absolute control over access to the data, etc., and that's our comfort zone. But for development and testing purposes, we like the ephemeral nature of the resources. The pay for what we use nature of resources and that is a perfectly fine workload to run in cloud. That's one possible line.

Another is burst capacity, where you operate your on-prem clusters at some sustainable utilization, but you don't want to go, let's say, about 70% utilization on CPU for your hardware or some measure like that. During peak period, you actually want to stretch your services and have some of them run in burst capacity in cloud. So those are two examples, and I think there are others. But there will be lots of reasons for which people will select one or the other or both.

[SPONSOR MESSAGE]

[00:35:40] JM: OpenShift is a Kubernetes platform from Red Hat. OpenShift takes the Kubernetes container orchestration system and adds features that let you build software more quickly. OpenShift includes service discovery, CI/CD built-in monitoring and health management, and scalability. With OpenShift, you can avoid being locked into any of the particular large cloud providers. You can move your workloads easily between public and private cloud infrastructure as well as your own on-prem hardware.

OpenShift from Red Hat gives you Kubernetes without the complication. Security, log management, container networking, configuration management, you can focus on your application instead of complex Kubernetes issues.

OpenShift is open source technology built to enable everyone to launch their big ideas. Whether you're an engineer at a large enterprise, or a developer getting your startup off the ground, you can check out OpenShift from Red Hat by going to softwareengineeringdaily.com/redhat. That's softwareengineeringdaily.com/redhat.

I remember the earliest shows I did about Kubernetes and trying to understand its potential and what it was for, and I remember people saying that this is a platform for building platforms. So Kubernetes was not meant to be used from raw Kubernetes to have a platform as a service. It was meant as a lower level infrastructure piece to build platforms as a service on top of, which is why OpenShift came into manifestation.

So you could check it out by going to softwareengineeringdaily.com/redhat and find out about OpenShift.

[INTERVIEW CONTINUED]

[00:37:48] JM: When you're thinking about architecting PKS, do you try to keep your service just kind of at parity with Kubernetes services that are on other platforms or are there ways in which you can differentiate or want to differentiate? I can almost imagine not even wanting to differentiate, because you just want to kind of keep pace with the market and make sure you're compliant and make sure you have optionality. But then again, maybe you do want to differentiate in certain ways. Do you want to differentiate as a Kubernetes platform provider?

[00:38:28] BM: The area in which we focus a lot of our energy is on the operations of the Kubernetes clusters themselves, like let's make it easy to operate and so dev ops teams can have essentially a completely reliable service on which to build their apps. There's a lot that we do in that area to really reduce all of the effort required from an operations front, and we've touched on a few of these things. There are certainly monitoring. What's the health of my clusters? There is a watching utilization and scaling as necessary cluster itself, I mean. Then there is the management of network access and isolation that we automate when a cluster is created. We automate all the steps necessary to make sure that you have ingress north-south network connectivity as well as east-west network connectivity within the cluster. So there's a large surface area of need there that we're working to address.

Now, you were alluding to services, and I think you're alluding to services that applications want to leverage. So maybe a database service, a storage service, a message bus service, that sort of thing, and we are developing a marketplace, a partner network in which the vendors who are specialists in those areas could make their offerings available on our platform. In public cloud, there is latching on to the service broker mechanism to enable clusters deployed on our platform or applications running on those clusters to leverage services offered by the cloud provider which they're running.

[00:40:14] JM: That's the open service broker project?

[00:40:17] BM: Yeah, that's right.

[00:40:18] JM: Can you explain more what that project is doing?

[00:40:22] BM: So when an application wants to make use of a service, the key things that that application needs is to how do I reach that service and how do I fit authenticate with that service? Those are the two key areas of functionality that the open service broker API attempts to solve.

[00:40:42] JM: And does that refer to like managed services, like things that you're buying from a cloud provider, or is it referring to like a service that I've built myself?

[00:40:53] BM: So it could be either, and I think in most cases it's both. But typically we're talking about services that are running outside of the cluster, because within –

[00:41:00] JM: Okay.

[00:41:01] BM: Yeah. So if it's a service that you've built yourself that's running inside the cluster, you don't need to use that broker. Kubernetes itself has the mechanism with which to discover and leverage microservices running within the cluster that you're running on.

[00:41:16] JM: And so the service broker, does that define protocols, or can you talk in more detail what exactly it's defining for me?

[00:41:24] BM: Well, it gives you a way to discover which services are available. It's certainly API-driven. So it is an API, and you can say, "Which services are available? Okay, here's the one that I want to leverage. What's the endpoint that I could reach it at and here are the credentials that I'm allowed to use to access it?" Those are the general functionality that it provides you.

[00:41:47] JM: Another aspect of networking on Kubernetes is the service mesh. This is the idea that you have sidecar containers deployed next to some of your services, and then you have a control plane that allows you to define network policy, define routing, define authentication and security stuff essentially. What was your approach to the service mesh category and how does that affect the architecture of PKS?

[00:42:23] BM: Yeah. So there are a few service mesh offerings out there and I think that it is an area that's likely to continue to gain traction. At this point I'd say there is a lot more mind share associated with service mesh than there is actually utilization. There're a lot of great success stories with it. Our approach is to first of all make sure that the widely used service mesh options, and Istio seems to be emerging as the leader there, run smoothly on all of our Kubernetes offerings. The beauty is as we talked earlier, Kubernetes does present a conformant control plane and the service meshes themselves are able to run without any huge heavy lifting on the part of the people operating the clusters. So that's great.

We've also introduced a product called the NSX service mesh, and one of the really neat things that service mesh enables is this not just discovery of services, but also traceability, and we see one of use cases that we talked about earlier was this idea of maybe burst capacity where you're stretching a service from on-prem to cloud or maybe it's two different on-prem, it doesn't really matter, clusters. How do you then make sure that your service has a common identity across those clusters and that you could trace its usage across those clusters? And those are the sorts of problems that we're looking to solve with NSX service mesh.

[00:43:57] JM: You and I met and KubeCon in China. What was your experience talking to Kubernetes users in China?

[00:44:08] BM: First of all, that was a phenomenal event, and I've been going to China regularly because we have a very reactive engineering group there. I've seen the adoption of Kubernetes in China at least as strong as it's been in the U.S., and it's great to see that community now starting to participate more in the open source projects, Kubernetes itself and related ones.

[00:44:35] JM: Did you have any interesting conversations that stood out to you in KubeCon China, or I don't know if you went to the most recent KubeCon? I find those conferences pretty interesting, because there's a lot of stuff that gets said in between talks and outside of keynotes and whatnot. I'm just wondering if there's anything I can mine from your falling through the cracks gathering of information.

[00:44:59] BM: Yeah. One of the things that really surprised me is how many public cloud offerings there are in China. By the way, I didn't go to KubeCon in Seattle this time. I got KubeCon out in Shanghai, but I was just amazed that as I met people and, "What do you do?" "Oh! I work for this startup that's a cloud provider and we have a Kubernetes offer." So I think that that's just such a growth area in that country. It's really quite amazing.

[00:45:28] JM: There are enterprises today that are completely overwhelmed with all the options in the cloud native world. Do you have any advice for these kinds of enterprises on how to approach the market?

[00:45:44] BM: Well, so I guess one thing to think about, there're so much mind share and attention being paid in the media to Kubernetes that I think most CIOs, the decision-makers, are saying, "Gee! I'd better get me a Kubernetes. Otherwise I'm not keeping up with the times." I'd say don't get Kubernetes just because you think it's the trend that you need to follow. Find a real business problem where taking an existing application and re-factoring it or building a new application in a cloud native way actually has a real business benefit and have that business object to be the driver for adopting Kubernetes or probably any new technology for that matter.

[00:46:31] JM: VMware acquired Heptio a while ago. Have you gotten any insights about your Kubernetes strategy from the Heptio acquisition?

[00:46:41] BM: Well, first of all, so much of what we had been doing pre-acquisition is so well-aligned with the division that Heptio has. As you probably know, the two founders of Heptio, Craig McLuckie and Joe Beda, are two of the three founders of Kubernetes itself back when they worked at Google. So these guys have a great vision through where do they see their baby going, and also they're amazed at where it's gone already. But the real delight in our discussions with them leading up to the acquisition is just how well-aligned our visions were in terms of what the potential is for this technology and how we can help our customers continue to innovate and really get the best out of it.

[00:47:28] JM: What's on the roadmap for PKS in the near future?

[00:47:32] BM: We talked about the areas in which we want to differentiate, and I think we have a long way to go before we realized the full potential of just the making things simple and have clusters that are completely essentially autonomic, easy to operate. So we're going to continue working on that. I also think we touched on this notion of hybrid as well and how it's not right to have just one big cluster. There're lots of reasons why multiple clusters are desirable and multiple clusters in different places. So I also see a big opportunity for coming up with technologies that enable our customers to make best use of that hybrid footprint, and we talked about our service mesh approach as one example of that, and I think there's plenty more that we could do in that realm.

[00:48:22] JM: Pivotal and VMware worked together on PKS, and Pivotal and VMware have some shared lineage and some shared, I guess, ownership. I think Dell owns both VMware and Pivotal these days. What was the process for collaborating with Pivotal on taking PKS to market?

[00:48:45] BM: Yes, this has been the true partnership with Pivotal across the entire product development process. I mean, we started by getting some of our experienced engineers. We had a lot of context. Pivotal has a lot of history. Maybe one of the organizations with the most history on what it means to build cloud native apps and containerized workloads the cloud

foundry offering. So we brought some of their leaders together with ours and we shaped the product together. We co-develop it. Our engineers actually work in the same facility and we market it together and we actually sell it together too. So every facet of this project has been a collaboration.

[00:49:28] JM: Okay. Well, Brad, it's been really great talking to you about VMware and PKS, and I am looking forward to seeing what else develops out of VMware in the future.

[00:49:38] BM: Thank you very much. It was a pleasure.

[END OF INTERVIEW]

[00:49:43] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source, it's free to use, and GoCD recently launched a test drive service that makes it easier than ever to try out GoCD. You can go to go.cd/test-drive-go.cd.

If you've been wondering about what continuous delivery tool you should use for your cloud native software, GoCD is worth checking out, and now it's easier than ever to just try it out and see if this looks like something that you would want. Just go to go.cd/test-drive-go.cd and find out how GoCD fits your workflow.

GoCD has support for Kubernetes and it was built with the learnings of the ThoughtWorks engineering team. If you want to try it out, go to go.cd/test-drive-go.cd.

[END]